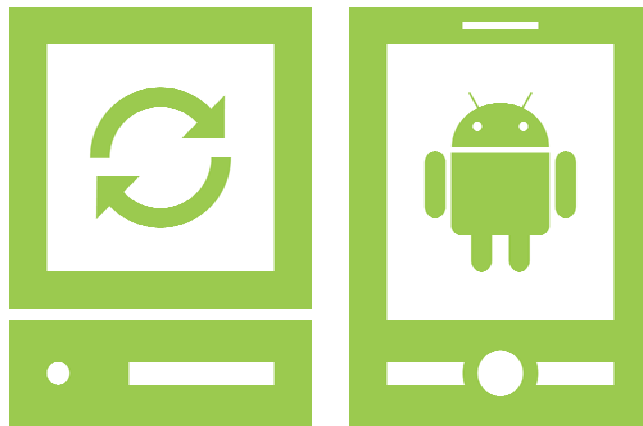


Task-Management Control-Center for Android

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil



Frühlingssemester 2011

Autoren: Daniela Meier, Ramona Rudnicki
Betreuer: Thomas Letsch
Projektpartner: Letsch Informatik, Wald ZH
Experte: Prof. Dr. Martin Zimmermann
Gegenleser: Oliver Augenstein

Inhaltsverzeichnis

Aufgabenstellung	1
Erklärung über die eigenständige Arbeit	2
Vereinbarung über Urheber- und Nutzungsrechte	3
Abstract	4
Vision	5
Projektplan	6
Risikomanagement	7
Arbeitspakete	8
Anforderungsspezifikation	9
Domainanalyse	10
GUI-Design	11
Software Architektur Dokument	12
Design Dokumentation	13
Testdokumentation	14
User Guide	15
Glossar	16
Literaturverzeichnis	17
Poster	18
Technischer Bericht	19
Persönliche Berichte	20
Zeiterfassung	21

Task-Management Control-Center for Android (TCA)

Studenten

- Daniela Meier
- Ramona Rudnicki

Einführung

Android (<http://www.android.com>) ist ein Betriebssystem sowie auch eine Software-Plattform für mobile Geräte wie Smartphones, Mobiltelefone und Netbooks, welches von der Open Handset Alliance entwickelt wird.

In vorangegangenen Studienarbeiten wurden für Android zwei Frameworks erstellt:

- Task-Management-Framework on Smart-Phone (TaMaF) [1]
Ein Framework zur Erstellung von Android-Applikationen im Bereich des Task-Managements.
- Android Control Framework (AnCoF) [2]
Ein Framework, welches den Zugriff auf bestimmte Funktionen des Android-Systems ab einem PC erlaubt.

Aufgabenstellung

Es soll eine Applikation erstellt werden, welche einem Benutzer auf einem PC die Funktionalitäten der beiden Frameworks "Task-Management-Framework TaMaF" und "Android Control Framework AnCoF" mittels einem GUI zur Verfügung stellt.

Die Applikation muss in Java mit Swing erstellt sein und unabhängig vom darunter liegenden Betriebssystem funktionsfähig sein (Windows, Linux, Mac OS).

Spezielle Anforderungen welche zu berücksichtigen sind:

- grosse Datenbestände
- permanente Synchronisation mit einer PC-basierter Datenquellen resp. -senken
- schnelle und effiziente Bedienung

Unter Berücksichtigung von aktuellen Software-Engineering-Methoden soll ein geeigneter Entwicklungsprozess definiert und darauf basierend die Applikation entsprechend gebaut werden.

Technologien

- Android
- Java/Swing
- Eclipse
- Enterprise Architect

Generelles

- Die Vorgaben der Abteilung Informatik [3] sind einzuhalten.
- Die "Generelle Richtlinien für Studien- und Bachelorarbeiten" [4] sind einzuhalten.
- Mit dem CASE-Tool Enterprise Architect ist ein UML-Modell zu führen, welches synchron mit den Programm-Sourcen und der Projekt-Dokumentation ist.
- Ein Java-Entwickler muss mit der Projekt-Dokumentation in die Lage versetzt werden, die Applikation in Betrieb zu nehmen und weiter entwickeln zu können.

Termine

- Montag, 21.02.11 Beginn der Studienarbeit
- Donnerstag, 17.06.11 12:00 Uhr Abgabe der Studienarbeit

Betreuung

Thomas Letsch
tlletsch@hsr.ch

055 - 22 24 567 (HSR Büro 5.204); 055 - 214 43 50 (Geschäft)

Referenzen

- [1] Studienarbeit „Task-Management-Framework on Smart-Phone (TaMaF)“, 17.12.2009
- [2] Studienarbeit „Android Control Framework (AnCoF)“, 23.12.2010
- [3] www.hsr.ch>HSR-intern>Bachelor-Studiengänge>Informatik>Allgemeine Infos Diplom-, Bachelor- und Studienarbeiten
<https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>
- [4] "Generelle Richtlinien für Studien- und Bachelorarbeiten" (v1.4 / 19.09.2010, Thomas Letsch)

Rapperswil, 21.Februar 2011



Thomas Letsch

Task-Management Control-Center for Android

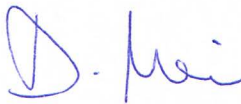
Erklärung

Wir erklären hiermit,

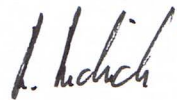
- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt haben, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde,
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.

Ort, Datum:

Rapperswil, 9.6.2011



Daniela Meier



Ramona Rudnicki



Daniela Meier



Ramona Rudnicki

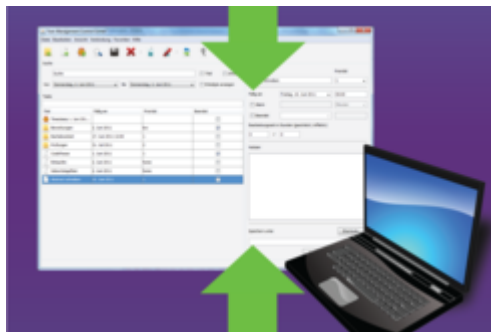
Diplomanden	Daniela Meier, Ramona Rudnicki
Examinator	Thomas Letsch
Experte	Prof. Dr. Martin Zimmermann, Hochschule Offenburg, D-77723 Gengenbach
Themengebiet	Software
Projektpartner	Letsch Informatik, Wald, ZH

Task-Management Control-Center for Android TCA

Interoperables Task-Management zwischen beliebiger Java-Plattform und Android



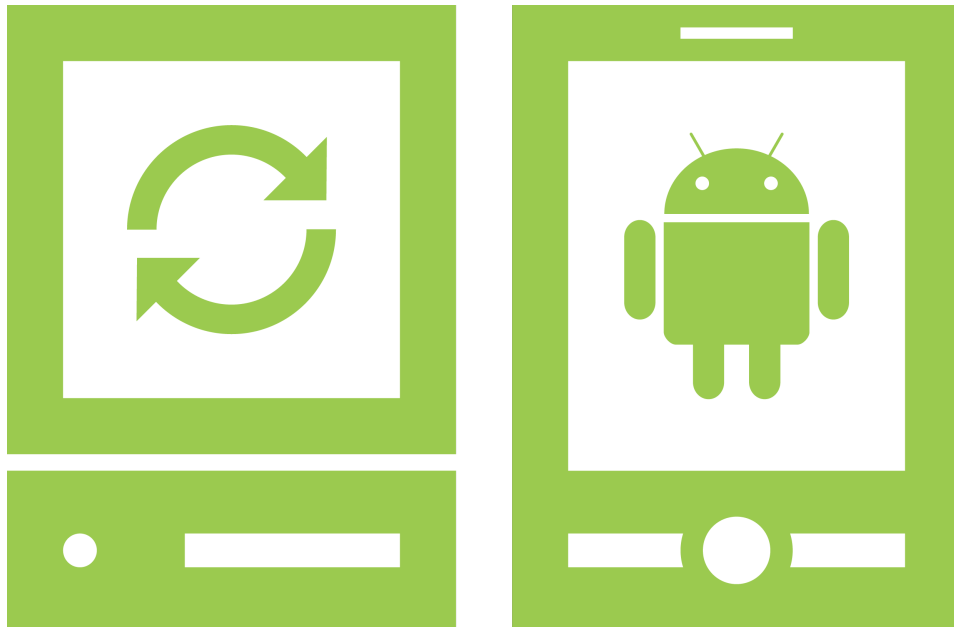
Ausgangslage: Notizen oder Aufgaben, welche auf einem Android- Gerät erstellt werden, können zurzeit nur umständlich mit dem PC ausgetauscht werden, geschweige denn mit einem Firmennetzwerk synchronisiert oder effizient durchsucht werden. Basierend auf dem "Android Control Framework" (AnCoF) soll eine Oberflächen-Applikation auf PC- sowie Mobile-Seite geschaffen werden, welche es erlaubt, Dateien und Notizen zu verwalten und zu synchronisieren. AnCoF ist ein Framework für Android-Geräte, das Remote-Zugriff auf ausgewählte Funktionen des Android-Systems bietet und damit die Realisation gängiger Aufgaben einer PC-Suite ermöglicht.



Vorgehen/Technologien: Um Notizen verwalten zu können, müssen diese erst gespeichert werden. Für die Persistierung der Notizen wurde auf eine SQLite Datenbank zurückgegriffen. Sind die Notizen gespeichert, müssen diese zwischen den Geräten ausgetauscht werden. Dieser Austausch, welcher der Synchronisation dient, wurde über die bereitgestellten Schnittstellen von AnCoF realisiert. Die Dateien werden ebenfalls mit Hilfe von AnCoF synchronisiert. Die Benutzeroberflächen wurden auf der PC- Seite in Java respektive Swing implementiert und auf der Android-Seite wurde eine android-spezifische Kombination aus Java (für die Logik) und XML (für die Darstellung) verwendet. Aufgrund der unterschiedlichen Voraussetzungen von Android- und Java-Code wurde die Arbeit in zwei Projekte aufgeteilt. Das eine Projekt realisiert die Implementation der Android-Oberfläche und das andere jene auf dem PC.



Ergebnis: Als Lösung ist das "Task- Management and Control- Center for Android" (TCA) hervorgegangen. Auf der PC- sowie der Mobile-Seite können über eine Benutzeroberfläche Notizen verfasst, aktualisiert und gelöscht werden. Der Übersichtlichkeit wegen können diese mittels Ordnern strukturiert werden. Durch die automatische Synchronisation beim Verbinden des PCs mit dem Mobile erlaubt TCA, jederzeit die aktuellste Version auserwählter Dateien und aller Notizen auf dem entsprechenden Gerät abzurufen. Um zu bestimmen, welche Dateien jeweils synchronisiert werden, können die Pfade zu den jeweiligen Ordnern in einer Datei festgelegt werden. Der Austausch von Notizen mit einem Firmennetzwerk wurde bereits analysiert und geplant, jedoch als Erweiterung für einen nächsten Release aufgeschoben.



Vision

«Task-Management Control-Center for Android»

Version 1.2, 9. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Projektvision	2
2.1 Daten	2
2.1.1 Spezielle Aufgaben	2
2.2 Datenbearbeitung	2
2.2.1 Verbindung	2
2.2.2 Statusabfrage	2
2.2.3 Synchronisation	3
2.2.4 Suche	3
2.2.5 Sicherheit	3
2.3 Darstellung	3

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 6

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Projektvision

TCA bietet eine PC- sowie eine Mobile-Device-Applikation, welche es ermöglicht, Dateien und Tasks zu verwalten respektive diese in einem synchronen Zustand zu halten. Über eine Schnittstelle ist es möglich, Dateinamen und Tasks auch mit anderen Netzwerken, z.B. einem Firmennetzwerk, auszutauschen. Die Benutzeroberfläche ist einfach und übersichtlich gestaltet, damit die Aufgaben effizient erledigt werden können. Nachfolgend werden die Aufgabenbereiche dargestellt, welche zu erfüllen sind.

2.1 Daten

TCA muss mit grossen Datenbeständen zurechtkommen. Diese gliedern sich in Tasks (Aufgaben oder Notizen) und Dateien auf. Tasks müssen in einer Baumstruktur in Tasks und «Subtasks» verschachtelt werden können.

2.1.1 Spezielle Aufgaben

Es soll ermöglicht werden, nebst Standardaufgaben wie Datei- und Tasksynchronisationen auch Audiofunktionen und Zeitstempel zu handhaben.

Audioaufnahmen und -wiedergaben

Es sollen Audiodateien verwaltet werden können. Diese erlauben es, Kurznotizen in digitaler Form aufzunehmen und diese dann auf der PC-Seite abzuhören. Wenn möglich, sollten bereits bestehende Programme für die Audiofunktionen genutzt werden.

Timestamps

Über diesen Task kann ein Zeitstempel erfasst werden. Dies ist ein Task mit minimalem Inhalt, das heisst, er beinhaltet maximal eine Kurznotiz. Zudem sollte der Zeitstempel sehr einfach («mit einem Klick») zu erfassen sein.

2.2 Datenbearbeitung

2.2.1 Verbindung

Für eine ständige Synchronisation der Dateien und Tasks mittels TCA wird eine aktive Verbindung zwischen Mobile-Device und PC benötigt. Möchte man das Mobile-Device vom System trennen, so muss dies mit oder ohne Dateisynchronisation möglich sein, da unter Umständen eine vollständige Synchronisation aller Dateien zu lange dauern würde.

2.2.2 Statusabfrage

Es muss möglich sein, via Telefon einen (z.B. Server-) Systemstatus abzufragen. Die aktive Verbindung soll nach einem zuvor festgelegten Zeitraum automatisch getrennt werden.

2.2.3 Synchronisation

Beim Verbinden des PCs mit dem Mobile-Device werden Dateien und Tasks in beide Richtungen synchronisiert. Danach sollen Tasks permanent vom PC zum Mobile-Device synchronisiert werden, Dateien hingegen nur auf Anfrage. Wünschenswert wäre eine rekursive Synchronisation. Die synchronisierten Dateien beziehungsweise Tasks müssen auch an der Schnittstelle für andere Netzwerke bereitgestellt werden.

2.2.4 Suche

Die vorhandenen Tasks (Tasknamen und Inhalt) müssen effizient durchsucht werden können. Auf der PC- wie auch der Mobile-Device-Seite soll eine inkrementelle Durchsuchung aller Tasks oder auch nur eines Teilbaums ermöglicht werden, welche es erlaubt, die Datenbestände mittels Regular Expression (RegEx) abzufragen. Die Ergebnisse können zur Übersicht zusätzlich in ihrer hierarchischen Struktur, als Baum dargestellt, angezeigt werden. Weiter muss auch der Inhalt der Tasks inkrementell durchsucht werden können.

2.2.5 Sicherheit

Um die Sicherheit beim Verlust des Mobile-Devices zu gewährleisten, müssen alle Tasks verschlüsselt abgelegt werden. Keinesfalls sollten diese im Klartext einsehbar sein. Bei Dateien hingegen kann die Verschlüsselung delegiert werden und ist nicht Aufgabe von TCA. Aus Performance-Gründen findet die Verschlüsselung auf der PC-Seite statt.

2.3 Darstellung

Wie bereits erwähnt, sollte das GUI möglichst einfach und intuitiv zu bedienen sein. Die Darstellung der Tasks (z.B. Schriftgröße, Hintergrundfarbe) muss konfigurierbar sein.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	24.02.2011	DM	Erster Vorschlag für Vision erstellt.
1.0rc03	25.02.2011	RR	Kleinere Inhaltliche Änderungen und Rechtschreibung angepasst.
1.0rc04	03.03.2011	DM	Anpassungen gemäss Sitzung vom 03.03.2011.
1.0rc05	10.03.2011	DM	Anpassungen gemäss Sitzung vom 10.03.2011.
1.0	10.03.2011	DM	Akzeptierte Version 1.0.
1.2rc01	09.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.2	09.06.2011	RR	Akzeptierte Version 1.2.

Tabelle 2.1: Revisionshistorie

Abkürzungsverzeichnis

Ba Bachelorarbeit

GUI Graphical User Interface

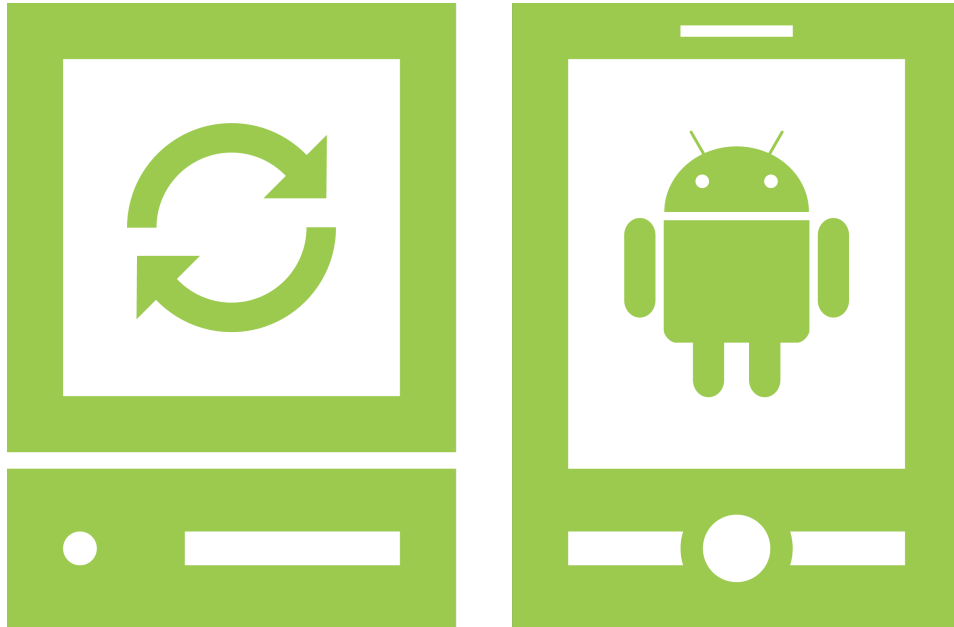
PC Personal Computer

RegEx Regular Expression

TCA Task-Management Control-Center for Android

Tabellenverzeichnis

2.1 Revisionshistorie	4
---------------------------------	---



Projektplan

«Task-Management Control-Center for Android»

Version 1.4, 13. Juni 2011

Daniela Meier (d2meier@hsr.ch)

Ramona Rudnicki (rrudnick@hsr.ch)

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Projektübersicht	2
2.1 Ziel und Zweck	2
2.2 Vorgehen	2
2.3 Annahmen und Einschränkungen	2
3 Projektorganisation	3
3.1 Organisationsstruktur	3
3.2 Externe Schnittstellen	3
4 Managementabläufe	4
4.1 Projektdauer	4
4.2 Projektplan	4
4.2.1 Zeitplanübersicht	4
4.2.2 Beschreibung der Artefakte	6
4.2.3 Iterationsplanung	9
4.2.4 Releases	10
4.2.5 Code Reviews	10
4.3 Besprechungen	11
5 Risiko-Management	12
6 Arbeitspakete	13
7 Infrastruktur	14
7.1 Räumlichkeiten	14
7.2 Hardware	14
7.3 Software	14
7.4 Kommunikation	15
8 Qualitätsmassnahmen	16
8.1 Dokumentation	16
8.1.1 Allgemein	16
8.1.2 Sitzungsprotokolle	16
8.1.3 Zeitplan	16
8.2 Style Guides	16
8.2.1 Code	16
8.2.2 Dokumente	17
8.3 Reviews	18
8.3.1 Dokumentations-Reviews	18
8.3.2 Code Reviews	18
8.3.3 Iteration Assessments	18

8.4	Versionsverwaltungssystem	18
8.5	Tests	18
8.5.1	Unit Tests	18
8.5.2	Systemtests	18
8.6	Logging	19
8.6.1	Loglevels	19

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 6.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Projektübersicht

2.1 Ziel und Zweck

TCA basiert auf dem Android Control Framework (AnCoF) und erweitert dieses um eine Oberflächen-Applikation auf PC- und Android-Seite. Diese ermöglicht es, Dateien und Notizen zu verwalten und zu synchronisieren. Über eine Schnittstelle können diese zusätzlich mit einem Firmennetzwerk ausgetauscht werden. TCA erlaubt es, jederzeit die aktuellste Version ausgewählter Dateien und Notizen auf dem jeweiligen Gerät abzurufen.

Weitere Details können dem Dokument «Vision» entnommen werden.

2.2 Vorgehen

Die Vorgehensweise entspricht dem Unified Process (UP), wie er in [2] beschrieben wird.

2.3 Annahmen und Einschränkungen

Die Soll-Arbeitszeit pro Projektmitglied und Woche entspricht in etwa 23 Stunden. Details können dem Kapitel 4.1, Seite 4 entnommen werden. Längere Absenzen sind keine zu erwarten. Bei Ausfällen wegen Krankheit oder Unfall wird der Zeitplan angepasst beziehungsweise die Arbeitszeiten erweitert.

3 Projektorganisation

3.1 Organisationsstruktur

Tabelle 3.1 zeigt die am Projekt beteiligten Personen und ihre Zuständigkeitsbereiche.

Bereich	Verantwortlich	Artefakten
Projekt Management	Daniela Meier (DM)	Projektplan (Pp)
Requirements und Analyse	DM	Anforderungsspezifikation (As), Domainanalyse (Da)
Design	Ramona Rudnicki (RR)	Software Architektur Dokument (SAD), Design Dokument (DD), Designmodell
Implementation	RR	Prototyp, Code, Releases
Testen	DM	Unit Tests, Systemtests
Infrastruktur	RR	Siehe Kapitel 7, Seite 14
Weitere Dokumentation	DM	Weitere, oben nicht aufgeführte Dokumente

Tabelle 3.1: Organisationsstruktur

3.2 Externe Schnittstellen

Thomas Letsch (TL) Projektbetreuer

4 Managementabläufe

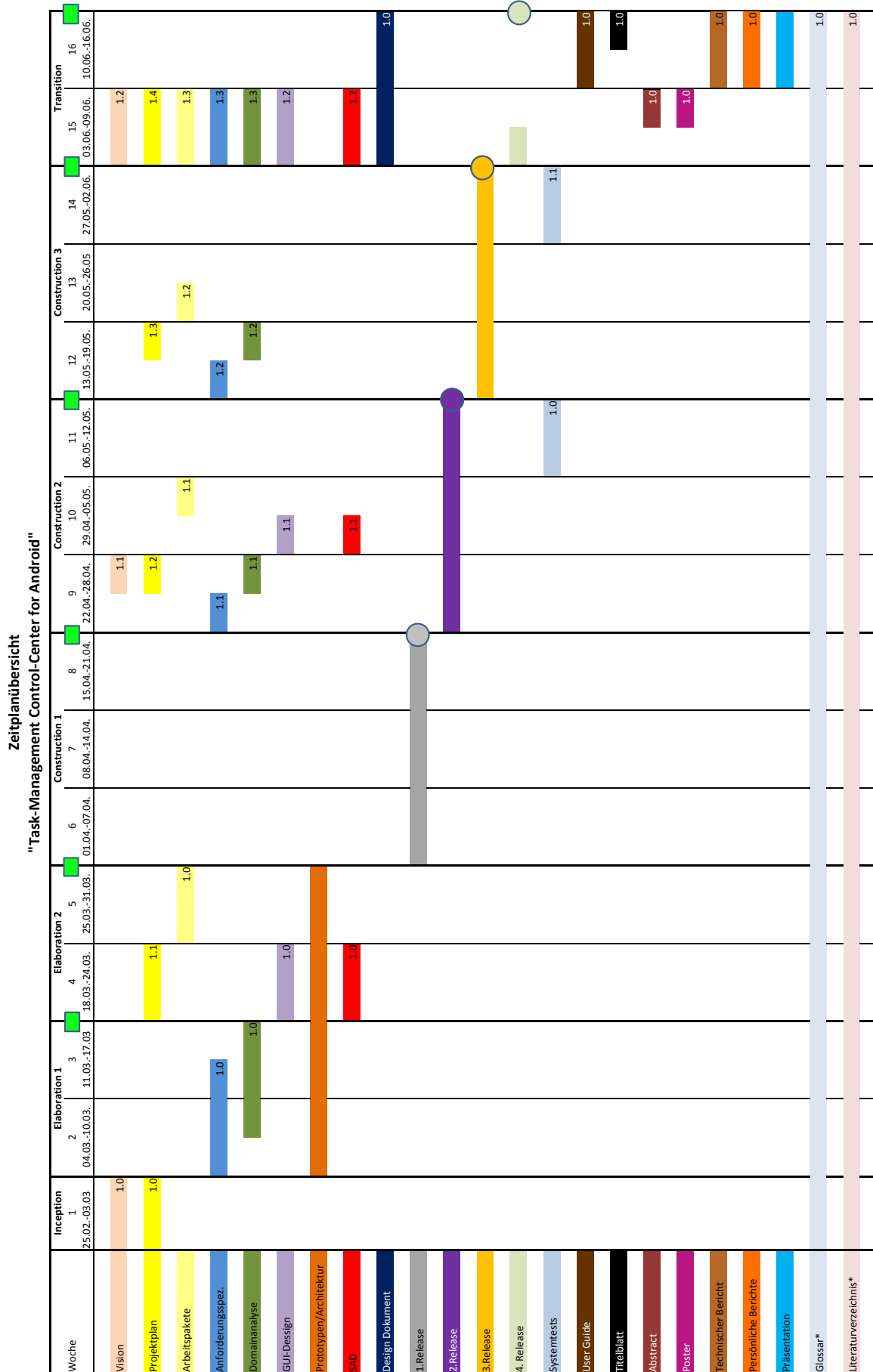
4.1 Projektdauer

Der Zeitplan basiert auf einer Arbeitszeit von 370 Stunden pro Teammitglied, was ein Total von 740 Stunden ergibt. Zudem wird davon ausgegangen, dass die Arbeitszeit gleichmässig über alle 16 Wochen verteilt wird. Das Projekt beginnt am 21.02.2011 und endet mit der Abgabe am 17.06.2011.

4.2 Projektplan

4.2.1 Zeitplanübersicht

Die folgende Abbildung zeigt, wann, welche Artefakte erstellt werden:



* Glossar und Literaturverzeichnis werden laufend nachgeführt und erhalten deshalb keine laufende Versionsnummer
 ■ Meilensteine, für Details siehe Projektplan, Kapitel 4.2.3, Seite 9

4.2.2 Beschreibung der Artefakte

Vision

Im Dokument Vision wird die Projektidee grob beschrieben. Geplante Versionen:

Version	Inhalt
1.0	Vision bis und mit Release 1
1.1	Entspricht Version 1.0, da keine neuen Anforderungen hinzugekommen sind bzw. zu ergänzen waren.
1.2	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.1: Vision

Projektplan

Der Pp definiert die Planung und Organisation des Projekts. Ausserdem dient er als Grundlage für weitere Projektdokumente. Geplante Versionen:

Version	Inhalt
1.0	Planung bis und mit Meilenstein 1
1.1	Planung bis und mit Meilenstein 3
1.2	Planung bis und mit Meilenstein 4
1.3	Planung bis und mit Meilenstein 5
1.4	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.2: Projektplan

Anforderungsspezifikation

Im Dokument As sind die funktionalen und nichtfunktionalen Anforderungen an das Produkt TCA ersichtlich. Geplante Versionen:

Version	Inhalt
1.0	Anforderungen an CRUD-Operationen der Tasks und Synchronisation von Tasks und Dateien.
1.1	Überarbeitung der Synchronisationsanforderungen und Ergänzen jener für die «Timestamps».
1.2	Entspricht Version 1.1, da keine neuen Anforderungen hinzugekommen sind bzw. zu ergänzen waren.
1.3	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.3: Anforderungsspezifikation

Domainanalyse

Die Da ermöglicht es, sich eine Übersicht über die Business Logik des Projektes zu verschaffen. Geplante Versionen:

Version	Inhalt
1.0	Analyse von CRUD-Operationen der Tasks und Synchronisation von Tasks und Dateien.
1.1	Überarbeiten der Synchronisationsanalyse und Ergänzen der Analyse von «Timestamps».
1.2	Analyse definitiv umgesetzter Funktionalitäten (entspricht Version 1.1, da keine neuen Anforderungen hinzugekommen sind bzw. zu ergänzen waren).
1.3	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.4: Domainanalyse

GUI-Design

Das GUI-Design enthält die gestalterische und benutzerspezifische Planung des User Interfaces (UIs). Folgende Versionen sind geplant:

Version	Inhalt
1.0	Erster Entwurf des GUIs für das Mobile-Device und den PC entsprechend der As. Personas und Szenarios erstellt.
1.1	Anpassungen am Entwurf der Mobile-Device-Oberfläche.
1.2	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.5: GUI-Design

Prototypen

Es werden mehrere kleine Prototypen erstellt. Aus diesen Prototypen wird danach die Architektur erstellt, welche als Grundgerüst für die nachfolgenden Construction-Iterationen dient.

Software Architektur Dokument

Das SAD liefert eine Übersicht der Architektur respektive über den logischen und physischen Aufbau von TCA. In den einzelnen Kapiteln wird eingehend auf die «Logical view», die «Process view», die «Deployment view» und die «Data view» eingegangen, wie sie in [2] beschrieben sind.

Geplante Versionen:

Version	Inhalt
1.0	Architekturübersicht über die einzelnen Subsysteme und Schichten betreffend CRUD-Operationen der Tasks und Synchronisation von Tasks und Dateien.
1.1	Anpassungen der einzelnen Subsysteme gemäss effektiver Architektur.
1.2	Final Review, keine inhaltlichen Änderungen, nur Überprüfung Rechtschreibung, Verweise, Abkürzungen.

Tabelle 4.6: Software Architektur Dokument

Design Dokument

Das DD enthält Informationen über das Design respektive Design-Entscheide. Es ist eine Version geplant, Version 1.0, welche erst zum Schluss erstellt wird.

Releases

Siehe Kapitel 4.2.4, Seite 10.

User Guide

Der User Guide (UG) ist eine Anleitung, mit der ein Benutzer erlernen kann, wie das Programm genutzt wird. Es ist eine Version geplant, Version 1.0.

Systemtests

Die Systemtests werden in zwei Teilen durchgeführt. Erste Tests finden in der Woche 11 statt, weitere Tests folgen in der Woche 14. Die Resultate werden in der Testdokumentation festgehalten. Diese enthält die Dokumentation aller durchgeführten Systemtests und eine Übersicht aller Bereiche, welche durch Unit Tests abgedeckt sind.

Weitere Dokumente

Titelblatt, Abstract, Poster, technischer und persönlicher Bericht werden gemäss den Vorgaben der Abteilung Informatik erstellt.

Präsentation

Ziele und Resultate des Projektes werden präsentiert.

Glossar

Das Glossar (Gl) listet wichtige Abkürzungen und Begriffe im Rahmen des Projektes auf und wird laufend ergänzt.

Literaturverzeichnis

Das Literaturverzeichnis enthält Angaben zur verwendeten Literatur und wird laufend ergänzt. Jeweils am Ende jedes Dokuments befindet sich eine Übersicht der referenzierten Werke. Somit wird zum Verständnis der einzelnen Dokumente nicht zusätzlich ein Literaturverzeichnis benötigt.

4.2.3 Iterationsplanung

Iteration Assessments

Die Inhalte der Iterationen werden an vorangehenden Iteration Assessments festgelegt. Für Details zu den Assessments siehe Kapitel 8.3.3, Seite 18.

Iterationen

Im UP sind vier Phasen definiert, welche in mehreren Iterationen durchlaufen werden. Die Tabelle 4.7 listet die geplanten Iterationen sowie deren Inhalte auf. Die genauen Daten, die Dauer und der Inhalt der Iterationen können/kann der Zeitplanung entnommen werden (siehe Kapitel 8.1.3, Seite 16).

Name	Beschreibung
Inception	Start und Detailplanung des Projekts.
Elaboration 1	Erfassung der Anforderungen und Analyse von CRUD-Operationen der Tasks und Synchronisation von Tasks und Dateien.
Elaboration 2	Ausarbeitung des GUIs, Prototypen und Architekturskelett, Anpassung des Projektplans und Erstellung des SADs.
Construction 1	Fokus auf 1. Release.
Construction 2	Fokus auf 2. Release, Durchführung erster Systemtests.
Construction 3	Fokus auf 3. Release, Durchführung abschliessender Systemtests.
Transition	Verfassen vorgegebener Dokumente gemäss [1] und abschliessende Arbeiten für 4. Release.

Tabelle 4.7: Liste der Iterationen

Meilensteine

Tabelle 4.8 listet die geplanten Meilensteine sowie die vorgesehenen Arbeitsergebnisse auf. Die genauen Daten der Meilensteine können der Zeitplanung entnommen werden (siehe Kapitel 8.1.3, Seite 16).

Nr.	Name	Arbeitsergebnis
MS1	Pp, As und Da	Anforderungen an das Projekt und dessen Analyse durchgeführt.
MS2	Prototyp	Architekturgerüst und Prototypen für GUI von PC und Mobile-Device erstellt.
MS3	1. Release	CRUD-Operationen für Tasks und Synchronisation von Tasks und Dateien implementiert.
MS4	2. Release	Oberflächen von Mobile und PC mit den CRUD-Operationen verlinkt, Synchronisation ausgearbeitet, Timestamps implementiert.
MS5	3. Release	Austausch von Tasks und Dateien implementiert.
MS6	4. Release	Präsentation und Abgabe des Projekts.

Tabelle 4.8: Liste der Meilensteine

4.2.4 Releases

Am Ende von jeder Construction-Iteration und der Transitionphase wird ein Release der Software erstellt. Die genauen Daten können der Zeitplanung entnommen werden (siehe Kapitel 8.1.3, Seite 16), die Releases und ihre Inhalte der Tabelle 4.9. Zu jedem Release wird ein Release-Dokument erstellt, welches Informationen zu speziellen Konfigurationen enthält. Die Inhalte der Releases werden in der jeweiligen Projektplan-Version festgelegt.

Name	Iteration	Inhalt
0.1	Construction 1	Entspricht MS3.
0.2	Construction 2	Entspricht MS4.
0.3	Construction 3	Entspricht MS5.
0.4	Transition	Abgabe der Ba.

Tabelle 4.9: Liste der Releases

4.2.5 Code Reviews

Während den Construction-Iterationen werden regelmässig Code Reviews durchgeführt (für Details siehe Kapitel 8.3.2, Seite 18). Die Daten und Inhalte der durchgeführten Code Reviews können der Tabelle 4.10 entnommen werden.

Datum	Review-Gegenstand	Revision
07.04.2011	Android: Startbildschirm, Alle Klassen der Taskübersicht; PC: Aufbau der Applikation, GUI-Aufbau, DB-Klassen.	210
18.04.2011	Android: Datenbankzugriff; PC: Tasks auf Domainschicht, Anpassungen an GUI.	241
28.04.2011	Android GUI: Taskübersicht und neuer Task; PC: Tasks auf Domainschicht, Refactorings an GUI.	280
06.05.2011	Android GUI: Taskübersicht und Task editieren; PC: Refactorings an GUI, Operationen auf Domainschicht.	312
12.05.2011	Android: Coderefactorings; PC: XferAgent.	340
30.05.2011	Android, PC: Abschluss Review.	414

Tabelle 4.10: Liste der Code Reviews

4.3 Besprechungen

Wöchentliche Teamsitzungen sind donnerstags zwischen 10:10 und 11:50 angesetzt, jene mit dem Betreuer finden donnerstags zwischen 08:10 und 10:10 statt. Die Dauer kann je nach Bedarf variieren. Zusätzlich finden kürzere Besprechungen nach Absprache statt.

Teilnehmer: DM, RR
Ort: Hochschule für Technik Rapperswil (HSR), 1.262
Zeit: Donnerstag, 10:10 bis 11:50 (wöchentlich)

Teilnehmer: DM, RR, TL
Ort: HSR, 1.223
Zeit: Donnerstag, 8:10 bis 10:10 (wöchentlich)

5 Risiko-Management

Die Risiken, ihren Einfluss auf den Projekterfolg und die Gegenmassnahmen sind im separaten Dokument «Risikomanagement» beziehungsweise `Risikomanagement.xlsx` beschrieben.

6 Arbeitspakete

Die einzelnen Arbeitspakete (Ap) sind im Dokument Ap ersichtlich, welches als Ergänzung zum Zeitplan (siehe Kapitel 8.1.3, Seite 16) dient. Es definiert die Inhalte beziehungsweise Resultate der Ap, ihre Prioritäten, die Abhängigkeiten von anderen Ap sowie die Personen, welche die Umsetzung der Ap überwachen.

7 Infrastruktur

7.1 Räumlichkeiten

Teamsitzungen, Reviews und Präsentationen finden in den Räumlichkeiten der HSR statt.

7.2 Hardware

Jedes Projektmitglied arbeitet mit dem privaten PC und/oder an dem von der HSR zur Verfügung gestellten Arbeitsplatz im Raum 1.262. Beim Ausfall einer Arbeitsstation stehen Ersatzrechner zur Verfügung.

7.3 Software

- Technologien:
 - Android
 - Java Swing
- Programmiersprache:
 - Java 6
- Entwicklungsumgebung:
 - Eclipse
 - Enterprise Architect (EA)
 - NetBeans (als GUI-Builder)
- Versionsverwaltungssoftware:
 - Subversion¹, Zuständigkeit: RR
 - Dropbox² (für die Ablage der PDF-Versionen der Dokumente und Releases)
- Code-Analyse:
 - Findbugs³
 - Metrics⁴
 - EclEmma⁵
- Dokumentation: \LaTeX
 - MikTeX ⁶
 - Texmaker ⁷

¹<https://svns.hsr.ch/TaskManagementControlCenterforAndroid>

²<https://www.dropbox.com>

³<http://findbugs.sourceforge.net>

⁴<http://www.stateofflow.com/projects/16/eclipsemetrics>

⁵<http://www.eclEmma.org>

⁶<http://miktex.org>

⁷<http://www.xmlmath.net/texmaker>

7.4 Kommunikation

Folgende Kommunikationsmittel werden eingesetzt:

- E-Mail
- Windows Messenger

8 Qualitätsmassnahmen

8.1 Dokumentation

8.1.1 Allgemein

Die Personen, welche für die einzelnen Ap verantwortlich sind, identifizieren, welche Dokumente im Zuge der Umsetzung eines Ap erstellt, aktualisiert beziehungsweise erweitert werden müssen. Sie kümmern sich auch um deren Umsetzung. Anschliessend werden die Dokumente einem Review unterzogen (siehe Kapitel 8.3.1, Seite 18).

8.1.2 Sitzungsprotokolle

Sitzungen mit dem Betreuer werden protokolliert und innerhalb von 24 Stunden dem anderen Projektmitglied respektive dem Betreuer per Mail als Anhang weitergeleitet. So wird sichergestellt, dass alle Entscheidungen und Diskussionspunkte festgehalten sind, ausserdem können Missverständnisse vermieden werden. An den Teamsitzungen werden TODO-Listen erstellt, in welchen die anstehenden Arbeiten festgehalten werden.

8.1.3 Zeitplan

Jedes Teammitglied erfasst seine Arbeitszeiten selbst und ist dafür verantwortlich, den Bedarf für die einzelnen Ap in die Gesamtübersicht des Dokuments «Zeitplan» respektive `Zeitplan.xlsx` einzutragen. Die Erfassung der täglichen Arbeiten erfolgt in einer separaten Excel-Tabelle «Zeiterfassung» beziehungsweise `Zeiterfassung.xlsx`, welche jedes Teammitglied für sich selbst zur eigenen Übersicht führt.

8.2 Style Guides

8.2.1 Code

Der Programmcode richtet sich grundsätzlich nach den Code Conventions von Sun¹. Diese werden mithilfe des Profils «Task-Management Control-Center for Android» von Eclipse durchgesetzt. Folgende Anpassungen gegenüber den Standardeinstellungen wurden vorgenommen:

- Indentation
 - «Align fields in columns» gesetzt.
 - «Statements within 'switch' body» gesetzt.
- White Space
 - Expression: Bei «Type cast» kein Abstand zwischen schliessender Klammer und Objekt.
 - Arrays: Bei «Array initializer» keinen Abstand nach öffnender und vor schliessender geschweifter Klammer.
- Blank Lines

¹<http://java.sun.com/docs/codeconv/html/CodeConvTOC.doc.html>

- Zusätzliche Leerzeile vor erster Deklaration in einer Klasse.
- New Lines
 - Keine neuen Zeilen in jeglichen leeren Klassen, Methoden und Blöcken respektive zwischen leeren geschweiften Klammern.
- Control Statements
 - Bei einzeiligem «if-Statement» kann alles auf eine Zeile geschrieben werden, sofern die Zeilenlänge dies zulässt.
- Line Wrapping
 - 'enum' declaration: Bei Konstanten erfolgt Zeilenumbruch dort, wo es nötig ist.
- Comments
 - «Enable header comment formatting» gesetzt.

Zudem wurden folgende Punkte beschlossen:

- Interfacenamen beginnen immer mit einem I.
- Member-Attribute
 - werden nur initialisiert, wenn dies erforderlich ist.
 - werden im Android-Projekt gemäss Android-Richtlinien mit einem vorangestellten `m` gekennzeichnet.
 - werden im Java-Projekt nur, falls erforderlich, mit `this.name` adressiert.
- Konstanten werden aufgrund der Übersichtlichkeit am Ende einer Klasse aufgeführt. Dies wird so gehandhabt, da in einigen Klassen viele Konstanten erforderlich sind. Wenn diese am Anfang der Klasse stünden, müsste ständig über diese «hinweggescrolled» werden.
- Folgende Sektionen werden mittels Kommentaren gekennzeichnet (die Aufzählung unten entspricht der Reihenfolge im Code):
 - «Helpers»: Falls sinnvoll werden Hilfsmethoden, welche in mehreren public Methoden benutzt werden, in dieser Sektion zusammengefasst.
 - «Getters and Setters»
 - «Inner Classes»
 - «Constants»

Weitere Sektionen können je nach Bedarf ergänzt werden.

8.2.2 Dokumente

Für Dokumente stehen \LaTeX -Dokumentvorlagen zur Verfügung, die eine einheitliche Formatierung begünstigen.

Wird auf ein anderes Kapitel referenziert, geschieht dies in der Form: Dokument (falls sich die besagte Stelle in einem anderen Dokument befindet), Kapitel, Seite, z.B. «Projektplan», Kapitel 4.2.2, Seite 6. Bei Tabellen und Abbildungen wird nur die Tabellenummer respektive Abbildungsnummer angegeben und falls sich die referenzierte Stelle in einem anderen Kapitel befindet zusätzlich noch die Seitenzahl, z.B. Tabelle 4.8, Seite 10.

Gängige Abkürzungen, wie z.B. PC, werden nur in der Kurzform verwendet, und zur Übersicht im Abkürzungsverzeichnis festgehalten. Andere Abkürzungen werden bei der ersten Verwendung ausgeschreiben, z.B. Abkürzung (Ak).

8.3 Reviews

8.3.1 Dokumentations-Reviews

Dokumente werden von der zuständigen Person erarbeitet und vom jeweils anderen Teammitglied gegengelesen. Danach werden die Entwürfe im Team besprochen und inhaltlich überarbeitet. Eine Person übernimmt jeweils die Abschlussredaktion, welche die Überarbeitung durch zwei Korrekturlesungen beinhaltet. Jeder Revisor trägt sich in die «Revisionshistorie» ein, damit Änderungen und Ergänzungen jederzeit verfolgt werden können.

8.3.2 Code Reviews

Es werden regelmässig Code Reviews wichtiger Komponenten durchgeführt. Dabei liegt der Schwerpunkt bei folgenden Kriterien:

- Code-Richtlinien (nicht alles lässt sich mit dem erstellten Profil «Task-Management Control-Center for Android» formatieren)
- «Code Smell»
- Verständlichkeit

Unterstützt werden die Code Reviews durch eine Code-Analyse mit Findbugs, Metrics und EcEmma, welche das Projekt unter anderem auf duplizierten oder nicht erreichbaren Code untersuchen. Die Code Reviews werden im Check-in-Kommentar erfasst.

8.3.3 Iteration Assessments

An der letzten Sitzung vor dem Ende jeder Iteration wird ein Iteration Assessment durchgeführt. Es wird der aktuelle Status der Arbeiten beurteilt und die zu bearbeitenden Arbeitspakete und Ziele für die nächste Iteration festgelegt. In diesem Rahmen wird der Pp aktualisiert beziehungsweise revidiert.

8.4 Versionsverwaltungssystem

Im Subversion (SVN)-Repository werden sämtliche UP-Artefakte, die Dokumentation sowie der gesamte Quellcode der Software verwaltet. Alle Projektteilnehmer können auf diese Weise gemeinsam am Projekt arbeiten. Es dürfen nur funktionsfähige, funktional zusammenhängende Patches eingechekkt werden.

PDF-Versionen der Dokumente und die Releases werden aus Platzgründen auf die Dropbox ausgelagert.

8.5 Tests

8.5.1 Unit Tests

Es wird auf Test Driven Development (TDD) gesetzt, das heisst, die Projektmitglieder sind verpflichtet, fortlaufend Unit Tests zu schreiben und gegen diese Tests zu entwickeln.

8.5.2 Systemtests

Anhand der Use Cases (UCs) werden Systemtests durchgeführt und protokolliert, um sicherzustellen, dass das System den funktionalen Anforderungen genügt.

8.6 Logging

Es wird ein Logging mit zweidimensionaler Laufzeit-Konfiguration eingesetzt. Dabei wird nach Kategorie (z.B. Package) und Log-Level gelogged.

Bei dichtem Logging, z.B. in Loops, wird zuerst der momentane Loglevel abgefragt, um eine unnötige Aufbereitung von Logausgaben zu verhindern.

8.6.1 Loglevels

Die Loglevels werden unverändert von AnCoF übernommen. Details können dem «Projektplan» von AnCoF, Kapitel 8.7.1, Seite 16 entnommen werden.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	22.02.2011	DM	Dokument erstellt.
1.0rc02	22.02.2011	DM, RR	Erster Vorschlag für Pp erstellt.
1.0rc03	24.02.2011	RR	Überarbeiten der Kapitel gemäss Vision und Besprechung.
1.0rc04	03.03.2011	DM, RR	Anpassung der Artefakte gemäss Besprechung vom 03.03.2011.
1.0	10.03.2011	DM	Akzeptierte Version 1.0.
1.1rc01	21.03.2011	DM	Inhalt der Artefakte gemäss Planung aktualisiert.
1.1	24.03.2011	DM	Akzeptierte Version 1.1.
1.2rc01	07.04.2011	DM, RR	Qualitätsmassnahmen zum Code ergänzt, Ziel und Zweck des Dokuments erfasst.
1.2rc02	19.04.2011	DM	Releasenummerierung angepasst.
1.2rc03	25.04.2011	DM	Inhalt der Dokumente entsprechend aktueller Planung bis und mit Release 2 angepasst.
1.2	28.04.2011	DM	Akzeptierte Version 1.2.
1.3rc01	28.04.2011	RR	Codeanalyse angepasst.
1.3rc02	09.05.2011	RR	Codeanalyse aktualisiert.
1.3rc03	12.05.2011	DM, RR	Qualitätsmassnahmen und Artefaktinhalte aktualisiert.
1.3	19.05.2011	RR	Akzeptierte Version 1.3.
1.4rc01	09.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.4	09.06.2011	RR	Akzeptierte Version 1.4.

Tabelle 8.1: Revisionshistorie

Abkürzungsverzeichnis

AnCoF	Android Control Framework
Ap	Arbeitspakete
As	Anforderungsspezifikation
Ba	Bachelorarbeit
CRUD	create, read, update, delete
Da	Domainanalyse
DB	Datenbank
DD	Design Dokument
DM	Daniela Meier
EA	Enterprise Architect
GI	Glossar
GUI	Graphical User Interface
HSR	Hochschule für Technik Rapperswil
PC	Personal Computer
Pp	Projektplan
RR	Ramona Rudnicki
SAD	Software Architektur Dokument
SVN	Subversion
TCA	Task-Management Control-Center for Android
TDD	Test Driven Development
TL	Thomas Letsch
UC	Use Case
UG	User Guide
UI	User Interface
UP	Unified Process

Tabellenverzeichnis

3.1	Organisationsstruktur	3
4.1	Vision	6
4.2	Projektplan	6
4.3	Anforderungsspezifikation	6
4.4	Domainanalyse	7
4.5	GUI-Design	7
4.6	Software Architektur Dokument	8
4.7	Liste der Iterationen	9
4.8	Liste der Meilensteine	10
4.9	Liste der Releases	10
4.10	Liste der Code Reviews	10
8.1	Revisionshistorie	20

Literaturverzeichnis

[1] HSR: *Anleitung: Dokumentation Semester-, Bachelor- und Diplomarbeiten*

[2] LARMAN, Craig: *Applying UML and Patterns*. Prentice Hall, 2004. – ISBN 978-0-13-148906-2

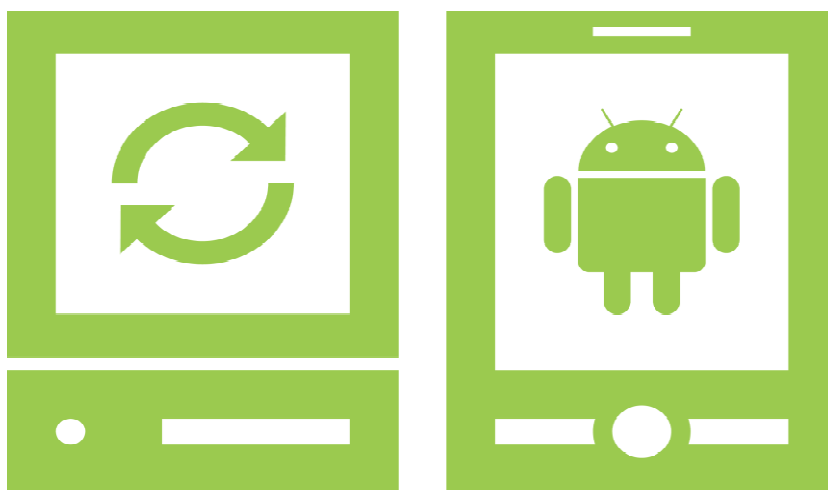
Risikoanalyse

«Task Management Control-Center for Android»

Daniela Meier (d2meier@hsr.ch)

Ramona Rudnicki (rrudnick@hsr.ch)

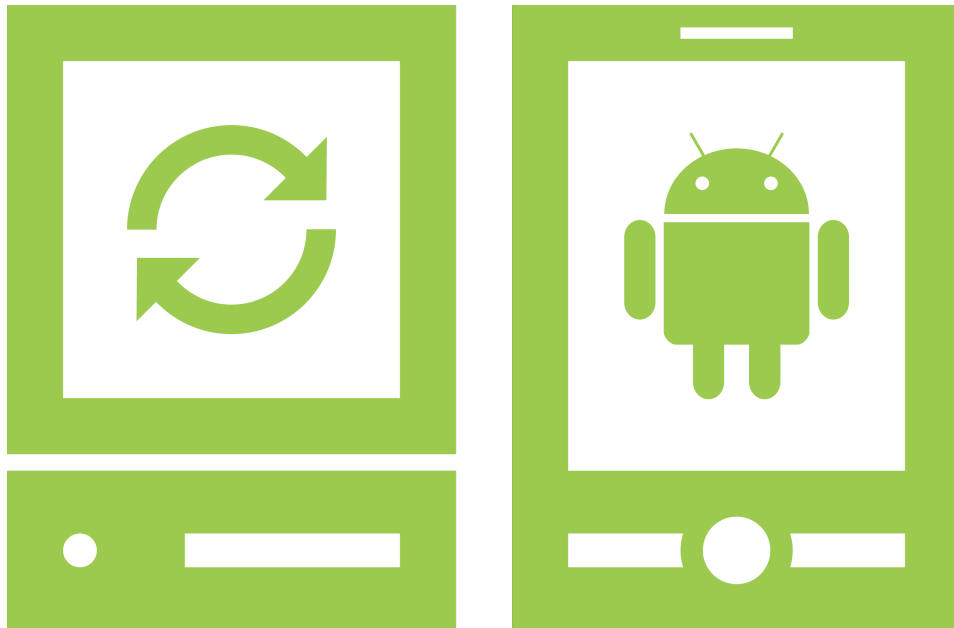
10. Juni 2011



Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	22.02.2011	DM	Dokument erstellt
1.0rc02	03.03.2011	DM	Risiken dem Projekt angepasst und ergänzt
1.0rc03	28.04.2011	DM	Risiken überprüft und angepasst
1.0	09.06.2011	RR	Akzeptierte Version 1.0

Risiko-Nr	Risikotitel	Risikobeschreibung	max. Schaden [h]	Eintrittswahrscheinlichkeit [%]	gewichteter Schaden [h]	Massnahmen zur Vermeidung/Verminderung	Vorgehen bei Eintreffen
R1	Ausfall Arbeitsstation	HW eines Projektmitgliedes fällt aus	8	5.00 %	0.40	Alternative HW organisieren und mind. alle 4 Stunden erledigte Arbeiten im Repository einchecken	an HSR-Rechner bzw. eigenem Laptop arbeiten
R2	Ausfall SVN-Server	SVN-Server der HSR fällt aus	20	2.00 %	0.40	lokale Kopien aktuell halten	Ausweichen auf anderen SVN-Server (z.B. Google)
R4	(Teil-) Ausfall eines Projektmitgliedes	Ausfall eines Projektmitgliedes aufgrund Krankheit, Unfall, Studiumabbruch	360	2.00 %	7.20	Verantwortlichkeiten in Team klar definieren, Stellvertreter definieren	Mehrarbeit in Folgewochen, zusätzliche Arbeit auf anderes Teammitglied verteilen, Funktionsumfang kürzen
R5	Erfahrung der Projektmanager	Die Projektmanager haben zu wenig Erfahrung mit ähnlichen Projekten und können daher den Aufwand der Arbeitspakete nicht korrekt schätzen	10	30.00 %	3.00	kritische Arbeitspakete frühzeitig bearbeiten und bei Unklarheiten mit Betreuer Rücksprache halten, iterativ planen	nicht alle geplanten Funktionen und Features umsetzen, Mehrarbeit, Sitzung mit Betreuer, um weitere Planung zu besprechen
R6	Probleme mit Technologien	Verwendete Technologien führen zu Problemen da das Basiswissen fehlt	30	25.00 %	7.50	Versuchen zu vereinfachen, iterativ planen	Dokumentationen und Beispiele im Internet suchen, systematisch in Themen einarbeiten
R7	Ausfall des Testgerätes	Das verwendete Android-Telefon ist defekt und muss repariert/ersetzt werden	20	5.00%	1.00	Alternativgerät organisieren	soweit möglich Emulator benutzen; neues Gerät organisieren; später angesetzte Funktionalitäten vorziehen falls Emulator für Tests nicht ausreicht
R8	Probleme mit Darstellung von GUI	Das UI wird nicht auf allen Plattformen gleich dargestellt (Linux, Windows)	30	40.00%	12.00	GUI regelmässig auf allen Plattformen testen, einfaches GUI gestalten	GUI-Komponenten einzeln testen, um möglichst schnell Fehlerquelle zu finden und zu beheben
Summe			478	109.00 %	31.50		



Arbeitspakete

«Task-Management Control-Center for Android»

Version 1.3, 10. Juni 2011

Daniela Meier (d2meier@hsr.ch)

Ramona Rudnicki (rrudnick@hsr.ch)

Name	Resultat	Abhängigkeit	Priorität	Wer
Projekt Management				
Vision	Vision erstellen	Aufgabenstellung	mittel	DM
Projektplan (Pp)	Pp erstellen	Vision	hoch	DM
Zeitplanung	Zeitplan erstellen und in Pp einfließen lassen	Releases	hoch	RR
Risiko Management	Risiken analysieren und Gegenmassnahmen festlegen, in Pp einfließen lassen	-	mittel	DM
Qualitätsmassnahmen	Qualitätsmassnahmen festlegen und in Pp einfließen lassen	-	tief	DM
Arbeitspakete (Ap)	Ap definieren	Software Architektur	tief	RR
Artefakte	Inhalt Artefakte definieren	ment (SAD) Pp	mittel	DM
Requirements				
Use Case (UC)s brief	UC brief erstellen	Vision	hoch	DM, RR
Anforderungsspezifikation (As)	As erstellen	-	mittel	DM
UC Diagramm	UC Diagramm erstellen	UCs brief	tief	RR
Analyse				
Domainanalyse (Da)	Da erstellen	-	mittel	RR
Domainmodell	Domainmodell erstellen	UCs brief	hoch	DM, RR
Sequenzdiagramme	System-Sequenz-Diagramm (SSD) erstellen	UCs brief	mittel	DM, RR
Operation Contract (OC)s	OCs erstellen	SSD	tief	DM
Design				
Designmodell	Designmodell erstellen	Domainmodell	hoch	RR
SAD	SAD erstellen	As	mittel	RR
Design Dokument (DD)	DD erstellen	Designmodell	hoch	DM, RR
GUI-Design	GUI-Entscheidung dokumentieren	Paperprototype	mittel	RR
Paperprototype	Paperprototypes erstellen	UCs brief	hoch	DM, RR
Implementation				
Prototypen	Erstellen erster Versuche und Prototypen	As	mittel	DM, RR
GUI-Prototyp PC	Erste Implementations-Versuche des GUI für den PC	Paperprototype	hoch	DM
GUI-Prototyp Mobile	Erste Implementations-Versuche des GUI für das Mobile	Paperprototype	mittel	DM
Architekturskelett	Erstellen der Packagestruktur und der wichtigsten Klassen	Prototypen, As, Da	hoch	RR
CRUD-Operationen für Tasks	Domainlogik für Task-Behandlung implementieren	Architekturskelett, As, Da	hoch	DM, RR
Dateisynchronisation	Domainlogik für die Dateisynchronisation implementieren	Prototypen, As, Da	hoch	DM
GUI Mobile	Erstellen des GUI für das Mobile-Device	GUI-Prototyp Mobile, As, Da	mittel	DM

Name	Resultat	Abhängigkeit	Priorität	Wer
GUI PC	Erstellen des GUI für den PC	GUI-Prototyp PC, As, Da	mittel	RR
Tasksynchronisation	Taskaustausch zwischen den Devices implementieren	CRUD-Operationen für Tasks	hoch	RR
Allgemeine Arbeiten	«Aufräumarbeiten»	-	tief	DM, RR
Code Review	Kontinuierliche Überprüfung der Code-Qualität wichtiger Komponenten	-	mittel	DM, RR
Test				
Unit-Tests	Unit-Tests erstellen für TDD	Artefakte Design	mittel	DM, RR
Systemtests	Funktionale Systemtests definieren und durchführen	Artefakte Implementierung	tief	DM, RR
Bugfixing	Verbliebene Fehler beheben	-	mittel	DM, RR
Dokumentation				
Titelblatt	Titelblatt für Gesamtabgabe erstellen	alle Dokumente	tief	RR
Abstract	Übersicht über Projekt	-	mittel	DM
Poster	Übersicht über Projekt	-	hoch	RR
Glossar (GI)	Definition projektspezifischer Begriffe	-	tief	DM
Literaturverzeichnis	Übersicht über alle benutzten Werke und Internetreferenzen	-	tief	DM
User Guide (UG)	Anleitung für Benutzer die Task-Management Control-Center for Android (TCA) nutzen	Artefakte Implementierung	hoch	RR
Technischer Bericht	Übersicht über Aufgabenstellung, Arbeit und Resultate	-	mittel	DM
Persönlicher Bericht	Erfahrungsbericht	-	tief	DM, RR
Präsentation	Präsentation Bachelorarbeit (Ba)	-	hoch	DM, RR
Allgemeine Bereinigung	Korrekturlesung	alle Dokumente	tief	DM
Dokumente	Vorlagen erstellen	-	tief	DM
Dokumentvorlagen	Alle Funktionen beschreiben	As	hoch	DM, RR
Javadoc				
Sitzungen				
Sitzungen mit Betreuer	Gemeinsame Projektplanung	-	-	-
Teamsitzungen	Detailplanung	Sitzungen mit Betreuer	-	-
Infrastruktur				
Arbeitsplatz einrichten	Subversion (SVN), Eclipse einrichten	-	tief	DM, RR

Tabelle 0.1: Liste der Arbeitspakete

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	28.03.2011	RR	Arbeitspakete geplant.
1.0	31.03	RR	Akzeptierte Version 1.0.
1.2rc01	12.05.2011	DM, RR	Arbeitspakete bereinigt.
1.2	12.05.2011	RR	Akzeptierte Version 1.2.
1.3rc01	09.06.2011	DM, RR	Rechtschreibung und Abkürzungen überprüft.
1.3	09.06.2011	RR	Akzeptierte Version 1.3.

