

Arbeits- und Projektzeiterfassung mit Windows 10 IoT und Microsoft Azure

Bachelorarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2016

Autoren:	Andreas Stefanits, Fabian Schäfer
Betreuer:	Prof. Hansjörg Huser, Jürg Jucker
Project Partner:	S3CC GmbH, Pfäffikon
Externer Experte:	Stefan Zettel, Ascentive AG, Zürich

Abstract

Ausgangslage

Das Ziel der vorliegenden Bachelorarbeit war es, einen Prototyp für ein Zeiterfassungssystem zu realisieren. Die grundlegende Architektur sollte dabei mit den folgenden Komponenten und Technologien umgesetzt werden:

- **Server:** Server- sowie Datenbanklogik basierend auf Microsoft Azure Cloud
- **Terminal-Client:** Zeiterfassung mittels Raspberry Pi Terminal mit Touch-Display, NFC Reader und Windows 10 IoT
- **Admin-Client:** Systemverwaltung mittels Admin-Client in Form einer Windows 10 Universal App

Vorgehensweise/Technologien

Die Arbeit wurde in eine Architektur-, eine Entwicklungs- und eine Testphase aufgeteilt.

Die Architekturphase begann mit der Erfassung und Analyse der Anforderungen. Im nächsten Schritt erfolgte die Evaluation der Technologien, wobei der Fokus auf der Ermittlung einer geeigneten Client-Server Schnittstelle lag. Diese sollte herstellerunabhängig, sinnvoll erweiterbar sowie einfach aufgebaut sein. Die Entscheidung fiel auf eine HTTP basierte REST-Schnittstelle, welche alle diese Kriterien erfüllt. Im letzten Schritt wurden, basierend auf den gesammelten Erkenntnissen, Domänenmodelle und Klassendiagramme für Client und Server entwickelt, sowie GUI Konzepte für die Clients erarbeitet.

Zu Beginn der Umsetzungsphase teilten wir uns auf, um die Entwicklung des Servers und Terminal-Clients parallel voranzutreiben. Dank der zuvor detailliert ausgearbeiteten Definition der REST-API, hat das Zusammenspiel zwischen Client und Server auf Anhieb funktioniert. Der im zweiten Teil der Umsetzungsphase entwickelte Admin-Client liess sich ebenfalls nahtlos in die REST-Schnittstelle integrieren.

In der letzten Phase, der Testphase, war die Ausarbeitung des Testplans die zentrale Aufgabe. Dieser diente dann als Grundlage für die Systemtests auf dem Terminal- und Admin-Client, um darüber die Funktionalität und Zuverlässigkeit des gesamten Systems zu testen. Ein weiteres wichtiges Testelement waren die Integrationstests, welche clientseitig mit automatisierten Tests über die REST-API die korrekte Funktionsweise des Servers überprüften.

Ergebnis

Das Ergebnis dieser Arbeit ist ein cloudbasiertes Zeiterfassungssystem mit einer Dokumentation der Architektur und der API, sowie einem Benutzerhandbuch.

Einige "Nice-to-have" Anforderungen, die den Rahmen dieser Arbeit gesprengt hätten, können dank der einfach erweiterbaren und gut dokumentierten REST-API mit wenig Aufwand auch nachträglich noch implementiert werden.

Deklaration

Wir erklären hiermit,

- dass wir die vorliegende Arbeit selber und ohne fremde Hilfe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde, durchgeführt haben.
- dass wir sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben haben.
- dass wir keine durch Copyright geschützten Materialien (z.B. Bilder) in dieser Arbeit in unerlaubter Weise genutzt haben.

Ort, Datum:

Rapperswil, 16. Juni 2016



Andreas Stefanits



Fabian Schäfer

Inhalt

1	Management Summary	1
1.1	Ausgangslage	1
1.2	Vorgehen und Technologien.....	1
1.3	Ergebnisse.....	1
1.4	Ausblick	1
2	Einleitung	2
2.1	Aufgabenstellung.....	2
3	Problemstellung	3
3.1	Funktionale Anforderungen	3
3.2	Spezifikationen	4
3.2.1	<i>Funktionalität</i>	4
3.2.2	<i>Sicherheit</i>	7
3.2.3	<i>Benutzerbarkeit</i>	7
3.2.4	<i>Änderbarkeit</i>	8
3.2.5	<i>Übertragbarkeit</i>	8
4	User Interface Design	9
4.1	Terminal App	9
4.1.1	<i>Einleitung</i>	9
4.1.2	<i>Anforderungen</i>	9
4.1.3	<i>Ablaufdiagramm</i>	10
4.1.4	<i>Mockups</i>	11
4.2	Admin App.....	12
4.2.1	<i>Einleitung</i>	12
4.2.2	<i>Anforderungen</i>	12
4.2.3	<i>Navigationsstruktur</i>	12
4.2.4	<i>Navigationselemente</i>	13
4.2.5	<i>Kommandoelemente</i>	13
4.2.6	<i>Mockups</i>	14
5	Software Architektur	15
5.1	Technologieevaluation	15

5.1.1	Server.....	15
5.1.2	Client	16
5.1.3	Client-Server Kommunikation	17
5.1.4	Near Field Communication (NFC).....	18
5.2	Software Layers.....	19
5.2.1	Data Access Layer.....	19
5.2.2	Domain Models	19
5.2.3	Business Logic	19
5.2.4	REST.....	19
5.2.5	Data Transfer Objects.....	19
5.3	Server Domain Model.....	20
5.3.1	Project.....	20
5.3.2	Employee	20
5.3.3	Position.....	21
5.3.4	TimeCategory	21
5.3.5	ProjectEmployee	21
5.3.6	PositionEmployee	21
5.3.7	PositionTimeCategory	21
5.3.8	TimeRecording	22
5.3.9	ProjectTimeRecording.....	22
5.3.10	Break.....	22
5.4	Admin-Client Domain Model.....	23
5.4.1	Unterschiede zu Server Domain Model.....	23
5.5	Terminal-Client	24
5.6	Server Klassendiagramm.....	25
5.6.1	DataAccess	25
5.6.2	DomainModel.....	25
5.6.3	DTOs.....	25
5.6.4	BusinessLogic	26
5.6.5	BackendWebApp.....	26
5.7	Clients Klassendiagramm.....	27
5.7.1	ClientApi	27

5.7.2	<i>DTOs</i>	27
5.7.3	<i>AdminClient</i>	27
5.7.4	<i>TerminalClient</i>	27
5.8	REST Definition	28
5.8.1	<i>CRUD HTTP Methoden</i>	28
5.8.2	<i>Report Ressourcen</i>	28
5.8.3	<i>Control Ressourcen</i>	29
5.8.4	<i>DTO Ressourcen</i>	30
5.8.5	<i>HTTP Status Codes</i>	32
5.9	Sicherheit.....	33
5.9.1	<i>Authentifizierung</i>	33
5.9.2	<i>Autorisierung</i>	33
6	Umsetzung	34
6.1	Technologien.....	34
6.2	Interface Definition.....	35
6.3	Server.....	38
6.3.1	<i>Aufbau</i>	38
6.3.2	<i>GenericDataRepository</i>	39
6.4	Admin-Client.....	40
6.4.1	<i>Aufbau</i>	40
6.4.2	<i>Bearbeitungsmodus</i>	41
6.5	Terminal-Client.....	42
6.5.1	<i>Aufbau</i>	42
6.5.2	<i>Sequenzdiagramm</i>	43
6.5.3	<i>Zustands-Management</i>	44
6.6	REST Client API.....	45
6.6.1	<i>ClientApi</i>	45
6.6.2	<i>RestHttpClient</i>	45
6.7	Raspberry Pi.....	46
6.7.1	<i>Inbetriebnahme</i>	46
6.7.2	<i>NFC Reader Konfiguration</i>	47
6.8	Testplan.....	48

6.8.1	<i>Admin-Client</i>	48
6.8.2	<i>Terminal-Client</i>	80
7	Zusammenfassung	85
7.1	Ergebnisse.....	85
7.2	Ausblick	86
7.2.1	<i>Erweiterungsmöglichkeiten</i>	86
7.2.2	<i>Verbesserungsvorschläge</i>	87
8	Referenzen	88
8.1	Quellen	88
8.2	Abbildungen.....	88
9	Danksagung	89
10	Anhang	90
10.1	Projektantrag.....	90
10.2	Persönliche Berichte	94
10.2.1	<i>Andreas Stefanits</i>	94
10.2.2	<i>Fabian Schäfer</i>	95
10.3	Projektplanung	96
10.3.1	<i>Projekt Zeitplan</i>	96
10.3.2	<i>Risiko Analyse</i>	98
10.4	Zeit Management	99
10.4.1	<i>Zeit pro Woche Andreas Stefanits</i>	99
10.4.2	<i>Zeit pro Woche Fabian Schäfer</i>	99
10.4.3	<i>Zeit pro Kategorie Andreas Stefanits</i>	100
10.4.4	<i>Zeit pro Kategorie Fabian Schäfer</i>	100
10.5	Sitzungsprotokolle	101
10.6	Benutzerhandbücher	116
10.6.1	<i>Terminal-Client</i>	116
10.6.2	<i>Admin-Client</i>	119
10.7	Testbericht.....	129
10.7.1	<i>Admin-Client</i>	129
10.7.2	<i>Terminal-Client</i>	133

1 Management Summary

1.1 Ausgangslage

Diese Bachelorarbeit befasst sich mit der Planung und Umsetzung eines Arbeits- und Projektzeiterfassungssystems für eine Firma mittlerer Grösse.

Den Mitarbeitern sollen fest installierte Terminals zur Verfügung stehen, an denen sie sich anmelden und ihr Kommen und Gehen erfassen können.

Die Verwaltung des Systems soll über eine Administrations Applikation erfolgen, die bei den verantwortlichen Personen auf dem PC installiert wird.

Alle dabei anfallenden Daten sollen zentral in der Cloud gespeichert werden, um nicht von der firmeninternen Infrastruktur abhängig zu sein.

1.2 Vorgehen und Technologien

Die Durchführung war in eine Architektur-, Umsetzungs- und Testphase untergliedert.

Im Zentrum der Architekturphase standen die Gespräche mit dem Auftraggeber um ein genaues Bild des gewünschten Endprodukts und dessen Funktionsumfang zu bekommen. Die dabei von uns entworfenen Konzepte und Modelle wurden in mehreren Sitzungen mit dem Auftraggeber überarbeitet und verfeinert, bis schliesslich die finale Architektur der Software definiert war.

Basierend auf dieser Software-Architektur entwickelten wir dann in der Umsetzungsphase die verschiedenen Komponenten des Systems. Das zentrale Element war die Serverseite mit der Datenbank, welche in Microsofts Azure Cloud aufgebaut wurde. Das Zeiterfassungsterminal wurde mit einem Raspberry Pi und Windows 10 IoT umgesetzt.

In der letzten Phase definierten wir zuerst den Testplan und prüften dann damit die Funktionalität des Systems. Auch die Dokumentation für den Auftraggeber wurde in der dritten Phase erstellt.

1.3 Ergebnisse

Alle in der Ausgangslage beschriebenen Komponenten haben wir im Rahmen dieser Arbeit entwickelt und getestet.

Die grundlegenden Funktionen des Zeiterfassungssystems sind gemäss den Anforderungen des Auftraggebers implementiert und können produktiv eingesetzt werden.

1.4 Ausblick

Einige "Nice to have" Funktionen wurden aus Zeitgründen im Rahmen dieser Arbeit nicht implementiert und können in Nachfolgeprojekten noch umgesetzt werden. Bei der Implementierung wurde stets auf die Erweiterbarkeit geachtet um eine spätere Erweiterung der Software so einfach wie möglich zu gestalten.

2 Einleitung

Der Auftraggeber dieser Bachelorarbeit ist die Softwareentwicklungs-Firma S3CC. Das Ziel des Projekts ist es, einen Prototypen zu bauen um für die Firma S3CC die Machbarkeit einer cloudbasierten Zeiterfassungslösung zu evaluieren.

2.1 Aufgabenstellung

Die Aufgabe dieser Bachelorarbeit ist es, ein Arbeits- und Projektzeiterfassungssystem für mittelgrosse Unternehmen zu implementieren. Zusätzlich soll evaluiert werden ob sich Microsoft Azure Cloud als Server und Windows 10 IoT Core als Betriebssystem für den Raspberry Pi Client eignen.

Die Mitarbeiter des Unternehmens sollen sich an fest installierten Terminal-Clients anmelden und ihre Arbeits- und Projektzeiten sowie Pausen erfassen können.

Für die Verwaltung der Mitarbeiter und Projekte soll zusätzlich ein Administrations Applikation erstellt werden, welche auf einem PC mit Windows 10 läuft.

Damit die Unternehmen keine eigene Server Infrastruktur zu Verfügung stellen müssen, sollen alle Daten zentral in der Cloud gespeichert werden.

Um zu beweisen, dass die entwickelten Funktionalitäten korrekt laufen soll ein Testplan für die Clients und Unit-Tests welche die Server-Schnittstelle testen erstellt werden.

Bei der ganzen Umsetzung soll das Augenmerk primär auf die technische Machbarkeit und weniger auf die Funktionalität gelegt werden.

3 Problemstellung

3.1 Funktionale Anforderungen

Die folgende Tabelle zeigt eine Liste der funktionalen Anforderungen und deren Priorität, die mit dem Auftraggeber vereinbart wurden:

Nr.	Anforderung	Prio	Zusätzliche Informationen
1	Der Benutzer kann seine Arbeitszeit erfassen.	1	Inklusive Pausenerfassung
2	Der Benutzer kann seine Arbeitszeit auf ein Projekt buchen.	1	
3	Der Benutzer muss sich in der Applikation anmelden.	1	NFC Tag oder Mail-Adresse
4	Der Benutzer soll in mindestens eine Berechtigungsgruppe sein.	1	Beispiele für Berechtigungsgruppen: <ul style="list-style-type: none">- Mitarbeiter- Administrator
5	Der Administrator soll neue Mitarbeiter erfassen und bearbeiten können	1	CRUD Operationen
6	Der Administrator soll neue Projekte erfassen und bearbeiten können.	1	CRUD Operationen
7	Der Administrator soll Zeitbuchungen nachträglich erfassen und bearbeiten können	1	
8	Der Administrator soll verschiedene Auswertungen erstellen können.	2	Beispiele für Auswertungen: <ul style="list-style-type: none">- Arbeitszeiten- Projektzeiten

3.2 Spezifikationen

In diesem Kapitel sind die Funktionalen und Nichtfunktionalen Anforderungen im Detail spezifiziert und kategorisiert.

3.2.1 Funktionalität

Name	Arbeitszeit erfassen	Prio	1	ID	FU 1	Anforderung	1
Beschreibung							
Der Benutzer kann seine Arbeitszeit erfassen indem die Start- und Endzeit erfasst werden.							

Name	Pausen erfassen	Prio	1	ID	FU 2	Anforderung	1
Beschreibung							
Der Benutzer kann seine Pausen erfassen indem die Start- und Endzeit erfasst werden.							

Name	Stunden auf ein Projekt buchen	Prio	1	ID	FU 3	Anforderung	2
Beschreibung							
Der Benutzer kann seine geleisteten Stunden einem Projekt zuweisen.							

Name	Anmeldung per NFC	Prio	1	ID	FU 4	Anforderung	3
Beschreibung							
Der Benutzer kann sich an einem fest installierten Terminal per NFC Karte anmelden.							

Name	Registrierung per Benutzer-ID	Prio	1	ID	FU 5	Anforderung	3
Beschreibung							
Der Benutzer kann die initiale Registrierung seiner NFC-Karte mithilfe seiner Benutzer-ID durchführen.							

Name	Anmeldung mit Benutzer/PW	Prio	1	ID	FU 6	Anforderung	3
Beschreibung							
Der Benutzer kann sich auf eigenen Geräten (nicht NFC Terminals) mit einem Benutzernamen und Passwort anmelden.							

3 Problemstellung

Name	Anmeldung Windows Live ID	Prio	3	ID	FU 7	Anforderung	3
Beschreibung							
Der Benutzer kann sich auf eigenen Geräten (nicht NFC Terminals) mit Windows Live ID anmelden.							

Name	Zuteilung zu Benutzergruppen	Prio	1	ID	FU 8	Anforderung	4
Beschreibung							
Der Benutzer wird einer Benutzergruppe zugewiesen um dessen Berechtigungen zu definieren.							

Name	Benutzergruppe "Mitarbeiter"	Prio	1	ID	FU 9	Anforderung	4
Beschreibung							
Die Benutzergruppe "Mitarbeiter" erlaubt es dem Benutzer seine Stunden zu erfassen.							

Name	Benutzergruppe "Administrator"	Prio	1	ID	FU 10	Anforderung	4
Beschreibung							
Die Benutzergruppe "Administrator" erlaubt es dem Benutzer administrative Tätigkeiten zu erledigen.							

Name	Neue Benutzer anlegen	Prio	1	ID	FU 11	Anforderung	4, 5
Beschreibung							
Die Benutzergruppe "Administrator" kann neue Benutzer anlegen.							

Name	Benutzer bearbeiten	Prio	1	ID	FU 12	Anforderung	4, 5
Beschreibung							
Die Benutzergruppe "Administrator" kann Benutzer bearbeiten.							

Name	Benutzer löschen	Prio	1	ID	FU 13	Anforderung	4, 5
Beschreibung							
Die Benutzergruppe "Administrator" kann Benutzer löschen.							

Name	Neue Projekte anlegen	Prio	1	ID	FU 14	Anforderung	4, 6
Beschreibung							
Die Benutzergruppe "Administrator" kann neue Projekte anlegen und Benutzer zuweisen.							

Name	Projekte bearbeiten	Prio	1	ID	FU 15	Anforderung	4, 6
Beschreibung							
Die Benutzergruppe "Administrator" kann Projekte bearbeiten.							

Name	Projekte löschen	Prio	1	ID	FU 16	Anforderung	4, 6
Beschreibung							
Die Benutzergruppe "Administrator" kann Projekte löschen.							

Name	Arbeitszeit manuell erfassen	Prio	1	ID	FU 17	Anforderung	4, 7
Beschreibung							
Die Benutzergruppe "Administrator" kann Arbeits- sowie Projektzeiten eines Benutzers manuell erfassen (auch nachträglich).							

Name	Arbeitszeit bearbeiten	Prio	1	ID	FU 18	Anforderung	4, 7
Beschreibung							
Die Benutzergruppe "Administrator" kann erfasste Arbeitszeiten eines Benutzers nachträglich bearbeiten.							

Name	Arbeitszeit Auswertung	Prio	2	ID	FU 19	Anforderung	4, 8
Beschreibung							
Die Benutzergruppe "Administrator" kann Auswertungen der Arbeitszeiten aller Benutzer erstellen.							

Name	Projektzeit Auswertung	Prio	2	ID	FU 20	Anforderung	4, 8
Beschreibung							
Die Benutzergruppe "Administrator" kann Auswertung der geleisteten Stunden pro Projekt erstellen.							

3.2.2 Sicherheit

Name	Passwortverschlüsselung	Prio	1	ID	SI 1	Anforderung	3
Beschreibung							
Alle Passwörter sind verschlüsselt zu übermitteln und zu speichern.							

3.2.3 Benutzerbarkeit

Name	Raspberry Pi mit NFC-Leser	Prio	1	ID	BE 1	Anforderung	1, 2, 3
Beschreibung							
Die Software soll auf einem Raspberry Pi Terminal welches mit einem NFC-Leser verbunden ist laufen.							

Name	Windows Phone und Tablet	Prio	2	ID	BE 2	Anforderung	1, 2, 3
Beschreibung							
Die Software soll auf Windows Phones und Tablets laufen.							

Name	Android Phone und Tablet	Prio	3	ID	BE 3	Anforderung	1, 2, 3
Beschreibung							
Die Software soll auf Android Phones und Tablets laufen.							

Name	iOS Phone und Tablet	Prio	3	ID	BE 4	Anforderung	1, 2, 3
Beschreibung							
Die Software soll auf iOS Phones und Tablets laufen.							

Name	Windows Universal App	Prio	1	ID	BE 5	Anforderung	1 - 8
Beschreibung							
Die Software soll per Windows Universal App verwendet werden können.							

Name	ASP.Net Webseite	Prio	4	ID	BE 6	Anforderung	1 - 8
Beschreibung							
Die Software soll als ASP.Net Webseite verwendet werden können.							

3.2.4 Änderbarkeit

Name	Möglichkeit für Erweiterungen	Prio	1	ID	AE 1	Anforderung	1 - 8
Beschreibung							
Die Architektur der Software soll ermöglichen effizient neue Funktionalitäten hinzuzufügen.							

3.2.5 Übertragbarkeit

Name	Shared Code	Prio	1	ID	UE 1	Anforderung	1 - 8
Beschreibung							
Die Architektur der Software soll ermöglichen gemeinsame Funktionalitäten mit dem gleichen Code Fragment zu verwenden.							

Name	Xamarin	Prio	3	ID	UE 2	Anforderung	1 - 8
Beschreibung							
Die verwendeten Technologien sollten so gewählt werden, damit diese auch mit dem Xamarin Framework lauffähig sind.							

4 User Interface Design

4.1 Terminal App

4.1.1 Einleitung

Die Terminal App ist ein zentrales Element des Arbeits- und Projektzeiterfassungssystem. Sie wird auf mehreren, fest installierten Terminals innerhalb der Firma laufen und den Mitarbeitern einen schnellen Zugriff auf ihre Zeiterfassung ermöglichen. Diese Terminals werden mit relativ kleinen 5" Bildschirmen ausgestattet sein.

4.1.2 Anforderungen

Bei der Entwicklung eines User Interfaces steht die Usability im Zentrum. Daher wurden folgende Anforderungen bei der Planung des GUIs berücksichtigt:

- Die GUI Elemente wie Buttons, Texte und Listen müssen so gross sein, dass sie gut lesbar sind und sich einfach antippen lassen
- Da das Display relativ klein ist, dürfen daher nicht zu viele Elemente pro Screen angezeigt werden, da diese sonst zu klein wären
- Die wichtigsten und am häufigsten genutzten Funktionen wie "Arbeitszeit starten", "Pause starten" und "Arbeitszeit beenden" müssen in maximal zwei Schritten (zweimal tippen) erreicht werden können
- Die Anordnung der Kontrollelemente muss über alle Screens hinweg einheitlich sein um Benutzern das Erlernen des Bedienkonzepts zu erleichtern
- Der Benutzer muss immer darüber Bescheid wissen, was sein aktueller Status ist, auf welchem Screen er sich gerade befindet und wohin er von da aus navigieren kann

4.1.3 Ablaufdiagramm

Basierend auf den zuvor genannten Anforderungen wurde folgendes Ablauf Diagramm erarbeitet. Die Terminal App verhält sich wie eine Zustandsmaschine, wobei die Zustandsübergänge (Pfeile) immer vom jeweiligen Zustand (Screen) abhängig sind.

Die Kästchen stellen dabei die verschiedenen Screens dar und die Pfeile zeigen Benutzer oder Systemaktionen auf. Bei den grünen Screens werden nur Informationen angezeigt, es ist keine Eingabe vom Benutzer erforderlich.

Zur vereinfachten Darstellung wurden Endpunkte eingefügt. Da das Terminal aber im Endlosbetrieb läuft, springt das Programm beim Erreichen eines Endpunktes automatisch wieder zum Anfang.

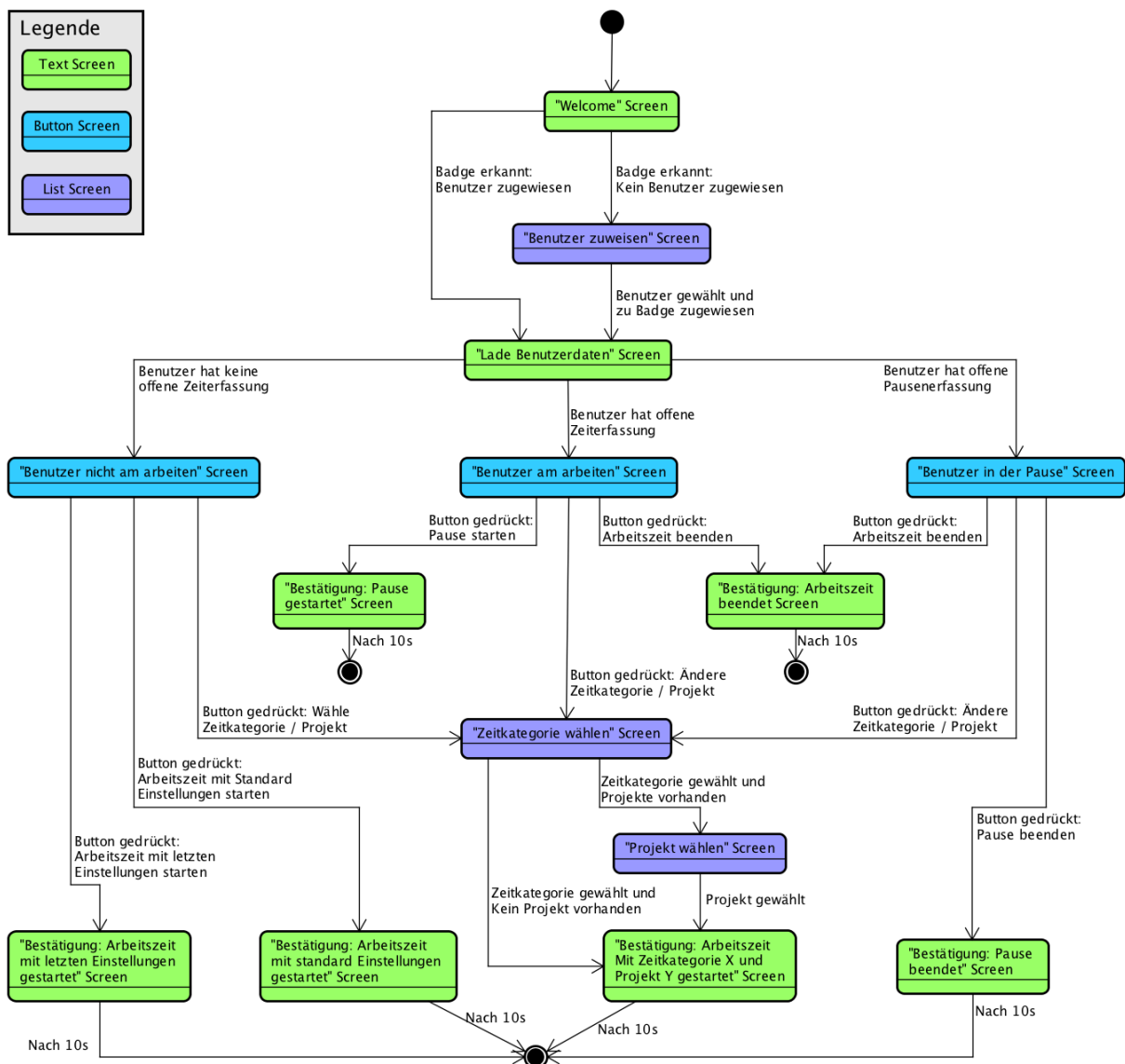


Abbildung 1: Ablaufdiagramm Terminal-Client

4.1.4 Mockups

Das gesamte Ablaufdiagramm haben wir mit GUI-Mockups umgesetzt. Da diese den Rahmen dieser Dokumentation sprengen würden, beschränkt sich die folgende Auflistung auf jeweils einen repräsentativen Mockup pro Screen-Typ.

