

digitalSTROM Windows 8 App

Bachelorarbeit

Abteilung Informatik

Hochschule für Technik Rapperswil

Herbstsemester 2013

Autoren: Richard Schulthess, Annrita Egli
Betreuer: Professor Hansjörg Huser, Franziska Schäpper
Projektpartner: Noser Engineering AG
Experte: Stefan Zettel, Ascentive Zürich
Gegenleser: Professor Andreas Rinkel

Abstract

Ausgangslage:

Die Firma aizo AG verkauft digitalSTROM Einrichtungen und bietet diese als komfortabel und einfach zu bedienen an. Die Stromgeräte eines Hauses können über ein Webinterface eingestellt und gesteuert werden. Auch über Schalter im Hause ist die direkte Steuerung möglich. Für digitalSTROM Kunden ist es aber am einfachsten, ihre Einrichtungen mit einem Tablet über eine App zu steuern. Da bisher keine solche App für Windows-Tablets existierte, war die Zielsetzung der Arbeit eine Windows 8.1 Store App zu entwickeln.

Vorgehen/Technologien:

Mit den im Development Kit enthaltenen Komponenten von aizo AG konnte eine digitalSTROM Einrichtung simuliert und getestet werden. Über einen Login von aizo AG wurde die Kommunikation mit dem Gerät hergestellt, um seine Funktionen mit der App steuern zu können. Es galt dabei die Fülle an Kombinationen, die durch verschiedene Räume und deren voreingestellten Stimmungen entstehen, in eine übersichtliche Darstellung zu bringen. Befehle an das digitalSTROM Gerät, für Ausführungen wie zum Beispiel das Einschalten einer Stimmung, werden über einen REST-Webservice gesendet. Als Entwicklungsumgebung wurden die neusten Microsofttechnologien wie Visual Studio 2013, Blend 2013 und ReSharper 8.0.2 eingesetzt.

Ergebnis:

Die digitalSTROM Windows 8.1 App ermöglicht den Besitzern von digitalSTROM Einrichtungen ein einfaches Ein- und Umstellen ihrer Stromgeräte mit dem Windows Tablet. Die App verfügt über die bekannte Windows App Oberfläche und Navigation, womit ihre Benutzerfreundlichkeit gewährleistet ist. Sobald der User auf der App eingeloggt ist, kann er mit wenigen Klicks seine gewünschten Stimmungen in den Räumen einstellen oder Befehle für das gesamte Haus aussenden. Als zusätzliches Feature können Stimmungen mittels einer Zeitschaltuhr zu bestimmten Zeiten eingeschaltet werden.

Arbeitsaufteilung

Leistung	Wer	Funktion
UI und Navigation Photo Page	Richard Schulthess	Implementation und Dokumentation
Core und Gerätesteuerung Zeitschaltuhr	Annrita Egli	Implementation und Dokumentation
Inputs für Appoberfläche und Funktionen	aizo AG	Projektpartner
Code Review	Noser AG	Projektpartner
Betreuung während der Arbeit	Prof. H. Huser	Verantwortlicher Professor
Betreuung und praktische Hilfe während der Arbeit	Franziska Schäpper	Assistentin des verantwortlichen Professors

Erklärung

Ich erkläre hiermit,

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

Ort, Datum:

Name, Unterschrift:

Ich erkläre hiermit,

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

Ort, Datum:

Name, Unterschrift:

Management Summary

Ausgangslage

Im Wohnen der Zukunft sind alle Stromverbraucher innerhalb eines Hauses oder einer Wohnung intelligent miteinander vernetzt. Die Schweizer Firma aizo AG hat das System digitalSTROM entwickelt, das die bereits bestehende Strominfrastruktur für die Vernetzung nutzen kann. Dafür müssen die Geräte mit einem integrierten Chip versehen sein oder es muss vor jedem Gerät eine Miniklemme mit eingebautem Chip montiert werden. Mit zusätzlichen Chips können problemlos weitere Geräte angeschlossen und untereinander verbunden werden. Diese Chips messen den Stromverbrauch, regeln die Funktionen der Stromgeräte und kommunizieren mit einem Server. Dieser Server verfügt über eine offene Schnittstelle, die es ermöglicht, neue Anwendungen zu programmieren.

Die elektrischen Geräte eines Hauses können über ein Webinterface über die Schnittstelle des digitalSTROM Servers eingestellt und gesteuert werden. Auch über Schalter im Hause ist die Steuerung möglich. Für die digitalSTROM Kunden ist es aber am einfachsten, ihre Einrichtungen mit einem Tablet über ein App zu steuern. Da bisher keine solche App für Windows Tablets existierte, war die Zielsetzung dieser Arbeit eine Windows 8.1 Store App zu entwickeln.

Vorgehen, Technologien

Um mit der Entwicklung der App beginnen zu können, hat aizo für Entwickler ein Developer Kit zusammengestellt. Dieses besteht aus einem Sicherungskasten mit FI-Schalter, einem eingebauten digitalSTROM Server mit Ethernetanschluss, einem digitalSTROM-Meter, der den Stromverbrauch misst und drei digitalSTROM Klemmen. Damit konnte eine mit digitalSTROM ausgerüstete Wohnung simuliert und getestet werden.

Mit dem Webinterface des digitalSTROM Servers kann man verschiedene Voreinstellungen programmieren. So konnte die Kommunikation von der App zum Server nachverfolgt werden, wie zum Beispiel das Einschalten einer Stimmung. Die Verbindung ist über SSL verschlüsselt und arbeitet mit dem kompakten Datenformat JSON.

Die App läuft auf Windows RT Tablets oder Desktop Computern mit Windows 8.1. Um die App benutzerfreundlich zu gestalten, galt es die Fülle an Kombinationen, die durch verschiedene Räume und deren voreingestellten Stimmungen entstehen, in eine übersichtliche Darstellung zu bringen.

Ergebnisse

Das digitalSTROM Windows 8.1 App ermöglicht den Besitzern von digitalSTROM Einrichtungen ein einfaches Ein- und Umstellen ihrer Stromgeräte mit dem Windows Tablet. Die App verfügt über die bekannte Windows App Oberfläche und Navigation, womit die digitalSTROM Einrichtungen benutzerfreundlich gesteuert werden können. Ist der User auf der App eingeloggt, kann er mit wenigen Klicks seine gewünschten Stimmungen in den Räumen einstellen. Stimmungen, die öfters ausgewählt werden, können den Favoriten hinzugefügt werden, damit sie nicht zuerst im entsprechenden Raum gesucht werden müssen. Es ist auch möglich Befehle, welche das gesamte Haus betreffen, wie zum Beispiel alle elektrischen Verbraucher auszuschalten oder die Handlung „Nach Hause kommen“ auszuführen, damit alle Geräte so eingeschaltet werden, wie man es sich beim nach Hause kommen wünscht. Die App gibt auch einen Überblick über den Stromverbrauch. Jeder im Haus vorhandene Schaltkreis zeigt den momentanen sowie den gesamten Stromverbrauch an.

Mit Hilfe der Kamera am Windows Tablet können Bilder aufgenommen und einem Raum oder einer Stimmung zugeordnet werden. Die Räume und Stimmungen können dort vom User umbenannt werden. So erhält die App ein sehr individuelles und ans Haus angepasstes Aussehen.

Anstelle der anfänglich geplanten Funktion des Zeichnens eines Gebäudeplanes, können als zusätzliches Feature Stimmungen mittels einer Zeitschaltuhr zu bestimmten Zeiten eingeschaltet respektive ausgeschaltet werden. Diese zeitgebundenen Events können mit Hilfe der App erstellt, bearbeitet oder gelöscht werden.

Ausblick

Es sind durchaus verschiedene Erweiterungen der App denkbar. Sofern die Firma aizo und ihre Kunden zufrieden sind mit der App, kann man neue Funktionen entwickeln. Es gibt auf dem Webinterface, wo die digitalSTROM Einrichtung gesteuert werden kann, bereits einige zusätzliche Webapps, welche auf dem Windows App integriert werden könnten, so wie es mit der Zeitschaltuhr geschehen ist.

Inhaltsverzeichnis

0.	Inhaltsverzeichnis	7
1.	Einleitung	11
1.1.	Aufgabenstellung, Ziele.....	12
2.	Ergebnisse	12
2.1.	Allgemein.....	12
2.2.	Grundfunktionen Endprodukt	13
2.3.	Features Endprodukt.....	13
2.3.1.	Foto Page	13
2.3.2.	Zeitschaltuhr	14
3.	Schlussfolgerungen	14
3.1.	Beurteilung und Vergleich	14
3.2.	Bedeutung der erreichten Ergebnisse.....	14
3.3.	Ausblick	14
4.	Szenarios	15
5.	Use Cases	16
5.1.	UC 01 Login	16
5.2.	UC 02 Eine Stimmung einschalten	16
5.3.	UC 03 Favoriten hinzufügen	16
5.4.	UC 04 Zeitschaltuhr.....	17
5.5.	UC 05 Stromverbrauch messen.....	17
5.6.	UC 06 Steuerung durch Sprache	17
5.7.	UC 07 Ändern der Raum-, Stimmung- oder Apartmentbenennung.....	17
6.	Code.....	17
6.1.	Anzahl Codezeilen	17
6.2.	Unit Test	17
7.	digitalSTROM.Core Beschrieb.....	18
7.1.	Allgemein.....	18
7.1.1.	PCL.....	18
7.1.2.	Architektur.....	19
7.2.	digitalSTROM.Core.Network	21
7.2.1.	Login	21
7.2.2.	Überprüfung des Logins	24
7.2.3.	Apartment Funktionen	24
7.2.3.1.	Aufruf GetApartmentName()	24
7.2.3.2.	Aufruf SetApartmentName().....	25
7.2.3.3.	Aufruf GetStructure()	25
7.2.3.4.	Aufruf GetReachableGroups()	26
7.2.3.5.	Aufruf GetCircuits()	27
7.2.3.6.	Aufruf CallSceneBySceneNumber()	28

7.2.4.	Circuit Funktionen.....	29
7.2.4.1.	Aufruf GetConsumption()	29
7.2.4.2.	Aufruf GetEnergyMeterValue().....	29
7.2.5.	Zonen Funktionen (Räume).....	30
7.2.5.1.	Aufruf ZoneCallScene().....	30
7.2.5.2.	Aufruf GetReachableScenes()	30
7.2.5.3.	Aufruf GetSceneName().....	31
7.2.5.4.	Aufruf SetSceneName()	31
7.2.5.5.	Aufruf SetRoomName().....	32
7.2.6.	Geräteaufrufe	32
7.2.6.1.	Aufruf GetOutputValue()	32
7.2.7.	Zeitschaltuhr Funktionen	33
7.2.7.1.	Aufruf GetTimedEvents()	33
7.2.7.2.	Aufruf SetTimedEvents()	34
7.2.7.3.	Aufruf DisableTimedEvent()	35
7.2.7.4.	Aufruf EnableTimedEvent().....	35
7.2.7.5.	Aufruf DeleteTimedEvent().....	36
7.3.	digitalSTROM.Core DataCache.cs	37
7.3.1.	Public Apartment Funktionen	38
7.3.1.1.	Aufruf GetApartmentAsync()	38
7.3.1.2.	Aufruf SetApartmentNameAsync().....	38
7.3.1.3.	Aufruf CallActivity().....	38
7.3.2.	Public Zone Methoden.....	38
7.3.2.1.	Aufruf ZoneCallSceneAsync().....	38
7.3.2.2.	Aufruf SetRoomNameById()	39
7.3.2.3.	Aufruf SetSceneNameAsync()	39
7.3.3.	Public Zeitschaltuhr Methoden	39
7.3.3.1.	Aufruf GetTimedEventsAsync().....	39
7.3.3.2.	Aufruf SetTimedEventAsync().....	40
7.3.3.3.	Aufruf DisableTimedEventAsync()	40
7.3.3.4.	Aufruf EnableTimedEventAsync ()	40
7.3.3.5.	Aufruf DeleteTimedEventAsync ().....	40
7.3.4.	Nicht verwendete Public Methode	41
7.3.4.1.	Aufruf IsActiveScene()	41
7.4.	digitalSTROM.Core.DataModel.....	41
7.4.1.	digitalSTROM.Core.DataModel.StructureData.....	42
7.4.2.	digitalSTROM.Core.DataModel.CircuitData	43
7.4.3.	digitalSTROM.Core.DataModel.ZoneData	44
7.4.4.	digitalSTROM.Core.DataModel.SceneData	44
7.5.	digitalSTROM.Core.JsonConverters	45
7.6.	Timed Events (Zeitschaltuhr) [17]	45