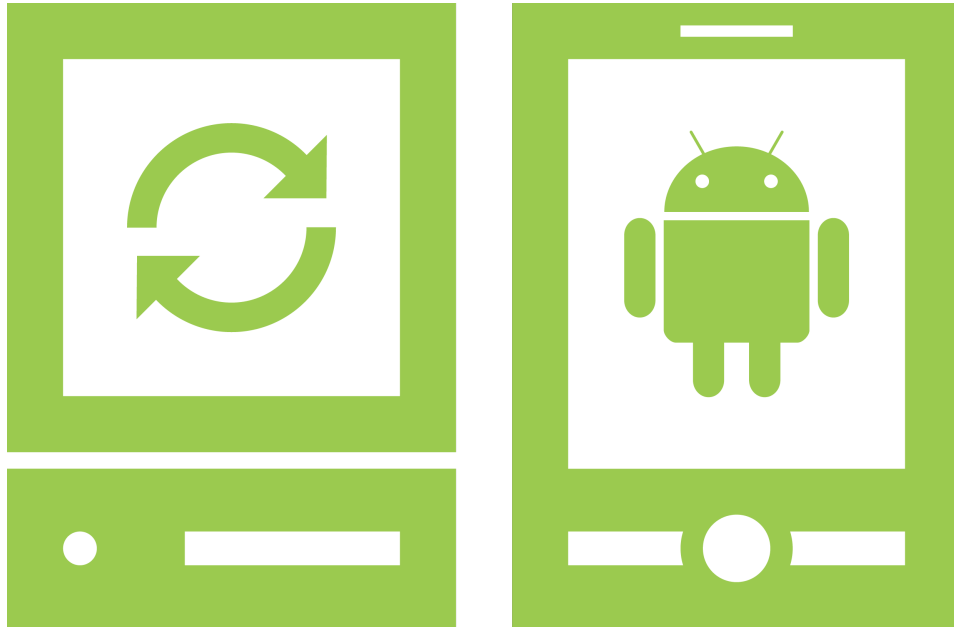
Tabelle 0.2: Revisionshistorie

Abkürzungsverzeichnis

Ap Arbeitspakete
As Anforderungsspezifikation
Ba Bachelorarbeit
CRUD create, read, update, delete
Da Domainanalyse
DD Design Dokument
GI Glossar
GUI Graphical User Interface
OC Operation Contract
PC Personal Computer
Pp Projektplan
SAD Software Architektur Dokument
SSD System-Sequenz-Diagramm
SVN Subversion
TCA Task-Management Control-Center for Android
TDD Test Driven Development
UC Use Case
UG User Guide

Tabellenverzeichnis

0.1	Liste der Arbeitspakete	2
0.2	Revisionshistorie	3



Anforderungsspezifikation

«Task-Management Control-Center for Android»

Version 1.3, 10. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Funktionale Anforderungen	2
2.1 Aktoren	2
2.2 Use Case-Diagramm	2
2.2.1 Dateisynchronisation	2
2.2.2 Tasks	3
2.2.3 Verbindung und Konfiguration	4
2.3 Use Cases brief	4
2.3.1 Dateisynchronisation	4
2.3.2 Tasks	5
2.3.3 Verbindung	6
2.3.4 Konfiguration	6
2.4 Schnittstelle	6
2.4.1 Dateien	6
2.4.2 Tasks	7
2.5 Anpassungen an bestehende Frameworks	7
2.5.1 Task-Management-Framework on Smart-Phone	7
2.5.2 Android Control Framework	7
3 Nichtfunktionale Anforderungen	8
3.1 Funktionalität	8
3.2 Zuverlässigkeit	8
3.3 Erlernbarkeit	8
3.4 Wartbarkeit	8
3.5 Effizienz	8
3.6 Unterstützte Systeme	8

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 6.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Funktionale Anforderungen

2.1 Aktoren

Der Benutzer agiert als Aktor im System. Diese Rolle fällt zudem auch dem Firmennetzwerk zu, da es aktiv an Use Cases (UCs) teilnimmt.

2.2 Use Case-Diagramm

Das UC-Diagramm wird zur Übersicht in mehrere Bilder aufgeteilt. Dies hat jedoch keinen Einfluss auf die UCs.

2.2.1 Dateisynchronisation

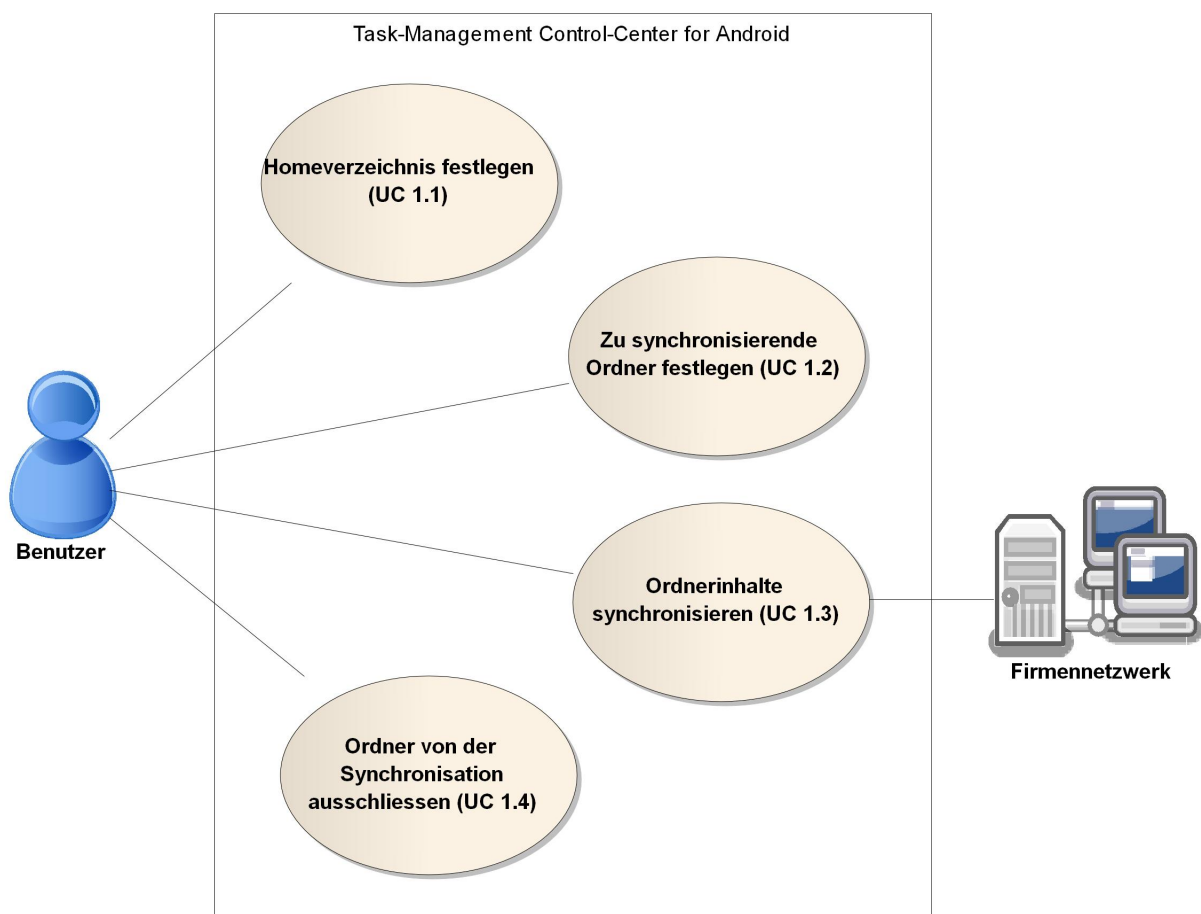


Abbildung 2.1: UC-Diagramm Dateisynchronisation

2.2.2 Tasks

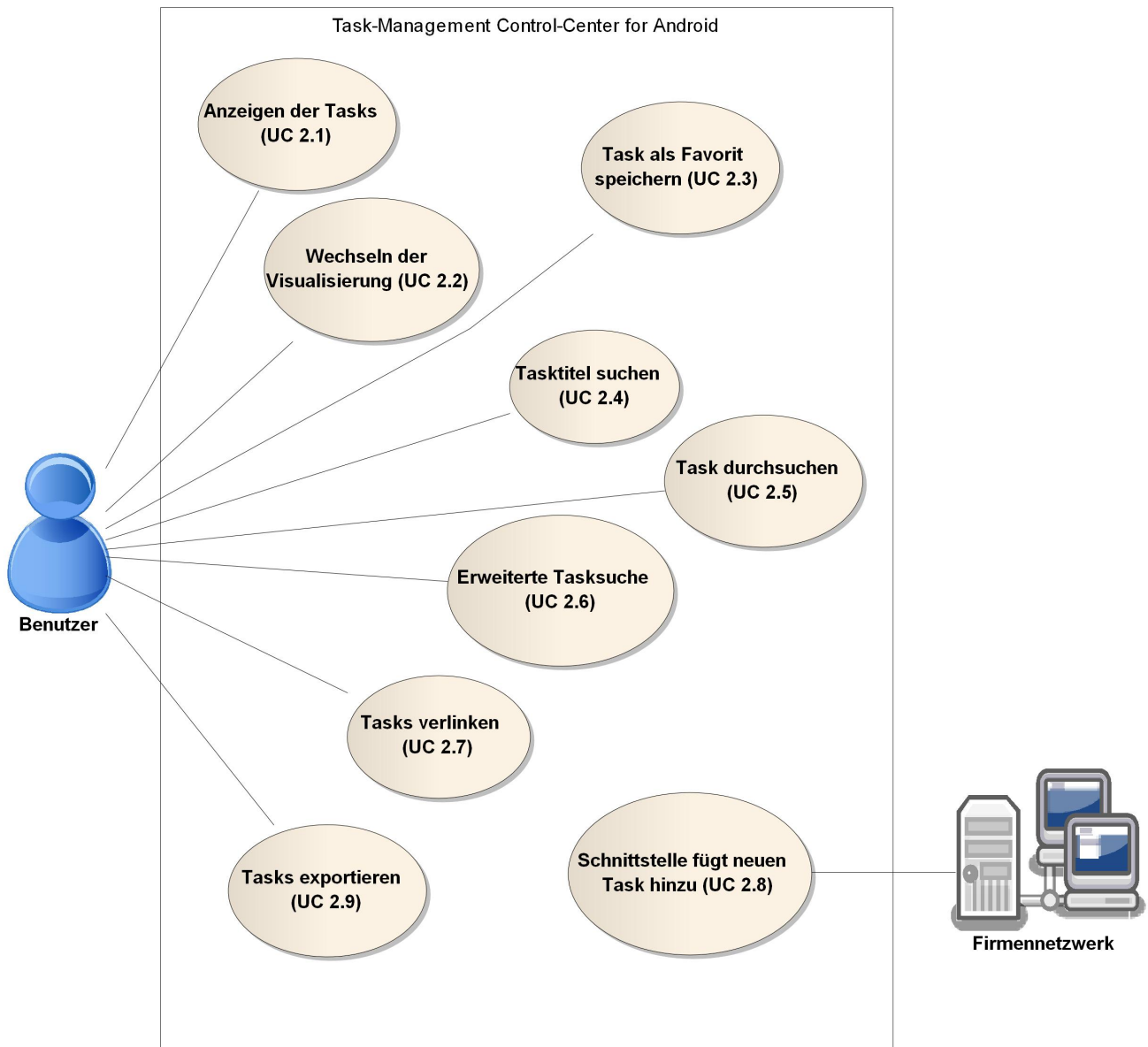


Abbildung 2.2: UC-Diagramm Tasks

2.2.3 Verbindung und Konfiguration

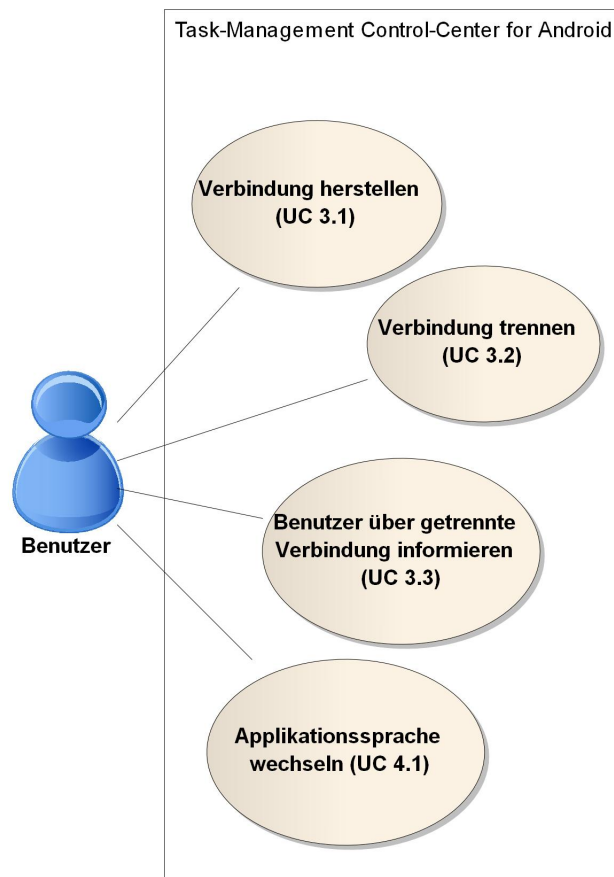


Abbildung 2.3: UC-Diagramm Verbindung und Konfiguration

2.3 Use Cases brief

Die letzten beiden Zahlen einer UC-Überschrift, durch einen Punkt getrennt, entsprechen jeweils der UC-Nummer (2.3.x.y - Hier entspricht x.y der UC-Nummer).

2.3.1 Dateisynchronisation

2.3.1.1 Homeverzeichnis festlegen

Es wird ein Homeverzeichnis für die Dateisynchronisation festlegen (auf PC- sowie Mobile-Device-Seite).

2.3.1.2 Zu synchronisierende Ordner festlegen

Die Ordner, deren Inhalt synchronisiert werden soll, werden relativ zum jeweiligen Homeverzeichnis oder absolut festgelegt. Um zwei Ordner zu synchronisieren, muss mindestens derjenige auf der PC-Seite angegeben werden.

2.3.1.3 Ordnerinhalte synchronisieren

Der Ordnerinhalt der angegebenen Verzeichnisse wird synchronisiert. Falls auf dem Mobile-Device kein Ordner angegeben wird, wird die Ordnerstruktur der PC-Seite übernommen. Für jede synchronisierte Datei wird eine Nachricht, welche die Pfadangaben enthält, an die Schnittstelle gesendet.

2.3.1.4 Ordner von der Synchronisation ausschliessen

Ein Ordner wird auf der Synchronisationsliste auskommentiert und somit temporär von der Synchronisation ausgeschlossen.

2.3.2 Tasks

Für create, read, update, delete (CRUD)-Operationen wird auf die vom Task-Management-Framework on Smart-Phone (TaMaF) zur Verfügung gestellten Funktionen zurückgegriffen. Erweiterungen der Taskeigenschaften werden nicht als eigene UCs aufgeführt, bei Bedarf jedoch vorgenommen. Besteht eine Verbindung zwischen dem Mobile-Device und dem PC, werden die Tasks automatisch und fortlaufend vom PC zum Mobile-Device synchronisiert. Beim Verbindungsaufbau werden diese in beide Richtungen ausgetauscht. Jeder synchronisierte Task wird zudem an die Schnittstelle gesendet.

2.3.2.1 Anzeigen der Tasks

Die Tasks werden entsprechend der gewählten Visualisierung («Baum»- oder «Zeitstrahl»-Darstellung) angezeigt. Der Pfad eines Tasks ist dabei immer ersichtlich. Für Details zur Darstellung siehe «GUI-Design».

2.3.2.2 Wechseln der Visualisierung

Die Darstellung der Tasks wird entsprechend gewechselt, selektierte Tasks bleiben markiert und werden weiterhin angezeigt.

2.3.2.3 Task als Favorit speichern

Ein Eintrag wird zur Favoritenliste hinzugefügt (analog der Favoritenliste im Windows Explorer).

2.3.2.4 Tasktitel suchen

Es wird inkrementell nach einem bestimmten Tasktitel gesucht.

2.3.2.5 Task durchsuchen

Der Inhalt des offenen Tasks wird inkrementell durchsucht.

2.3.2.6 Erweiterte Tasksuche

Folgende Parameter können für die Suche eingestellt werden:

- Pfad: Alle Tasks und Subtasks im angegebenen Pfad werden durchsucht.
- Titel/Inhalt: Es werden entweder nur die Titel oder aber der gesamte Taskinhalt mitsamt Titel durchsucht. Die Suche ist wiederum inkrementell.

2.3.2.7 Tasks verlinken

Der selektierte Task wird unidirektional mit einem anderen Task verlinkt (analog eines Shortcuts in Windows).

2.3.2.8 Schnittstelle fügt neuen Task hinzu

Der von der Schnittstelle kommende Task wird auf dem PC hinzugefügt. Dieser übernimmt wie bei jedem neuen Task das Weiterleiten an das Mobile-Device.

2.3.2.9 Tasks exportieren

Tasks eines definierten Zeitraums werden serialisiert und in ein File gespeichert.

2.3.3 Verbindung

2.3.3.1 Verbindung herstellen

Eine Verbindung zwischen dem Mobile-Device und dem PC wird hergestellt. Der UC 1.3 wird ausgeführt und die Tasks synchronisiert.

2.3.3.2 Verbindung trennen

Eine Verbindung zwischen dem Mobile-Device und dem PC wird getrennt. Optional wird der UC 1.3 ausgeführt. In jedem Fall werden aktive Tasksynchronisationen beendet und die Verbindung zur Schnittstelle unterbrochen.

2.3.3.3 Benutzer über getrennte Verbindung informieren

Will der Benutzer, nach dem der UC 3.2 ausgeführt wurde, weiterarbeiten, so wird er vom System über die fehlende Verbindung zum Mobile-Device informiert. Dies geschieht, sobald ein Task hinzugefügt oder verändert wird.

2.3.4 Konfiguration

2.3.4.1 Applikationssprache wechseln

Die Sprache des GUIs wird entsprechend der Auswahl angepasst.

2.4 Schnittstelle

Das TCA stellt eine Schnittstelle zur Verfügung, um synchronisierte Tasks und Dateien an ein Firmennetzwerk zu melden. Meldungen von der Schnittstelle an das Firmennetzwerk werden in ein File geschrieben.

2.4.1 Dateien

TCA übermittelt die Dateinamen und Pfadinformationen, sowohl den Pfad auf dem Mobile-Device als auch den Pfad auf dem PC, an die Schnittstelle zum Firmennetzwerk.

Ein Übermitteln dieser Daten vom Firmennetzwerk an TCA wird nicht benötigt. Zum Synchronisationszeitpunkt kann TCA selbständig erkennen, welche Dateien verändert wurden.

2.4.2 Tasks

Wird ein Task erstellt, verändert oder gelöscht, wird die Operation zusammen mit dem Task der Schnittstelle übergeben.

Die Operationen `new(task, path)`, `update(task, path)` und `delete(task, path)` führen die entsprechende Operation auf der PC-Applikation aus. Der Pfad spezifiziert dabei, wo der Task liegt bzw. liegen soll. Diese drei Funktionen sind an der Schnittstelle verfügbar und können von aussen aufgerufen werden.

2.5 Anpassungen an bestehende Frameworks

2.5.1 Task-Management-Framework on Smart-Phone

Taskeigenschaften

Die Tasks werden mit folgende Eigenschaften ergänzt:

- Estimated Time: Entspricht einer Schätzung des Zeitaufwandes zur Erledigung des Tasks.
- Effective Time: Entspricht der effektiv verwendeten Zeit zur Erledigung des Tasks.
- Sortierungsnummer: Wird zur Sortierung gleichpriorisierter Tasks innerhalb eines Ordners verwendet.
- Task-Links: Eine Liste von «Shortcuts» zu anderen Tasks.
- Child-List: Eine Liste von allen zugehörigen Subtasks zu einem bestimmten Task.

Die Taskpriorisierung wird neu alphanumerisch sein (anstatt rein numerisch).

2.5.2 Android Control Framework

Klasse DataCollection

Die Klasse `DataCollection` wird um das Attribut `fileSize` erweitert, damit detektiert werden kann, ob die Synchronisation erfolgreich war.

3 Nichtfunktionale Anforderungen

3.1 Funktionalität

Das TCA bietet die Möglichkeit, über ein einfach zu bedienendes GUI Tasks und Dateien zu verwalten. Details können dem Dokument «Vision» entnommen werden.

3.2 Zuverlässigkeit

Wird ein aktiver Synchronisationsprozess unterbrochen, z.B. bei einem Verbindungsverlust durch Trennen der USB-Verbindung, darf kein inkonsistenter Zustand entstehen. Nicht vollständig synchronisierte Dateien müssen gelöscht werden, für die Behandlung der Tasks sei auf TaMaF verwiesen. Nach einem erneuten Verbindungsaufbau müssen die Dateien der fehlgeschlagenen Synchronisation erneut synchronisiert werden können.

3.3 Erlernbarkeit

Das GUI wird so gestaltet, dass es intuitiv benutzbar ist. Es sollen keine umfangreichen Schulungen benötigt werden, um die Benutzung des Programms zu erlernen. Falls trotzdem Unklarheiten auftauchen sollten, kann der «User Guide» zurate gezogen werden.

3.4 Wartbarkeit

TCA muss zu jedem Zeitpunkt erweiterbar beziehungsweise veränderbar sein, damit zusätzliche Anforderungen und Anpassungen aufgrund Kundenwünschen jederzeit in die Applikation eingebunden werden können.

3.5 Effizienz

Das System muss Daten in $O(n)$ synchronisieren können. Einfügeoperationen müssen in $O(\log(n))$ und Sortieroperationen in $O(n\log(n))$ erfolgen.

3.6 Unterstützte Systeme

TCA unterstützt Android-Betriebssysteme ab Version 2.1 Update 1, beziehungsweise SDK 7 und Java ab Version 1.6. Auf der PC-Seite werden folgende Betriebssysteme unterstützt:

- Windows 7
- Linux Fedora 14

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	03.03.2011	DM, RR	Use Cases erstellt.
1.0rc03	03.03.2011	RR	Kapitel Schnittstelle erstellt.
1.0rc04	04.03.2011	DM	Kapitel NF Anforderungen erstellt.
1.0rc05	10.03.2011	DM	Anpassungen gemäss Sitzung vom 10.03.2011.
1.0rc06	17.03.2011	RR	Anpassungen gemäss Sitzung vom 17.03.2011.
1.0	17.03.2011	RR	Akzeptierte Version 1.0
1.1rc01	26.04.2011	RR	UC 3.3 hinzugefügt und Diagramm angepasst.
1.1	17.03.2011	DM	Akzeptierte Version 1.1
1.3rc01	10.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.3	10.06.2011	RR	Akzeptierte Version 1.3

Tabelle 3.1: Revisionshistorie

Abkürzungsverzeichnis

AnCoF Android Control Framework

Ba Bachelorarbeit

CRUD create, read, update, delete

GUI Graphical User Interface

PC Personal Computer

SDK Software Development Kit

TCA Task-Management Control-Center for Android

TaMaF Task-Management-Framework on Smart-Phone

UC Use Case

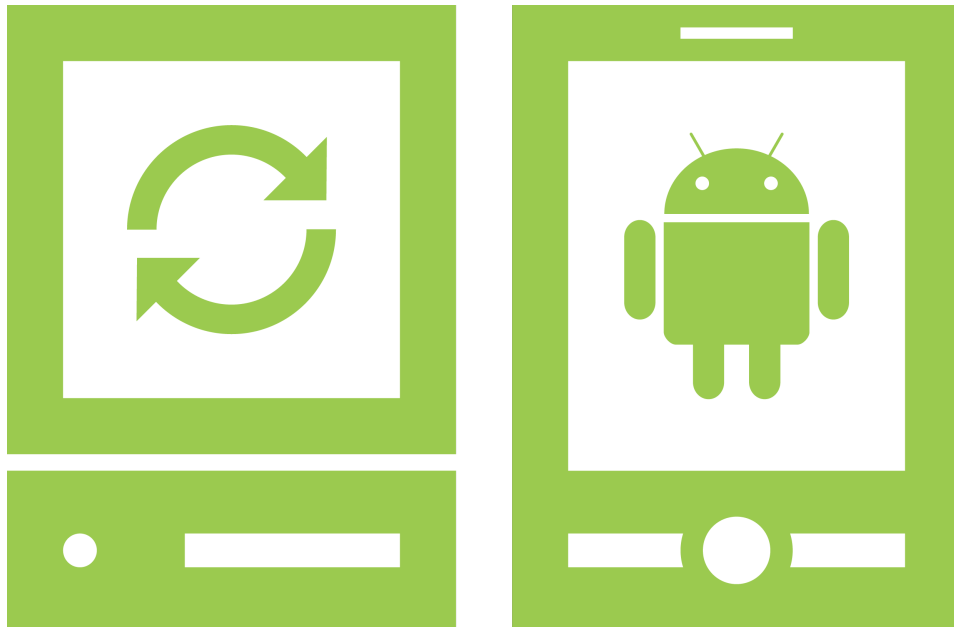
USB Universal Serial Bus

Abbildungsverzeichnis

2.1	UC-Diagramm Dateisynchronisation	2
2.2	UC-Diagramm Tasks	3
2.3	UC-Diagramm Verbindung und Konfiguration	4

Tabellenverzeichnis

3.1 Revisionshistorie	9
---------------------------------	---



Domainanalyse

«Task-Management Control-Center for Android»

Version 1.3, 10. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck	1
1.2 Gültigkeitsbereich	1
2 Domainmodel	2
2.1 Strukturdiagramm	2
2.2 Konzeptbeschreibung	3
3 System-Sequenz-Diagramm (SSD)	6
3.1 Dateisynchronisation	6
3.1.1 UC 1.3 Ordnerinhalte synchronisieren	6
3.2 Tasks	7
3.2.1 UC 2.8 Schnittstelle fügt neuen Task hinzu	7
3.3 Verbindung	7
3.3.1 UC 3.1 Verbindung herstellen	7
3.3.2 UC 3.2 Verbindung trennen	8
3.4 Erweiterung bestehender Frameworks	9
3.4.1 Android Control Framework	9
4 Operation Contracts	10
4.1 syncFiles	10
4.2 usePCFolderStructure	10
4.3 doSync	11
4.4 fileSynchronized	11
4.5 fileSynchronizedInfo	12
4.6 addTask	12
4.7 mobileConnected	13
4.8 disconnectMobile	13

1 Einführung

1.1 Zweck

Siehe «Projektplan», Kapitel 4.2.2, Seite 7.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Domainmodel

2.1 Strukturdiagramm

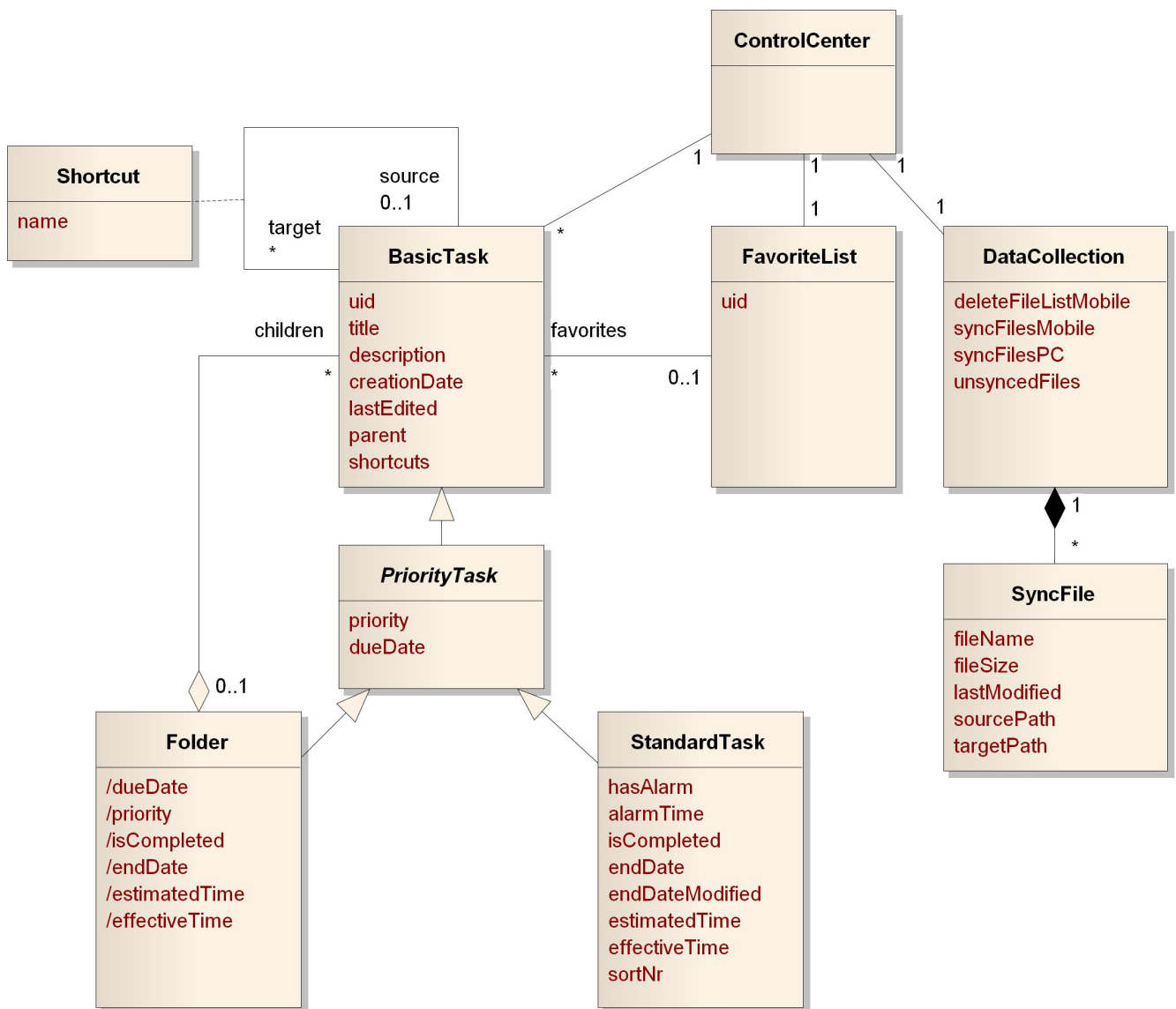


Abbildung 2.1: Domain Model

2.2 Konzeptbeschreibung

ControlCenter

Das Objekt ControlCenter stellt die Funktionen zur Verfügung, welche für die Synchronisation von Dateien und Tasks gebraucht werden.

Attribute	– –
Beziehungen	Benutzt BasicTask und DataCollection für die Synchronisation und enthält zudem eine FavoriteList.

Tabelle 2.1: ControlCenter

DataCollection

Details zur DataCollection können der Dokumentation von Android Control Framework (AnCoF) entnommen werden. Die Klasse wird nur zur Übersicht dargestellt.

SyncFile

Hilfsklasse, welche alle relevanten Informationen zur Übermittlung einer Datei enthält.

Attribute	fileName	Dateiname (inklusive Dateiondung)
	fileSize	Dateigrösse
	lastModified	Bearbeitungsdatum
	sourcePath	Pfad der Dateiquelle
	targetPath	Pfad des Zielorts der Datei
Beziehungen	Ist Teil einer DataCollection.	

Tabelle 2.2: SyncFile

BasicTask

Basisklasse, welche die wichtigsten Felder eines Tasks enthält und die Verschachtelung derselben ermöglicht.

Attribute	uid	Eindeutige Identifikationsnummer
	title	Überschrift des Tasks (muss innerhalb eines Ordners eindeutig sein)
	description	Notiz (muss vom Benutzer nicht ausgefüllt werden)
	creationDate	Erstelldatum
	lastEdited	Bearbeitungsdatum
	parent	Referenz auf übergeordneten Task (falls es sich hierbei um einen «Root»-Task handelt, kann «parent» auch null sein)
	shortcuts	Links auf andere BasicTasks
Beziehungen	Benutzt Shortcuts.	

Tabelle 2.3: BasicTask

Shortcut

Assoziationsklasse, welche das Erstellen von Shortcuts ermöglicht.

Attribute	name	Bezeichnung des Shortcuts
Beziehungen	Ein Shortcut besitzt jeweils eine source sowie ein target, welche auf einen BasicTask referenzieren.	

Tabelle 2.4: Shortcut

PriorityTask

Ein BasicTask, welcher eine Priorität und eine Fälligkeit haben kann. Da er nie als eigener Task verwendet wird, ist der PriorityTask abstrakt.

Attribute	priority	Priorität des Tasks
	dueDate	Fälligkeitsdatum und -zeit
Beziehungen	Ist ein BasicTask.	

Tabelle 2.5: PriorityTask

Folder

Dient als Behälter für BasicTasks und ermöglicht eine hierarchische Datenstruktur. Da der Folder nur als Strukturierungselement dient, enthält er keine spezifischen Taskinformationen.

Attribute	dueDate	Ist das aktuellste Datum der darin enthaltenen BasicTasks. Falls der Folder keine BasicTasks enthält, wird das manuell gesetzte dueDate verwendet.
	priority	Priorität des Folders. Falls nicht manuell gesetzt, wird die höchste Priorität der darin enthaltenen BasicTasks genutzt.
	isCompleted	Wird gesetzt, falls alle darin enthaltenen Tasks als erledigt markiert wurden.
	endDate	Entspricht dem endDate des zuletzt beendeten, darin enthaltenen BasicTasks.
	estimatedTime	Wird aus den Informationen der darin enthaltenen BasicTasks berechnet.
	effectiveTime	Wird aus den Informationen der darin enthaltenen BasicTasks berechnet.
Beziehungen	Ist ein PriorityTask und enthält BasicTasks als «children».	

Tabelle 2.6: Folder

StandardTask

Ein Task, der spezifische Angaben enthält.

Attribute	hasAlarm	Ist gesetzt, wenn ein Alarm ausgelöst werden soll.
	alarmTime	Zeit in Minuten vor der Fälligkeitszeit
	isCompleted	Bearbeitungsstatus
	endDate	Wann der Task beendet wurde (wird vom System gespeichert und ist unveränderlich)
	endDateModified	Wann der Task beendet wurde (kann manuell verändert werden)
	estimatedTime	Geschätzte Bearbeitungszeit
	effectiveTime	Tatsächlich verwendete Bearbeitungszeit
	sortNr	Hilfsvariable zur Sortierung der Tasks
Beziehungen	Ist ein PriorityTask.	

Tabelle 2.7: StandardTask

FavoriteList

Eine Hilfsklasse, die das Übertragen der Favoriten an das Mobile-Device ermöglicht.

Attribute	uid	Eindeutige Identifikationsnummer (wird gebraucht um die Übertragung zu ermöglichen)
Beziehungen	Enthält BasicTasks.	

Tabelle 2.8: FavoriteList

3 SSD

3.1 Dateisynchronisation

3.1.1 UC 1.3 Ordnerinhalte synchronisieren

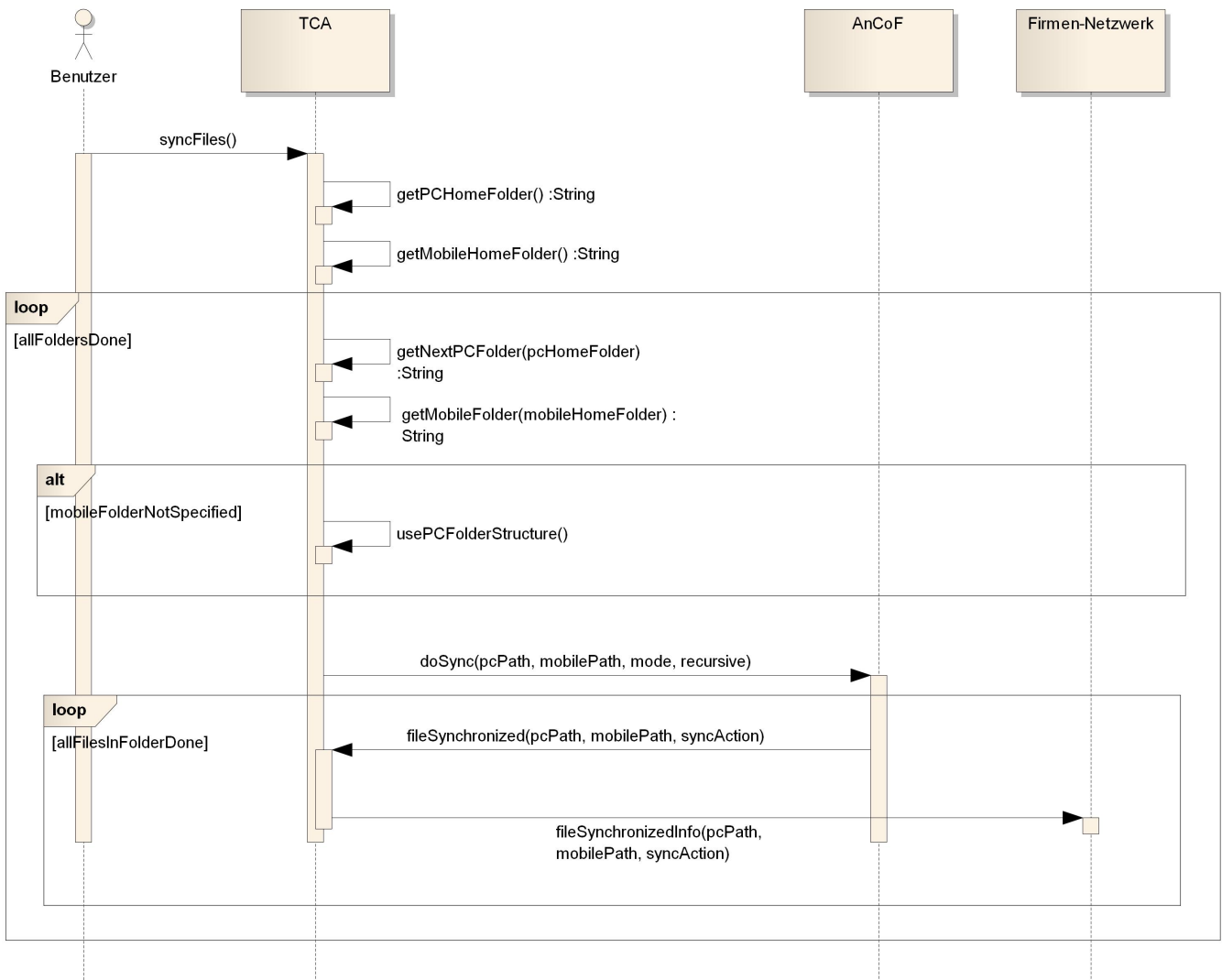


Abbildung 3.1: Ordnerinhalte synchronisieren

3.2 Tasks

3.2.1 UC 2.8 Schnittstelle fügt neuen Task hinzu

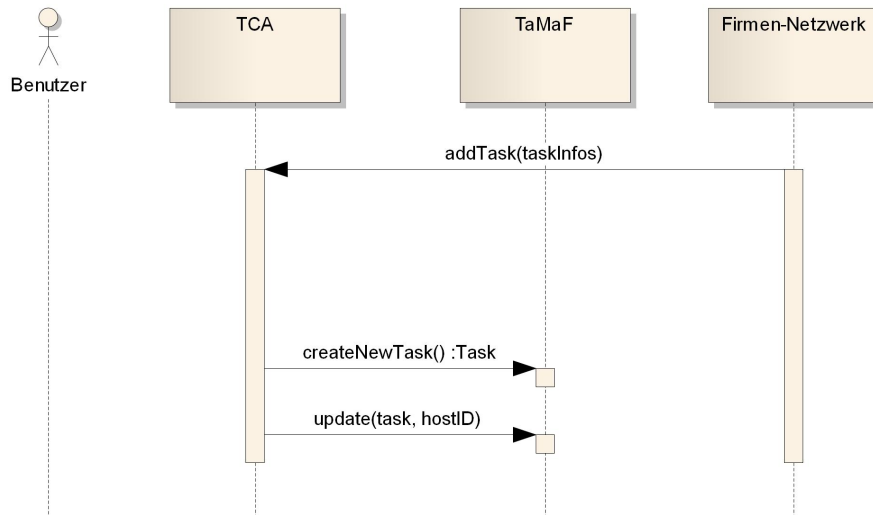


Abbildung 3.2: Schnittstelle fügt neuen Task hinzu

3.3 Verbindung

3.3.1 UC 3.1 Verbindung herstellen

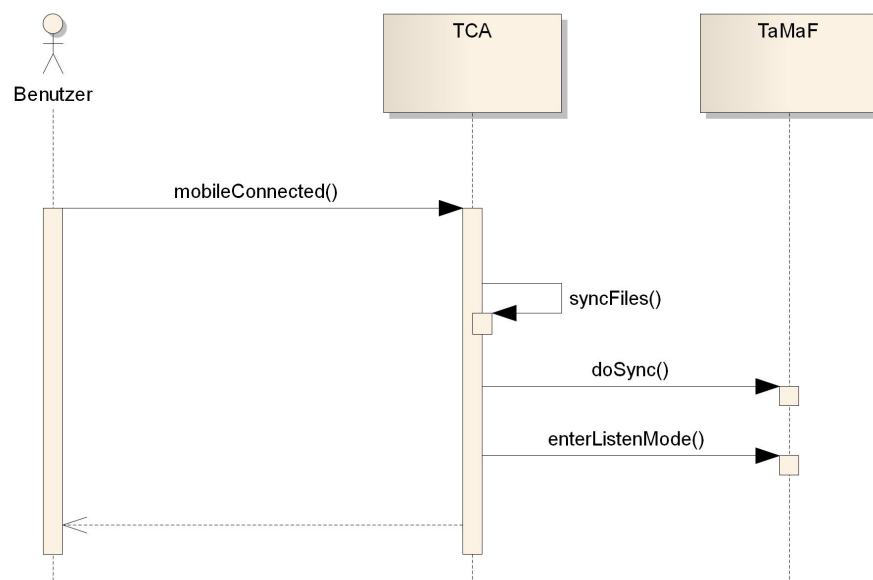


Abbildung 3.3: Verbindung herstellen

3.3.2 UC 3.2 Verbindung trennen

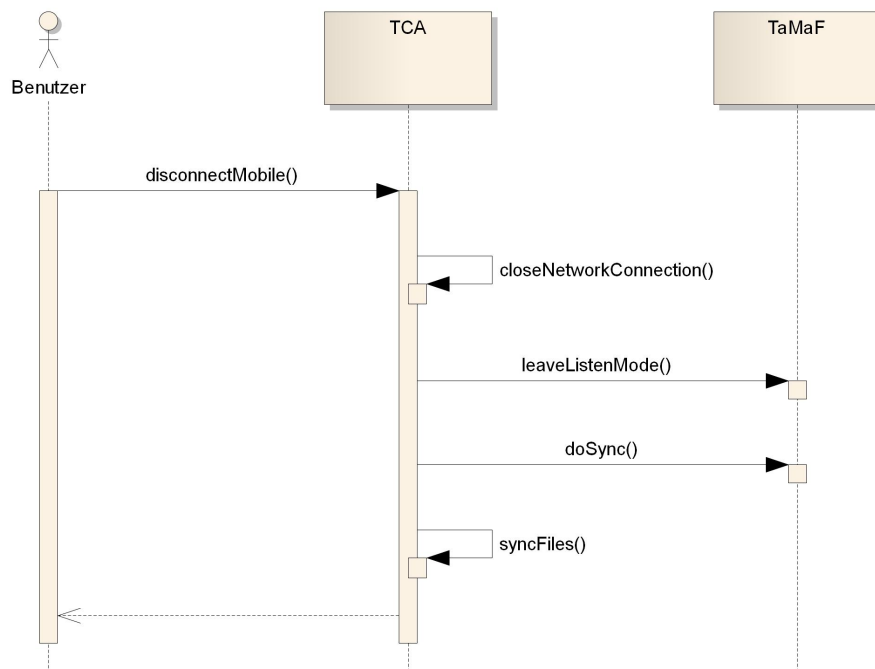


Abbildung 3.4: Verbindung trennen

3.4 Erweiterung bestehender Frameworks

3.4.1 Android Control Framework

Das AnCoF wird um die grün markierten Funktionen erweitert. Diese erlauben es, vor der Synchronisation zu überprüfen, ob der Pfad des Mobile-Devices existiert und erstellen diesen gegebenenfalls.

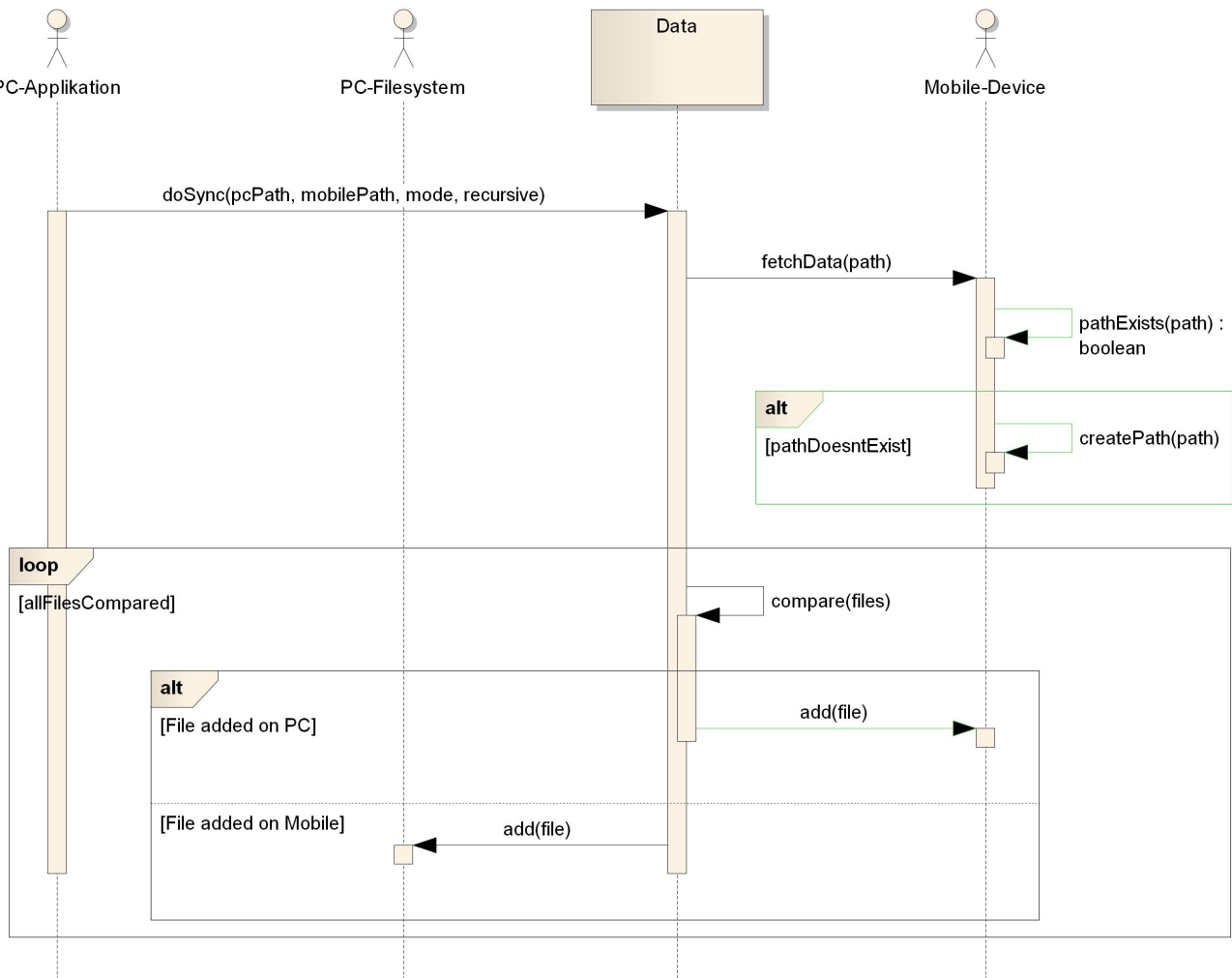


Abbildung 3.5: AnCoF, Synchronisieren

4 Operation Contracts

Die letzte Zahl einer Operation Contract (OC)-Überschrift entspricht jeweils der OC-Nummer (4.x - Hier entspricht x der OC-Nummer).

4.1 syncFiles

Operation	syncFiles()
Cross References	UC 1.3
Preconditions	<ul style="list-style-type: none"> • Das Homeverzeichnis auf dem PC sowie dem Mobile-Device ist angegeben. • Es ist mindestens ein Ordner auf der PC-Seite angegeben.
Postconditions	<ul style="list-style-type: none"> • Der Ordnerinhalt der angegebenen Verzeichnisse wurde synchronisiert.