Text Screen

Der nebenstehende Mockup zeigt einen Text Screen. In diesem Beispiel handelt es sich um den Text Screen, den man nach dem starten einer Zeiterfassung mit Projekt sieht.

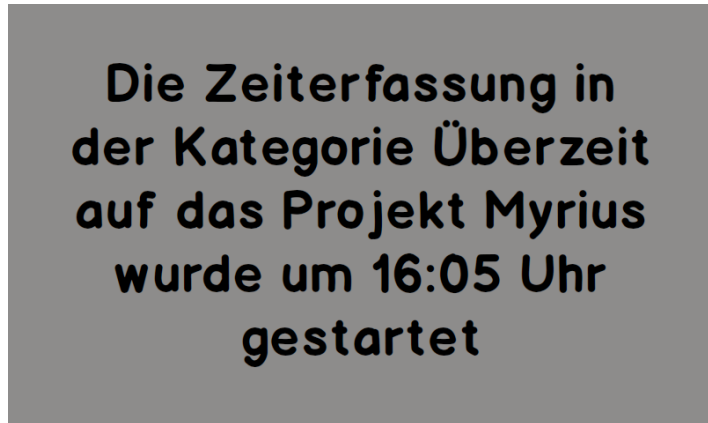


Abbildung 2: Terminal-Client Text Screen Mockup

Button Screen

Der nebenstehende Mockup zeigt einen Button Screen. In diesem Beispiel handelt es sich um den Button Screen, den man nach dem Anmelden mit aktiver Zeiterfassung sieht.

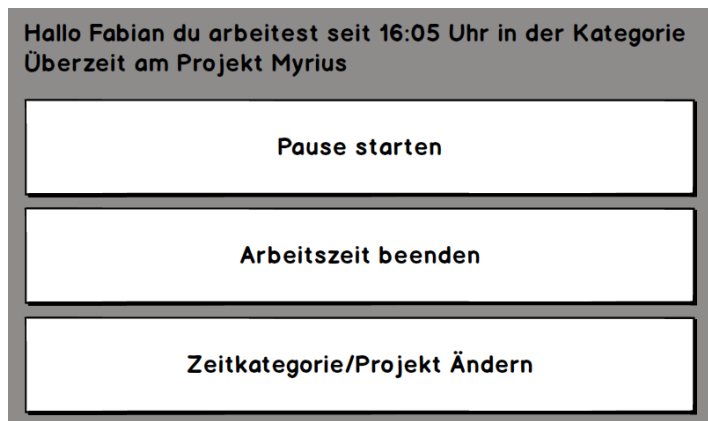


Abbildung 3: Terminal-Client Button Screen Mockup

List Screen

Der nebenstehende Mockup zeigt einen List Screen. In diesem Beispiel handelt es sich um den List Screen, den man beim Ändern der Zeitkategorie sieht.

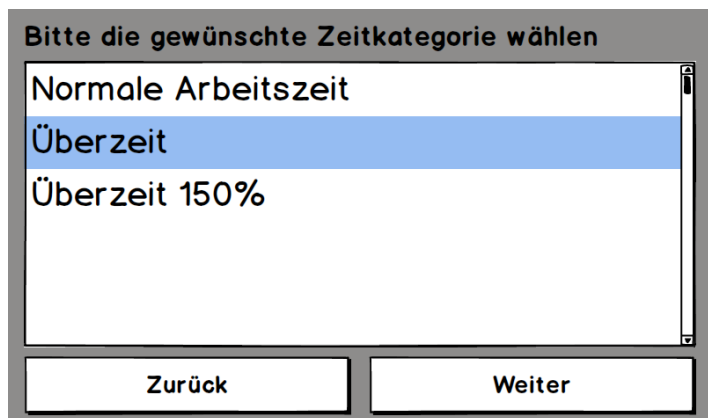


Abbildung 4: Terminal-Client List Screen Mockup

4.2 Admin App

4.2.1 Einleitung

Die Admin App dient primär der Erfassung der Stammdaten, welche dann von der Terminal App als Basis zur Zeiterfassung verwendet werden. Dazu ermöglicht sie die nachträgliche Manipulation von Arbeits- und Projektzeiten sowie Pausen. Die Applikation kommt auf den PCs bzw. Tablets der Administratoren zum Einsatz.

4.2.2 Anforderungen

Folgende Anforderungen wurden bei der Entwicklung der Admin App berücksichtigt:

- Die Navigationsstruktur soll schnell erfassbar und einheitlich aufgebaut sein
- Die Applikation muss auch auf Tablets laufen können, daher keine Pop-up Fenster
- Die gängigen Best Practices von Microsoft sollen verwendet werden

4.2.3 Navigationsstruktur

In der folgenden Tabelle werden die Ebenen der Navigationsstruktur, ausgehend von den vier Hauptelementen in der Ebene 1 visualisiert.

	Ebene 1	Ebene 2	Ebene 3	Ebene 4
👤	Mitarbeiter	Mitarbeiterauswahl	Mitarbeitereditor	Projektzuweisung
				Zeiterfassung
				Zeitauswertung*
📁	Projekte	Projektauswahl	Projekteditor	Mitarbeiterzuweisung
				Zeitauswertung*
☰	Positionen	Positionsauswahl	Positionseditor	Kategorienzuweisung
📅	Zeitkategorien	Kategorienauswahl	Kategorieneditor	Positionszuweisung
				Zeitauswertung*

*Diese Elemente wurden im Rahmen dieser Arbeit nicht implementiert

Das Element "Zeiterfassung" in der 4. Ebene des Mitarbeiters, ist aufgrund seiner Komplexität wiederum in mehrere Ebenen unterteilt.

Ebene Z1	Ebene Z2	Ebene Z3	Ebene Z4
Arbeitszeitauswahl	Arbeitszeiteditor	Pausenauswahl	Pauseneditor
		Projektzeitauswahl	Projektzeiteditor








4.2.4 Navigationselemente

Die Navigationselemente sind nach Microsoft Best Practices definiert. Die Vorgaben sowie detaillierte Erklärungen zu den verschiedenen Navigationselementen findet man auf der MSDN Webseite¹. In der nachfolgenden Tabelle ist definiert für welche Ebene welches Navigationselement verwendet wird.

Navigationselement	Ebene	Beschreibung	Begründung
Registerkarten und Pivots	1	Zeigt am oberen Rand dauerhaft eine Liste mit Links zu allen Seiten der Ebene 1 an	Ermöglicht das schnelle hin und her wechseln zwischen den Hauptebenen
Master/Detail	2/3 Z1/2 Z3/4	Zeigt auf der linken Seite eine Liste (Ebene 2, bzw. Z1 und Z3) und auf der rechten die Details des selektierten Objekts (Ebene 3, bzw. Z2 und Z4)	Es lässt sich schnell in der Liste das gewünschte Objekt finden und bearbeiten
Overlay	4	Zeigt die View als Overlay über den aktuellen Master/Detail View	Legt den Fokus auf die aktuelle View und blendet momentan unwichtiges aus

4.2.5 Kommandoelemente

Die Kommandoelemente dienen der Manipulation der Daten. In den Master/Detail Ansichten sind sie jeweils am unteren Rand in einer Command Bar angeordnet. Die Kommandoelemente sind gemäss Microsoft Best Practices auf der MSDN Webseite² definiert.

	Kommando	Ebene	Erklärung
	Filtern	Master	Öffnet ein Flyout mit Filteroptionen für die Liste
	Sortieren		Öffnet ein Flyout mit Sortieroption für die Liste
	Hinzufügen		Fügt einen neuen Datensatz in die Liste ein
	Entfernen		Entfernt einen Datensatz aus der Liste
	Bearbeiten	Detail	Aktiviert den Bearbeitungsmodus eines Datensatzes
	Speichern		Speichert die Änderungen an einem Datensatz
	Abbrechen		Verwirft die Änderungen an einem Datensatz

¹ Microsoft: "Navigation design basics for UWP apps", MSDN Online: <https://msdn.microsoft.com/en-us/windows/uwp/layout/navigation-basics>

² Microsoft: "Command design basics for UWP apps", MSDN Online: <https://msdn.microsoft.com/en-us/windows/uwp/layout/commanding-basics>

4.2.6 Mockups

Die gesamte Navigationsstruktur aus Kapitel 4.2.3 haben wir in GUI Mockups implementiert. Eine repräsentative Auswahl dieser Mockups sind hier gelistet.

Master/Detail View

Dieser Mockup zeigt die Master/Detail Ansicht der Mitarbeiter, welche repräsentativ für alle Master/Detail Ansichten in der Admin App steht.

Auf der linken Seite ist die Liste aller Mitarbeiter, sowie die Kontrollelemente zum Filtern, Sortieren, Hinzufügen und Löschen darunter. Auf der rechten Seite befindet sich die Detailansicht des gewählten Mitarbeiters sowie die Kontrollelemente zum Bearbeiten, Speichern und Abbrechen darunter.

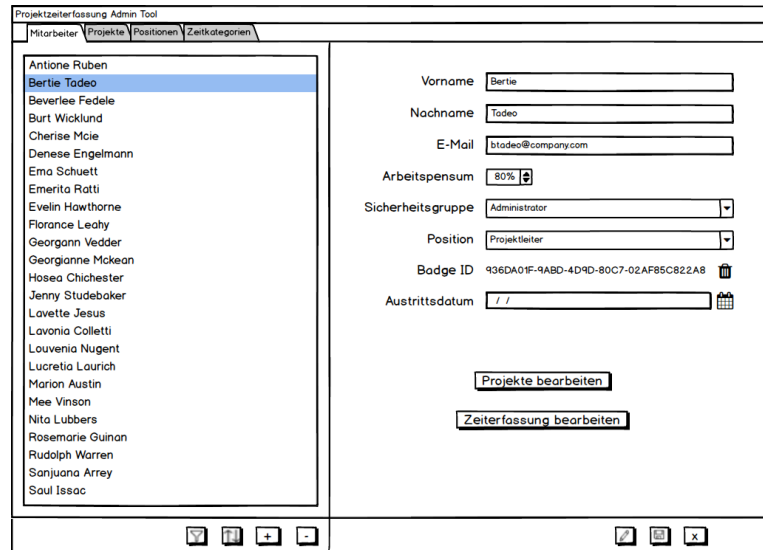


Abbildung 5: Admin-Client Master/Detail View Mockup

Overlay View

Dieser Mockup zeigt die "Zeitkategorie zu Position zuweisen" Overlay Ansicht, welche repräsentativ für alle Zuweisungs-Ansichten in der Admin App steht.

Die Ansicht liegt als Overlay über der aktuellen Position Master/Detail Ansicht und kann über den Pfeil in der oberen linken Ecke verlassen werden. In der oberen Hälfte sind alle nicht zugewiesenen und in der unteren Hälfte die zugewiesenen Zeitkategorien in Form von Kacheln ersichtlich. Über die zwei Knöpfe in der Mitte lassen sich Kategorien zuweisen und entfernen.

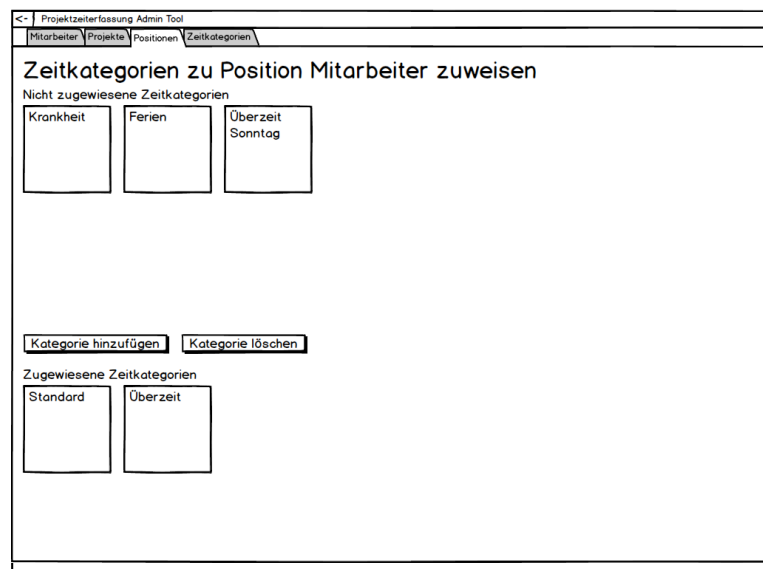


Abbildung 6: Admin-Client Overlay View Mockup

5 Software Architektur

5.1 Technologieevaluation

5.1.1 Server

Microsoft Azure

Im Projektantrag der Bachelorarbeit wurde definiert, dass der Server auf Microsoft Azure Cloud laufen soll. Als erstes wurde dieser Evaluert und geprüft ob Microsoft Azure geeignet ist als Server. Dazu wurden die folgenden Kriterien analysiert.

Kriterium	Kommentar	Ergebnis
Installierte .NET Version	Auf Microsoft Azure lassen sich folgende Versionen einstellen: v3.5 und v4.6	Für den entwickelten Server wird .NET v4.6.1 verwendet, Azure unterstützt dieses
SQL Sever	Microsoft Azure bietet nur MSSQL Server an, jedoch kann für jede Datenbank eine Skalierung vorgenommen werden	Dank Entity Framework lässt sich jede relationale Datenbank ansteuern, auch das skalieren ist von Vorteil
Integration in Visual Studio 2015	Microsoft Azure kann direkt als Server verbunden und konfiguriert werden, auch das Publizieren von Änderungen ist sehr einfach	Dank der guten Integration in Visual Studio 2015, kann alles über die IDE gesteuert werden
Scale-Up und Scale-Out	Microsoft Azure bietet die Möglichkeit für beide Technologien: <ul style="list-style-type: none"> - Für Scale-Up gibt es vom Gratis-Angebot bis zur Highend-Lösung alles. - Für Scale-Out können neue Instanzen dazu geschalten werden 	Für dieses Projekt kann das Gratis-Tier verwendet werden, da in der Testphase keine grosse Performance benötigt wird Im laufenden Betrieb kann der Server an die Bedürfnisse angepasst werden

Das Fazit ist, dass Microsoft Azure alle Funktionalitäten bietet, die für den Server-Teil dieses Projektes benötigt werden. Der Server und die Datenbank können aber auch selber auf einem IIS mit SQL-Server gehostet werden.

5.1.2 Client

Universal Windows App

Da der Terminal-Client auf verschiedenen Systemen laufen soll, wurde Universal Windows App (UWP), das neue Framework für GUI Entwicklung angeschaut. In der folgenden Tabelle sind die Vor- und Nachteile von UWP gegenüber WPF aufgelistet.

Vorteile UWP	Nachteile UWP
<ul style="list-style-type: none"> - Look and Feel von Windows 10 - Standardisiertes Design für die App - Läuft auf allen neuen Windows Geräten, wie Windows Phone, Windows 10 IoT Core und Windows 10 - Selber Code lässt sich auf verschiedenen Gerätetypen installieren - GUI Elemente sind für dynamisch verfügbare Bildschirmfläche ausgelegt - App lässt sich einfach entfernen ohne Spuren zu hinterlassen - GUI wird vorkompiliert und ist daher schneller 	<ul style="list-style-type: none"> - Läuft nur unter Windows 10 - Einige WPF Features verhalten sich anders (z.B. x:Bind) oder wurden ganz entfernt (z.B. x:Type bindings) - Noch nicht vollständig in Visual Studio integriert (der WYSIWYG Editor hat noch einige Fehler) - Fehlermeldungen in Visual Studio nicht immer klar wegen Vorkompilierung des GUIs - Durch Windows 10 Look and Feel auch viele Einschränkungen wie die App aussehen muss

Da ein Teil dieses Projektes ein Client für ein Raspberry Pi ist war schnell klar, dass auf die neue Technologie eingesetzt werden muss. Dies aufgrund dessen, dass WPF von Windows IoT nicht gut, UWP hingegen voll unterstützt wird und auch die Verteilung vereinfacht wird.

Wenn der Terminal-Client mit UWP erstellt wird und daher auch die gemeinsamen Libraries UWP kompatibel sein müssen, liegt es nahe auch für den Admin-Client UWP zu verwenden. Dies bringt auch den Vorteil, dass sich der gewohnte Windows 10 Style in der Applikation widerspiegelt.

5.1.3 Client-Server Kommunikation

In diesem Kapitel soll evaluiert werden ob für die Kommunikation zwischen Client und Server WCF oder REST verwendet werden soll. Hierfür werden die wichtigsten Unterschiede der beiden Technologien gegenübergestellt. Die folgende Tabelle wurde mit Hilfe der MSDN Webseite³ erstellt.

Kriterium	WCF	REST
Protokoll	Mehrere (TCP, HTTP, UDP und proprietäre Protokolle)	HTTP
Encoding	Text, MTOM, Binary	JSON, XML und weitere HTTP Media-Types
.NET Framework Integration	Integriert	Integriert als ASP.NET Web API
Kompatibilität	Kann nur mit .NET Framework verwendet werden	Kann mit allen Sprachen und Frameworks verwendet werden die REST unterstützen
Beschreibungssprache	WCF SOAP unterstützt WSDL das die Autogenerierung des Clients erlaubt	Selbstbeschreibend und braucht daher kein WSDL
Sicherheit	Unterstützt den WS-Security Standard	Verwendet Sessions und Cookies für Zugriffsschutz

Ein wichtiger Punkt welcher REST hervorhebt ist, dass der REST-Service auch mit anderen Sprachen und Frameworks verwendet werden kann. Das wäre mit dem WCF-Service nicht möglich, da dieser nur vom .NET-Framework unterstützt wird. Da der Server später jedoch auch von Android und iOS Geräten verwendet werden soll, kann ein reiner .NET-Framework Service nicht verwendet werden.

Aufgrund dieser Argumente wird das System mit einer REST-Schnittstelle entwickelt und verwendet. Als Encoding kommt JSON zum Einsatz, da dieses Format das kompakteste der oben genannten ist und somit am wenigsten Overhead erzeugt.

³ Microsoft: "WCF and ASP.NET Web API", MSDN Online: <https://msdn.microsoft.com/en-us/library/jj823172.aspx>

5.1.4 Near Field Communication (NFC)

Die Identifizierung der Benutzer am Zeiterfassungsterminal soll mittels NFC-Tags in Form von Smartcards erfolgen. Dazu wird die eindeutige ID der Smartcard in der Datenbank beim Benutzer hinterlegt. Es gibt folgende Varianten um eine Smartcard eindeutig zu identifizieren:

- **UID:** Jeder NFC-Tag besitzt eine unveränderbare eindeutige Hardware ID, die sogenannte UID (Unique Identifier). Diese UIDs sind jedoch vom Format her nicht einheitlich und können je nach Hersteller unterschiedlich aufgebaut sein.
- **NDEF:** Die meisten NFC-Tags (auch die mit denen wir in diesem Projekt arbeiten) unterstützen das NFC Data Exchange Format (NDEF), dieses erlaubt es uns beim Erstkontakt mit der Smartcard eine generierte GUID darauf zu schreiben und diese später wieder zu lesen.

Die zwei genannten Varianten habe beide Vor- und Nachteile die in folgender Tabelle veranschaulicht werden sollen.

	UID-Variante	NDEF-Variante
Vorteile	<ul style="list-style-type: none"> - Eindeutige UID ist von Werk gegeben und unveränderlich 	<ul style="list-style-type: none"> - Die NDEF Informationen können mit der vorhandenen Library ProximityDevice gelesen und geschrieben werden - Es können auch weitere Informationen (z.B. ob es ein Mitarbeiter oder Projekt Badge ist) darauf gespeichert werden - Die generierten GUIDs sind alle im selben Format
Nachteile	<ul style="list-style-type: none"> - Die UID kann mit der integrierten C# NFC Library (ProximityDevice) nicht ausgelesen werden - Die UIDs haben kein einheitliches Format 	<ul style="list-style-type: none"> - Die GUID muss beim Erstkontakt mit der Smartcard generiert und geschrieben werden

Aufgrund der überwiegenden Vorteile der NDEF-Variante haben wir uns für diese entschieden. Für das Generieren und Parsen von NDEF Records, bzw. Messages verwenden wir die NDEF Library⁴.

⁴ Jakl, Andreas: "NDEF Library for Proximity APIs / NFC", GitHub Online: <https://github.com/andijakl/ndef-nfc>

5.2 Software Layers

Das folgende Diagramm zeigt den Aufbau des Systems unterteilt in die Software Layers.

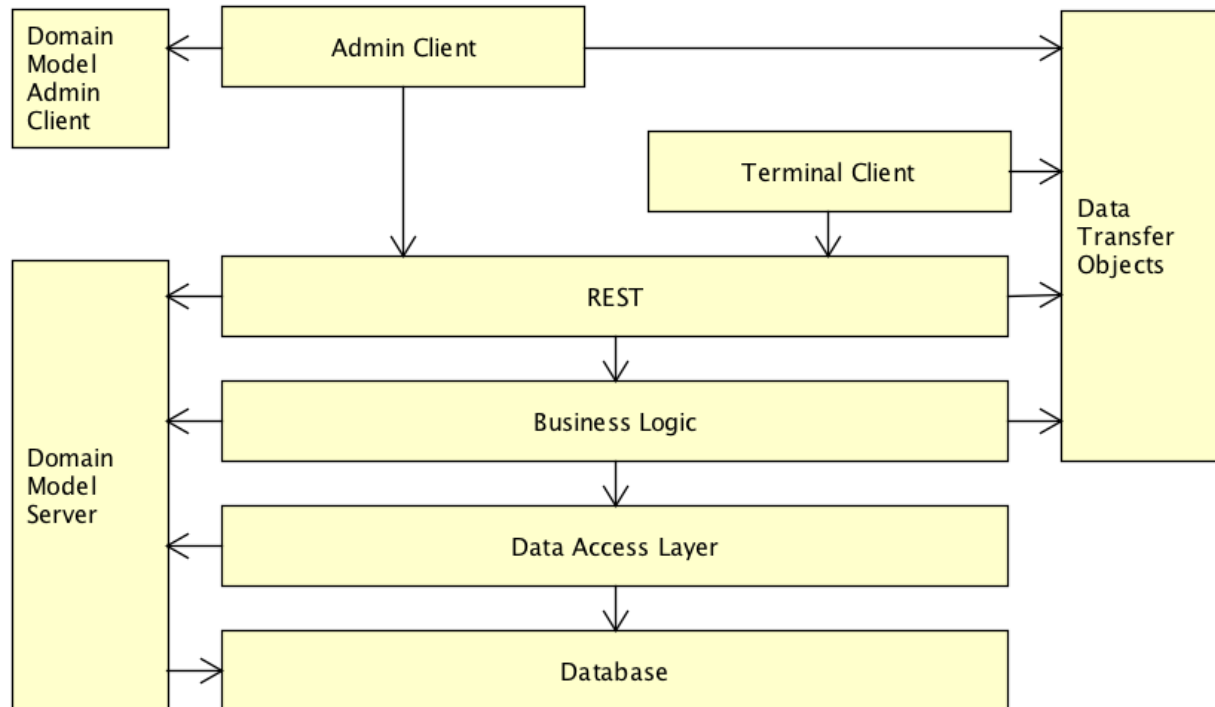


Abbildung 7: Software Layers

5.2.1 Data Access Layer

Der Data Access Layer (DAL) ist zuständig für alle Datenbankzugriffe. Durch den DAL werden direkten Datenbankzugriffe verhindert.

5.2.2 Domain Models

Die Domain Models sind eine Abbildung der Datenstruktur der Software. Diese werden im Kapitel 5.3 für den Server und Kapitel 5.4 für den Admin-Client genauer beschrieben.

5.2.3 Business Logic

Der Business Logic Layer ist für die Funktionalitäten zuständig welche für die REST-Schnittstelle benötigt werden. Dies sind in erster Linie die Implementation der Use-Cases der Software, wie zum Beispiel das Starten der Arbeitszeit. Dazu kommen die Konverter mit der Logik um zwischen DTOs und Domain Model zu konvertieren.

5.2.4 REST

Der REST Layer enthält alle Funktionalitäten welche über die REST-Schnittstelle verwendet werden können. Die genaue Definition der REST-Schnittstelle ist im Kapitel 5.8 ersichtlich.

5.2.5 Data Transfer Objects

Die Data Transfer Objects (DTOs) werden für den Austausch von Objekten zwischen den Clients und der REST-Schnittstelle verwendet (siehe Kapitel 5.8.4).

Die DTOs entsprechen den Klassen des Client Domain Models (siehe Kapitel 5.4).

5.3 Server Domain Model

Die folgende Abbildung zeigt das Domain Model das Serverseitig zum Einsatz kommt.

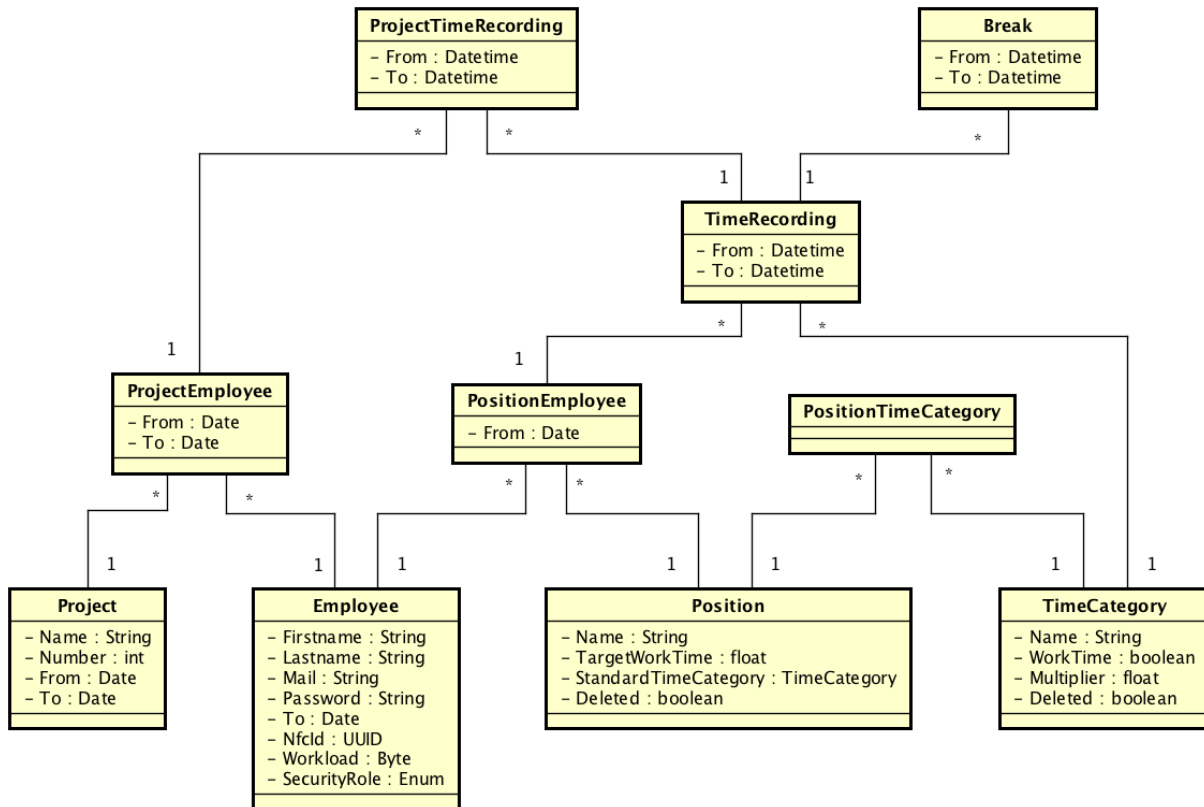


Abbildung 8: Server Domain Model

5.3.1 Project

Die Stammdatenklasse Project bildet die Projekte der Firma ab. Jedes Projekt verfügt über einen Projektnamen, eine Projektnummer sowie ein Anfangs- und Enddatum.