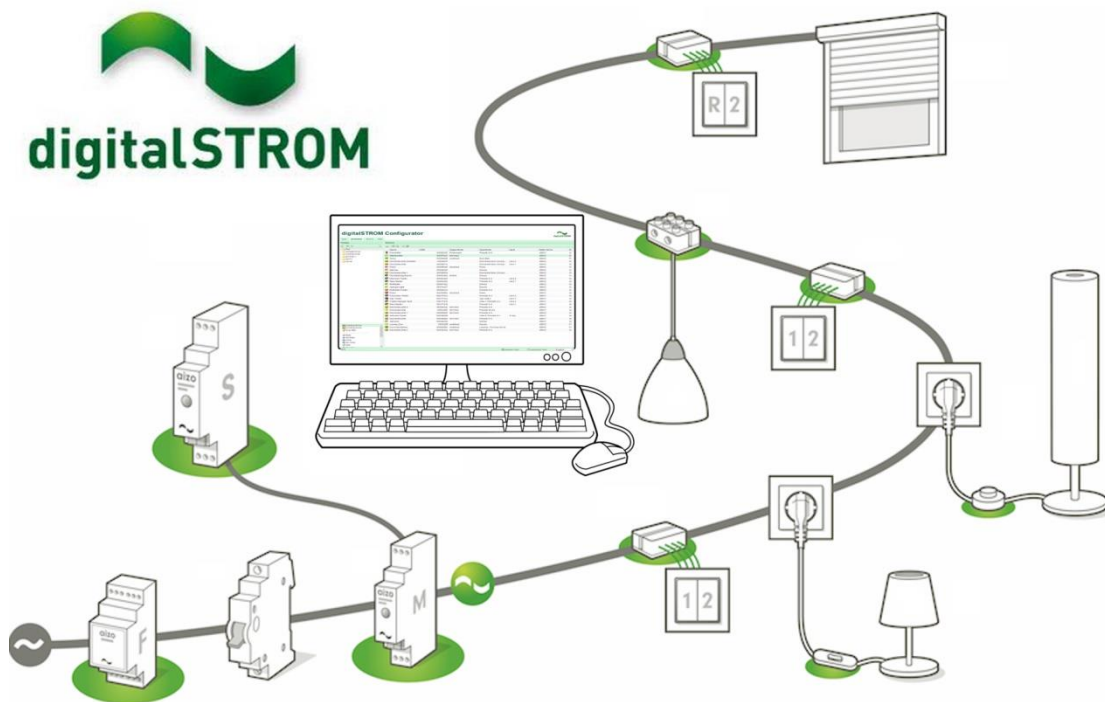
7.6.1.	Manuelles Parsen aufgrund Inkonsistenz in JSON Antwort.....	45
7.6.2.	Anzeigen der Timed Events	47
7.6.3.	Hinzufügen / Bearbeiten / Löschen der Timed Events	48
7.7.	namespace digitalSTROM.Storage.....	48
7.7.1.	Speichern der Favoriten	48
7.7.2.	Laden der Favoriten	49
7.7.3.	Speichern von Bildern	49
7.7.4.	Laden von Bildern.....	49
8.	digitalSTROM.View.....	49
8.1.	namespace digitalStrom.DataSource.....	49
8.2.	Lokalisierung	50
8.3.	Hubpage.....	50
8.4.	HubPageViewModel.....	51
8.5.	Photo Command	52
8.6.	Remove ICommand	52
8.7.	ZonePage.....	52
9.	User Interface	52
9.1.	Papierprototyp.....	53
9.2.	Navigationsdesign	53
9.2.1.	Favoriten.....	54
9.2.2.	Räume	55
9.2.3.	Handlungen	56
9.2.4.	Einstellungen	56
9.2.4.1.	Zeitschaltuhr.....	56
9.2.5.	Foto Seite	58
10.	Projektmethodik.....	59
10.1.	Sitzungsprotokolle.....	59
10.2.	Code Review	59
10.3.	Team Foundation Server	59
10.4.	Sprints	59
10.4.1.	Sprint 1	59
10.4.2.	Sprint 2	60
10.4.3.	Sprint 3	60
10.4.4.	Sprint 4	60
10.4.5.	Sprint 5	61
10.4.6.	Projektabschluss	61
11.	Anforderungsanalyse.....	63
11.1.	Einführung	63
11.1.1.	Zweck	63
11.2.	<i>Allgemeine Beschreibung</i>	63
11.2.1.	<i>Produktperspektive</i>	63

11.2.2.	Produkt <i>Funktionen</i>	63
11.2.2.1.	Hauptfunktionen der App	63
11.2.2.2.	Priorität 2 Funktionen der App	63
11.2.2.3.	Priorität 3 Funktionen der App	63
11.2.3.	Benutzer Charakteristiken	64
11.2.4.	Abhängigkeiten	64
12.	Persönliche Berichte	65
12.1.	Annrita Egli	65
12.2.	Richard Schulthess	65
13.	Glossar	67
14.	Literaturverzeichnis	69
14.1.	Bücher	69
14.2.	Internet	69
14.3.	Dokumente	69
15.	Tabellenverzeichnis	71
16.	Abbildungsverzeichnis	72
17.	Risikomanagement	73
18.	Diagramme	74
19.	TFS History	76

Technischer Bericht der Arbeit

0. Einleitung

Von der Firma aizo AG werden digitalSTROM Einrichtungen [2] vertrieben. Diese gelten als komfortabel und einfach zu bedienen. Beschliesst ein Hausbesitzer sein Haus mit digitalSTROM einzurichten, so muss er dies zuerst von einem Elektriker installieren lassen, welcher alle elektrischen Verbraucher im Haus entsprechend ausrüstet und zusammenschliesst. Danach können über die webbasierte Benutzeroberfläche am digitalSTROM-Server Einstellungen vorgenommen werden. Wie man sich mit diesem Server verbinden und damit Einstellungen vornehmen kann, ist im digitalSTROM Installationshandbuch [13] [14] beschrieben. In jedem Zimmer kann man zum Beispiel Stimmungen programmieren und bei jeder Lampe einstellen welche Helligkeit sie bei welcher Stimmung aufweisen soll. Es gibt auch Stimmungen welche das ganze Haus betreffen, wie zum Beispiel der Aufruf „Haus verlassen“, welcher alle elektrische Verbraucher ausschaltet. Die Bewohner können diese Stimmungen über einen Schalter im Haus oder über das Webinterface aufrufen. Die folgende Abbildung zeigt wie die digitalSTROM Geräte verbunden sind und das Webinterface auf dem Laptop.



Quelle: digitalSTROM Dokumentation, aizo AG

Abbildung 1 digitalSTROM bisher

Für die Bewohner eines digitalSTROM Hauses ist es jedoch praktischer, wenn sie Aufrufe von Stimmungen über eine App steuern können. Ein Vorteil davon ist, dass man auch von unterwegs überprüfen kann, ob alle Geräte im Haus ausgeschaltet sind oder man schon vor der Heimkehr die Lichter bereits wieder angehen lassen kann.

Für Android und für iOS Geräte existierte bereits eine App für digitalSTROM. Man kann damit die Stimmungen im Haus ein- und ausschalten und den Stromverbrauch ablesen. Was bisher fehlte, war eine App für Windows Plattformen. Der Firma aizo AG liegt viel daran, dass es solche Apps gibt und stellt darum ihre Gerätedokumentationen öffentlich zur Verfügung. aizo organisiert regelmässig Developer Days und liefert Informationen über die App Entwicklung für digitalSTROM.

0.1. Aufgabenstellung, Ziele

Da bisher keine digitalSTROM App für Windows Tablets existierte, beauftragte aizo AG die Firma Noser AG eine solche zu erstellen. Noser erteilte diesen Auftrag weiter an die HSR. Daraus entstand die Aufgabenstellung eine Windows 8 App für den Windows App Store zu entwickeln. Der Umfang der App wurde vorher nicht genau festgelegt, also musste zuerst abgesteckt werden, welche Funktionen die App ausweisen soll. In enger Zusammenarbeit mit Mitarbeitern von aizo AG wurde abgeklärt, welches die wichtigsten Funktionen sind, die eine digitalSTROM App ausweisen soll und welche zusätzlichen Features am nötigsten sind. Für die Entwicklung der App stellte aizo AG ein Development Kit zur Verfügung. Dieses besteht aus einem digitalSTROM Server und drei Lampen und simuliert damit eine digitalSTROM Einrichtung.



Abbildung 2 Demogerät von aizo AG

1. Ergebnisse

1.1. Allgemein

Über einen Login von aizo wurde die Kommunikation mit dem digitalSTROM Server über SSL möglich. Es galt nun die Schnittstelle zum implementieren, über welche die App mit dem Server kommunizieren kann. Welche Aufrufe der Server verwalten kann, ist in der Demogerät Entwicklungs-API von aizo AG [15] enthalten. Die folgende Abbildung zeigt die Development Kit Anlage in Gebrauch.



Abbildung 3 Development Kit von aizo

1.2. Grundfunktionen Endprodukt

Mit der in diesem Projekt erstellten digitalSTROM Windows 8.1 App können die Geräte einer digitalSTROM Einrichtung von der Firma aizo AG gesteuert werden. Mit dieser App ist ein Überblick über alle Räume des Hauses möglich und der momentane und gesamte Stromverbrauch der einzelnen Schaltkreise im Haus kann abgelesen werden. Man kann über die Hauptseite in die Räume navigieren, wo alle dort installierten digitalSTROM Geräte und die voreingestellten Stimmungen in diesem Raum aufgelistet werden. Von dort aus kann der Benutzer die voreingestellten Stimmungen aufrufen. Wird eine Stimmung öfters vom Benutzer benötigt, so kann er diese zu den Favoriten hinzufügen, damit sie nachher von der Hauptseite aus aufgerufen werden kann. Ebenfalls auf der Hauptseite ist ein direkter Aufruf der wichtigsten, das ganze Haus betreffenden Handlungen möglich.

1.3. Features Endprodukt

Nebst den wichtigsten Grundfunktionen die auf einer digitalSTROM App benötigt werden, wurden zwei Features umgesetzt.

1.3.1. Foto Page

Die Windows 8 App besitzt eine Foto Page. Mit der Kamera am Tablet können Fotos aufgenommen werden, welche danach den Räumen und favorisierten Stimmungen im Haus zugewiesen werden können. Es ist ebenfalls möglich, Fotos aus der eigenen Bildbibliothek den Räumen oder Stimmungen zuzuweisen. So kann der User an den Bildern sehen was eine eingeschaltete Stimmung bewirkt. Die Hauptseite der App listet alle Bilder der Räume auf, der User sieht also die gesamte Wohnung oder das ganze Haus vor sich und kann so schnell durch die Räume navigieren.