Tabelle 4.1: Operation Contract 1

4.2 usePCFolderStructure

Operation	usePCFolderStructure()
Cross References	UC 1.3
Preconditions	<ul style="list-style-type: none"> • Es ist kein Ordner auf der Mobile-Device-Seite angegeben.
Postconditions	<ul style="list-style-type: none"> • Der Ordner beziehungsweise die Ordnerstruktur wurde entsprechend angelegt.

Tabelle 4.2: Operation Contract 2

4.3 doSync

Operation	doSync(pcPath, mobilePath, mode, recursive)
Cross References	UC 1.3
Preconditions	<ul style="list-style-type: none"> • Der Ordner unter «pcPath» existiert.
Postconditions	<ul style="list-style-type: none"> • Die Pfadangabe des Ordners «mobilePath» auf der Mobile-Device-Seite existiert. • Die Ordnerinhalte von «pcPath» und «mobilePath» wurden synchronisiert.

Tabelle 4.3: Operation Contract 3

4.4 fileSynchronized

Operation	fileSynchronized(pcPath, mobilePath, syncAction)
Cross References	UC 1.3
Preconditions	<ul style="list-style-type: none"> • Die Ordner auf der PC- und der Mobile-Device-Seite existieren. • Die Datei wurde bereits ausgetauscht.
Postconditions	–

Tabelle 4.4: Operation Contract 4

4.5 fileSynchronizedInfo

Operation fileSynchronizedInfo(pcPath, mobilePath, syncAction)

Cross References UC 1.3

Preconditions

- Die Datei wurde synchronisiert.

Postconditions

- Die Dateiinformationen wurden an die Schnittstelle weitergereicht.
-

Tabelle 4.5: Operation Contract 5

4.6 addTask

Operation addTask(taskInfos)

Cross References UC 2.7

Preconditions

- Ein neuer Task wurde von der Firmen-Netzwerk-Schnittstelle empfangen.

Postconditions

- Der Task wurde auf dem PC hinzugefügt.
-

Tabelle 4.6: Operation Contract 6

4.7 mobileConnected

Operation mobileConnected()

Cross References UC 3.1

Preconditions

- Das Mobile-Device und der PC sind eingeschaltet.
- Das Mobile-Device und der PC sind physisch, z.B. durch ein USB-Kabel, miteinander verbunden.

Postconditions

- Es wurde eine Verbindung hergestellt.
 - Der Benutzer wurde über den Status der Verbindungsherstellung benachrichtigt.
 - Alle Tasks und definierten Ordnerinhalte wurden synchronisiert.
 - TCA befindet sich im «Listenmode».
-

Tabelle 4.7: Operation Contract 7

4.8 disconnectMobile

Operation disconnectMobile()

Cross References UC 3.2

Preconditions

- Das Mobile-Device und der PC sind verbunden.

Postconditions

- Die Verbindung wurde getrennt.
 - Der Benutzer wurde über den Status der Verbindungstrennung benachrichtigt.
 - Die Verbindung zur Netzwerkschnittstelle wurde geschlossen.
 - Alle Tasks und Dateien wurden vollständig synchronisiert.
-

Tabelle 4.8: Operation Contract 8

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	07.03.2011	DM, RR	Kapitel Domainmodel und SSD erstellt.
1.0rc03	10.03.2011	DM, RR	Domainmodell angepasst.
1.0rc04	11.03.2011	DM	Operation Contracts erstellt.
1.0rc05	17.03.2011	DM, RR	Anpassungen gemäss Sitzung vom 17.03.2011.
1.0	17.03.2011	RR	Akzeptierte Version 1.0.
1.1rc01	24.03.2011	DM, RR	Anpassungen gemäss Sitzung vom 24.03.2011.
1.1rc02	26.04.2011	RR	Eindeutigkeit von Titeln vermerkt.
1.1rc03	28.04.2011	RR	Domainmodell angepasst.
1.1	28.04.2011	RR	Akzeptierte Version 1.1.
1.2rc01	19.05.2011	RR	Strukturdiagramm gemäss Sitzung angepasst.
1.2	19.05.2011	RR	Akzeptierte Version 1.2.
1.3rc01	10.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.3	10.06.2011	RR	Akzeptierte Version 1.3.

Tabelle 4.9: Revisionshistorie

Abkürzungsverzeichnis

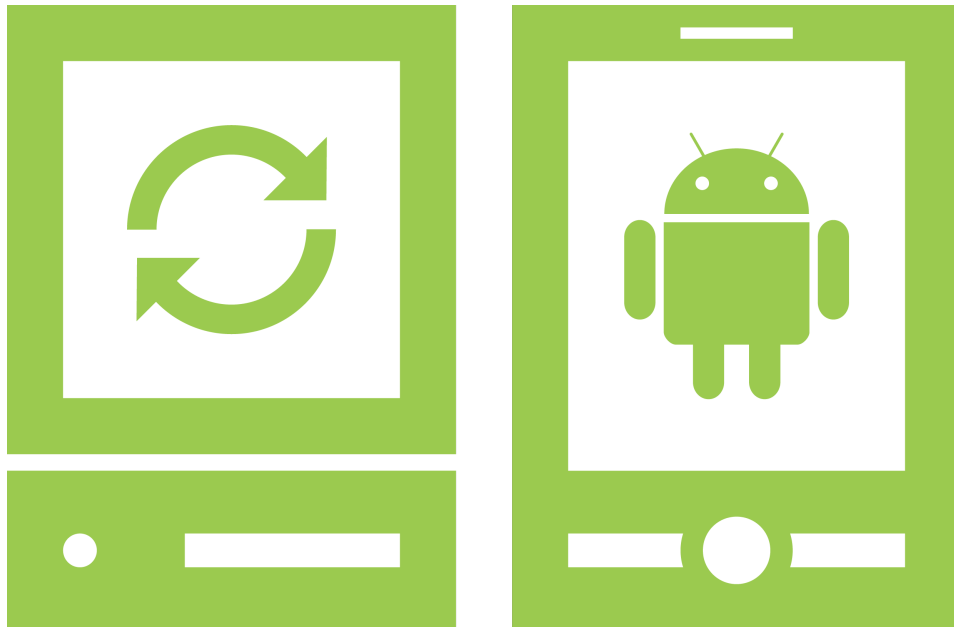
AnCoF Android Control Framework
Ba Bachelorarbeit
OC Operation Contract
PC Personal Computer
SSD System-Sequenz-Diagramm
TCA Task-Management Control-Center for Android
UC Use Case
USB Universal Serial Bus

Abbildungsverzeichnis

2.1	Domain Model	2
3.1	Ordnerinhalte synchronisieren	6
3.2	Schnittstelle fügt neuen Task hinzu	7
3.3	Verbindung herstellen	7
3.4	Verbindung trennen	8
3.5	AnCoF, Synchronisieren	9

Tabellenverzeichnis

2.1	ControlCenter	3
2.2	SyncFile	3
2.3	BasicTask	3
2.4	Shortcut	4
2.5	PriorityTask	4
2.6	Folder	4
2.7	StandardTask	5
2.8	FavoriteList	5
4.1	Operation Contract 1	10
4.2	Operation Contract 2	10
4.3	Operation Contract 3	11
4.4	Operation Contract 4	11
4.5	Operation Contract 5	12
4.6	Operation Contract 6	12
4.7	Operation Contract 7	13
4.8	Operation Contract 8	13
4.9	Revisionshistorie	14



GUI Design

«Task-Management Control-Center for Android»

Version 1.2, 10. Juni 2011

Inhaltsverzeichnis

1	Einführung	1
1.1	Zweck des Dokuments	1
1.2	Gültigkeitsbereich	1
2	Persona	2
2.1	Paulo Preciso	2
2.2	Renata Rapida	3
2.3	Isidoro Impaziente	4
2.4	Danilo Desinteressato	5
3	Soll Szenarios	6
3.1	Notfalldienst für Heizölbestellungen	6
3.2	Der eilige Vorgesetzte	6
3.3	Taskerfassung für Mitarbeiter	6
3.4	Notfall zu Hause	7
3.5	Besichtigung der Infrastruktur	7
3.6	Während der Heimfahrt	7
3.7	Alles dabei	7
4	Paperprototype	8
4.1	Allgemeine Darstellungsentscheide	8
4.2	PC-Applikation	9
4.2.1	Explorer-View	9
4.2.2	Zeitstrahl-View	10
4.2.3	Task ansehen	11
4.2.4	Task erstellen oder ändern	12
4.2.5	Shortcuts von Task oder Ordner editieren	13
4.2.6	Suchresultate	14
4.3	Mobile-Device	15
4.3.1	Startbildschirm	15
4.3.2	Taskübersicht Explorer-View	15
4.3.3	Taskansicht Zeitstrahl-View	16
4.3.4	Taskansicht	16
4.3.5	Task erstellen oder ändern	17
4.3.6	Timestamp erzeugen	18
4.3.7	Ordner erstellen oder ändern	18
4.3.8	Standard Suche	19
4.3.9	Erweiterte Suche	19

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 7.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Persona

2.1 Paulo Preciso

- 58 Jahre
- Kaufmännische Lehre
- Buchhalter eines Transport- und Ölhandelsunternehmens
- ledig, kinderlos
- seit 20 Jahren als Buchhalter tätig
- gute Windows- und Officekenntnisse

Paulo erledigt die Buchhaltung in einem kleinen Transport- und Ölhandelsunternehmen. Für diesen Aufgabenbereich ist er alleine verantwortlich, benötigt jedoch immer die aktuellsten Daten des Einkäufers, des Verkaufsteams und der Chauffeure für die Verrechnung.

Ganz selten arbeitet er von zu Hause aus, meist an Wochenenden, wenn kurzfristig viele Aufträge zu erwarten sind. In solchen Situationen kann es vorkommen, dass er das Verkaufsteam unterstützen muss. Daher besitzt er einen firmeneigenen Laptop, auf dem die nötigen Programme installiert sind und ein Smartphone, um im Notfall erreichbar zu sein. Einen grossen Teil der Arbeit erledigt er jedoch im Betrieb, wo er sich mit dem Einkäufer und einer Sekretärin ein gemeinsames Büro teilt. Dort herrscht oft hektisches Treiben, da regelmässig die Chauffeure vorbeischaun, um sich über den aktuellsten Stand der Aufträge zu informieren.

Um einen reibungslosen Arbeitsablauf zu garantieren, spricht er sich regelmässig mit dem Einkäufer und dem Verkaufsteam ab, damit die Leistungen zur Zeit verrechnet werden können und immer genügend Liquidität vorhanden ist. «Ohne die regelmässigen Sitzungen im Team wäre es kaum möglich, den Überblick über die Auftragslage zu behalten und offene Rechnungsbeträge rechtzeitig einzufordern.»

Paulo ist gut organisiert und protokolliert alle Sitzungen, damit Beschlüsse dauerhaft festgehalten werden können und keine offenen Fragen entstehen. Damit auch andere Teammitglieder jederzeit Auskunft geben können, legt er die Dokumente und Rechnungen in einem Ordner ab, sortiert nach Kundennamen und Datum.

Paulo kennt die Produkte des Transports- und Ölhandelsunternehmens. «Sind neue innovative Lösungen in der Ölbranche auf dem Markt, informiere ich mich jeweils über diese, schliesslich muss ich Bescheid wissen, für was ich Rechnungen ausstelle.» Sind offene Fragen oder Unklarheiten vorhanden, holt er sich die fehlenden Informationen jeweils bei den betreffenden Spezialisten ein.

Oft ärgert sich Paulo über die unorganisierte Datenablage seines Vorgesetzten. Vielfach sind dessen Daten gar nicht oder nur unvollständig abgelegt. Manche Gesprächsnotizen, die er sich während oder kurz nach einem Kundengespräch macht, sind dann nur auf dem Smartphone des Vorgesetzten vorhanden und für Paulo unzugänglich. «Ich wünschte mir ein Tool, welches Daten auf dem Smartphone automatisch mit jenen auf dem PC synchronisiert, sobald dieses mit dem PC verbunden wird. So würde mein Vorgesetzter sofort daran erinnert werden, seine Daten in die Datenablage einzuspeisen, sobald er sein Smartphone zwecks Stromaufladung an den PC anschliesst.»

2.2 Renata Rapida

- 40 Jahre
- Kaufmännische Lehre
- Sekretärin einer Bauunternehmung
- verheiratet, Mutter zweier Kinder (3 und 5)
- seit 1 Jahr als Abteilungssekretärin tätig
- gute Officekenntnisse, was Textverarbeitungsprogramme angeht

Renata arbeitet als Abteilungssekretärin in einer grossen Bauunternehmung. Da sie als «Familienmanagerin» nur zu 50% arbeiten kann, teilt sie sich diesen Posten mit einem Kollegen.

Einen grossen Teil der Arbeit erledigt sie vor Ort, wo sie sich mit ihrem Kollegen einen Arbeitsplatz teilt. Da es manchmal vorkommt, dass sie aufgrund ihrer Kinder verspätet oder gar nicht im Betrieb arbeiten kann respektive kurzfristig früher gehen muss, besitzt sie einen Laptop und kann sich über das Virtual Private Network (VPN) ins Firmennetzwerk einloggen. Daher ist es wichtig, dass ihr immer die aktuellsten Daten zur Verfügung stehen.

Als Mutter zweier Kinder im Vorschulalter ist sie es gewohnt, schnell und konzentriert zu arbeiten. Die ihr zur Verfügung stehende Zeit für Hausarbeit und Kinder muss gut eingeteilt werden. Diese Eigenschaft macht sie sich auch im Arbeitsbereich zu nutzen und wird aufgrund dessen von anderen Mitarbeiter sehr geschätzt. Durch ihre gut strukturierte Arbeitsweise unterlaufen ihr selten Fehler und anstehende Arbeiten liefert sie immer termingerecht ab.

Renata ist gut organisiert. Leider lassen sich Familie und Arbeit durch die Doppelbelastung nicht immer klar trennen, was sich darin bemerkbar macht, dass Renata trotz Absprachen mit ihrem Partner und der Kinderbetreuung des öfteren kurzfristig im Betrieb fehlt und dann Sitzungen verschoben werden müssen.

Durch den Arbeitsunterbruch, bedingt durch die «Babypause» verlor Renata den Anschluss an die neuen Technologien. Diesen Nachteil versucht sie in mühseliger Selbstarbeit wieder wettzumachen. Sie liest sich gerne in neue Thematiken ein und kennt mittlerweile die von ihr täglich benutzten Programme wieder bestens. «Ich bin mir dessen bewusst, dass der Rückstand in technischen Fragen, welchen ich durch die Abwesenheit vom Arbeitsmarkt bekommen habe, sich nicht von heute auf morgen wieder wettmachen lässt, doch bemühe ich mich, diesen von Tag zu Tag zu verkleinern.»

Renata ärgert sich manchmal, dass sie nicht immer über VPN aufs Firmennetzwerk zugreifen kann. «Oftmals sind mir die Hände gebunden, wenn ich in den Abendstunden Zeit hätte, mich um liegengebliebene Arbeiten zu kümmern. Denn unsere Internetverbindung ist leider nicht immer funktionstüchtig.»

2.3 Isidoro Impaziente

- 31 Jahre
- Ausbildung zum Informatiker mit Berufsmittelschule (BMS)
- Kundenberater eines Informatikbüros
- Systemverantwortlicher bei einigen Klein- und Kleinstfirmen
- ledig, in Partnerschaft lebend
- seit 3 Jahren als Kundenberater tätig, zuvor 2 Jahre als Systemadministrator bei der selben Firma
- gutes technisches Wissen, gute Mac- und Windowskenntnisse, sehr gute Linux-Kenntnisse

Isidoro ist einer von drei Kundenberater des Betriebs. Kundentermine werden im Normalfall vom Sekretariat organisiert und den Kundenberatern zugewiesen. Isidoro hat feste Kunden, die er schon seit längerem betreut, er führt jedoch auch Verkaufsgespräche mit potenziell neuen Kunden.

Die Beratungsgespräche finden fast immer (zu ca. 90%) direkt beim Kunden statt. Für seine Arbeit steht ihm ein Laptop zur Verfügung, welchen er während den Kundengesprächen nutzt. Da allerdings einige der Aufträge direkt vor Ort, zum Beispiel in einem Serverraum, besprochen werden müssen, kann der Laptop nicht überall genutzt werden. In diesen Fällen arbeitet Isidoro mit einem herkömmlichen Notizblock oder einem Diktiergerät und tippt die Notizen danach ab.

Die zusätzliche Aufgabe als Systemverantwortlicher bereitet ihm oft Probleme, da er ständig unterwegs ist. Um die Systeme regelmässig zu überprüfen, nutzt er eine Funktion, welche es ermöglicht den Systemstatus per Telefon abzuhören. Gerne würde er dies im Auto erledigen, da er jedoch während dem Fahren keine Nummer eingeben und auch nicht hinterher nach der «Austaste» suchen will, macht er diese Arbeit vorwiegend auf Raststätten oder Parkplätzen. Unter diesem Umstand leidet die Fahrkonzentration, da keine Zeit übrig bleibt, sich während einer langen Autofahrt die Beine zu vertreten.

Isidoro ist sehr ungeduldig und nicht immer gut organisiert. Für ihn reicht es, wenn er weiss, wo sich was befindet. Einzelne Mitarbeiter haben sich deswegen bereits geärgert, da sich Isidoro selten im Büro aufhält und dann nur telefonisch zu erreichen ist. Zudem ärgert sich Isidoro immer, wenn er im Büro Zeit damit verbringt, die handschriftlichen Notizen abzuschreiben. Auch die vielen Hilfsmittel wie Laptop, Telefon, Notizblock, Diktiergerät, etc., die er bei einem Kundengespräch mitnehmen muss, ärgern ihn: «Ich möchte nicht immer mit einer Wagenladung voller Geräte ankommen. Es sollte ein Hilfsmittel geben, welches klein genug ist, damit ich es überall hin mitnehmen kann, aber dennoch gross genug, damit es alle erforderlichen Funktionen erfüllt.»

Neue technische Geräte findet Isidoro faszinierend und kann sich deshalb auch schnell an neue Programme anpassen. Er ist der Meinung, dass ein Computerprogramm, welches sich nicht in den ersten fünf Minuten grundsätzlich bedienen lässt, nicht brauchbar ist. Zudem möchte er die wichtigsten Funktionen mit möglichst wenigen Klicks benutzen können. Als überzeugter Linux-Nutzer möchte er, dass ein Programm sowohl auf Linux, wie auch auf anderen Systemen verfügbar ist, damit er seinen Kunden Programme empfehlen kann, die er selbst bereits getestet hat und bestens kennt.

2.4 Danilo Desinteressato

- 47 Jahre
- Ingenieurstudium im Bereich Elektrotechnik
- Besitzer eines kleinen Unternehmens mit 25 Mitarbeitern
- verheiratet, 3 fast erwachsene Kinder (16, 18 und 21)
- arbeitet sowohl an Aufträgen als auch als Repräsentant der Firma.
- sehr gutes technisches Fachwissen, gute Windowskenntnisse, gute Kenntnisse der für die Arbeit relevanten Programmen

Danilo hat die Firma vor 19 Jahren als kleinen Dienstleistungsbetrieb mit zwei Mitarbeitern gegründet. Bis heute ist die Firma auf 25 Mitarbeiter angewachsen, die alle ihre eigenen Projekte betreuen und an anderen Projekten mitarbeiten. Er selbst arbeitet auch an diversen Projekten mit und springt dort ein, wo «Not-am-Mann» ist, da er die einzelnen Arbeiten sehr gut kennt. Als Geschäftsinhaber und Repräsentant der Firma hat er viele Kundenbesuche, welche spontan stattfinden können.

Für Kundenbesuche nimmt er nebst Unterlagen auf Papier immer einen Laptop, vielfach einen Prototypen, an dem gearbeitet wird und meist noch einen Beamer mit. Zudem besitzt er ein Smartphone, welches er im Moment ausschliesslich zum telefonieren verwendet.

Seine Freizeit verbringt er am liebsten mit seiner Familie. Während dieser Zeit möchte er sich nicht um die Firma oder deren technischen Geräten kümmern. Deshalb dauert es oft sehr lange, bis Danilo alle Funktionen eines neuen «Gadgets» erlernt hat oder er ignoriert diese ganz, wie er dies im Moment mit seinem Smartphone macht.

Danilo ist nicht mehr der jüngste und sieht daher nicht mehr so gut, da er aber eitel ist, möchte er auf eine Lesebrille verzichten. Deshalb ist er froh, wenn er in Programmen die Schriftgrösse oder die Hintergrundfarbe verändern kann, damit er auf dem Bildschirm ohne Anstrengung lesen kann.

In seiner Firma gibt es im Moment mehrere auf Microsoft Excel basierte Lösungen, um Arbeiten und Termine zu koordinieren. Er und einige seiner Mitarbeiter verwenden den in Microsoft Outlook integrierten Kalender, um ihre Termine zu erfassen. Andere Mitarbeiter verwenden jedoch eigene, nicht kompatible Programme, so dass Terminvereinbarungen immer noch mit Notizzetteln oder Mails ausgetauscht werden müssen. Danilo hofft, dieses «Softwaredurcheinander» mit einem Programm ersetzen zu können, um mindestens die anstehenden Aufgaben einheitlich verwalten zu können. «Ich möchte ein Tool, welches es uns ermöglicht, Tasks auszutauschen und diese möglichst einfach an Kundenterminen abzurufen oder neue zu erfassen.»

3 Soll Szenarios

3.1 Notfalldienst für Heizölbestellungen

Paulo hat dieses Wochenende wieder einmal einen seiner ungeliebten Notfalldienste und muss daher eingehende Telefonanrufe entgegennehmen und erforderliche Öltransporte in die Wege leiten. Gerade als er sich am Samstagnachmittag hinsetzen möchte, um die Zeitung zu lesen, klingelt sein Smartphone. Etwas genervt über den Unterbruch meldet er sich und nimmt die Bestellung auf. Dank der «Aufnahmefunktion» der neuen Applikation auf seinem Smartphone kann er die Bestellung für den anstehenden Heizöltransport direkt aufzeichnen. Er ist froh darüber, da er ansonsten die Daten auf seinem Notizblock hätte festhalten müssen, um diese später in das Verrechnungssystem einspeisen zu können. Nun informiert er den diensthabenden Chauffeur über die Bestellung, welche dieser dann prompt zur Zufriedenheit des Kunden ausführt und diesen mit dem dringend benötigten Öl beliefert. Paulo ist froh, dass er dank der neuen Applikation am Montag morgen keine Zettelsammlung mehr im System eingeben muss, da er die aufgenommenen «Audionotizen» durch die automatische Synchronisation seines Smartphones mit seinem PC direkt auf diesem abrufen und entsprechend verrechnen kann. So besteht auch nicht mehr die Gefahr, dass verloren gegangene Zettelchen erst spät oder im schlimmsten Fall gar nicht verrechnet werden.

3.2 Der eilige Vorgesetzte

Es ist Freitagnachmittag und Paulo freut sich bereits auf das anstehende freie Wochenende, als sein Vorgesetzter ins Büro stürmt. Er kommt direkt von einer Sitzung und kaum hat er das Büro betreten, sollte er bereits schon beim nächsten Kundentermin sein. Glücklicherweise muss er seine kostbare Zeit nicht mit langen Erklärungen und Arbeitsanweisungen aufgrund der Sitzungsergebnisse verbringen, denn mittels der «Synchronisationfunktion» seiner Smartphone-Applikation kann er die Gesprächsnotizen direkt bei Paulo auf den PC übertragen. Schon nach wenigen Minuten, in denen er die Unterlagen für die letzte Sitzung dieser Woche zusammensuchen konnte, kann er die Smartphone-Verbindung mit dem PC wieder trennen. Er wünscht allen anwesenden Mitarbeitern noch ein schönes Wochenende und macht sich auf den Weg zum Kunden. Paulo ist erfreut über die erhaltenen Notizen. Ohne die neue Applikation wären wertvolle Minuten mit Erklärungen verloren gegangen und er hätte möglicherweise im Eifer des Gefechts nicht alle wichtigen Informationen korrekt auf einem Notizzettel festhalten können. Dank der erfolgreichen Synchronisation kann sich Paulo nach dem Wochenende am Montagmorgen sofort an die Arbeit machen, die erhaltenen Informationen ins System einzuspeisen und muss nicht erst warten, bis sein Vorgesetzter im Büro erscheint.

3.3 Taskerfassung für Mitarbeiter

Es ist Mittwoch abends, nach einem anstrengenden Tag findet Renata endlich Zeit, sich an liegengebliebene Schreiarbeiten des Betriebs zu machen. Da die Kinder nachmittags frei hatten, war es ihr nicht möglich, ohne Unterbrechungen zu arbeiten. Gerne opfert sie daher die Abendstunden, um das Protokoll der gestrigen Sitzung zu studieren und entsprechende Aufgaben für die Bau- und Projektleiter zu verfassen. Diese werden so im System eingetragen, dass die Mitarbeiter am nächsten Morgen nur ihre Smartphones mit der neuen Applikation verbinden müssen, um die erfassten Tasks sodann mittels der automatischen Synchronisation direkt auf ihren Geräten zu erhalten. Renata muss am nächsten Tag unglücklicherweise zu Hause

bleiben, da ihre Tochter krank ist. Deshalb trifft sich diese Automatisierung besonders gut. Ansonsten hätte sie jeden einzelnen Mitarbeiter telefonisch informieren müssen.

3.4 Notfall zu Hause

Wie jede Woche steht Renata am Donnerstag den ganzen Tag im Betrieb zur Verfügung. Die Kinder werden während dieser Zeit von einer Tagesmutter betreut. Als sie sich gerade daran macht, das Abnahmeprotokoll eines kleinen Umbaus zu schreiben, klingelt ihr Telefon. Ihre Tochter ist während dem Spielen unglücklich gestürzt und muss deswegen zum Arzt. Da Renata die nötigen Audioaufnahmen, welche der Bauleiter auf der Baustelle während der Abnahme gemacht hat, bereits durch die Synchronisation ihres Laptops mit dem Smartphone des Bauleiters erhalten hat, kann sie sich sofort auf den Weg machen. Nach erfolgtem Arzttermin kann sie aufatmen, die Verletzung ihrer Tochter ist glücklicherweise nichts mehr als eine Verstauchung des Fusses. Endlich zu Hause angekommen, kann sich Renata daran machen, das ausstehende Protokoll zu vervollständigen, so dass es am nächsten Morgen an die betroffenen Parteien verschickt werden kann.

3.5 Besichtigung der Infrastruktur

Isidoro ist an einem Kundengespräch. Der Kunde ist begeistert von der Lösung, welche Isidoro präsentiert und unterschreibt sogleich den Arbeitsauftrag. Um keine Zeit zu verlieren, sollen die ersten Arbeiten bereits besprochen und geplant werden. Während der Kunde schnell seinen Terminkalender holen geht, erstellt Isidoro bereits die Ordnerstruktur für den neuen Auftrag mit seinem Laptop. Nun werden einzelne Aufgaben erfasst, welche Isidoro sogleich als Task anlegt. Damit er sich ein genaueres Bild über die Infrastruktur machen kann, führt ihn der Kunde direkt in den Serverraum. Für Isidoro ist dies kein Problem, da er nur sein Smartphone vom Laptop trennen muss und so alle Daten mitnehmen kann. Im Serverraum angekommen, bittet ihn der Kunde sogleich mit der Arbeit zu beginnen. Da Isidoro noch Zeit hat, bevor er zum nächsten Kundentermin muss, willigt er ein. Bevor er beginnt, erstellt er einen Zeitstempel, damit er später noch weiss, wie lange er für die Tätigkeit brauchte und seinen Stundenzettel ohne Probleme ausfüllen kann.

3.6 Während der Heimfahrt

Während Isidoro von einem kurzen Kundengespräch nach Hause fährt, bemerkt er, dass eine der Firmen, deren Server er betreut, gleich an der Strecke liegt. Bereits vor Wochen hat Isidoro die Servicenummer des Systems in sein Smartphone programmiert, welches sich in der Halterung am Armaturenbrett befindet. Er startet den Anruf bequem mit einem Tastendruck. Darauf ertönt kurz der Wählton und anschliessend wird ihm der Systemstatus von einer Computerstimme mitgeteilt. Währenddessen fährt Isidoro auf der Autobahn weiter. Nach den zuvor eingestellten 25 Sekunden wird das Gespräch automatisch unterbrochen.

3.7 Alles dabei

Danilo verbringt seine Ferien gerne zusammen mit seiner Familie, meist unternehmen sie dann kleinere Tagesausflüge. Seine Mitarbeiter wissen, dass er für Notfälle immer erreichbar ist, aber nicht ständig gestört werden möchte.

Sein stellvertretender Projektleiter hat einen Anruf erhalten und sollte die Informationen an eine andere, am Projekt beteiligte Person, weiter leiten. Leider weiss er von dieser Person weder E-Mailadresse noch Telefonnummer und die Kontaktdaten auf Danilos PC sind passwortgeschützt. Deshalb ruft er schnell bei seinem Chef an. Danilo setzt den Anruf auf «Halten» und ruft den Task mit den entsprechenden Kontaktinformationen auf. Diese kann er seinem Projektleiter weitergeben, obwohl er gerade an einer Stadtführung in Bern teilnimmt.

4 Paperprototype

Für die effektive Umsetzung des Paperprototyps und die realisierten Funktionen können die Dokumente «Design Dokument (DD)» und «User Guide (UG)» zurate gezogen werden.

4.1 Allgemeine Darstellungsentscheide

Folgende Punkte gelten für alle Darstellungen sowohl auf dem PC als auch auf dem Mobile-Device.

- Tasks müssen nicht zwingend eine Priorität besitzen.
- Sortierreihenfolge
 - Die Reihenfolge der Tasks wird wie folgt festgelegt: Datum, Priorität, Zeit oder benutzerdefinierte Reihenfolge.
 - Tasks ohne Priorität werden innerhalb ihres Datums zu oberst einsortiert.
 - Tasks mit gleichem Datum aber ohne Zeitangabe werden nach denjenigen mit Zeitangabe einsortiert.
- Wird bei einem Task kein Fälligkeitsdatum angegeben, wird das aktuelle Datum verwendet.
- Shortcuts werden durch das Icon gekennzeichnet.
- Shortcuts von Tasks können nur über die Detailansicht aufgerufen werden.
- Es kann angegeben werden, wie viele erledigte Tasks und Ordner angezeigt werden.
- Favoriten
 - Die Reihenfolge ist editierbar.
 - Die Standardsortierung entspricht dem Hinzufügedatum.

4.2 PC-Applikation

4.2.1 Explorer-View

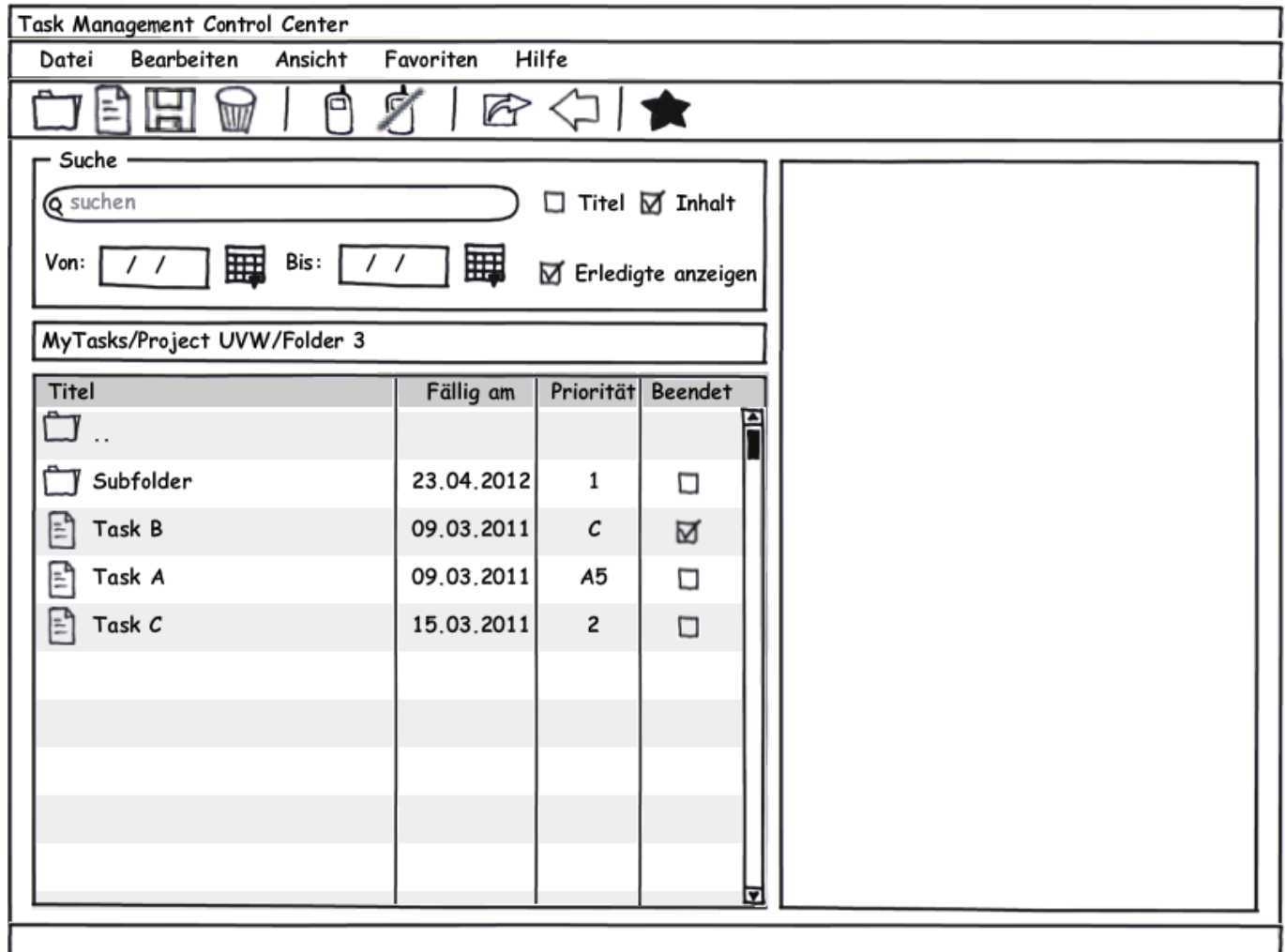


Abbildung 4.1: Explorer-View

- Selektieren einer Zeile öffnet den Bearbeitungsmodus rechts.
- Doppelklick auf einen Task öffnet eine Textanzeige in eigenem Fenster, welches sich immer vor dem Hauptfenster befindet, dieses aber nicht sperrt.
- .. wechselt in übergeordneten Ordner.
- Ordner werden immer am Anfang dargestellt.
- Ein Ordner wird erst als beendet markiert, wenn alle enthaltenen Tasks abgeschlossen sind.

4.2.2 Zeitstrahl-View

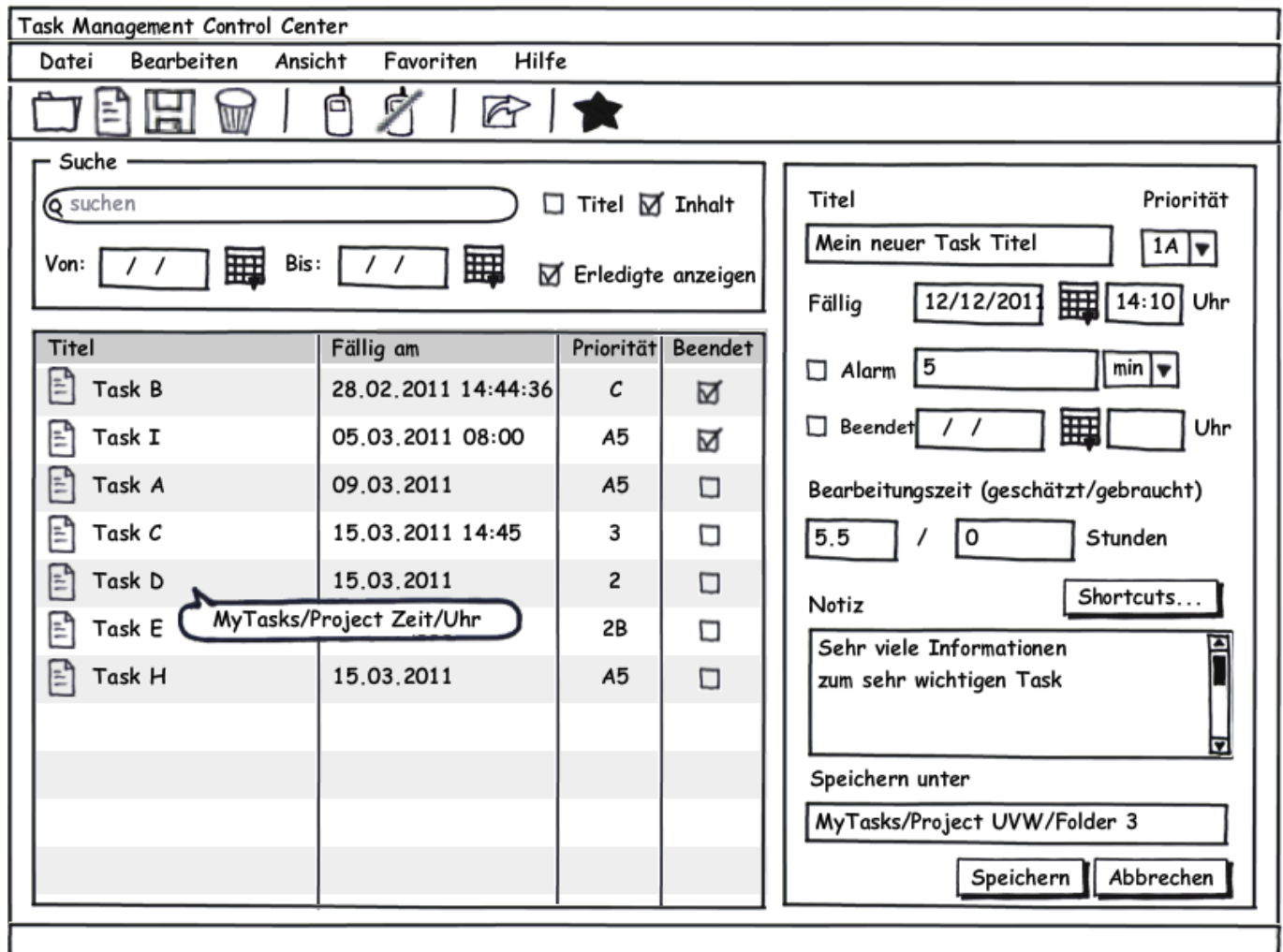


Abbildung 4.2: Zeitstrahl-View

- Beendete Tasks werden «ausgegraut».
- Timestamps mit Sekunden wurden vom System generiert, ohne sind es Benutzereingaben.
- Tasks, die ein Fälligkeitsdatum mit Zeit haben, werden immer am Anfang ihrer Priorität einsortiert.
- Tasks mit gleicher Priorität können innerhalb ihrer Prioritätsklasse neu sortiert werden.
- Die Anzahl dargestellter, erledigter Tasks ist konfigurierbar.

4.2.3 Task ansehen

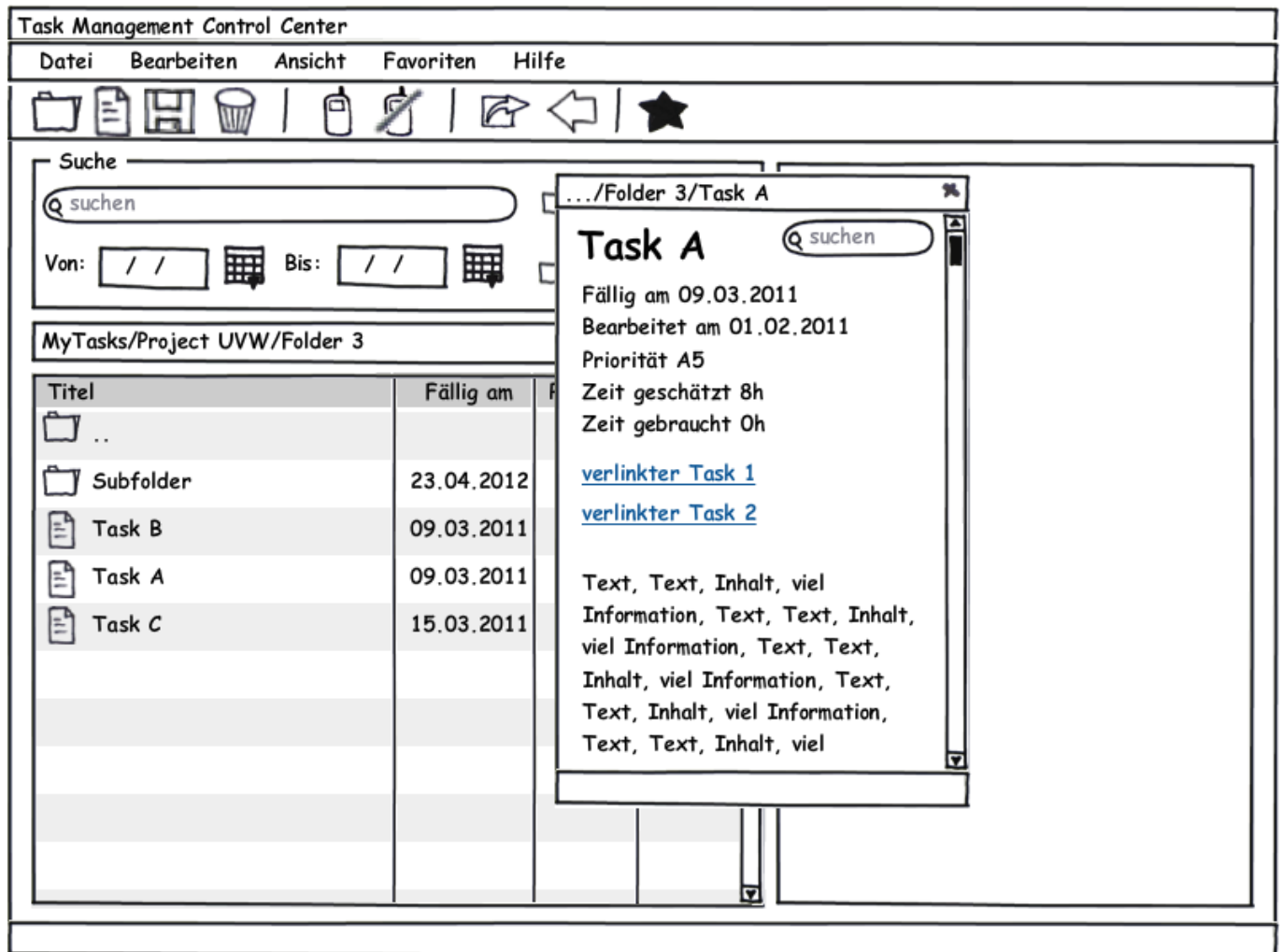


Abbildung 4.3: Task ansehen

4.2.4 Task erstellen oder ändern

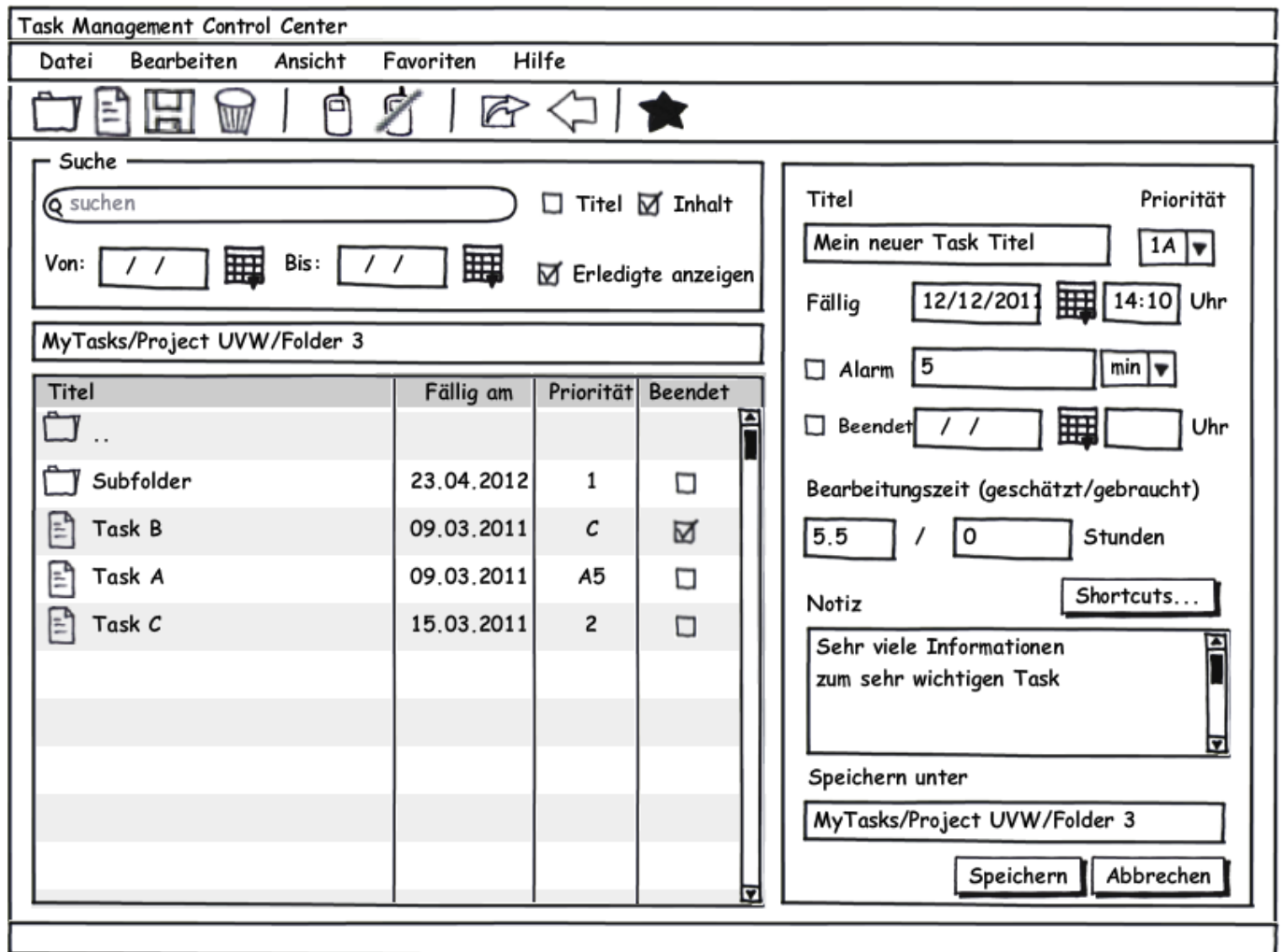


Abbildung 4.4: Neuer Task erstellen, Task ändern

- Die Prioritätsklassen werden gespeichert und sind zur Laufzeit konfigurierbar.
- «Speichern unter» ist standardmässig mit dem angezeigten Pfad gefüllt.
- Wird ein Ordner selektiert, werden die Informationen ebenfalls rechts angezeigt, nicht bearbeitbare Felder wie z.B. Bearbeitungszeit werden trotzdem angezeigt.

4.2.5 Shortcuts von Task oder Ordner editieren

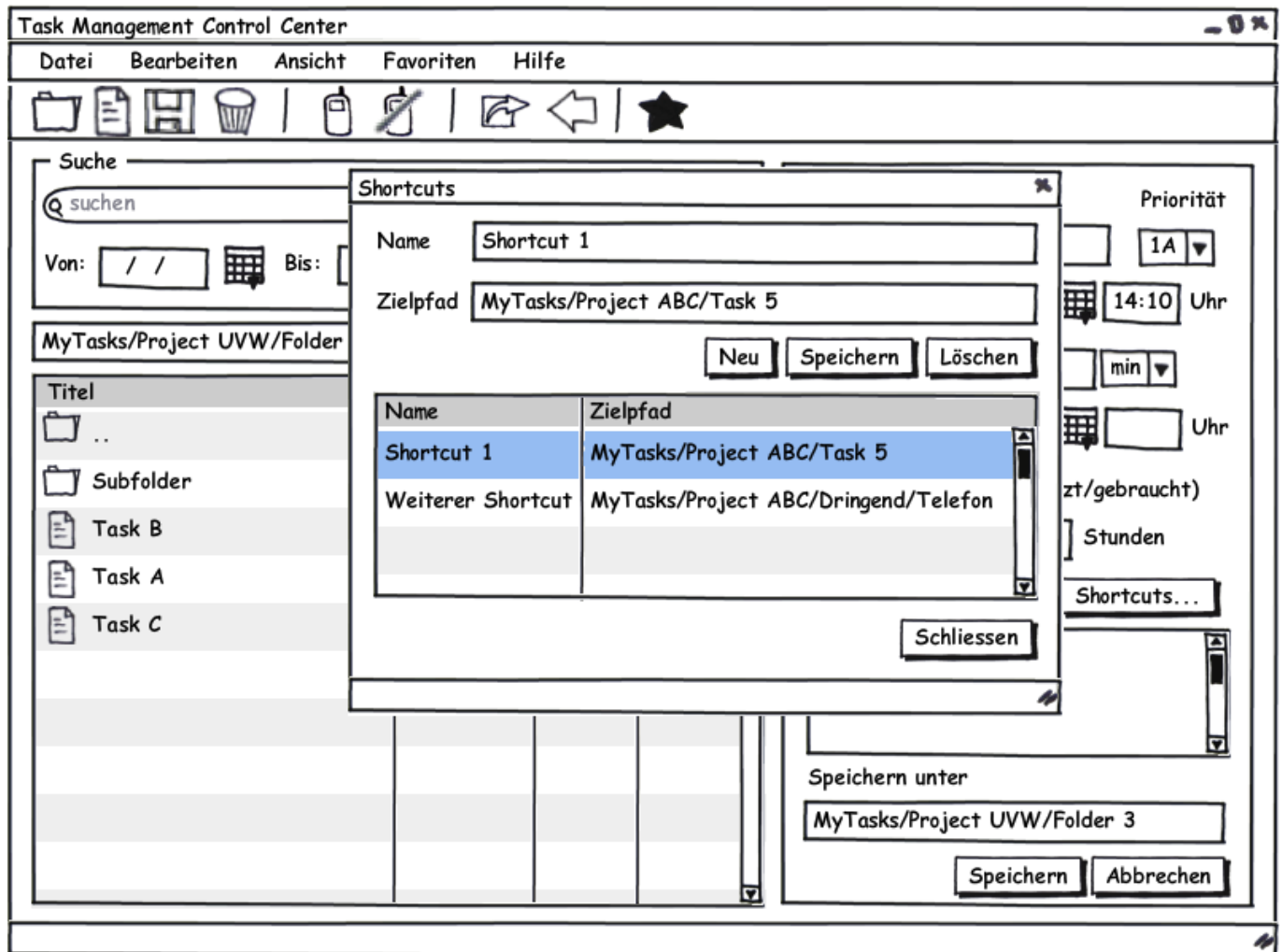


Abbildung 4.5: Shortcuts von Task oder Ordner editieren

4.2.6 Suchresultate

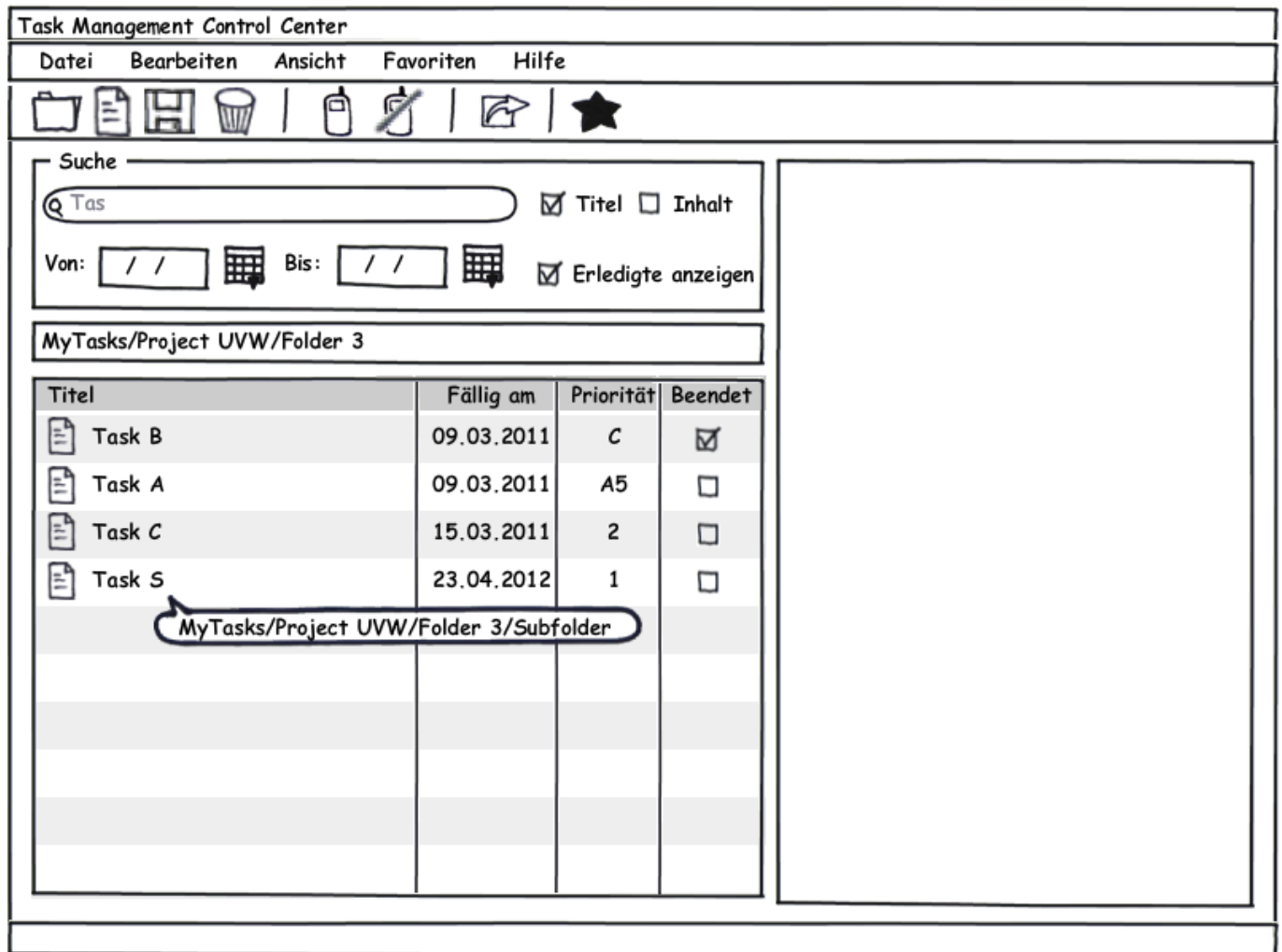


Abbildung 4.6: Suchresultate

- Der Pfad wird mit Hilfe eines Tooltips dargestellt.
- Selektieren bzw. Doppelklick funktioniert analog der normalen Darstellung.
- Um Suchresultate zu löschen, muss die Suche aktiv beendet werden (analog Windows Explorer).