Ein Projekt gilt als gelöscht bzw. abgeschlossen, wenn das Enddatum in der Vergangenheit liegt.

5.3.2 Employee

Die Stammdatenklasse Employee bildet die Mitarbeiter der Firma ab. Jeder Mitarbeiter hat einen Namen, eine Mailadresse und ein definiertes Arbeitspensum sowie optional ein Austrittsdatum und eine NFC ID (siehe Kapitel 5.1.4).

Da die Employees auch zur Anmeldung ans System verwendet werden, besitzt jeder Employee ein Passwort sowie eine Sicherheitsrolle (Administrator, User oder Terminal). Das Passwort wird in Kombination mit der Mailadresse als Login verwendet.

Ein Mitarbeiter gilt als gelöscht bzw. ausgetreten, wenn das Austrittsdatum in der Vergangenheit liegt.

5.3.3 Position

Die Stammdatenklasse Position bildet die Positionen ab. Jede Position hat einen Namen (z.B. Projektleiter), eine wöchentliche Soll-Stundenzahl und eine Standard Zeitkategorie.

Eine Position kann nicht gelöscht werden, da sonst die darauf referenzierenden Zeiterfassungen auch gelöscht werden müssen. Aus diesem Grund gibt das "Gelöscht"-Feld darüber Auskunft ob eine Position noch aktiv ist.

5.3.4 TimeCategory

Die Stammdatenklasse TimeCategory bildet die Zeitkategorien ab. Eine Zeitkategorie hat einen Namen (z.B. Überzeit Sonntag) und einen Multiplikator (z.B. 150% bei Überzeit Sonntag), sowie ein Feld das angibt ob es sich um Arbeitszeitkategorie handelt oder nicht (z.B. wäre die Kategorie "Krankheit" keine Arbeitszeit).

Wie die Position kann auch eine Zeitkategorie nicht gelöscht werden. Daher gibt es auch hier des "Gelöscht"-Feld welches definiert ob die Kategorie noch aktiv ist.

5.3.5 ProjectEmployee

Die Klasse ProjectEmployee bildet die Beziehung zwischen Mitarbeitern und Projekten ab. Jeder Mitarbeiter kann in mehreren Projekten tätig sein und jedes Projekt kann mehrere Mitarbeiter beschäftigen.

Da die Mitarbeit an einem Projekt immer zeitlich beschränkt ist, ist in der Mitarbeiter-Projekt Beziehung ein Anfangs- und Enddatum definiert.

5.3.6 PositionEmployee

Die Klasse PositionEmployee bildet die Beziehung zwischen Mitarbeitern und Positionen ab. Jeder Mitarbeiter kann während seiner Anstellungsdauer in mehreren Positionen arbeiten und pro Position gibt es mehrere Mitarbeiter.

Da ein Mitarbeiter immer nur in einer Position aktiv sein kann, wird in der Position-Mitarbeiter Beziehung lediglich das Startdatum angegeben. Der Mitarbeiter ist dann nur in der Position aktiv tätig die das neuste Startdatum hat. Die älteren Zuweisungen werden nur zu Historisierungszwecken für die darauf referenzierenden Zeiterfassungen benötigt.

5.3.7 PositionTimeCategory

Die Klasse PositionTimeCategory bildet die Beziehung zwischen Positionen und Zeitkategorien ab. Jede Position kann mehrere Zeitkategorien haben und jede Zeitkategorie kann von mehreren Positionen verwendet werden.

Eine Position-Zeitkategorie Beziehung kann komplett gelöscht werden, da keine Referenzen darauf bestehen.

5.3.8 TimeRecording

Die Bewegungsdatenklasse TimeRecording bildet die Zeiterfassungen ab. Jede Zeiterfassung hat einen Start- und Endzeitpunkt.

Zeiterfassungen können gelöscht werden, jedoch werden die darauf referenzierenden Projektzeiterfassungen sowie Pausen mitgelöscht.

5.3.9 ProjectTimeRecording

Die Bewegungsdatenklasse ProjectTimeRecording bildet die Projektzeiterfassungen ab. Jede Projektzeiterfassung hat einen Start- und Endzeitpunkt, der innerhalb der Zeitspanne der referenzierten Zeiterfassung liegen muss.

Projektzeiterfassungen können ohne weiteres gelöscht werden.

5.3.10 Break

Die Bewegungsdatenklasse Break bildet die Pausen ab. Jede Pause hat einen Start- und Endzeitpunkt, der innerhalb der Zeitspanne der referenzierten Zeiterfassung liegen muss.

Pausen können ohne weiteres gelöscht werden.

5.4 Admin-Client Domain Model

Nachfolgende Abbildung zeigt das Domain Model des Admin-Clients. Dieses stimmt grösstenteils mit dem Domain Model des Servers überein, jedoch wurden einige, für den Client nicht relevante, Zwischentabellen weggelassen.

Die N:N und N:1 Beziehungen werden nicht direkt auf den Objekten abgebildet, sondern sind über die REST-Schnittstelle (siehe Kapitel 5.8) abfragbar. Die einzelnen Klassen entsprechen den DTOs (siehe Kapitel 5.2.5).

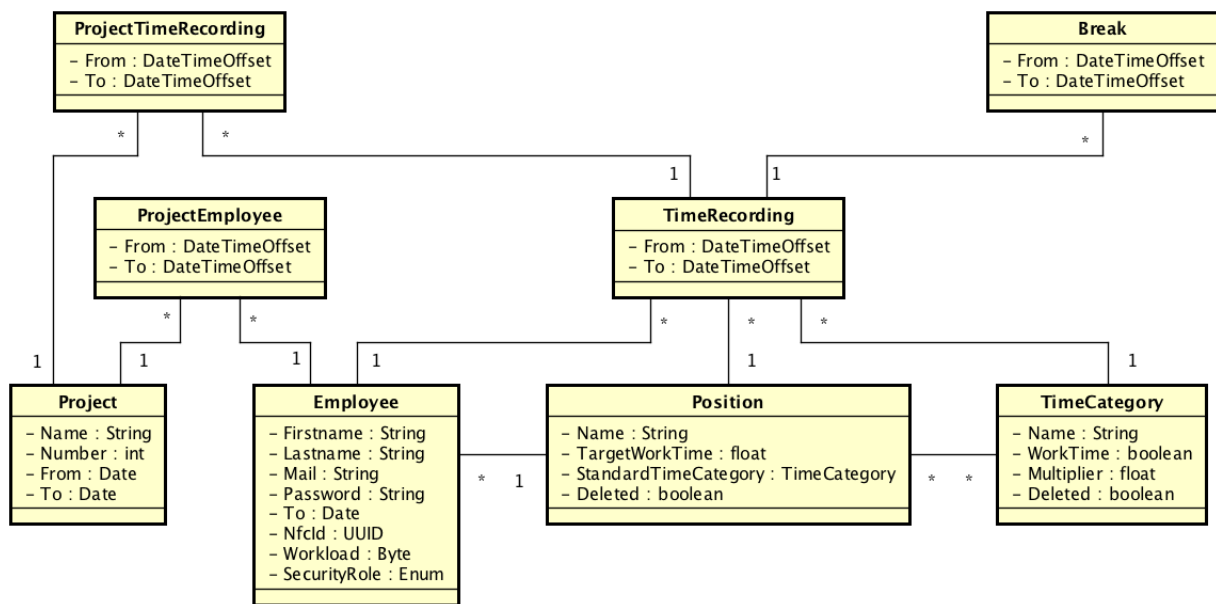


Abbildung 9: Admin-Client Domain Model

5.4.1 Unterschiede zu Server Domain Model

Das Domain Model des Admin-Clients unterscheidet sich in folgenden Punkten von dem des Server:

- **Fehlende PositionEmployee Tabelle**, da die Beziehung zwischen Mitarbeiter und Position (wie im Kapitel 5.3.6 bereits beschrieben) eigentlich eine 1:N Beziehung ist, die jedoch aus Historisierungsgründen Serverseitig als N:N betrachtet wird. Für den Client ist dies jedoch irrelevant, weshalb auf dem Mitarbeiter eine direkte Referenz auf die Position gemacht wird.
- **Fehlende PositionTimeCategory Tabelle**, da die Beziehung zwischen Position und TimeCategory auf dem Admin-Client ohne Zwischentabelle umgesetzt werden kann. Über die REST-Schnittstelle können die Zeitkategorien einer Position, bzw. die Positionen einer Zeitkategorie direkt abgefragt werden. Die Zuweisung funktioniert ebenfalls über eine Methode auf der REST-API (siehe Kapitel 5.8.3).

5.5 Terminal-Client

Der Terminal-Client entspricht in seiner Funktionsweise einer Zustandsmaschine und hat kein eigenes Domain Model, sondern bezieht lediglich gezielt Informationen daraus. Die verschiedenen Zustände sowie Zustandsübergänge des Terminal-Clients sind in Kapitel 4.1.3 ausführlich beschrieben.

Folgende Daten bezieht bzw. ändert der Terminal-Client auf dem Domain Model des Servers:

- Employee
 - Laden eines einzelnen Employees anhand der NFC ID, bei der Anmeldung mit Badge
 - Laden einer Liste aller Employees die keine NFC ID haben, bei Erkennung eines nicht zugewiesenen Badges
 - Änderung der NFC ID eines einzelnen Employee bei der Zuweisung eines neue Badges
- Projects
 - Laden einer Liste aller aktiven dem aktuellen Mitarbeiter zugewiesenen Projekte
- TimeCategories
 - Laden einer Liste aller aktiven dem aktuellen Mitarbeiter zugewiesenen Zeitkategorien

Alle anderen Manipulationen am Domain Model finden direkt auf dem Server statt und werden vom Terminal-Client über die Control Ressourcen der REST-API initiiert (siehe Kapitel 5.8.3).

5.6 Server Klassendiagramm

Die folgende Abbildung zeigt das vereinfachte Klassendiagramm welches für den Server verwendet wurde.

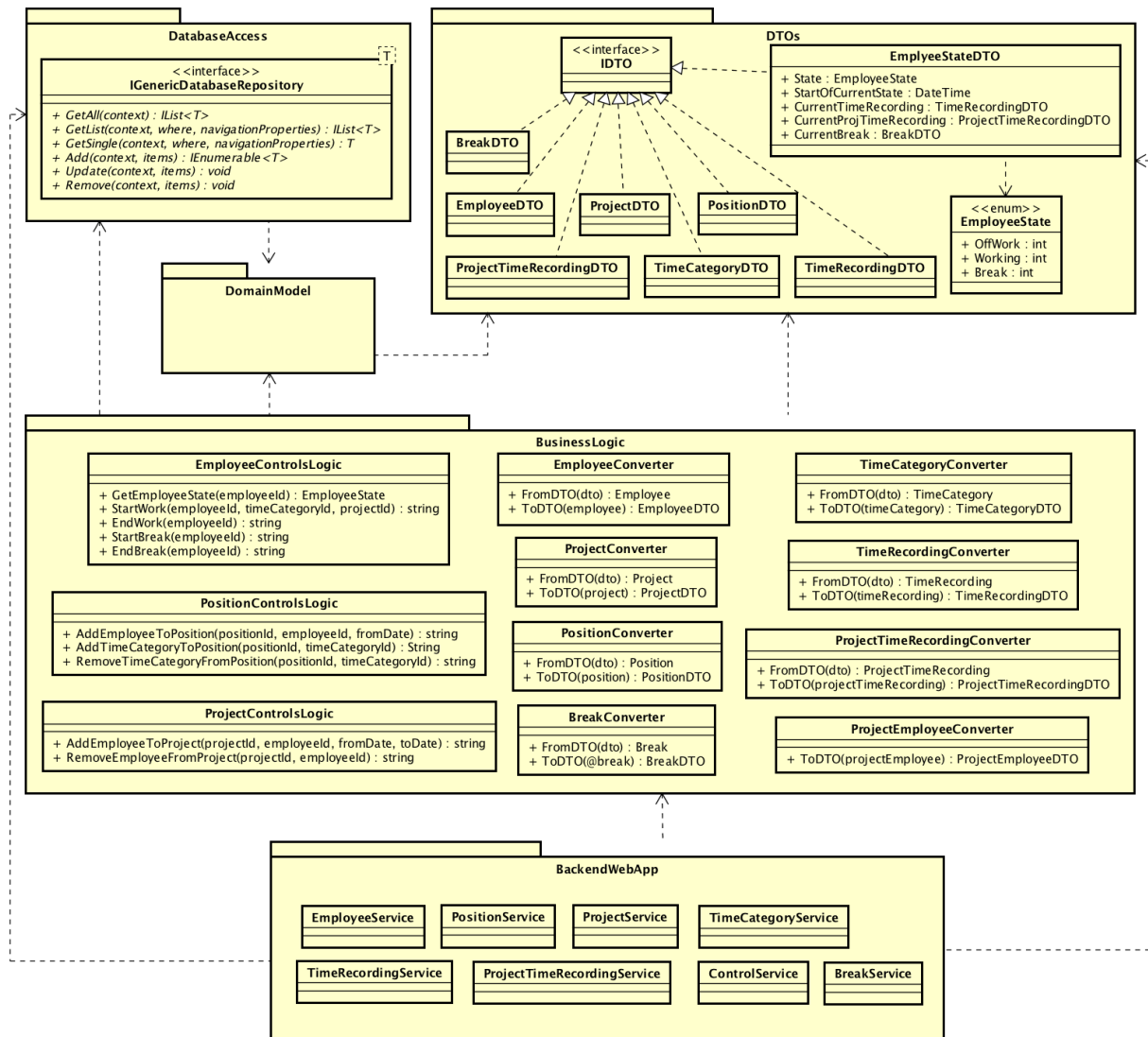


Abbildung 10: Klassendiagramm Server

5.6.1 DataAccess

Das Packet DataAccess enthält alle Komponente welche für das Entity Framework benötigt werden. Für jede Entität existiert ein Repository welches von IGenericRepository ableitet, in diesem Diagramm werden diese zur vereinfachten Darstellung nicht angezeigt.

5.6.2 DomainModel

Das DomainModel ist im Kapitel 5.3 genauer beschrieben.

5.6.3 DTOs

Im Packet DTOs sind alle Data Transfer Objects zu finden welche über die REST-Schnittstelle gesendet werden.

5.6.4 BusinessLogic

Die BusinessLogic enthält alle Konverter welche das DomainModel in das entsprechende DTO wandelt und umgekehrt. Zusätzlich ist einige Logik, welche für die Controls der REST-Schnittstelle benötigt werden in die BusinessLogic ausgelagert.

5.6.5 BackendWebApp

Im BackendWebApp ist die REST-Schnittstelle definiert und implementiert. Im Kapitel 5.8 sind die angebotenen Funktionen im Detail beschrieben.

5.7 Clients Klassendiagramm

Die folgende Abbildung zeigt das vereinfachte Klassendiagramm welches für die Clients verwendet wurde.

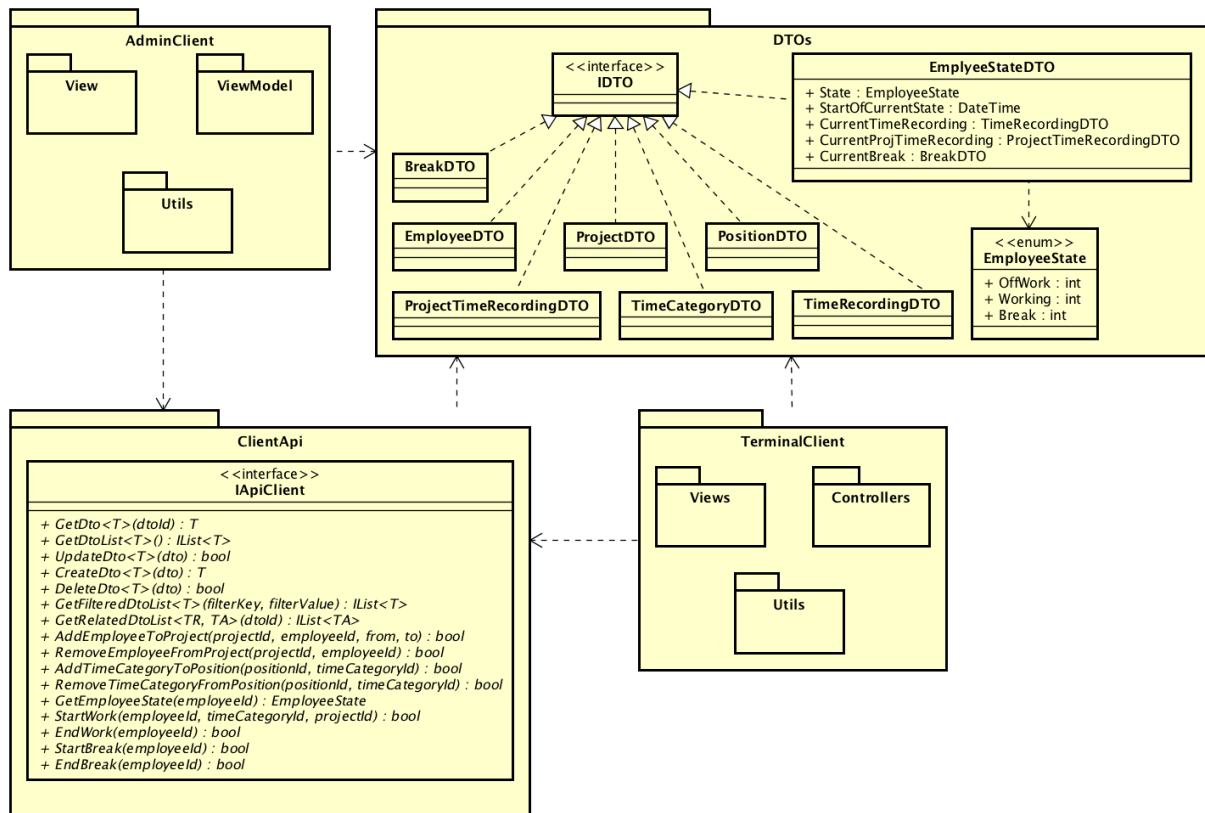


Abbildung 11: Klassendiagramm Clients

5.7.1 ClientApi

Im Packet ClientApi wird die Kommunikation mit der REST-Schnittstelle implementiert, über welche die Clients kommunizieren können. Das Interface IApiClient wird sehr generisch gehalten, damit eine Implementation für alle DTO Ressourcen verwendet werden kann. Die detaillierte Beschreibung der einzelnen Methoden ist im Kapitel 0 ersichtlich.

5.7.2 DTOs

Im Packet DTOs sind alle Data Transfer Objects zu finden welche über die REST-Schnittstelle ausgetauscht werden.

5.7.3 AdminClient

Der AdminClient wird mit dem MVVM-Ansatz implementiert, dadurch können die Views von der Logik getrennt werden.

5.7.4 TerminalClient

Der TerminalClient wird mit dem MVC-Ansatz implementiert, durch dies können viele Views für verschiedene Screens wiederverwendet werden.

5.8 REST Definition

Die Kommunikation zwischen Clients und Server erfolgt über eine HTTP REST-Schnittstelle. Bei der Planung der REST-Schnittstelle wurden drei grundlegende Ressourcen-Typen definiert:

- **Report:** Um serverseitige Auswertungen durchzuführen (z.B. Gesamte Arbeitszeit im aktuellen Monat)
- **Control:** Um serverseitige Abläufe zu starten (z.B. Pause beginnen) und konsolidierte Informationen (z.B. aktueller Status vom Mitarbeiter) vom Server abzufragen.
- **DTO:** Für die Manipulation (CRUD) einzelner Objekte oder Listen davon

5.8.1 CRUD HTTP Methoden

Im HTTP Standard sind für alle Arten von Anfragen Methoden (auch "Verben" genannt) vorgesehen. Diese stimmen jedoch nicht 1:1 mit den CRUD-Ausdrücken Create, Read, Update und Delete überein. Deshalb hier die Definition der Zuordnungen von CRUD und HTTP Verben.

CRUD	HTTP Verb	Anwendbar auf
Create	POST	- Ressourcen Listen z.B. /api/employees/
Read	GET	- Einzelne Ressourcen z.B. /api/employees/{id}/ - Ressourcen Listen z.B. /api/employees/ - Gefilterte Listen z.B. /api/employees?mail={mail} - Report Ressourcen (siehe Kapitel 5.8.2) - Control Ressourcen (siehe Kapitel 5.8.3)
Update	PUT	- Einzelne Ressourcen z.B. /api/employees/{id}/
Delete	DELETE	- Einzelne Ressourcen z.B. /api/employees/{id}/

5.8.2 Report Ressourcen

Die Report Ressourcen haben die Funktion, serverseitige Reports aus der Datenbank zu generieren und die Daten in konsolidierter Form an den Client zu übermitteln. Sie sind unter <https://{host}/api/reports/> angesiedelt und werden lediglich mit dem HTTP Verb GET verwendet. Die Report Ressourcen wurden im Rahmen dieser Arbeit nicht näher betrachtet, deshalb folgt hier lediglich eine kurze Auflistung mit möglichen Reports:

- Geleistete Arbeitsstunden eines Mitarbeiters in der letzten Woche
- Anzahl Arbeitsstunden die für ein Projekt seit Beginn geleistet wurden
- Anzahl Überstunden der ganzen Firma im aktuellen Monat

5.8.3 Control Ressourcen

Die Control Ressourcen haben die Funktion serverseitige Aktionen auszulösen. Sie sind alle unter <https://{host}/api/controls/> angesiedelt und werden lediglich mit dem HTTP Verb **GET** verwendet. Die Control Ressourcen sind für folgende Arten von Anfragen konzipiert:

- Wenn viele Daten aus der Datenbank verarbeiten müssen und nur wenig oder gar keinen Output generiert wird (z.B. Arbeitszeit beginnen) → Es wird kein unnötiger Netzwerkverkehr erzeugt da der DB Zugriff direkt auf dem Server stattfindet
- Um komplexe Operationen auf die Business Logik des Servers auszulagern (z.B. Mitarbeiter Status ermitteln) → Clients (insbesondere Terminal-Clients) können einfacher programmiert werden, wenn die Logik auf dem Server ist
- Um Relationen zwischen Ressourcen herzustellen, die nicht in DTOs abgebildet sind (z.B. Zeitkategorien zu Position zuweisen) → Die Clients müssen bei gewissen N:N Relationen nur die referenzierten Ressourcen sehen, jedoch nicht direkt die Zwischentabelle manipulieren können (z.B. `/api/positions/{id}/timecategories`)

In der nachfolgenden Tabelle werden die Control Ressourcen beschrieben (die Klammern werden durch Werte ersetzt).

Ressource (unter <code>/api/controls/</code>)	GET Parameter	Beschreibung
<code>projects/{id}/addemployee</code>	- fromdate - todate	Mitarbeiter zu Projekt hinzufügen
<code>projects/{id}/removeemployee</code>		Mitarbeiter von Projekt entfernen
<code>positions/{id}/addtimecategory</code>		Zeitkategorie zu Position hinzufügen
<code>positions/{id}/removetimecategory</code>		Zeitkategorie von Position entfernen
<code>employees/{id}/state</code>		Gibt den Status als EmployeeStateDTO zurück
<code>employees/{id}/startwork</code>	- timecategoryid - (projectid)	Arbeitszeit (und Projektzeit) starten (kann auch zum Kategorie und/oder Projekt ändern aufgerufen werden)
<code>employees/{id}/endwork</code>		Alle offenen Arbeits- und Projektzeiten sowie Pausen beenden
<code>employees/{id}/startbreak</code>		Zeiterfassung pausieren
<code>employees/{id}/endbreak</code>		Zeiterfassung fortsetzen

5.8.4 DTO Ressourcen

Die DTO Ressourcen sind die klassischen REST Ressourcen und haben die Funktion Serverseitige Objekte zu manipulieren. Jede DTO Ressource ist direkt unter <https://{host}/api/{ressource}/> angesiedelt und kann mit allen CRUD Operationen verwendet werden.

In der nachfolgenden Tabelle werden die DTO Ressourcen sowie Subressourcen der vier Stammdatenklassen beschrieben (die Klammern werden durch Werte ersetzt):

Ressource	Subressource	Beschreibung
projects	/	Alle Projekte
	{id}	Einzelnes Projekt
	{id}/employees/	Alle aktuellen Mitarbeiter eines Projekts
	{id}/projectemployees/	Alle Mitarbeiterzuweisungen eines Projekts (inkl. beendete/zukünftige)
	{id}/projecttimerecordings/	Alle Projektzeiterfassungen eines Projekts
employees	/	Alle Mitarbeiter
	{id}	Einzelner Mitarbeiter
	{id}/projects/	Alle aktuellen Projekte eines Mitarbeiters
	{id}/projectemployees/	Alle Projektzuweisungen eines Mitarbeiters (inkl. ausgetretene)
	{id}/timecategories/	Alle Zeitkategorien eines Mitarbeiters
	{id}/timerecordings/	Alle Zeiterfassungen eines Mitarbeiters
positions	/	Alle Positionen
	{id}	Einzelne Position
	{id}/employees/	Alle Mitarbeiter einer Position
	{id}/timecategories/	Alle Zeitkategorien einer Position
timecategories	/	Alle Zeitkategorien
	{id}	Einzelne Zeitkategorie
	{id}/positions/	Alle Positionen einer Zeitkategorie
	{id}/timerecordings/	Alle Zeiterfassungen einer Zeitkategorie

Bei einigen Ressourcen aus der vorhergehenden Tabelle können GET Parameter mitgegeben werden um eine gefilterte Liste der Ressourcen zu erhalten:

Ressource	GET Parameter	Beschreibung
projects	number	Projekte nach Projektnummer filtern
employees	mail	Mitarbeiter nach Mailadresse filtern
	nfcid	Mitarbeiter nach NFC ID filtern

In der nachfolgenden Tabelle werden die DTO Ressourcen sowie Subressourcen der drei Bewegungsdatenklassen beschrieben (die Klammern werden durch Werte ersetzt):

Ressource	Subressource	Beschreibung
timerecordings	/	Alle Zeiterfassungen
	{id}	Einzelne Zeiterfassung
	{id}/projecttimerecordings /	Alle Projektzeiterfassungen einer Zeiterfassung
	{id}/breaks/	Alle Pausen einer Zeiterfassung
projecttimerecordings	/	Alle Projektzeiterfassungen
	{id}	Einzelne Projektzeiterfassung
breaks	/	Alle Pausen
	{id}	Einzelne Pause

5.8.5 HTTP Status Codes

Im Folgenden werden die HTTP Status Codes, welche für die Server-Responses definiert wurden, beschrieben. Die Status Codes geben darüber Auskunft wie ein HTTP Request vom REST Server verarbeitet wurde und ob Fehler aufgetreten sind.