1.3.2. Zeitschaltuhr

Ein weiteres Feature der App ist die Zeitschaltuhr. Mit der Windows App können neue zeitbasierte Events erstellt oder gelöscht werden. Ebenfalls ist es möglich, die Events auf aktiv oder passiv zu stellen oder ihnen zum Beispiel andere Wochentage zuzuweisen. Für diese Implementierung galt es, mit dem digitalSTROM Webservice klarzukommen, der nicht immer zuverlässige Antworten lieferte.

2. Schlussfolgerungen

2.1. Beurteilung und Vergleich

Mit der digitalSTROM Windows 8.1 App wurde eine App erstellt, die die wichtigsten Funktionen und dazu einige Features für einen digitalSTROM Besitzer beinhaltet. Alle in der Anforderungsspezifikation aufgelisteten Hauptfunktionen und eine Zusatzfunktion konnten umgesetzt und in die App integriert werden. Ansätze für die Weiterentwicklung der App sind ebenfalls vorhanden. Damit kann die App als ein gereifter Prototyp angesehen werden, der den digitalSTROM Apps für Android und iOS ebenbürtig ist.

2.2. Bedeutung der erreichten Ergebnisse

Da bisher keine digitalSTROM App für Windows existierte, werden die digitalSTROM und Windows Tablet Besitzer über ihr Erscheinen froh sein. Sie haben jetzt die Möglichkeit ihre elektrischen Verbraucher über das Tablet zu steuern. Nebst dem Aufrufen der verschiedenen Stimmungen können sie den Räumen auch Bilder zuordnen oder Handlungen durchführen, die das gesamte Haus betreffen. Die App bietet auch das Bearbeiten der zeitgebundenen Events an. Damit können Aktivitäten zu einer bestimmten Zeit durchgeführt werden. Auf der App ist es möglich diese zu erstellen, zu löschen, inaktiv oder aktiv zu stellen oder zu bearbeiten.

2.3. Ausblick

Die Firma aizo bietet ihren Kunden sehr viele Features an. So gibt es für das Webinterface immer wieder neue Apps. Die digitalSTROM Windows 8.1 App wurde so erstellt, dass sie beliebig erweitert werden kann. Sobald der Webservice neue Funktionen anbietet, können diese auch auf der App integriert werden. Die App ist gewissermassen eine Vorlage für weitere Features von aizo. Bisher nicht umgesetzt wurde zum Beispiel die Sprachsteuerung, von der es gemäss aizo bereits Spezifikationen gibt. Damit kann die App gesprochene Befehle ausführen.

Denkbar ist auch eine Erweiterung die dem Benutzer erlaubt, auf der App Einstellungen vorzunehmen, die er bis jetzt nur auf dem Webinterface einstellen kann. Man könnte dem Benutzer ermöglichen, dass er Stimmungen bearbeiten kann, also den Geräten einen anderen Wert für die Stimmung zuzuweisen.

Projektdokumentation

3. Szenarios

Szenario 1: Haus verlassen

Beim Verlassen des Hauses mit digitalSTROM können über die App alle Lichter ausgeschaltet werden. Sofern man dies in der Eile vergisst, ist es auch von unterwegs möglich diese Handlung durchzuführen.

Szenario 2: Nach Hause kommen und Zeitschaltuhr

Wenn der Bewohner des Hauses mit digitalSTROM nach Hause kommt, kann er dies auf der App bekanntgeben, womit im Gang und in der Küche des Hauses die Lichter eingeschaltet werden und zusätzlich die Kaffeemaschine eingeschaltet wird. Es ist auch möglich diese Handlung so einzustellen, dass sie ohne weiteres Zutun vom User an bestimmten Tagen zu einer bestimmten Zeit ausgeführt wird.

Szenario 3: Lesen

Setzt sich der Bewohner des Hauses mit digitalSTROM hin zum Lesen, so gibt er dies auf der App bekannt, womit sich die Leselampen einschalten und die restlichen Lichter im Haus ausgeschaltet werden.

Szenario 4: Klingeln

Hat ein Bewohner des Hauses mit digitalSTROM Hörschwierigkeiten, so kann eingestellt werden, dass die Lampen im Haus blinken sobald die Klingel betätigt wird.

Szenario 5: Radio & Storen vom Bett aus bedienen

Nimmt der Bewohner des Hauses mit digitalSTROM das Windows Tablet mit auf den Nachttisch, so kann er der App Bescheid geben, dass er aufgewacht ist. Damit schaltet das Radio an, es öffnen sich die Storen und die Kaffeemaschine schaltet ein.

4. Use Cases

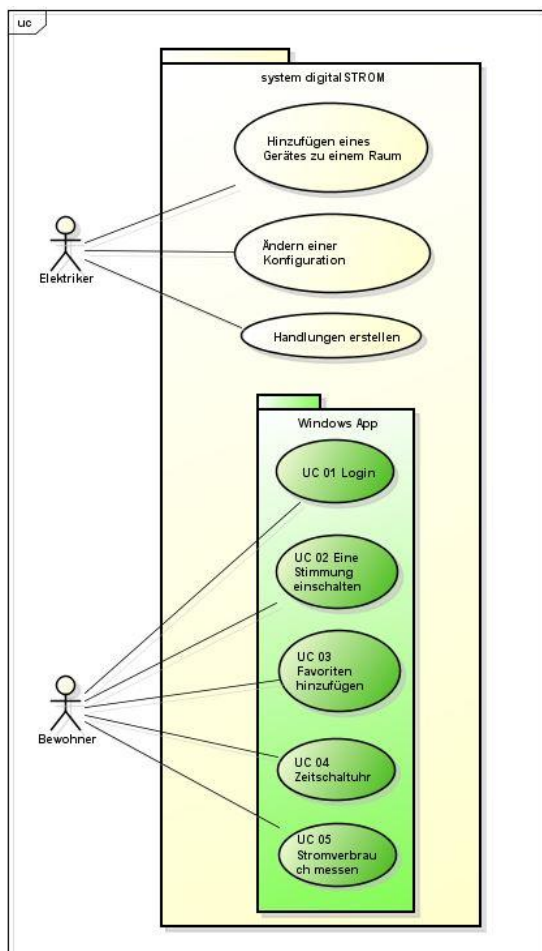


Abbildung 4 Use Case Diagramm

4.1. UC 01 Login

Nach einer erfolgreichen Konfiguration am Computer via Web Browser kann die Windows 8.1 App in Betrieb genommen werden. Als erstes muss der Benutzer die Server-IP und Port Adresse in die Login Maske eintragen. Danach muss er seinen Username und Passwort eingeben. Startet die App das erste Mal, wird er nach einem Verbinden mit der Cloud gefragt. Dieser Dialog kann abgebrochen oder mit „weiter“ bestätigt werden. Für die Cloudanbindung braucht er eine Remote-Verbindung und ein Login von der Firma aizo.

4.2. UC 02 Eine Stimmung einschalten

Über die Windows App kann der User verschiedene Stimmungen aufrufen. Dazu klickt der Benutzer entweder im Hauptfenster bei den Favoriten auf das entsprechende Symbol, oder er öffnet auf der App den entsprechenden Raum und wählt dort die gewünschte Stimmung aus.

4.3. UC 03 Favoriten hinzufügen

Auf dem Startscreen der App kann der Benutzer seine Favoriten hinzufügen. Gibt es eine Handlung oder eine Stimmung in einem Raum welche der Benutzer öfters einstellt, so kann er dieses Icon zu den Favoriten hinzufügen.

4.4. UC 04 Zeitschaltuhr

Mit der Windows App können dem Server Zeitschaltungen mitgeteilt und konfiguriert werden.

4.5. UC 05 Stromverbrauch messen

Auf der App kann der Bewohner des Hauses mit digitalSTROM nachsehen, wie hoch sein Stromverbrauch an den verschiedenen Tagen ist, und es wird ihm aufgezeigt, wie hoch der Stromverbrauch der einzelnen Geräte ist. So kann er Geräte mit hohem Stromverbrauch seltener gebrauchen oder durch bessere Produkte ersetzen.

4.6. UC 06 Steuerung durch Sprache

Mit der Anbindung an die Cloud wird der App eine Sprachsteuerung zur Verfügung gestellt. Ohne diese ist die App mit Touch Elementen steuerbar. Durch einen Knopf im Hauptfenster schaltet sich die Sprachsteuerung ein. Mit einem Signal wird der Benutzer informiert, dass er einen Befehl sprechen kann. Danach erscheint ein Feedback auf dem Display, das dem Benutzer anzeigt, was genau verstanden und was ausgeführt wurde.

4.7. UC 07 Ändern der Raum-, Stimmung- oder Apartmentbenennung

Ändert ein Benutzer seine Wohnungseinrichtung, kann er den Räumen oder Stimmungen eine neue Bezeichnung geben. Auch den Namen seiner gesamten digitalSTROM Einrichtung kann er jederzeit anpassen.

5. Code

5.1. Anzahl Codezeilen

Tabelle 1 Codemetrik Berechnung

Hierarchie	Wartbarkeitsindex	Zyklomatische Komplexität	Vererbungstiefe	Klassenkopplung	Codezeilen
digitalSTROM	86	774	7	200	1344
digitalSTROM.Core	91	385	2	138	679
digitalSTROM.UnitTest	78	43	1	49	159
digitalSTROM.UnitTest.View	79	5	1	8	13
				Total	2246

Die Anzahl Codezeilen ist durch häufigen Gebrauch von Linq [8] und vielen Verlagerungen in Framework Funktionalitäten niedrig gehalten.

5.2. Unit Test

Es wurden zwei Arten von Unit Tests geschrieben um den Core zu testen. Es gibt Unit Tests für den ausgeloggten und für den eingeloggten Zustand. Ist eine Verbindung zum Server möglich, so werden

zuerst die Setter-Methoden aufgerufen um ein Objekt zu setzen, welches dann mit der Getter-Methode wieder geholt und mit dem Erwartungswert verglichen wird.

6. digitalSTROM.Core Beschrieb

6.1. Allgemein

6.1.1. PCL

Damit die Core Library unseres Projektes auch für eine Windows Phone 8 App gebraucht werden könnte, müsste das Core Projekt mit einer Portable Class Library (PCL) [9] geschrieben werden.

Der Server von aizo besitzt ein self signed Certificate, welches beim Login (Kapitel 6.2.1) ignoriert werden muss um eine https Verbindung zu erhalten. Auf einem Windows Phone ist aber ein solcher ignore Befehl nicht möglich. Somit gibt es in einer PCL Library auch keine solchen Klassen welche das Ignorieren dieses Zertifikates erlauben.