4.3 Mobile-Device

4.3.1 Startbildschirm



Abbildung 4.7: Startbildschirm

Menü-Einträge:

- Einstellungen
- Verbindung
- Erweiterte Suche

4.3.2 Taskübersicht Explorer-View

Die Taskübersichten respektive die Anzeige der Favoriten werden in einer Tabview dargestellt. So können jederzeit die Vorteile der einzelnen Übersichten genutzt werden. Wird ein Ordner angeklickt, wird dessen Inhalt angezeigt. Wird ein Task angeklickt, so werden dessen Detailinformationen in einer weiteren View angezeigt.

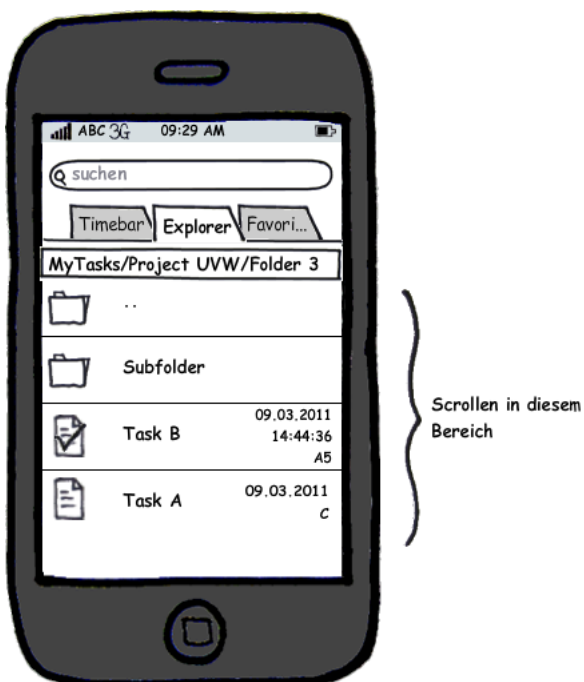


Abbildung 4.8: Taskübersicht

Menü-Einträge:

- Neuer Task
- Neuer Ordner
- Timestamp
- Audio Aufnahme
- zu Favoriten hinzufügen
- Ordneigenschaften
- Erweiterte Suche
- Löschen

4.3.3 Taskansicht Zeitstrahl-View

Die Zeistrah-View wird gleich dargestellt, wie die Suchresultate, für Details siehe Kapitel 4.3.8, Seite 19. Die Sortierung erfolgt wie bei der PC-Applikation. Als Bedienungserleichterung werden am oberen Bildschirmrand zusätzlich noch zwei Datumfelder zur Einschränkung der angezeigten Tasks hinzugefügt.

4.3.4 Taskansicht

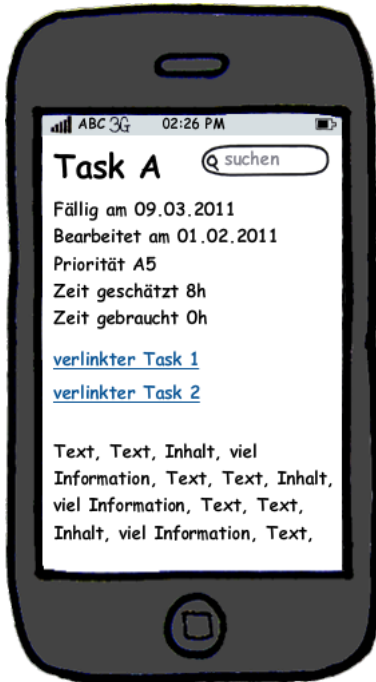


Abbildung 4.9: Taskansicht

Menü-Einträge:

- Bearbeiten
- Löschen
- zu Favoriten hinzufügen
- Ok

4.3.5 Task erstellen oder ändern

Titel

Priorität

Fällig am

Alarm

Beendet am

Bearbeitungszeit
(geschätzt/gebraucht)
 / Stunden

Shortcuts
Name
Ziel

Speichern unter

Notiz

- Ohne Mobile dargestellt, da scrollen erforderlich.
- Der Pfad in «Speichern unter» wird von der Anzeige übernommen, kann aber verändert werden.

Abbildung 4.10: Task erstellen oder ändern

4.3.6 Timestamp erzeugen



Angezeigt wird der Standardtitel, dieser ist jedoch änderbar

Abbildung 4.11: Timestamp erzeugen

4.3.7 Ordner erstellen oder ändern



Anzeigebildschirm, um einen Ordner zu bearbeiten.

- Ohne Mobile dargestellt, da scrollen erforderlich.
- Der Pfad in «Speichern unter» wird von der Anzeige übernommen, kann aber verändert werden.
- Bearbeitungszeit wird automatisch berechnet.

Abbildung 4.12: Ordner erstellen oder ändern

4.3.8 Standard Suche



Menü-Einträge:

- Neuer Task
- Neuer Ordner
- zu Favoriten hinzufügen

Abbildung 4.13: Standard Suche

4.3.9 Erweiterte Suche



Um die Suchresultate zu sehen, kann das Suchfeld an den oberen Bildschirmrand verschoben werden.

Abbildung 4.14: Erweiterte Suche

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	25.02.2011	DM	Dokument erstellt.
1.0rc02	28.02.2011	DM	Personas Paulo und Renata erstellt.
1.0rc03	01.03.2011	RR	Personas Isidoro und Danilo erstellt.
1.0rc04	09.03.2011	DM	Szenarios zu Paulo und Renata erstellt.
1.0rc05	09.03.2011	RR	Szenarios zu Isidoro und Danilo erstellt.
1.0rc06	11.03.2011	RR	Paperprototype-Bilder eingefügt.
1.0rc07	17.03.2011	DM, RR	Paperprototype gemäss Sitzung vom 17.03.2011 überarbeitet.
1.0rc08	24.03.2011	DM, RR	Änderungen gemäss Sitzung vom 24.03.2011 eingearbeitet.
1.0	24.03.2011	DM	Akzeptierte Version 1.0.
1.1rc01	25.04.2011	DM	Paperprototype-Ansichten für Mobile angepasst.
1.1	05.05.2011	DM	Akzeptierte Version 1.1.
1.2rc01	10.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.2	10.06.2011	RR	Akzeptierte Version 1.2

Tabelle 4.1: Revisionshistorie

Abkürzungsverzeichnis

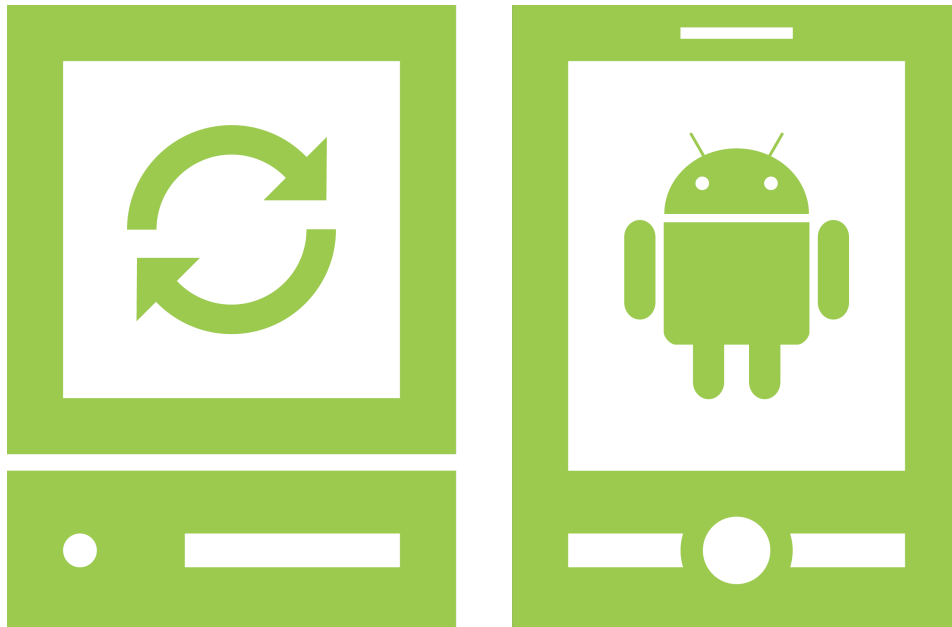
Ba Bachelorarbeit
BMS Berufsmittelschule
DD Design Dokument
PC Personal Computer
TCA Task-Management Control-Center for Android
UG User Guide
VPN Virtual Private Network

Abbildungsverzeichnis

4.1	Explorer-View	9
4.2	Zeitstrahl-View	10
4.3	Task ansehen	11
4.4	Neuer Task erstellen, Task ändern	12
4.5	Shortcuts von Task oder Ordner editieren	13
4.6	Suchresultate	14
4.7	Startbildschirm	15
4.8	Taskübersicht	15
4.9	Taskansicht	16
4.10	Task erstellen oder ändern	17
4.11	Timestamp erzeugen	18
4.12	Ordner erstellen oder ändern	18
4.13	Standard Suche	19
4.14	Erweiterte Suche	19

Tabellenverzeichnis

4.1 Revisionshistorie	20
---------------------------------	----



Software Architektur Dokument

«Task-Management Control-Center for Android»

Version 1.2, 11. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Logical View	2
2.1 PC-Applikation	2
2.1.1 gui	2
2.1.2 application	3
2.1.3 domain	3
2.1.4 networkinterface	3
2.2 Mobile Applikation	4
2.2.1 gui	4
2.2.2 domain	4
3 Deployment View	5
4 Process View	6
5 Data View	7

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 7.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Logical View

2.1 PC-Applikation

«TaMaF» und «AnCoF» sind keine Packages im Sinne der Logical View. Sie dienen nur der Übersichtlichkeit und stellen den Zugriff auf die benutzten Frameworks dar.

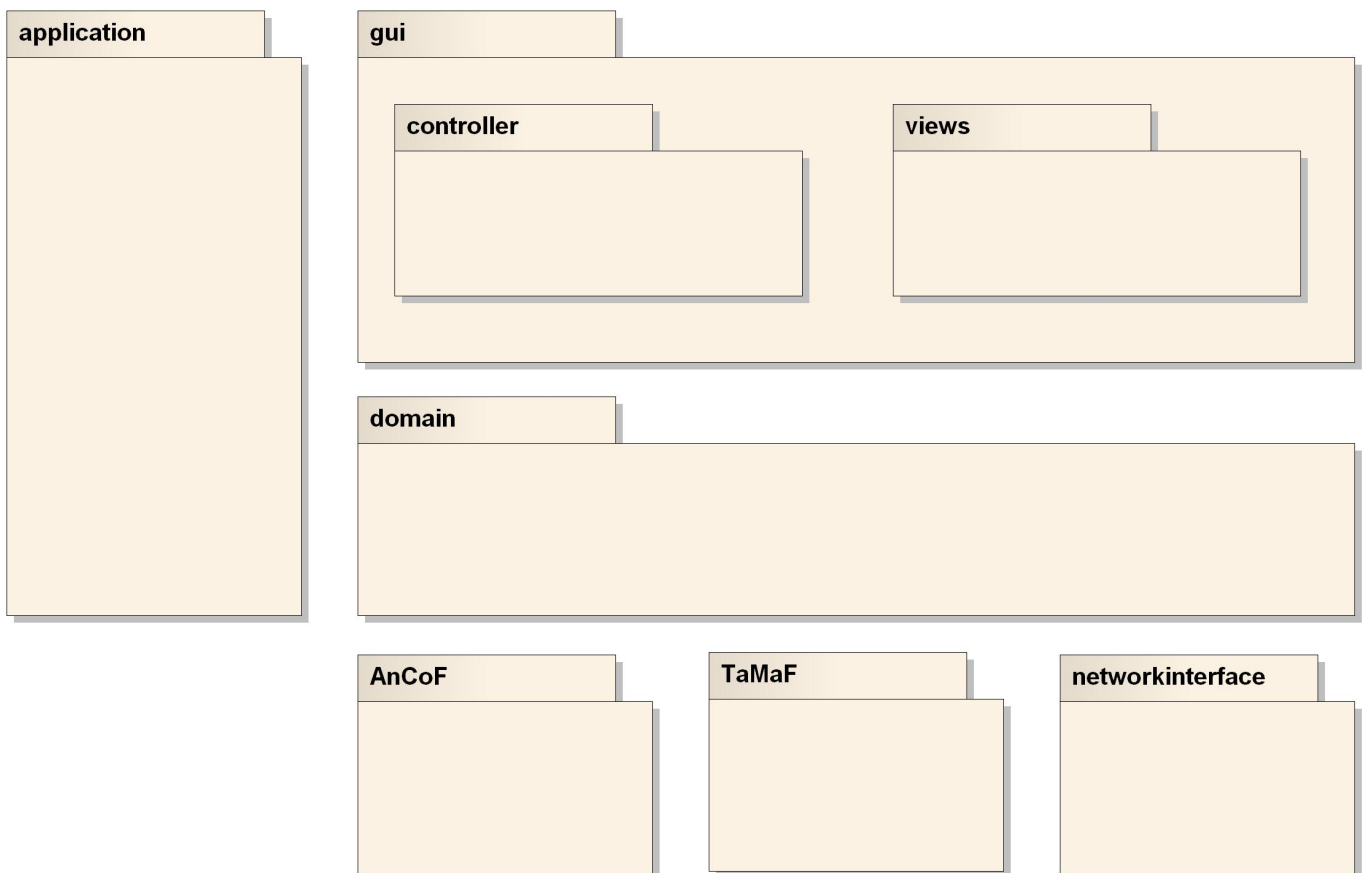


Abbildung 2.1: Packages PC-Applikation

2.1.1 gui

Dieses Package fasst alle Klassen zusammen, die in die Darstellung der Applikation involviert sind.

view

Alle Klassen, welche die Darstellung definieren, werden in diesem Package vereint. Diese sind so weit wie möglich generiert und sollten ohne GUI-Builder nicht verändert werden. Zudem enthalten die Klassen keine eigene Logik.

controller

Die Klassen des Packages «controller» steuern alle Aktionen der «view»-Klassen an. Reine GUI-Aktionen werden direkt an die zugehörige Viewklasse weitergegeben. Aktionen, welche zu persistierende Daten beinhalten, werden nach dem Model-View-Controller (MVC)-Pattern verarbeitet.

2.1.2 application

Das Package «application» steuert den Aufbau der Applikation sowie das Zusammenspiel der einzelnen Komponenten.

2.1.3 domain

In diesem Package wird die Applikations-Logik zusammengefasst. Zudem dient es als Verbindung zu dem Mobile-Device und dem Netzwerkinterface.

2.1.4 networkinterface

Alle Klassen, welche die Verbindung zum Firmen-Netzwerk regeln, werden im Package «networkinterface» zusammengefasst.

2.2 Mobile Applikation

Wie bei der PC-Applikation werden die Packages «TaMaF» und «AnCoF» ebenfalls nur zur Übersicht dargestellt.

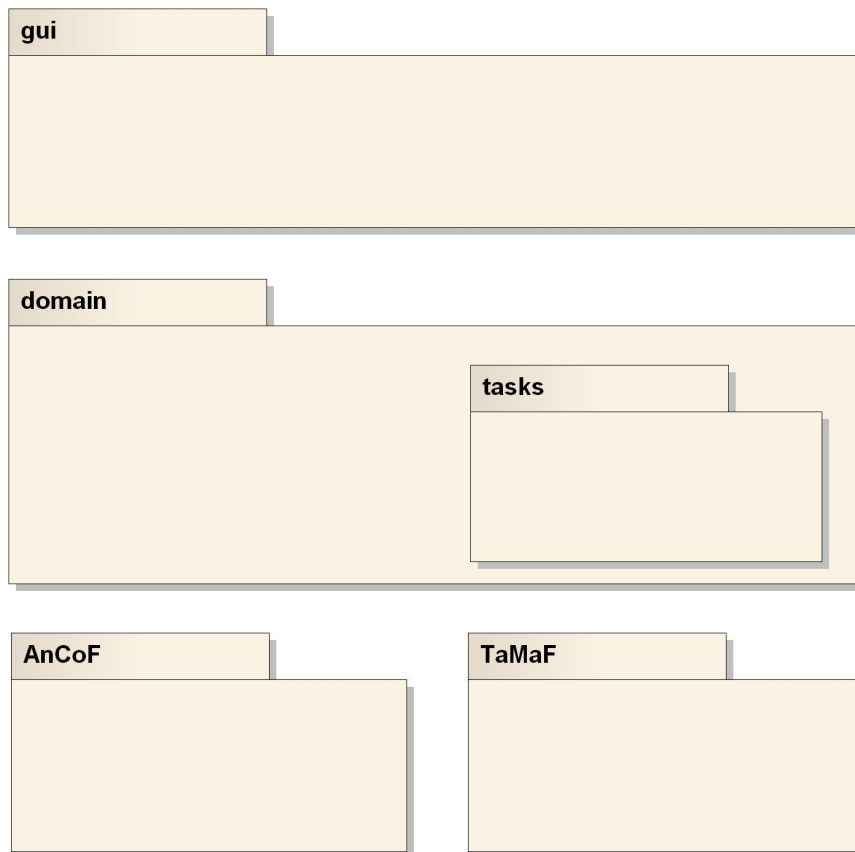


Abbildung 2.2: Packages Mobile Applikation

2.2.1 gui

Dieses Package enthält alle Klassen, welche die Darstellung steuern.

2.2.2 domain

Das Package «domain» fasst alle Klassen zusammen, welche die vom PC erhaltenen Daten empfangen und für die weitere Verwendung aufbereiten.

tasks

Dieses Package wird sowohl für das PC-Projekt als auch für das Mobile-Projekt verwendet. Es enthält die Task-Hierarchie wie sie im Dokument «Domainanalyse», Kapitel 2, Seite 2 beschrieben wird. Im Unterschied zu den Task-Klassen, welche Task-Management-Framework on Smart-Phone (TaMaF) besitzt, sind die Beziehungen zu anderen Tasks mit Referenzen anstelle der «uid» gelöst.

3 Deployment View

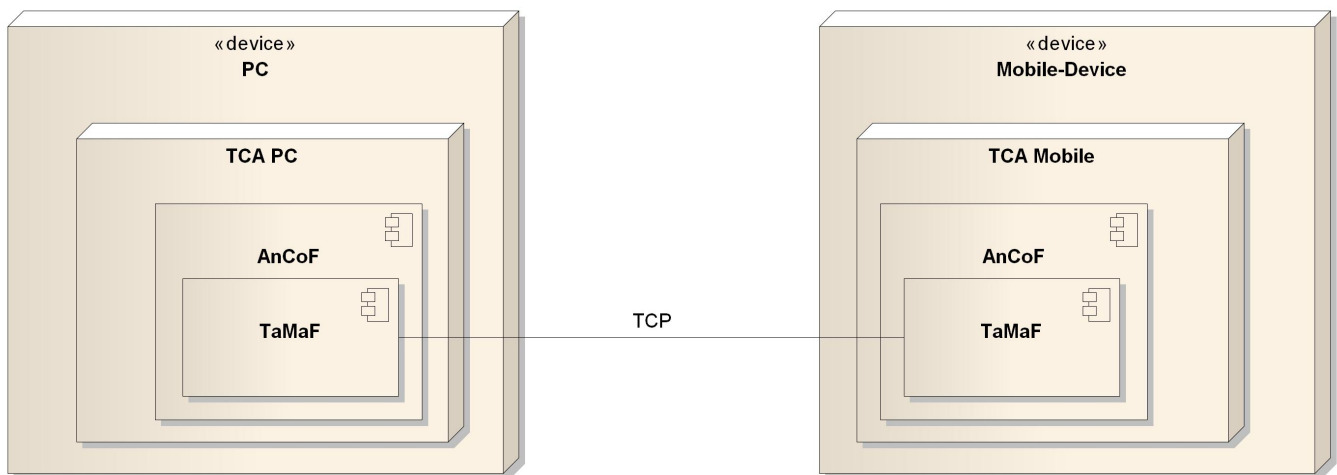


Abbildung 3.1: Deployment

Das TCA umfasst zwei Applikationen, eine PC- als auch eine Mobile-Applikation. Diese bauen auf dem Android Control Framework (AnCoF) auf, welches die Verbindung und den Datenaustausch zwischen den beiden Applikationen regelt.

4 Process View

Die Behandlung der durch den Datenaustausch entstehenden Parallelität wird von AnCoF bzw. TaMaF übernommen.

Die Verbindung zum Firmennetzwerk für eingehende Tasks muss in einem eigenen Thread ablaufen, damit die Schnittstelle immer sofortigen Zugriff auf das TCA hat.

Die Erfassung von Tasks und die Synchronisation von Dateien müssen unabhängig voneinander sein. Neu hinzukommende Tasks werden bei der Synchronisation von mehreren Dateien «dazwischen geschoben». Deshalb muss die Dateisynchronisation in einem eigenen Thread stattfinden.

5 Data View

Die Datenhaltung wird von AnCoF beziehungsweise TaMaF übernommen, Details können den jeweiligen Dokumentationen entnommen werden. Die Datenbank von TaMaF muss ersetzt werden, um die neuen Task-Eigenschaften unterstützen zu können. Die Beschreibung der Datenbank geschieht im «Design Dokument», Kapitel 4, Seite 5.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	21.03.2011	RR	Kapitel 1-5 erstellt.
1.0rc03	24.03.2011	DM, RR	Änderungen gemäss Sitzung vom 24.03.2011 eingearbeitet.
1.0	24.03.2011	DM	Akzeptierte Version 1.0.
1.1rc01	02.05.2011	RR	Dateview angepasst und Package tasks hinzugefügt.
1.1	05.05.2011	DM	Akzeptierte Version 1.1.
1.2rc01	10.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.2	10.06.2011	RR	Akzeptierte Version 1.2.

Tabelle 5.1: Revisionshistorie

Abkürzungsverzeichnis

AnCoF Android Control Framework

Ba Bachelorarbeit

GUI Graphical User Interface

MVC Model-View-Controller

PC Personal Computer

TCA Task-Management Control-Center for Android

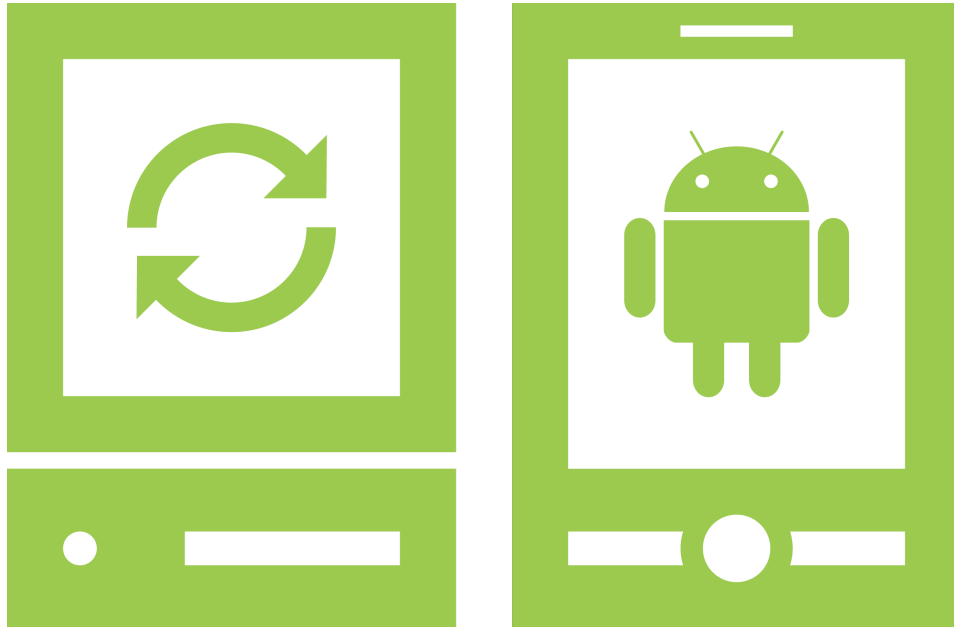
TaMaF Task-Management-Framework on Smart-Phone

Abbildungsverzeichnis

2.1	Packages PC-Applikation	2
2.2	Packages Mobile Applikation	4
3.1	Deployment	5

Tabellenverzeichnis

5.1 Revisionshistorie	8
---------------------------------	---



Designdokument

«Task-Management Control-Center for Android»

Version 1.0, 13. Juni 2011

Inhaltsverzeichnis

1	Einführung	1
1.1	Zweck des Dokuments	1
1.2	Gültigkeitsbereich	1
1.3	Anmerkung zu verwendeten Begriffen	1
2	TCA	2
3	AnCoF	3
3.1	Nutzung von Android Control Framework (AnCoF) mit onBind()	3
3.2	ch.hsr.sync.tamaf	4
4	Datenbank	5
5	TCA PC	7
5.1	Die Konfigurationsdateien	7
5.1.1	Konstanten	7
5.1.2	Imagename	7
5.2	Klassenstruktur	8
5.3	ch.hsr.tca.application	9
5.4	ch.hsr.tca.domain	10
5.5	ch.hsr.tca.gui	15
5.5.1	ch.hsr.tca.gui.views	15
5.5.2	ch.hsr.tca.gui.controller	25
5.5.3	ch.hsr.tca.gui.validation	30
5.5.4	ch.hsr.tca.gui.util	33
5.6	ch.hsr.tca.util	35
5.7	Kommunikation mit Events	35
5.7.1	Events der Tool- und Menubar	36
5.8	Mehrsprachigkeit der Applikation	38
5.9	Logging	38
6	TCA Android	39
6.1	Bugs in Android	39
6.2	Klassenstruktur	39
6.3	ch.hsr.tca.domain	39
6.3.1	ch.hsr.tca.domain.helpers	39
6.3.2	ch.hsr.tca.domain.providers	40
6.3.3	ch.hsr.tca.domain.tasks	40
6.4	ch.hsr.tca.gui	43
6.5	ch.hsr.tca.util	49
6.6	res: Oberflächengestaltung	51
6.6.1	Bilder	51
6.6.2	Layouts	51
6.6.3	Menues	52
6.6.4	Sprachen	53

6.7	Android Manifest	53
6.8	Android Testprojekt	53
7	Taskaustausch	54
8	Erweiterungen und Einschränkungen	56
8.1	TCA	56
8.2	AnCoF	56
8.3	Datenbank	56
8.4	TCA PC	56
8.4.1	Noch nicht implementierte Bereiche	56
8.4.2	Pfad-Eingabefeld	56
8.4.3	Alarm auslösen	56
8.4.4	Erkennen von verändertem Beendet-Datum	57
8.5	TCA Android	57
8.5.1	ch.hsr.tca.domain	57
8.5.2	ch.hsr.tca.gui	57
8.5.3	ch.hsr.tca.util	58
8.6	Sprachen	58
8.7	Taskaustausch	58
8.8	Verwendete Libraries und Bilder	58
8.8.1	Libraries	58
8.8.2	Bilder	59
9	Anhang	60

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 8.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

1.3 Anmerkung zu verwendeten Begriffen

Der Begriff «Task» wird, wenn nicht anders vermerkt, als Oberbegriff für alle Arten von Tasks verwendet. Wird auf den Typ eines Tasks verwiesen, werden immer die Grundnamen benutzt, also z.B. BasicTask und nicht DomainBasicTask.

2 TCA

Das TCA besteht zur Hauptsache aus zwei einzelnen Projekten, je eines für die PC-Seite und für die Mobile-Device-Seite. Diese Aufteilung ist erforderlich, da die Laufzeitumgebung von Android auf der «Dalvik virtual machine» basiert. Diese ist optimiert für Mobile-Devices. Nebst diesen beiden Projekten existiert das Projekt AnCoF, welches als Library in die beiden oben erwähnten Projekte eingebunden wurde. AnCoF stellt die Schnittstellen für die Datei- und Tasksynchronisation und diejenigen für den Verbindungsaufbau bereit. Genauere Informationen zu AnCoF können dem Kapitel 3, Seite 3 entnommen werden. Um das Android-Projekt zu testen, musste ein separates Testprojekt erstellt werden, «TCAAndroidTest». Details zum Testprojekt können dem Kapitel 6.8, Seite 53 entnommen werden.

Der Code der beiden "Hauptprojekte" ist gleich strukturiert und unterscheidet sich nur in wenigen Bereichen, sprich Graphical User Interface (GUI) und Datenbankzugriff. Wo möglich wurden die gleichen Klassen verwendet. Um keinen doppelten Code zu schreiben, wurden die betroffenen Packages im Android-Projekt erstellt und im PC-Projekt als «class folder» dem «build path» hinzugefügt. Diese Klassen mussten im Android-Projekt angesiedelt werden, da die «Dalvik virtual machine» Klassen aus einem Java-Projekt nicht fehlerfrei kompilieren kann.

Weitere Informationen zu den beiden Projekten können den Kapiteln 5, Seite 7 und 6, Seite 39 entnommen werden.

3 AnCoF

AnCoF wurde als bestehendes Projekt übernommen und entsprechend den Anforderungen des TCA erweitert und angepasst. Folgende Packages wurden in TCA verwendet:

- ch.hsr.ancof
- ch.hsr.ancof.data
- ch.hsr.sync.tamaf

Aus dem Package «ch.hsr.ancof.data» wurde die entsprechende Schnittstelle implementiert, im Package «ch.hsr.sync.tamaf» erfolgten nebst Implementierung der benötigten Schnittstellen zusätzlich strukturelle Änderungen am Code der einzelnen Klassen.

3.1 Nutzung von AnCoF mit onBind()

Der Service «AnCoFService» wurde dahingehend erweitert, dass dieser nun sowohl mit «start» als auch mit «bind» in eine Applikation eingebunden werden kann.

Dies wurde nötig, da zur Laufzeit erkannt werden muss, ob der Service bereits gestartet wurde.

3.2 ch.hsr.sync.tamaf

Die Taskstruktur und die vorgesehene Datenbank von Task-Management-Framework on Smart-Phone (TaMaF) wurden komplett ausgewechselt. Eine Beschreibung der neuen Datenbank kann im Kapitel 4 auf Seite 5 gefunden werden.

Alle Taskklassen überschreiben die Methode «equals», um einen einfachen Mechanismus zur Überprüfung der Tasks bereitzustellen. Die neue Taskstruktur und ihre Eigenschaften wird in der folgenden Abbildung gezeigt. Zu beachten ist, dass die Referenzen auf andere Tasks (z.B. für die Children-Liste oder das Parent) mit einer long-Variable gelöst wurden.

Der PriorityTask ist eine abstrakte Klasse, da er keine direkte Bedeutung für den Benutzer besitzt und zu «Code-Reuse»-Zwecken erstellt wurde.



Abbildung 3.1: Die neue Taskstruktur

4 Datenbank

Die Datenbank ist sowohl auf der PC- wie auch der Mobile-Seite gleich aufgebaut. Um auf die gleiche Datenbanktechnologie setzen zu können, wurde SQLite gewählt.

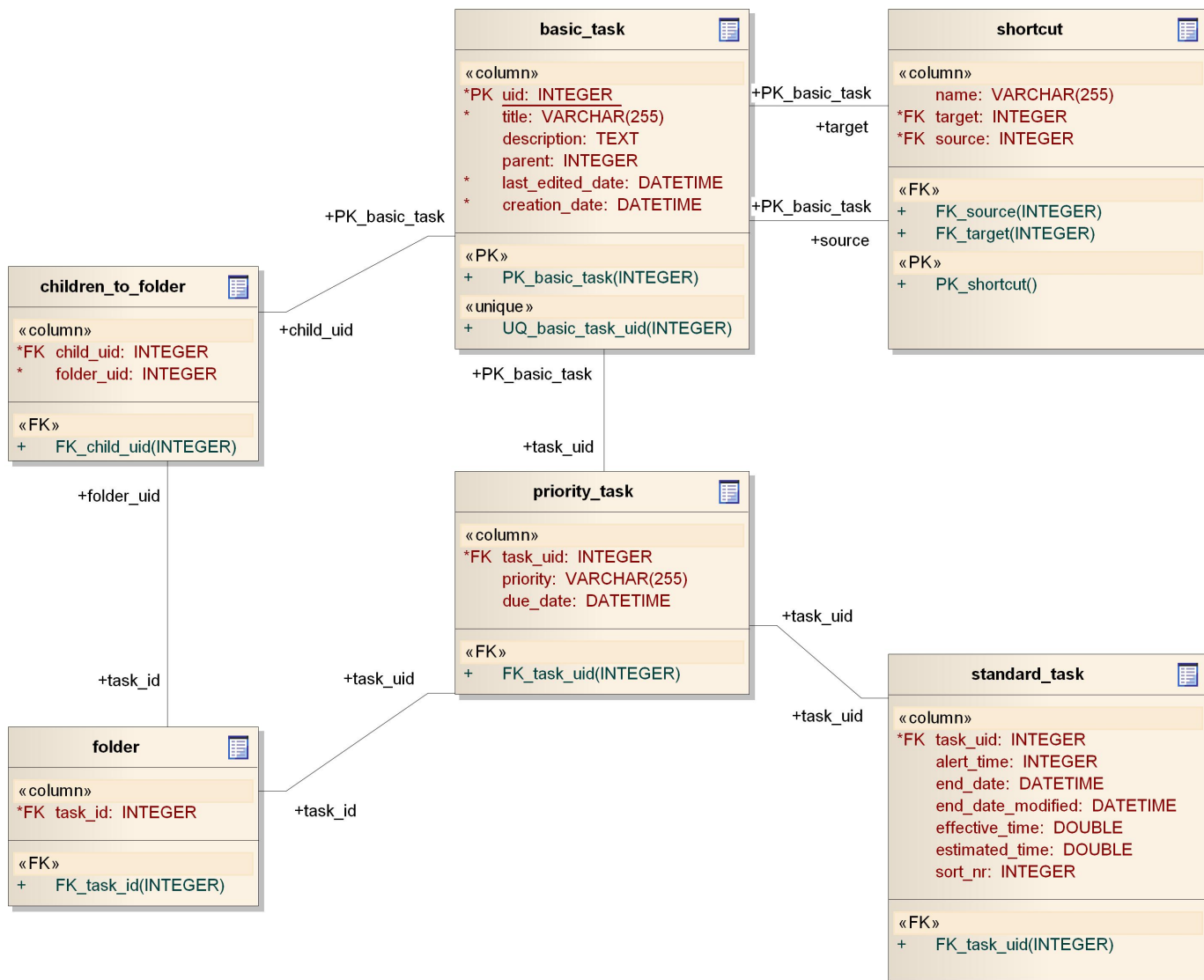


Abbildung 4.1: Datenbankschema

Folgende Tabellen werden erstellt:

- basic_task: Haupttabelle, welche die wichtigsten Felder eines Tasks enthält.
- shortcut: Wird für das Speichern eines Shortcuts benötigt.
- priority_task: Erweitert den BasicTask um eine Priorität und ein Fälligkeitsdatum.
- folder: Ermöglicht die Verschachtlung von Tasks.

- children_to_folder: Wird für die Zuordnung eines Subtasks zu einem folder benötigt.
- standard_task:Erweitert den PriorityTask um folgende Felder:
 - Alarmzeit
 - Enddatum, dass automatisch vom System generiert wird
 - Enddatum, dass manuell vom Benutzer gesetzt wird
 - effektive Bearbeitungszeit
 - geschätzte Bearbeitungszeit
 - Sortierungsnummer, die für die individuelle Sortierung der Tasks benötigt wird (wird momentan nicht benutzt)

5 TCA PC

Das PC-Projekt beinhaltet alle Packages, welche für die Logik und Darstellung der PC-Applikation benötigt werden.

5.1 Die Konfigurationsdateien

5.1.1 Konstanten

Die Datei «constants.properties» enthält alle wichtigen Konstanten der Applikation. Sie ist mittels Kommentaren in einzelne logische Abschnitte unterteilt. Es muss beachtet werden, dass gewisse Konstanten nicht einzeln geändert werden dürfen. Ist dies der Fall, wird in den entsprechenden Kapiteln darauf hingewiesen.

Die wichtigste Konstante ist «MainFolder». Diese zeigt an, wo die Benutzerdaten gespeichert werden. Alle weiteren Konstanten im Abschnitt «File Locations», welche mit dem PC Filesystem zu tun haben, werden relativ dazu angegeben und müssen einen vorangestellten Doppelbackslash aufweisen.

5.1.2 Imagenames

Die Bilder, welche für die PC-Applikation benötigt werden liegen im Ordner «pictures». Die korrekte Datenstruktur der Bilder wird im Kapitel 5.5.4 auf Seite 33 beschrieben.

Alle Bildnamen werden in der Datei «imagenames.properties» in logische Namen übersetzt. Dies wurde eingeführt, damit ein Bild umbenannt/ausgetauscht werden kann, ohne dass der Source Code verändert werden muss. Dies ist z.B. der Fall, wenn ein Bild an drei Orten verwendet wird, und nur an zwei Orten ausgetauscht werden soll.

Alle Bildnamen werden ohne Dateierweiterung gespeichert, da dies die Klasse «ImageLoader» übernimmt.

5.2 Klassenstruktur

Folgende Abbildung gibt einen Überblick über die einzelnen Packages und deren Klassen bzw. Subpackages.

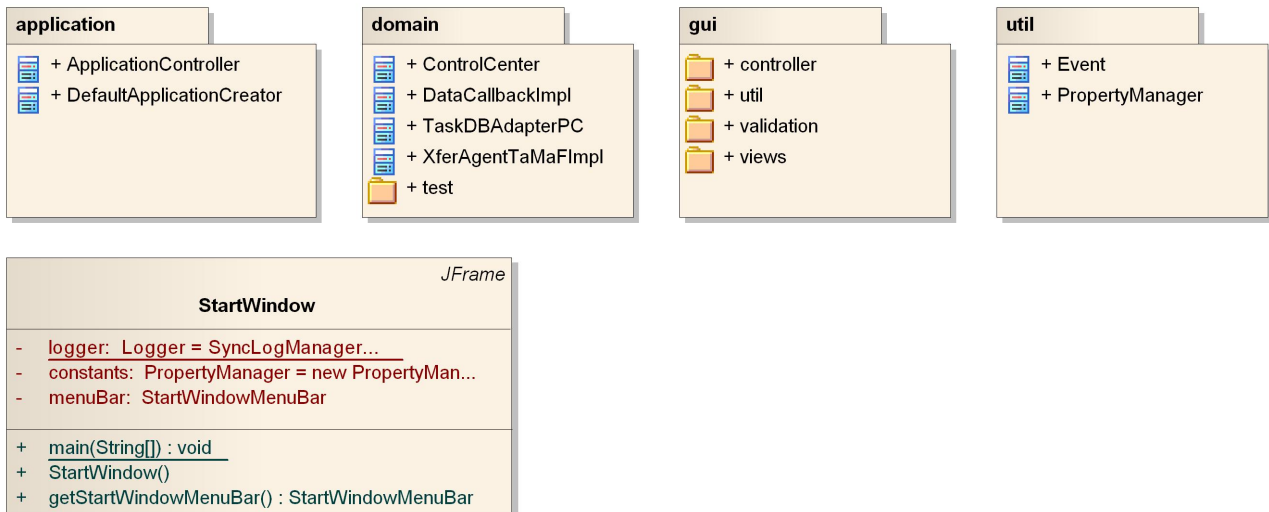


Abbildung 5.1: Klassenstruktur

Um Code-Duplikationen zu vermeiden, wurden wie bereits im Kapitel 2, Seite 2 erwähnt, einzelne Packages als «class folder» im «build path» eingebunden. Namentlich sind dies die folgenden zwei Packages, welche sich im Projekt «TCA_Android» befinden:

- ch.hsr.tca.domain.tasks
- ch.hsr.tca.util

Beim Package «ch.hsr.tca.domain.tasks» handelt es sich um Hilfsklassen, welche, im Unterschied zu den «Task-Klassen» im TaMaF, die Beziehungen zu anderen Tasks mit Referenzen anstelle der UIDs löst. In «ch.hsr.tca.util» sind Klassen, wie die «Comparatoren» oder eigene Exceptions enthalten. Weitere Informationen zum Inhalt dieser Packages können den Kapiteln 6.3.3, Seite 40 und 6.5, Seite 49 entnommen werden.

ch.hsr.tca.StartWindow

Die Klasse «StartWindow», welche sich ausserhalb der Subpackages befindet, ist eine Implementation von «JFrame» und dient der PC-Applikation als Hauptfenster. Sie enthält keine eigene Logik, startet sich jedoch selbst.

Im folgenden wird auf die einzelnen Packages eingegangen.

5.3 ch.hsr.tca.application

Das Package «application» dient als Zwischenstück zwischen GUI und Domainlogik.

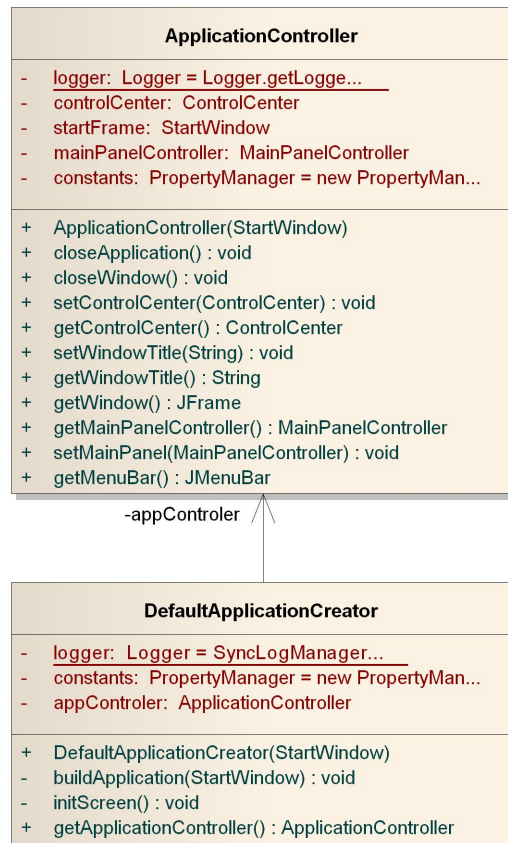


Abbildung 5.2: Package ch.hsr.tca.application

ch.hsr.tca.application.DefaultApplicationCreator

Die Klasse «DefaultApplicationCreator» dient als aufbauende Klasse. Sie erstellt den eigentlichen «ApplicationController» und macht das «MainPanel» mit ihm bekannt.

ch.hsr.tca.application.ApplicationController

Der ApplicationController wird an alle GUI-Controller weitergegeben. Er ist der oberste Controller und kennt alle «Subcontroller». Zudem initiiert er alle «Aufräumarbeiten», wenn die Applikation geschlossen werden soll.

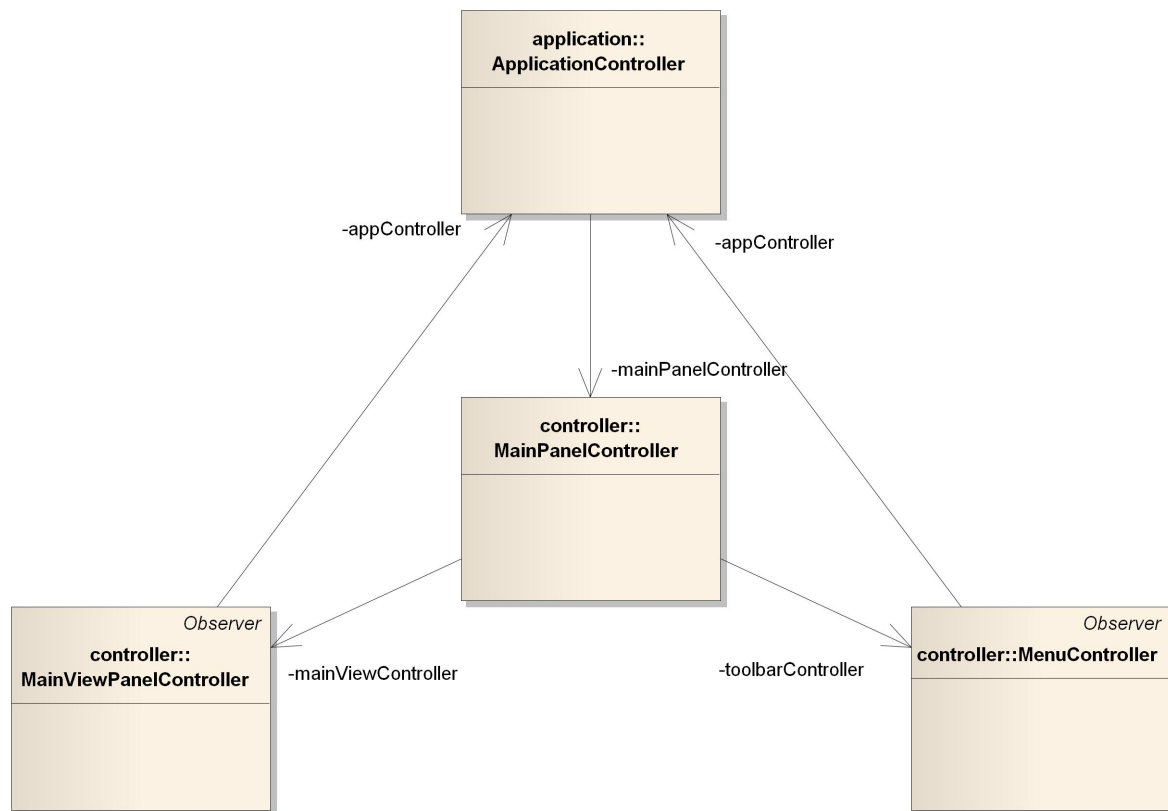


Abbildung 5.3: Struktur der Controller in TCA

Die anderen Controller, welche der Abbildung 5.3 entnommen werden können, werden im Detail im Kapitel 5.5.2 ab Seite 25 beschrieben.

5.4 ch.hsr.tca.domain

In diesem Package befindet sich die eigentliche Logik des Programms.

test

- + ControlCenterTest
- + DBAdapterTest
- + LoadTest
- + XferAgentTest

TaskDBAdapterPC

- logger: Logger = SyncLogManager...
- constants: PropertyManager = new PropertyMan...
- connection: Connection
- connectionInfo: String = "jdbc:sqlite:..."

- # getConnection() : Connection
- + getTask(long) : BasicTask
- + getChildTasks(long) : List<BasicTask>
- fillInFolderChild(ResultSet) : Folder
- fillInStandardTaskChild(ResultSet) : StandardTask
- + getBasicTask(long) : BasicTask
- # getFolder(long) : Folder
- # getStandardTask(long) : StandardTask
- # getAllStandardTask() : List<StandardTask>
- getChildren(long) : List<Long>
- # getShortcuts(long) : List<Shortcut>
- fillInBasicTaskVariables(ResultSet, BasicTask) : void
- fillInPriorityTaskVariables(ResultSet, PriorityTask) : void
- fillInStandardTaskVariables(ResultSet, StandardTask) : void
- getCalendarFromDbValue(ResultSet, String) : Calendar
- + updateTask(BasicTask) : void
- + updateTask(Folder) : void
- + updateTask(StandardTask) : void
- + updateShortcut(long, long, Shortcut) : void
- prepareUpdateBasicTask(BasicTask, Connection) : PreparedStatement
- prepareUpdatePriorityTask(PriorityTask, Connection) : PreparedStatement
- prepareUpdateStandardTask(StandardTask, Connection) : PreparedStatement
- updateAsChild(BasicTask, Connection) : void
- prepareUpdateChildOfFolder(long, long, Connection) : PreparedStatement
- prepareUpdateShortcut(Shortcut, long, long, Connection) : PreparedStatement
- + saveTask(BasicTask) : void
- + saveTask(Folder) : void
- + saveTask(StandardTask) : void
- + saveShortcut(long, Shortcut) : void
- saveShortcuts(BasicTask, Connection) : void
- saveAsChild(BasicTask, Connection) : void
- prepareSaveBasicTask(BasicTask, Connection) : PreparedStatement
- prepareSaveShortcut(long, Shortcut, Connection) : PreparedStatement
- prepareSavePriorityTask(PriorityTask, Connection) : PreparedStatement
- prepareSaveFolder(Folder, Connection) : PreparedStatement
- prepareSaveStandardTask(StandardTask, Connection) : PreparedStatement
- prepareSaveChildOfFolder(long, long, Connection) : PreparedStatement
- + deleteTask(BasicTask) : void
- + deleteTask(Folder) : void
- + deleteTask(StandardTask) : void
- + deleteShortcut(long, Shortcut) : void
- prepareDeleteBasicTask(BasicTask, Connection) : PreparedStatement
- prepareDeletePriorityTask(PriorityTask, Connection) : PreparedStatement
- prepareDeleteFolder(Folder) : PreparedStatement
- prepareDeleteStandardTask(StandardTask, Connection) : PreparedStatement
- prepareDeleteShortcuts(BasicTask, Connection) : PreparedStatement
- prepareDeleteShortcut(long, Shortcut, Connection) : PreparedStatement
- prepareDeleteChild(BasicTask, Connection) : PreparedStatement
- createStatementWithArguments(String, Connection, Object) : PreparedStatement

Abbildung 5.4: Package ch.hsr.tca.domain Teil 1

ch.hsr.tca.domain.TaskDBAdapterPC

Der TaskDBAdapterPC dient als Bindeglied zwischen der SQLite-Datenbank und der Domainschicht der PC-Applikation. Die Klasse besteht aus sechs Teilen, die sich in folgende Bereiche gliedern lassen:

1. einer Methode um die Datenbankverbindung zu erstellen
2. Auslesen der Tasks
3. Speichern eines neuen Tasks
4. Updaten eines Tasks
5. Löschen eines Tasks
6. einer Hilfsmethode, die für das Erstellen der PreparedStatements¹ verantwortlich ist.

Jeder der Bereiche 3-4 besteht aus vier public Methoden, um alle Taskarten und Shortcuts bearbeiten zu können. Hierzu werden alle Tabellen zuerst einzeln vorbereitet und danach zusammen an die Datenbank übermittelt.

Um die Rückgabe unvollständiger Tasks zu vermeiden, werden in den privaten Methoden keine Exceptions gefangen. Diese Aufgabe übernimmt die «Hauptmethode» und bricht somit den ganzen Vorgang ab.

¹<http://download.oracle.com/javase/1.4.2/docs/api/java/sql/PreparedStatement.html>



Abbildung 5.5: Package ch.hsr.tca.domain Teil 2

ch.hsr.tca.domain.ControlCenter

Das ControlCenter ist das Herzstück der Applikation.

Laden der Tasks

Nach dem Starten werden alle Taskinformationen aus der Datenbank geladen und in «Domain Tasks» (siehe Kapitel 6.3.3, Seite 40) gewandelt.

In einem zweiten Schritt werden alle Shortcutinformationen, welche zuvor aus der Datenbank geladen wurden, in Referenzen auf die entsprechenden Taskobjekte umgewandelt. Zudem wird die Map «tasksToShortcuts» gefüllt. Diese Map wird gebraucht, um die Shortcuts, welche auf den Task zeigen, zu erkennen, wenn der Task gelöscht wird.

create, read, update, delete (CRUD)-Operationen der Tasks

Das ControlCenter leitet die CRUD-Operationen, welche vom GUI kommen, an den «TaskDBAdapterPC» weiter. Zudem organisiert es das Umhängen von Referenzen, falls der Task an einen neuen Ort verschoben wurde. Weiter kann ein Task mit diesen Funktionen an das Mobile-Device weitergeleitet werden.

Erstellen eines Speicherpfades

Die Speicherpfad-Strings werden ebenfalls im ControlCenter berechnet, da dieses bereits eine Referenz auf alle Tasks hält. Dasselbe geschieht, wenn von einem Pfad auf den Task geschlossen werden muss. Etwas schwieriger gestaltet sich die Berechnung des Pfades, falls nur die UID vorliegt. In diesem Fall muss die Datenbank zu Hilfe genommen werden.

Diese Methoden werden sehr oft gebraucht, da die einzelnen Pfade nur virtuell als Referenzen auf andere Tasks bestehen. Um diese Referenzen dem Nutzer zugänglich zu machen, müssen sie zuerst in einen für Menschen verständlichen String umgewandelt werden.

Im Detail wird, um einen Pfad-String zu erstellen, dem existierenden Teilpfad so lange der Name des nächst höheren Parents vorangestellt, bis das Parent «null» ist und somit der «root» entspricht. Dieser Vorgang ist möglich, da ein Kindtask ohne ein Parent, zwingend in der «root» liegen muss.

Soll ein Task anhand eines Pfades gesucht werden, wird dieser Vorgang umgekehrt, indem zuerst aus dem ersten Teil des Pfades der passende «Roottask» gesucht wird. Von diesem wird das zum nächsten Teil des Pfades passende Kind gesucht. Dieser Vorgang wird wiederholt, bis entweder kein Kind gefunden wird oder alle Teilpfade abgearbeitet wurden. Im ersten Fall handelt es sich um einen inkorrekten Pfad. Im zweiten Fall ist der zuletzt gefundene Task korrekt und kann zurückgegeben werden.

Verbindung zum Mobile-Device und Austausch der Dateien

Der PC übernimmt die Rolle des Client in der Verbindung zum Mobile-Device, d.h. das Mobile-Device muss die Verbindung initiieren. Details zum Verbindungsaufbau können der Dokumentation von TaMaF im Dokument «Developer Guide», Kapitel 4.2, Seite 7 entnommen werden.

Nachdem die Verbindung zustande gekommen ist, werden zuerst die Prioritäten, welche sich momentan in der GUI-Combobox befinden, in die Prioritäten-Datei zurückgeschrieben. Diese Datei wird nun als erste Datei ausgetauscht. Dies ist wichtig, da auf dem Mobile-Device keine Prioritäten hinzugefügt werden können. Der weitere Austausch wird danach über die Klasse «DataCallbackImpl» und die Methode «syncNextPath» geregelt (siehe Kapitel 5.4, Seite 15).

Soll die Verbindung zum Mobile-Device getrennt werden, wird, wie beim Verbindungsaufbau, die Prioritäten-Datei erneut als erste Datei ausgetauscht.

ch.hsr.tca.domain.XferAgentTaMaFImpl

Diese Klasse regelt das Verhalten bei einem Taskaustausch. Der Austauschvorgang wird detailliert im Kapitel 7 auf Seite 54 beschrieben.

ch.hrs.tca.domain.DataCallbackImpl

Diese Klasse ist eine Implementation des Interfaces «DataCallback» von AnCoF. Details zum Synchronisationsalgorithmus können der Dokumentation von AnCoF entnommen werden.