HTTP Status Code	HTTP Body	Beschreibung
200 OK	Die angeforderte Ressource, bzw. Liste von Ressourcen	Der Request wurde erfolgreich durchgeführt
201 Created	Die erstellte Ressource	Eine Ressource wurde erfolgreich erstellt (gibt es nur bei POST)
204 No Content	-	Der Request wurde erfolgreich durchgeführt
400 Bad Request	Message mit Detaillierter Fehlerbeschreibung	Der Request enthält ungültige Parameter
401 Unauthorized	Message, dass der Zugriff auf diese Ressource nicht erlaubt ist	Der Request enthält keinen gültigen WWW-Authenticate header
404 Not Found	-	Die Ressource im Request existiert nicht
405 Method Not Allowed	Message, dass die Methode nicht erlaubt ist	Die Methode (z.B. PUT) ist auf dieser Ressource nicht verfügbar
500 Internal Server Error	Message mit Detaillierter Fehlerbeschreibung	Der Server hat einen internen Fehler der nichts mit dem Request zu tun hat

5.9 Sicherheit

5.9.1 Authentifizierung

Für die Authentifizierung wird HTTP Basic Authentication verwendet. Dies eine der einfachsten, jedoch nicht die sicherste Variante, da Benutzername und Passwort in Base64 kodiert (reversibel) mitgesendet werden. Aus diesem Grund empfiehlt es sich den Server mit TLS (HTTPS) laufen zu lassen, damit Clientverbindungen verschlüsselt werden.

Der Client schickt mit jedem Request das kodierte Login im HTTP **WWW-Authenticate** Header mit. Falls der Server eine Anfrage ohne gültiges Login im Header erhält, antwortet er mit dem Statuscode **401 Unauthorized**.

Die Mailadresse eines Mitarbeiters ist gleichzeitig sein Benutzername. Das Login wird vom Server gegenüber dem **Mail** und dem **Password** Feld in der **Employee** Datenbank geprüft.

5.9.2 Autorisierung

Die Autorisierung geschieht basierend auf der dem Mitarbeiter zugewiesenen Sicherheitsrolle (siehe Kapitel 5.3.2). In der folgenden Tabelle sind die jeweiligen Berechtigungen einer Rolle auf die verschiedenen Ressourcen definiert.

Rolle	DTO Ressourcen (Rechte)	Control Ressourcen (Rechte)
Administrator	Alle (Read/Write)	Alle (Read/Write)
User	Alle mit eigenem Employee referenzierten (Read)*	Alle Subressourcen des eigenen Employee (Read/Write)*
Terminal	Alle Employees (Read/Write) mit Subressourcen (Read)*	Alle Employees mit Subressourcen (Read/Write)*

**Die effektive Umsetzung wurde mit Read Rechten auf alle Ressourcen vereinfacht*

6 Umsetzung

6.1 Technologien

Da die Verwendung von Windows 10 IoT Core und Microsoft Azure Cloud für dieses Projekt vorgegeben ist, waren die Technologien schnell gewählt. Die folgende Liste zeigt alle Technologien welche verwendet wurden.

Microsoft Visual Studio 2015 mit Microsoft .NET 4.6.1 und C# 6.0

Um das Projekt umzusetzen haben wir Microsoft Visual Studio 2015 Enterprise mit .NET 4.6.1, welche den C# Standard 6.0 enthält, verwendet.

Windows SDK 10

Um den Admin-Client und Terminal-Client zu erstellen wurde das Windows SDK 10 verwendet. Dieses wird benötigt um Universal Apps für Windows 10 und Windows 10 IoT Core zu schreiben.

MSTest

Um die REST-Schnittstelle zu testen haben wir Unit-Tests mit MSTest geschrieben.

Microsoft Azure Cloud

Als Backend und Datenbank Server wurde in diesem Projekt Microsoft Azure verwendet. Dies bietet eine Cloudbasierte Lösung, welche schnell aufgesetzt und verwendet werden kann. Auch ist die Integration in Microsoft Visual Studio 2015 sehr gut und Änderungen lassen sich schnell publizieren.

ASP.Net Web API

Für die REST-Schnittstelle wurde die ASP.NET Web API verwendet. Mit diesem Framework lassen sich einfach REST-Schnittstellen erstellen und verwenden.

Entity Framework 6.0

Für den Datenbankzugriff und das Transaktionsmanagement wurde Entity Framework mit Code-First Ansatz verwendet.

Visual Studio Online

Als Source-Control für dieses Projekt wurde Visual Studio Online verwendet. Dieses kann gratis von bis zu 5 Benutzern verwendet werden und war daher gut geeignet für dieses Projekt.

6.2 Interface Definition

Im Folgenden werden die public Methoden der wichtigsten server- und clientseitigen Interfaces der Applikation beschrieben.

IGenericRepository

Das Interface IGenericRepository ist im Projekt DataAccess zu finden und definiert die Standard Methoden für den Zugriff auf die Entitäten in der Datenbank.

Name	Return	Parameter	Beschreibung
GetAll	IList<T>	Entities	Liest alle Entitäten vom Typ T
GetList	IList<T>	Entities, Func<T, bool>, params Expression<Func<T, object>>[]	Liest alle Entitäten vom Typ T, welche dem Filter entsprechen
GetSingle	T	Entities, Func<T, bool>, params Expression<Func<T, object>>[]	Liest genau eine Entität vom Typ T, welche dem Filter entspricht
Add	IEnumerable<T>	Entities, T[]	Speichert die gegebenen Entitäten in der Datenbank
Update	void	Entities, T[]	Updatet die gegebenen Entitä- ten in der Datenbank
Remove	void	Entities, T[]	Entfernt die gegebenen Entitä- ten aus der Datenbank

ITimeRecordingRepository

Das Interface ITimeRecordingRepository ist im Projekt DataAccess zu finden und ist für den Zugriff auf die Entität TimeRecording zuständig.

Name	Return	Parameter	Beschreibung
UpdateFK	Void	Entities, Posi- tionEmploy- ee, TimeCategory , params TimeRecording[]	Updatet die übergebenen PositionEmployee und Time- Category Foreign Keys von den übergebenen TimeRe- cordings.

IPositionRepository

Das Interface IPositionRepository ist im Projekt DataAccess zu finden und ist für den Zugriff auf die Entität Position zuständig.

Name	Return	Parameter	Beschreibung
UpdateFK	Void	Entities, TimeCategory, params Position[]	Updatet den übergebenen TimeCategory Foreign Key von den übergebenen Positionen

IProjectTimeRecordingRepository

Das Interface IProjectTimeRecordingRepository ist im Projekt DataAccess zu finden und ist für den Zugriff auf die Entität ProjectTimeRecording zuständig.

Name	Return	Parameter	Beschreibung
UpdateFK	Void	Entities, ProjectEmployee, params ProjectTimeRecording[]	Updatet den übergebenen ProjectEmployee Foreign Key von den übergebenen ProjectTimeRecordings

IClientApi

Das Interface IApiClient ist im Projekt ClientApi zu finden und definiert die Methoden für den Zugriff auf die REST-Schnittstelle über die Client API.

Name	Return	Parameter	Beschreibung
GetDto<T>	T	int	Fragt das DTO für den gegebenen Typ T welches die übergebene ID hat vom Server ab
GetDtoList<T>	IList<T>	void	Fragt alle DTOs für den gegebenen Typ T vom Server ab
UpdateDto<T>	bool	T	Updatet das übergebene DTO auf dem Server und gibt den Status zurück
CreateDto<T>	T	T	Erstellt das übergebene DTO auf dem Server und gibt das erstellte DTO zurück.
DeleteDto<T>	bool	Int	Löscht das übergebene DTO auf dem Server und gibt den Status zurück

Name	Return	Parameter	Beschreibung
GetFilteredDtoList<T>	IList<T>	string, string	Fragt alle DTOs für den gegebenen Typ T welche dem Filter entsprechen vom Server ab
GetRelatedDtoList<TR,TA>	IList<TA>	int	Fragt die zugehörigen DTOs (Typ TR) von dem DTO (Typ TA) mit der übergebenen ID vom Server ab
AddEmployeeToProject	bool	int, int, DateTimeOffset, DateTimeOffset	Fügt den Mitarbeiter mit der übergebenen ID dem übergebenen Projekt hinzu. Optional kann ein Von- und Bis-Datum gesetzt werden.
RemoveEmployeeFromProject	bool	int, int	Entfernt den Mitarbeiter mit der übergebenen ID vom übergebenen Projekt.
AddTimeCategoryToPosition	bool	int, int	Fügt die Zeitkategorie mit der übergebenen ID der übergebenen Position hinzu.
RemoveTimeCategoryFromPosition	bool	int, int	Entfernt die Zeitkategorie mit der übergebenen ID von der übergebenen Position.
GetEmployeeState	EmployeeState DTO	int	Fragt den Status des Mitarbeiters mit der übergebenen ID ab
StartWork	bool	int, int, int	Startet die Zeiterfassung für den Mitarbeiter mit der übergebenen ID für die Zeitkategorie. Optional kann noch eine Projekt-ID mitgeben werden.
EndWork	bool	int	Beendet die Zeiterfassung für den Mitarbeiter mit der übergebenen ID.
StartBreak	bool	int	Startet die Pause für den Mitarbeiter mit der übergebenen ID.
EndBreak	bool	int	Beendet die Pause für den Mitarbeiter mit der übergebenen ID.

6.3 Server

6.3.1 Aufbau

Der Server ist in verschiedene Projekte unterteilt welche in diesem Abschnitt beschrieben werden.

BackendWebApp

Das Projekt BackendWebApp, welches ein ASP.NET Web API Projekt ist, enthält die REST-Schnittstelle und die Logik für die Authentifizierung. Für die Authentifizierung wurde das integrierte Interface `IAuthorizationFilter` implementiert. Damit der erstellte Filter auch verwendet wird muss der IIS durch die Klasse `WebApiApplication` so konfiguriert werden, dass dieser nicht die integrierte Authentifizierung von IIS sondern die im Code definierte verwendet. Ausserdem muss für den Filter die `AuthenticateAsync`-Methode überschrieben werden, welche die Logik für das Authentifizieren enthält. In diesem Projekt wurde für die Authentifizierung die Mail-Adresse und das Passwort des Benutzers verwendet.

Die REST-Schnittstelle ist in die verschiedenen Ressourcen unterteilt, das bedeutet jede Ressource hat eine eigene Klasse und eine eigene URI.

ASP.NET Web API bietet für die Implementierung einer REST-Schnittstelle einige Attribute, mit denen zum Beispiel die Security, die URI und die HTTP-Methode definiert werden können. Auch wurde in dieser Arbeit immer der `ResponseType` definiert, somit wird direkt vom Server geprüft ob das korrekte DTO im Response-Body zurückgesendet wird. Zur besseren Identifizierung der aufgerufenen HTTP-Operation, wurde das Attribut `SwaggerOperation` eingefügt, mit dem die HTTP-Operation im Response-Header gesetzt werden kann.

Jeder Parameter einer REST-Methode hat jeweils ein Attribut davor das angibt, ob dieser Parameter aus dem Body oder der URI des Requests gelesen werden soll (z.B. bei einem Request auf `/employees/{id}` würde der Parameter `id` aus der URI bezogen werden).

Speziell gehandhabt werden muss auch die Verwendung von Query Strings in der URI (z.B. `/projects?number=5`). Die Parameter in der entsprechenden Methode müssen für Datentypen welche nicht nullable sind entweder als nullable oder optionale Parameter angegeben werden (z.B. `GetProjects([FromUri]int? number = null)`).

BusinessLogic

Das Projekt BusinessLogic enthält die ausgelagerte Logik für die Controls der REST-Schnittstelle, damit die Trennung von REST und Logik garantiert werden kann.

Auch enthalten sind die verschiedenen Konverter, welche das wandeln von DTO zu Domain Model und umgekehrt übernehmen. Dies Konverter erlauben das Auflösen von Zwischentabellen, welche nur für das Speichern in der Datenbank benötigt werden, jedoch für die Clients nicht relevant sind.

DataAccess

Das Projekt DataAccess enthält alle Repositories welche für den Datenbank-Zugriff benötigt werden. Auch die Klasse Entities, welche als DbContext für das Entity Framework benötigt wird, ist darin enthalten. Diese Klasse konfiguriert das Entity Framework, das Lazy-Loading und die zu verwendenden Proxy-Objekte. Die Proxy-Objekte werden zwingend benötigt damit das Lazy-Loading korrekt funktioniert. Sie enthalten die persistierten Daten, können jedoch schneller erstellt werden als das reelle Objekt.

Auch das Cascade-on-Delete wird in der Entities Klasse für die meisten Entitäten deaktiviert. Dies wird gemacht, da in diesem Projekt die meisten Entitäten nicht aus der Datenbank entfernt, sondern nur per Tabellenfeld als gelöscht markiert werden.

DomainModel

Das Projekt enthält das Domain Model des Servers, welches in Kapitel 5.3 Detailliert beschrieben ist.

6.3.2 GenericDataRepository

Eine Spezialität des Servers ist das GenericDataRepository welches die Standard Funktionalitäten (CRUD) für den Datenbank-Zugriff anbietet. Jedoch bietet das GenericDataRepository auch die Möglichkeit die einzelnen Methoden zu überschreiben, zum Beispiel muss die Update Methode überschrieben werden damit die referenzierten Objekte im Kontext angehängt werden und das Updaten korrekt funktioniert. Der Vorteil ist, dass die Read-Methoden nur einmal erstellt werden müssen und für alle Entitäten gleich sind.

Als zusätzlicher Parameter für die Read-Methoden können Navigation-Properties übergeben werden, wodurch die referenzierten Entitäten direkt per "Join" in die SQL-Abfrage eingebunden werden und das Resultat mit "Where"-Abfragen verfeinert werden kann.

Das GenericDataRepository wird auch als Best Practice für das Arbeiten mit dem Entity Framework angegeben, es wurde mit der Hilfe des Blogs von Magnus Montin⁵ erstellt.

⁵ Montin, Magnus: "Implementing a generic data access layer using Entity Framework", Online Blog Article: <https://blog.magnusmontin.net/2013/05/30/generic-dal-using-entity-framework/>

6.4 Admin-Client

6.4.1 Aufbau

Der Admin-Client ist als klassische MVVM Applikation aufgebaut. Er besteht hauptsächlich aus der Main-View und den folgenden vier Stammdaten-Views. Die Main-View ermöglicht die Navigation zwischen den Stammdaten-Views mittels Registerkarten und Pivots:

- **Employee:** Zur Manipulation von Mitarbeitern
- **Project:** Zur Manipulation von Projekten
- **Position:** Zur Manipulation von Positionen
- **TimeCategory:** Zur Manipulation von Zeitkategorien

Jede dieser Stammdaten-Views ist nach dem Master-Detail Prinzip aufgebaut, das bedeutet, dass auf der linken Seite eine Liste aller verfügbaren Datensätze (Master) und auf der rechten Seite die Details zu einem ausgewählten Datensatz ersichtlich sind.

Zu jeder Master-Detail View gibt es ein entsprechendes View-Model, das die Daten vom Backend so aufbereitet, dass sie in den Views als Bindings genutzt werden können.

Neben den vier Stammdaten-Views gibt es drei Bewegungsdaten-Views, die ebenfalls nach dem Master-Detail Prinzip aufgebaut sind:

- **TimeRecording:** Diese Master-Detail-View wird aus der Detail-View eines Employees aufgerufen und erlaubt die Manipulation der Zeiterfassungen des entsprechenden Mitarbeiters
- **ProjectTimeRecording:** Diese Master-Detail-View wird aus der Detail-View eines TimeRecordings aufgerufen und erlaubt die Manipulation der Projektzeiterfassungen der entsprechenden Zeiterfassung
- **Break:** Diese Master-Detail-View wird aus der Detail-View eines TimeRecordings aufgerufen und erlaubt die Manipulation der Pausen der entsprechenden Zeiterfassung

Dazu kommen zwei weitere Views die sich dasselbe View-Model teilen, da sie von zwei verschiedenen Seiten an denselben Daten arbeiten:

- **AssignProjectToEmployee:** Wird auf der Employee Detail-View aufgerufen und erlaubt die Zuweisung von Projekten zum ausgewählten Mitarbeiter
- **AssignEmployeeToProject:** Wird auf der Project Detail-View aufgerufen und erlaubt die Zuweisung von Mitarbeitern zum ausgewählten Projekt

Die folgenden Views haben kein View-Model, da sie nicht einen Teil des Admin-Client Domain Models, sondern lediglich die Beziehung zwischen zwei Klassen darstellen:

- **AssignPositionToTimeCategory:** Wird auf der TimeCategory Detail-View aufgerufen und erlaubt die Zuweisung von Positionen zur ausgewählten Zeitkategorie
- **AssignTimeCategoryToPosition:** Wird auf der Position Detail-View aufgerufen und erlaubt die Zuweisung von Zeitkategorien zur ausgewählten Position

6.4.2 Bearbeitungsmodus

Eine Spezialität des Admin-Clients ist die Umsetzung des Bearbeitungsmodus. Dieser muss folgende Anforderungen erfüllen:

- Änderungen am GUI müssen über das Two-Way Binding direkt in das referenzierte DTO im View-Model übernommen werden
- Beim Speichern kann direkt die aktualisierte Version des DTOs im View-Model auf den Server hochgeladen werden, ohne zuerst die aktuellen Daten aus dem GUI abfragen zu müssen
- Beim Abbrechen müssen die Originalwerte des DTOs wieder im GUI angezeigt werden können ohne das Original DTO erneut vom Server zu laden

Dies wurde mit einer vereinfachten Version des Memento Patterns realisiert. Dadurch, dass alle Properties im View-Model direkt auf das entsprechende DTO referenzieren, kann durch ein Klonen des DTOs eine exakte Kopie der gesamten Daten des View-Models angefertigt werden.

Der Ablauf beim Bearbeiten und Abbrechen sieht folgendermassen aus:

1. Der Benutzer wechselt den aktuellen Datensatz in den Bearbeitungsmodus
2. Auf dem View-Model wird ein Klon des referenzierten DTOs gemacht und die Referenz auf das Original DTO in eine Backup-Variabel gesichert
3. Der Benutzer beendet den Bearbeitungsmodus durch Abbrechen
4. Auf dem View-Model wird die Original DTO Referenz aus der Backup-Variabel zurückkopiert und das GUI Benachrichtigt, dass sich die Properties geändert haben
5. Durch das Binding lädt sich das GUI wieder die alten Properties des Original DTOs

Der Ablauf beim Bearbeiten und Speichern sieht folgendermassen aus:

1. Der Benutzer wechselt den aktuellen Datensatz in den Bearbeitungsmodus
2. Auf dem View-Model wird ein Klon des referenzierten DTOs gemacht und die Referenz auf das Originalobjekt in eine Backup-Variabel gesichert
3. Der Benutzer beendet den Bearbeitungsmodus durch Speichern
4. Auf dem View-Model wird das bearbeitete DTO an den Server übermittelt

6.5 Terminal-Client

6.5.1 Aufbau

Der Terminal-Client ist eigentlich ein endlicher Automat, bei dem jeder angezeigte Screen einem Zustand und jede Benutzeraktion einem Zustandsübergang entspricht. Dies wurde bereits mit dem Ablaufdiagramm (entspricht dem Zustandsübergangdiagramm) in Kapitel 4.1.3 veranschaulicht.

Der Terminal-Client ist aus den folgenden drei Komponenten aufgebaut.

MainController

Der MainController ist das Herzstück des Terminal-Clients. Er erfüllt folgende Aufgaben innerhalb des Terminal-Clients:

- Kommunikation mit dem Server über die REST-API
- Bereitstellung der Serverdaten für die ScreenControllers
- Bereitstellung der Zustandslogik und Navigation zwischen den ScreenControllern

ScreenControllers

Es gibt pro Screen (gemäss Ablaufdiagramm in Kapitel 4.1.3) einen separaten ScreenController der für den jeweiligen Screen-Inhalt zuständig ist. Wie im Ablaufdiagramm beschrieben gibt es drei Arten von ScreenControllern, die jeweils dasselbe Interface implementieren:

- **Text Screen:** Diese Controller dienen lediglich der Anzeige eines Textes auf dem Bildschirm und implementieren das Interface `ITextViewController`
- **Button Screen:** Diese Controller dienen der Anzeige eines Titels sowie drei Knöpfen und implementieren das Interface `IButtonViewController`
- **List Screen:** Diese Controller dienen der Anzeige eines Titels, einer Liste sowie zwei Knöpfen (Zurück und Weiter) und implementieren das Interface `IListViewController`

Views

Die Views definieren das effektive Aussehen eines Screens. Es gibt pro Screen Typ (Text, Button und List) eine separate View. Die Bindings dieser Views sind jeweils gegenüber den oben genannten Interfaces der ScreenControllers gemacht.

6.5.2 Sequenzdiagramm

Das gesamte Programm ist nach dem "Inversion of Control" Prinzip aufgebaut. Der Main-Controller gibt die Kontrolle an den ScreenController ab und dieser gibt sie wiederum an die View ab.

Das folgende Sequenzdiagramm zeigt einen Ausschnitt aus dem Gesamtablauf des Programms, um das Zusammenspiel zwischen MainController, ScreenControllern und Views zu veranschaulichen. Es beginnt mit dem Aufruf des "Projekt wählen"-Screens (gemäß Kapitel 4.1.3).

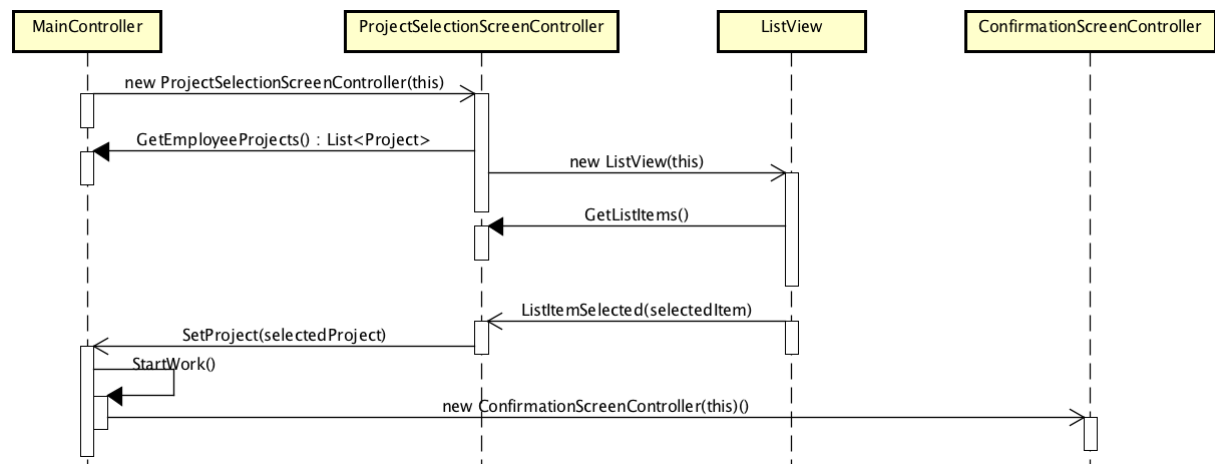


Abbildung 12: Sequenzdiagramm Terminal-Client

1. Der MainController (MC) erzeugt eine neue ProjectSelectionScreenControllers (PSSC) Instanz und gibt sich selber als Parameter mit (damit der PSSC den MC kennt)
2. Der PSSC holt sich vom MC die Liste aller verfügbaren Projekte des aktuellen Benutzers (diese hat der MC in einem früheren Schritt vom Server geladen)
3. Der PSSC erzeugt eine neue ListView Instanz und gibt sich selber als Parameter mit (Damit die ListView den PSSC kennt)
4. Die ListView holt sich beim PSSC die ListItems (im IListViewController Interface definiert) und bekommt die vom PSSC konvertierte Liste aus Schritt 2 zurück
5. Sobald der Benutzer in der ListView ein Item ausgewählt und bestätigt hat, informiert die ListView den PSSC darüber (hier vereinfacht als ListItemSelected dargestellt, im Code wird hier mit Binding gearbeitet)
6. Der PSSC konvertiert das selectedItem zurück in ein Project und ruft die SetProject Methode mit dem selectedProject als Parameter auf dem MC auf
7. Der MC hat jetzt alle Informationen (der hier gezeigt Ausschnitt war der letzte dazu notwendige Schritt) um beim Server die Zeiterfassung zu starten (was hier vereinfacht als StartWork dargestellt wird)
8. Sobald der MC die Bestätigung vom Server hat, wird der ConfirmationScreenController instanziiert und der gesamte Ablauf beginnt wieder von vorne

6.5.3 Zustands-Management

Damit der MainController bei der Anmeldung eines Mitarbeiters weiss, welchen Screen-Controller er aufrufen, bzw. welche Informationen er dem ScreenController bereitstellen muss, braucht er folgende Zustandsinformationen:

- **Status:** Ist der Mitarbeiter bei der Arbeit, in der Pause oder abwesend?
- **Startzeit:** Seit wann hat der Mitarbeiter den aktuellen Status? (Wird für Anzeigen wie „Du bist seit 09:35 Uhr in der Pause“ benötigt)
- **Zeitkategorie:** Welche ist die aktuelle Zeitkategorie des Mitarbeiters, oder wenn er nicht am Arbeiten ist, welche war seine letzte? (Wird benötigt um die aktuelle Kategorie anzuzeigen oder beim Arbeitsbeginn mit den Settings vom Vortag weiterzumachen)
- **Projekt:** Welches ist das aktuelle Project des Mitarbeiters, oder wenn er nicht am Arbeiten ist, welches war sein letztes? (Wird benötigt um das aktuelle Projekt anzuzeigen oder beim Arbeitsbeginn mit den Settings vom Vortag weiterzumachen)

Um das herauszufinden, müsste der MainController alle TimeRecordings, ProjectTimeRecordings und Breaks des Mitarbeiters vom Server laden und auswerten. Da dies sehr viel Netzwerkverkehr und Rechenleistung auf dem Client bedeuten würde, haben wir uns entschieden diese Logik auf den Server auszulagern.