Eine weitere Möglichkeit dieses Ignorieren des Zertifikats zu umgehen wäre noch mit einer Vorinstallation des Zertifikats gewesen. Doch wie aizo AG mitteilte, hat jedes digitalSTROM Gerät ein eigenes Zertifikat, damit ist eine Vorinstallation nutzlos. [4] [5]

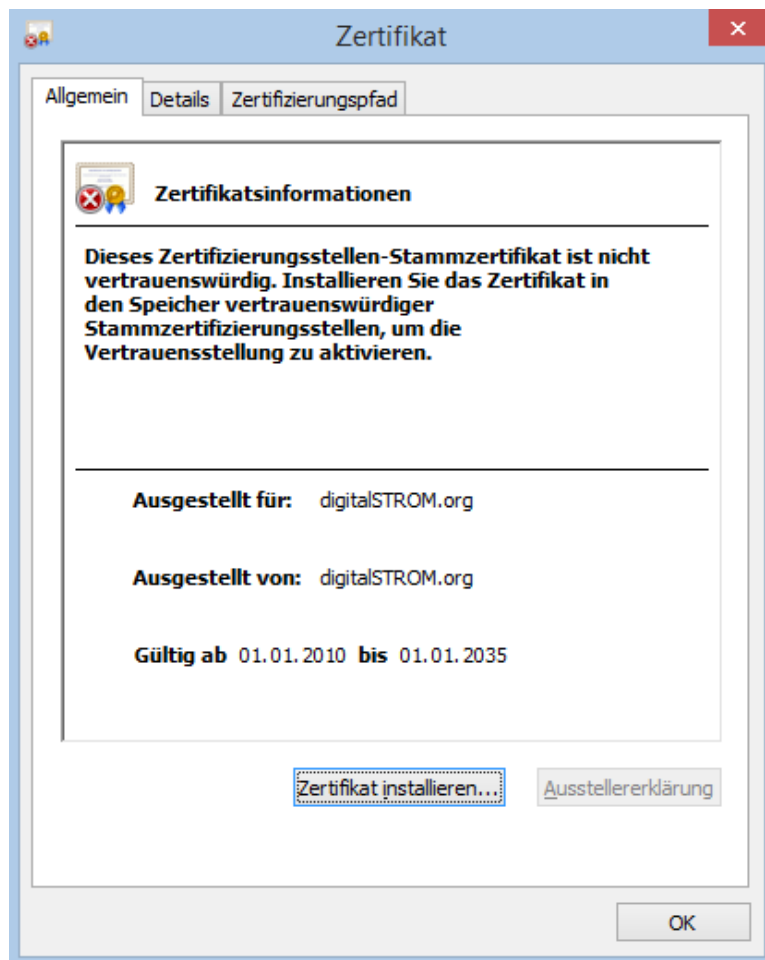


Abbildung 5 Zertifikat von digitalSTROM

6.1.2. Architektur

Die Netzwerkaufrufmethoden befinden sich alle in der Klasse Communication.cs, im Namespace digitalSTROM.Core.Network. Deserialisiert und gecacht werden die Antworten aus den Netzwerkaufrufen in der Klasse DataCache, im Namespace digitalSTROM.Core. Diese greift für die Deserialisierung auf die Klassen im Namespace digitalSTROM.Core.DataModel zu.

Das File DemoData.cs wird gebraucht um alle JSON Antworten zu speichern, welche die Communication.cs Klasse, im Fall dass keine Verbindung vorhanden ist, zurückgibt.

Die gecachten Daten aus DataCache.cs werden über die DataSource an die ViewModels weitergegeben.

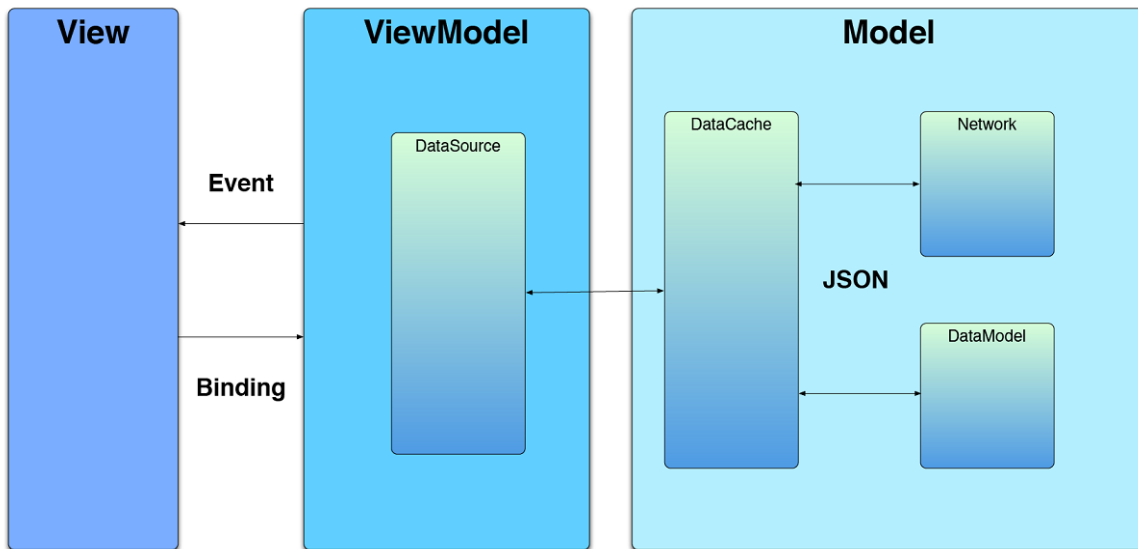


Abbildung 6 Architektur MVVM

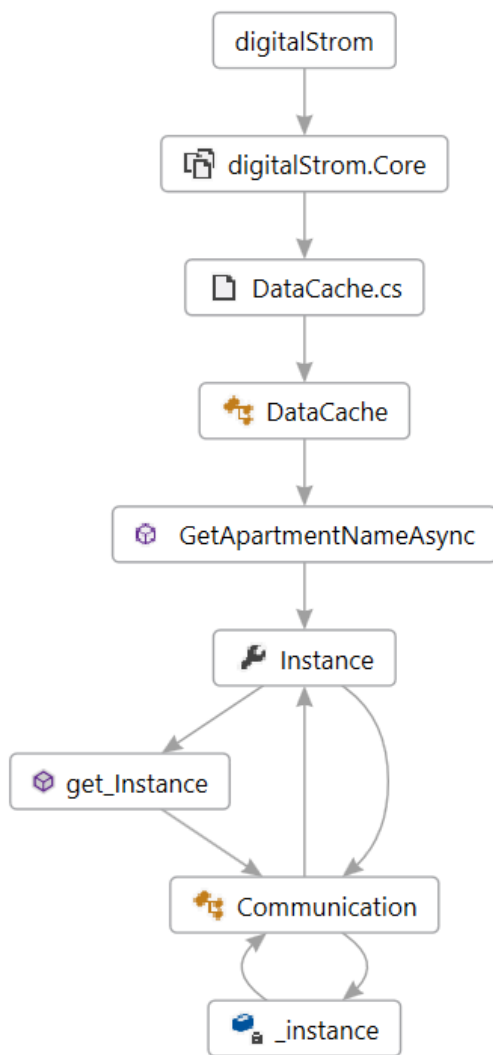


Abbildung 7 Aufruf am Beispiel der Methode GetApartmentNameAsync

6.2. digitalSTROM.Core.Network



Abbildung 8 Klassen im namespace digitalSTROM.Core.Network

6.2.1. Login

Tabelle 2 Login Methode

Methode	Login()
Parameter	Username (string) Password (string) Servername (string aus Server IP und Port)
Antworttyp	JSON

Ablauf Login:**Schritt 1 Filter setzen um nicht vertrauenswürdige Zertifikat zu ignorieren:**

```
1. HttpBaseProtocolFilter filter = new HttpBaseProtocolFilter {
2. ServerCredential = new Windows.Security.Credentials.PasswordCredential {
3. Password = password, UserName = userName
4. }
5. };
6. filter.IgnoreableServerCertificateErrors.Add(ChainValidationResult.Untrusted);
```

Schritt 2 Logout:

Applikation Token kann nur im ausgeloggten Zustand empfangen werden

URL: <https://169.254.80.81/json/system/logout>

Antworttyp: JSON

Antwort:

```
{
    "ok":true
}
```

Schritt 3 Gibt es bereits einen Application Token?

Überprüfen, ob bereits ein Application Token in Application Data vorhanden ist. Wenn vorhanden, springe zu Schritt 9, sonst weiter bei Schritt 4.

Schritt 4 Erbitten des Application Tokens:

URL: <https://169.254.80.81/json/system/requestetApplicationToken?applicationName=digitalStrom>

Antworttyp: JSON

Beispielantwort:

```
{
    "ok":true,
    "result":{
        "applicationToken":"bd4156e17486885da155088b26bfedaf632416b91"
    }
}
```

Schritt 5 Parsen des Application Tokens:

JsonObject.Parse(applicationTokenJson)

Antwort: string applicationToken

Schritt 6 Erbitten eines temporären Session Tokens:

Url: <https://169.254.80.81/json/system/login?user=username&password=password>

Antworttyp: JSON

Beispielantwort:

```
{
    "ok":true,
    "result":{
        "token":"35e11982877fe072c03cd0964377a22559fbbdd1b7eb2df685fff"
    }
}
```

```
    }  
}
```

Schritt 7 Parsen von User Auth Token aus dem Content:

JsonObject.Parse(temporarySessionTokenJson)

Antworttyp: string temporarySessionToken

Schritt 8 Aktivieren des Application Tokens:

URL: [https://169.254.80.81/json/system/enableToken?applicationToken=applicationToken
&token=temporarySessionToken](https://169.254.80.81/json/system/enableToken?applicationToken=applicationToken&token=temporarySessionToken)

Antworttyp: JSON

Antwort:

```
{  
    "ok":true  
}
```

Schritt 9 Erbitten des echten Session Tokens:

URL: <https://169.254.80.81/json/system/loginApplication?loginToken=applicationToken>

Antworttyp: JSON

Beispielantwort:

```
{  
    "ok":true,  
    "result":{  
        "token":"cb944ad861925e207f4fc8682c5740bc92adebd97ca7949dfce2"  
    }  
}
```

Schritt 10 Parsen des echten Tokens

JsonObject.Parse(tokenJson)

Antwort: string token

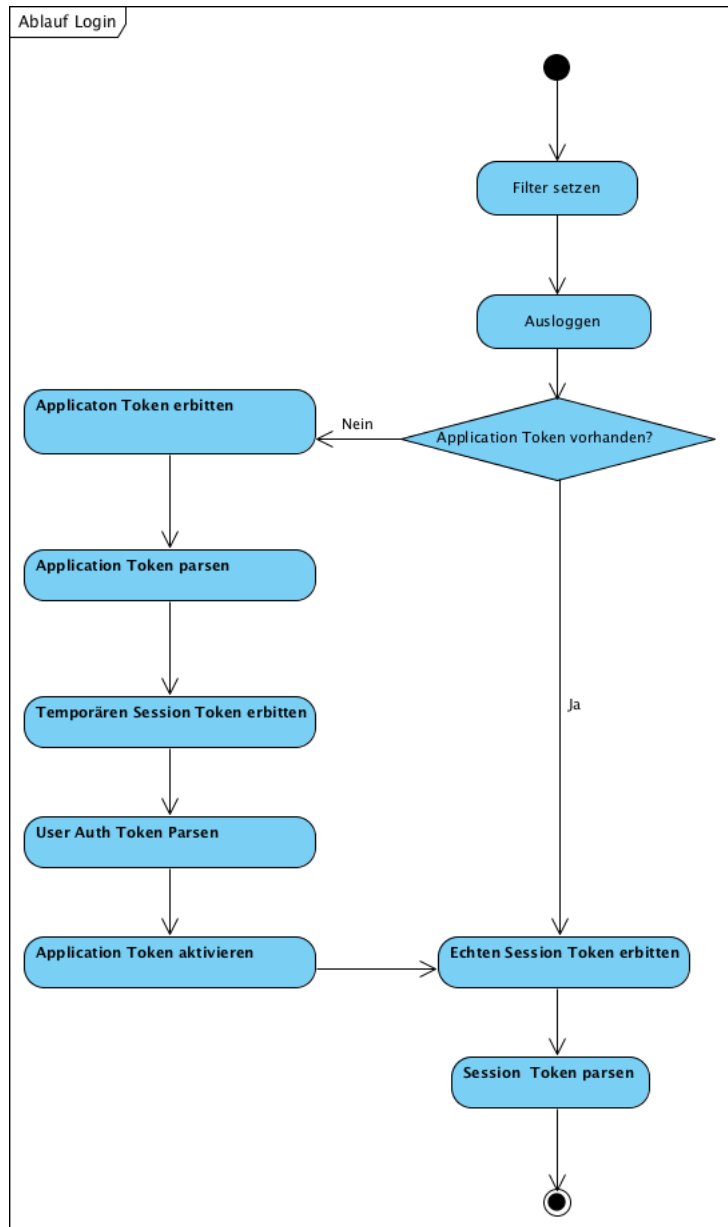


Abbildung 9 Ablauf Login

6.2.2. Überprüfung des Logins

Bei jedem Netzwerkaufruf muss geprüft werden, ob die Verbindung zum Server noch immer vorhanden ist. Dafür wird im Konstruktor der Communication.cs Klasse die Methode `SubscribeToNetworkStatusChange()` aufgerufen.