Die Synchronisierung der Dateien findet asynchron statt. Sobald ein Ordner synchronisiert wurde, wird von AnCoF die Methode «onDoSyncFinished» aufgerufen. Diese beginnt mit Hilfe des «ControlCenters», den nächsten Ordner zu synchronisieren.

5.5 ch.hsr.tca.gui

Im Package «ch.hsr.tca.gui» liegen alle Klassen, welche für die Darstellung und Verarbeitung von GUI-Aktionen verantwortlich sind.

Die Kommunikation zwischen den darstellenden Klassen des Packages «views» und den steuernden Klassen des Packages «controller» geschieht mit Observern und der Klasse «ch.hsr.tca.util.Event» (siehe Kapitel 5.6, Seite 35).

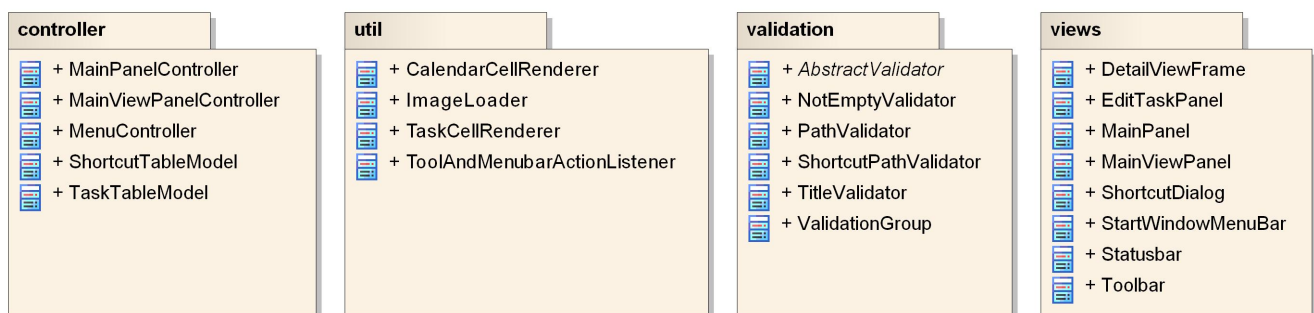


Abbildung 5.6: Package ch.hsr.tca.gui

In den folgenden Kapiteln sollte beachtet werden, dass die Beschreibung des GUIs immer mit deutschen Begriffen erfolgt. Die englische Übersetzung kann dem Resourcebundle im Ordner «languages» entnommen werden.

5.5.1 ch.hsr.tca.gui.views

Die Klassen dieses Packages enthalten selbst keine Logik, sie leiten lediglich erkannte Benutzeraktionen an ihre Controller weiter. Die «initComponents»-Methode der GUI-Klasse wurde zum einen Teil mit dem GUI-Builder von Netbeans erstellt und zum anderen Teil von Hand angepasst bzw. erweitert.

Die Methode «addListeners» definiert das Verhalten der einzelnen GUI-Elemente. Sie wurde eingeführt, um die reine Darstellung der Elemente, welche in der Methode «initComponents» definiert wird, von derjenigen des Verhaltens zu trennen.

In den folgenden Abbildungen wird die Klasse «EditPanel» zweimal gezeigt. Dies dient dem besseren Verständnis, die Klasse selbst ist nur einmal vorhanden.



Abbildung 5.7: Package ch.hsr.tca.gui.views Teil 1

ch.hsr.tca.gui.views.MainPanel

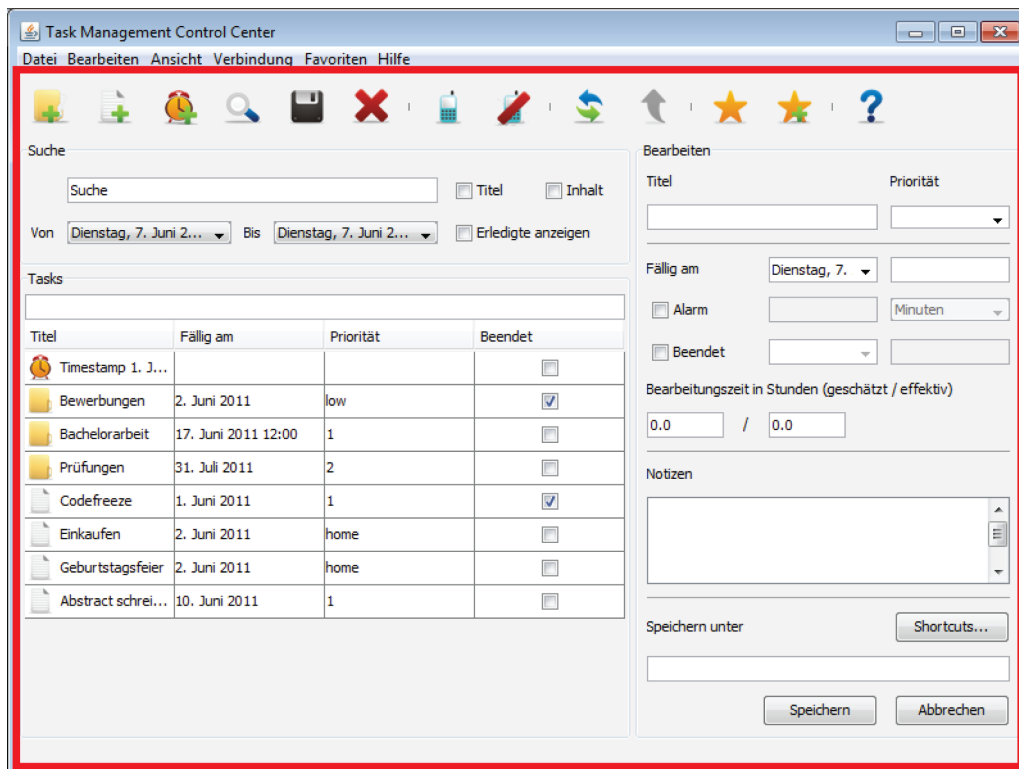


Abbildung 5.8: Die Klasse MainPanel im GUI

Das MainPanel ist die «unterste» Panel-Schicht des GUIs und enthält nebst der «MenuBar», alle anderen «JPanel». Namentlich sind dies die «ToolBar», das «MainViewPanel» und die «Statusbar».

Um die Komponenten anzuordnen, wurde ein «BoxLayout», angeordnet an der y-Achse, verwendet.

ch.hsr.tca.gui.views.Toolbar

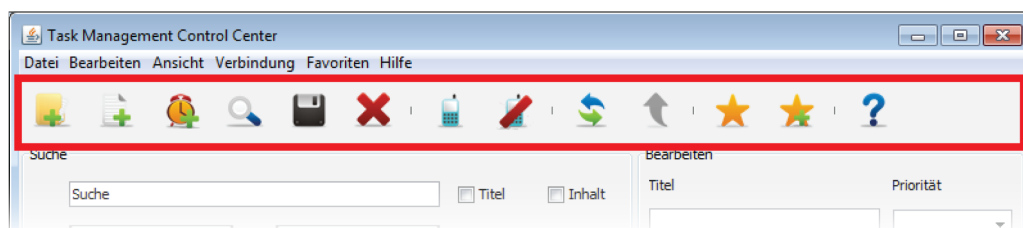


Abbildung 5.9: Die Klasse Toolbar im GUI

Die Toolbar enthält eine Auswahl der Buttons der Menubar. Deshalb haben Tool- und Menubar den gleichen Controller, den «MenuController».

Während dem Initialisieren werden die Buttons in der Map «buttons» gespeichert. Der zugehörige Value ist ein enum-Wert, welcher in der Klasse «ToolAndMenubarActionListener» definiert wird (siehe Kapitel 5.5.4, Seite 34). Entspricht ein Toolbar-Button einem MenuBar-Button, muss in beiden Klassen zwingend der gleiche Value verwendet werden. Dieses System vereinfacht das Hinzufügen des «ActionListeners», da jedem Button in der Map der gleiche «ActionListener» hinzugefügt werden kann.

Die Buttons werden in der Methode «createToolbarButton» erstellt. Die Grösse des Buttons ist dabei von den in «constants.properties» gespeicherten Werten «ToolbarButtonWidth» und «ToolbarButtonHeight»

abhängig. Die Buttongrösse wurde mit der Icongrösse «ToolBarImageSize» von 32x32 Pixel abgestimmt. Diese drei Werte sollten nur zusammen verändert werden, wobei beachtet werden muss, dass «ToolBarImageSize» einen Wert des enums «ch.hsr.tca.gui.util.ImageLoader.ImageSize» verwenden muss.

ch.hsr.tca.gui.views.MainViewPanel

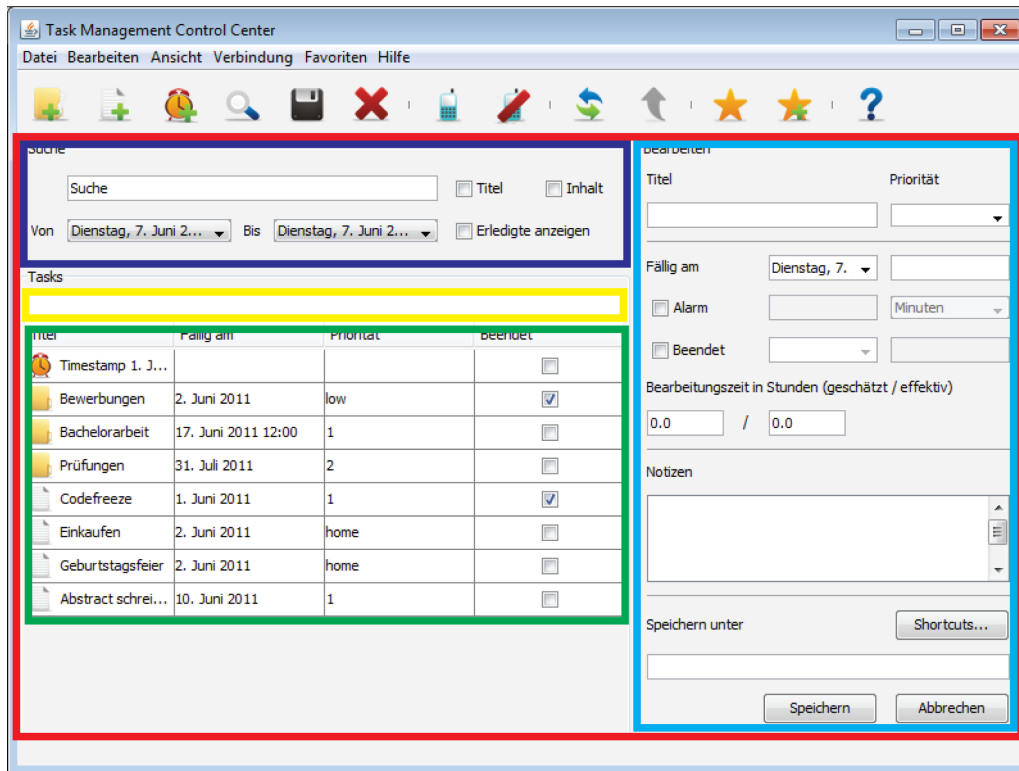


Abbildung 5.10: Die Klasse MainViewPanel im GUI (rot) aufgeteilt in die Bereiche Suchformular (dunkelblau), Pfad-Eingabefeld (gelb), Tasktabelle (grün) und Bearbeitungsbereich (hellblau)

Das MainViewPanel enthält das Suchformular, je nach Ansicht ein Pfad-Eingabefeld, die Tabelle mit den Tasks und rechts ein «JPanel», welches der Bearbeitung eines Tasks dient und ein- bzw. ausgeblendet werden kann. Angeordnet wurden diese mit einem «GridBagLayout».

Das Suchformular

Das Suchformular besitzt noch keine Aktionen, es wurde als Platzhalter für spätere Suchoperationen, wie sie im Dokument «Anforderungsspezifikation», in den Kapiteln 2.3.2.4 - 2.3.2.6, Seite 5 beschrieben werden, erstellt.

Das Pfad-Eingabefeld

Das Pfad-Eingabefeld wird nur in der Baumdarstellung angezeigt und informiert den Benutzer über den aktuell geöffneten Ordnerpfad. Es besitzt einen Keylistener, welcher sowohl auf die Eingabe eines Schrägstrichs als auch auf die Enter-Taste reagiert. Wird ein Schrägstrich eingegeben und der Pfad vor dem Schrägstrich ist korrekt, wird in der Tasktabelle sofort der Inhalt des Ordners angezeigt. Die Enter-Taste wird als abschliessende Aktion aufgefasst und es wird versucht, in den entsprechenden Ordner zu wechseln.

Die Tasktabelle

Die Tasktabelle stellt dem Benutzer eine Übersicht über seine Tasks dar. Sie wird über das «TaskTableModel» gefüllt.

Die Darstellung des Tasktitels und des Fälligkeitsdatums erfolgt mit den beiden CellRenderern «TaskCellRenderer» und «CalendarCellRenderer». Die beiden Spalten «Priorität» und «Beendet» benutzen standard Java Renderer für String bzw. boolean.

Sowohl ein Mouse- (den «RowClickListener») als auch ein Keylistener wurden auf der Tabelle registriert. Dies, damit z.B. ein Ordner sowohl mit einem Doppelklick als auch mit der Enter-Taste geöffnet werden kann.

Der Bearbeitungsbereich

Der Bearbeitungsbereich wird wegen seiner Komplexität durch ein eigenes JPanel, das «EditTaskPanel», realisiert, welches im Kapitel 5.5.1 auf Seite 21 erläutert wird.

ch.hsr.tca.gui.view.MainViewPanel.RowClickListener

Die innere Klasse RowClickListener erkennt Einfach- und Doppelklicks, indem sie, immer wenn ein Einfachklick erfolgt, einen Timer startet. Läuft der Timer ab, wurde nur einmal geklickt und der Event für einen Einfachklick wird ausgelöst. Wird, bevor der Timer abgelaufen ist, ein zweiter Klick registriert, wird der Timer abgebrochen und ein Doppelklickevent ausgelöst. Die Zeit des Timers wird über die Konstante «DoubleClickSpeed» gesteuert, welche die Zeit, in welcher zwei Klicks als Doppelklick erkannt werden, in Millisekunden angibt.

ch.hsr.tca.gui.views.Statusbar



Abbildung 5.11: Die Klasse Statusbar im GUI

Die Statusbar ist zur Zeit nur ein leeres JPanel. Sie kann zu einem späteren Zeitpunkt benutzt werden, um dem Benutzer wichtige Informationen darzustellen.

JPanel

EditTaskPanel

- logger: Logger = SyncLogManager...
- bundle: ResourceBundle
- constants: PropertyManager = new PropertyMan...
- panelObserver: Observable
- alertCheckBox: JCheckBox
- cancelButton: JButton
- completedCheckBox: JCheckBox
- completedDate: JCalendarCombo
- completedTimeFormattedTextField: JFormattedTextField
- detailsSeparator: JSeparator
- dueDate: JCalendarCombo
- dueDateLabel: JLabel
- dueTimeFormattedTextField: JFormattedTextField
- editingTimeLabel: JLabel
- effectiveTimeTextField: JFormattedTextField
- estimatedTimeTextField: JFormattedTextField
- noteLabel: JLabel
- noteScrollPane: JScrollPane
- noteTextArea: JTextArea
- pathSeparator: JSeparator
- priorityComboBox: JComboBox
- priorityLabel: JLabel
- saveAsLabel: JLabel
- saveAsTextField: JTextField
- saveButton: JButton
- shortcutsButton: JButton
- alertTimeTextField: JFormattedTextField
- alertTimeUnitComboBox: JComboBox
- titleLabel: JLabel
- titleSeparator: JSeparator
- titleTextField: JTextField
- versusLabel: JLabel
- parentFrame: JFrame
- validationGroup: ValidationGroup = new ValidationG...
- completedDateModified: boolean = false
- filepath: String = constants ...

- + EditTaskPanel(JFrame, ApplicationController)
- + addObserver(Observer) : void
- + addListeners() : void
- + showEditPanel(DomainBasicTask, String) : void
- setBasicTaskFields(DomainBasicTask, String) : void
- setPriorityTaskFields(DomainPriorityTask) : void
- setFolderFields(DomainFolder) : void
- setStandardTaskFields(DomainStandardTask) : void
- setAlertFields(DomainStandardTask) : void
- setCompletedFields(DomainStandardTask) : void
- + showEditPanelForFolderCreation(String) : void
- + showEditPanelForStandarTaskCreation(String) : void
- disableAllCriticalFields() : void
- enableFolderFields() : void
- enableStandardTaskFields() : void
- + saveValuesToBasicTask() : DomainBasicTask
- saveValuesToBasicTask(DomainBasicTask) : void
- saveValuesToPriorityTask(DomainPriorityTask) : void
- + saveValuesToFolder() : DomainFolder
- + saveValuesToStandardTask() : DomainStandardTask
- saveAlertValues(DomainStandardTask) : void
- saveCompletedValues(DomainStandardTask) : void
- saveEstAndEffTime(DomainStandardTask) : void
- getCalendarFromViewValues(JCalendarCombo, JFormattedTextField) : Calendar
- calculateAlertMinutes() : long
- cleanPanelContents() : void
- + persistPriorities() : void
- + getSavePath() : String
- + setSavePath(String) : void
- initComponents() : void
- initValidation(ApplicationController) : void

DefaultComboBoxModel

«static»

PriorityComboBoxModel

- priorities: ArrayList<String>
- selectedPriority: String

- + PriorityComboBoxModel(String)
- loadFromFile(String) : void
- + saveToFile(String) : void
- + getSize() : int
- + setSelectedItem(Object) : void
- + selectedItem() : Object

Abbildung 5.12: Package ch.hsr.tca.gui.views Teil 2

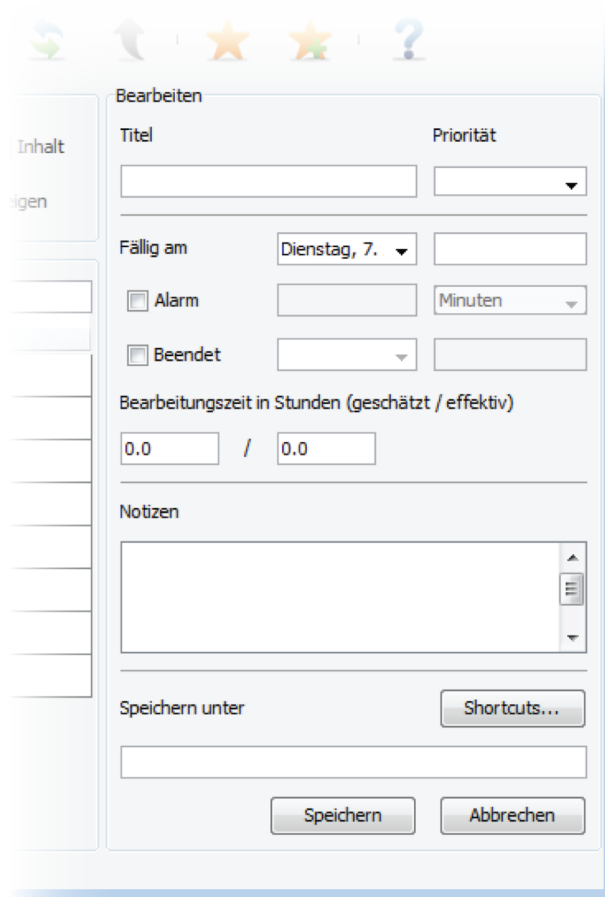
ch.hsr.tca.gui.views.EditTaskPanel

Abbildung 5.13: Die Klasse EditTaskPanel im GUI

Das EditTaskPanel wird nur angezeigt, wenn ein Task erstellt bzw. bearbeitet wird. Falls ein Ordner oder Timestamp bearbeitet wird, werden die nicht benötigten Felder ausgegraut.

Bearbeiten des «Beendet-Datums» erkennen

Um festzustellen, dass der Zeitpunkt des Beendens eines Tasks verändert wurde, wurde die Hilfsvariable «completedDateModified» eingeführt. Wird eines der beiden Felder verändert, wird «completedDateModified» auf false gesetzt, was zur Folge hat, dass sowohl die Variable «endDate» als auch «endDateModified» eines Tasks gespeichert werden.

ch.hsr.tca.gui.views.EditTaskPanel.PriorityComboBoxModel

Während dem Applikationsstart werden die Prioritäten aus einer Datei in alphabetischer Reihenfolge in die Combobox geladen. Um die Prioritäten beim nächsten Applikationsstart erneut zu Verfügung zu haben, werden alle Einträge der Combobox beim Schliessen der Applikation zurück in die Prioritäten-Datei geschrieben. Sobald ein Mobile-Device angeschlossen oder getrennt wird, werden die Prioritäten ebenfalls in die Datei geschrieben, um ausgetauscht werden zu können.



Abbildung 5.14: Package ch.hsr.tca.gui.views Teil 3

ch.hsr.tca.gui.views.DetailViewFrame

Abbildung 5.15: Die Klasse DetailViewFrame als GUI-Element

Der DetailViewFrame stellt mit Hilfe eines JEditorPanes die Informationen eines Tasks dar. Die Taskinformationen werden zur Formatierung in einen HTML-String geschrieben.

Die Shortcuts werden als Hyperlinks markiert (im Bild nicht ersichtlich), damit sie später als einfach zu bedienende Sprungmarken verwendet werden können.

ch.hsr.tca.gui.views.StartWindowMenuBar

Der Aufbau der StartWindowMenuBar gleicht demjenigen der Toolbar. Sie besitzt ebenfalls eine Map, in welcher die einzelnen «MenuItems» mit einem enum-Value abgelegt werden. Zudem besitzen die MenuItem-Items bzw. Buttons der beiden Klassen denselben ActionListener, den «ToolAndMenuBarActionListener» (s.h. Kapitel 5.5.4, Seite 34). Ist ein Befehl sowohl in derMenuBar als auch in derToolbar vorhanden, wird dies mit demselben Symbol neben dem MenuItem angezeigt.

Die Einträge der StartWindowMenuBar sind in der Tabelle 5.5 38 ersichtlich.

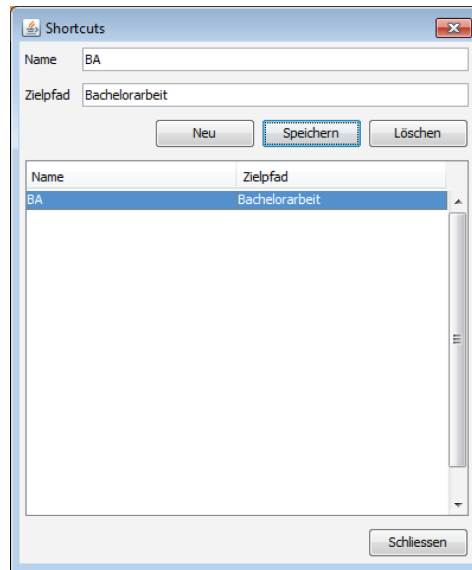
ch.hsr.tca.gui.views.ShortcutDialog

Abbildung 5.16: Die Klasse ShortcutDialog als GUI-Element

Der ShortcutDialog ist ein modaler Dialog. Er ermöglicht dem Nutzer einen oder mehrere Shortcuts anzulegen. Es wurde die Methode «showDialog» eingeführt, welche eine Referenz auf die im TableModel verwendete Shortcut-Liste zurück gibt.

ch.hsr.tca.gui.view.ShortcutDialog.ShortcutTableRowListener

Der ShortcutTableRowListener erkennt einen Klick auf eine Tabellenzeile und füllt die Felder «nameTextField» und «targetTextField» mit den entsprechenden Informationen.

5.5.2 ch.hsr.tca.gui.controller

Die Klassen des Packages «ch.hsr.tca.gui.controller» steuern die Aktionen der Klassen des Packages «ch.hsr.tca.gui.views».

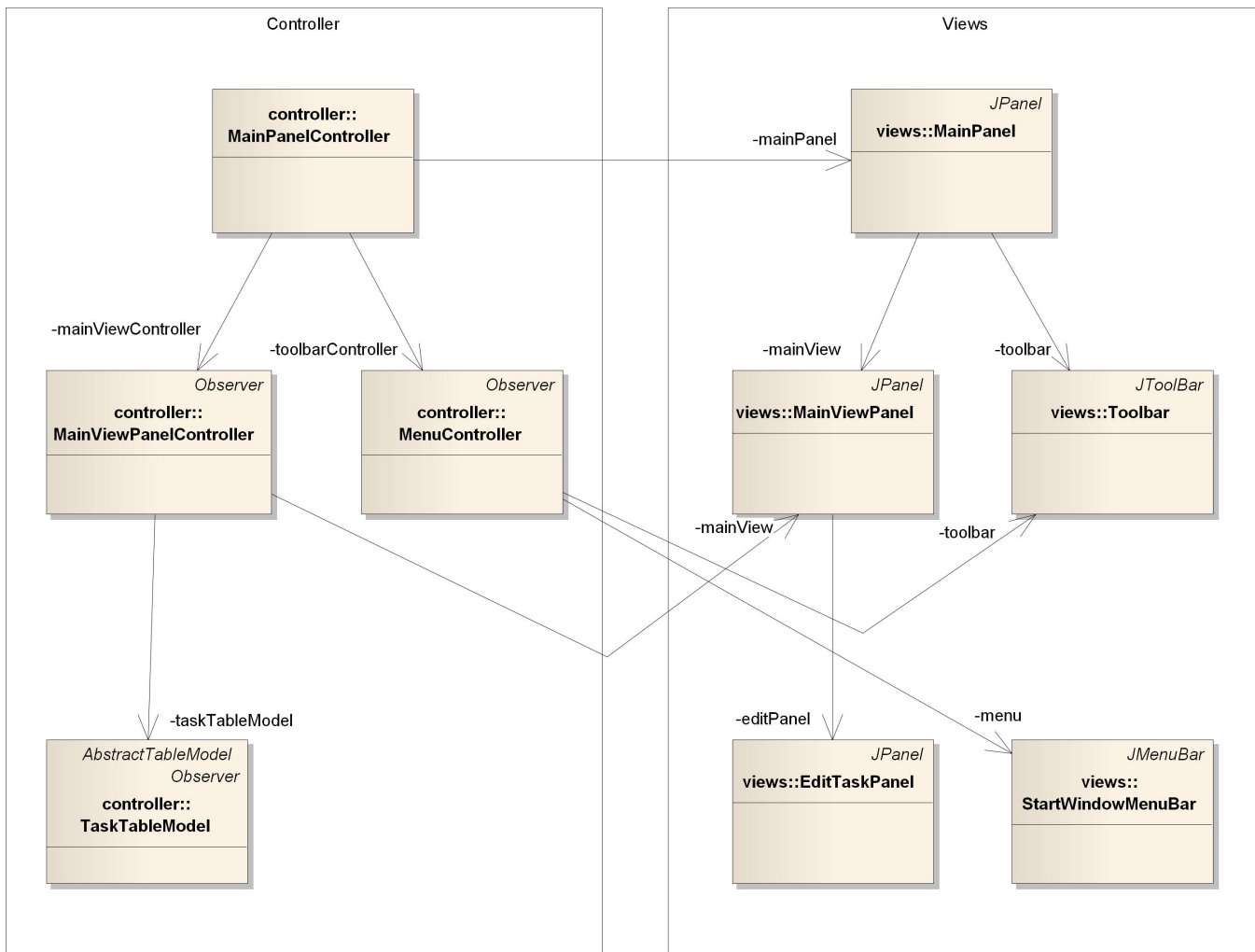


Abbildung 5.17: Die Controller- und ihre View-Klassen

Die drei Hauptcontroller kennen ihre Viewklasse(n). Sie werden bereits während dem Initialisieren der Applikation vom nächst höheren bzw. vom ApplicationController miteinander bekannt gemacht.

Das TaskTableModel hilft dem MainViewPanelController, die Tasktabelle des MainViewPanels zu steuern. Da das EditTaskPanel ein Teil des MainViewPanels ist, übernimmt der MainViewPanelController die Steuerung beider JPanels.

Die Klasse StatusBar besitzt momentan noch keinen eigenen Controller, da sie noch keine Funktion übernimmt. Wird dies geändert, muss auch für die StatusBar ein eigener Controller, auf gleicher Ebene wie der MainViewPanel- bzw. MenuController, eingeführt werden.

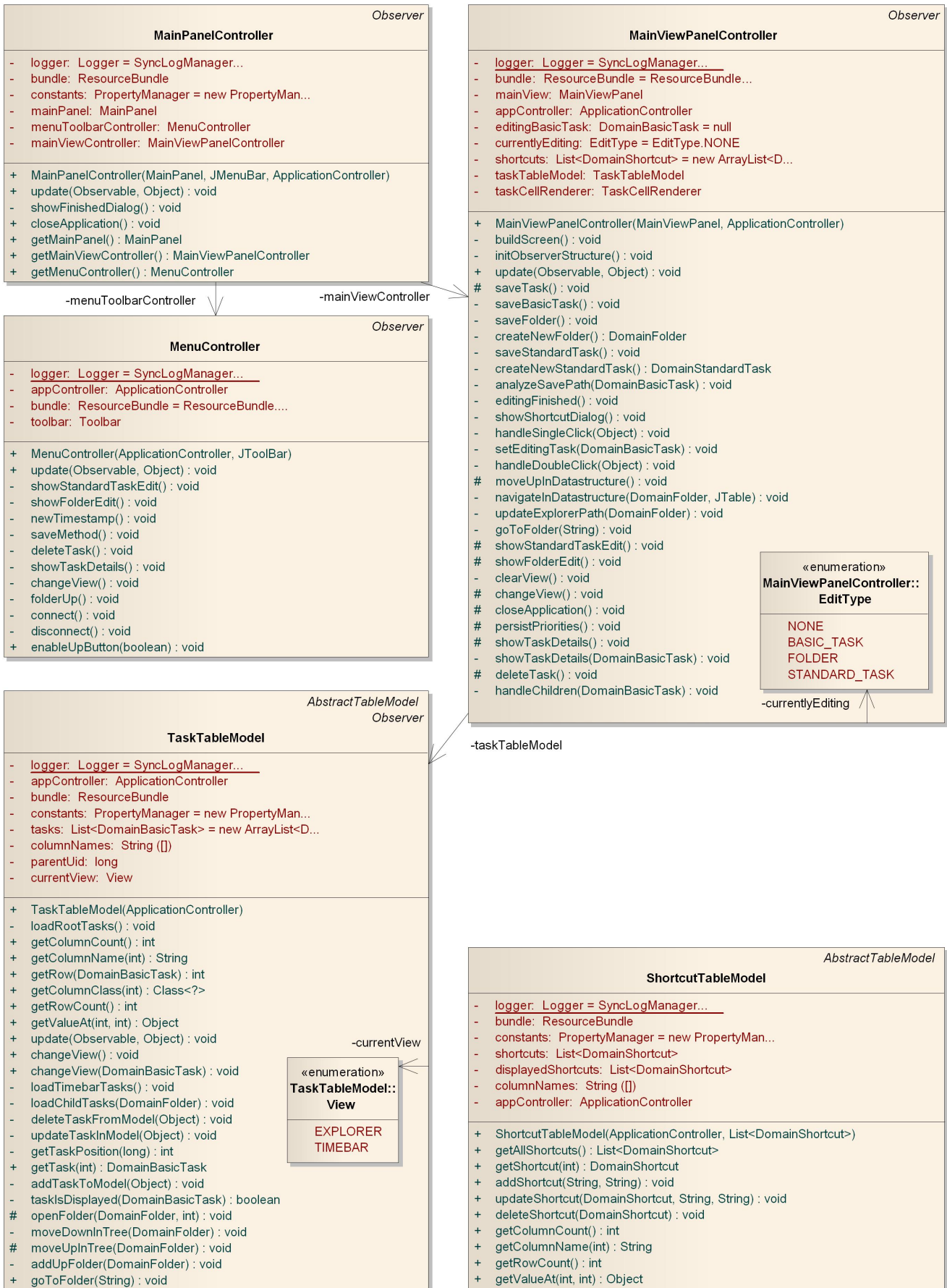


Abbildung 5.18: Package ch.hsr.tca.gui.controller

ch.hsr.tca.gui.controller.MainPanelController

Der MainPanelController besitzt wenig eigene Logik, da das MainPanel keine eigenen Funktionen übernimmt. Er ist lediglich dazu da, die «Subcontroller» zu initialisieren und mit ihren View-Komponenten bekannt zu machen. Zudem leitet er das Schliessen der Applikation an die «Subcontroller» weiter und gewährt dem ApplicationController Zugriff auf die «Subcontroller».

Events

Der MainPanelController ist ein Observer und fängt einen Event ab. Details zu der Kommunikation mit Events kann dem Kapitel 5.7, Seite 35 entnommen werden.

Event	Aktion
SYNCHRONISATION_FINISHED	Die Dateisynchronisation wurde beendet. Es soll ein Dialog-Fenster angezeigt werden.

Tabelle 5.1: Abgefangener Event des MainPanelController und dessen Aktion

ch.hsr.tca.gui.controller.MainViewPanelController

Dies ist der Hauptcontroller des GUIs. Er regelt alle Abläufe ausser denjenigen, die ausschliesslich die Tool- oder Menubar betreffen.

Bearbeiten oder Erstellen eines Tasks

Die Variable «editingBasicTask» enthält immer den Task, der aktuell im EditTaskPanel angezeigt wird bzw. null, falls ein neuer Task erstellt wird. Dies ist notwendig, da die Informationen des ursprünglichen Tasks zum Zeitpunkt des Speicherns noch vorhanden sein müssen. Zudem wird in der Variable «currentlyEditing» der Typ des bearbeiteten/erstellten Tasks gespeichert.

Soll der Task gespeichert werden, geschehen folgende Schritte:

1. Der Typ wird anhand der Variable «currentlyEditing» ermittelt.
2. Die Informationen, welche vom EditTaskPanel bereitgestellt werden, werden in einen temporären Task geschrieben. Diesem wird die Shortcutliste hinzugefügt.
3. Es wird überprüft, ob die Variable «editingBasicTask» null ist.
 - a) Die Variable ist null: Es muss ein neuer Task erstellt werden. Der temporäre Task kann sofort als neuer Task an das ControlCenter weitergeleitet werden. Dieser Fall darf für einen BasicTask niemals vorkommen, da ein BasicTask direkt erstellt wird.
 - b) Die Variable enthält einen Task: Der temporäre Task und der ursprüngliche Task werden zusammen für ein Update an das ControlCenter weitergereicht.
4. Die Felder des EditTaskPanel werden geleert und das Panel wird ausgeblendet.

Events

Der MainViewPanelController ist ein Observer und fängt eine Reihe von Events ab. Details zu der Kommunikation mit Events kann dem Kapitel 5.7, Seite 35 entnommen werden.

Event	Aktion
ABORT_NEW_TASK	Das Bearbeiten/Erstellen eines Tasks soll abgebrochen werden. Die Felder des EditTaskPanels werden geleert und das Panel auf invisible gesetzt.
DOUBLE_CLICK_ON_TABLE_ROW	Ein Doppelklick erfolgte auf einer Tabellenzeile. Es muss unterschieden werden, ob die Zeile einen Ordner enthält oder nicht. Im Falle eines Ordners wird dem TaskTableModel die Anweisung gegeben, in den Ordner zu springen (siehe. Kapitel 5.5.2, Seite 30). Enthält die Zeile keinen Ordner, wird ein neuer DetailViewFrame erstellt.
GO_TO_FOLDER	Es soll in den, im angehängten Objekt mitgegebenen Ordner gesprungen werden. Das TaskTableModel wird angewiesen, die Inhalte des entsprechenden Ordners anzuzeigen.
SAVE_TASK	Der momentan in Bearbeitung befindliche Task soll erstellt bzw. verändert werden. Der MainViewPanelController ist, wie im vorigen Abschnitt erläutert, in der Lage, zu erkennen was und wohin er speichern soll.
SHOW_SHORTCUT_DIALOG	Öffnet den ShortcutDialog. Da der Dialog den GUI-Thread sperrt, wird sofort eine Referenz auf die Shortcuts-Liste gespeichert. Diese wird benötigt, um später den bearbeiteten Task speichern zu können.
SINGLE_CLICK_ON_TABLE_ROW	Ein Einfachklick erfolgte auf einer Tabellenzeile. Das EditTaskPanel wird mit den Taskinformationen gefüllt und sichtbar gemacht. Zudem wird der Task und dessen Typ (Basic- oder StandardTask oder Folder) im Controller zwischengespeichert.

Tabelle 5.2: Abgefangene Events des MainViewPanelController und ihre Aktionen

ch.hsr.tca.gui.controller.MenuController

Der MenuController ist zuständig für alle Aktionen, welche von der StartWindowMenuBar oder der Toolbar ausgelöst wurden. Die Klasse ToolAndMenubarActionListener erzeugt entsprechend dem angeklickten Button einen Event.

Events

Der MenuController ist ein Observer und fängt eine Reihe von Events ab. Details zu der Kommunikation mit Events kann dem Kapitel 5.7, Seite 35 entnommen werden.

Event	Aktion
CHANGE_VIEW	Es soll zwischen Baum- und Zeitstrahlview gewechselt werden. Der MainViewController wird angewiesen, dies in die Wege zu leiten. Wird die Zeitstrahldarstellung angezeigt und es ist ein Task selektiert, wird dieser Task auch in der Baumdarstellung angezeigt. Das heisst, es wird in den übergeordneten Ordner des selektierten Tasks gesprungen.
CONNECT_TO_DEVICE	Ein Mobile-Device soll verbunden werden. Das ControlCenter wird angewiesen, eine Verbindung aufzubauen.
DELETE_TASK	Der MainViewPanelController wird angewiesen, den selektierten Task zu löschen.
DISCONNECT_FROM_DEVICE	Das ControlCenter wird angewiesen, die bestehende Verbindung zu schliessen.
EXIT_APPLICATION	Die PC-Applikation soll geschlossen werden. Mithilfe des Application-Controllers wird der Schliessvorgang gestartet.
FOLDER_UP	Es soll in den darüber liegenden Ordner gewechselt werden. Dieser Event kann nur entstehen, wenn es einen übergeordneten Ordner gibt, da der Toolbar-Button ansonsten deaktiviert ist. Mithilfe des MainView-Panels wird der Inhalt des neuen Ordners angezeigt.
NEW_FOLDER	Es soll ein neuer Folder erstellt werden. Mit Hilfe des MainViewPanelControllers werden die nötigen Aktionen ausgeführt, um das Edit-TaskPanel darzustellen.
NEW_TASK	Es soll ein neuer StandardTask erstellt werden. Mit Hilfe des MainViewPanelControllers werden die nötigen Aktionen ausgeführt, um das Edit-TaskPanel darzustellen.
NEW_TIMESTAMP	Es soll ein neuer BasicTask erstellt werden. Dieser wird direkt zur Speicherung an das ControlCenter übermittelt.
SAVE_TASK	Der Speicherbefehl wird an den MainViewPanelController weitergeleitet.
SHOW_DETAILS	Der MainViewPanelController wird angewiesen, den DetailViewFrame eines Tasks zu erzeugen.

Tabelle 5.3: Abgefangene Events des MenuControllers und ihre Aktionen

ch.hsr.tca.gui.controller.TaskTableModel

Das TaskTableModel ist für den Inhalt der Tasktabelle des MainViewPanels zuständig. Standardmässig sind nach dem Starten der Applikation die Roottasks (also in der Baumdarstellung) geladen.

Das TaskTableModel wechselt bei Bedarf den Inhalt der Tabelle aus. Einerseits kann es alle Tasks, sortiert nach dem Datum, enthalten, andererseits kann auch nur ein Teil der Tasks dargestellt werden, wenn die Baumdarstellung genutzt wird. Sortiert werden die Tasks mit den zwei Klassen «Explorer-» bzw. «TimebarComparator», Details siehe Kapitel 6.5, Seite 49.

Um die Navigation mit einem «..»-Ordner zu ermöglichen, muss künstlich ein neues Folder-Objekt erstellt werden. Dieses Objekt hat den Titel «..» und das gleiche Parent, wie der aktuell geöffnete Ordner. Dieser neue «Hilfsordner» wird immer zu oberst in der Liste einsortiert.

ch.hsr.tca.gui.controller.ShortcutTableModel

Das ShortcutTableModel regelt den Inhalt der Tabelle im ShortcutDialog.

Die Klasse besitzt zwei Listen, welche zur Darstellung bzw. zur Speicherung der Shortcuts benötigt werden. Dies ist nötig, da ein Shortcut nicht einfach aus der Liste gelöscht werden kann, sondern zum Löschen markiert werden muss. Das eigentliche Löschen übernimmt das ControlCenter, welches den Löschvorgang zudem an die Datenbank weitergibt. Um dem Nutzer anzuzeigen, dass der Shortcut gelöscht wird, wird die zweite Liste benötigt, von welcher der Shortcut gelöscht wird und so aus der angezeigten Tabelle verschwindet.

5.5.3 ch.hsr.tca.gui.validation

Das Package «ch.hsr.tca.gui.validation» enthält alle Klassen welche zur Validierung eines Textfeldes benötigt werden. Die Hauptklasse AbstractValidator bestimmt dabei, was während der Validierung geschieht und wie ein Fehler angezeigt wird. Die abgeleiteten Klassen implementieren lediglich das Validations-Kriterium.

Um die implementierenden Subklassen des AbstractValidators nutzen zu können, muss eine Instanz der Subklasse mit «setInputVerifier» an das gewünschte JTextField gehängt werden.



Abbildung 5.19: Package ch.hsr.tca.gui.validation

ch.hsr.tca.gui.validation.AbstractValidator

Eine fehlgeschlagene Validierung wird mit einem JDialog (ohne Fenster) und der veränderten Hintergrundfarbe des JTextFields angezeigt.

Die Hintergrundfarben des JDialogs und des JTextFields werden durch die Konstanten «ErrorPopupColor» und «TextFieldErrorColor» bestimmt.

Die eigentliche Validierung findet in der Methode «verify» statt. Diese sollte nicht verändert werden, da sie Auswirkungen auf alle abgeleiteten Validatoren hat. Es ist sinnvoller, eine abgeleitete Klasse zu erstellen und die abstrakte Methode «validationCriteria» zu implementieren.

Die Klasse implementiert das Interface «KeyListener», damit der Errortext gelöscht bzw. die Hintergrundfarbe zurückgesetzt werden kann, nachdem eine Eingabe erfolgte und das Feld erneut validiert werden muss.

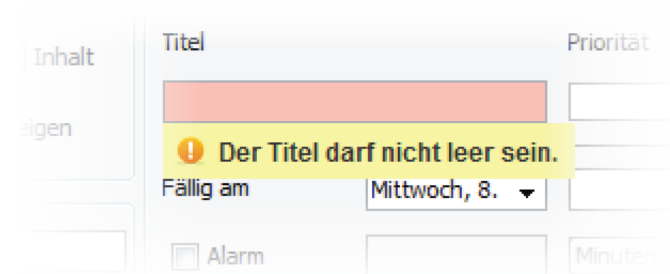


Abbildung 5.20: Beispiel einer fehlgeschlagenen Validierung

ch.hsr.tca.gui.validation.PathValidator

Der PathValidator erkennt, ob der eingegebene Pfad existiert und auf einen Ordner zeigt. Er wird für das Pfad-Eingabefeld des MainViewPanels und das Pfadfeld des EditTaskPanels gebraucht.

ch.hsr.tca.gui.validation.NotEmptyValidator

Der NotEmptyValidator überprüft, ob das TextFeld ausgefüllt wurde oder nicht. Er wird im ShortcutDialog gebraucht, um das Namentextfeld zu überprüfen.

ch.hsr.tca.gui.validation.ShortcutPathValidator

Der ShortcutPathValidator ähnelt sehr dem PathValidator. Er lässt jedoch auch korrekte Pfade, welche einen Basic- oder StandardTask als Ziel haben, zu. Er wird in der Klasse ShortcutDialog zur Überprüfung des Shortcutziels gebraucht.

ch.hsr.tca.gui.validation.TitleValidator

Der TitleValidator überprüft einerseits, ob der eingegebene Titel ein gültig ist und andererseits, ob er innerhalb des zu speichernden Ordners eindeutig ist. Er wird im EditTaskPanel zur Validierung des Titelfeldes benötigt.

ch.hsr.tca.gui.validation.ValidationGroup

Mehrere Validatoren können in einer ValidationGroup zusammengefasst und zur selben Zeit ausgewertet werden. Dies ist vorallem dann nützlich, wenn z.B. ein Speichern-Button gedrückt wurde.

5.5.4 ch.hsr.tca.gui.util

Im Package «ch.hsr.tca.gui.util» werden alle Hilfsklassen, welche ausschliesslich für das GUI verwendet werden, zusammengefasst.

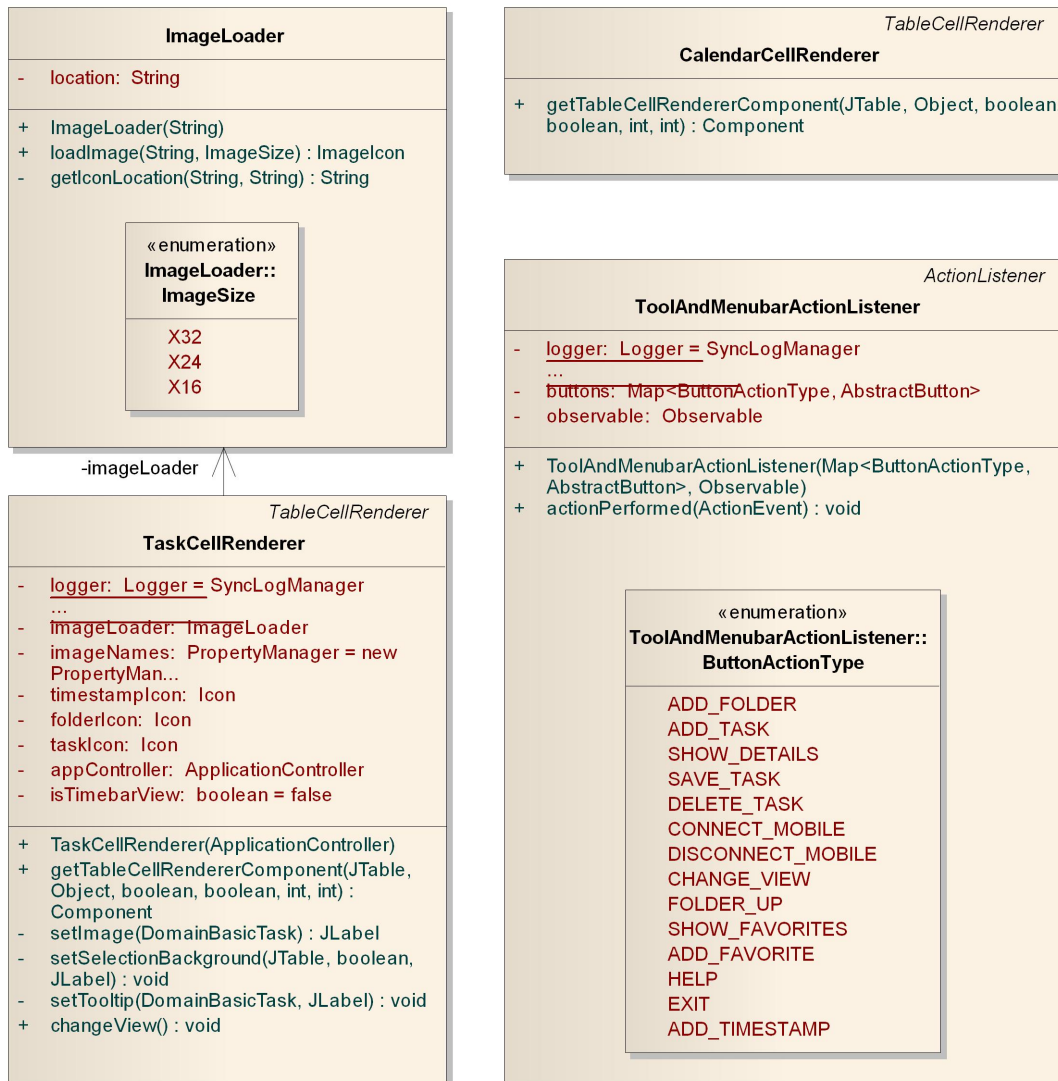


Abbildung 5.21: Package ch.hsr.tca.gui.util

ch.hsr.tca.gui.util.ImageLoader

Die Klasse ImageLoader wird dazu gebraucht, um Bilder einer bestimmten Grösse zu laden. Dazu kann im Konstruktor der Klasse der Speicherort angegeben werden.

Die Ordnerstruktur des Speicherorts muss wie folgt gestaltet sein: Alle Bilder der Grösse 32x32 Pixel liegen in einem Unterordner mit dem Namen «X32» und werden durch den enum-Wert «ImageSize.X32» beschrieben. Dies kann beliebig erweitert werden, jeder neue enum-Wert muss einem Ordernamen ent-

sprechen. Somit können Bilder gleichen Namens in verschiedenen Grössen auf einfache Art und Weise geladen werden.

ch.hsr.tca.gui.util.TaskCellRenderer

Der TaskCellRenderer implementiert das Interface «TableCellRenderer» und konstruiert, aus den Informationen eines Tasks, ein in einer Tabelle darstellbares Label. Dieses Label enthält ein Icon, welches den Tasktyp und den Titel des Tasks darstellt. Zudem wird in dieser Klasse der Tooltip mit dem Speicherpfad eines Tasks erstellt.

ch.hsr.tca.gui.util.CalendarCellRenderer

Der CalendarCellRenderer implementiert das Interface «TableCellRenderer» und stellt das Fälligkeitsdatum eines Tasks dar. Er unterscheidet sich minimal vom Standard-CellRenderer für ein Datum. Er stellt die Zeit nur dar, falls die Zeit nicht 0:00 lautet.

ch.hsr.tca.gui.util.ToolAndMenubarActionListener

Die Klasse ToolAndMenubarActionListener dient den beiden View-Klassen StartWindowMenuBar und Toolbar als ActionListener ihrer MenuItems bzw. Buttons. Im Konstruktor erhält er eine Map der Original-Buttons sowie einen Observer, welchen er bei einem ActionEvent zu informieren hat. Die überschriebene Methode «actionPerformed» überprüft, um welchen Button es sich handelt und erstellt dann, entsprechend dem ButtonActionType, den nötigen Event. Eine Übersicht über die Verknüpfung von Button und Event kann dem Kapitel 5.7 auf Seite 35 entnommen werden.

5.6 ch.hsr.tca.util

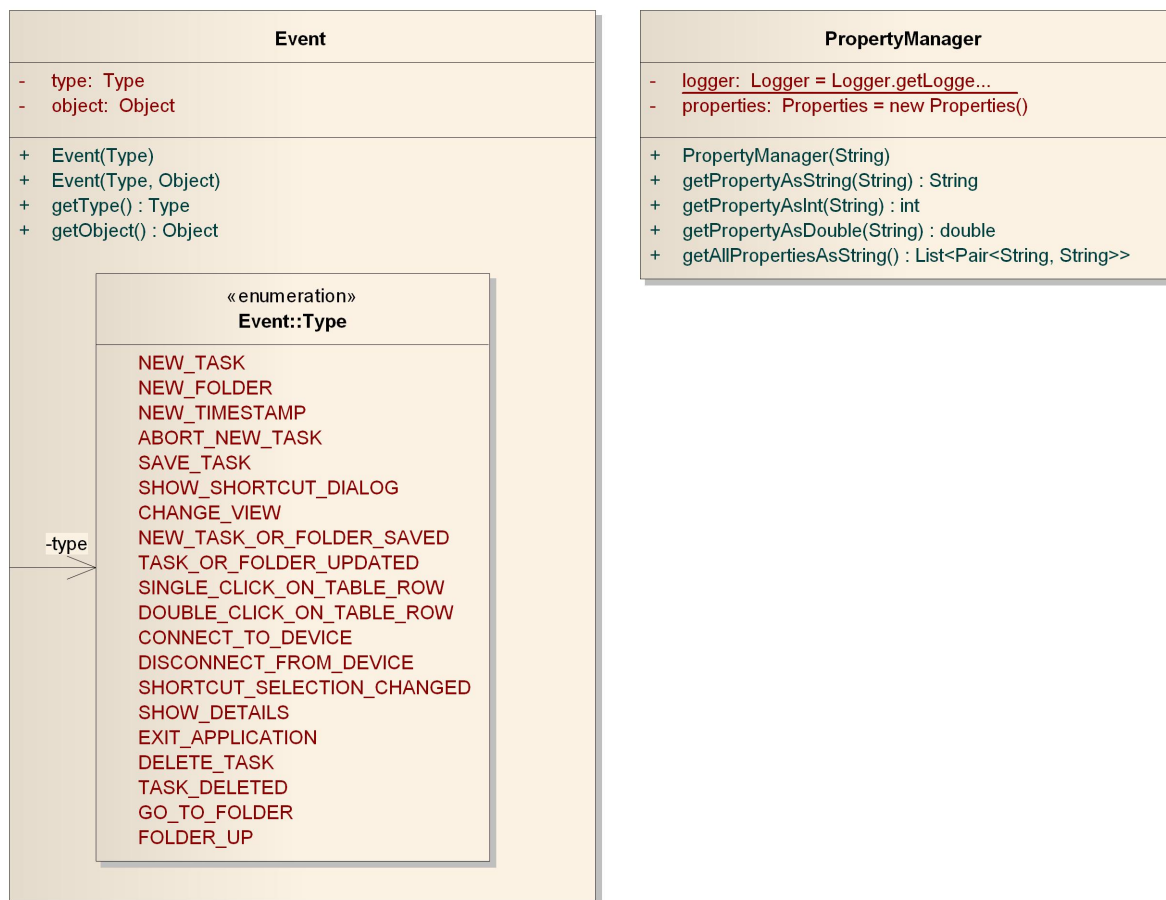


Abbildung 5.22: Package ch.hsr.tca.util

ch.hsr.tca.util.Event

Die Klasse Event wird gebraucht, um die Observerklassen über ein aufgetretenes Ereignis zu informieren. Ein Event hat zwingend einen Typ und optional ein Objekt. Der Event wird erstellt, sobald in einer Observableklasse die Methode «notifyObservers» aufgerufen wurde. Eine Übersicht, welcher Event von wem erzeugt und welche Klasse ihn verarbeitet, kann im Kapitel 5.7, Seite 35 gefunden werden.

ch.hsr.tca.util.PropertyManager

Mit einem PropertyManager kann eine Properties-Datei geladen und die Inhalte nach den entsprechenden Schlüsseln durchsucht werden. Anders als die Javaklasse «Properties», welche dies auch kann, kann er die Variablen auch als int oder double zur Verfügung stellen.