Alle oben genannten Informationen werden vom Server in ein **EmployeeStateDTO** gepackt und können vom Client über die Control-API (siehe Kapitel 5.8.3) abgefragt werden. Das EmployeeStateDTO enthält folgende Felder:

- **State:** ENUM mit den möglichen Werten OffWork, Working und Break
- **StartOfCurrentState:** DateTimeOffset mit Startzeitpunkt des aktuellen State
- **CurrentTimeCategory:** TimeCategoryDTO der aktuellen Zeitkategorie
- **CurrentProject:** ProjectDTO des aktuellen Projekts (kann auch null sein)

6.6 REST Client API

Die REST Client API stellt den verschiedenen Clients alle auf dem Server verfügbaren REST-Ressourcen in Form eines einheitlichen Interfaces zur Verfügung. Die genaue Definition dieses IClientApi Interfaces ist in Kapitel 0 zu finden.

Die Hauptaufgabe der REST Client API ist es, Methodenaufrufe in REST-Anfragen und REST-Antworten in DTOs umzuwandeln.

6.6.1 ClientApi

Die ClientApi Klasse implementiert das IClientApi Interface und hat die Aufgabe Interface-Methodenaufrufe in RestHttpClient-Methodenaufrufe zu konvertieren. Dabei wurden folgende CRUD Operationen für den Zugriff auf die DTO-Ressourcen implementiert:

- **GetDto:** Zur Abfrage eines einzelnen DTOs mit ID
- **GetDtoList:** Zur Abfrage aller DTOs einer Ressource
- **UpdateDto:** Zum Aktualisieren eines DTOs
- **CreateDto:** Zur Erstellung eines DTOs
- **DeleteDto:** Zur Löschung eines DTOs
- **GetFilteredDtoList:** Zur gefilterten Abfrage aller DTOs einer Ressource
- **GetRelatedDtoList:** Zur Abfrage aller DTOs einer referenzierten Ressource

Dazu kommen einige Methoden, welche für den Zugriff auf Control-Ressourcen konzipiert sind, die aber hier nicht näher beschrieben werden müssen.

Da die oben genannten Methoden wie erwähnt generisch sind, musste eine Methode geschrieben werden, um die DTO Namen (die dem Methodenaufruf als Typen-Parameter mitgegeben werden) in REST-Ressourcen-Namen zu konvertieren. Da die Namensgebung der REST-Ressourcen und DTOs konsistent ist, kann die Konvertierung in ein paar wenigen Schritten vollzogen werden:

1. Wandle den Typennamen in Kleinbuchstaben (z.B. EmployeeDTO → employeedto)
2. Entferne die letzten drei Zeichen (z.B. employeedto → employee)
3. Falls nötig, ersetze das 'y' am Ende durch 'ie' (z.B. timecategory → timecatergorie)
4. Hänge ein 's' an (employee → employees)

6.6.2 RestHttpClient

Die RestHttpClient Klasse liegt zwischen der REST-API des Servers und der ClientApi Klasse des Clients und hat folgende Aufgaben:

- Aufbau und Aufrechterhaltung einer HTTP Verbindung zur REST-API des Servers
- Serialisierung und Versand von DTOs und andern Anfragen
- Empfang und Deserialisierung von DTOs und andern Antworten
- Auswertung der HTTP Status-Codes des Servers

6.7 Raspberry Pi

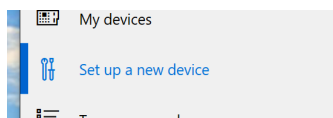
6.7.1 Inbetriebnahme

In diesem Kapitel wird beschrieben, wie ein Raspberry Pi von Grund auf mit Windows 10 IoT Core aufgesetzt wird. Folgende Komponenten sind gegeben:

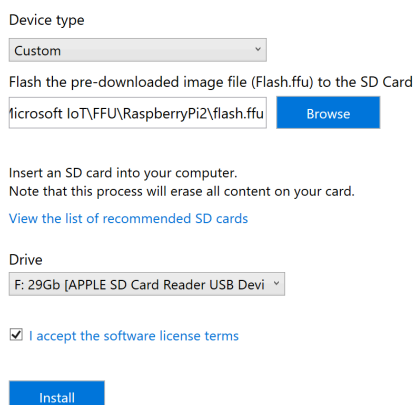
- Raspberry Pi 3
- Raspberry Pi 7" Touchscreen Display von element14
- 32GB microSD Karte mit microSD zu SD Adapter
- PC mit Windows 10 und SD Kartenleser (für die Installation)

Die Installation wird grösstenteils auf dem PC durchgeführt. Hier eine Schritt für Schritt Beschreibung dazu:

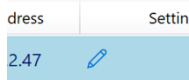
1. Den Installer des Windows 10 IoT Core Dashboard unter folgenden Link herunterladen: <https://iottools.blob.core.windows.net/iotdashboard/setup.exe>
2. Heruntergeladenes setup.exe starten → Das Windows 10 IoT Core Dashboard wird automatisch installiert
3. Da das Windows 10 IoT Core Image für den Raspberry Pi 3 erst als Insider Preview verfügbar ist, muss man sich zuerst unter <https://insider.windows.com> für das Windows Insider Programm anmelden
4. Das ISO-Image für Raspberry Pi 3 Windows 10 IoT Core unter folgendem Link herunterladen: <http://go.microsoft.com/fwlink/?LinkID=733603>
5. Das heruntergeladene ISO-Image mounten
6. Das darin befindliche MSI-File installieren
7. Im Windows 10 IoT Core Dashboard "Setup a new device" wählen:



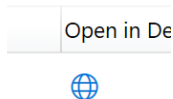
8. Als "Device type" **Custom** auswählen und folgenden Pfad angeben: C:\Program Files (x86)\Microsoft IoT\FFU\RaspberryPi2\flash.ffu
9. SD Card auswählen, "software license terms" Akzeptieren und auf **Install** klicken:



10. Die microSD in den Raspberry Pi einsetzen und mit Strom und Netzwerk verbinden
11. Der Raspberry Pi macht den initial-Boot, was etwas länger dauern kann
12. Im Windows 10 IoT Core Dashboard unter "My devices" beim Raspberry Pi auf das Bearbeitungssymbol klicken



13. Das Passwort unter "Change device password" ändern. Das Standard Passwort ist: p@ssw0rd
14. Durch einen Klick auf das Globus Symbol gelangt man auf die Managementseite:



15. Auf der Managementseite unter "Time zone" die richtige Zeitzone einstellen

6.7.2 NFC Reader Konfiguration

Dieses Kapitel beschreibt die Inbetriebnahme des NFC Readers. Die Verbindung mit dem Reader erfolgt über den I²C Bus mittels der GPIO Pins des Raspberry Pis. Die Treiber für den I²C Bus sind im Windows 10 IoT Core bereits enthalten. Microsoft schlägt in ihrer Windows 10 IoT Core Hardware Kompatibilitätsliste⁶ unter anderem das OM5577 Demoboard von NXP vor. Dieses wurde im Rahmen dieser Arbeit verwendet und dessen Konfiguration ist hier beschrieben:

1. Das ZIP mit dem NFC Treiber von der Herstellerseite herunterladen und entpacken: <http://cache.nxp.com/documents/software/SW3497.zip>
2. Im Windows Explorer auf das C: Laufwerk des Raspberry Pis verbinden (die IP ist im Windows 10 IoT Core Dashboard ersichtlich): \\<IP des Raspberry Pis>\c\$
3. Die heruntergeladenen Dateien **acpitabl.dat** und **NXPPN71x0.inf** auf den Raspberry Pi nach **C:\Windows\system32** kopieren
4. Die PowerShell-Konsole als Administrator starten
5. Folgende Befehle der Reihe nach in der PowerShell-Konsole ausführen:
 - a. Net start WinRM
 - b. Set-Item WSMAN:\localhost\Client\TrustedHosts -Value <Raspberry Pi IP>
 - c. Enter-PSSession -ComputerName <Pi IP> -Credential Administrator
 - d. Im Passwortprompt das Administrator Passwort des Raspberry Pis eingeben
 - e. devcon -dp_add C:\Windows\system32\NXPPN71x0.inf
6. Nach einem Neustart des Raspberry Pis ist der NFC Reader einsatzbereit

⁶ Microsoft: "Hardware Compatibility List", Windows Dev Center Online: <https://developer.microsoft.com/en-us/windows/iot/win10/supportedinterfaces#NFC>

6.8 Testplan

6.8.1 Admin-Client

Test 1		
Funktionalität	Konfiguration des Admin-Client	
Testkondition, Startpunkt	Admin-Client wurde noch nicht konfiguriert und kein Benutzer ist eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Starte den Admin-Client 2. Trage in die Felder folgende Informationen ein: <ol style="list-style-type: none"> a. Mail: admin@apze.local b. Passwort: admin c. URL: https://apze.azurewebsites.net/api/ 3. Auf Login drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Der Admin-Client starte und zeigt den Loginmenu an. • Der Login funktioniert und das Loginmenu wird geschlossen 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 2	
Funktionalität	Erstellen einer Zeitkategorie
Testkondition, Startpunkt	Dieser Test startet nach Test 1. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Zeitkategorie2. Eine neue Zeitkategorie anlegen per +3. Folgende Informationen in das neue Formular eintragen:<ol style="list-style-type: none">a. Zeitkategorienname: Standard Kategorieb. Zeit-Typ: Arbeitszeitc. Zeit-Multiplikator: 14. Auf Speichern drücken
Erwartete Resultate	<ul style="list-style-type: none">• Es werden keine Zeitkategorien angezeigt• Der rechte Teil des Tools zeigt das leere Formular für eine Zeitkategorie an• Die Zeitkategorie wurde gespeichert und wird in der Liste angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 3	
Funktionalität	Bearbeiten einer Zeitkategorie
Testkondition, Startpunkt	Dieser Test startet nach Test 2 Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Zeitkategorie 2. Die Zeitkategorie «Standard Kategorie» auswähle 3. Bearbeitungsmodus aktivieren 4. Folgende Informationen ändern: <ol style="list-style-type: none"> a. Zeitkategorienname: Standard b. Zeit-Multiplikator: 1.5 5. Zeitkategorie Speichern
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird die Zeitkategorie «Standard Kategorie» angezeigt • Es werden die Informationen für die Zeitkategorie «Standard» angezeigt • Die Zeitkategorie wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 4	
Funktionalität	Erstellen einer Position
Testkondition, Startpunkt	Dieser Test startet nach Test 1. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Position2. Eine neue Position anlegen per +3. Folgende Informationen in das neue Formular eintragen:<ol style="list-style-type: none">a. Positionsname: Mitarbeiterb. Standard Zeitkategorie: Standardc. Soll-Arbeitszeit: 404. Auf Speichern drücken
Erwartete Resultate	<ul style="list-style-type: none">• Es werden keine Positionen angezeigt• Der rechte Teil des Tools zeigt das leere Formular für eine Position an• Die Position wurde gespeichert und wird in der Liste angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 5		
Funktionalität	Bearbeiten einer Position	
Testkondition, Startpunkt	Dieser Test startet nach Test 4. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Position 2. Die Position «Mitarbeiter» auswähle 3. Bearbeitungsmodus aktivieren 4. Folgende Informationen ändern: <ol style="list-style-type: none"> a. Positionsname: Projektleiter b. Soll-Arbeitszeit: 44 5. Position Speichern 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird die Position «Mitarbeiter» angezeigt • Es werden die Informationen für die Position «Mitarbeiter» angezeigt • Die Zeitkategorie wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 6	
Funktionalität	Erstellen eines Benutzers
Testkondition, Startpunkt	Dieser Test startet nach Test 1. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Einen neuen Mitarbeiter anlegen per +3. Folgende Informationen in das neue Formular eintragen:<ol style="list-style-type: none">a. Vorname: Hansb. Nachname: Muster1c. Mail: h.muster@apze.locald. Passwort: 1234e. Workload: 100%f. Sicherheitsrolle: Userg. Position: Projektleiter4. Mitarbeiter speichern
Erwartete Resultate	<ul style="list-style-type: none">• Es wird nur der Admin angezeigt in der Liste• Es wird das leere Formular für einen Mitarbeiter angezeigt• Der erstellte Mitarbeiter wird in der Liste angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 7	
Funktionalität	Bearbeiten eines Benutzers
Testkondition, Startpunkt	Dieser Test startet nach Test 6. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster1» auswählen3. Bearbeitungsmodus aktivieren4. Folgende Information ändern:<ol style="list-style-type: none">a. Nachname: Muster5. Mitarbeiter speichern
Erwartete Resultate	<ul style="list-style-type: none">• Es werden zwei Benutzer in der Liste angezeigt• Es werden die Informationen von «Hans Muster1» angezeigt• Die Felder werden für die Bearbeitung aktiviert• Der Mitarbeiter wird aktualisiert und in der Liste mit dem neuen Namen dargestellt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 8	
Funktionalität	Löschen eines Benutzers
Testkondition, Startpunkt	Dieser Test startet nach Test 7. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster» auswählen3. Den Mitarbeiter per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es werden zwei Benutzer in der Liste angezeigt• Es werden die Informationen von «Hans Muster» angezeigt• Der Mitarbeiter wurde gelöscht und wird nicht mehr in der Liste angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 9	
Funktionalität	Reaktivieren eines Benutzers
Testkondition, Startpunkt	Dieser Test startet nach Test 8. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Filter aktivieren das deaktivierte Mitarbeiter angezeigt werden 3. Den Mitarbeiter «Hans Muster» auswählen 4. Bearbeitungsmodus aktivieren 5. Das Austrittsdatum löschen 6. Mitarbeiter speichern 7. Den Filter deaktivieren das gelöschte Mitarbeiter angezeigt werden
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird nur der Administrator in der Liste angezeigt • Der Mitarbeiter «Hans Muster» wird in der Liste angezeigt • Die Informationen vom Mitarbeiter «Hans Muster» werden angezeigt • Die Felder werden für die Bearbeitung aktiviert • Der Mitarbeiter wird gespeichert und das Zeichen für gelöscht ist nicht mehr sichtbar • Es werden zwei Mitarbeiter angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 10	
Funktionalität	Erstellen eines Projektes
Testkondition, Startpunkt	Dieser Test startet nach Test 1. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Projekt2. Unterhalb der Auflistung der Projekte auf das + drücken3. Folgende Informationen in das neue Formular eintragen:<ol style="list-style-type: none">a. Projektname: TestProjekt1b. Projektnummer: 100c. Startdatum: aktuelles Datumd. Enddatum: aktuelles Datum + 1 Jahr4. Projekt Speichern
Erwartete Resultate	<ul style="list-style-type: none">• Es werden keine Projekte angezeigt• Der rechte Teil des Tools zeigt das leere Formular für ein Projekt an• Das Projekt wurde gespeichert und wird in der Liste angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 11	
Funktionalität	Bearbeiten eines Projektes
Testkondition, Startpunkt	Dieser Test startet nach Test 10. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Projekt2. Das Projekt «TestProjekt1» auswähle3. Bearbeitungsmodus aktivieren4. Folgende Informationen ändern:<ol style="list-style-type: none">a. Projektname: TestProjektb. Projektnummer: 1015. Projekt Speichern
Erwartete Resultate	<ul style="list-style-type: none">• Es wird das Projekt «TestProjekt1» angezeigt• Der rechte Teil des Tools werden die Informationen für das Projekt angezeigt• Das Projekt wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 12	
Funktionalität	Löschen eines Projektes
Testkondition, Startpunkt	Dieser Test startet nach Test 11. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Projekt2. Das Projekt «TestProjekt» auswähle3. Das Projekt per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird das Projekt «TestProjekt» angezeigt• Es werden die Informationen für das Projekt «TestProjekt» angezeigt• Das Projekt wurde gelöscht und wird nicht mehr in der List angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 13		
Funktionalität	Reaktiveren eines Projektes	
Testkondition, Startpunkt	Dieser Test startet nach Test 12. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Projekt 2. Den Filter aktivieren das abgeschlossene Projekte angezeigt werden 3. Das Projekt «TestProjekt» auswähle 4. Das Enddatum auf das aktuelle Datum + 1 Jahr 5. Projekt speichern 6. Den Filter deaktivieren das gelöschte Projekte angezeigt werden 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird kein Projekt angezeigt • Es wird das Projekt «TestProjekt» angezeigt • Es werden die Informationen für das Projekt «TestProjekt» angezeigt • Das Projekt wurde gespeichert • Das Projekt «TestProjekt» wird angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 14	
Funktionalität	Zuweisen einer Zeitkategorie zu einer Position
Testkondition, Startpunkt	Dieser Test startet nach Test 3 und Test 5. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt, Es wurde noch eine zweite Zeitkategorie erstellt namens «Überzeit»
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Position2. Die Position «Projektleiter» auswählen3. Auf Zeitkategorie zuweisen drücken4. Die erstelle Zeitkategorie «Überzeit» der Position zuweisen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird die Position «Projektleiter» angezeigt• Es werden die Detailinformationen der Position «Projektleiter» angezeigt• Es wird die Zeitkategorie «Standard» als zugewiesene Zeitkategorien angezeigt• In der Liste der nicht zugewiesenen wird die Zeitkategorie «Überzeit» angezeigt• Die Zeitkategorie «Überzeit» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 15	
Funktionalität	Zuweisung einer Zeitkategorie zu einer Position löschen
Testkondition, Startpunkt	Dieser Test startet nach Test 14. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Position2. Die Position «Projektleiter» auswählen3. Auf Zeitkategorie zuweisen drücken4. Die erstelle Zeitkategorie «Überzeit» von der Position entfernen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird die Position «Projektleiter» angezeigt• Es werden die Detailinformationen der Position «Projektleiter» angezeigt• Es werden keine nicht zugewiesenen Zeitkategorien angezeigt• In der Liste der zugewiesenen wird die Zeitkategorien «Standard» und «Überzeit» angezeigt• Die Zeitkategorie «Überzeit» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 16	
Funktionalität	Zuweisen einer Position zu einer Zeitkategorie
Testkondition, Startpunkt	Dieser Test startet nach Test 3 und Test 5. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Zeitkategorie2. Die Position «Überzeit» auswählen3. Auf Position zuweisen drücken4. Die erstelle Position «Projektleiter» der Zeitkategorie zuweisen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird die Zeitkategorien «Standard» und «Überzeit» angezeigt• Es werden die Detailinformationen der Zeitkategorie «Überzeit» angezeigt• Es werden keine zugewiesenen Positionen angezeigt• In der Liste der nicht zugewiesenen wird die Position «Projektleiter» angezeigt• Die Position «Projektleiter» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 17	
Funktionalität	Zuweisung einer Position zu einer Zeitkategorie entfernen
Testkondition, Startpunkt	Dieser Test startet nach Test 16. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Zeitkategorie 2. Die Position «Überzeit» auswählen 3. Auf Position zuweisen drücken 4. Die erstelle Position «Projektleiter» von der Zeitkategorie entfernen
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird die Zeitkategorien «Standard» und «Überzeit» angezeigt • Es werden die Detailinformationen der Zeitkategorie «Überzeit» angezeigt • Es werden keine nicht zugewiesenen Positionen angezeigt • In der Liste der zugewiesenen wird die Position «Projektleiter» angezeigt • Die Position «Projektleiter» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 18	
Funktionalität	Zuweisen eines Mitarbeiters zu einem Projekt
Testkondition, Startpunkt	Dieser Test startet nach Test 9 und Test 13. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Projekt2. Das Projekt «TestProjekt» auswählen3. Auf Mitarbeiter zuweisen drücken4. Der erstelle Mitarbeiter «Hans Muster» dem Projekt zuweisen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird das Projekt «TestProjekt» angezeigt• Es werden die Detailinformationen des Projektes «TestProjekt» angezeigt• Es werden keine zugewiesenen Mitarbeiter angezeigt• In der Liste der nicht zugewiesenen wird der Mitarbeiter «Hans Muster» angezeigt• Der Mitarbeiter «Hans Muster» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 19		
Funktionalität	Zuweisung eines Mitarbeiters zu einem Projekt entfernen	
Testkondition, Startpunkt	Dieser Test startet nach Test 18. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Projekt 2. Das Projekt «TestProjekt» auswählen 3. Auf Mitarbeiter zuweisen drücken 4. Der erstelle Mitarbeiter «Hans Muster» aus dem Projekt entfernen 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird das Projekt «TestProjekt» angezeigt • Es werden die Detailinformationen des Projektes «TestProjekt» angezeigt • Es werden keine nicht zugewiesenen Mitarbeiter angezeigt • In der Liste der zugewiesenen wird der Mitarbeiter «Hans Muster» angezeigt • Der Mitarbeiter «Hans Muster» wechselt die Liste 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 20	
Funktionalität	Zuweisen eines Projektes zu einem Mitarbeiter
Testkondition, Startpunkt	Dieser Test startet nach Test 9 und Test 13. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster» auswählen3. Auf Projekte zuweisen drücken4. Das erstelle Projekt «TestProjekt» dem Mitarbeiter zuweisen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird der Mitarbeiter «Hans Muster» angezeigt• Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt• Es werden keine zugewiesenen Projekte angezeigt• In der Liste der nicht zugewiesenen wird das Projekt «TestProjekt» angezeigt• Das Projekt «TestProjekt» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 21	
Funktionalität	Zuweisung eines Projektes zu einem Mitarbeiter entfernen
Testkondition, Startpunkt	Dieser Test startet nach Test 20. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster» auswählen3. Auf Projekte entfernen drücken4. Das erstelle Projekt «TestProjekt» von dem Mitarbeiter entfernen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird der Mitarbeiter «Hans Muster» angezeigt• Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt• Es werden keine nicht zugewiesenen Projekte angezeigt• In der Liste der zugewiesenen wird das Projekt «TestProjekt» angezeigt• Das Projekt «TestProjekt» wechselt die Liste
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 22		
Funktionalität	Erstellen einer Zeiterfassung	
Testkondition, Startpunkt	Dieser Test startet nach Test 1. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt. Es ist der Benutzer «Hans Muster» erstellt welcher der Position «Projektleiter» und dem Projekt «TestProjekt» zugewiesen ist. Auch ist die Zeitkategorie «Standard» erstellt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Per + eine neue Zeiterfassung erstellen 5. Folgende Informationen eintragen: <ol style="list-style-type: none"> a. Von: 07.06.2016 08:00 b. Bis: 07.06.2016 17:00 c. Zeitkategorie: Standard d. Position: Projektleiter 6. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es wird das leere Formular für die Zeiterfassung angezeigt • Die Zeiterfassung wurde gespeichert und wird in der Liste angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 23		
Funktionalität	Bearbeiten einer Zeiterfassung	
Testkondition, Startpunkt	Dieser Test startet nach Test 22. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Bearbeitungsmodus aktivieren 6. Folgende Informationen bearbeiten: <ol style="list-style-type: none"> a. Bis: 07.06.2016 18:00 7. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Der Bearbeitungsmodus wurde aktiviert • Die Zeiterfassung wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 24		
Funktionalität	Erstellen einer Pause	
Testkondition, Startpunkt	Dieser Test startet nach Test 23. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Auf «Pause bearbeiten» drücken 6. Per + eine neue Pause erfassen 7. Folgende Informationen eintragen: <ol style="list-style-type: none"> a. Von: 07.06.2016 10:00 b. Bis: 07.06.2016 10:10 8. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Es werden alle Pausen angezeigt • Es wird das leere Formular für das erfassen eine Pause angezeigt • Die Pause wurde gespeichert und wird in der Liste angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 25		
Funktionalität	Bearbeiten einer Pause	
Testkondition, Startpunkt	Dieser Test startet nach Test 24. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Auf «Pause bearbeiten» drücken 6. Die erstelle Pause auswählen 7. Den Bearbeitungsmodus aktivieren 8. Folgende Informationen bearbeiten: <ol style="list-style-type: none"> a. Bis: 07.06.2016 10:15 9. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Es werden alle Pausen angezeigt • Es werden die Detailinformationen der Pause angezeigt • Der Bearbeitungsmodus wurde aktiviert • Die Pause wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 26	
Funktionalität	Löschen einer Pause
Testkondition, Startpunkt	Dieser Test startet nach Test 25. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster» auswählen3. Auf «Zeiterfassungen bearbeiten» drücken4. Die erstellte Zeiterfassung auswählen5. Auf «Pause bearbeiten» drücken6. Die erstelle Pause auswählen7. Die Pause per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird der Mitarbeiter «Hans Muster» angezeigt• Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt• Es werden alle Zeiterfassungen angezeigt• Es werden die Detailinformationen der Zeiterfassung angezeigt• Es werden alle Pausen angezeigt• Es werden die Detailinformationen der Pause angezeigt• Die Pause wurde gelöscht und wird nicht mehr angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 27		
Funktionalität	Erstellen einer Projektzeiterfassung	
Testkondition, Startpunkt	Dieser Test startet nach Test 23. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Auf «Projektzeit bearbeiten» drücken 6. Per + eine neue Projektzeit erfassen 7. Folgende Informationen eintragen: <ol style="list-style-type: none"> a. Von: 07.06.2016 10:00 b. Bis: 07.06.2016 12:00 c. Projekt: TestProjekt 8. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Es werden alle Projektzeiten angezeigt • Es wird das leere Formular für das erfassen einer Projektzeit angezeigt • Die Projektzeit wurde gespeichert und wird in der Liste angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 28		
Funktionalität	Bearbeiten einer Projektzeiterfassung	
Testkondition, Startpunkt	Dieser Test startet nach Test 27. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Auf «Projektzeit bearbeiten» drücken 6. Die erstellte Projektzeit auswählen 7. Den Bearbeitungsmodus aktivieren 8. Folgende Informationen bearbeiten: <ol style="list-style-type: none"> a. Von: 07.06.2016 11:00 b. Bis: 07.06.2016 14:00 9. Auf Speichern drücken 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Es werden alle Projektzeiten angezeigt • Es werden die Detailinformationen der Projektzeiterfassung angezeigt • Der Bearbeitungsmodus wurde aktiviert • Die Projektzeit wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 29		
Funktionalität	Löschen einer Projektzeiterfassung	
Testkondition, Startpunkt	Dieser Test startet nach Test 28. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.	
Ablauf	<ol style="list-style-type: none"> 1. Über die Navigation auf Mitarbeiter 2. Den Mitarbeiter «Hans Muster» auswählen 3. Auf «Zeiterfassungen bearbeiten» drücken 4. Die erstellte Zeiterfassung auswählen 5. Auf «Projektzeit bearbeiten» drücken 6. Die erstellte Projektzeit auswählen 7. Die erstellte Projektzeit per - löschen 	
Erwartete Resultate	<ul style="list-style-type: none"> • Es wird der Mitarbeiter «Hans Muster» angezeigt • Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt • Es werden alle Zeiterfassungen angezeigt • Es werden die Detailinformationen der Zeiterfassung angezeigt • Es werden alle Projektzeiten angezeigt • Es werden die Detailinformationen der Projektzeiterfassung angezeigt • Der Projektzeiterfassung wurde gelöscht 	
Erhaltene Resultate		
Passed / Failed	Passed <input type="checkbox"/>	Failed <input type="checkbox"/>