6.2.3. Apartment Funktionen

6.2.3.1. Aufruf `GetApartmentName()`

Tabelle 3 Webservice `GetApartmentName`

Methodenname	<code>GetApartmentName()</code>
Beschreibung	Gibt die Bezeichnung des Apartments zurück.
Parameter	-

Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/apartment/getName
Antworttyp	JSON (NameResponse)

Webrequest Beispiel:

<https://169.254.80.81:8080/json/apartment/getName>

JSON Antwort:

```
{
    "ok":true,"result":{
        "name":"Zuhause"
    }
}
```

6.2.3.2. Aufruf SetApartmentName()**Tabelle 4 Webservice SetApartmentName**

Methode	SetApartmentName()
Beschreibung	Setzt den Namen des Apartments auf der Einstellungsseite.
Parameter	newApartmentName (string)
Rückgabotyp	bool
Netzwerkaufruf	https://{serverName}/json/apartment/setName?newName={newApartmentName}
Antworttyp	JSON (NameResponse)

Webrequest Beispiel:

<https://169.254.80.81:8080/json/apartment/setName?newName=Bahnhof>

JSON Antwort:

```
{
    "ok":true
}
```

6.2.3.3. Aufruf GetStructure()**Tabelle 5 Webservice GetStructure**

Methode	GetStructure()
Beschreibung	Gibt die Struktur des Apartments zurück mit allen Räumen (zones) und deren dort angeschlossenen Geräten (devices).
Parameter	-
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/apartment/getStructure
Antworttyp	JSON (StructureResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/apartment/getStructure>

Antwort:

```
{ "ok": true, "result":
  { "apartment":
    { "zones": [
      { "id": 0, "name": "", "isPresent": true, "devices": [
        {
          "id": "3504175fe0000000000000d1", "name": "Dinning lights",
          "functionID": 4120, "productRevision": 774, "productID": 200,
          "meterDSID": "3504175fe0000010000004ce", "busID": 66, "isPresent": true, "lastDiscovered": "2011-04-11 07:38:35",
          "firstSeen": "2011-03-29 13:58:14", "on": false, "locked": false
        },
        {
          "id": "3504175fe0000000000000d2", "name": "Lamp couch",
          "functionID": 4120, "productRevision": 774, "productID": 200,
          "meterDSID": "3504175fe0000010000004ce", "busID": 67, "isPresent": true, "lastDiscovered": "2011-04-11 07:38:36",
          "firstSeen": "2011-03-29 13:58:14",
          "on": false, "locked": false
        }
      ]
    },
    { "groups": [
      {
        "id": 0, "name": "broadcast", "isPresent": false,
        "devices": [
          "3504175fe0000000000000d1",
          "3504175fe00000000000000d2",
          "3504175fe00000000000000d3",
          "3504175fe00000000000000d4"
        ]
      }
    ]
  }
}
```

6.2.3.4. Aufruf GetReachableGroups()**Tabelle 6 Webservice GetReachableGroups**

Methode	GetReachableGroups ()
Beschreibung	Gibt alle vorhandenen Gruppen in den Räumen (zones) des Apartments zurück.
Parameter	-
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/apartment/getReachableGroups
Antworttyp	JSON (ZoneResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/apartment/callScene?getReachableGroups>

Antwort:

```

{
  "ok":true,"result":{
    "zones":[{
      "zoneID":3,"name":"dsm2","groups":[1,2,16]
    },
    {
      "zoneID":2,"name":"Flur","groups":[1]
    },
    {
      "zoneID":1466,"name":"dsm1","groups":[1,2,16]
    },
    {
      "zoneID":1,"name":"Wohnzimmer","groups":[1,5]
    },
    {
      "zoneID":4,"name":"Schlafzimmer","groups":[]
    },
    {
      "zoneID":5,"name":"Zimmer A","groups":[1]
    },
    {
      "zoneID":6,"name":"Kammer","groups":[]
    },
    {
      "zoneID":7,"name":"Berlin1","groups":[]
    }
  ]}
}

```

6.2.3.5. Aufruf GetCircuits()**Tabelle 7 Webservice GetCircuits**

Methode	GetCircuits ()
Beschreibung	Gibt alle Sicherungen (circuits) des Apartments zurück. An diesen Sicherungen kann der Stromverbrauch abgelesen werden.
Parameter	-
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/apartment/getCircuits
Antworttyp	JSON (CircuitResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/apartment/getCircuits>

Antwort:

```
{
  "ok":true,"result":{
    "circuits":[
      {"name":"dSM1","dsid":"3504175fe00000100000005ba",
        "hwVersion":285278560,"armSwVersion":17825792,
        "dspSwVersion":17039616,"apiVersion":519,"hwName":"","
        "isPresent":true,"isValid":true,
        "ignoreActionsFromNewDevices":false
      },
      {"name":"dSM2","dsid":"3504175fe0000010000000585",
        "hwVersion":285278560,"armSwVersion":17825792,
        "dspSwVersion":17039616,"apiVersion":519,"hwName":"","
        "isPresent":true,"isValid":true,
        "ignoreActionsFromNewDevices":false
      }
    ]
  }
}
```

6.2.3.6. Aufruf CallSceneBySceneNumber()**Tabelle 8 Webservice CallSceneBySceneNumber**

Methode	CallSceneBySceneNumber ()
Beschreibung	Ruft die Stimmung (scene) auf. Diese ist unabhängig von einem Raum, wird also im gesamten Haus aufgerufen. Raumunabhängige Stimmungen werden als Handlungen bezeichnet.
Parameter	sceneNumber (string)
Rückgabetyt	void
Netzwerkaufruf	https://{serverName}/json/apartment/callScene?sceneNumber={sceneNumber}
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/apartment/callScene?sceneNumber=5>

Antwort:

```
{
  "ok":true
}
```

6.2.4. Circuit Funktionen

6.2.4.1. Aufruf *GetConsumption()*

Tabelle 9 Webservice GetConsumption

Methode	GetConsumption ()
Beschreibung	Gibt den momentanen Verbrauch in Watt des gewählten Schaltkreises zurück.
Parameter	circuitId (string)
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/circuit/getConsumption?id={circuitId}
Antworttyp	JSON (ConsumptionResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/circuit/getConsumption?id=53504175fe0000010000005ba>

Antwort:

```
{
  "ok":true,"result":{
    "consumption":2
  }
}
```

6.2.4.2. Aufruf *GetEnergyMeterValue()*

Tabelle 10 Webservice GetEnergyMeterValue

Methode	GetEnergyMeterValue ()
Beschreibung	Gibt den Gesamtverbrauch in Watt/Sekunden des gewählten Schaltkreises zurück.
Parameter	circuitId (string)
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/circuit/getEnergyMeterValue?id={circuitId}
Antworttyp	JSON (MeterValueResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/circuit/getEnergyMeterValue?id=53504175fe0000010000005ba>

Antwort:

```
{
  "ok":true,"result":{
    "meterValue":235518984
  }
}
```

6.2.5. Zonen Funktionen (Räume)

6.2.5.1. Aufruf ZoneCallScene()

Tabelle 11 Webservice ZoneCallScene

Methode	ZoneCallScene()
Beschreibung	Ruft eine der Stimmungen (scene) im entsprechenden Raum (zone) auf. Kann auf der ZonePage im entsprechenden Raum aufgerufen werden.
Parameter	zoneId(string) sceneNumber(string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/zone/callScene?id={zoneNumber}&groupNumber=1 &sceneNumber={sceneNumber}&force=true
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/zone/callScene?id=1&groupNumber=1&sceneNumber=17&force=true>

Antwort:

```
{
    "ok":true
}
```

6.2.5.2. Aufruf GetReachableScenes()

Tabelle 12 Webservice GetReachableScenes

Methode	GetReachableScenes()
Beschreibung	Gibt die Stimmungen (scenes) eines Raumes (zone) zurück
Parameter	zoneId (string)
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/zone/getReachableScenes?id={zoneId}
Antworttyp	JSON (SceneResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/zone/getReachableScenes?id=3>

Antwort:

```
{
    "ok":true,"result":{
        "reachableScenes":[5,17,18,19]
    }
}
```

6.2.5.3. Aufruf GetSceneName()

Tabelle 13 Webservice GetSceneName

Methode	GetSceneName()
Beschreibung	Gibt den Namen der Stimmung der Gruppe Licht im entsprechenden Raum zurück
Parameter	zoneId (string) sceneNumber (int) groupId (int) Die App unterstützt die Gruppe mit der Id 1 (Licht)
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/zone/sceneGetName?id={zoneId}&sceneNumber={sceneNumber}&groupID={groupId}
Antworttyp	JSON (NameResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/zone/sceneGetName?id=2&sceneNumber=5&groupID=1>

Antwort:

```
{
  "ok":true,"result":{
    "name":"Lesen"
  }
}
```

6.2.5.4. Aufruf SetSceneName()

Tabelle 14 Webservice SetSceneName

Methode	SetSceneName()
Beschreibung	Gibt der ausgewählten Stimmung auf der Photopage einen neuen Namen.
Parameter	zoneId (string) sceneNumber (string) groupId(int) Die App unterstützt die Gruppe mit der Id 1 (Licht) newSceneName (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/zone/sceneSetName?id={zoneId}&groupID={groupId}&sceneNumber={sceneNumber}&newName={newSceneName}
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/zone/sceneSetName?id=2&groupID=1&sceneNumber=5&newName=Putzen>

Antwort:

```
{
    "ok":true
}
```

6.2.5.5. Aufruf SetRoomName()**Tabelle 15 Webservice SetRoomName**

Methode	SetRoomName()
Beschreibung	Gibt dem ausgewählten Raum auf der Photopage einen neuen Namen.
Parameter	zoneId (string) newRoomName (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/zone/setName?id={zoneId}&newName={newRoomName}
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/zone/setName?id=2&newName=Salon>

Antwort:

```
{
    "ok":true
}
```

6.2.6. Geräteaufrufe**6.2.6.1. Aufruf GetOutputValue()****Tabelle 16 Webservice GetOutputValue**

Methode	GetOutputValue()
Beschreibung	Prüft anhand des output value des Geräts mit der Id „deviceId“, ob das Gerät eingeschaltet ist. Ist der Wert null, so ist das Gerät ausgeschaltet.
Parameter	deviceId (string)
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/device/getOutputValue?dsid={deviceId}&offset=0
Antworttyp	JSON (OutputValueResponse)

Aufrufbeispiel:

<https://169.254.80.81:8080/json/device/getOutputValue?dsid=35000000035757ab&offset=0>

Antwort:

```
{
  "ok":true,"result":{
    "offset":0,"value":0
  }
}
```

6.2.7. Zeitschaltuhr Funktionen**6.2.7.1. Aufruf GetTimedEvents()****Tabelle 17 Webservice GetTimedEvents**

Methode	GetTimedEvents ()
Beschreibung	Gibt alle Events zurück, die in der Zeitschaltuhr aufgeführt sind.
Parameter	-
Rückgabotyp	string
Netzwerkaufruf	https://{serverName}/json/property/query?query=/scripts/system-addon-timed-events/entries/*(*)/*(*)/*(*)/*(*)
Antworttyp	JSON (EventResponse)

Aufrufbeispiel:

[https://169.254.80.81:8080/json/property/query?query=/scripts/system-addon-timed-events/entries/*\(*\)/*\(*\)/*\(*\)/*\(*\)](https://169.254.80.81:8080/json/property/query?query=/scripts/system-addon-timed-events/entries/*(*)/*(*)/*(*)/*(*))