Diese Klasse kann einfach erweitert werden, wenn weitere Datentypen wie z.B. boolean benötigt werden. Sie nimmt der aufrufenden Klasse das Casten des Properties ab.

5.7 Kommunikation mit Events

Um eine möglichst unabhängige Navigation einzuführen, wurde beschlossen, dass View und Controller nur über eine Observer-Struktur verbunden sind. Somit ist es egal, wieviele Klassen auf einen Event reagieren.










Die folgende Tabelle bietet eine Übersicht über die durch die Observerstruktur verbundenen Klassen und die ausgetauschten Events.

Observable	Observer	ausgetauschte Events
ControlCenter	TaskTableModel	NEW_TASK_OR_FOLDER_SAVED TASK_DELETED TASK_OR_FOLDER_UPDATED
	MainPanelController	SYNCHRONISATION_FINISHED
EditTaskPanel	MainViewPanelController	ABORT_NEW_TASK SAVE_TASK SHOW_SHORTCUT_DIALOG
MainViewPanel	MainViewPanelController	DOUBLE_CLICK_ON_TABLE_ROW GO_TO_FOLDER SINGLE_CLICK_ON_TABLE_ROW
StartWindowMenuBar (mit Hilfe des ToolAndMenubarActionListeners)	MenuController	Die ausgetauschten Events werden in der folgenden Tabelle näher erläutert.
Toolbar (mit Hilfe des ToolAndMenubarActionListeners)	MenuController	Die ausgetauschten Events werden in der folgenden Tabelle näher erläutert.

Tabelle 5.4: Observer-Struktur und ausgetauschte Events

5.7.1 Events der Tool- und Menubar

In der folgenden Tabelle werden die Events, welche von der Menu- bzw. Toolbar versendet werden, aufgelistet. Die Events werden nicht in den Klassen selbst generiert, sondern mit Hilfe des ToolAndMenubarActionListeners kontrolliert. Da Tool- und Menubar einen Grossteil der Buttons gemeinsam besitzen, werden nur die Icons der Toolbar und die Namen der Menüitems im GUI erwähnt. Enthält eine Zeile keinen Event, wurde die Funktion des Buttons noch nicht implementiert.

Button	ausgetauschter Event	ausgeführte Aktion
Menu Datei		
Beenden	EXIT_APPLICATION	Die Applikation wird geschlossen.
Menu Bearbeiten		
 Neuer Ordner	NEW_FOLDER	Im Bearbeitungsbereich rechts kann ein neuer Ordner erstellt werden.
 Neuer Task	NEW_TASK	Im Bearbeitungsbereich rechts kann ein neuer StandardTask erstellt werden.
 Neuer Timestamp	NEW_TIMESTAMP	Es wird ein neuer BasicTask den Roottasks hinzugefügt.
 Speichern	SAVE_TASK	Der sich aktuell in Bearbeitung befindliche Task wird gespeichert.
 Löschen	DELETE_TASK	Der in der Tasktabelle selektierte Task wird gelöscht.
Menu Ansicht		
 Ansicht wechseln	CHANGE_VIEW	Die Taskdarstellung wird von Baum- zu Zeitstrahldarstellung gewechselt, bzw. von Zeitstrahl- zu Baumdarstellung.
 Details anzeigen	SHOW_DETAILS	Die Details eines Tasks werden in einem eigenen Frame angezeigt.
Menu Verbindung		
 Verbindung herstellen	CONNECT_TO_DEVICE	Es wird eine Verbindung zu dem Mobile-Device aufgebaut.
 Verbindung trennen	DISCONNECT_FROM_DEVICE	Die aktive Verbindung zum Mobile-Device wird geschlossen.





Button	ausgetauschter Event	ausgeführte Aktion
Menu Favoriten		
 Zu Favoriten hinzufügen -		Ein Task wird im Favoritenmenü abgelegt.
Menu Hilfe		
 Hilfe -		Das Hilfefenster wird geöffnet.
Toolbarbuttons ohne Entsprechung in der Menubar		
 -	FOLDER_UP	Die Tasktabelle zeigt den Inhalt des übergeordneten Ordners an.
 -	-	Die als Favoriten markierten Tasks werden angezeigt.

Tabelle 5.5: Tool- und Menubar Buttons und ihre Events

5.8 Mehrsprachigkeit der Applikation

Die Applikation ist auf Mehrsprachigkeit ausgelegt. Deshalb wurden mehrere Properties-Dateien erstellt, welche als «ResourceBundle» in die Applikation eingebunden werden. Die Sprache lässt sich über die Konstante «GUILanguage» steuern. Es ist weder möglich, die Sprache zur Laufzeit zu wechseln, noch ist das Menü Ansicht/Sprache mit dieser Konstanten verknüpft.

Momentan stehen die Sprachen Englisch und Deutsch zur Verfügung.

5.9 Logging

Das Logging wird von den beiden Konstanten «LogLevel» und «Logpackage» gesteuert. Der Benutzer kann somit frei bestimmen, welche Logausgaben erstellt werden.

6 TCA Android

Das Android-Projekt beinhaltet alle Packages, welche für die Logik und Darstellung der Android-Applikation benötigt werden. Nebst diesen Packages sind auch jene enthalten, welche Klassen beinhalten, die sowohl im PC- als auch im Android-Projekt benötigt werden.

6.1 Bugs in Android

Da in Android Returnstatements «doppelt» ausgeführt werden, musste jeweils eine Rückgabe-Variable eingeführt werden, welcher der Return-Wert zugewiesen werden kann. Anbei ein Beispiel zum Verständnis:

```
if(true) return 0;
return 1;
```

Der Rückgabewert wird wider Erwarten immer 1 sein, da Android in jedem Fall das letzte Return-Statement ausführen wird. Für weitere Informationen sei auf http://android.git.kernel.org/?p=platform/dalvik.git;a=blob_plain;f=docs/debugger.html;hb=HEAD verwiesen.

6.2 Klassenstruktur

Folgende Abbildung gibt einen Überblick über die einzelnen Packages und deren Klassen.

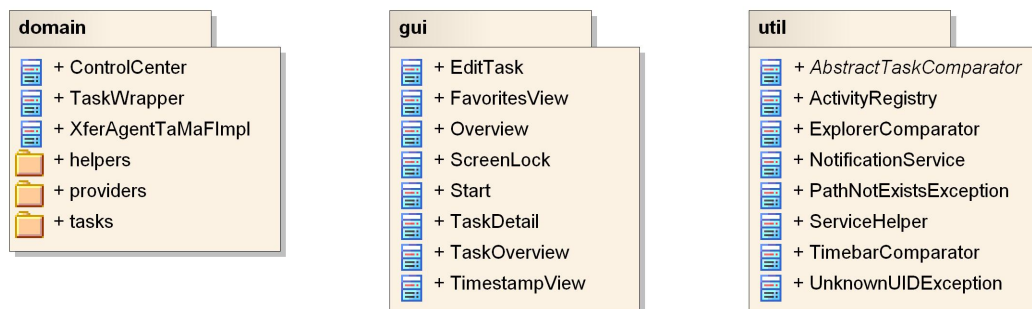


Abbildung 6.1: Klassenstruktur Mobile

Im folgenden wird auf die einzelnen Packages eingegangen.

6.3 ch.hsr.tca.domain

In diesem Package befindet sich die eigentliche Logik des Programms.

6.3.1 ch.hsr.tca.domain.helpers

Für jede Datenbanktabelle existiert eine eigene «Helper-Klasse», welche die, für die Erstellung und Bearbeitung der Tabellen erforderlichen Konstanten zur Verfügung stellt. Diese Hilfsklassen ermöglichen eine

saubere Implementation der Datenbank und verhindern, dass Tippfehler entstehen. Folgende Hilfsklassen werden angeboten:

- BasicTaskHelper
- ChildrenHelper
- FolderHelper
- PriorityTaskHelper
- ShortcutHelper
- StandardTaskHelper

Details zum Datenbankschema können dem Kapitel 4, Seite 5 entnommen werden.

6.3.2 ch.hsr.tca.domain.providers

ch.hsr.tca.domain.providers.TaskContentProvider

Die Klasse TaskContentProvider implementiert die Datenschnittstelle, über die eine Anwendung ihre privaten Daten ganz oder teilweise für andere Anwendungen des gleichen Systems zugänglich machen kann. Der Entscheid, den Datenbankzugriff mittels eines ContentProviders zu implementieren, wurde daher gefällt, da in Android Datenbanken nur über einen ProviderTestCase¹ ausreichend getestet werden können und hierfür ist ein ContentProvider erforderlich.

Die Overrides der folgenden Methoden werfen eine SQLException, die im TaskWrapper gefangen wird.

- query()
- insert()
- update()
- delete()

Diese Exceptions werden erst dort in den public Methoden gefangen, damit der ganze Vorgang abgebrochen werden kann und keine halben Tasks zurückgegeben werden können. Details zum TaskWrapper können der Seite 42 entnommen werden.

6.3.3 ch.hsr.tca.domain.tasks

Diese Tasks wurden erstellt, um Referenzen zwischen den Objekten ausnutzen zu können und die Taskverbindungen nicht ausschliesslich über die UID lösen zu müssen. Dieses Package liegt im Android-Projekt, da Android im Gegensatz zu Java ein externes Package nicht akzeptiert.

Im Allgemeinen «wrappen» diese Klassen die Task-Klassen, welche für den Austausch mit TaMaF gebraucht werden. Sie besitzen alle eine Methode «getAs...» um den darin enthaltenen ursprünglichen TaMaF-Task zurück zu bekommen.

¹<http://developer.android.com/reference/android/test/ProviderTestCase2.html>



Abbildung 6.2: Package ch.hsr.tca.domain.tasks

ch.hsr.tca.domain.tasks.DomainBasicTask

Ein DomainBasicTask besitzt eine Referenz auf sein Parent. Wird die «getAsBasicTask»-Methode aufgerufen, wird zuerst die unter Umständen neue UID in den BasicTask geschrieben.

ch.hsr.tca.domain.tasks.DomainPriorityTask

Der DomainPriorityTask ist wie seine TaMaF-Klasse PriorityTask abstrakt. Er gibt den ableitenden Klassen jedoch vier neue Methoden zur Implementation vor: «getEstimatedTime», «getEffectiveTime», «isCompleted» und «getEndDate». Diese Methoden werden sowohl für den DomainFolder als auch für den DomainStandardTask benötigt.

ch.hsr.tca.domain.tasks.DomainFolder

Der DomainFolder enthält eine Map mit den Tasktiteln als Keys und den Childtasks als Values. Auch hier gilt wie beim DomainBasicTask, bevor der Folder in der Methode «getAsFolder» zurückgegeben wird, werden alle UIDs der Children gespeichert.

Da der DomainFolder weder beendet werden kann, noch eine geschätzte bzw. gebrauchte Zeit aufweist, werden die Werte, wie im Dokument «Domainanalyse», Kapitel 2.2, Seite 4 gefordert, aus dem Inhalt des DomainFolders berechnet.

ch.hsr.tca.domain.tasks.DomainStandardTask

Der DomainStandardTask besitzt keine eigene Logik, er leitet alle Aufrufe direkt an den darin enthaltenen StandardTask weiter. Er besitzt eine Variable, um einen NotificationService (siehe Kapitel 6.5, Seite 49) zu speichern.

ch.hsr.tca.domain.tasks.DomainShortcut

Der DomainShortcut besitzt eine Variable, um auf den Zieltask referenzieren zu können. Zudem wurde ein «changedFlag» eingeführt. Dies ist nötig, damit bei einem bestehenden Task die Shortcutliste verändert werden kann, ohne alle DomainShortcuts mit der vorigen Version vergleichen zu müssen. Mögliche Werte für das «changedFlag» sind durch den enum «ShortcutDirty» gegeben.

ch.hsr.tca.domain.ControlCenter

Das ControlCenter bildet auch hier das Herzstück der Applikation. Es ist analog demjenigen in der PC-Applikation aufgebaut und übernimmt dieselben Funktionen. Details dazu können dem Kapitel 5.4, Seite 14 entnommen werden.

ch.hsr.tca.domain.TaskWrapper

Der TaskWrapper dient als Bindeglied zwischen dem TaskContentProvider und dem ControlCenter der Mobile-Applikation. Die Klasse besteht aus ähnlichen Teilen wie der TaskDBAdapterPC (siehe Kapitel 5.4, Seite 12). Folgende Teilbereiche sind vorhanden:

1. Auslesen der Tasks
2. Speichern eines neuen Tasks
3. Updaten eines Tasks
4. Löschen eines Tasks

5. Hilfsmethoden für das Auffüllen der «ContentValues» (diese werden über den TaskContentProvider der Datenbank übermittelt)

Diese Bereiche enthalten public Methoden, welche es erlauben, alle Task-Arten und die «Shortcuts» zu bearbeiten. «Shortcuts» werden auf der Mobile-Seite nur im Zusammenhang mit dem Löschen eines Tasks entfernt, ansonsten können sie weder bearbeitet noch erstellt werden.

Entgegen der Lösung auf der PC-Seite werden die «Shortcuts» nicht atomar mit dem zugehörigen Task gelöscht. Es kann daher sein, dass sich noch «Shortcut»-Reste in der Datenbank befinden, falls ein Task erfolgreich gelöscht wird, das Löschen der zugehörigen «Shortcuts» jedoch fehlschlägt. Dies hat jedoch auf die sichtbaren «Shortcuts» keine Konsequenzen, da «Shortcuts» nur im Zusammenhang mit einem Task angezeigt werden und keine eigenständige, von den zugehörigen Tasks unabhängige Darstellung besitzen.

ch.hsr.tca.domain.XferAgentTaMaFImpI

Diese Klasse regelt das Verhalten bei einem Taskaustausch. Der Austauschvorgang wird detailliert im Kapitel 7 auf Seite 54 beschrieben.

6.4 ch.hsr.tca.gui

Alle in diesem Package enthaltenen Klassen dienen der Visualisierung der Benutzeroberfläche und erben von der Klasse «Activity». Activities verknüpfen die im Ordner res enthaltenen Ressourcen mit der Anwendung. Sie implementieren die Geschäftsprozesse, werten Dialogeingaben aus oder reagieren auf eine Menüauswahl und andere Steuerungsoperationen des Benutzers der Applikation. Gestartet wird die Applikation über die Start-Activity, welche dem Startbildschirm von TCA entspricht. Über diese Activity wird auch eine Verbindung des Mobile-Devices mit dem PC realisiert.

ch.hsr.tca.gui.Start

Die Klasse Start dient dem Start der Applikation. Möchte man eine Verbindung zwischen dem PC und dem Mobile-Device herstellen, geschieht dies ebenfalls in dieser Klasse. Die Verbindung mit dem PC muss zwingend in einem neuen Thread ablaufen, da ansonsten das GUI einfriert.

ch.hsr.tca.gui.ScreenLock

Da während einer bestehenden Verbindung mit dem PC auf dem Mobile-Device weder Tasks bearbeitet noch erstellt werden dürfen, wird der Bildschirm bis zum Beenden einer Verbindung (der Verbindungsabbruch wird auf dem PC initiiert) gesperrt. Sobald das Mobile-Device einen Verbindungsverlust detektiert, wird wieder auf den Startbildschirm zurückgekehrt. Dieser Vorgang funktioniert noch nicht einwandfrei, daher kann momentan der ScreenLock-Bildschirm jederzeit über das Drücken des «Backbuttons» verlassen werden.

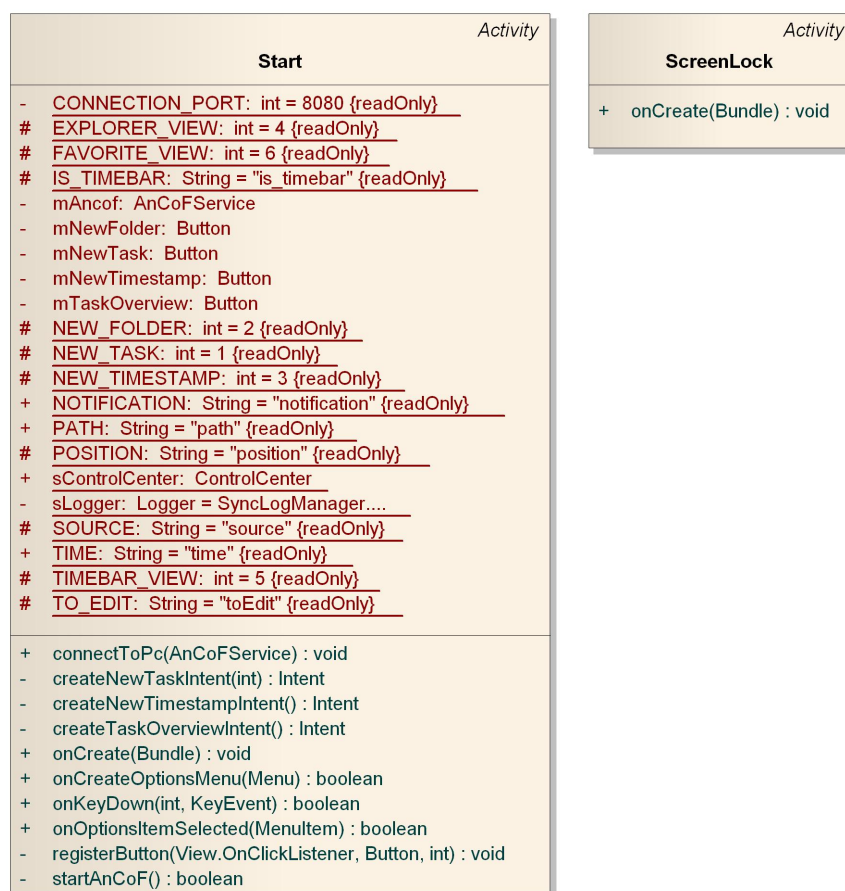


Abbildung 6.3: Package ch.hsr.tca.gui Startildschirm

ch.hsr.tca.gui.Overview

Diese Activity dient der Erstellung einer TabView für die Darstellung der Taskübersichten respektive der Favoriten. Um keinen doppelten Code zu erzeugen, wurde darauf verzichtet, für die Zeitstrahl- und die Baumdarstellung eigene Klassen zu verwenden. Beide Darstellungen werden über die TaskOverview-Activity realisiert, welche je nach angewähltem Tab die entsprechenden Felder lädt.

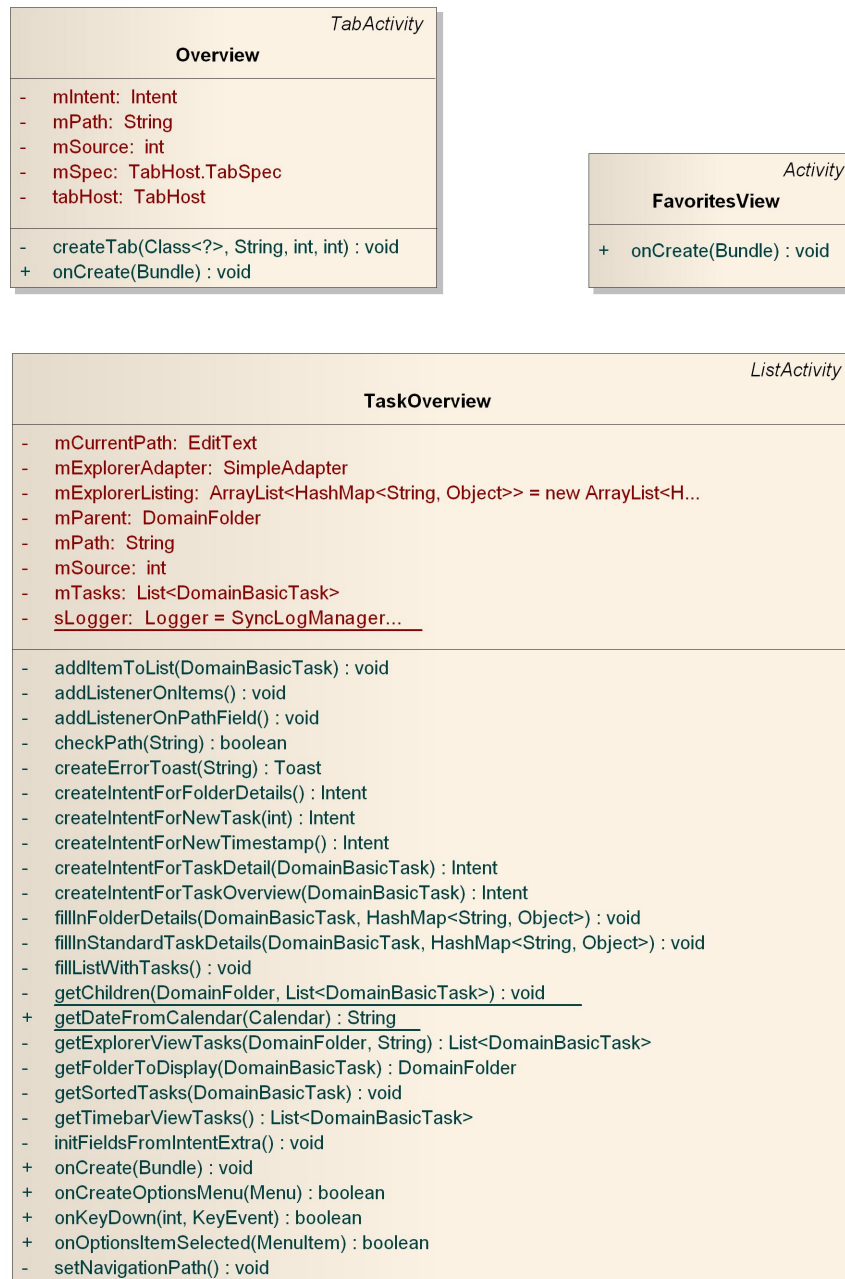


Abbildung 6.4: Package ch.hsr.tca.gui Taskübersicht

ch.hsr.tca.gui.TaskOverview

Die Klasse TaskOverview dient, wie bereits erläutert, der Visualisierung der Baum- und der Zeitstrahldarstellung. Über sogenannte «Extras», die einem Intent mitgegeben werden können, wird detektiert, ob es sich aktuell um eine Baum- oder eine Zeitstrahlview handelt. Ein «Intent» repräsentiert einen konkreten Aufruf einer anderen Activity, eines Hintergrundprozesses oder einer externen Anwendung.

ch.hsr.tca.gui.FavoritesView

Diese Klasse wurde noch nicht implementiert und dient zur Zeit nur als «Platzhalter»-Klasse.

ch.hsr.tca.gui.TaskDetail

Über die TaskDetail-Activity werden die Detail-Informationen eines bestimmten Tasks respektive Ordners angezeigt.

TaskDetail		Activity
-	mCurrentTask: DomainBasicTask	
-	mNotificationNr: int	
-	mPath: String	
-	mSource: int	
-	sLogger: <u>Logger = SyncLogManager...</u>	
-	createDialog(): void	
-	createTaskEditIntent(): Intent	
-	createTaskOverviewDeleteIntent(): Intent	
-	createTaskOverviewOkIntent(): Intent	
-	createTimestampEditIntent(): Intent	
-	deleteShowedTask(): void	
-	getDateFromCalendar(Calendar): String	
-	initFieldsFromIntentExtra(): void	
+	onCreate(Bundle): void	
+	onCreateOptionsMenu(Menu): boolean	
+	onOptionsItemSelected(MenuItem): boolean	
-	setCreationDateOfBasicTask(Textview): void	
-	setDateContent(): void	
-	setDueDateOfPriorityTask(Textview): void	
-	setDueOrEndDateContentOfPriorityTask(Textview): void	
-	setEffectiveTimeContent(): void	
-	setEndDateOfPriorityTask(Textview): void	
-	setEstimatedTimeContent(): void	
-	setLastEditContent(): void	
-	setNotes(): void	
-	setPriorityContent(): void	
-	setShortcuts(): void	
-	setTaskContent(): void	

Abbildung 6.5: Package ch.hsr.tca.gui Taskinformationen

ch.hsr.tca.gui.EditTask

Über die EditTask-Activity wird die Bearbeitung respektive Erstellung eines neuen Tasks oder Ordners realisiert. Wird auf «Speichern» gedrückt, wird ein neuer Task / Ordner erstellt und mithilfe der vom Control-Center zur Verfügung gestellten Methoden in der Datenbank gespeichert oder entsprechend aktualisiert. Das Bearbeiten oder Erstellen eines Tasks funktioniert analog dem Vorgang auf der PC-Seite, Details dazu können dem Kapitel 5.5.2, Seite 27 entnommen werden.

Je nach dem ob ein Ordner oder ein Task bearbeitet oder erstellt wird, werden die entsprechenden Felder aus- oder eingeblendet. Dies wird wiederum über ein «Extra» detektiert.

Activity

EditTask

```

-   completedDateSetListener: DatePickerDialog.OnDateSetListener = new DatePickerD...
-   completedTimeSetListener: TimePickerDialog.OnTimeSetListener = new TimePickerD...
-   dueDateSetListener: DatePickerDialog.OnDateSetListener = new DatePickerD...
-   dueTimeSetListener: TimePickerDialog.OnTimeSetListener = new TimePickerD...
-   EFFECTIVE: String = "effective" {readOnly}
-   ESTIMATED: String = "estimated" {readOnly}
-   mAlarm: CheckBox
-   mAlarmTime: EditText
-   mButtonPickCompletedDate: Button
-   mButtonPickCompletedTime: Button
-   mButtonPickDueDate: Button
-   mButtonPickDueTime: Button
-   mCompleted: CheckBox
-   mCompletedDateModified: boolean = false
-   mCurrentDay: int
-   mCurrentHour: int
-   mCurrentMinute: int
-   mCurrentMonth: int
-   mCurrentYear: int
-   mEditingBasicTask: DomainBasicTask
-   mEffective: EditText
-   mEstimated: EditText
-   mIsFolderEdit: boolean = false
-   mIsTimebar: boolean = false
-   mNote: TextView
-   mPath: String
-   mPriorityAdapter: ArrayAdapter<CharSequence>
-   mSavePath: EditText
-   mShortcuts: List<DomainShortcut> = new ArrayList<D...
-   mSource: int
-   mTimeUnit: Spinner
-   mTitle: EditText
-   mToEdit: boolean = false
-   sLogger: Logger = SyncLogManager...
-   TCA_PATH: String = "/Priorities" {readOnly}

-   addAlertTimeListener(): void
-   addCheckBoxListener(): void
-   analyzeSavePath(String): DomainFolder
-   calculateAlarmMinutes(Long): Long
-   checkSavePath(DomainFolder, String): DomainFolder
-   createAlertTimeSpinner(): Spinner
-   createCalendarWithCurrentDate(): Calendar
-   createCancelIntent(): Intent
-   createErrorToast(String): Toast
-   createIntent(String): Intent
-   createNewFolder(): DomainFolder
-   createNewStandardTask(): DomainStandardTask
-   createNotification(DomainStandardTask): void
-   createNumberErrorToast(): DomainStandardTask
-   createPrioritySpinner(): Spinner
-   disableTaskFields(): void
-   editingFinished(): void
-   externalStorageState(): boolean
-   getPathFromParent(DomainBasicTask): String
-   getSelectedDate(Calendar): Calendar
-   initButtons(): void
-   initFields(): void
-   initFieldsFromIntentExtra(): void
-   initGUIComponents(): void
-   loadFromFile(ArrayAdapter<CharSequence>): void
+  onCreate(Bundle): void
#  onCreateDialog(int): Dialog
-   pad(int): String
-   replaceComma(String): String
-   saveFolder(): DomainFolder
-   saveStandardTask(): DomainStandardTask
+  saveValuesToFolder(): DomainFolder
+  saveValuesToStandardTask(): DomainStandardTask
-   setAlertTime(DomainStandardTask): void
-   setAlertTimeContent(): void
-   setCalculatedAlertTime(DomainStandardTask): long
-   setCompletedDate(DomainStandardTask): void
-   setContentToEdit(): void
-   setDate(Calendar): void
-   setDisabled(View): void
-   setFolderContentToEdit(): void
-   setPreparedTime(DomainStandardTask, String, String): void
-   setSelectedPriority(DomainPriorityTask): void
-   setStandardTaskContentToEdit(): void
-   setTaskOverviewSource(Intent): void
-   titleExists(DomainFolder, String): boolean
-   titlesEmpty(String): boolean
-   unitFamily(DomainFolder, DomainBasicTask): void
-   updateDatesOnButtons(): void
-   updateDisplayedDate(Button): void
-   updateDisplayedTime(Button): void
                
```

«enumeration»
EditTask::DialogId

DUE_DATE_DIALOG
COMPLETED_DATE_DIALOG
DUE_TIME_DIALOG
COMPLETED_TIME_DIALOG

Attributes
~ mId: int

~ DialogId(int)
~ getId(): int

Abbildung 6.6: Package ch.hsr.tca.gui Task erstellen / bearbeiten

ch.hsr.tca.gui.TimestampView

Da der Timestamp sich grundlegend von der Bearbeitung oder Erstellung eines neuen Tasks oder Ordners unterscheidet (es wird nur ein Bruchteil der Informationen/Felder benötigt), wurde entschieden, eine eigene Klasse dafür zu realisieren. Die TimestampView-Activity ermöglicht die Bearbeitung oder Erstellung eines Timestamps. Dies ist ein spezieller Task, der primär dazu dient, «Zeitpunkte» festzuhalten. Die wichtigste Eigenschaft ist daher die Zeit der Erstellung dieses Tasks, welche im default-Titel festgehalten wird. Timestamps werden immer in der «root» gespeichert.

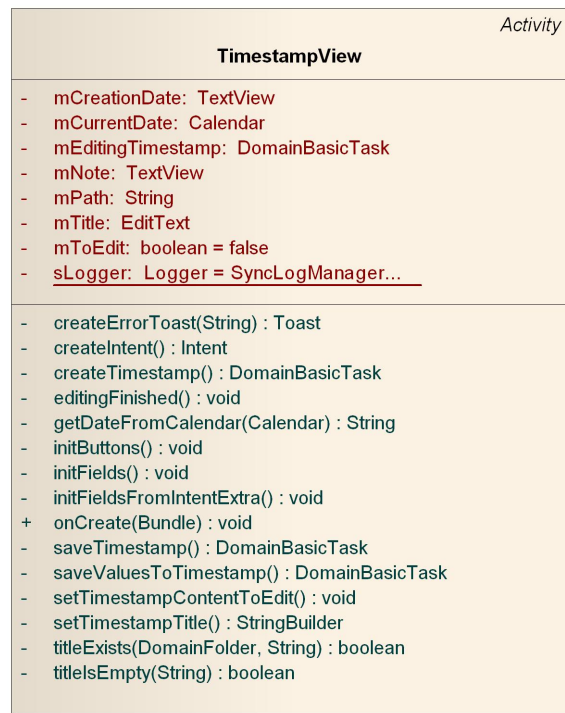


Abbildung 6.7: Package ch.hsr.tca.gui Zeitstempel erstellen

6.5 ch.hsr.tca.util

ch.hsr.tca.util.AbstractTaskComparator

Da je nach Darstellung (Baum oder Zeitstrahl) die Tasks respektive Ordner unterschiedlich sortiert sein müssen, wurde ein Comparator verwendet. Die benötigten Comparatoren (ExplorerComparator und TimebarComparator) implementieren den AbstractTaskComparator, welcher die gemeinsamen Methoden enthält.

ch.hsr.tca.util.ExplorerComparator

Der ExplorerComparator wird verwendet, um die Tasks in der Baumdarstellung entsprechend zu sortieren. Die Sortierung ist wie folgt: Timestamp, Folder, StandardTask. Innerhalb der Task-Klassen wird zuerst nach Priorität, dann nach Datum (je nach Task nach Erstell-, Fälligkeits- oder Enddatum) und zuletzt nach dem Tasktitel.

ch.hsr.tca.util.TimebarComparator

Der TimebarComparator dient dazu, die Tasks der Zeitstrahldarstellung in der richtigen Reihenfolge einzuordnen, welche nach Datum (je nach Task nach Erstell-, Fälligkeits- oder Enddatum) sortiert wird.

ch.hsr.tca.util.NotificationService

Über den NotificationService kann einem Task entsprechend der gesetzten Alarmzeit eine Erinnerung mitgegeben werden. Die Erinnerung wurde in einem Service implementiert, da eine Activity die Nachricht nur dann absetzen würde, wenn der Lebenszyklus der Activity noch nicht beendet wurde. Ein Service hingegen löst die Benachrichtigung auch dann aus, wenn die Applikation zum Beispiel geschlossen wurde. Wird jedoch das Programm neu gestartet und die Alarmzeit verändert oder der Alarm ausgeschaltet, werden diese Änderungen zur Zeit noch nicht detektiert und der ursprüngliche Alarm respektive die Notification trotzdem ausgelöst. Änderungen an der Alarmzeit werden momentan nur dann übernommen, wenn das Programm in der Zwischenzeit nie beendet wurde.

ch.hsr.tca.util.ServiceHelper

Der ServiceHelper implementiert das «ServiceConnection» Interface, welches es ermöglicht, Informationen zu erhalten, ob ein Service läuft oder gestoppt ist. Weitere Information betreffend diesem Interface können der Seite <http://developer.android.com/reference/android/content/ServiceConnection.html> entnommen werden.

ch.hsr.tca.util.ActivityRegistry

Die ActivityRegistry stellt Methoden zur Verfügung, die es erlauben, gestartete Activities zu registrieren, um diese, falls gewünscht, zu einem bestimmten Zeitpunkt wieder zu beenden.

ch.hsr.tca.util.PathNotExistsException

Diese Exception wird geworfen, falls ein Task aufgrund eines inkorrekten Pfades nicht gefunden werden kann.

ch.hsr.tca.util.UnknownUIDException

Diese Klasse wird genutzt, falls ein Task aufgrund einer falschen UID nicht gefunden werden kann.

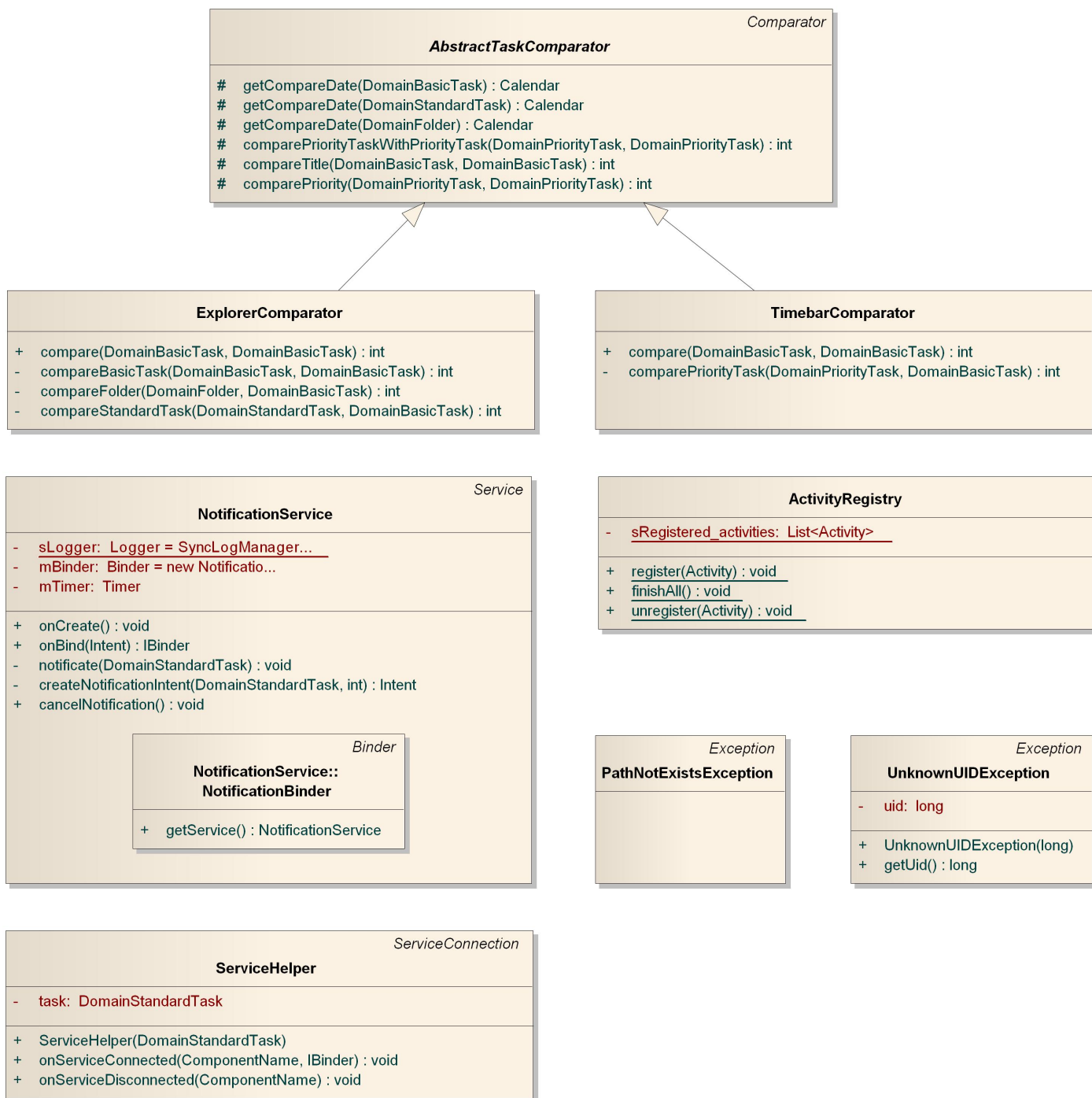


Abbildung 6.8: Package ch.hsr.tca.util

6.6 res: Oberflächengestaltung

Die Oberflächengestaltung wird in Android mittels XML-Dateien konfiguriert. Diese werden entsprechend ihrer Funktion in unterschiedlichen Ordnern abgelegt, welche sich im Hauptordner «res» befinden. Aus diesen XML-Dateien wird die Java-Klasse `R.java` generiert. In den entsprechenden View-Klassen (siehe Kapitel 6.4, Seite 43) kann nun über die generierte Klasse auf die entsprechenden View-Elemente, die in den XML-Dateien konfiguriert wurden, zugegriffen werden und deren Inhalt gelesen respektive gelöscht oder bearbeitet werden.

6.6.1 Bilder

Bilder befinden sich in den folgenden Subordnern:

- `res\drawable`
- `res\drawable-hdpi`
- `res\drawable-mdpi`
- `res\drawable-ldpi`

In den einzelnen Ordnern befinden sich die Bilder in der entsprechenden Grösse, damit aus der Anwendung heraus das optimale Bild angezogen werden kann. Wären die Bilder zu gross, würde das Laden zu lange dauern, wären sie zu klein, wäre die Auflösung nicht zufriedenstellend.

6.6.2 Layouts

Ähnlich wie bei Swing-Anwendungen werden verschiedene Layouts für den Aufbau der einzelnen Fenster verwendet. Die XML für diese Fenster befinden sich im Subordner `res/layout`.

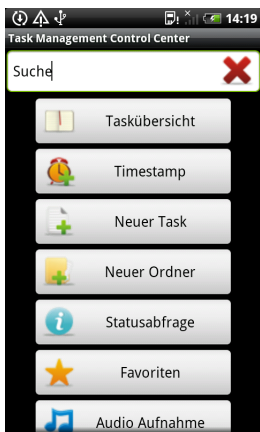


Abbildung 6.9: Startbildschirm



Abbildung 6.10: Erinnerung

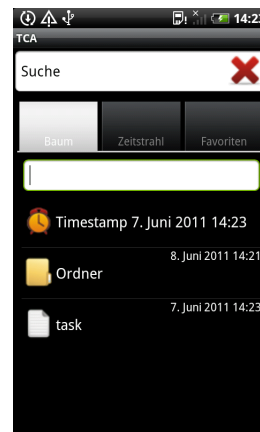


Abbildung 6.11: Baumdarstellung



Abbildung 6.12: Zeitstrahldarstellung



Abbildung 6.13: Detailinformationen

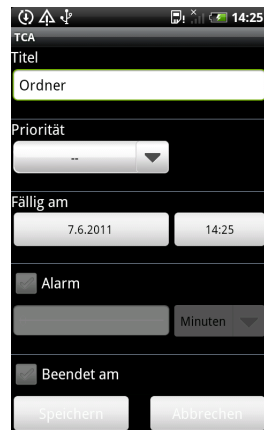


Abbildung 6.14: Ordner erstellen / bearbeiten

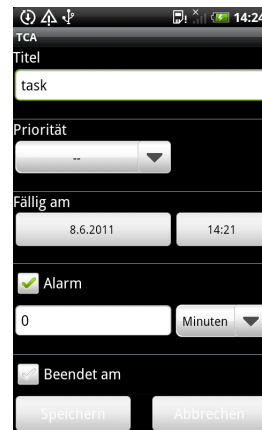


Abbildung 6.15: Task erstellen / bearbeiten

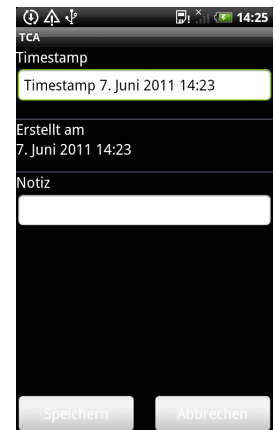


Abbildung 6.16: Zeitstempel erstellen / bearbeiten

Folgende Layouts sind vorhanden:

- res/layout.main.xml - für den Startbildschirm
- res/layout.screen_lock.xml - für die Bildschirmsperre
- res/layout.task_overview.xml - für die Erstellung der TabView der Taskübersicht
- res/layout.explorer_main.xml - für den Baum-, Zeitstrahl-Tab
- res/layout.explorer - für die Auflistung der Tasks im Baum-, Zeitstrahl-Tab
- res/layout.favorites.xml - für den Favoriten-Tab
- res/layout.task_detail.xml - für die Detailinformationen
- res/layout.edit_task.xml - für die Erstellung / Bearbeitung eines Tasks oder Ordners
- res/layout.timestamp.xml - für die Erstellung / Bearbeitung eines Zeitstempels

6.6.3 Menues

Einzelne Menues, welche in die unterschiedlichen Views eingebunden werden können, befinden sich im Subordner res/menu. Folgende Menues sind vorhanden:

- res/menu.main_menu.xml
 - «Verbindung» - um eine Verbindung mit dem PC herzustellen
- res/menu.explorer_menu.xml
 - «Neuer Task» - um einen Task zu erstellen
 - «Neuer Ordner» - um einen Ordner zu erstellen
 - «Ordneigenschaften» - um die Eigenschaften des Ordners, in dem man sich momentan befindet, anzuzeigen
 - «Timestamp» - um einen Zeitstempel zu erstellen
- res/menu.task_detail.xml

- «Bearbeiten» - um einen Ordner / Task / Zeitstempel zu bearbeiten
- «Löschen» - um einen Ordner / Task / Zeitstempel zu löschen
- «OK» - um zur Taskübersicht zurückzukehren

6.6.4 Sprachen

Die Internationalisierung der Applikation geschieht bei Android über die Subordner «value-*», wobei der * durch den entsprechenden Ländercode ersetzt wird. In diesen Ordnern werden die Strings definiert, welche in der Applikation für die Beschriftung von Eingabefeldern etc. eingesetzt werden. Entsprechend dem Ländercode müssen die Strings in die passende Sprache übersetzt werden.

- res/values.strings.xml
- res/values-en.strings.xml

Zur Zeit wird die Sprache der Applikation entsprechend der Betriebssystemsprache des Geräts gesetzt.

6.7 Android Manifest

Im Manifest werden die Activities und Services eingetragen und die entsprechenden Einstellungen und Rechte gesetzt, die für die Applikation benötigt werden. Dies ist nötig, damit die Laufzeitumgebung weiss, aus welchen Komponenten TCA zusammengesetzt ist.

Activities müssen entsprechend mit `<activity android:name="Package-Pfad"/>` und Services mit `<service android:name="Package-Pfad"/>` eingetragen werden. Falls bestimmte Rechte benötigt werden, so können diese mit `<uses-permission android:name="android.permission.BENÖTIGTE_PERMISSION"/>` angegeben werden.

6.8 Android Testprojekt

Um die Datenbankzugriffe zu testen, wurde eigens ein Testprojekt geschaffen, welches von ProviderTestCase2 erbt. Über dieses werden die CRUD-Operationen getestet.

Details können der Homepage von Android entnommen werden: <http://developer.android.com/reference/android/test/ProviderTestCase2.html>

7 Taskaustausch

Die grundlegenden Funktionen des Taskaustauschs werden von TaMaF übernommen, z.B. das Löschen eines Tasks sowohl auf dem Mobile-Device als auch auf dem PC.

Um die Eindeutigkeit der Pfade zu garantieren, müssen unter Umständen einige Anpassungen an den Tasks vorgenommen werden. Dabei agieren Mobile-Device und PC unabhängig und passen die Tasks in der eigenen Datenbank entsprechend an. Diese Unabhängigkeit ist nötig, da ansonsten Inkonsistenzen auf dem einen Device entstehen können.

Die folgende Übersicht stellt dar, wie die Tasks in den einzelnen Fällen angepasst werden müssen.

- Ein neuer Task wird empfangen:
 - **PC:** Der Pfad des neuen Task wird auf seine Eindeutigkeit überprüft. Falls dies nicht der Fall ist, wird der Titel des neuen Tasks durchnummeriert. Nun kann der Task als «neu» gespeichert werden.
 - **Mobile:** Der Pfad des neuen Task wird auf seine Eindeutigkeit überprüft. Falls dies nicht der Fall ist, wird der Titel des alten Tasks durchnummeriert. Es wird also nicht der gerade empfangene Task angepasst, sondern der ursprüngliche! Nun kann der Task als «neu» gespeichert werden.
- Ein veränderter Task wird empfangen:
 - **PC:** Der Pfad des veränderten Task wird auf seine Eindeutigkeit überprüft. Falls dies nicht der Fall ist, wird der Titel des veränderten Tasks durchnummeriert. Dabei wird zusätzlich überprüft, ob der Titel des veränderten Tasks dem Titel des original Tasks auf der PC-Seite entspricht. In diesem Fall muss der Titel nicht angepasst werden. Der veränderte Task kann nun gespeichert werden.
 - **Mobile:** Der Pfad des veränderten Task wird auf seine Eindeutigkeit überprüft. Falls dies nicht der Fall ist, wird der Titel des Mobile-Tasks durchnummeriert. Dabei wird ebenfalls überprüft, ob es sich um den gleichen Task handelt. Der veränderte Task kann nun gespeichert werden.
- Ein Löschbefehl wird erhalten: Dieser Fall birgt keine Probleme. Die Tasks können auf beiden Seiten normal gelöscht werden.

Die folgende Abbildung stellt exemplarisch dar, was passieren muss, wenn sowohl auf dem Mobile-Device als auch auf dem PC ein neuer Task gleichen Titels gespeichert wird. Dabei darf angenommen werden, dass die Verarbeitung der Tasks auf den beiden Devices parallel erfolgt und nicht sequentiell wie in der Abbildung gezeigt.

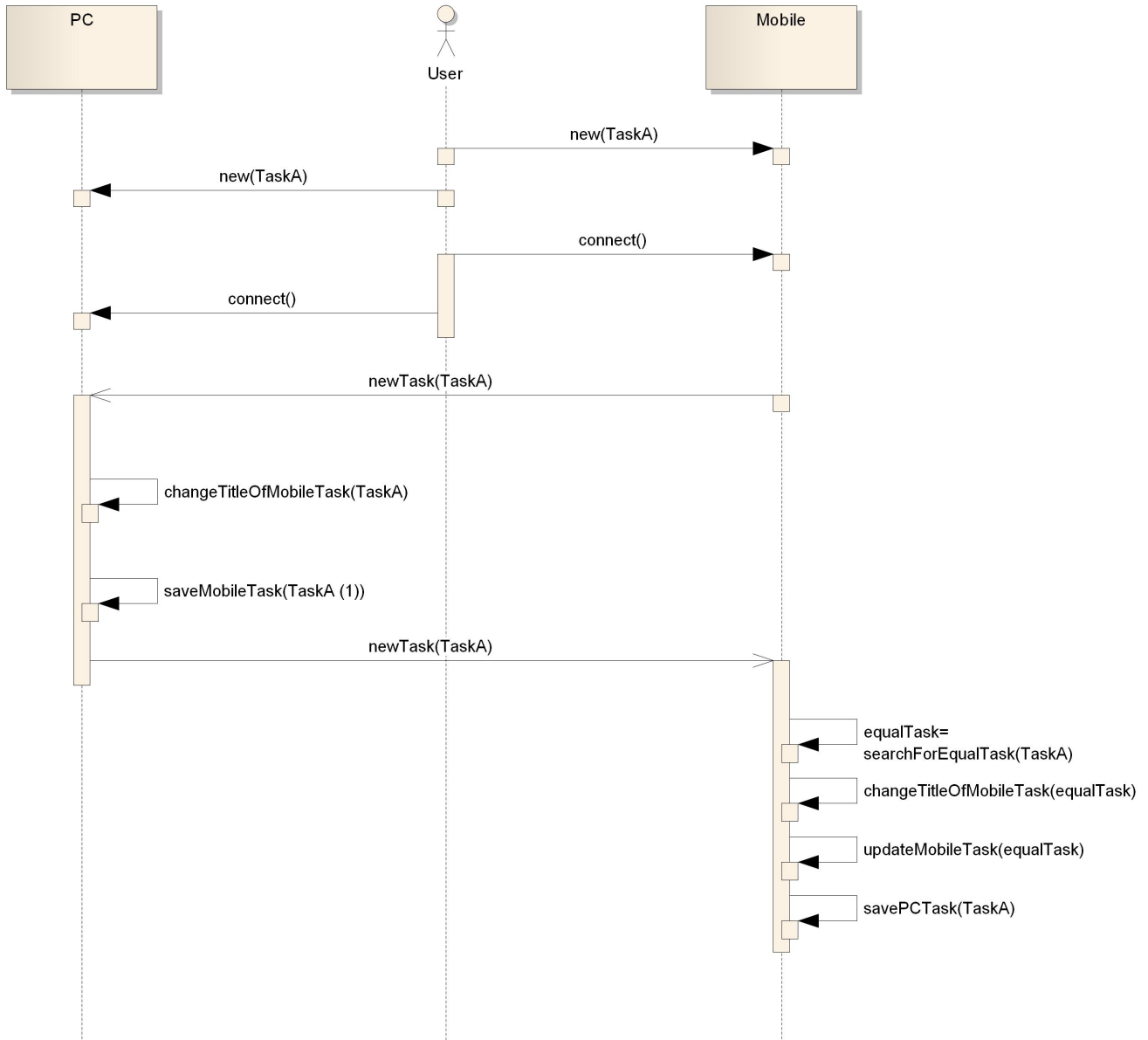


Abbildung 7.1: Beispiel eines Taskaustauschs

8 Erweiterungen und Einschränkungen

Zu den Erweiterungen und Einschränkungen sollten auch die «TODOs» im Code beachtet werden.

8.1 TCA

Das Durchsuchen der Tasks musste aus Zeitgründen ausgelassen werden. Details zu den Suchfunktionen können den Use Cases (UCs) («Anforderungsspezifikation (As)», Kapitel 2.3.2.4 - 2.3.2.6, Seite 5) entnommen werden.

8.2 AnCoF

Um die Synchronisation von Dateien zu Optimieren, wäre eine rekursive Synchronisation wünschenswert. Diese ist bereits in den von AnCoF zur Verfügung gestellten Schnittstellen vorgesehen, jedoch noch nicht implementiert.

8.3 Datenbank

Die Datenbank sollte soweit implementiert sein, dass keine grossen Änderungen oder Einschränkungen zu erwarten sind. Es sind nur noch kleine Anpassungen zu erwarten, falls neue «Features» hinzukommen.

8.4 TCA PC

8.4.1 Noch nicht implementierte Bereiche

Folgende Bereiche konnten aus Zeitgründen nicht implementiert werden:

- Statusbar
- Suchformular
- Aktives Verlinken der Shortcuts durch HTML-Hyperlinks.

Details können dem Kapitel 5 ab Seite 7 entnommen werden.

8.4.2 Pfad-Eingabefeld

Im Moment können im Pfad-Eingabefeld nur Ordnerpfade eingegeben werden. In Zukunft wäre es möglich den Pfad eines «normalen» Tasks einzugeben, um so die Detail-Ansicht des Tasks zu erhalten.

8.4.3 Alarm auslösen

Zum gespeicherten Alarmzeitpunkt wird noch kein Signal ausgelöst. Dies muss, ähnlich dem NotificationService auf dem Mobile-Device implementiert werden.

8.4.4 Erkennen von verändertem Beendet-Datum

Es besteht noch keine Möglichkeit, dass der Benutzer erkennen kann, ob das Beendet-Datum verändert wurde. Dies könnte gelöst werden, indem das automatisch generierte Datum präziser angegeben wird, als das manuell eingegebene.

8.5 TCA Android

8.5.1 ch.hsr.tca.domain

ch.hsr.tca.domain.TaskWrapper

Um alle Subtasks eines bestimmten Ordners zu erhalten, werden momentan die einzelnen Einträge aus den Tabellen `basic_task`, `folder` und `standard_task` in eigenen Abfragen ausgelesen, um diese dann einzeln auszusortieren, damit keine doppelten Einträge aufgrund der Vererbungshierarchie mehr vorhanden sind.

Dies sollte, falls Android es zulässt, mittels eines «Joins» bereits in der Datenbankabfrage optimiert werden, wie dies bereits auf der PC-Seite der Fall ist.

ch.hsr.tca.domain.XferAgentTaMaFImpl

Zur Zeit wird der Verlust respektive Verbindungsabbruch seitens des PCs auf der Mobile-Seite nicht zuverlässig detektiert. Die Override-Methode `public void onConnectionLost(int HostID)` in der Klasse `XferAgentTaMaFImpl`, welche entsprechende Schritte bei einem Verbindungsverlust einleitet, wird daher nicht aufgerufen und dementsprechend werden auch keine weiterführenden Aktionen ausgelöst, welche für einen reibungslosen Ablauf des Programms erforderlich wären. Details zum «Workaround» können dem Abschnitt `ch.hsr.tca.gui.ScreenLock` entnommen werden.

8.5.2 ch.hsr.tca.gui

Im GUI wurden folgende vorgesehene Bereiche noch nicht implementiert:

- Statusabfrage
- Favoriten
- Audio Aufnahme
- Shortcuts erstellen oder bearbeiten

ch.hsr.tca.gui.ScreenLock

Wird der Verbindungsabbruch wie im obigen Kapitel beschrieben nicht detektiert, so muss nach dem Beenden einer Verbindung mittels Drücken des «Backbuttons» auf den Startbildschirm zurückgekehrt werden.