Test 30	
Funktionalität	Löschen einer Zeiterfassung
Testkondition, Startpunkt	Dieser Test startet nach Test 23. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt.
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Mitarbeiter2. Den Mitarbeiter «Hans Muster» auswählen3. Auf «Zeiterfassungen bearbeiten» drücken4. Die erstellte Zeiterfassung auswählen5. Die erstellte Zeiterfassung per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird der Mitarbeiter «Hans Muster» angezeigt• Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt• Es werden alle Zeiterfassungen angezeigt• Es werden die Detailinformationen der Zeiterfassung angezeigt• Die Zeiterfassung wurde gelöscht
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 31	
Funktionalität	Löschen einer Zeitkategorie
Testkondition, Startpunkt	Dieser Test startet nach Test 3. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Zeitkategorie2. Die Zeitkategorie «Standard» auswähle3. Die Zeitkategorie per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird die Zeitkategorie «Standard» angezeigt• Es werden die Informationen für die Zeitkategorie «Standard» angezeigt• Die Zeitkategorie wurde gelöscht und wird nicht mehr in der List angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 32	
Funktionalität	Löschen einer Position
Testkondition, Startpunkt	Dieser Test startet nach Test 5. Admin-Client Tool ist gestartet und der User ist als admin eingeloggt
Ablauf	<ol style="list-style-type: none">1. Über die Navigation auf Position2. Die Position «Projektleiter» auswähle3. Die Position per - löschen
Erwartete Resultate	<ul style="list-style-type: none">• Es wird die Position «Projektleiter» angezeigt• Es werden die Informationen für die Position «Projektleiter» angezeigt• Die Position wurde gelöscht und wird nicht mehr in der List angezeigt
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

6.8.2 Terminal-Client

Test 1	
Funktionalität	Konfiguration des Terminalclients
Testkondition, Startpunkt	Terminal wurde noch nicht konfiguriert und kein Benutzer ist eingeloggt. Der unten genannte Terminal Benutzer wurde vor-gängig mittels Admin-Client konfiguriert.
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Trage in die Felder folgende Informationen ein: <ol style="list-style-type: none"> a. Mail: terminal@apze.local b. Passwort: terminal c. URL: https://apze.azurewebsites.net/api/ 3. Auf Login drücken
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient starte und zeigt den Konfigurations-screen an. • Der Login funktioniert und der Terminalclient zeigt den Startscreen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 2	
Funktionalität	Badge zuweisen
Testkondition, Startpunkt	Terminal wurde konfiguriert und Terminal Benutzer ist eingeloggt. Ein neuer Badge ist vorhanden
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den neuen Badge auf den NFC-leser legen 3. Den Benutzer «Hans Muster» auswählen und mit ok bestä-tigen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es wird die Liste mit allen Benutzern ohne Badge ange-zeigt • Der Badge wurde zugewiesen
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 3	
Funktionalität	Arbeitszeit ohne Projekt beginnen
Testkondition, Startpunkt	Dieser Test startet nach Test 2
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Zeitkategorie / Projekt wählen 4. Zeitkategorie «Standard» auswählen und auf ok 5. Kein Projekt auswählen und auf ok
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Es werden alle Zeitkategorien angezeigt welcher der Benutzer wählen kann • Es werden alle Projekte angezeigt welche der Benutzer wählen kann • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 4	
Funktionalität	Pause beginnen
Testkondition, Startpunkt	Dieser Test startet nach Test 3
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Pause beginnen wählen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 5	
Funktionalität	Pause beenden
Testkondition, Startpunkt	Dieser Test startet nach Test 4
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Pause beenden wählen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 6	
Funktionalität	Zeitkategorie ändern
Testkondition, Startpunkt	Dieser Test startet nach Test 5
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Zeitkategorie / Projekt wählen 4. Zeitkategorie «Überzeit» auswählen und auf ok 5. Projekt «Testprojekt» auswählen und auf ok
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Es werden alle Zeitkategorien angezeigt welcher der Benutzer wählen kann • Es werden alle Projekte angezeigt welche der Benutzer wählen kann • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 7	
Funktionalität	Arbeitszeit beenden
Testkondition, Startpunkt	Dieser Test startet nach Test 6
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Arbeitszeit beenden wählen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 8	
Funktionalität	Letzte Einstellungen verwenden
Testkondition, Startpunkt	Dieser Test startet nach Test 7
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. «Überzeit / Testprojekt» wählen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 9	
Funktionalität	Normale Arbeitszeit beginnen
Testkondition, Startpunkt	Dieser Test startet nach Test 7
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. «Normale Arbeitszeit beginnen» wählen
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

Test 10	
Funktionalität	Arbeitszeit mit Projekt beginnen
Testkondition, Startpunkt	Dieser Test startet nach Test 7
Ablauf	<ol style="list-style-type: none"> 1. Starte den Terminalclient 2. Den konfigurierten Badge auf den NFC-leser legen 3. Zeitkategorie / Projekt wählen 4. Zeitkategorie «Standard» auswählen und auf ok 5. Projekt «TestProjekt» auswählen und auf ok
Erwartete Resultate	<ul style="list-style-type: none"> • Der Terminalclient startet. • Es werden die verschiedenen Optionen angezeigt • Es werden alle Zeitkategorien angezeigt welcher der Benutzer wählen kann • Es werden alle Projekte angezeigt welche der Benutzer wählen kann • Der Bestätigungsscreen zeigt nochmals alle Informationen an
Erhaltene Resultate	
Passed / Failed	Passed <input type="checkbox"/> Failed <input type="checkbox"/>

7 Zusammenfassung

7.1 Ergebnisse

Das Ergebnis dieser Bachelorarbeit sind Server, Terminal-Client und Admin-Client eines Zeiterfassungssystems, welche voll funktionsfähig sind. Um die erstellten Komponenten zu testen wurde pro Client ein Testplan erstellt und für den Server wurden Unit-Tests geschrieben, welche die Funktionalitäten der REST-Schnittstelle testen.

Der Terminal-Client wurde als endlicher Automat entwickelt, da sich dieser je nach Status und Aktion des eingeloggten Mitarbeiters nach einem vorgegebenen Schema verhält. Für die Anmeldung am Terminal, wird der Badge des Mitarbeiters verwendet von dem die GUID ausgelesen wird.

Damit der Terminal-Client einfach zu benutzen ist, wurde ein einheitliches Design für alle Screens verwendet, durch dies weiss der Benutzer immer wo er sich befindet und kommt einfach zum Ziel.

Der Admin-Client wurde als Universal Windows App (UWP) mit MVVM-Pattern entwickelt. Dank UWP konnte das gewohnte Look and Feel von Windows 10 widerspiegelt werden, wodurch sich der Benutzer schnell zurecht findet.

Der Admin-Client kann dank der Verwendung von UWP auch einfach auf Mobile-Geräte portiert werden, dafür müssen lediglich die Screens für dynamische Bildschirmgrösse angepasst werden.

Der Server wurde mit der ASP.NET Web API erstellt und konnte durch das mit einer REST-Schnittstelle ausgestattet werden. Die REST-Schnittstelle wurde in verschiedene Ressourcen aufgeteilt und bietet unter anderem auch Control-Ressourcen, welche automatisierte Funktionen wie StartWork anbieten. Dank dieser Control-Ressourcen konnte der Terminal-Client schlank gehalten und die Netzwerkkommunikation reduziert werden.

Das Persistieren der Daten wurde mit dem Entity Framework, welches sich sehr gut für einfache Zugriffe auf die Datenbank eignet, realisiert.

Der Server inklusive MSSQL-Datenbank wurde, den Anforderung entsprechend, auf Microsoft Azure Cloud publiziert. Dies ermöglicht es dem Unternehmen welches diese Lösung später einsetzt ohne eigene Server-Infrastruktur auszukommen.

Das in dieser Bachelorarbeit entwickelte System kann direkt Produktiv eingesetzt und auch nachträglich noch einfach mit zusätzlichen Features erweitert werden. Alle Anforderungen mit Priorität 1 wurden umgesetzt und getestet.

Das Ziel der Bachelorarbeit war unter anderem auch die Machbarkeit einer solchen Lösung zu evaluieren. Wenn man also das Resultat dieser Arbeit betrachtet, kann man die Machbarkeit eines solchen Systems ohne Vorbehalte bestätigen.

7.2 Ausblick

7.2.1 Erweiterungsmöglichkeiten

Dieses Kapitel beschreibt einige Erweiterungsmöglichkeiten, die zu einem späteren Zeitpunkt hinzugefügt werden können.

Eingeschränkter Zugriff auf Ressourcen

Um die Sicherheit zu erhöhen kann der Zugriff auf die Ressourcen noch weiter eingeschränkt werden. Mit der momentanen Implementation hat ein normaler Benutzer lese Zugriff auf alle Ressourcen, was nicht optimal ist. Mit einer Erweiterung der Zugriffsverwaltung könnten die Lese-Rechte soweit eingeschränkt werden, dass ein Benutzer nur noch auf seine eigenen Ressourcen Zugriff hat.

Reports

Die REST-API des Servers könnte mit einer Reporting-Funktionalität erweitert werden, mit der die Zeiterfassungen für ausgewählte Benutzer oder Projekte über einen bestimmten Zeitraum ausgewertet werden können. In einem zweiten Schritt könnten diese Reports dann ins GUI des Admin Clients integriert werden.

Clients für Mobile-Geräte

Die Clients sind momentan jeweils für die in der Arbeit beschriebenen Plattformen optimiert, könnten aber in Zukunft so angepasst werden, dass die Views auch auf anderen Geräten korrekt dargestellt werden. Dies betrifft vor allem den Admin-Client, bei dem auf eine skalierbare Grösse verzichtet wurde, da der Hauptfokus auf der Desktop- und Tablet-App lag.

Konfigurations-Badge

Um die Konfiguration des Terminal-Clients zu vereinfachen, könnte dieser so angepasst werden, dass er die Einstellungen von einem Konfigurations-Badge beziehen könnte. Da jeder Badge einen beschreibbaren Speicher hat (siehe Kapitel 5.1.4) könnte dieser mit den nötigen Server- und Login-Informationen beschrieben werden. Dies wäre dann von Vorteil, wenn ein Unternehmen viele Terminal-Clients die es konfigurieren muss, da es die Konfiguration erheblich erleichtert.

7.2.2 Verbesserungsvorschläge

Clientseitige Validierung

Momentan wird die Validierung der Daten auf dem Server vorgenommen, die Userexperience könnte jedoch erheblich verbessert werden, wenn bereits auf dem Client eine Validierung stattfindet. Dies würde das Anzeigen von spezifischen Fehlermeldungen im GUI erlauben, statt die Änderungen einfach auf dem Server zu verwerfen.

Security

Die Authentifizierung auf dem Server geschieht momentan per HTTP-Basic Authentication, welche nur so sicher wie die darunterliegende Verbindung (im besten Fall TLS) ist. Eine Verbesserung wäre, wenn die Authentifizierung über OAuth stattfindet. Alternativ könnte auch ein einmaliges HTTP-Basic Authentication gemacht werden bei der dann eine Session gestartet wird, welche für alle weiteren Anfragen verwendet wird. Durch eine dieser Technologien wäre die Sicherheit des Servers erhöht und die Login-Daten müssten nicht bei jeder Anfrage mitgesendet werden.

Error Handling

Das Error Handling geschieht meisten über HTTP-Statuscodes, welche mit einer spezifischen Nachricht ergänzt werden. Dies reicht jedoch nicht aus, um auf den Clients sinnvolle Fehlermeldungen zu generieren. Das Error Handling könnte beispielsweise durch ErrorDTOs verbessert werden. Solche ErrorDTOs würden genaue Informationen über den Error enthalten und der Client könnte anhand dessen ein spezifisches Recovery machen oder eine genaue Fehlermeldung anzeigen.

8 Referenzen

8.1 Quellen

Alle externen Quellen in dieser Arbeit sind direkt auf den entsprechenden Seiten in den Fussnoten ersichtlich.

8.2 Abbildungen

Alle folgenden Abbildungen sind eigene Darstellungen die im Rahmen dieser Arbeit erstellt wurden:

Abbildung 1: Ablaufdiagramm Terminal-Client	10
Abbildung 2: Terminal-Client Text Screen Mockup.....	11
Abbildung 3: Terminal-Client Button Screen Mockup	11
Abbildung 4: Terminal-Client List Screen Mockup	11
Abbildung 5: Admin-Client Master/Detail View Mockup.....	14
Abbildung 6: Admin-Client Overlay View Mockup.....	14
Abbildung 7: Software Layers.....	19
Abbildung 8: Server Domain Model.....	20
Abbildung 9: Admin-Client Domain Model.....	23
Abbildung 10: Klassendiagramm Server	25
Abbildung 11: Klassendiagramm Clients.....	27
Abbildung 12: Sequenzdiagramm Terminal-Client.....	43

9 Danksagung

Wir möchten unseren Dank an folgenden Personen aussprechen:

- Prof. Hansjörg Huser vom Institute for Networked Solutions (INS), HSR Rapperswil, für die Unterstützung und Betreuung in allen Bereichen
- Herrn Jürg Jucker von S3CC GmbH, Pfäffikon, für die Unterstützung und Betreuung in allen Bereichen
- Unseren Familien und Freunden für die tatkräftige Unterstützung während unserer Bachelorarbeit

10 Anhang

10.1 Projektantrag

Arbeitszeit und Projektzeiterfassung mit Windows 10 IoT und Microsoft Azure

Aufgabenstellung für Fabian Schäfer und Andreas Stefanits

In einem Prototyp soll eine Arbeitszeit- und Projektzeiterfassung mittels Microsoft Azure (Backend) und Windows IoT Core und Universal Apps realisiert werden. Dabei soll das Augenmerk auf der technischen Machbarkeit gelegt werden und weniger auf Funktionalität.

Funktionalität

Folgende Clients mit ähnlicher Funktionalität sollen unterstützt werden:

- Windows IoT Client (P1)
Diese Windows Universal App soll auf einem Raspberry PI mit Windows 10 IoT laufen und folgende Funktionen für einen Mitarbeiter unterstützen:
 - Kommen und Gehen
Arbeitszeitbeginn und Ende
 - Pause / Mittagspause etc.
Pausenstart und Arbeitsfortsetzung
 - Erfassung von Zeiten auf ein Projekt (Beginn und Ende),
dabei müssen Pausen berücksichtigt werden.

Mitarbeiter sollen sich entweder über NFC Tags oder über die Eingabe einer Personalnummer (Touch Display) identifizieren.
- SmartPhones / Tablets (Windows 10 (P2), iOS und Android (P3))
Dieser Client soll die folgenden Funktionen für einen Mitarbeiter, welcher sich über ein User (Username/Passwort) anmeldet (ev. Windows Live Id), unterstützen:
 - Kommen und Gehen
Arbeitszeitbeginn und Ende
 - Pause / Mittagspause etc.
Pausenstart und Arbeitsfortsetzung
 - Erfassung von Zeiten auf ein Projekt (Beginn und Ende),
dabei müssen Pausen berücksichtigt werden
- Windows 10 Universal App (Desktop) (P1)
Dieser Client soll die folgenden Funktionen für einen Mitarbeiter, welcher sich über ein User (Username/Passwort) anmeldet (ev. Windows Live Id), unterstützen:
 - Administrations- und Auswertungsfunktionen
 - Erfassen und Verwalten von Mitarbeitern
 - Erfassen und Verwalten von Projekten
 - Erstellen von Auswertungen
 - Arbeitszeiten
 - Projektzeiten
 - Mitarbeiterfunktionen
 - Kommen und Gehen
Arbeitszeitbeginn und Ende
 - Pause / Mittagspause etc.
Pausenstart und Arbeitsfortsetzung
 - Erfassung von Zeiten auf ein Projekt (Beginn und Ende),
dabei müssen Pausen berücksichtigt werden

- ASP.NET MVC Website (on Microsoft Azure) (P4)
Dieser Client soll dieselben Funktionen unterstützen wie der Windows 10 Universal App (Desktop) Client.
- Backend (Datenhaltung) auf Azure oder Server, mit REST oder WCF-Schnittstelle

Vorgehensvorschlag

1. Inbetriebnahme Raspberry Pi (Windows IoT) Core mit TouchScreen
2. Test Universal App (oder Mono) auf Raspberry Pi Plattform
3. Entwicklung Universal App für (Desktop) und Windows IoT Core für Kommen/Gehen/Pause mit Personalnummer (inkl. Backend)
4. Anschluss NFC Leser and Raspberry Pi.
5. Erweiterung App für NFC Leser
6. Erweiterung App für Surface mit NFC Leser
7. Entwicklung SmartPhone/Tablet App mit Universal App.
8. Erweiterung Funktionalität
9. ASP.NET MVC Client (Optional)

Projektteam

Fabian Schäfer, fabian.schaefer@hsr.ch
Andreas Stefanits, andreas.stefanits@hsr.ch

Auftraggeber

S3CC GmbH
Eichenstrasse 11
CH-8808 Pfäffikon SZ

Betreuung HSR

Jürg Jucker, jjucker@hsr.ch, Tel: 079 756 92 75
Hansjörg Huser, hhuser@hsr.ch, Tel: 055 222 49 12 , 079 276 43 09(HSR Raum 8.242)

Experte

Stefan Zettel, szettel@ascentive.ch, Ascentive Zürich

Gegenleser

tbd

Projektabwicklung

Termine:

- Siehe auch: <https://www.hsr.ch/Termine-Bachelor-und-Studiena.5142.0.html>
- Beginn der Arbeit: **Mo., 22. Feb. 2016**
- Zwischenpräsentation für Gegenleser und evtl Experte: ca. Ende April.
- Abgabetermin Abstract für DA-Broschüre: **8 .Juni 2016**
- Abgabetermin Kurzfassung/Poster/Mgmt-Summary zum Review: : **8 .Juni 2016**
- Abgabetermin Poster/Mgmt-Summary: **12. Juni 2016**
- Abgabetermin BA: **16. Juni 2016, 12.00 Uhr**
- Präsentation der Bachelorarbeiten, **16. Juni 2016, 16 bis 20 Uhr**
- Mündliche BA-Prüfung **Juli/Aug. 2016** : genaues Datum folgt

Arbeitsaufwand

Für die erfolgreich abgeschlossene Arbeit werden 12 ECTS angerechnet. Dies entspricht einer Arbeitsleistung von mind. 360 Stunden pro Student. (14 Wochen zu ca. 21h, 2 Wochen zu 45h)

Hinweise für die Gliederung und Abwicklung des Projektes:

Gliedern Sie Ihre Arbeit in 4 bis 5 Teilschritte. Schliessen Sie jeden Teilschritt mit einem Meilenstein ab. Definieren Sie für jeden Meilenstein, welche Resultate dann vorliegen müssen!

Folgende Teilschritte bzw. Meilensteine sollten Sie in Ihrer Planung vorsehen:

- Schritt 1: Projektauftrag inkl. Projektplan (mit Meilensteinen),
- Meilenstein 1: Review des Projektauftrages abgeschlossen. Projektauftrag von Auftraggeber und Dozent genehmigt
 Letzter Meilenstein: Systemtest abgeschlossen
 Termin: ca. eine Woche vor Abgabe
- Entwickeln Sie Ihre SW in einem iterativen, inkrementellen Prozess: Planen Sie möglichst früh einen ersten lauffähigen Prototypen mit den wichtigsten und kritischsten Kernfunktionen. In die folgenden Phasen können Sie dieses Kernsystem schrittweise ausbauen und testen.
- Falls Sie in Ihrer Arbeit neue oder Ihnen unbekannte Technologien einsetzen, sollten Sie parallel zum Erarbeiten des Projektauftrages mit dem Technologiestudium beginnen.
- Verwalten Sie ihre Software und Dokumente auf einem geeigneten Repository. Stellen Sie sicher, dass der/die Betreuer jederzeit Zugriff auf das Repository haben und dass das Projekt anhand des Repositories jederzeit wiederhergestellt werden kann.
- Achten Sie auf die Einhaltung guter Programmier- und Designprinzipien
- Halten Sie sich im Übrigen an die Vorgaben aus dem Modul SE-Projekt.

Projektadministration

- Führen Sie ein individuelles Projekttagebuch aus dem ersichtlich wird, welche Arbeiten Sie durchgeführt haben (inkl. Zeitaufwand). Diese Angaben sollten u.a. eine individuelle Beurteilung ermöglichen.
- Dokumentieren Sie Ihre Arbeiten laufend. Legen Sie Ihre Projektdokumentation mit der aktuellen Planung und den Beschreibungen der Arbeitsergebnisse elektronisch in einem Projektordner ab. Dieser Projektordner sollte jederzeit einsehbar sein.

Inhalt der Dokumentation

Bei der Abgabe muss jede Arbeit folgende Inhalte haben:

- Dokumente gemäss Vorgabe: <https://www.hsr.ch/Allgemeine-Infos-Diplom-Bach.4418.0.html>
- Aufgabenstellung
- Technischer Bericht
- Projektdokumentation
- Die Abgabe ist so zu gliedern, dass die obigen Inhalte klar erkenntlich und auffindbar sind.
- Zitate sind zu kennzeichnen, die Quelle ist anzugeben.
- Verwendete Dokumente und Literatur sind in einem Literaturverzeichnis aufzuführen.
- Projekttagebuch, Dokumentation des Projektverlaufes, Planung etc.
- Weitere Informationen: <https://www.hsr.ch/Ablaeufe-und-Regelungen-Studie.7479.0.html>
 (Dokument Ablaeufe_und_Regelungen_Studien-_und_Bachelorarbeiten_141027_.pdf)

Fortschrittsbesprechung:

Regelmässig findet zu einem fixen Zeitpunkt eine Fortschrittsbesprechung statt.
 Teilnehmer: Dozent und Studenten, bei Bedarf auch Vertreter der Auftraggeber

Termin: jeweils Mittwoch, 17h bis 18h, Raum 8.242 (Abweichungen werden rechtzeitig kommuniziert)

Traktanden

- Was wurde erreicht, was ist geplant, welche Probleme stehen an
- Review von Code/Dokumentation (Abgabe jeweils einen Tag vor dem Meeting)

Falls notwendig, können weitere Besprechungen / Diskussionen einberufen werden.

Sie erstellen zu jeder Besprechung ein Kurzprotokoll, welches Sie spätestens 2 Tage nach der Sitzung per e-mail an den Betreuer senden.

Rapperswil, 22. Feb. 2016
Hansjörg Huser, Jürg Jucker

10.2 Persönliche Berichte

10.2.1 Andreas Stefanits

Wie schon bei der Semesterarbeit habe ich die Bachelorarbeit mit Fabian Schäfer gemacht, jedoch haben wir dieses mal kein eigenes Thema bearbeitet, sondern haben uns über den Regulären Prozess für ein Thema beworben. Die Themawahl war schnell besprochen, denn Fabian und ich interessierten uns beide stark für diese Arbeit. Was zusätzlich von Vorteil war, war das Herr Huser die Betreuung der Arbeit machte und wir sehr gute Erfahrung in der Semesterarbeit mit ihm gemacht haben.

Was mich selbst sehr gefreut hat war, dass wir ein Projekt von Anfang bis Ende machen konnten. Was ich persönlich schon länger nicht mehr machen konnte und es mir viel Spass macht die Architektur von Grund auf aufzubauen. Bei der Architektur haben Fabian und ich viele Stunden investiert, damit diese korrekt funktioniert und auch erweiterbar ist. So haben wir nur schon für das DomainModel des Servers vier Versionen erstellt bis wir auf das aktuelle DomainModel kamen.

Auch über das GUI-Design haben wir viel gesprochen, aber Fabian hatte zu den meisten Bedenken und Problemen schnell eine Lösung. Auch bei anderen Problemen hatte Fabian stets eine sehr gute Idee wie wir dies lösen können.

Die ersten Wochen der Umsetzungsphase haben wir uns aufgeteilt um schneller einen Prototyp zu haben, welcher den Durchstich durch alle Layers gezeigt hat. Somit hat Fabian den Terminal-Client und ich den Server-Teil übernommen. Das Arbeiten mit C# und REST war neu für mich, jedoch hatte ich mich schnell in die Materie eingearbeitet. Und konnte schnell eine REST-Schnittstelle erstellen, welche wir im vorher genau definiert hatten. Das Anbinden des Terminal-Client an den Server verlief ohne grössere Problem, was mich sehr gefreut hat. Dies war dank der guten Vorbereitung und der gut definierten Architektur möglich.

Nachdem der Terminal-Client lauffähig war, widmeten wir uns dem Admin-Client. Am Anfang war es für mich recht schwer da ich noch nie mit UWP gearbeitet hatte und sich dieses doch in einigen Punkten von mir bekannten WPF unterscheidet. Jedoch nach einiger Zeit war es kein Problem mehr die Screens zu implementieren und bugfixes zu machen.

In der Testphase hat Fabian dank den von ihm erstellten Unit-Tests einige Fehler und fehlende Pre-conditions entdeckt. Auch haben wir während der Testphase festgestellt, dass das Context-Handling welches wir in Verbindung mit Entity Framework verwendeten, nicht das gewünschte Resultat bringt. Somit haben wir dies nochmals korrigiert, was viel Zeit gekostet aber sich gelohnt hat.

Ich möchte mich nochmals herzlich bei Fabian Schäfer für die gute Zusammenarbeit bedanken und werde die Zeit in der HSR steht's in guter Erinnerung halten.

10.2.2 Fabian Schäfer

Ich habe bereits meine Semesterarbeit zusammen mit Andreas Stefanits gemacht und da wir damals sehr gut miteinander harmonierten, entschieden wir uns die Bachelorarbeit ebenfalls gemeinsam zu bestreiten.

Anders als bei der Semesterarbeit hatten wir dieses Mal kein eigenes Thema, sondern mussten am regulären Bewerbungsverfahren auf die ausgeschriebenen Arbeiten teilnehmen. Diese Arbeit war dort meine erste Wahl, da mich das Thema sehr interessierte und ich auch schon privat damit zu tun hatte. Der Zufall wollte es so, dass ausgerechnet Herr Huser, dessen Betreuung wir bereits bei unserer Semesterarbeit sehr schätzten, erneut unser zugeteilte Dozent für diese Arbeit war.

Schon in den ersten Wochen war ich voller Motivation bei der Planung der Architektur und der GUIs. Manchmal kamen mir sogar spät nachts im Bett noch neue Ideen zum Aufbau der Architektur. Immerhin war es mein erstes grösseres Projekt bei dem ich von A-Z alles selber planen und aufbauen konnte. Natürlich immer in Zusammenarbeit mit meinem Projektpartner, dessen Expertise und langjährige Erfahrung ich stets zu schätzen wusste.

Der Start in die Umsetzungsphase verlief bei mir ein wenig harzig. Es war das erste Mal, dass ich eine UWP App programmierte, daher verbrachte ich die erste Woche damit, mich ins Thema einzuarbeiten und verschiedene Demoprogramme nachzubauen. In der zweiten Entwicklungswoche manifestierte sich dann schon die ersten Erfolgserlebnisse in Form eines rudimentär funktionellen Prototyps. Zwar musste ich den Terminal-Client in den darauffolgenden Wochen noch einmal komplett umkrempeln, aber ich habe auch viel dabei gelernt, das mir bei der späteren Entwicklung des Admin-Clients zugutekam.