Antwort:

```
{
  "ok":true,
  "result":{
    "entries":[{"
      "name":"Beleuchtung aus","id":2,
      "scope":"system-addon-timed-events","deleteCounter":0,
      "lastExecuted":"2013-12-06 08:07:45","2":[{"name":[]},
      {"id":[]}, {"enabled":true,"conditions":[
        {"enabled":[]}]}, {"actions":[{"type":"zone-scene",
        "zone":6,"group":1,"scene":0,"force":false,"delay":0,
        "0":[{"},{},{},{},{},{},{}]}, {"timeBase":"sunrise",
        "offset":600,"recurrenceBase":"weekly","time":[
        {"timeBase":[]}, {"offset":[]}, {"recurrenceBase":[]},
        {"0":"MO","1":"TU","2":"WE","3":"TH",
        "recurrence":[{"},{},{},{},{},{},{}]},
        {"scope":[]}, {"deleteCounter":[]}, {"lastExecuted":[]}]
      },
      {
        "name":"Blinken","id":"8","scope":"system-addon-timed-
        events","deleteCounter":0,
        "lastExecuted":"2013-12-05 16:00:01","8":[{"name":[]},
        {"id":[]}, {"enabled":true,"conditions":[
        {"enabled":[]}, {"holiday":"off","states":[{}]}]},
        {"actions":[{"type":"device-scene",
```

```

        "dsid":"3504175fe00000000000582e", "scene":14,
        "force":false, "delay":0, "category":"timer",
        "0":[{}], {"timeBase":"daily",
        "offset":57600, "recurrenceBase":"weekly", "time":[
        {"timeBase":[]}, {"offset":[]}, {"recurrenceBase":[]},
        {"0":"MO", "1":"TH", "2":"FR", "3":"SA", "4":"SU",
        "recurrence":[{}], {"scope":[]},
        {"deleteCounter":[]}, {"lastExecuted":[]}]
    }
}

```

6.2.7.2. Aufruf SetTimedEvents()

Table 18 Webservice SetTimedEvents

Methode	SetTimedEvents ()
Beschreibung	Erzeugt einen neuen Event oder bearbeitet einen bestehenden Event, der an die Zeitschaltuhr gebunden ist. Der Event hat einen Namen, einen Zeitpunkt wann er aufgeföhrt wird (offset), ein Array von Wochentagen an welchen er ausgeföhrt wird, eine Raum (zone) und Stimmung (scene) welche dem Event zugeordnet werden und kann aktiv oder passiv sein (enabled). Wird ein neuer Event erzeugt, wird ihm vom aizo Gerät automatisch eine noch freie Id zugeordnet. Ein schon vorhandener Event behält seine Id.
Parameter	name (string) offset (int) weekdays (DayOfWeek[]) Das Array wird mittels english.DateTimeFormat.DayNames[(int) weekdays[i]].Substring(0, 2).ToUpper(); in den vom Netzwerkaufruf gewünschten String gebracht. groupId(int) Die App unterstützt die Gruppe mit der Id 1 (Licht) zoneId (string) sceneId (string) actionType(string) Die App unterstützt zum Setzen den „zone-scene“ Aktionstyp. enabled (bool) wird als true gesetzt. delay(int) Jedem Event wird ein Delay von 0 Sekunden zugeordnet. Der Event verfügt also über keinen Delay und muss pünktlich ausgeföhrt werden. eventId (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/event/raise?name=system-addon-timed-events.config¶meter=actions=save;value={„name“:“{name}“, „id“:null, „time“:„{offset“: {offset}, „timeBase“:“daily“, „recurrence“: {„timeArray“: {weekdays}, „recurrenceBase“:“weekly“, „actions“: [„type“:“{actionType}“, „zone“: zoneId, „group“: {groupId}, „scene“: {sceneId}, „force“: {enabled}, „delay“: {delay}]}}
Antworttyp	JSON

Aufrufbeispiel:

[https://169.254.80.81:8080/json/event/raise?_name=system-addon-timed-events.config¶meter=actions=save;_value={\"name\": \"Einzigartiger Name\", \"id\": null, \"time\": {\"offset\": 9000, \"timeBase\": \"daily\"}, \"recurrence\": {\"timeArray\": \[\"MO\", \"TU\", \"WE\"\], \"recurrenceBase\": \"weekly\"}, \"actions\": \[{\"type\": \"zone-scene\", \"zone\": 2077, \"group\": 1, \"scene\": 5, \"force\": false, \"delay\": 0}\]}\]](https://169.254.80.81:8080/json/event/raise?_name=system-addon-timed-events.config¶meter=actions=save;_value={\)

Antwort:

```
{
  "ok": true
}
```

6.2.7.3. Aufruf DisableTimedEvent()

Tabelle 19 Webservice DisableTimedEvent

Methode	DisableTimedEvent ()
Beschreibung	Setzt das Property IsEnabled vom EventItem auf inaktiv. Der Event wird so nicht durchgeführt.
Parameter	eventId (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/event/raise?name=system-addon-timed-events.config¶meter=actions=disable;id={eventId}
Antworttyp	JSON

Aufrufbeispiel:

https://169.254.80.81:8080/json/event/raise?_name=system-addon-timed-events.config¶meter=actions=disable;id=2

Antwort:

```
{
  "ok": true
}
```

6.2.7.4. Aufruf EnableTimedEvent()

Tabelle 20 Webservice EnableTimedEvent

Methode	EnableTimedEvent ()
Beschreibung	Setzt das Property IsEnabled vom EventItem auf aktiv. Der Event wird zum gewählten Zeitpunkt durchgeführt.
Parameter	eventId (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/event/raise?name=system-addon-timed-events.config¶meter=actions=enable;id={eventId}
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/event/raise?name=system-addon-timed-events.config¶meter=actions=enable;id=4>

Antwort:

```
{
    "ok":true
}
```

6.2.7.5. Aufruf DeleteTimedEvent()**Tabelle 21 Webservice DeleteTimedEvent**

Methode	DeleteTimedEvent ()
Beschreibung	Löscht den ausgewählten Event.
Parameter	eventId (string)
Rückgabotyp	void
Netzwerkaufruf	https://{serverName}/json/event/raise?name=system-addon-timed-events.config¶meter=actions=delete;id={eventId}
Antworttyp	JSON

Aufrufbeispiel:

<https://169.254.80.81:8080/json/event/raise?name=system-addon-timed-events.config¶meter=actions=delete;id=2>

Antwort:

```
{
    "ok":true
}
```

6.3. digitalSTROM.Core DataCache.cs

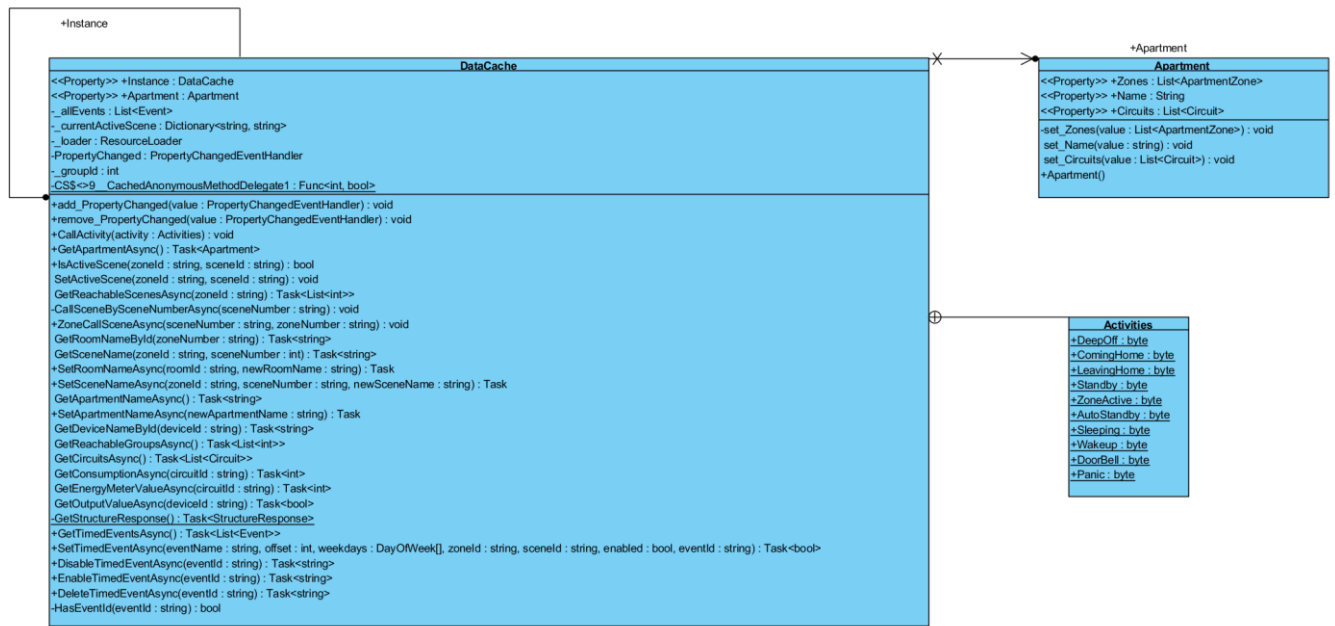


Abbildung 10 Klasse im namespace digitalSTROM.Core

Einige Id's, wie zum Beispiel die Id der Geräte, haben den Datentyp string, da diesen eine hexadezimale Id zugeordnet ist.

Damit die internal Methoden dieser Klasse ebenfalls mittels Unit Test getestet werden konnten, wurde im File AssemblyInfo.cs des digitalSTROM.Core Projekts die folgende Zeile eingefügt:

```
[assembly: InternalsVisibleTo("digitalSTROM.UnitTest")]
```

Auf der folgenden Abbildung ist dargestellt, wie die CommonDataSource die Daten aus dem DataCache erhält.

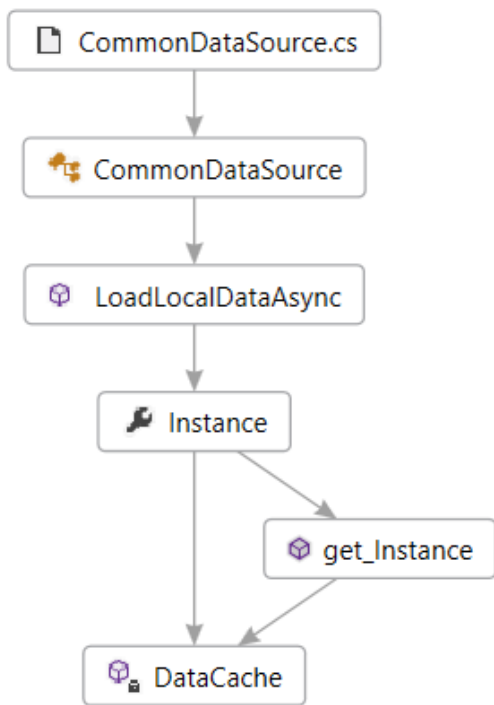


Abbildung 11 DataCache - CommonDataSource

6.3.1. Public Apartment Funktionen

6.3.1.1. Aufruf *GetApartmentAsync()*

Tabelle 22 DataCache GetApartmentAsync

Methode	GetApartmentAsync ()
Beschreibung	Gibt die deserialisierte Struktur des Apartments zurück mit allen Räumen (zones) und deren dort angeschlossenen Geräten (devices) und den Stimmungen (scenes).
Parameter	-
Rückgabotyp	Apartment

6.3.1.2. Aufruf *SetApartmentNameAsync()*

Tabelle 23 DataCache SetApartmentNameAsync

Methode	SetApartmentNameAsync ()
Beschreibung	Gibt dem Apartment einen neuen Namen
Parameter	newApartmentName (string)
Rückgabotyp	void