Während einer bestehenden Verbindung können Tasks erstellt oder bearbeitet werden. Dies sollte jedoch nicht gemacht werden, da es nicht vorgesehen ist und nur durch den Umstand, dass der Sperrbildschirm momentan nicht vollständig eine Bedienung des Gerätes verhindern darf, ermöglicht wird.

In Zukunft sollte dies nicht mehr möglich sein. Sobald die Detektion des Verbindungsabbruchs zuverlässig funktioniert, erlaubt der Sperrbildschirm während einer Verbindung keine Bedienung des Programms mehr.

ch.hsr.tca.gui.Overview

Momentan wird bei der Darstellung eines Tasks nicht zwischen manuell geändertem und den automatisch vom System generiertem Datum unterschieden.

Um diesen Unterschied sichtbar zu machen, könnten manuell geänderte Daten nur auf die Minute genau angezeigt werden und jene, welche automatisch generiert wurden, auf die Sekunde genau.

ch.hsr.tca.gui.FavoritesView

Dieser Bildschirm ist ein Platzhalter, welcher durch die zukünftige Implementierung der Favoriten ersetzt wird.

ch.hsr.tca.gui.TaskDetail

Shortcuts, also Links auf andere Tasks, werden momentan nur dargestellt, sind aber noch nicht verlinkt bzw. erlauben durch Drücken des Links kein «Springen» auf den entsprechenden Task. Auch können sie weder bearbeitet noch erstellt werden. Einzig das Löschen funktioniert, da dies automatisch geschieht, wenn die verlinkten Tasks gelöscht werden.

8.5.3 ch.hsr.tca.util

ch.hsr.tca.util.NotificationService

Einem Task kann aktuell eine Erinnerung gesetzt werden, diese wird auch zur richtigen Zeit ausgelöst. Wird jedoch in der Zwischenzeit die Applikation geschlossen und danach wieder geöffnet und die Erinnerung bearbeitet oder gelöscht, hat dies keine Auswirkungen auf die zuvor gesetzte Erinnerung. Es wird zwar eine korrekte, neue Erinnerung beim Task gespeichert, die alte kann jedoch nicht gelöscht werden, da diese in einem eigenen Thread abläuft, welcher im Hintergrund weiterläuft.

In Zukunft sollte dies behoben werden. Ein möglicher Ansatz wäre, mittels Thread-Id den Thread dem Task zuzuordnen, damit zu einem späteren Zeitpunkt darauf zugegriffen werden kann und die Änderungen entsprechend eingespielen werden können.

8.6 Sprachen

Momentan kann die Sprache der Applikation nicht manuell gesetzt werden und wird entsprechend der Betriebssystemsprache gesetzt.

8.7 Taskaustausch

Der andauernde Vorgang des Dateiaustausches wird in der PC-Applikation nur unzureichend dargestellt. Dies sollte geändert werden, um dem Nutzer durch einen Dialog bzw. durch Anzeigen in der Statusbar eine visuelle Bestätigung des Vorgangs zu geben.

Um Tasks austauschen zu können, muss zuvor eine erste Verbindung aufgebaut worden sein. Dies ist nicht benutzerfreundlich und sollte behoben werden.

8.8 Verwendete Libraries und Bilder

8.8.1 Libraries

Die Lizenz des verwendeten DatePicker kann im Anhang (Kapitel 9, Seite 9) gefunden werden. Für eine spätere Kommerzialisierung des TCA sollte sie nicht im Wege stehen.

8.8.2 Bilder

Es wurden Bilder unter dem Paket «coquette»¹ verwendet. Zudem wurden einige Bilder aus zwei bestehenden zusammengefügt. Dieses Icon-Set darf in einer kommerziellen Version des TCA nicht verwendet werden, da es unter der GNU-General-Public-License steht.

¹<http://dryicons.com/free-icons/preview/coquette-icons-set/>

9 Anhang

JCalendar License

Artistic License

Preamble

The intent of this document is to state the conditions under which a Package may be copied, such that the Copyright Holder maintains some semblance of artistic control over the development of the package, while giving the users of the package the right to use and distribute the Package in a more-or-less customary fashion, plus the right to make reasonable modifications.

Definitions:

- "Package" refers to the collection of files distributed by the Copyright Holder, and derivatives of that collection of files created through textual modification.
 - "Standard Version" refers to such a Package if it has not been modified, or has been modified in accordance with the wishes of the Copyright Holder.
 - "Copyright Holder" is whoever is named in the copyright or copyrights for the package.
 - "You" is you, if you're thinking about copying or distributing this Package.
 - "Reasonable copying fees" is whatever you can justify on the basis of media cost, duplication charges, time of people involved, and so on. (You will not be required to justify it to the Copyright Holder, but only to the computing community at large as a market that must bear the fee.)
 - "Freely Available" means that no fee is charged for the item itself, though there may be fees involved in handling the item. It also means that recipients of the item may redistribute it under the same conditions they received it.
1. You may make and give away verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you duplicate all of the original copyright notices and associated disclaimers.
 2. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.
 3. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:
 - a) place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

- b) use the modified Package only within your corporation or organization.
 - c) rename any non-standard executables so the names do not conflict with standard executables, which must also be provided, and provide a separate manual page for each non-standard executable that clearly documents how it differs from the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
4. You may distribute the programs of this Package in object code or executable form, provided that you do at least ONE of the following:
- a) distribute a Standard Version of the executables and library files, together with instructions (in the manual page or equivalent) on where to get the Standard Version.
 - b) accompany the distribution with the machine-readable source of the Package with your modifications.
 - c) accompany any non-standard executables with their corresponding Standard Version executables, giving the non-standard executables non-standard names, and clearly documenting the differences in manual pages (or equivalent), together with instructions on where to get the Standard Version.
 - d) make other distribution arrangements with the Copyright Holder.
5. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you do not advertise this Package as a product of your own.
6. The scripts and library files supplied as input to or produced as output from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.
7. C or perl subroutines supplied by you and linked into this Package shall not be considered part of this Package.
8. Aggregation of this Package with a commercial distribution is always permitted provided that the use of this Package is embedded; that is, when no overt attempt is made to make this Package's interfaces visible to the end user of the commercial distribution. Such use shall not be construed as a distribution of this Package.
9. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.
10. THIS PACKAGE IS PROVIDED AS IS AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The End

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	20.05.2011	DM	Dokument erstellt.
1.0rc02	08.06.2011	DM	Kapitel für Mobile erstellt.
1.0rc03	08.06.2011	RR	Kapitel für PC erstellt.
1.0rc04	09.06.2011	RR	Taskaustausch beschrieben.
1.0rc05	11.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.0	11.06.2011	RR	Akzeptierte Version 1.0.

Tabelle 9.1: Revisionshistorie

Abkürzungsverzeichnis

AnCoF Android Control Framework

As Anforderungsspezifikation

Ba Bachelorarbeit

CRUD create, read, update, delete

GUI Graphical User Interface

HTML Hypertext Markup Language

PC Personal Computer

TCA Task-Management Control-Center for Android

TaMaF Task-Management-Framework on Smart-Phone

UC Use Case

UID Unique Identification

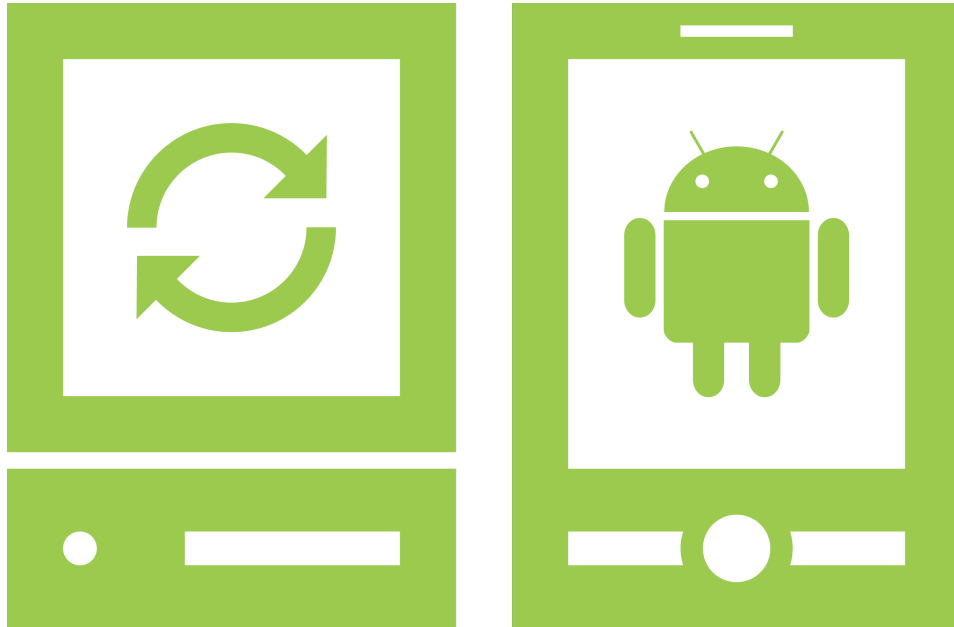
Abbildungsverzeichnis

3.1	Die neue Taskstruktur	4
4.1	Datenbankschema	5
5.1	Klassenstruktur	8
5.2	Package ch.hsr.tca.application	9
5.3	Struktur der Controller in TCA	10
5.4	Package ch.hsr.tca.domain Teil 1	11
5.5	Package ch.hsr.tca.domain Teil 2	13
5.6	Package ch.hsr.tca.gui	15
5.7	Package ch.hsr.tca.gui.views Teil 1	16
5.8	Die Klasse MainPanel im GUI	17
5.9	Die Klasse Toolbar im GUI	17
5.10	Die Klasse MainViewPanel im GUI (rot) aufgeteilt in die Bereiche Suchformular (dunkelblau), Pfad-Eingabefeld (gelb), Tasktabelle (grün) und Bearbeitungsbereich (hellblau)	18
5.11	Die Klasse Statusbar im GUI	19
5.12	Package ch.hsr.tca.gui.views Teil 2	20
5.13	Die Klasse EditTaskPanel im GUI	21
5.14	Package ch.hsr.tca.gui.views Teil 3	22
5.15	Die Klasse DetailViewFrame als GUI-Element	23
5.16	Die Klasse ShortcutDialog als GUI-Element	24
5.17	Die Controller- und ihre View-Klassen	25
5.18	Package ch.hsr.tca.gui.controller	26
5.19	Package ch.hsr.tca.gui.validation	31
5.20	Beispiel einer fehlgeschlagenen Validierung	32
5.21	Package ch.hsr.tca.gui.util	33
5.22	Package ch.hsr.tca.util	35
6.1	Klassenstruktur Mobile	39
6.2	Package ch.hsr.tca.domain.tasks	41
6.3	Package ch.hsr.tca.gui Startildschirm	44
6.4	Package ch.hsr.tca.gui Taskübersicht	45
6.5	Package ch.hsr.tca.gui Taskinformationen	46
6.6	Package ch.hsr.tca.gui Task erstellen / bearbeiten	47
6.7	Package ch.hsr.tca.gui Zeitstempel erstellen	48
6.8	Package ch.hsr.tca.util	50
6.9	Startbildschirm	51
6.10	Erinnerung	51
6.11	Baumdarstellung	51
6.12	Zeitstrahldarstellung	51
6.13	Detailinformationen	52
6.14	Ordner erstellen / bearbeiten	52
6.15	Task erstellen / bearbeiten	52
6.16	Zeitstempel erstellen / bearbeiten	52

7.1 Beispiel eines Taskaustauschs 55

Tabellenverzeichnis

5.1	Abgefangener Event des MainPanelController und dessen Aktion	27
5.2	Abgefangene Events des MainViewPanelController und ihre Aktionen	28
5.3	Abgefangene Events des MenuControllers und ihre Aktionen	29
5.4	Observer-Struktur und ausgetauschte Events	36
5.5	Tool- und Menubar Buttons und ihre Events	38
9.1	Revisionshistorie	62



Testdokumentation

«Task-Management Control-Center for Android»

Version 1.1, 11. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Systemtest	2
2.1 Voraussetzung	2
2.2 Vorbereitung	2
2.2.1 Setzen der Einstellungen	2
2.2.2 Starten der Anwendungen	2
2.2.3 Konfigurationsdatei für den Dateiaustausch	3
2.3 Ablauf	3
2.4 Systemtest	4
2.4.1 create, read, update, delete (CRUD)-Operationen	4
2.4.2 Dateisynchronisation	6
2.4.3 Tasksynchronisation	8
2.4.4 Tasks	9
2.4.5 Verbindung	12
2.4.6 Konfiguration	13
3 Unit Test	14

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 8.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Systemtest

2.1 Voraussetzung

Der Benutzer verfügt über ein Mobile-Device mit dem Betriebssystem Android der minimalen Version 2.1. Zudem wird ein PC mit einer Java Installation der minimalen Version 1.6 und einer Installation der Android Debug Brindge (ADB)¹ benötigt.

Um eine Verbindung zwischen dem Mobile-Device und dem PC aufzubauen, wird ein USB-Kabel benötigt.

2.2 Vorbereitung

Es wird davon ausgegangen, dass die Dateien in dem Verzeichnis liegen bleiben, in welchem sie ausgeliefert wurden. Dieses Verzeichnis wird im weiteren «Hauptverzeichnis» genannt. Auf dem Mobile-Device muss relativ zum Pfad «sdcard» ein Ordner «Priorities» erstellt werden. Hier wird jeweils die aktuellste Prioritäten-Datei vom PC abgelegt.

2.2.1 Setzen der Einstellungen

Im Hauptverzeichnis befindet sich ein Ordner «properties», in welchem sich die Datei «constants.properties» befindet. In dieser Datei muss unter

- «MainFolder» der Pfad des Speicherordners (muss nicht im Hauptverzeichnis sein) angegeben werden.
- «PriorityFileLocation» der Pfad, relativ zum «MainFolder» und dem Namen der Datei, in der die Prioritäten gespeichert werden, angegeben werden.
- «DatabaseLocation» der Pfad inklusive dem Namen der Datenbank, in welcher die Tasks gespeichert werden, relativ zum «MainFolder», angegeben werden.

2.2.2 Starten der Anwendungen

Bevor getestet werden kann, muss die Datei «TCA_Android.apk» auf dem Mobile-Device installiert werden. Um eine Verbindung zwischen dem Mobile-Device und dem PC herzustellen, muss das USB-Kabel beim PC und beim Mobile-Device eingesteckt werden.

Die Installation wird folgendermassen vorgenommen: Auf der Konsole ins Hauptverzeichnis wechseln und den Befehl `adb install AnCoF.apk` eingeben. Wurde die Installation erfolgreich abgeschlossen, kann die Applikation auf dem Mobile-Device wie gewohnt vom Menu aus gestartet werden.

Weiter muss die PC-Applikation «TCA_PC.jar» gestartet werden. Dies geschieht unter Windows mittels Doppelklick auf die Datei oder unter Linux wie auch Windows über den Konsolenbefehl: `java -jar TCA_PC.jar`.

Vor dem Testen muss eine erste Verbindung zwischen Mobile-Device und PC aufgebaut werden.

¹<http://developer.android.com/guide/developing/tools/adb.html>

2.2.3 Konfigurationsdatei für den Dateiaustausch

In der Datei «properties/path.properties» können die Ordner angegeben werden, deren Inhalt ausgetauscht werden soll.

2.3 Ablauf

Alle im Dokument Anforderungsspezifikation (As) beschriebenen Use Cases (UCs) werden getestet und die Resultate bzw. Abläufe festgehalten.

2.4 Systemtest

2.4.1 CRUD-Operationen

Die folgenden Testfälle wurden nicht mit UCs abgedeckt, da sie für eine erfolgreiche Synchronisation der Tasks vorausgesetzt werden.

Name	Task erstellen
Beschreibung	Ein Task wird erstellt und in der Datenbank gespeichert.
Erwartetes Resultat	Der Task wurde erstellt.
Voraussetzung	Es existiert kein Ordner oder Task mit dem gleichen Titel und Pfad.
Testdaten	–
Vorgehen	PC: Im Menü «Bearbeiten» auf «Neuer Task» klicken. Nachdem die Daten eingegeben wurden, auf «Speichern» klicken. Mobile: Auf dem Startscreen auf «Neuer Task» klicken. Nachdem die Daten eingegeben wurden, auf «Speichern» klicken.
Resultat	Der Task wurde auf den entsprechenden Devices / Betriebssystemen erstellt.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.1: Systemtest Task erstellen

Name	Ordner erstellen
Beschreibung	Ein Ordner wird erstellt und in der Datenbank gespeichert.
Erwartetes Resultat	Der Ordner wurde erstellt.
Voraussetzung	Es existiert kein Ordner oder Task mit dem gleichen Titel und Pfad.
Testdaten	–
Vorgehen	PC: Im Menü «Bearbeiten» auf «Neuer Ordner» klicken. Nachdem die Daten eingegeben wurden, auf «Speichern» klicken. Mobile: Auf dem Startscreen auf «Neuer Ordner» klicken. Nachdem die Daten eingegeben wurden, auf «Speichern» klicken.
Resultat	Der Ordner wurde auf den entsprechenden Devices / Betriebssystemen erstellt.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.2: Systemtest Ordner erstellen

Name	Verschachtelter Task / Ordner erstellen
Beschreibung	Ein verschachtelter Task / Ordner wird erstellt und in der Datenbank gespeichert.
Erwartetes Resultat	Der Task / Ordner wurde erstellt.
Voraussetzung	Es existiert kein Ordner oder Task mit dem gleichen Titel und Pfad. Es existiert ein Ordner, in welchem der Task / Ordner gespeichert werden kann.
Testdaten	–
Vorgehen	PC: Im Menü «Bearbeiten» auf «Neuer Task» bzw. «Neuer Ordner» klicken. Nachdem die Daten eingegeben wurden und der Pfad entsprechend gesetzt wurde, auf «Speichern» klicken. Mobile: Auf dem Startscreen auf «Neuer Task» bzw. «Neuer Ordner» klicken. Nachdem die Daten eingegeben wurden und der Pfad entsprechend gesetzt wurde, auf «Speichern» klicken.
Resultat	Der Task / Ordner wurde auf den entsprechenden Devices / Betriebssystemen erstellt.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.3: Systemtest Verschachtelter Task / Ordner erstellen

Name	Task / Ordner ändern
Beschreibung	Ein Task / Ordner wird verändert und in der Datenbank aktualisiert.
Erwartetes Resultat	Der Task / Ordner wurde aktualisiert.
Voraussetzung	Es existiert kein Ordner oder Task mit dem gleichen Titel und Pfad. Es existiert ein Task / Ordner, welcher verändert werden kann.
Testdaten	–
Vorgehen	PC: Task / Ordner in Tabelle selektieren und im Bereich «Bearbeiten» die Änderungen vornehmen. Nachdem die Daten verändert wurden, auf «Speichern» klicken. Mobile: Auf dem Startscreen auf «Taskübersicht» klicken und dort ein Task auswählen. Danach im Menü auf «Task bearbeiten» klicken. Falls ein Ordner verändert werden soll, den Ordner öffnen und im Menü «Ordneigenschaften» und danach im Menü «Task bearbeiten» wählen. Die Daten verändern und auf «Speichern» klicken.
Resultat	Der Task / Ordner wurde auf den entsprechenden Devices / Betriebssystemen aktualisiert.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.4: Systemtest Task / Ordner ändern

Name	Task / Ordner löschen
Beschreibung	Ein Task / Ordner wird gelöscht und aus der Datenbank entfernt.
Erwartetes Resultat	Der Task / Ordner wurde gelöscht.
Voraussetzung	Es existiert ein Task / Ordner, welcher gelöscht werden kann.
Testdaten	–
Vorgehen	PC: Task / Ordner in Tabelle selektieren und im Menü «Bearbeiten» auf «Löschen» klicken. Mobile: Auf dem Startscreen auf «Taskübersicht» klicken und dort ein Task anwählen. Falls ein Ordner gelöscht werden soll, den Ordner öffnen und im Menü «Ordneigenschaften» wählen. Danach im Menü auf «Löschen» klicken.
Resultat	Der Task / Ordner wurde auf den entsprechenden Devices / Betriebssystemen gelöscht.
Kommentar	–
Testdatum	10.05.2011

Tabelle 2.5: Systemtest Task / Ordner löschen

2.4.2 Dateisynchronisation

Name	UC 1.1: Homeverzeichnis festlegen
Beschreibung	Das Homeverzeichnis, in welchem alle ausgetauschten Dateien abgelegt werden, wird festgelegt.
Erwartetes Resultat	Der Pfad des Homeverzeichnisses steht in der entsprechenden Datei
Voraussetzung	–
Testdaten	Ein existierender Ordner, welcher als Homeverzeichnis dient.
Vorgehen	Die entsprechende Konfigurationsdatei öffnen und die Zeile mit dem Homeverzeichnis hinzufügen.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.6: Systemtest UC 1.1

Name	UC 1.2: Zu synchronisierende Ordner festlegen
Beschreibung	Die Ordner, welche synchronisiert werden sollen, werden festgelegt. Dies kann geschehen, indem ein absoluter Pfad angegeben wird, oder ein Pfad, relativ zum Homeverzeichnis.
Erwartetes Resultat	Der Pfad der zu synchronisierenden Ordner steht in der entsprechenden Datei.
Voraussetzung	Falls ein relativer Pfad angegeben wird, muss ein Homeverzeichnis angegeben sein.
Testdaten	Ein existierender Ordner, in dem die Dateien liegen.
Vorgehen	Die entsprechende Konfigurationsdatei öffnen und den Befehl hinzufügen.
Resultat	Die Ordner sind in der Konfigurationsdatei gespeichert.
Kommentar	Es kann kein Pfad relativ zum Homeverzeichnis angegeben werden.
Testdatum	31.05.2011

Tabelle 2.7: Systemtest UC 1.2

Name	UC 1.3: Ordnerinhalte synchronisieren
Beschreibung	Die Dateien, welche sich in einem Ordner befinden, werden synchronisiert.
Erwartetes Resultat	Die Dateien wurden entsprechend ihrem Datum ausgetauscht.
Voraussetzung	Es wurde mindestens ein zu synchronisierender Ordner festgelegt. In diesem Ordner befindet sich mindestens eine Datei.
Testdaten	Eine beliebige Datei.
Vorgehen	In der Mobile-Applikation im Menü auf «Verbindung» klicken. Danach in der PC-Applikation, im Menü «Verbindung» auf «Verbindung herstellen» klicken.
Resultat	Die Ordnerinhalte haben den gleichen Stand.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.8: Systemtest UC 1.3

Name	UC 1.4: Ordner von der Synchronisation ausschliessen
Beschreibung	Ein Ordner wird temporär vom Synchronisationsvorgang ausgeschlossen.
Erwartetes Resultat	Die Dateien im angegebenen Ordner wurden nicht synchronisiert und haben noch den selben Stand wie vor der Synchronisation.
Voraussetzung	Es wurde mindestens ein zu synchronisierender Ordner festgelegt. In diesem Ordner befindet sich mindestens eine Datei.
Testdaten	Eine beliebige Datei.
Vorgehen	Die entsprechende Konfigurationsdatei öffnen und die Zeile, in welcher der Ordner definiert ist, auskommentieren.
Resultat	Die Ordnerinhalte wurden nicht synchronisiert.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.9: Systemtest UC 1.4

2.4.3 Tasksynchronisation

PC und Mobile in der folgenden Tabelle entsprechen den Geräten, auf welchen die Task-Items erstellt, bearbeitet oder gelöscht werden. Wenn nach einer Operation der Task auf dem jeweils anderen Gerät erfolgreich aktualisiert wurde, kann dies in der Tabelle mittels eines «✓» eingetragen werden.

Name	Root-Tasks		Verschachtelung		Gleicher Titel	
	PC	Mobile	PC	Mobile	PC	Mobile
New Task	✓	✓	✓	✓	✓	✓
New Folder	✓	✓	✓	✓	✓	✓
New Timestamp	✓	✓	–	–	✓	✓
Update Task	✓	✓	✓	✓	✓	✓
Update Folder	✓	✓	✓	✓	✓	✓
Update Timestamp	✓	✓	–	–	✓	✓
Delete Task	✓	✓	✓	✓	–	–
Delete Folder	✓	✓	✓	✓	–	–
Delete Timestamp	✓	✓	✓	✓	–	–
Vorgehen	Entsprechend Test gewünschtes Item erstellen, bearbeiten oder löschen. Für Details zum Vorgehen siehe Kapitel 2.4.1, Seite 4. Nachdem die Items vorbereitet sind, muss die Synchronisation ausgelöst werden. Details hierfür können dem Kapitel 2.4.5, Seite 12 entnommen werden.					
Resultat	Auf beiden Geräten herrscht ein synchroner Zustand betreffend der Task-Items.					
Kommentar	Timestamps können nicht verschachtelt werden, diese Tests fallen daher weg. Das Löschen von Tasks respektive Folder oder Timestamps mit gleichem Titel stellt kein Problem dar und muss daher nicht getestet werden.					
Testdatum	31.05.2011					

Tabelle 2.10: Systemtests Tasksynchronisation

2.4.4 Tasks

Name	UC 2.1: Anzeigen der Tasks
Beschreibung	Die Tasks werden in einer der beiden Darstellungsformen (Baum oder Zeitstrahl) angezeigt. Dies ist sowohl auf dem Mobile-Device als auch auf dem PC möglich.
Erwartetes Resultat	PC: Die Tasks werden in der Übersichtstabelle im Hauptfenster angezeigt. Mobile: Die Tasks werden in der Tabview angezeigt.
Voraussetzung	Es wurde zuvor mindestens ein Task gespeichert.
Testdaten	Die zuvor gespeicherten Tasks.
Vorgehen	PC: Die Applikation starten. Mobile: Den Button «Taskübersicht» anklicken.
Resultat	Die Tasks werden in der Baumübersicht angezeigt.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.11: Systemtest UC 2.1

Name	UC 2.2: Wechseln der Visualisierung
Beschreibung	Die Tasks werden in der Baum- oder in der Zeitstrahldarstellung angezeigt.
Erwartetes Resultat	Die Visualisierung wurde von der Baum- in die Zeitstrahldarstellung umgewandelt, bzw. von der Zeitstrahl- in die Baumdarstellung.
Voraussetzung	Es wurden zuvor mehrere Tasks gespeichert, welche unterschiedliche Fälligkeitsdaten und Verschachtlungen haben.
Testdaten	Die zuvor gespeicherten Tasks.
Vorgehen	PC: Im Menü «Ansicht» auf «Ansicht wechseln» klicken. Mobile: Den entsprechenden Tab der Tabview anklicken.
Resultat	Die Ansicht wurde gewechselt.
Kommentar	Auf dem Mobile-Device kann in der Zeitstrahlansicht kein spezifischer Task selektiert werden, welcher, bei einem Wechsel in die Baumdarstellung, entsprechend angezeigt wird.
Testdatum	31.05.2011

Tabelle 2.12: Systemtest UC 2.2

Name	UC 2.3: Task als Favorit speichern
Beschreibung	Ein Task wird als Favorit markiert und wird fortan zusätzlich in einer weiteren Liste angezeigt.
Erwartetes Resultat	PC: Der Task erscheint im Menü «Favoriten». Mobile: Der Task wird in der «Favoriten-Liste» angezeigt.
Voraussetzung	Es wurde zuvor mindestens ein Task gespeichert.
Testdaten	Ein zuvor gespeicherter Task.
Vorgehen	PC: Im Menü «Favoriten» auf «Zu Favoriten hinzufügen» klicken. Mobile: Noch nicht implementiert.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.13: Systemtest UC 2.3

Name	UC 2.4: Tasktitel suchen
Beschreibung	Der Titel eines Tasks wird gesucht.
Erwartetes Resultat	PC: Der Task mit dem entsprechenden Titel wird in der Übersichtstabelle angezeigt. Mobile: Der Task wird unterhalb des Suchformulars angezeigt.
Voraussetzung	Es wurde zuvor ein Task mit dem entsprechenden Titel gespeichert.
Testdaten	Ein Task mit dem entsprechenden Titel.
Vorgehen	PC: Im Bereich «Suche» den Titel in das Suchfeld eingeben und die Checkbox «Titel» aktivieren. Mobile: Den Titel im Suchfeld oben rechts eingeben.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.14: Systemtest UC 2.4

Name	UC 2.5: Task durchsuchen
Beschreibung	Der Inhalt eines Tasks wird inkrementell durchsucht.
Erwartetes Resultat	Die dem Suchwort entsprechenden Stellen werden sofort nach einem Anschlag (neu-)markiert.
Voraussetzung	Die Detailansicht des Tasks wurde geöffnet.
Testdaten	Ein zuvor gespeicherter Task.
Vorgehen	Das Suchwort im Suchfeld oben rechts eingeben.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.15: Systemtest UC 2.5

Name	UC 2.6: Erweiterte Tasksuche
Beschreibung	Es können diverse Parameter einer normalen Suche (UC 2.4) hinzugefügt werden. So kann zusätzlich der Pfad, in welchem gesucht wird, angegeben werden und es kann ausgewählt werden, ob nur nach Titeln, nur nach Inhalt oder in beidem gesucht werden soll.
Erwartetes Resultat	PC: Den Suchkriterien entsprechende Tasks werden in der Übersichtstabelle im Hauptfenster angezeigt. Mobile: Den Suchkriterien entsprechende Tasks werden unterhalb des Suchformulars angezeigt.
Voraussetzung	Es wurde zuvor mindestens ein Task, welcher die Suchkriterien erfüllt, gespeichert.
Testdaten	Der zuvor gespeicherte Task.
Vorgehen	PC: Im Bereich «Suche» die entsprechenden Suchkriterien setzen. Mobile: In der Taskübersicht auf den Menüeintrag «Erweiterte Suche» klicken und damit die Such-Activity öffnen. Danach die entsprechenden Suchkriterien setzen.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.16: Systemtest UC 2.6

Name	UC 2.7: Tasks verlinken
Beschreibung	Ein Task (TaskA) wird unidirektional mit einem anderen Task (TaskB) verlinkt.
Erwartetes Resultat	Es wurde in TaskA ein Shortcut auf TaskB erstellt.
Voraussetzung	Es existieren zwei Tasks oder Ordner (TaskA und TaskB).
Testdaten	TaskA und TaskB
Vorgehen	PC: Den Bereich «Bearbeiten» öffnen indem auf TaskA geklickt wird. Der Button «Shortcuts...» öffnet den Shortcutdialog. In diesem Dialog den Namen des Shortcuts festlegen und den Pfad von TaskB eingeben, danach auf Speichern klicken und den Dialog schliessen. Nun muss TaskA nur noch gespeichert werden. Mobile: Noch nicht implementiert.
Resultat	Der Shortcut wurde erstellt.
Kommentar	–
Testdatum	10.05.2011

Tabelle 2.17: Systemtest UC 2.7

Name	UC 2.8: Schnittstelle fügt neuen Task hinzu
Beschreibung	Die Schnittstelle zum Firmennetzwerk fügt einen Task hinzu.
Erwartetes Resultat	Der Task wurde auf dem PC hinzugefügt und mit dem Mobile-Device synchronisiert.
Voraussetzung	Der von der Schnittstelle kommende Task existiert noch nicht.
Testdaten	–
Vorgehen	Noch nicht implementiert.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.18: Systemtest UC 2.8

Name	UC 2.9: Tasks exportieren
Beschreibung	Alle Tasks eines bestimmten Zeitraums werden in einer eigenen Datei gespeichert.
Erwartetes Resultat	Die Tasks wurden aus der Datenbank entfernt und in die Datei gespeichert.
Voraussetzung	Es wurde zuvor mindestens ein Task gespeichert.
Testdaten	Die zuvor gespeicherten Tasks.
Vorgehen	Noch nicht implementiert.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.19: Systemtest UC 2.9

2.4.5 Verbindung

Name	UC 3.1: Verbindung herstellen
Beschreibung	Es wird eine Verbindung zwischen Mobile-Device und PC aufgebaut und die Dateien und Tasks werden synchronisiert.
Erwartetes Resultat	PC und Mobile: Die Dateien und Tasks, welche auf dem Mobile-Device bearbeitet/hinzugefügt/gelöscht wurden, wurden entsprechend aktualisiert. Nur Mobile: Es wurde in den Listen-Mode gewechselt, in welchem auf dem Mobile-Device keine Änderungen an Tasks vorgenommen werden können.
Voraussetzung	Das Mobile-Device und der PC sind bereits physisch verbunden.
Testdaten	–
Vorgehen	Mobile: Im Menü auf «Verbindung» klicken. Dies muss vor dem «Verbindungsstart» auf dem PC geschehen. PC: Im Menü «Verbindung» auf «Verbindung herstellen» klicken.
Resultat	Das Mobile-Device und der PC sind verbunden, Dateien und Tasks wurden ausgetauscht.
Kommentar	–
Testdatum	31.05.2011

Tabelle 2.20: Systemtest UC 3.1

Name	UC 3.2: Verbindung trennen
Beschreibung	Es wird eine Verbindung zwischen Mobile-Device und PC getrennt, zuvor werden die Dateien und Tasks synchronisiert.
Erwartetes Resultat	PC und Mobile: Die Dateien und Tasks, welche auf dem Mobile-Device bearbeitet/hinzugefügt/gelöscht wurden, wurden entsprechend aktualisiert. Nur Mobile: Der Listen-Mode wurde Verlassen.
Voraussetzung	Das Mobile-Device und der PC sind bereits verbunden.
Testdaten	–
Vorgehen	PC: Im Menü «Verbindung» auf «Verbindung trennen» klicken.
Resultat	Die Synchronisation der ausstehenden Tasks und Dateien hat stattgefunden.
Kommentar	Der Synchronisationsbildschirm verschwindet nicht bei jedem Trennvorgang.
Testdatum	31.05.2011

Tabelle 2.21: Systemtest UC 3.2

Name	UC 3.3: Benutzer über getrennte Verbindung informieren
Beschreibung	Der Benutzer möchte trotz getrennter Verbindung einen Task auf dem PC speichern und wird auf die fehlende Verbindung hingewiesen.
Erwartetes Resultat	Der Benutzer sieht ein Popup-Fenster.
Voraussetzung	Das Mobile-Device und der PC sind physisch nicht verbunden.
Testdaten	–
Vorgehen	PC: Ein neuer Task erstellen.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.22: Systemtest UC 3.3

2.4.6 Konfiguration

Name	UC 4.1: Applikationssprache wechseln
Beschreibung	Die Sprache des GUI wird verändert.
Erwartetes Resultat	Die Applikationsoberfläche wird nun in der gewählten Sprache dargestellt.
Voraussetzung	–
Testdaten	–
Vorgehen	PC: Im Menü «Ansicht» unter «Sprache» die gewünschte Sprache anklicken.
Resultat	–
Kommentar	Noch nicht implementiert.
Testdatum	10.05.2011

Tabelle 2.23: Systemtest UC 4.1

3 Unit Test

Der Domain-Layer wurde, so weit möglich, mit Unit Tests abgedeckt.

Zudem ist zu beachten, dass Android ein eigenes Testprojekt besitzt in welchem sich die Unit Tests befinden.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	06.05.2011	RR	Dokument erstellt.
1.0rc02	09.05.2011	DM	Korrekturen und Ergänzungen angebracht.
1.0rc03	10.05.2011	DM, RR	Getestet und Ergebnisse festgehalten.
1.0	12.05.2011	DM	Akzeptierte Version 1.0.
1.1rc01	19.05.2011	DM	Testcases für die Tasksynchronisation erstellt.
1.1rc02	31.05.2011	DM, RR	Getestet und Ergebnisse festgehalten.
1.1rc03	11.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.1	11.06.2011	RR	Akzeptierte Version 1.1.

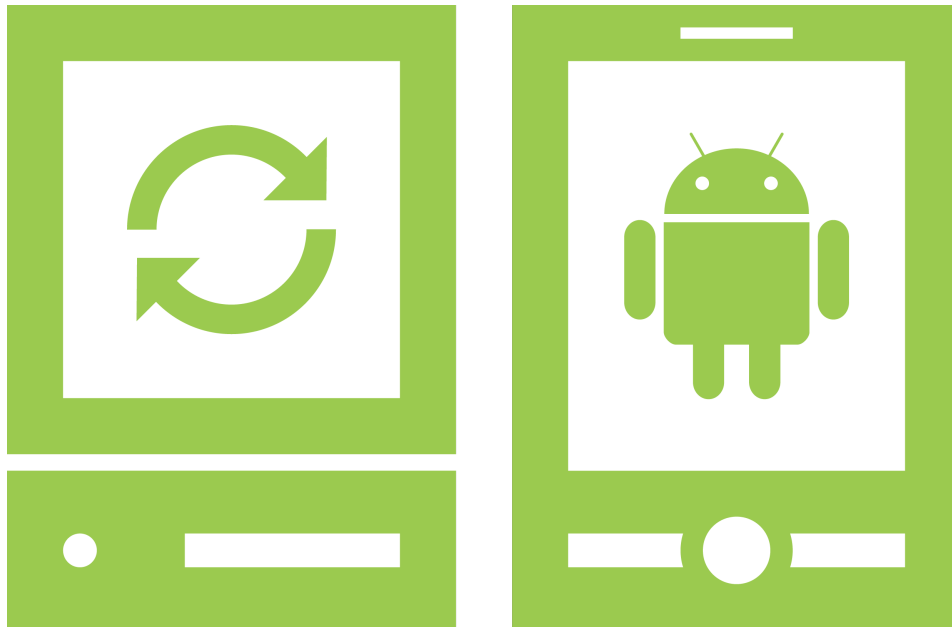
Tabelle 3.1: Revisionshistorie

Abkürzungsverzeichnis

ADB Android Debug Brindge
As Anforderungsspezifikation
Ba Bachelorarbeit
CRUD create, read, update, delete
GUI Graphical User Interface
PC Personal Computer
TCA Task-Management Control-Center for Android
UC Use Case
USB Universal Serial Bus

Tabellenverzeichnis

2.1	Systemtest Task erstellen	4
2.2	Systemtest Ordner erstellen	4
2.3	Systemtest Verschachtelter Task / Ordner erstellen	5
2.4	Systemtest Task / Ordner ändern	5
2.5	Systemtest Task / Ordner löschen	6
2.6	Systemtest UC 1.1	6
2.7	Systemtest UC 1.2	7
2.8	Systemtest UC 1.3	7
2.9	Systemtest UC 1.4	7
2.10	Systemtests Tasksynchronisation	8
2.11	Systemtest UC 2.1	9
2.12	Systemtest UC 2.2	9
2.13	Systemtest UC 2.3	10
2.14	Systemtest UC 2.4	10
2.15	Systemtest UC 2.5	10
2.16	Systemtest UC 2.6	11
2.17	Systemtest UC 2.7	11
2.18	Systemtest UC 2.8	12
2.19	Systemtest UC 2.9	12
2.20	Systemtest UC 3.1	12
2.21	Systemtest UC 3.2	13
2.22	Systemtest UC 3.3	13
2.23	Systemtest UC 4.1	13
3.1	Revisionshistorie	15



User Guide

«Task-Management Control-Center for Android»

Version 1.0, 11. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Installation	2
2.1 Vor der Installation	2
2.2 Installation der PC-Applikation	2
2.2.1 Konfigurieren der PC-Applikation	2
2.3 Installation der Mobile-Applikation	2
2.3.1 Debugging-Modus aktivieren	2
2.3.2 Installieren der Anwendung	3
2.3.3 Prioritäten	3
2.4 Abschliessende Schritte	3
3 Verbindung	4
3.1 Verbindung herstellen	4
3.2 Verbindung trennen	4
4 Tasks Verwalten	5
4.1 Arten von Tasks	5
4.2 Verwalten auf dem PC	5
4.2.1 Tasks anzeigen	5
4.2.2 Tasks, Ordner oder Timestamps hinzufügen	7
4.2.3 Tasks, Ordner oder Timestamps bearbeiten	7
4.2.4 Tasks, Ordner oder Timestamps löschen	7
4.3 Verwalten auf dem Mobile-Device	8
4.3.1 Tasks anzeigen	8
4.3.2 Tasks, Ordner oder Timestamps hinzufügen	8
4.3.3 Tasks, Ordner oder Timestamps bearbeiten	9
4.3.4 Tasks, Ordner oder Timestamps löschen	9
5 Dateiaustausch	10
5.1 Zu synchronisierende Ordner festlegen	10
5.2 Einen Ordner temporär von der Synchronisation ausschliessen	10

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 8.

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Installation

2.1 Vor der Installation

In dieser Installationsanleitung wird davon ausgegangen, dass Sie Ihr Mobile-Device bereits mittels USB mit dem PC verbunden haben, sowie die Treiber Ihres Mobile-Devices installiert sind.

Weiter wird davon ausgegangen, dass die Android Debug Brindge (ADB) bereits installiert wurde.

2.2 Installation der PC-Applikation

Kopieren Sie die Datei «TCA_PC.jar» und die Ordner «languages», «pictures», «properties» und «db» in das gewünschte Zielverzeichnis.

2.2.1 Konfigurieren der PC-Applikation

Öffnen Sie die Datei «constants.properties», welche sich im Ordner «properties» befindet und setzen Sie unter «MainFolder» den Pfad, unter welchem Sie Ihre Daten aufbewahren wollen.

Mittels «PriorityFileLocation» und «PriorityFileName» können Sie festlegen, wo die gespeicherten Prioritäten aufbewahrt werden. Es ist wichtig, dass diese Datei in einem eigenen Ordner liegt, da sie mit dem Mobile-Device synchronisiert werden muss.

Unter «DatabaseLocation» können Sie den Speicherpfad der Datenbank, in welcher Ihre Tasks gespeichert werden, relativ zum «MainFolder» angeben.

«PriorityFileLocationMobile» legt den Zielort der Prioritäten-Datei auf dem Mobile-Device fest. Weiteres dazu in Kapitel 2.3.3, Seite 3.

Kopieren sie nun die Datei «Tasks.db», welche Sie im Ordner «db» finden, an den Ort, welchen Sie unter «DatabaseLocation» angegeben haben und benennen Sie sie gegebenenfalls um. Zudem müssen Sie den Ordner, in welchem die Prioritäten-Datei abgelegt wird, erstellen. Die Prioritäten-Datei selbst, müssen Sie nicht erstellen.

2.3 Installation der Mobile-Applikation

2.3.1 Debugging-Modus aktivieren

Um TCA benutzen zu können, muss der Debugging-Modus auf dem Mobile-Device aktiviert sein.

Tippen Sie unter «Einstellungen» auf «Anwendungen», dann auf «Entwicklung» und aktivieren Sie «USB-Debugging».



Abbildung 2.1: Aktivieren des Debugging-Modus

2.3.2 Installieren der Anwendung

Öffnen sie auf dem PC die Konsole, wechseln Sie in den Ordner, in welchem sich die Datei «TCA_Android.apk» befindet und geben sie den folgenden Befehl ein:

```
adb install TCA_Android.apk
```

Die Anwendung wird nun automatisch installiert. Dies kann einen Moment dauern.

2.3.3 Prioritäten

Um die Prioritäten mit dem PC austauschen zu können, muss der Ordner, welcher in der Datei «constants.properties» unter «PriorityFileLocationMobile» angegeben wurde, zwingend existieren. Falls dieser Ordner noch nicht vorhanden ist, legen Sie ihn bitte an.

2.4 Abschliessende Schritte

Bevor Sie damit beginnen neue Tasks zu erfassen, müssen Sie eine erste Verbindung herstellen. Starten Sie beide Anwendungen und stellen Sie wie im Kapitel 3.1 auf Seite 4 beschrieben eine Verbindung her.

3 Verbindung

3.1 Verbindung herstellen

Um die Verbindung zu initiieren, öffnen Sie auf dem Mobile-Device den Hauptbildschirm und klicken Sie im Menü auf «Verbindung». Nun erscheint ein «Bitte warten»-Dialog.

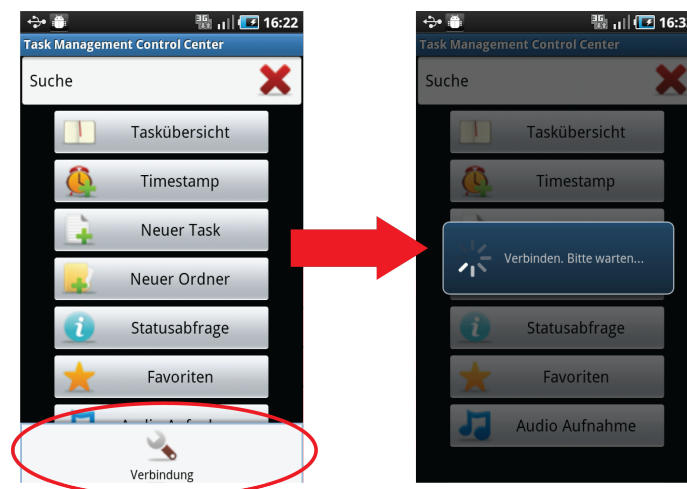


Abbildung 3.1: Verbindung initiieren

Wechseln Sie zur PC-Applikation und klicken Sie auf «Verbindung herstellen». Der Synchronisationsvorgang kann, abhängig von der auszutauschenden Datenmenge, einige Zeit dauern.



Abbildung 3.2: Verbindung herstellen

Die Verbindung wurde hergestellt. Die Tasks werden, solange das Mobile-Device mit dem PC verbunden bleibt, sofort ausgetauscht. Während einer bestehenden Verbindung dürfen auf dem Mobile-Device weder Tasks erstellt noch bearbeitet werden. Auf der PC-Seite kann normal weitergearbeitet werden.

3.2 Verbindung trennen

Um eine bestehende Verbindung zu trennen, klicken Sie in der PC-Applikation auf «Verbindung trennen».



Abbildung 3.3: Verbindung trennen

4 Tasks Verwalten

4.1 Arten von Tasks

Es gibt die folgenden drei Arten von Tasks:



Abbildung 4.1: vlnr: Timestamp, Ordner und Task

- **Timestamps:** Sie dienen dem schnellen Markieren eines Zeitpunkts mit einer kleinen Notiz.
- **Ordner:** Sie dienen der Sortierung der Tasks. Zudem können sie eine eigene Priorität, Notiz oder ein Fälligkeitsdatum haben, welches jedoch überschrieben wird, sobald sich andere Tasks und Ordner darin befinden. Pfade in einem Ordnerbaum werden mit einem «/» getrennt, z.B. Ordner 1/Ordner 2/Neuer Task.
- **Tasks:** Ein Task enthält alle wichtigen Informationen, welche Sie speichern wollen.

Bitte beachten Sie, dass ein Task-, Ordner oder Timestamptitel innerhalb eines Ordners immer eindeutig sein muss!

4.2 Verwalten auf dem PC

4.2.1 Tasks anzeigen

Bereits nach dem Start der Anwendung werden die Tasks in der Haupttabelle angezeigt.

Titel	Fällig am	Priorität	Beendet
Timestamp 1. Juni 2011 14:51			<input type="checkbox"/>
Bewerbungen	2. Juni 2011	low	<input checked="" type="checkbox"/>
Bachelorarbeit	17. Juni 2011 12:00	1	<input type="checkbox"/>
Prüfungen	31. Juli 2011	2	<input type="checkbox"/>
Codefreeze	1. Juni 2011	1	<input checked="" type="checkbox"/>
Einkaufen	2. Juni 2011	home	<input type="checkbox"/>
Geburtsfeier	2. Juni 2011	home	<input type="checkbox"/>
Abstract schreiben	30. Juni 2011	1	<input type="checkbox"/>

Abbildung 4.2: Die Haupttabelle

Befinden Sie sich in der Baumdarstellung, können Sie sich mittels Doppelklick auf einen Ordner dessen Inhalt anzeigen lassen. Wenn sie auf «..» klicken, gelangen Sie eine Ebene höher in der Hierarchie.

Falls Sie den exakten Pfad zum Zielordner kennen, können Sie diesen auch direkt im Feld oberhalb der Übersicht eingeben.

Wechseln der Visualisierung

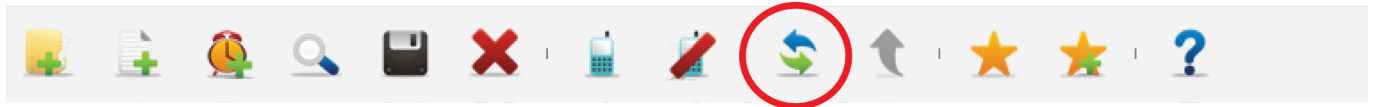


Abbildung 4.3: Wechseln der Visualisierung

Sie können zwischen Baum- und Zeitstrahldarstellung wechseln. Die Baumdarstellung ordnet die Tasks, ähnlich dem Windows-Explorer, in Ordner, Unterordner und Tasks an, die Zeitstrahldarstellung sortiert die Tasks nach dem Fälligkeits- bzw. Beendet-Datum an.

Anzeigen der Details

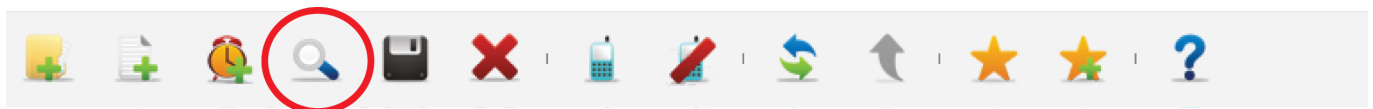


Abbildung 4.4: Details anzeigen

Die Details eines Task können Sie durch einen Doppelklick auf den gewünschten Task oder mit der «Enter»-Taste öffnen, weiter lässt sich dieser auch anzeigen, indem sie auf «Details anzeigen» klicken. Es öffnet sich das unten dargestellte Detail-Fenster.

Um die Details eines Ordners einsehen zu können, müssen Sie den Ordner zuerst markieren und danach auf «Details anzeigen» klicken.



Abbildung 4.5: Details eines Tasks

4.2.2 Tasks, Ordner oder Timestamps hinzufügen

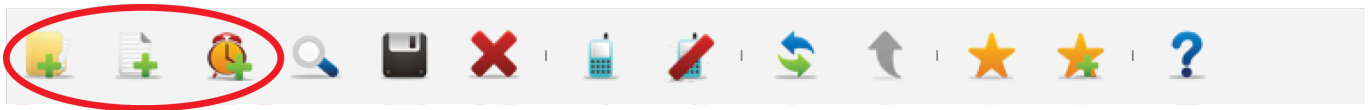


Abbildung 4.6: Einen Task, Ordner oder Timestamp hinzufügen

Klicken Sie auf «Neuer Task», «Neuer Ordner» oder «Neuer Timestamp», um einen Neuen Task, Ordner oder Timestamp zu erstellen.

Nach einem Klick auf «Neuer Task» oder «Neuer Ordner» öffnet sich rechts der Bearbeitungsbereich, in welchem Sie alle gewünschten Angaben vermerken können. Ein Timestamp wird sofort erstellt und gespeichert. Dieser erhält jeweils einen Standardtitel, der wie folgt aufgebaut ist: Timestamp gefolgt vom aktuellen Datum und der Uhrzeit, z.B. «Timestamp 7. Juni 2011 14:23».

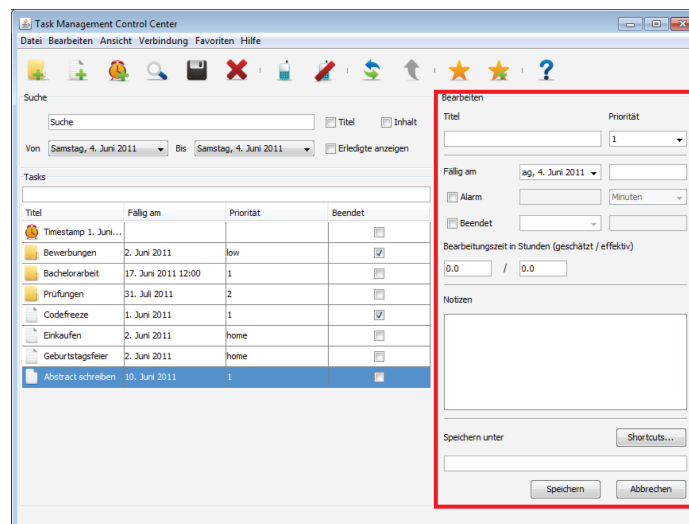


Abbildung 4.7: Der Bearbeitungsbereich

4.2.3 Tasks, Ordner oder Timestamps bearbeiten

Um eine Task, Ordner oder Timestamp zu bearbeiten, klicken Sie auf die entsprechende Tabellenzeile. Es öffnet sich der Bearbeitungsmodus rechts. Wird ein Ordner oder Timestamp bearbeitet, können einige Felder nicht genutzt werden, weil diese Felder für einen Ordner oder Timestamp nicht benötigt werden.

4.2.4 Tasks, Ordner oder Timestamps löschen



Abbildung 4.8: Einen Task, Ordner oder Timestamp löschen

Markieren Sie den Task, Ordner oder Timestamp welchen Sie löschen wollen und klicken Sie auf «Löschen»

4.3 Verwalten auf dem Mobile-Device

4.3.1 Tasks anzeigen

Tippen Sie im Hauptbildschirm auf «Taskübersicht» um die Tasks anzuzeigen.

Nun erscheint eine Übersicht aller unverschachtelter Tasks. Es kann zwischen Baum-Darstellung (analog zu z.B. Windows-Explorer), Zeitstrahl-Darstellung (Sortierung nach Datum und Zeit) und den Favoriten gewechselt werden. Wobei anzumerken ist, dass das Speichern von Favoriten in dieser Version der Software noch nicht möglich ist.

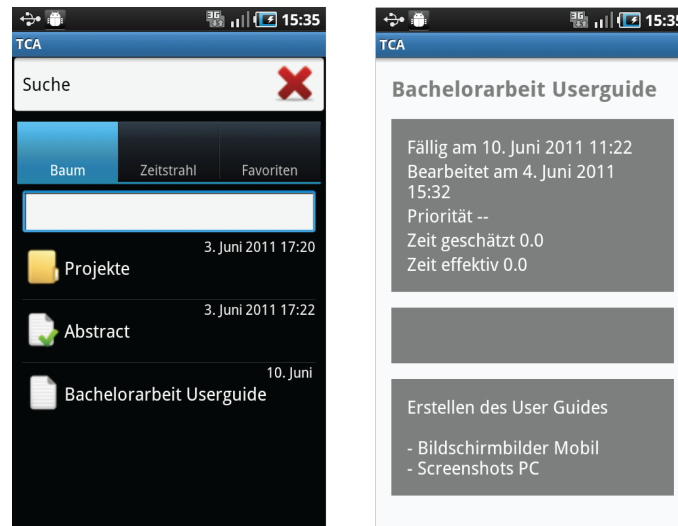


Abbildung 4.9: Links: Taskansicht in der Baumdarstellung, Rechts: Details eines Tasks

Um die Detailansicht eines Tasks oder Timestamp zu öffnen, tippen Sie auf den entsprechenden Task oder Timestamp. Die Details eines Ordners können Sie anzeigen, indem Sie zuerst in den Ordner wechseln und danach im Menü auf «Ordneigenschaften» tippen. Wenn Sie in der Baumdarstellung auf den «Bacbutton» tippen, gelangen Sie jeweils eine Ebene höher in der Hierarchie. Falls Sie den exakten Pfad zu einem Ordner kennen, können Sie diesen auch direkt im Feld oberhalb der Übersicht eingeben, um dorthin zu gelangen.