Die Entwicklung des Admin-Clients gestaltete sich insgesamt einfacher als die der Terminal App, da ich schon mit der Materie vertraut war und gleich mit der Umsetzung loslegen konnte. Zwar gab es auch hier einige unerwartete Hürden, die primär auf die grundlegenden Unterschiede zwischen UWP und WPF zurückzuführen waren, jedoch konnten wir dank der eingeplanten Reserveweche die Entwicklung im Zeitplan zum Ende bringen.

Die verbleibenden Wochen verflogen wie im Flug. Während der Testphase sind noch einige Fehler auf der Serverseite zum Vorschein gekommen, was auch noch unerwartet viel Zeit gekostet hat, da die betroffenen Teile des Servers nochmals von Grund auf korrigiert werden mussten. Aber auch das konnte durch die zweite eingeplante Reserveweche abgefangen werden.

In den letzten zwei Wochen lag der Fokus wie geplant auf der Dokumentation, die in der Hitze des Programmier-Gefechts ein wenig vernachlässigt wurde. Doch im Endeffekt hat unsere Zeitplanung ganz gut hingehauen.

Ich schaue nun mit einem lachenden und einem weinenden Auge auf diese Arbeit, die gleichzeitig auch das Ende meiner Zeit an der HSR symbolisiert, zurück. Einerseits bin ich froh endlich am Ziel angekommen zu sein, andererseits war es auch eine sehr schöne und lehrreiche Zeit für mich, die ich so schnell nicht vergessen werde.

10.3 Projektplanung

10.3.1 Projekt Zeitplan

	Woche 1						Woche 2						Woche 3								
	MO 22.02	DI 23.02	MI 24.02	DO 25.02	FR 26.02	SA 27.02	SO 28.02	MO 29.02	DI 01.03	MI 02.03	DO 03.03	FR 04.03	SA 05.03	SO 06.03	MO 07.03	DI 08.03	MI 09.03	DO 10.03	FR 11.03	SA 12.03	SO 13.03
Meilensteine													M1								
Phase	Kick-Off						Planungs-Phase						Architektur-Phase								
Arbeit							Requirements Definition, Projektplan						Requirements Analyse, Technologien Evaluation								
Resultate							Entwurf Projektplan						Genehmigter Projektplan, Spezifikationen								

	Woche 4						Woche 5						Woche 6								
	MO 14.03	DI 15.03	MI 16.03	DO 17.03	FR 18.03	SA 19.03	SO 20.03	MO 21.03	DI 22.03	MI 23.03	DO 24.03	FR 25.03	SA 26.03	SO 27.03	MO 28.03	DI 29.03	MI 30.03	DO 31.03	FR 01.04	SA 02.04	SO 03.04
Meilensteine													M2								
Phase	Architektur-Phase						Architektur-Phase						Architektur-Phase								
Arbeit	Backend Architektur						User Frontend Architektur und GUI Entwurf						Management Frontend Architektur und GUI Entwurf								
Resultate													Finale Architektur								

	Woche 7						Woche 8						Woche 9								
	MO 04.04	DI 05.04	MI 06.04	DO 07.04	FR 08.04	SA 09.04	SO 10.04	MO 11.04	DI 12.04	MI 13.04	DO 14.04	FR 15.04	SA 16.04	SO 17.04	MO 18.04	DI 19.04	MI 20.04	DO 21.04	FR 22.04	SA 23.04	SO 24.04
Meilensteine							M3														
Phase	Entwicklungs-Phase						Entwicklungs-Phase						Entwicklungs-Phase								
Arbeit	Programmierung User Frontend						Programmierung Backend						Integration Frontend und Backend								
Resultate							Erster lauffähiger Prototyp (Frontend)														

	Woche 10							Woche 11							Woche 12						
	MO 25.04	DI 26.04	MI 27.04	DO 28.04	FR 29.04	SA 30.04	SO 01.05	MO 02.05	DI 03.05	MI 04.05	DO 05.05	FR 06.05	SA 07.05	SO 08.05	MO 09.05	DI 10.05	MI 11.05	DO 12.05	FR 13.05	SA 14.05	SO 15.05
Meilensteine	M4																				
Phase	Entwicklungs-Phase							Entwicklungs-Phase							Entwicklungs-Phase						
Arbeit	Programmierung Management Frontend							GUI Entwicklung							Reserve, Implementation von "Nice to Have" Features						
Resultate	Zwischenpräsentation Architekturdurchstich																				

	Woche 13							Woche 14							Woche 15						
	MO 16.05	DI 17.05	MI 18.05	DO 19.05	FR 20.05	SA 21.05	SO 22.05	MO 23.05	DI 24.05	MI 25.05	DO 26.05	FR 27.05	SA 28.05	SO 29.05	MO 30.05	DI 31.05	MI 01.06	DO 02.06	FR 03.06	SA 04.06	SO 05.06
Meilensteine															M5						
Phase	Test-Phase							Test-Phase							Test-Phase						
Arbeit	Test Cases erarbeiten und durchführen							Ausbesserungen und Optimierungen							Reserve						
Resultate								Stabile lauffähige Applikation auf allen Layern							Systemtests abgeschlossen						

	Woche 16							Woche 17						
	MO 06.06	DI 07.06	MI 08.06	DO 09.06	FR 10.06	SA 11.06	SO 12.06	MO 13.06	DI 14.06	MI 15.06	DO 16.06	FR 17.06	SA 18.06	SO 19.06
Meilensteine								M6						
Phase	Dokumentation							Dokumentation						
Arbeit	Fertigstellung der Dokumentation							Reserve						
Resultate	Abgabe Abstract							Abgabe und Präsentation des fertigen Projektes						

10.3.2 Risiko Analyse

Die folgende Tabelle zeigt die identifizierten Risiken, deren Wahrscheinlichkeit (nachfolgend W'keit genannt) und Auswirkungen. Diese Risiken wurden zu Beginn der Bachelorarbeit identifiziert und haben sich im Laufe der Arbeit nur marginal geändert. Die Farben kennzeichnen das entsprechende Risiko (siehe Risikomatrix).

Risiko	W'keit	Auswirkung	Massnahmen	Bemerkung
Verlust eines Teammitglieds	Unwahrscheinlich	Katastrophal	-	Projekt müsste abgebrochen werden
Projekt ist nicht gemäss Anforderungen umsetzbar	Vorstellbar	Kritisch	Es wurden Vorabklärungen getroffen um das zu prüfen	Die zu verwendenden Technologien sind bekannt
Projekt kann nicht in der Vorgegebenen Zeit beendet werden	Unwahrscheinlich	Kritisch	Anforderungen sind so gestellt, dass Prio1 sicher machbar ist	-
Überforderung des Teams in Bezug auf technisches Können	Unwahrscheinlich	Kritisch	-	Beides gute Studenten mit Vorwissen in dem Bereich
Defekt oder Verlust eine Entwickler-PCs	Selten	Unwesentlich	Alle Wichtigen Daten sind in der Cloud	Dropbox für Doku, TFS Online für Code

Die folgende Matrix diene als Basis für die Risikoevaluation in der oberen Tabelle. Sie definiert das Risiko basierend auf der Eintrittswahrscheinlichkeit und der Auswirkung eines Ereignisses.

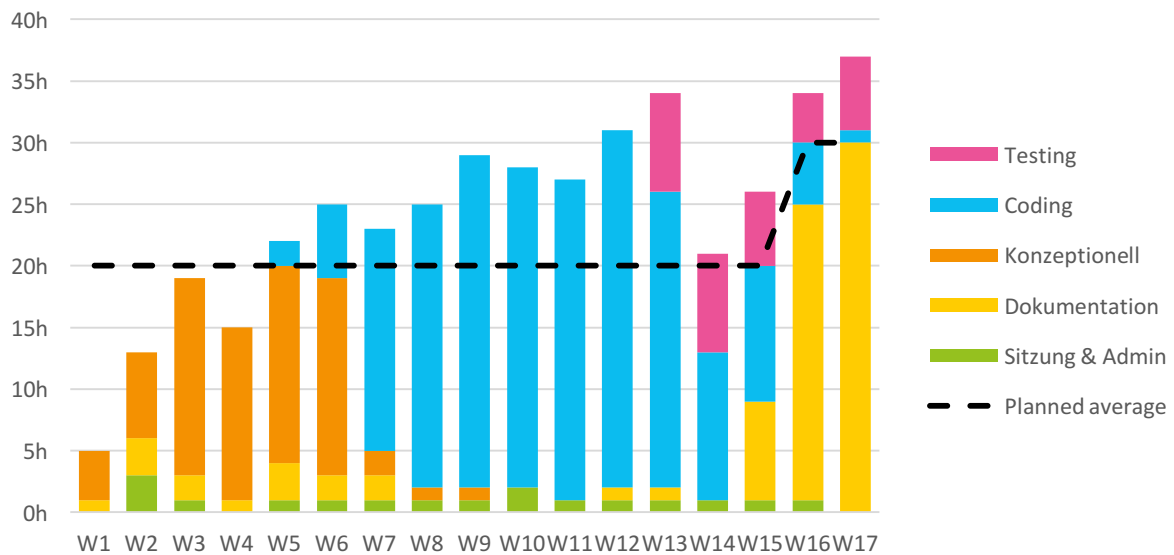
W'keit	Auswirkung			
	Unwesentlich	Gering	Kritisch	Katastrophal
Häufig	Hoch	Hoch	Extrem	Extrem
Wahrscheinlich	Mittel	Hoch	Hoch	Extrem
Vorstellbar	Mittel	Mittel	Hoch	Hoch
Unwahrscheinlich	Tief	Mittel	Mittel	Hoch
Selten	Tief	Tief	Mittel	Mittel

10.4 Zeit Management

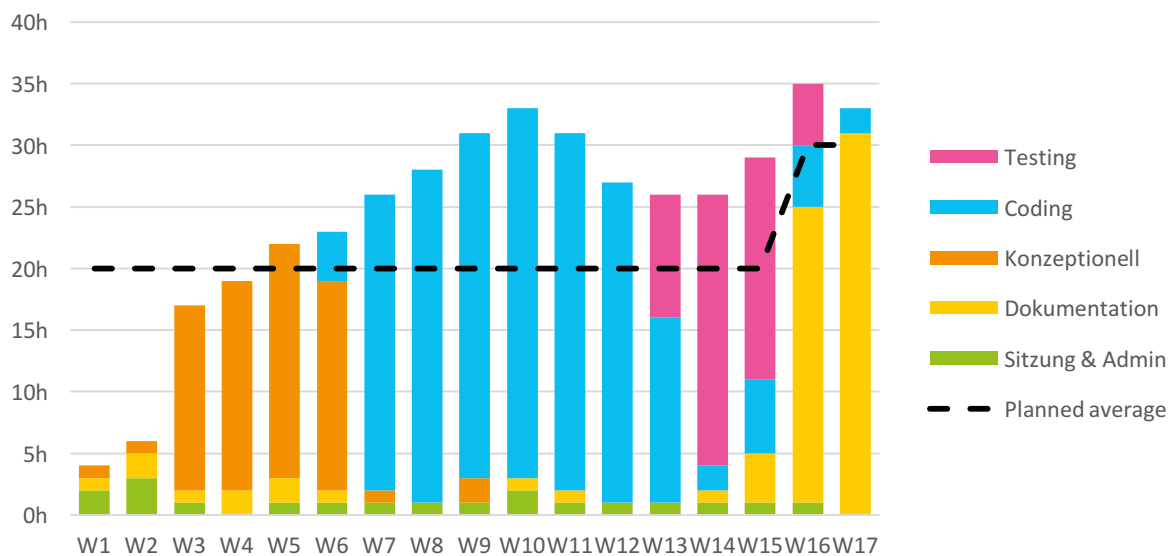
In diesem Kapitel sind die Zeiterfassungen die wir in den verschiedenen Kategorien gearbeitet haben dokumentiert.

	Andreas Stefanits	Fabian Schäfer
Erwartetes Stundentotal	360	360
Effektives Stundentotal	414	416
Zusätzliche Stunden	54	56

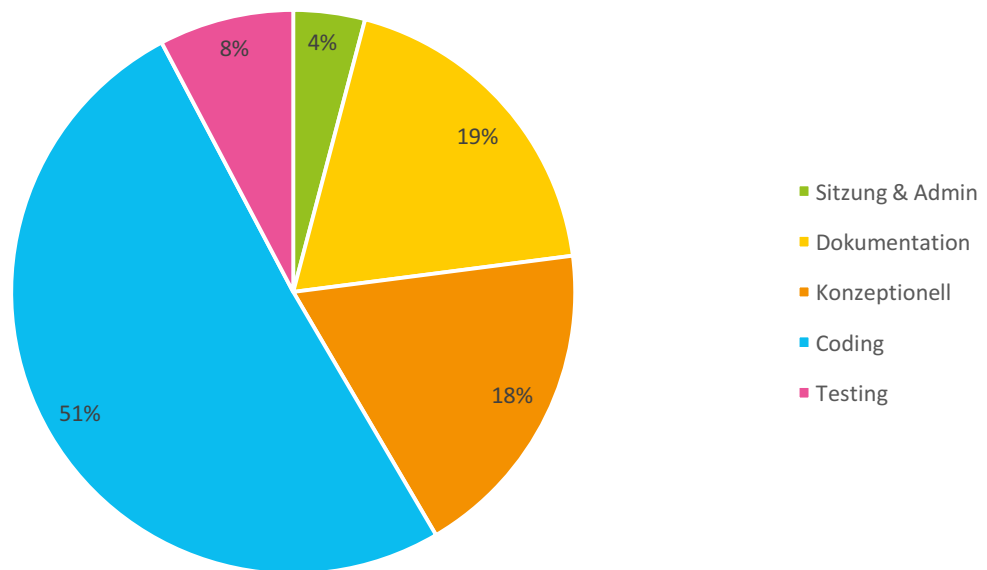
10.4.1 Zeit pro Woche Andreas Stefanits



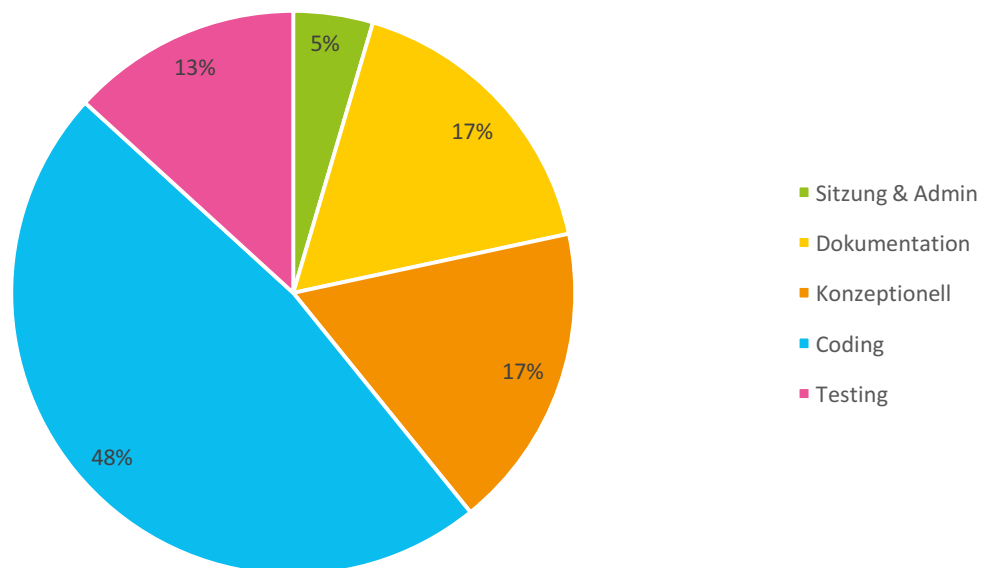
10.4.2 Zeit pro Woche Fabian Schäfer



10.4.3 Zeit pro Kategorie Andreas Stefanits



10.4.4 Zeit pro Kategorie Fabian Schäfer



10.5 Sitzungsprotokolle

Sitzungsprotokoll 1 (Kickoff)

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Abwesend

Andreas Stefanits (entschuldigt)

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-02-24, 17:00 – 17:45

Besprochene Punkte

- Aufgabenstellung durgeschaut und einige Punkte noch angepasst → Herr Huser schickt die finale Fassung
- Ziel ist es, code der nicht UI bezogen ist als Shared Code in einer portablen Library zu haben um die spätere Portierung auf Xamarin zu erleichtern
- Zwischenresultat (Architekturdurchstich) bis Ende April mit Zwischenpräsentation (die Studierenden organisieren den Termin)

Nächste Schritte

- Grobe Fassung des Projektplans
- Key für Azure Cloud wird von Herrn Huser an die Studierenden geschickt
- Sitzung jeweils mittwochs um 17:00 in Raum 8.242

Sitzungsprotokoll 2

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-03-02, 17:00 - 17:45

Besprochene Punkte

- Aufgabenstellung durgeschaut und einige Punkte noch angepasst → Herr Huser schickt die finale Fassung
- Ziel ist es, code der nicht UI bezogen ist als Shared Code in einer portablen Library zu haben um die spätere Portierung auf Xamarin zu erleichtern
- Zwischenresultat (Architekturdurchstich) bis Ende April mit Zwischenpräsentation (die Studierenden organisieren den Termin)

Nächste Schritte

- Review des Projektplans durch Herrn Huser und Herrn Jucker → Feedback mit Verbesserungsvorschlägen oder gewünschten Änderungen an uns. Finale Fassung bis zur nächsten Sitzung
- Finalisierung der Anforderungen und Spezifikationen mit anschliessendem Review. Finale Fassung bis zur nächsten Sitzung

Sitzungsprotokoll 3

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-03-09, 16:50 - 17:30

Besprochene Punkte

- Projektplan genehmigt
- Finale Anforderungen und Spezifikationen angeschaut und genehmigt
- Entwurf des Domain Models besprochen und Änderungsvorschläge aufgenommen

Nächste Schritte

- Nächste Woche fällt die Sitzung aus
- Finale Version der Architektur

Sitzungsprotokoll 4

Anwesend

Prof. Hansjörg Huser

Fabian Schäfer

Andreas Stefanits

Abwesend

Jürg Jucker

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-03-23, 17:10 - 17:45

Besprochene Punkte

- REST Definitionen besprochen
- Finales Klassendiagramm besprochen
- Entwurf des Paper Prototype besprochen

Nächste Schritte

- Technologie für Entwicklungsumgebung abklären (.NET Core?)

Sitzungsprotokoll 5

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-03-30, 17:00 - 18:00

Besprochene Punkte

- Finales Klassendiagramm erklärt und besprochen
- Paper Prototypes für Terminal und Admin App vorgestellt und besprochen
- REST Schnittstellendefinition besprochen
- Nächste Schritte und Termin für Zwischenpräsentation definiert. Die Zwischenpräsentation findet am 27.04.2016 in Raum 8.267 zusammen mit Herrn Rinkel statt.

Nächste Schritte

- Andreas Stefanits schickt Einladung für Zwischenpräsentation an alle beteiligten.

Sitzungsprotokoll 6

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-04-06, 17:00 - 17:30

Besprochene Punkte

- Kurzes Statusupdate: Entwicklung des Servers und Terminal-Clients begonnen, noch nicht viel zu zeigen

Nächste Schritte

- Prototyp vorbereiten

Sitzungsprotokoll 7

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-04-13, 17:00 - 17:30

Besprochene Punkte

- Kurzes Statusupdate: Entwicklung des Servers und Terminal-Clients ist am laufen
- Kurze Demo des aktuellen Stands des Terminal-Clients

Nächste Schritte

- Nächste Sitzung am Donnerstag 21. April 14:30 anstatt Mittwoch

Sitzungsprotokoll 8

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-04-21, 14:30 - 15:00

Besprochene Punkte

- Erste Version des Prototyps präsentiert

Nächste Schritte

- Zwischenpräsentation vorbereiten

Sitzungsprotokoll 9 (Zwischenpräsentation)

Anwesend

Prof. Hansjörg Huser

Andreas Rinkel

Fabian Schäfer

Andreas Stefanits

Abwesend

Jürg Jucker

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-04-27, 17:10 - 17:50

Besprochene Punkte

- Präsentation des aktuellen Status der Arbeit
- Präsentation der Architektur
- Demo des Prototyps
- Herr Rinkel möchte folgendes in der Doku sehen:
 - Big Picture der gesamten Arbeit
 - Lösungsansatz heruntergebrochen von gross nach klein
 - Systembeschreibung: Wer kommuniziert mit wem

Nächste Schritte

- 2-3 Wochen vor Abgabe können wir die Gliederung (Inhaltsverzeichnis) an Herrn Rinkel schicken um ein Feedback zu erhalten

Sitzungsprotokoll 10

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-05-04, 17:00 - 17:30

Besprochene Punkte

- Arbeit am Admin-Client begonnen, noch nicht viel zu zeigen

Nächste Schritte

- Admin-Client weiterentwickeln für kurze Präsentation nächste Woche

Sitzungsprotokoll 11

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-05-11, 17:00 - 17:30

Besprochene Punkte

- Kurze Präsentation der ersten Version des Admin-Clients

Nächste Schritte

- Admin-Client weiterentwickeln

Sitzungsprotokoll 12

Anwesend

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Prof. Hansjörg Huser

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-05-18, 17:00 - 17:30

Besprochene Punkte

- Präsentation der aktuellen Version des Admin-Clients

Nächste Schritte

- Admin-Client finalisieren

Sitzungsprotokoll 13

Anwesend

Prof. Hansjörg Huser

Jürg Jucker

Fabian Schäfer

Andreas Stefanits

Abwesend

Niemand

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-05-25, 17:00 - 17:30

Besprochene Punkte

- Umbau des Backends erläutert
- Vorläufig finale Version des Admin-Clients präsentiert
- Azure Pass abgelaufen → Neuen Zugriff angefordert

Nächste Schritte

- Erste Version des Abstracts verfassen
- Code Cleanup für Code Review nächste Woche

Sitzungsprotokoll 14

Anwesend

Prof. Hansjörg Huser

Fabian Schäfer

Andreas Stefanits

Abwesend

Jürg Jucker

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-06-01, 17:00 - 17:45

Besprochene Punkte

- Abstract besprochen und Verbesserungsvorschläge aufgenommen
- Inhalt der Doku besprochen und gemeinsam Inhaltsverzeichnis angeschaut
- Unit-Tests und Code angeschaut

Nächste Schritte

- Abstract überarbeiten
- Inhaltsverzeichnis an Herrn Rinkel schicken für Feedback von seiner Seite
- Herr Huser und Herr Jucker schauen sich den Code an
- Herr Huser organisiert Windows 10 Tablet

Sitzungsprotokoll 15

Anwesend

Prof. Hansjörg Huser

Fabian Schäfer

Andreas Stefanits

Abwesend

Jürg Jucker

Protokollführer

Fabian Schäfer

Ort, Datum und Uhrzeit

Rapperswil (Raum 8.242), 2016-06-08, 17:00 - 17:30

Besprochene Punkte

- Finale Version des Abstracts angeschaut
- Neue Hardware (Raspberry Pi, Display, Windows 10 Tablet) für Präsentation übergeben

Nächste Schritte

- Doku fertigstellen
- Abstract einreichen
- Poster erstellen
- Code finalisieren
- Die letzte Sitzung nächste Woche fällt aus

10.6 Benutzerhandbücher

10.6.1 Terminal-Client

Installationsanleitung

1. Beim ersten Start des Terminal-Clients muss dieser Konfiguriert werden. Der Terminal-Client zeigt folgendes Fenster an

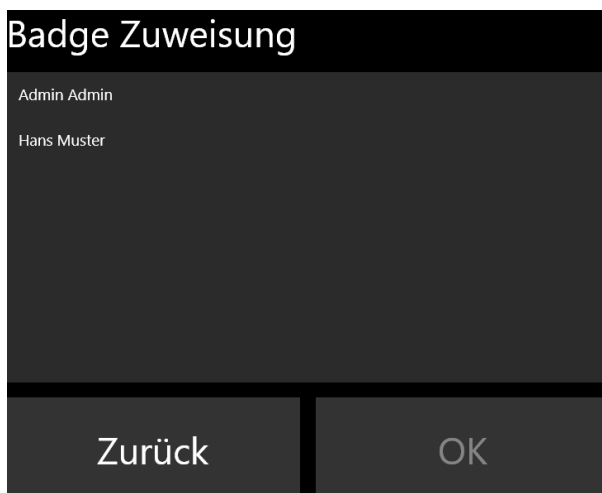


Einstellungen
Server URL:
https://apze.azurewebsites.net/api/
E-Mail Adresse:
terminal@apze.local
Passwort:
•••
Anmelden

2. Im Konfigurationsfenster die Mail-Adresse und das Passwort des Terminal Accounts sowie die Adresse des Servers (https) eintragen.

Badge Zuweisung

1. Den Badge welcher einem Mitarbeiter zugewiesen werden soll an den NFC-Leser halten.
2. Der Terminal-Client zeigt folgendes Fenster an, welches alle Mitarbeiter anzeigt welche keinen Badge zugewiesen haben.

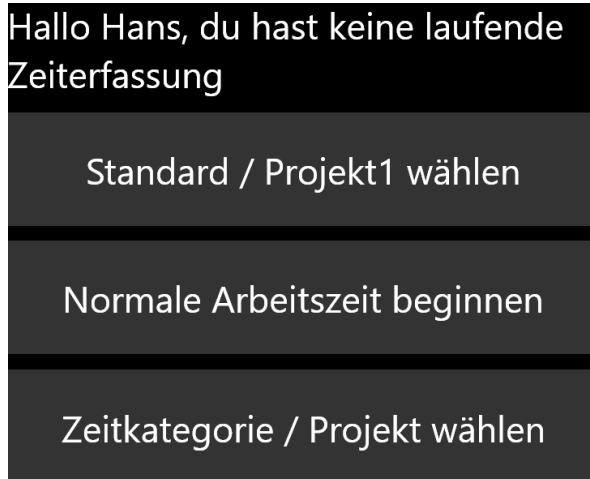


Badge Zuweisung
Admin Admin
Hans Muster
Zurück OK

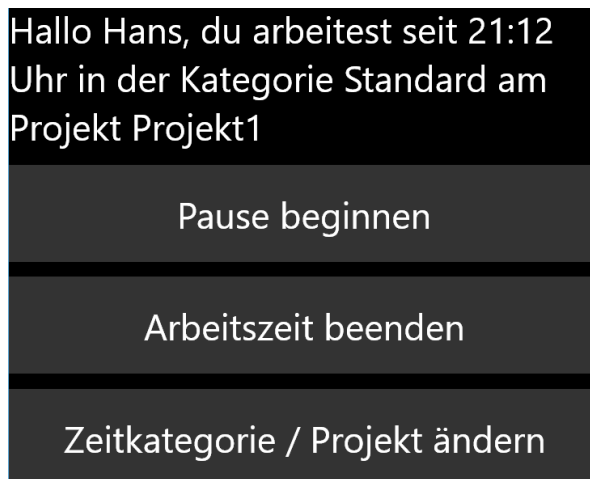
3. Um den Badge zuzuweisen den gewünschten Benutzer auswählen und mit «OK» bestätigen.