6.3.1.3. Aufruf *CallActivity()*

Tabelle 24 DataCache CallActivity

Methode	CallActivity ()
Beschreibung	Ruft die activity (Handlung) auf, welche im Parameter übergeben wurde und nimmt deren Id aus dem Enum Activities.
Parameter	Activities
Rückgabotyp	void

6.3.2. Public Zone Methoden

6.3.2.1. Aufruf *ZoneCallSceneAsync()*

Tabelle 25 DataCache ZoneCallSceneAsync

Methode	ZoneCallSceneAsync ()
Beschreibung	Ruft eine Stimmung (scene) im entsprechenden Raum (zone) auf. Die Stimmung muss über den Raum aufgerufen werden, da die sceneNumber der Räume keine unique Id ist, sondern dieselbe Nummer meist in mehreren Räumen vorkommt.
Parameter	sceneNumber (string) zoneld (string)
Rückgabotyp	void

6.3.2.2. Aufruf SetRoomNameById()

Tabelle 26 DataCache SetRoomNameById

Methode	SetRoomNameById ()
Beschreibung	Gibt dem entsprechenden Raum (zone) einen neuen Namen.
Parameter	roomId (string) newRoomName (string)
Rückgabotyp	Task

6.3.2.3. Aufruf SetSceneNameAsync()

Tabelle 27 DataCache SetSceneNameAsync

Methode	SetSceneNameAsync ()
Beschreibung	Gibt der entsprechenden Stimmung (scene) einen neuen Namen. Die Id des Raumes (zoneId) muss dafür ebenfalls übergeben werden, da die Nummer der Stimmung keine unique Id ist, sondern meistens in mehreren Räumen vorkommt.
Parameter	zoneId (string) sceneNumber (string) newSceneName (string)
Rückgabotyp	Task

6.3.3. Public Zeitschaltuhr Methoden**6.3.3.1. Aufruf GetTimedEventsAsync()**

Tabelle 28 DataCache GetTimedEventsAsync

Methode	GetTimedEventsAsync ()
Beschreibung	Gibt alle Zeitschaltuhr Events zurück
Parameter	-
Rückgabotyp	List<Event>

6.3.3.2. Aufruf SetTimedEventAsync()

Tabelle 29 DataCache SetTimedEventAsync

Methode	SetTimedEventAsync()
Beschreibung	Erzeugt einen neuen Zeitschaltuhr Event mit den entsprechenden Parametern oder verändert einen bestehenden Zeitschaltuhr Event.
Parameter	eventName (string) offset (int) weekdays (DayOfWeeks []) zoneld (string) sceneld (string) enabled (bool) eventId (string)
Rückgabetyyp	bool

6.3.3.3. Aufruf DisableTimedEventAsync()

Tabelle 30 DataCache DisableTimedEventAsync

Methode	DisableTimedEventAsync ()
Beschreibung	Setzt den entsprechenden Zeitschaltuhr Event auf inaktiv. Das heisst, der Event wird zwar aufgelistet, wird aber zur festgelegten Zeit nicht ausgeführt.
Parameter	eventId (string)
Rückgabetyyp	string

6.3.3.4. Aufruf EnableTimedEventAsync ()

Tabelle 31 DataCache EnableTimedEventAsync

Methode	EnableTimedEventAsync ()
Beschreibung	Setzt den entsprechenden Zeitschaltuhr Event auf aktiv. Das heisst, der Event wird zur festgelegten Zeit ausgeführt.
Parameter	eventId (string)
Rückgabetyyp	string

6.3.3.5. Aufruf DeleteTimedEventAsync ()

Tabelle 32 DataCache DeleteTimedEventAsync

Methode	DeleteTimedEventAsync ()
Beschreibung	Löscht den entsprechenden Zeitschaltuhr Event
Parameter	eventId (string)
Rückgabetyyp	string

6.3.4. Nicht verwendete Public Methode

6.3.4.1. Aufruf IsActiveScene()

Tabelle 33 DataCache IsActiveScene

Methode	IsActiveScene ()
Beschreibung	Prüft ob eine Stimmung (scene) in einem bestimmten Raum (zone) aktiv ist. Für die Methode existiert ein Unit Test, sie wird aber auf der jetzigen Version der App nicht verwendet. Sie könnte für den Use Case verwendet werden, falls man anzeigen will, ob eine Scene aktiv ist oder nicht.
Parameter	zoneld (string) sceneld (string)
Rückgabotyp	bool

6.4. digitalSTROM.Core.DataModel

Für den Aufruf der Daten welche sich im namespace digitalSTROM.Core.DataModel.StructureData befinden, stellte aizo AG Demodaten zur Verfügung, welche aufgerufen werden, sofern keine Verbindung mit dem digitalSTROM Server hergestellt werden kann. Zu den Datenabrufen der anderen DataModel namespaces gab es keine Demodaten. Diese mussten selbst erstellt werden und befinden sich in der Klasse DemoData.cs im namespace digitalSTROM.Core.Network. Die Zusammenstellung dieser Daten konnte mit Hilfe der Netzwerkantworten auf Aufrufe im Webinterface gefunden werden.

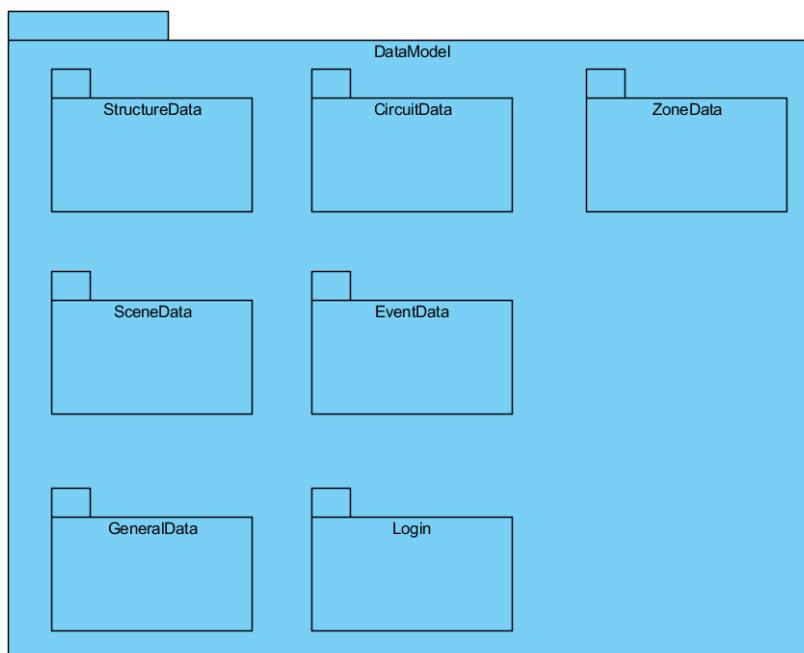


Abbildung 12 Überblick namespaces im DataModel

6.4.1. digitalSTROM.Core.DataModel.StructureData

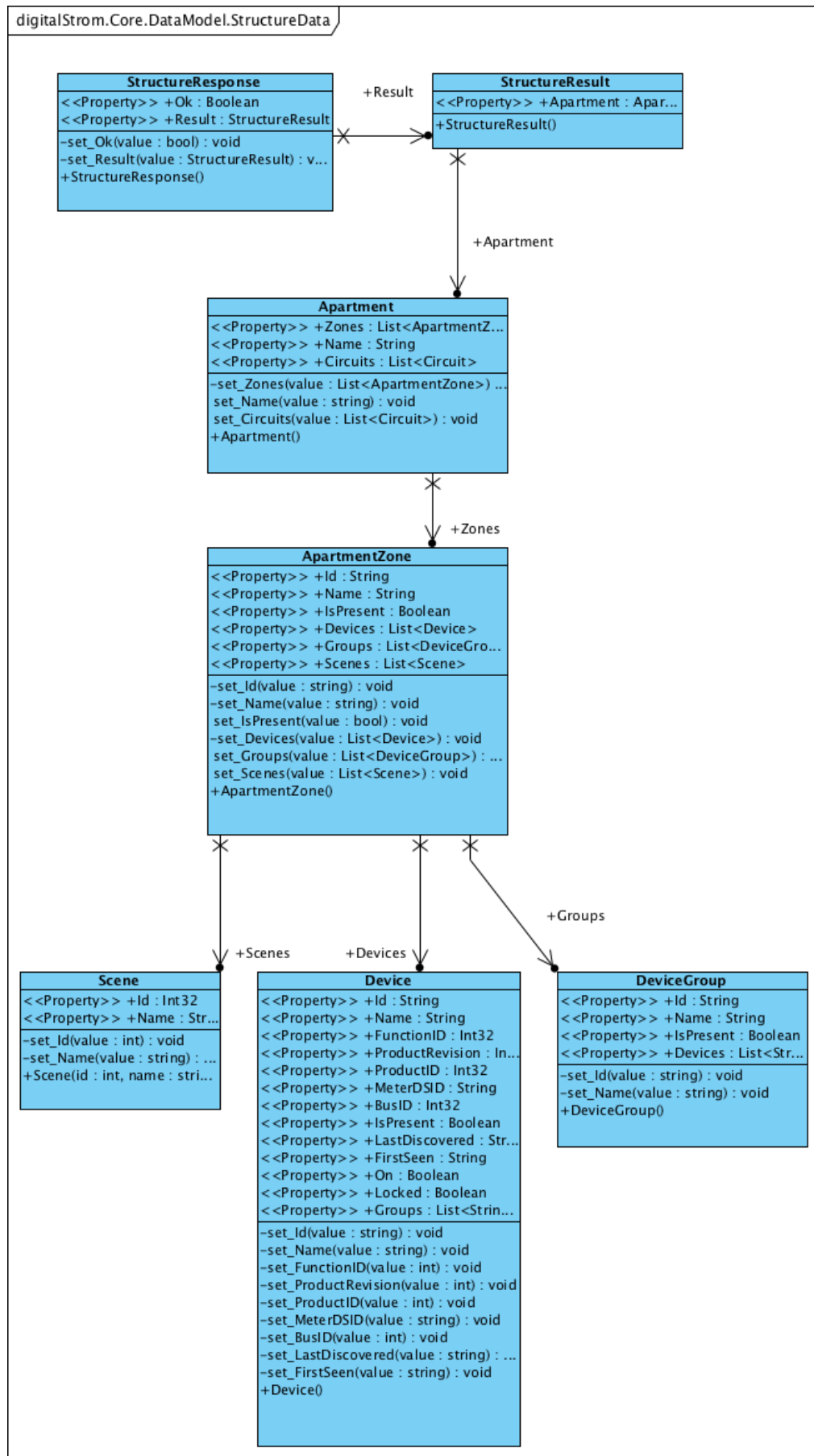


Abbildung 13 Datenmodell für die Struktur des Apartments

Der namespace digitalSTROM.Core.DataModel.StructureData beinhaltet die Datenstruktur des Apartments. Damit kann das gesamte Apartment aufgebaut werden.

6.4.2. digitalSTROM.Core.DataModel.CircuitData

In diesem namespace befinden sich die Daten die benötigt werden, um den Stromverbrauch abzulesen. In einem Apartment kann es ein bis mehrere Circuits geben, welche alle ihren derzeitigen Stromverbrauch und ihren gesamten Stromverbrauch messen. Durch die await Funktion für das Abrufen der Circuits wird ein anonymes Objekt generiert, was im Datenmodell als <<struct>> erscheint.