4.3.2 Tasks, Ordner oder Timestamps hinzufügen

Um einen neuen Task, Ordner oder Timestamp zu erstellen, tippen sie im Hauptbildschirm oder im Menü der «Taskübersicht» auf den entsprechenden Button.

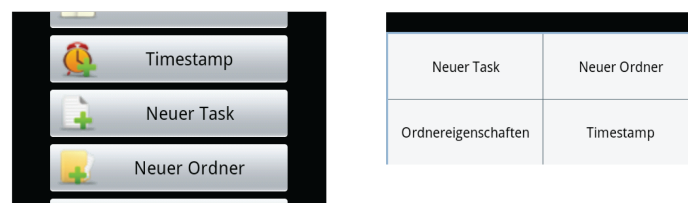


Abbildung 4.10: Einen neuen Task, Ordner oder Timestamp erstellen

Sie gelangen in den Bearbeitungsmodus, in welchem Sie alle Informationen des Tasks, Ordners oder Timestamps eingeben können. Beachten Sie dabei, dass ein Task und ein Ordner den gleichen Bearbeitungsbildschirm anzeigen. Jeweils nicht benötigte Felder sind deaktiviert.

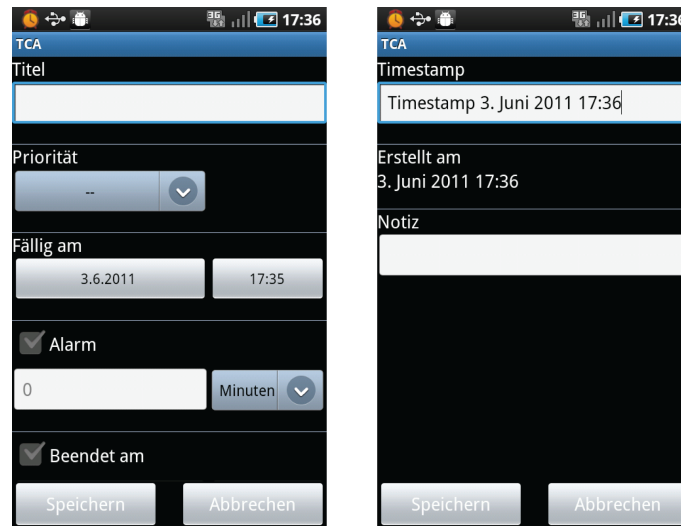


Abbildung 4.11: Links: Erstellen eines neuen Task oder Ordner, Rechts: Erstellen eines neuen Timestamp

4.3.3 Tasks, Ordner oder Timestamps bearbeiten

Öffnen Sie die Detailansicht eines Task, Ordners oder Timestamps und tippen Sie im Menü auf «Bearbeiten». Es öffnet sich der Bearbeitungsmodus, welcher im Kapitel 4.3.2 auf Seite 8 beschrieben wird.

Zu beachten ist, dass ein Timestamp nicht verschoben werden kann.

4.3.4 Tasks, Ordner oder Timestamps löschen

Wollen Sie einen Task, Ordner oder Timestamp löschen, öffnen Sie die Detailansicht und tippen Sie im Menü auf «Löschen».

Falls Sie einen Ordner löschen wollen, welcher nicht leer ist, müssen Sie dies zusätzlich in einem Dialog bestätigen.

5 Dateiaustausch

Um Dateien zwischen dem PC und dem Mobile-Device auszutauschen, öffnen Sie die Datei «path.properties», welche im Ordner «properties» liegt.

5.1 Zu synchronisierende Ordner festlegen

Für jedes Ordner-Paar, welches synchronisiert werden muss, erstellen sie folgenden Eintrag:

<Absoluter Pfad auf dem PC> = <Pfad auf dem Mobile relativ zum Verzeichnis «sdcard»>
z.B. C:\User\Projekt X\Austausch = /Projekt X

Dabei muss darauf geachtet werden, dass dem Doppelpunkt des PC-Laufwerks zwingend ein Backslash vorangestellt werden muss. Zudem müssen beide angegebenen Ordner auf dem jeweiligen System bereits existieren.

5.2 Einen Ordner temporär von der Synchronisation ausschliessen

Um einen Ordner temporär von der Synchronisation auszuschliessen, stellen sie der entsprechenden Zeile ein «#»-Zeichen voran.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	02.06.2011	RR	Dokument erstellt.
1.0rc02	04.06.2011	RR	Kapitel 2-5 erstellt.
1.0rc03	11.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.0	11.06.2011	RR	Akzeptierte Version 1.0.

Tabelle 5.1: Revisionshistorie

Abkürzungsverzeichnis

ADB Android Debug Brindge

Ba Bachelorarbeit

PC Personal Computer

TCA Task-Management Control-Center for Android

USB Universal Serial Bus

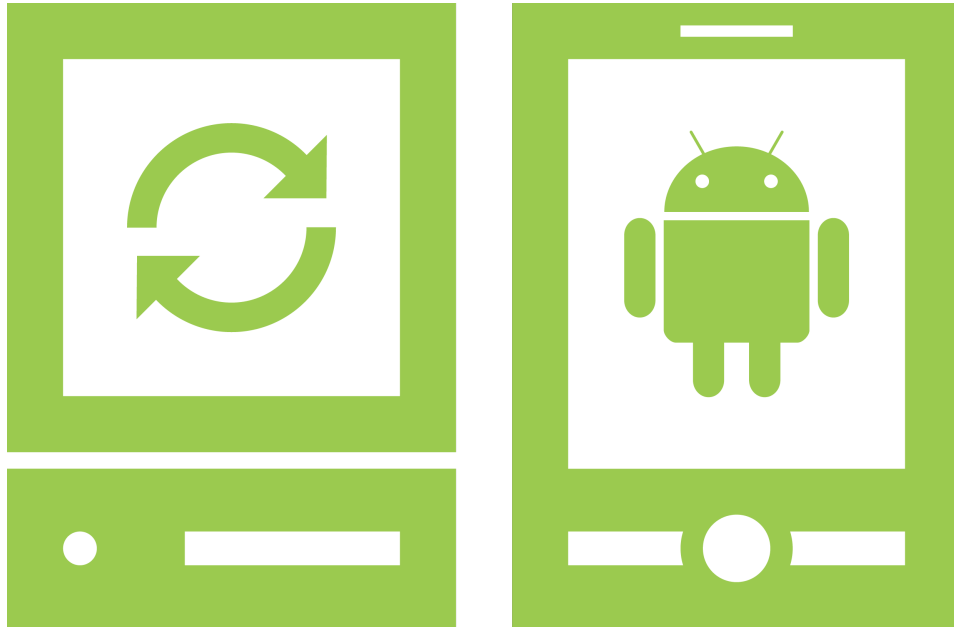
Abbildungsverzeichnis

2.1	Aktivieren des Debugging-Modus	3
3.1	Verbindung initiieren	4
3.2	Verbindung herstellen	4
3.3	Verbindung trennen	4
4.1	vInr: Timestamp, Ordner und Task	5
4.2	Die Haupttabelle	5
4.3	Wechseln der Visualisierung	6
4.4	Details anzeigen	6
4.5	Details eines Tasks	6
4.6	Einen Task, Ordner oder Timestamp hinzufügen	7
4.7	Der Bearbeitungsbereich	7
4.8	Einen Task, Ordner oder Timestamp löschen	7
4.9	Links: Taskansicht in der Baumdarstellung, Rechts: Details eines Tasks	8
4.10	Einen neuen Task, Ordner oder Timestamp erstellen	8
4.11	Links: Erstellen eines neuen Task oder Ordner, Rechts: Erstellen eines neuen Timestamp	9

Tabellenverzeichnis

5.1 Revisionshistorie	11
---------------------------------	----

Literaturverzeichnis



Glossar

«Task-Management Control-Center for Android»

Version 1.0, 11. Juni 2011

Inhaltsverzeichnis

1 Einführung	1
1.1 Zweck des Dokuments	1
1.2 Gültigkeitsbereich	1
2 Begriffe	2

1 Einführung

1.1 Zweck des Dokuments

Siehe «Projektplan», Kapitel 4.2.2, Seite 8

1.2 Gültigkeitsbereich

Dieses Dokument dient als Grundlage für das Projekt Task-Management Control-Center for Android (TCA) und behält daher seine Gültigkeit während der gesamten Projektdauer.

2 Begriffe

Tabelle 2.1 (Seite 4) listet wichtige Begriffe und ihre Bedeutung im Rahmen des Projekts auf. In der Spalte Kontext ist aufgeführt, in welchem Zusammenhang welche Definition des Begriffs verwendet wird.

Begriff	Kontext	Definition und Information
AnCoF	-	Das Android Control Framework (AnCoF) stellt Methoden zur Verfügung, um Funktionalitäten des Mobile-Devices über den PC abzuwickeln.
Android	-	Betriebssystem sowie Software-Plattform für mobile Geräte wie Smartphones, Mobiltelefone und Netbooks, die von der Open Handset Alliance entwickelt werden.
Ap	Planung	Dokument: Arbeitspakete (Ap)
As	Anforderungen	Dokument: Anforderungsspezifikation (As)
CRUD	-	create, read, update, delete (CRUD) Operationen.
Da	Analyse	Dokument: Domainanalyse (Da)
DD	Design	Dokument: Design Dokument (DD)
Device	-	Ein Gerät, welches AnCoF benutzen kann (z.B. PC, Mobile-Device). In der Dokumentation wird der Begriff «Gerät» als Synonym verwendet.
DRY	Dokumentation und Implementation	Don't repeat yourself (DRY), das heisst: Keinen doppelten Text oder Code verfassen.
EA	Design	Der Enterprise Architect (EA) ist ein Unified Modeling Language (UML)- und Business-Modellierungs-Tool.
Eclipse	Entwicklung	Entwicklungsumgebung
Folder	Dokumentation	Bezeichnet einen besonderen Task, welcher andere Tasks beinhalten kann. Wird synonym zu Ordner verwendet.
Gl	Dokumentation	Dokument: Glossar (Gl)
GUI-Design	Design	Dokument: GUI-Design
Inkrementelle Suche	-	Die Suche nach dem gewünschten Begriff startet bereits während dieser eingegeben wird.
Interface	-	Die Begriffe Interface und Schnittstelle werden synonym verwendet.
Java	Implementation	Objektorientierte Programmiersprache
Mobile-Device	-	Mobiles Gerät wie zum Beispiel ein Smartphone oder Mobiltelefon
Ordner	Dokumentation	Bezeichnet einen besonderen Task, welcher andere Tasks beinhalten kann. Wird synonym zu Folder verwendet.
Pp	Planung	Dokument: Projektplan (Pp)

Begriff	Kontext	Definition und Information
RegEx	-	Zeichenkette, die der Beschreibung von Mengen beziehungsweise Untermengen von Zeichenketten mit Hilfe bestimmter syntaktischer Regeln dient.
SAD	Design	Dokument: Software Architektur Dokument (SAD)
Schnittstelle	-	Schnittstelle von TCA zum Firmennetzwerk. Die Begriffe Interface und Schnittstelle werden synonym verwendet.
Standardtask	Dokumentation	Bezeichnet eine Task, welchen Zusatzinformationen wie z.B. ein Start- und ein Enddatum besitzt.
TaMaF	-	Mittels dem Task-Management-Framework on Smart-Phone (TaMaF) kann eine Verbindung über USB oder WLAN zwischen einem Mobile-Device und dem PC aufgebaut werden.
Task	-	Ein Task ist eine Aufgabe, welche es zu erledigen gilt. Task wird als Sammelbegriff für jegliche Arten von «Tasks» verwendet(Ordner, Standardtasks und Zeitstempel), wenn nichts genaueres definiert wird.
Td	Construction	Dokument: Testdokumentation (Td)
Timestamp	Dokumentation	Task, welcher einen bestimmten Zeitpunkt mit Angabe der genauen Zeit festhält. Wird synonym zu Zeitstempel verwendet.
UG	Entwicklung	Dokument: User Guide (UG)
USB	-	USB ist ein serielles Bussystem zur Verbindung eines PC's mit externen Geräten.
Vision	Planung	Dokument: Vision
Zeitstempel	Dokumentation	Task, welcher einen bestimmten Zeitpunkt mit Angabe der genauen Zeit festhält. Wird synonym zu Timestamp verwendet.

Tabelle 2.1: Liste der projektrelevanten Begriffe

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	24.02.2011	DM	Dokument erstellt.
1.0rc02	09.06.2011	DM	Glossar ergänzt.
1.0rc03	11.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.0	11.06.2011	RR	Akzeptierte Version 1.0.

Tabelle 2.2: Revisionshistorie

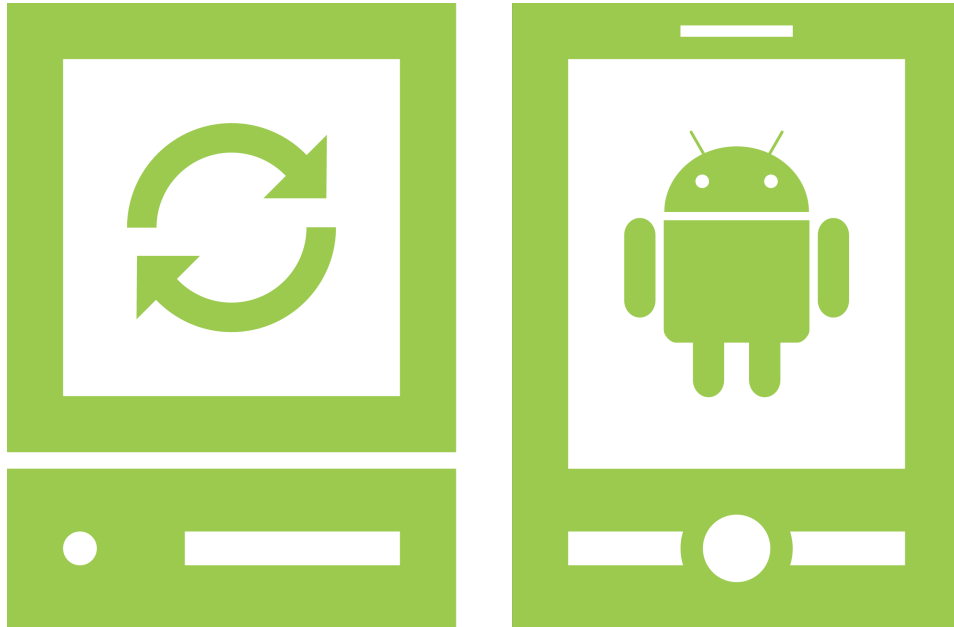
Abkürzungsverzeichnis

AnCoF Android Control Framework
Ap Arbeitspakete
As Anforderungsspezifikation
Ba Bachelorarbeit
CRUD create, read, update, delete
Da Domainanalyse
DD Design Dokument
DRY Don't repeat yourself
EA Enterprise Architect
GI Glossar
GUI Graphical User Interface
PC Personal Computer
Pp Projektplan
RegEx Regular Expression
SAD Software Architektur Dokument
TCA Task-Management Control-Center for Android
TaMaF Task-Management-Framework on Smart-Phone
Td Testdokumentation
UG User Guide
UML Unified Modeling Language
USB Universal Serial Bus
WLAN Wireless Local Area Network

Abbildungsverzeichnis

Tabellenverzeichnis

2.1	Liste der projektrelevanten Begriffe	4
2.2	Revisionshistorie	5



Literatur

«Task-Management Control-Center for Android»

Version 1.0, 11. Juni 2011

Verwendete Werke

Becker, Arno, Pant, Marcus (2009): Android. Grundlagen und Programmierung, 1. Auflage, Heidelberg.
Larman, Craig (2004): Applying UML and Patterns. 3. Auflage, Upper Saddle River.
Morris, Jason (2011): Android User Interface Development, 1. Auflage, Birmingham.

Verwendete Studienarbeiten

Android Control Framework (2010).
Task-Management-Framework on Smart-Phone (2009).

Internetreferenzen

Allgemein

Anleitung für Dokumentation Semester-, Bachelor- und Diplomarbeiten (2011):

https://www.hsr.ch/index.php?eID=tx_nawsecured1&u=5695&file=fileadmin/user_upload/customers/hsr/HSR-INTERN/Bachelor-Studiengaenge/Informatik/Vorlagen/DokuAnleitungBA_DA_SA_100304.pdf&t=1307611691&hash=ff5888025f7be8446e2143dfbff72bcb, 08. Juni 2011.

Android

Android: <http://developer.android.com/>, 08. Juni 2011.

Mobile-Applikation

ActivityRegistry:

<http://code.google.com/p/mensaupb/source/browse/trunk/Mensa2/src/mensa/upb/ActivityRegistry.java?spec=svn8&r=8>, 08. Juni 2011.

PC-Applikation

Datepicker:

Freixas, Toni: <http://flib.sourceforge.net/index.html>, 08. Juni 2011.

Bilder

Open Clipart Library: <http://www.openclipart.org/>, 08. Juni 2011.

dryicons: <http://dryicons.com/>, 08. Juni 2011.

Abkürzungsverzeichnis

Ba Bachelorarbeit

Task-Management Control-Center for Android



Daniela Meier



Ramona Rudnicki

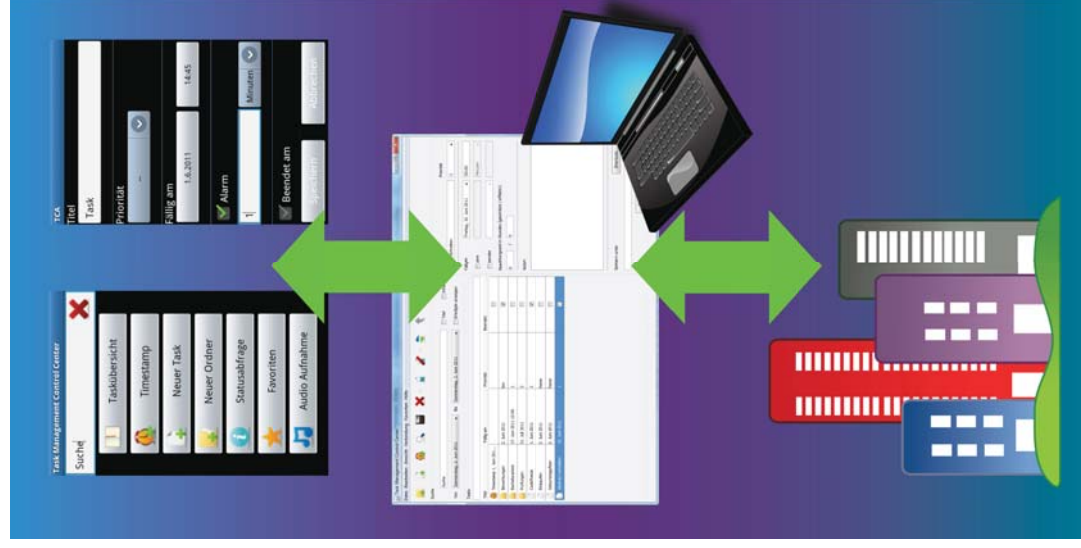
Betreuer: Thomas Letsch

Experte: Prof. Dr. Martin Zimmermann

Projektpartner: Letsch Informatik, Wald ZH

Bachelorarbeit Frühjahrssemester 2011

Themengebiet: Software



Voraussetzungen

Task-Management Control-Center for Android baut auf den Studienarbeiten...

- "Task Management Framework"
 - "Android Control Framework"
- ... auf.

Unterstützte Systeme

Task-Management Control-Center for Android unterstützt...

- Java ab Version 6 auf Windows und Linux
- Android ab Version 2.1

Erreichte Ziele

Task-Management Control-Center for Android ermöglicht...

- einfache Erfassung von Aufgaben und Terminen (Tasks)
- Strukturierte Ablage von Tasks
- Verwaltung und Synchronisation von Tasks und Dateien

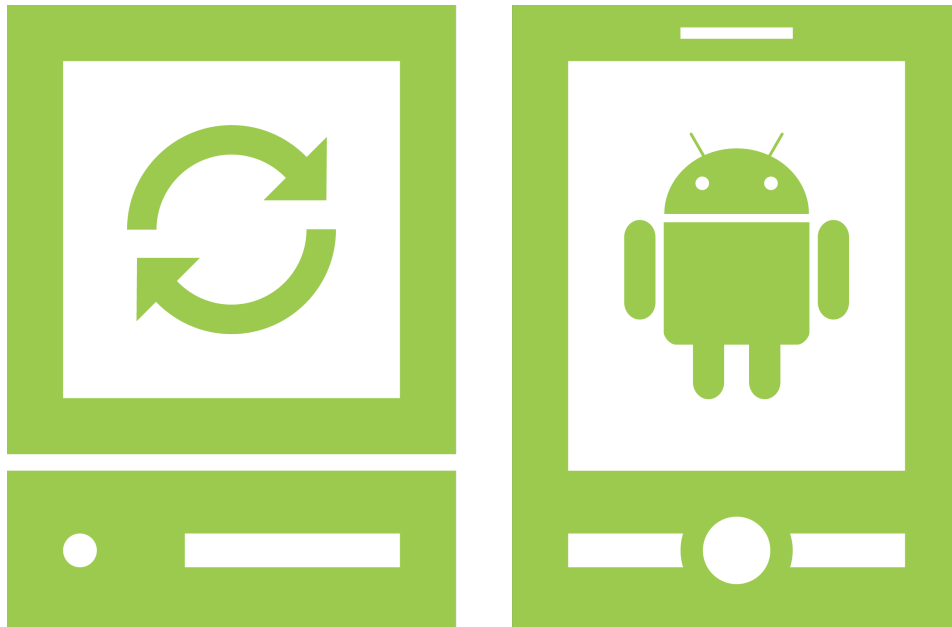
... auf dem PC sowie auf Android Smartphones.

Ausblick

Task-Management Control-Center for Android sieht für die Zukunft...

- Unterstützung von Mac OS
 - Einbindung in ein Firmennetzwerk
- ... vor.





Technischer Bericht

«Task-Management Control-Center for Android»

Version 1.0, 11. Juni 2011

Inhaltsverzeichnis

1	Einleitung	1
2	Ergebnisse	2
2.1	Taskpersistierung	2
2.1.1	PC-Applikation	2
2.1.2	Mobile-Applikation	2
2.2	Task-, Dateiverwaltung bzw. -synchronisation	2
2.2.1	Verbindung	2
2.3	Aussicht	3
3	Fazit	4
3.1	PC-Applikation	4
3.2	Mobile-Applikation	4

1 Einleitung

Um Notizen oder Termine (Tasks), welche auf einem Android-Smartphone erstellt wurden, mit dem PC zu synchronisieren, benötigt man zurzeit mehrere «Apps» respektive PC-Applikationen. Möchte man diese zusätzlich noch in ein Firmennetzwerk weiterleiten respektive die Tasks aus einem solchen Netzwerk mit dem PC oder Android-Gerät austauschen, stellt dies ein weiteres Hindernis dar, welches momentan nicht befriedigend gelöst werden kann.

Basierend auf dem «Android Control Framework (AnCoF)»¹ sollte eine Oberflächen-Applikation auf PC- sowie Mobile-Seite geschaffen werden, welche es erlaubt, Dateien und Notizen zu verwalten und zu synchronisieren. Um eine solche gesamtheitlich Lösung zu schaffen, musste auch auf das Task-Management-Framework on Smart-Phone (TaMaF) zurückgegriffen werden.² TaMaF bereits in AnCoF integriert, da für die Realisierung der Schnittstellen, welche AnCoF zur Verfügung stellt, auf Funktionen von TaMaF zurückgegriffen wird.

¹AnCoF ist ein Framework für Android-Geräte, das Remote-Zugriff auf ausgewählte Funktionen des Android-Systems bietet und damit die Realisation gängiger Aufgaben einer PC-Suite ermöglicht.

²TaMaF stellt Schnittstellen für den Verbindungsaufbau zwischen einem Mobile-Device und dem PC sowie für die Synchronisation von Tasks zur Verfügung.

2 Ergebnisse

Als Lösung für das «Applikationen-Wirrwarr» ist das «Task-Management Control-Center for Android» resultiert. Auf PC- sowie Mobile-Seite können über eine Benutzeroberfläche Notizen und Termine verfasst, aktualisiert und gelöscht werden. Der Übersichtlichkeit wegen können diese mittels Ordnern strukturiert werden. Durch die automatische Synchronisation beim Verbinden des PCs mit dem Mobile-Device erlaubt das Task-Management Control-Center for Android (TCA), jederzeit die aktuellste Version auserwählter Dateien und aller Tasks auf dem entsprechenden Gerät abzurufen.

2.1 Taskpersistierung

Um Tasks verwalten zu können, müssen diese gespeichert werden.

2.1.1 PC-Applikation

Für die Persistierung der Tasks auf der PC-Seite wurde auf eine SQLite-Datenbank zurückgegriffen. Die Grundlage für die Taskverwaltung wurde bereits durch TaMaF zur Verfügung gestellt. Die vorhandene Klassenstruktur respektive deren Abbildung auf die Datenbank musste jedoch an die zu erfüllenden Aufgaben angepasst werden. Änderungen erfolgten primär damit eine effiziente und hierarchische Taskstrukturierung möglich wurde.

2.1.2 Mobile-Applikation

Auf der Mobile-Seite wurde ebenfalls eine SQLite-Datenbank verwendet, welche dieselbe Abbildung der Klassenstruktur hat. Da jedoch der Datenbankzugriff nicht analog jenem in einer PC-Anwendung ist, konnten nicht dieselben Klassen verwendet werden. Der Datenbankzugriff geschieht hier über einen «ContentProvider», der die Zugriffe organisiert und von aussen her über eine Wrapper-Klasse aufgerufen wird, welche die Methoden für die Datenbank-Manipulationen zur Verfügung stellt.

2.2 Task-, Dateiverwaltung bzw. -synchronisation

Sind die Tasks gespeichert, müssen diese zwischen den Geräten ausgetauscht werden. Dieser Austausch, welcher der Synchronisation dient, wurde über die bereitgestellten Schnittstellen von TaMaF realisiert, auf die über AnCoF zugegriffen werden konnte. Die Dateien werden ebenfalls mit Hilfe von AnCoF synchronisiert.

2.2.1 Verbindung

Beim Verbinden werden jeweils die vorhandenen Tasks und alle Dateien synchronisiert. Um festzulegen, welche Dateien ausgetauscht werden sollen, müssen die Pfade zu den Ziel- und Quellordnern, welche die Dateien enthalten, in der «Pfad-Datei» notiert werden. Wird die Verbindung beendet, findet ebenfalls eine Synchronisation statt. Wurden in der Zwischenzeit Tasks oder Dateien auf der PC-Seite bearbeitet, werden diese nun ausgetauscht. Bei jedem Austausch wird zusätzlich die Prioritäten-Datei auf das Mobile-Device übertragen.

Bei einem unerwarteten Abbruch der Synchronisation, wenn z.B. die Verbindung vom Mobile-Device und dem PC vor der vollständigen Synchronisation unterbrochen wird, werden die Tasks zur Zeit nicht korrekt in der von TaMaF zur Verfügung gestellten «Communication-Datenbank», welche den Synchronisationsstatus der Tasks festhält, gespeichert. Dies bedeutet, dass diese Tasks bei einer erneuten Synchronisation nur ausgetauscht werden, wenn sie in der Zwischenzeit nochmals bearbeitet wurden.

2.3 Aussicht

Da die Taskverwaltung mehr Zeit in Anspruch genommen hat, wie erwartet, konnten nicht alle vorgenommenen Ziele erreicht werden. Unter anderem ist für eine zukünftige Version eine inkrementelle Durchsichtung aller Tasks geplant und auch der Austausch von Tasks und Dateien mit einem Firmennetzwerk musste als Erweiterung für einen nächsten Release aufgeschoben werden. Ebenfalls bereits geplant, jedoch noch nicht implementiert, sind folgende «Features»:

- Favoriten-Liste analog jener im Windows Explorer
- zur Laufzeit konfigurierbare Applikationssprache
- Audiotask, der es erlaubt, Informationen aufzuzeichnen
- Statusabfrage
- Export von Tasks in Datei zwecks Archivierung

3 Fazit

Rückblickend kann gesagt werden, dass sich trotz sorgfältiger Planung immer auch Verbesserungspotential findet. Nachfolgend die wichtigsten Punkte und Entscheidungen, welche bei der Umsetzung der Arbeit getroffen wurde.

TCA wurde in zwei Projekte aufgeteilt, eines für die Mobile- und eines für die PC-Applikation. Android-Applikationen können zwar in Java implementiert werden, jedoch unterscheidet sich die Dalvik Virtual Machine von der Java Virtual Machine, so dass ein einziges Java-Projekt nicht realisierbar gewesen wäre. Dieser Umstand bedeutete, dass einzelne Klassen schlussendlich sehr ähnlich ausschauen. Wo möglich, wurde doppelten Code jedoch vermieden, indem die betreffenden Klassen im Android-Projekt liegen und vom Java-Projekt her als class-Folder in den «Build-Path» eingebunden werden.

3.1 PC-Applikation

Die PC-Applikation bildet das Herzstück von TCA. Es wurde besonderen Wert auf das Graphical User Interface (GUI) gelegt, da dieses im Gegensatz zu bestehenden Anwendungen eine möglichst effiziente und einfache Taskverwaltung ermöglichen soll.

Damit Bilder, Pfade, die Sprache und Konstanten für die Fenstergrösse der Applikation jederzeit an die vorherrschende Situation angepasst werden kann, wurden diese Konfigurationsdaten in entsprechende «Property-Dateien» ausgelagert. Die Erarbeitung dieser Lösung erscheint momentan etwas umständlich, jedoch kann durch den zusätzlichen Aufwand, der für diese «dynamische» Konfigurationslösung getätigt wurde, bei Weiterführung des Projekts bzw. der Implementierung weiterer «Features» in Zukunft Zeit eingespart werden.

Aufgrund dieses Mehraufwands musste darauf verzichtet werden, die Alarm-Funktion der Tasks vollständig zu implementieren und auch die Anbindung ans Firmennetzwerk musste, wie bereits erwähnt, verschoben werden.

3.2 Mobile-Applikation

Hier gestaltete sich vor allem der Datenbank-Zugriff recht umständlich. Um die Datenbank testen zu können, musste ein «ContentProvider» implementiert werden. Zusätzlich zum «ContentProvider», welcher für den Datenbankzugriff zuständig ist, wurde entschieden, einen «TaskWrapper» zu implementieren. Dieser stellt Methoden für Datenbankzugriffe zur Verfügung. Der Umweg über diese zusätzliche Wrapper-Klasse scheint für Aussenstehende unnötig zu sein, jedoch erleichtert er das Testen der Datenbankzugriffe und erlaubt eine saubere Trennung zwischen der Datenbank und dem restlichen Code.

Revisionshistorie

Version	Datum	Person	Änderung
1.0rc01	06.06.2011	dm	Dokument erstellt.
1.0rc02	07.06.2011	dm	Kapitel verfasst.
1.0rc03	11.06.2011	DM, RR	Rechtschreibung und Verweise überprüft.
1.0	11.06.2011	RR	Akzeptierte Version 1.0.

Tabelle 3.1: Revisionshistorie

Abkürzungsverzeichnis

AnCoF Android Control Framework

Ba Bachelorarbeit

GUI Graphical User Interface

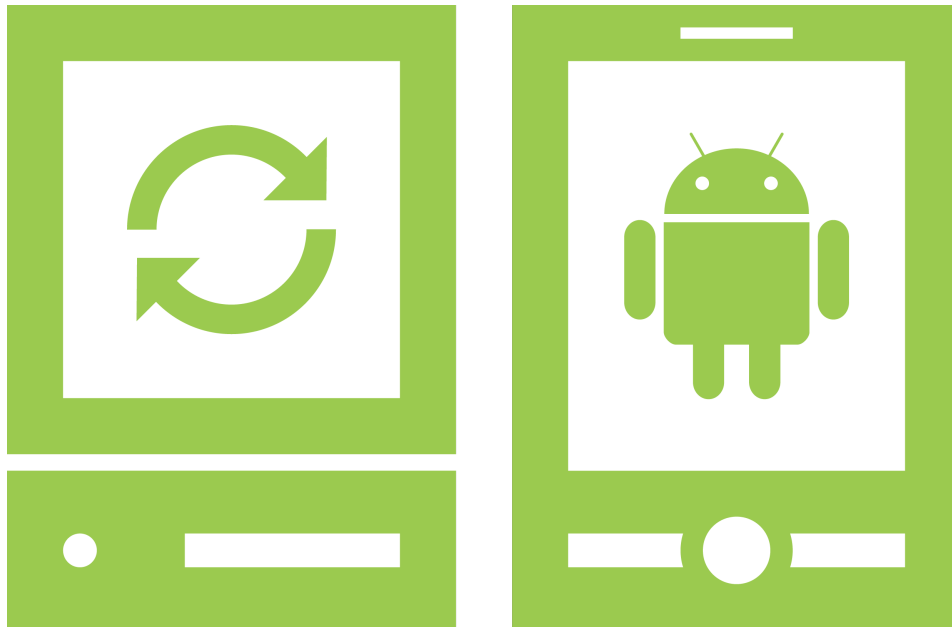
PC Personal Computer

TCA Task-Management Control-Center for Android

TaMaF Task-Management-Framework on Smart-Phone

Tabellenverzeichnis

3.1 Revisionshistorie	5
---------------------------------	---



Persönliche Berichte

«Task-Management Control-Center for Android»

Version 1.0, 13. Juni 2011

Persönlicher Bericht Daniela Meier

Die Bachelorarbeit (Ba), welche den Abschluss des Studiums bildet, war eine ganz besondere Herausforderung für mich. Die Erfahrungen aus der Studienarbeit hatten gezeigt, dass es noch einiges an Verbesserungspotenzial für die Ba gibt, was nun auch umgesetzt werden sollte. Da wir beschlossen hatten, eine Fortsetzungsarbeit zu machen, waren wir direkt mit der Arbeit und ihren «Tücken» konfrontiert, welche wir im vorigen Semester implementiert und dokumentiert hatten.

Der Beginn der Arbeit gestaltete sich recht ähnlich wie in der Studienarbeit, es wurde wiederum vorwiegend geplant, angepasst und spezifiziert. Da jedoch grosse Teile der erforderlichen Dokumente sich ähnlich gestalteten, konnten wir auf die bereits fertig gestellten Unterlagen des letzten Semesters zurückgreifen und brauchten daher weniger Zeit, die entsprechenden Papiere zu erstellen. Im Gegensatz zur letzten Arbeit begannen wir dieses mal bereits sehr früh mit der Implementierung erster Prototypen und des Architekturskeletts.

Da es sich beim Task-Management Control-Center for Android (TCA) um eine respektive zwei Oberflächen-Applikationen handelt, wurde besonders viel Zeit in die Gestaltung des Graphical User Interface (GUI)s investiert. Vor allem auf dasjenige auf der Personal Computer (PC)-Seite wurde besonders geachtet, da es eine möglichst einfache Bedienung unseres Programms erlauben sollte und bereits konkrete Vorstellungen betreffend des Aussehens vorgegeben wurden. Dank der Hilfe eines GUI-Builders konnten diese Vorgaben relativ schnell umgesetzt und ein erster Prototyp der graphischen Oberfläche vorgelegt werden. Entgegen den Erwartungen wurde für die Oberflächengestaltung auf der Mobile-Seite mehr Zeit benötigt als angenommen. Manche Bilder bzw. Vorstellungen der einzelnen Bildschirme, die bereits vor Augen schwebten, gestalteten sich in der Umsetzung als schwierig oder gar nicht realisierbar.

Bei der Logik des Programms war es glücklicherweise gerade umgekehrt, die verlorene Zeit für die Gestaltung des Mobile-GUIs konnte hier wieder egalisiert werden, da grosse Teile ähnlich jenen auf der PC-Seite waren und zu grossen Teilen nur übernommen und den Gegebenheiten angepasst werden konnten.

Rückblickend kann ich sagen, dass wieder einiges mehr an Erfahrung dazu gekommen ist und ich bestätigt wurde, das richtige Studium gewählt zu haben. Planen, spezifizieren und implementieren macht immer noch genau soviel Spass wie am Anfang und die Zeit ging wie im Fluge vorbei. Auch war ich froh, dass ich nebst der Ba nicht noch der Belastung ausgesetzt war, weitere Module besuchen zu müssen. Auch wenn dies zeitlich gesehen machbar gewesen wäre, ist es einfacher gesagt als getan. Denn dies hätte vorausgesetzt, dass die übrige Zeit voll ausgenutzt hätte werden müssen und 8 Stunden am Stück an einer Arbeit zu schaffen, ist ermüdend und erfordert unglaublich viel Konzentration.

Persönlicher Bericht Ramona Rudnicki

Zu Beginn der Ba, noch bevor wir die genaue Aufgabenstellung erhielten, fragte ich mich, wie wir wohl während einem Semester so viel Arbeit in ein Projekt investieren können. Es erschien wie eine winzige Aufgabe mit sehr viel Zeit. Ich kannte die grundlegende Aufgabe bereits, da wir uns entschieden hatten, eine Fortsetzungsarbeit vom Android Control Framework (AnCoF) zu schreiben. Die genaue Aufgabenstellung, welche uns Herr Letsch zu Semesterbeginn zukommen liess, war sehr offen formuliert und liess viel Raum für Interpretationen und eigene Ideen offen. Während dem Ausarbeiten der Spezifikationen erwies sich diese Unschärfe in der Aufgabenstellung als sehr angenehm, da wir auch unsere Vorstellungen einbringen konnten. Einerseits war ich sehr froh, etwas persönliches einbringen zu können, andererseits brachte auch Herr Letsch so viele Ideen ein, dass ich dachte, wir würden Unmengen an Überstunden anhäufen. Ich war Herr Letsch sehr dankbar, dass er immer wieder betonte, dass wir im Moment alle möglichen Anforderungen sammeln und zu einem späteren Zeitpunkt entscheiden, welche der Anforderungen wir tatsächlich umsetzen.

Am Anfang der Studienarbeit haben wir zu viel Zeit für das Planen verwendet bzw. verstreichen lassen. In diesem Semester gab ich mir grosse Mühe, bereits in den aller ersten Wochen mit ersten Implementations-tests zu beginnen. Dies erwies sich in den darauffolgenden Wochen als sehr wertvoll. Die Grundarchitektur stand etwas früher als geplant und es konnte rechtzeitig mit dem Implementieren der Features begonnen werden. Da wir die GUI-Implementierung von Android zeitlich unterschätzt haben, waren wir, als der erste Release fällig war, sehr froh um dieses «Zeitpolster».

Sehr erstaunt war ich, wie einfach sich die Aufgabenteilung gestaltete. Da Daniela zu Beginn für die Dokumentation verantwortlich war, machte ich mich daran, die Architektur aufzubauen und zu planen. Als die ersten Dokumente standen, war die Architektur noch nicht so weit, dass sie bereits genutzt werden konnte. Deshalb begann sich Daniela in die GUI-Implementation von Android einzuarbeiten. Somit waren die Hauptaufgabengebiete verteilt, Daniela beschäftigte sich hauptsächlich mit der Android-Oberfläche und ich trieb die PC-Applikation weiter voran. Dass ich deshalb weniger über die Android-Programmierung gelernt habe, bedaure ich etwas, dafür konnte ich meine Kenntnisse in Java Swing auffrischen.

Als ich begann, AnCoF in das Grundgerüst der Applikationen einzubauen, fiel mir auf, dass ich bereits wenige Monate nach der Implementation des Frameworks nicht mehr auswendig wusste, wie es benutzt wird. Mit Hilfe des Developer Guides und den Beispielprojekten von AnCoF konnte ich mein Gedächtnis auffrischen.

Alles in allem ist es sehr schade, dass es bereits Sommer ist. Ich hätte sehr gerne noch die eine oder andere Anforderung umgesetzt. Die Ba war sehr lehrreich und hat mir nochmals den Entwicklungsprozess einer Software vor Augen geführt. Zudem wurde mir erneut bewusst, dass mir immer noch die Erfahrung fehlt, Zeitaufwände im voraus richtig einzuschätzen. Da dies nur mit der Zeit «behoben» werden kann, hoffe ich, dass ich meinem Ziel mit diesem Semester bereits etwas näher gekommen bin.

Abkürzungsverzeichnis

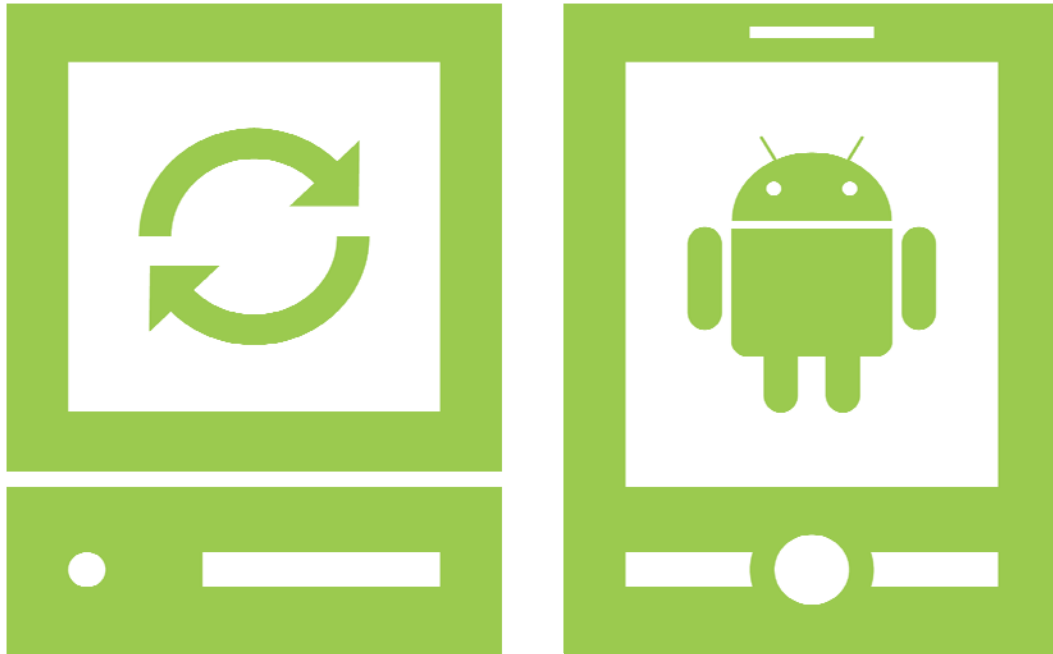
AnCoF Android Control Framework

Ba Bachelorarbeit

GUI Graphical User Interface

PC Personal Computer

TCA Task-Management Control-Center for Android



Zeiterfassung

«Task-Management Control-Center for Android»

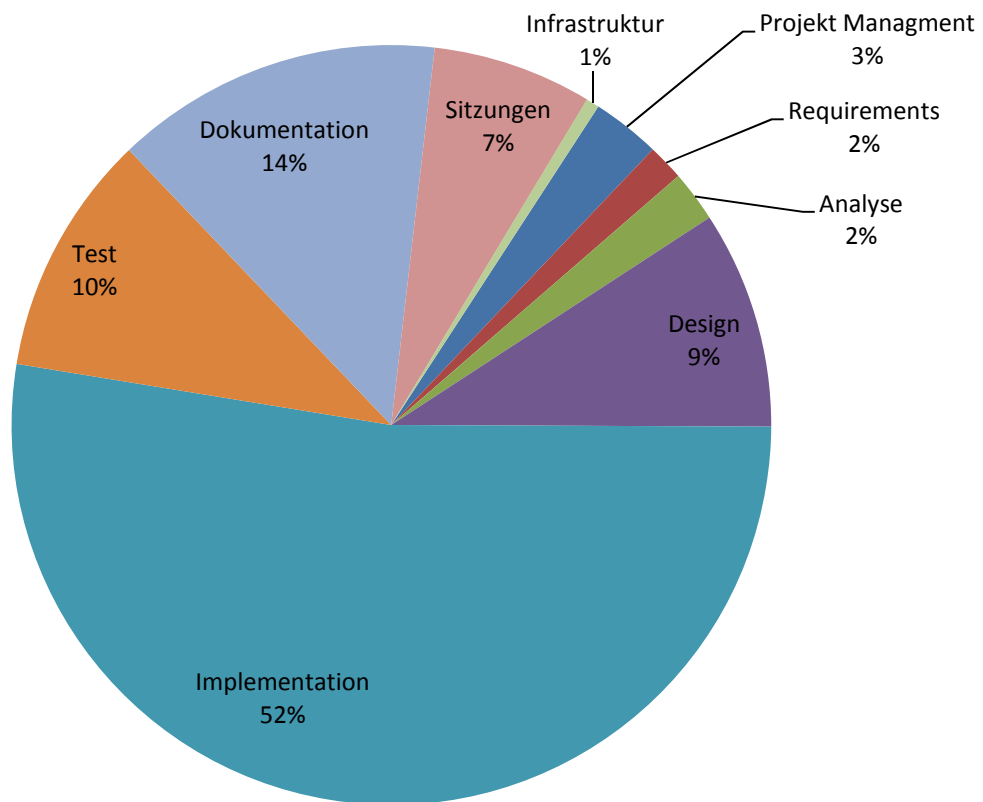
13.06.2011

Auswertung nach Aufgabe

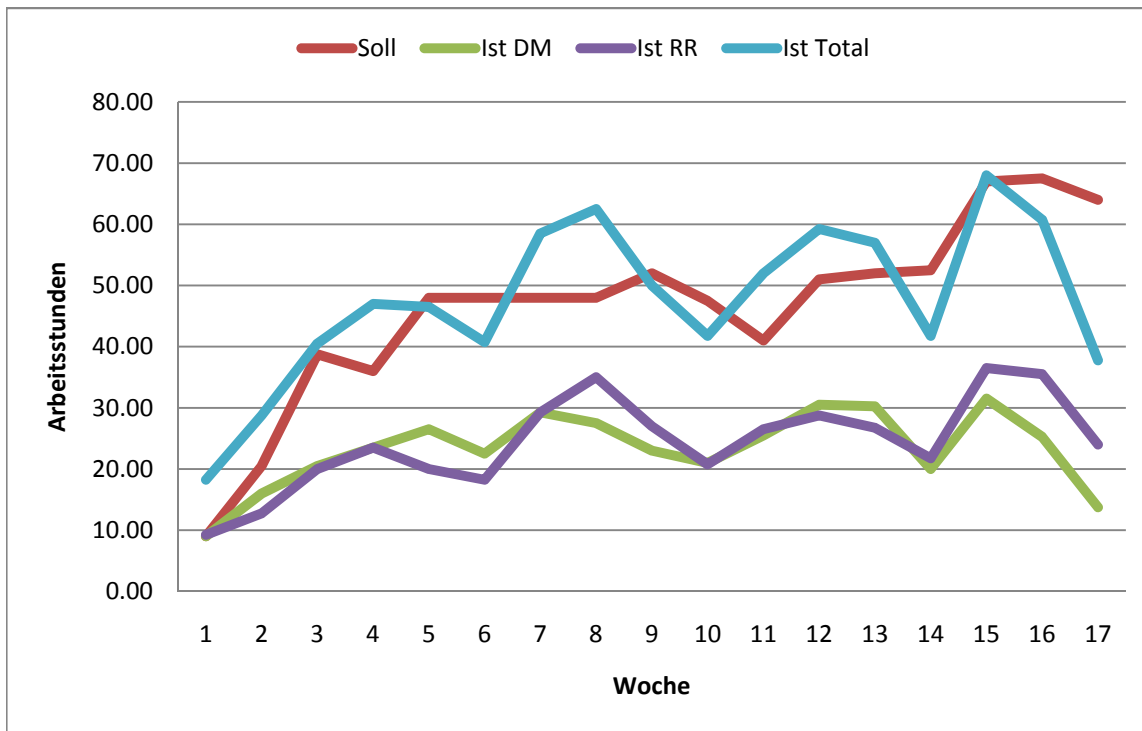
	Soll	Ist DM	Ist RR	Ist Total	Differenz	Differenz in %
Projekt Management	21.50	11.00	12.50	23.50	2.00	9.30
Vision	3.25	3.25	1.25	4.50	1.25	38.46
Projektplan	2.75	3.50	4.75	8.25	5.50	200.00
Zeitplanung	12.00	4.75	4.25	9.00	-3.00	-25.00
Risiko Management	1.75	0.50	0.50	1.00	-0.75	-42.86
Qualitätsmassnahmen	1.00	0.50	0.50	1.00	0.00	0.00
Arbeitspakete	3.00	0.50	1.00	1.50	-1.50	-50.00
Artefakte	1.00	1.25	1.50	2.75	1.75	175.00
Requirements	9.00	5.25	7.25	12.50	3.50	38.89
Use Case brief	4.00	3.00	3.50	6.50	2.50	62.50
Anforderungsspezifikation	3.50	2.25	2.25	4.50	1.00	28.57
UseCase Diagramm	1.50	0.00	1.50	1.50	0.00	0.00
Analyse	14.00	9.25	8.25	17.50	3.50	25.00
Domainanalyse	3.50	0.75	2.00	2.75	-0.75	-21.43
Domainmodell	5.00	4.25	4.25	8.50	3.50	70.00
Sequenzdiagramme	4.00	2.00	2.00	4.00	0.00	0.00
Operation Contracts	1.50	2.25	0.00	2.25	0.75	50.00
Design	63.00	25.25	49.50	74.75	11.75	18.65
Designmodell	7.00	0.75	2.00	2.75	-4.25	-60.71
Software Architecture Document	3.50	0.25	2.00	2.25	-1.25	-35.71
Design Dokument	24.00	11.75	26.00	37.75	13.75	57.29
GUI-Design	8.00	4.50	7.25	11.75	3.75	46.88
Paperprototype	20.50	8.00	12.25	20.25	-0.25	-1.22
Implementation	418.50	204.50	219.00	423.50	5.00	1.19
Prototypen	14.50	5.25	6.00	11.25	-3.25	-22.41
GUI-Prototyp PC	21.00	15.75	0.25	16.00	-5.00	-23.81
GUI-Prototyp Mobile	28.00	54.50	0.00	54.50	26.50	94.64
Architekturskelet	38.00	0.00	30.75	30.75	-7.25	-19.08
CRUD-Operationen Task	47.00	7.75	35.25	43.00	-4.00	-8.51
Dateisynchronisation	51.00	0.00	20.50	20.50	-30.50	-59.80
GUI-Mobile	72.75	97.00	1.75	98.75	26.00	35.74
GUI-PC	61.25	3.00	77.25	80.25	19.00	31.02
Tasksynchronisation	60.00	4.00	28.25	32.25	-27.75	-46.25
Allgemeine Arbeiten	4.00	3.25	5.00	8.25	4.25	106.25
Code Review	21.00	14.00	14.00	28.00	7.00	33.33
Test	68.00	54.25	28.75	83.00	15.00	22.06
Unit Tests	26.00	23.50	14.00	37.50	11.50	44.23
Systemtests	14.00	6.50	8.75	15.25	1.25	8.93
Bugfixing	28.00	24.25	6.00	30.25	2.25	8.04
Dokumentation	76.50	52.50	60.00	112.50	36.00	47.06
Titelblatt	1.00	1.50	1.00	2.50	1.50	150.00
Abstract	1.00	3.50	3.00	6.50	5.50	550.00
A0-Poster	4.00	2.75	3.00	5.75	1.75	43.75
Glossar	3.00	0.25	0.00	0.25	-2.75	-91.67
Literaturverzeichnis	0.50	0.75	0.00	0.75	0.25	50.00
User Guide	4.00	0.50	7.25	7.75	3.75	93.75
Technischer Bericht	2.00	2.75	0.00	2.75	0.75	37.50
Persönlicher Bericht	3.00	0.75	1.00	1.75	-1.25	-41.67
Präsentation	8.00	5.50	3.00	8.50	0.50	6.25
Allgemeine Bereinigung Dokumente	19.00	8.00	16.75	24.75	5.75	30.26

	Soll	Ist DM	Ist RR	Ist Total	Differenz	Differenz in %
Dokumentvorlagen erstellen	3.00	7.75	1.75	9.50	6.50	216.67
Javadoc	28.00	18.50	23.25	41.75	13.75	49.11
Sitzungen	85.00	27.75	27.00	54.75	-30.25	-35.59
Sitzungen mit Betreuer	51.00	17.25	17.25	34.50	-16.50	-32.35
Teamsitzungen	34.00	10.50	9.75	20.25	-13.75	-40.44
Infrastruktur	4.00	2.50	2.00	4.50	0.50	12.50
Arbeitsplatz einrichten	4.00	2.50	2.00	4.50	0.50	12.50
Rückstellungen	30.00	0.00	0.00	0.00	-30.00	-100.00
Rückstellungen	30.00	0.00	0.00	0.00	-30.00	-100.00
Total	789.50	392.25	414.25	806.50	17.00	2.15

Arbeitsaufwand effektiv



Auswertung nach Woche



Woche	Soll	Ist DM	Ist RR	Ist Total	Differenz	Differenz in %
0	9.00	9.00	9.25	18.25	9.25	102.78
1	20.50	16.00	12.75	28.75	8.25	40.24
2	38.75	20.50	20.00	40.50	1.75	4.52
3	36.00	23.50	23.50	47.00	11.00	30.56
4	48.00	26.50	20.00	46.50	-1.50	-3.13
5	48.00	22.50	18.25	40.75	-7.25	-15.10
6	48.00	29.25	29.25	58.50	10.50	21.88
7	48.00	27.50	35.00	62.50	14.50	30.21
8	52.00	23.00	27.00	50.00	-2.00	-3.85
9	47.50	21.00	20.75	41.75	-5.75	-12.11
10	41.00	25.50	26.50	52.00	11.00	26.83
11	51.00	30.50	28.75	59.25	8.25	16.18
12	52.00	30.25	26.75	57.00	5.00	9.62
13	52.50	20.00	21.75	41.75	-10.75	-20.48
14	67.00	31.50	36.50	68.00	1.00	1.49
15	67.50	25.25	35.50	60.75	-6.75	-10.00
16	64.00	13.75	24.00	37.75	-26.25	-41.02
Total	790.75	395.50	415.50	811.00	20.25	2.56