Start Screens

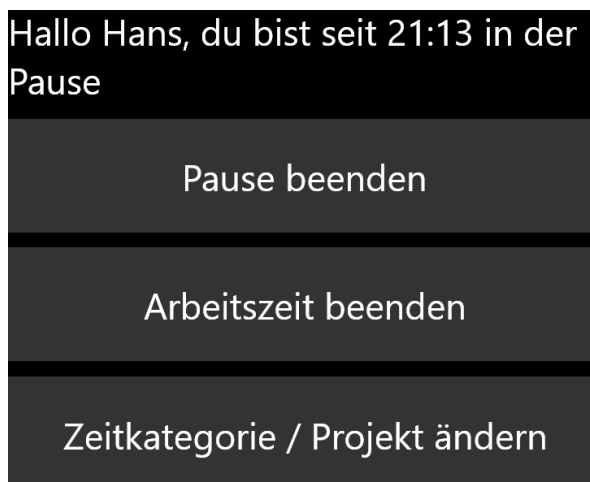
1. Nachdem der Mitarbeiter den Badge an den NFC-Leser gehalten hat wird einer der folgenden Screens angezeigt, je nach Status der Zeiterfassung.
2. Start Screen, wenn keine laufende Zeiterfassung vorhanden ist



3. Start Screen bei laufender Zeiterfassung



4. Start Screen bei laufender Pause



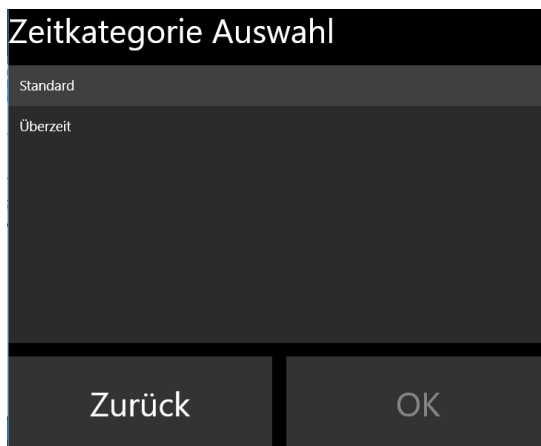
5. Wird «Pause beenden» oder «Arbeitszeit beenden» gewählt wird eines der folgenden Bestätigungsfenster angezeigt

Deine Arbeitszeiterfassung wurde erfolgreich pausiert!

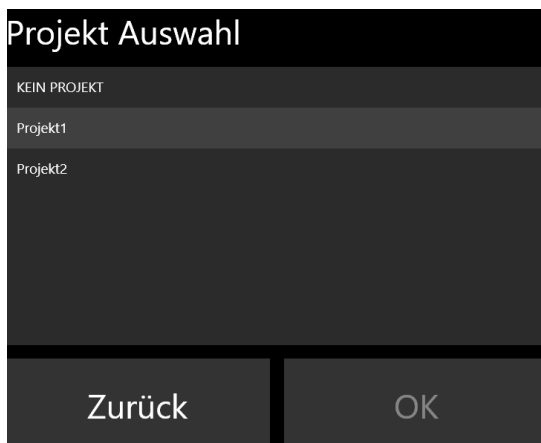
Deine Arbeitszeiterfassung wurde erfolgreich beendet!

Zeitkategorie / Projekt ändern

1. Wählt der Benutzer «Zeitkategorie / Projekt ändern» wird das folgende Zeitkategorie Auswahl Fenster angezeigt



2. Nach Auswahl der Zeitkategorie und betätigen des «OK»-Knopfes wird auf die Projekt Auswahl weitergeleitet.



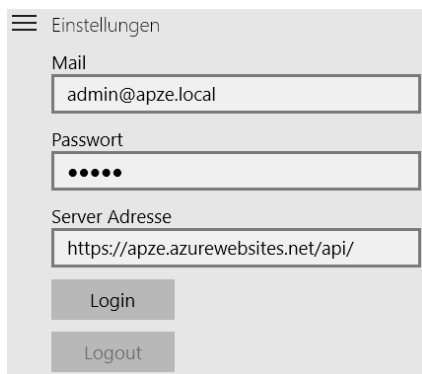
3. Hier kann optional ein Projekt ausgewählt werden, soll keine Projektzeiterfassung laufen kann «KEIN PROJEKT» ausgewählt werden.
4. Nach betätigen des «OK»-Knopfes wird das Bestätigungsfenster mit den ausgewählten Einstellungen angezeigt.

**Deine Arbeitszeiterfassung wurde erfolgreich gestartet!
Du arbeitest in der Zeitkategorie Standard am Projekt Projekt1**

10.6.2 Admin-Client

Installationsanleitung

1. Das Gerät auf welchem der Admin-Client installiert werden soll muss zuerst in den Development Mode gesetzt werden, damit wird erlaubt das Windows Universal Apps manuell installiert werden dürfen.
2. Den Ordner "Admin-Client" lokal auf das Gerät kopieren und das PowerShell-Skript "Add-AppDevPackage.ps1" ausführen.
3. Beim ersten Start des Admin-Clients muss dieser Konfiguriert werden. Der Admin-Client zeigt folgendes Fenster an:

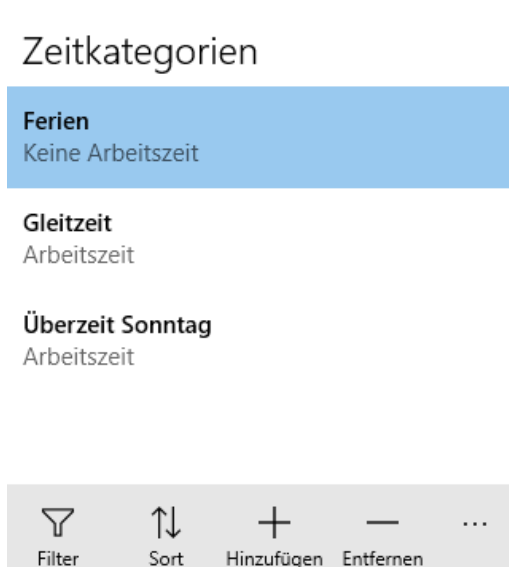


The screenshot shows a configuration window titled "Einstellungen" (Settings). It has a hamburger menu icon in the top left. The window contains three input fields: "Mail" with the text "admin@apze.local", "Passwort" (Password) with four dots, and "Server Adresse" (Server Address) with the text "https://apze.azurewebsites.net/api/". Below the input fields are two buttons: "Login" and "Logout".

4. Im Konfigurationsfenster die Mail-Adresse und das Passwort eines Admin Accounts sowie die Adresse des Servers (https) eintragen. Der Standard Admin Account hat folgendes Login, welches nach der ersten Anmeldung geändert werden sollte:
 - Mail: admin@apze.local
 - Passwort: admin
5. Damit der Admin bearbeitet werden kann muss dieser einer Position zugewiesen werden. Dies ist notwendig damit alle Funktionen korrekt funktionieren.

Zeitkategorien

1. Für die Verwaltung der Zeitkategorien den entsprechenden Reiter auswählen
2. Im linken Bereich der Applikation sind alle Zeitkategorien und die Funktionen um diese zu erstellen und zu löschen aufgelistet:



3. Wird eine neue Zeitkategorie erstellt oder eine bestehende ausgewählt öffnet sich auf der rechten Seite das Formular für die Zeitkategorie. Um eine bestehende Kategorie zu bearbeiten muss der Bearbeitungsmodus im unteren rechten Bereich aktiviert werden. Am selben Ort befinden sich auch das Speichern und Abbrechen der Bearbeitung:

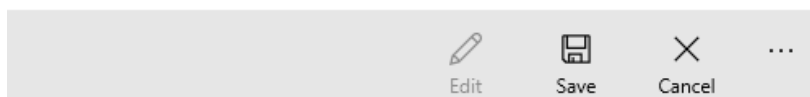
Zeitkategorie Details Gleitzeit

Zeitkategorienname

Zeit-Multiplikator

Zeit-Typ

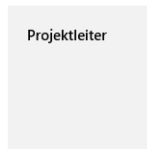
[Positionen zur Zeitkategorie hinzufügen](#)



4. Um einer Zeitkategorie mehrere Positionen zuzuweisen muss die Zeitkategorie ausgewählt und auf "Positionen zur Zeitkategorie hinzufügen" geklickt werden

5. Im oberen Bereich sind alle Positionen welche der Zeitkategorie nicht zugewiesen sind, im unteren Bereich sind die zugewiesenen. Um eine Position hinzuzufügen oder zu entfernen muss die Position ausgewählt und der entsprechende Knopf geklickt werden:

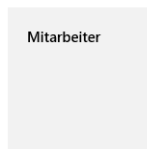
Nicht zugewiesene Positionen



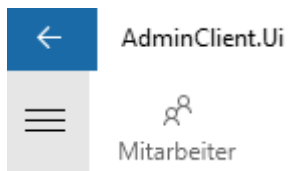
Position Hinzufügen

Position Entfernen

Zugewiesene Positionen



6. Mit einem Klick auf den oberen linken Pfeil gelangt man wieder zurück auf die Detail-Ansicht der Zeitkategorie:



Positionen

1. Für die Verwaltung der Positionen den entsprechenden Reiter auswählen
2. Im linken Bereich der Applikation sind alle Positionen und die Funktionen um diese zu erstellen und zu löschen aufgelistet:

Positionen

Mitarbeiter

Zeitkategorie Gleitzeit

Projektleiter

Zeitkategorie Gleitzeit



3. Wird eine neue Position erstellt oder eine bestehende ausgewählt öffnet sich auf der rechten Seite wieder das Formular für die Bearbeitung:

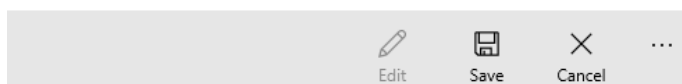
Position Details Projektleiter

Positionsname

Standard Zeitkategorie

Soll-Arbeitszeit

[Zeitkategorien zu Position hinzufügen](#)

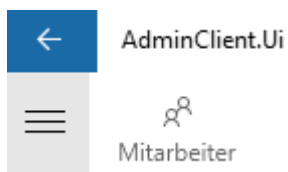


4. Um einer Position mehr Zeitkategorien zuzuweisen muss die Zeitkategorie ausgewählt und auf "Positionen zur Zeitkategorie hinzufügen" geklickt werden

5. Im oberen Bereich sind alle Zeitkategorien welche der Position nicht zugewiesen sind, im unteren Bereich sind die zugewiesenen. Um eine Zeitkategorie hinzuzufügen oder zu entfernen muss die Zeitkategorie ausgewählt und der entsprechende Knopf geklickt werden:

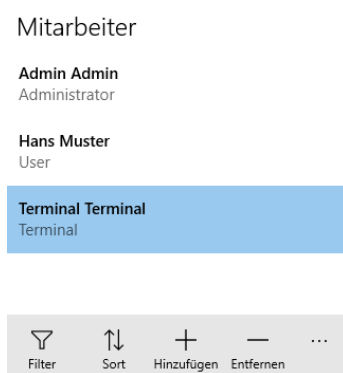


6. Mit einem Klick auf den oberen linken Pfeil gelangt man wieder zurück auf die Detail-Ansicht der Position:



Mitarbeiter

1. Für die Verwaltung der Mitarbeiter den entsprechenden Reiter auswählen
2. Im linken Bereich der Applikation sehen sie alle erstellten Mitarbeiter und die Funktionen um diese zu erstellen und löschen.



3. Wird ein neuer Mitarbeiter erstellt oder ein bestehender ausgewählt öffnet sich auf der rechten Seite das Formular für den Mitarbeiter. Um ein bestehender Mitarbeiter zu bearbeiten muss zuerst der Bearbeitungsmodus aktiviert werden.

Mitarbeiter Details Hans Muster

Vorname

Nachname

Mail Adresse

Neues Passwort

Arbeitspensum

Sicherheitsrolle

Position

Austrittsdatum

Badge-ID **Kein Badge registriert**

[Mitarbeiter zu Projekten zuweisen](#)

[Zeiterfassungen bearbeiten](#)



4. Um dem Mitarbeiter Projekt zuzuweisen wählen sie den Mitarbeiter aus und klicken sie auf «Mitarbeiter zu Projekten zuweisen».

Nicht zugewiesene Projekte

Projekt2

Startdatum:

Enddatum:

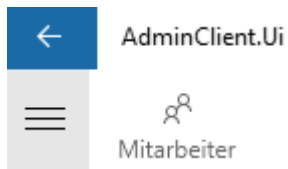
Projekt Hinzufügen

Projekt Entfernen

Zugewiesene Projekte

Projekt1
10.06.2016
29.06.2016

5. Im oberen Bereich sind alle Projekte welche nicht dem Mitarbeiter zugewiesen wurden, im unteren Bereich sind die zugewiesenen. Um ein Projekt hinzuzufügen oder zu entfernen wählen sie das Projekt aus und klicken sie auf den entsprechenden Knopf. Für das Hinzufügen wird noch das Start- und Enddatum benötigt.
6. Um wieder auf die Detail-Ansicht des Mitarbeiters zu kommen klicken sie oben links auf den Pfeil.



7. Damit ein gelöschter Mitarbeiter wieder reaktiviert werden kann, setzten sie den Filter damit gelöschte Mitarbeiter angezeigt werden. Wurde der Filter gesetzt wählen sie den entsprechenden Mitarbeiter aus und löschen das Austrittsdatum.

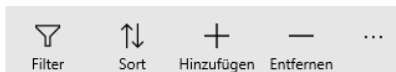
Projekte

1. Für die Verwaltung der Projekte den entsprechenden Reiter auswählen
2. Im linken Bereich der Applikation sehen sie alle erstellten Projekte und die Funktionen um diese zu erstellen und löschen.

Projekte

Projekt1
Projekt Nr. 123

Projekt2
Projekt Nr. 1

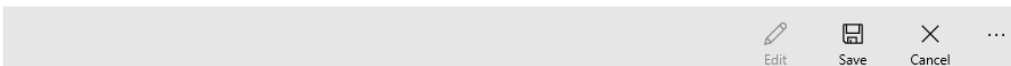


3. Wird ein neues Projekt erstellt oder ein bestehender ausgewählt öffnet sich auf der rechten Seite das Formular für das Projekt. Um ein bestehendes Projekt zu bearbeiten muss zuerst der Bearbeitungsmodus aktiviert werden.

Projekt Details Projekt1

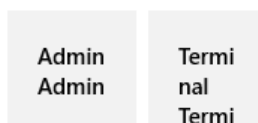
Projektname	<input type="text" value="Projekt1"/>
Projektnummer	<input type="text" value="123"/>
Startdatum	<input type="text" value="10.06.2016"/>
Enddatum	<input type="text" value="23.06.2016"/>

[Mitarbeiter zu Projekt hinzufügen](#)



4. Um Mitarbeiter zu einem Projekt hinzuzufügen wählen sie das Projekt aus und klicken sie auf «Mitarbeiter zu Projekt hinzufügen».

Nicht zugewiesene Mitarbeiter



Startdatum:

Enddatum:

Mitarbeiter Hinzufügen

Mitarbeiter Entfernen

Zugewiesene Mitarbeiter



5. Im oberen Bereich sind alle Mitarbeiter welche nicht dem Projekt zugewiesen wurden, im unteren Bereich sind die zugewiesenen. Um ein Mitarbeiter hinzuzufügen oder zu entfernen wählen sie die Mitarbeiter aus und klicken sie auf den entsprechenden Knopf. Für das Hinzufügen wird noch das Start- und Enddatum benötigt.
6. Um wieder auf die Detail-Ansicht des Projektes zu kommen klicken sie oben links auf den Pfeil.



7. Damit ein gelöschttes Projekt wieder reaktiviert werden kann, setzten sie den Filter damit gelöschte Projekte angezeigt werden. Wurde der Filter gesetzt wählen sie das entsprechende Projekt aus und setzten das Enddatum in die Zukunft.

Zeiterfassung bearbeiten

1. Für die Verwaltung der Mitarbeiter den entsprechenden Reiter auswählen
2. Als nächstes klicken sie auf «Zeiterfassung bearbeiten»

Zeiterfassung Hans Muster

Mittwoch, 8. Juni 2016
Gleitzeit

Freitag, 10. Juni 2016
Gleitzeit

Freitag, 10. Juni 2016
Gleitzeit

Zeiterfassung Details Mittwoch, 8. Juni 2016

Von **08.06.2016** **15:58**

Bis **08.06.2016** **15:58**

Position **Mitarbeiter**

Zeitkategorie **Gleitzeit**

[Pausen bearbeiten](#)

[Projektzeiten bearbeiten](#)

⏮ ⏪ ⏩ ⏭ ⋮

✎ 📁 ✕ ⋮

3. Auf der linken Seite werden alle Zeiterfassungen des Mitarbeiters aufgelistet. Auch hier sind im unteren Bereich die das Hinzufügen und löschen zu finden. Hilfreich sind die Filter welche das Zeitfenster der Zeiterfassungen einschränken lässt.
4. Um alle Projektzeiten oder Pausen für diese Zeiterfassung anzuzeigen klicken sie auf «Pausen bearbeiten» oder «Projektzeiten bearbeiten».
5. Diese funktionieren gleich wie die Zeiterfassung, auf der linken Seite sind alle Pausen bzw. Projektzeiten und rechts das Formular.

Pausen am 08.06.2016

15:58 Uhr bis 16:00 Uhr

Pause Details 15:58 Uhr bis 16:00 Uhr

Von **08.06.2016** **15:58**

Bis **08.06.2016** **16:00**

⏮ ⏪ ⏩ ⏭ ⋮

✎ 📁 ✕ ⋮

Projektzeiten am 08.06.2016

Mittwoch, 8. Juni 2016
Projekt1

Projekt-Zeiterfassung Details Mittwoch, 8. Juni 2016

Von **08.06.2016** **15:58**

Bis **08.06.2016** **16:58**

Projekt **Projekt1**

⏮ ⏪ ⏩ ⏭ ⋮

✎ 📁 ✕ ⋮

10.7 Testbericht

10.7.1 Admin-Client

Test #	Erhaltenes Resultat	Passed
1	Der Admin-Client starte und zeigt den Loginmenu an. Der Login funktioniert und das Loginmenu wird geschlossen	✓
2	Es werden keine Zeitkategorien angezeigt, Der rechte Teil des Tools zeigt das leere Formular für eine Zeitkategorie an, Die Zeitkategorie wurde gespeichert und wird in der Liste angezeigt	✓
3	Es wird die Zeitkategorie «Standard Kategorie» angezeigt, Es werden die Informationen für die Zeitkategorie «Standard» angezeigt, Die Zeitkategorie wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓
4	Es werden keine Positionen angezeigt, Der rechte Teil des Tools zeigt das leere Formular für eine Position an, Die Position wurde gespeichert und wird in der Liste angezeigt	✓
5	Es wird die Position «Mitarbeiter» angezeigt, Es werden die Informationen für die Position «Mitarbeiter» angezeigt, Die Zeitkategorie wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓
6	Es wird nur der Admin angezeigt in der Liste, Es wird das leere Formular für einen Mitarbeiter angezeigt, Der erstellte Mitarbeiter wird in der Liste angezeigt	✓
7	Es werden zwei Benutzer in der Liste angezeigt, Es werden die Informationen von «Hans Muster1» angezeigt, Die Felder werden für die Bearbeitung aktiviert, Der Mitarbeiter wird aktualisiert und in der Liste mit dem neuen Namen dargestellt	✓
8	Es werden zwei Benutzer in der Liste angezeigt, Es werden die Informationen von «Hans Muster» angezeigt, Der Mitarbeiter wurde gelöscht und wird nicht mehr in der Liste angezeigt	✓
9	Es wird nur der Administrator in der Liste angezeigt, Der Mitarbeiter «Hans Muster» wird in der Liste angezeigt, Die Informationen vom Mitarbeiter «Hans Muster» werden angezeigt, Die Felder werden für die Bearbeitung aktiviert, Der Mitarbeiter wird gespeichert und das Zeichen für gelöscht ist nicht mehr sichtbar, Es werden zwei Mitarbeiter angezeigt	✓
10	Es werden keine Projekte angezeigt, Der rechte Teil des Tools zeigt das leere Formular für ein Projekt an, Das Projekt wurde gespeichert und wird in der Liste angezeigt	✓

Test #	Erhaltenes Resultat	Passed
11	Es wird das Projekt «TestProjekt1» angezeigt, Der rechte Teil des Tools werden die Informationen für das Projekt angezeigt, Das Projekt wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓
12	Es wird das Projekt «TestProjekt» angezeigt, Es werden die Informationen für das Projekt «TestProjekt» angezeigt, Das Projekt wurde gelöscht und wird nicht mehr in der List angezeigt	✓
13	Es wird kein Projekt angezeigt, Es wird das Projekt «TestProjekt» angezeigt, Es werden die Informationen für das Projekt «TestProjekt» angezeigt, Das Projekt wurde gespeichert, Das Projekt «TestProjekt» wird angezeigt	✓
14	Es wird die Position «Projektleiter» angezeigt, Es werden die Detailinformationen der Position «Projektleiter» angezeigt, Es wird die Zeitkategorie «Standard» als zugewiesene Zeitkategorien angezeigt, In der Liste der nicht zugewiesenen wird die Zeitkategorie «Überzeit» angezeigt, Die Zeitkategorie «Überzeit» wechselt die Liste	✓
15	Es wird die Position «Projektleiter» angezeigt, Es werden die Detailinformationen der Position «Projektleiter» angezeigt, Es werden keine nicht zugewiesenen Zeitkategorien angezeigt, In der Liste der zugewiesenen wird die Zeitkategorien «Standard» und «Überzeit» angezeigt, Die Zeitkategorie «Überzeit» wechselt die Liste	✓
16	Es wird die Zeitkategorien «Standard» und «Überzeit» angezeigt, Es werden die Detailinformationen der Zeitkategorie «Überzeit» angezeigt, Es werden keine zugewiesenen Positionen angezeigt, In der Liste der nicht zugewiesenen wird die Position «Projektleiter» angezeigt, Die Position «Projektleiter» wechselt die Liste	✓
17	Es wird die Zeitkategorien «Standard» und «Überzeit» angezeigt, Es werden die Detailinformationen der Zeitkategorie «Überzeit» angezeigt, Es werden keine nicht zugewiesenen Positionen angezeigt, In der Liste der zugewiesenen wird die Position «Projektleiter» angezeigt, Die Position «Projektleiter» wechselt die Liste	✓
18	Es wird das Projekt «TestProjekt» angezeigt, Es werden die Detailinformationen des Projektes «TestProjekt» angezeigt, Es werden keine zugewiesenen Mitarbeiter angezeigt, In der Liste der nicht zugewiesenen wird der Mitarbeiter «Hans Muster» angezeigt, Der Mitarbeiter «Hans Muster» wechselt die Liste	✓

Test #	Erhaltenes Resultat	Passed
19	Es wird das Projekt «TestProjekt» angezeigt, Es werden die Detailinformationen des Projektes «TestProjekt» angezeigt, Es werden keine nicht zugewiesenen Mitarbeiter angezeigt, In der Liste der zugewiesenen wird der Mitarbeiter «Hans Muster» angezeigt, Der Mitarbeiter «Hans Muster» wechselt die Liste	✓
20	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden keine zugewiesenen Projekte angezeigt, In der Liste der nicht zugewiesenen wird das Projekt «TestProjekt» angezeigt, Das Projekt «TestProjekt» wechselt die Liste	✓
21	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden keine nicht zugewiesenen Projekte angezeigt, In der Liste der zugewiesenen wird das Projekt «TestProjekt» angezeigt, Das Projekt «TestProjekt» wechselt die Liste	✓
22	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es wird das leere Formular für die Zeiterfassung angezeigt, Die Zeiterfassung wurde gespeichert und wird in der Liste angezeigt	✓
23	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Der Bearbeitungsmodus wurde aktiviert, Die Zeiterfassung wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓
24	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Pausen angezeigt, Es wird das leere Formular für das erfassen eine Pause angezeigt, Die Pause wurde gespeichert und wird in der Liste angezeigt	✓
25	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Pausen angezeigt, Es werden die Detailinformationen der Pause angezeigt, Der Bearbeitungsmodus wurde aktiviert, Die Pause wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓

Test #	Erhaltenes Resultat	Passed
26	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Pausen angezeigt, Es werden die Detailinformationen der Pause angezeigt, Die Pause wurde gelöscht und wird nicht mehr angezeigt	✓
27	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Projektzeiten angezeigt, Es wird das leere Formular für das Erfassen einer Projektzeit angezeigt, Die Projektzeit wurde gespeichert und wird in der Liste angezeigt	✓
28	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Projektzeiten angezeigt, Es werden die Detailinformationen der Projektzeiterfassung angezeigt, Der Bearbeitungsmodus wurde aktiviert, Die Projektzeit wurde gespeichert und wird in der Liste mit den neuen Informationen angezeigt	✓
29	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Es werden alle Projektzeiten angezeigt, Es werden die Detailinformationen der Projektzeiterfassung angezeigt, Der Projektzeiterfassung wurde gelöscht	✓
30	Es wird der Mitarbeiter «Hans Muster» angezeigt, Es werden die Detailinformationen des Mitarbeiters «Hans Muster» angezeigt, Es werden alle Zeiterfassungen angezeigt, Es werden die Detailinformationen der Zeiterfassung angezeigt, Die Zeiterfassung wurde gelöscht	✓
31	Es wird die Zeitkategorie «Standard» angezeigt, Es werden die Informationen für die Zeitkategorie «Standard» angezeigt, Die Zeitkategorie wurde gelöscht und wird nicht mehr in der List angezeigt	✓
32	Es wird die Position «Projektleiter» angezeigt, Es werden die Informationen für die Position «Projektleiter» angezeigt, Die Position wurde gelöscht und wird nicht mehr in der List angezeigt	✓

10.7.2 Terminal-Client

Test #	Erhaltenes Resultat	Passed
1	Login erfolgreich, Startscreen wird angezeigt	✓
2	Die Zuweisung war erfolgreich und der Benutzer wird angemeldet	✓
3	Der Bestätigungsscreen zeigt die richtigen Informationen an	✓
4	Der Bestätigungsscreen zeigt an dass die Pause begonnen wurde	✓
5	Der Bestätigungsscreen zeigt an dass die Pause beendet wurde	✓
6	Der Bestätigungsscreen zeigt die neue Zeitkategorie an	✓
7	Der Bestätigungsscreen zeigt an dass die Arbeitszeit beendet wurde	✓
8	Der Bestätigungsscreen zeigt die letzten Einstellungen an	✓
9	Der Bestätigungsscreen zeigt dass Standard gestartet wurde	✓
10	Der Bestätigungsscreen zeigt dass das Projekt gestartet wurde	✓