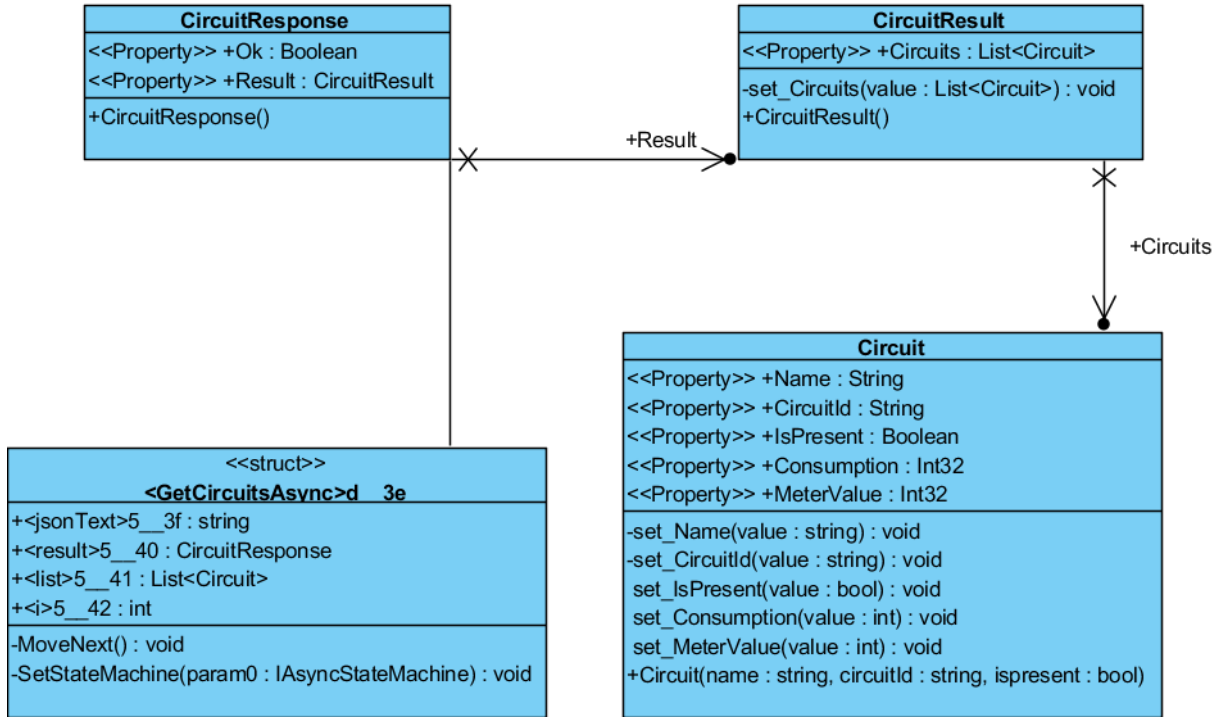


Abbildung 14 Datenmodell um die Circuits des Apartments herauszulesen

Der gesamte Stromverbrauch kann am „EnergyMeterValue“ herausgelesen werden. Dieses Datenmodell sieht folgendermassen aus:

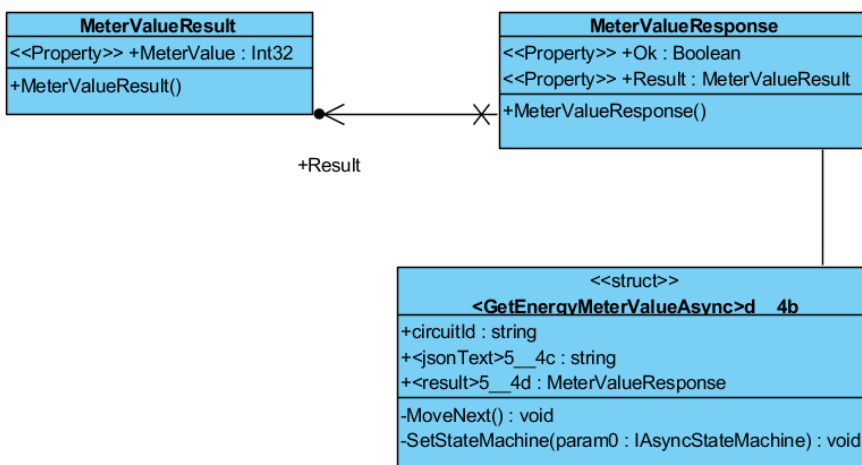


Abbildung 15 Datenmodell um den EnergyMeterValue herauszulesen

Dasselbe Bild zeigt sich beim momentanen Stromverbrauch was als „Consumption“ zurückgegeben wird:

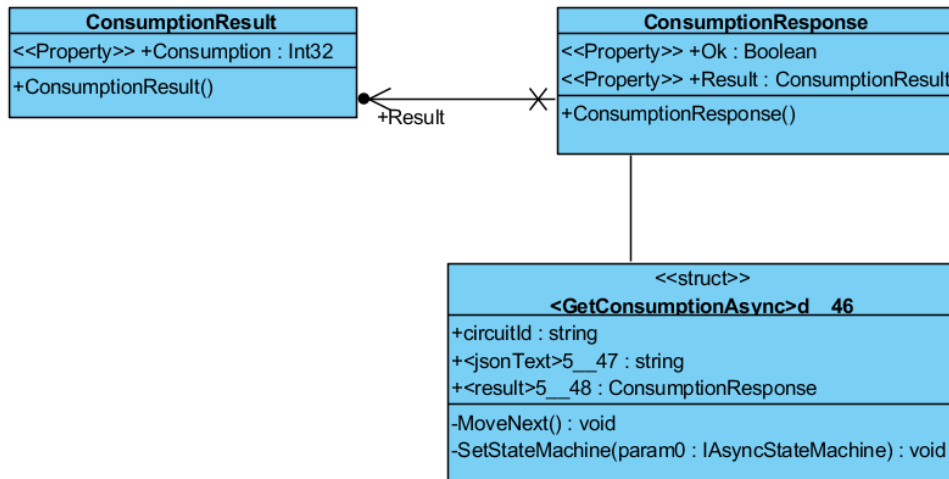


Abbildung 16 Datenmodell um Consumption herauszulesen

6.4.3. digitalSTROM.Core.DataModel.ZoneData

Auf die Zonen kann über den namespace ZoneData direkt zugegriffen werden. So müssen bei einzelnen Abfragen über die Zone nicht die gesamten StructureData heruntergeladen werden.

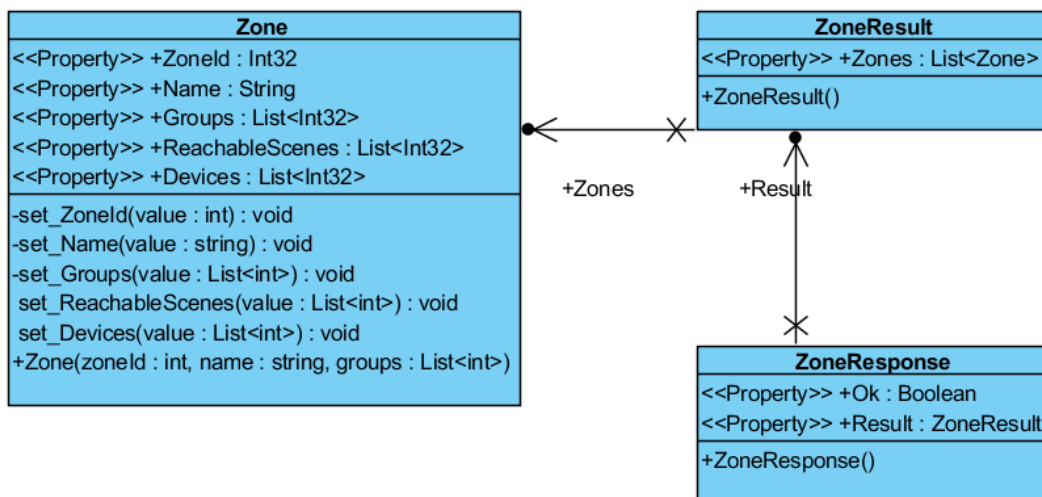
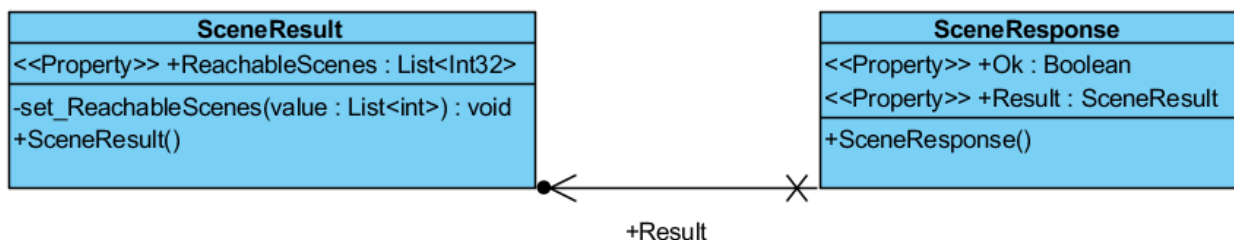


Abbildung 17 Datenmodell Zone

6.4.4. digitalSTROM.Core.DataModel.SceneData



6.5. digitalSTROM.Core.JsonConverters

Im diesem namespace befinden sich zwei JSON Converters. Wofür diese nötig sind wird im nächsten Kapitel 6.6 Timed Events beschrieben.

EventTimeConverter
+CanConvert(objectType : Type) : bool
+ReadJson(reader : JsonReader, objectType : Type, existingValue : Object, serializer : JsonSerializer) : Object
+ParseDay(dayValue : string) : DayOfWeek
+WriteJson(writer : JsonWriter, value : Object, serializer : JsonSerializer) : void
+EventTimeConverter()

EventResultConverter
-CS\$<9 CachedAnonymousMethodDelegate8 : Func<JToken,<>f AnonymousType0<JToken,JObject>><JToken,<>f AnonymousType0'2>
-CS\$<9 CachedAnonymousMethodDelegate9 : Func<<>f AnonymousType0<JToken,JObject>.bool><<>f AnonymousType0'2,Boolean>
-CS\$<9 CachedAnonymousMethodDelegatea : Func<<>f AnonymousType0<JToken,JObject>.bool><<>f AnonymousType0'2,Boolean>
-CS\$<9 CachedAnonymousMethodDelegateb : Func<<>f AnonymousType0<JToken,JObject>.IEnumerable<JProperty>><<>f AnonymousType0'2,IEnumerable'1>
-CS\$<9 CachedAnonymousMethodDelegatec : Func<<>f AnonymousType0<JToken,JObject>.JProperty,JProperty><<>f AnonymousType0'2,JProperty,JProperty>
-CS\$<9 CachedAnonymousMethodDelegated : Func<JObject, bool>
+CanConvert(objectType : Type) : bool
+ReadJson(reader : JsonReader, objectType : Type, existingValue : Object, serializer : JsonSerializer) : Object
+WriteJson(writer : JsonWriter, value : Object, serializer : JsonSerializer) : void
+EventResultConverter()

Abbildung 18 Klassen im namespace digitalSTROM.Core.JsonConverters

6.6. Timed Events (Zeitschaltuhr) [17]

6.6.1. Manuelles Parsen aufgrund Inkonsistenz in JSON Antwort

Die Antworten zu Event-Aufrufen sind kein normales JSON. Folgendes Beispiel zeigt einen Ausschnitt der Antwort aus dem Aufruf „GetTimedEvents“:

```
{
  "ok":true,
  "result":{
    "entries":[
      {
        "name":"Weihnachtsbeleuchtung aus",
        "id":2,
        "scope":"system-addon-timed-events",
        "lastExecuted":"2013-12-06 08:07:45",
        "2":[]
      },
      {
        "name":"Alles aus",
        "id":"0",
        "scope":"system-addon-timed-events",
        "lastExecuted":"2013-12-06 00:00:00",
        "0":[]
      },
      {
        "name":"Licht Flur",
        "id":4,
        "scope":"system-addon-timed-events ",
        "lastExecuted":"2013-12-06 00:10:00",
        "4":[]
      },
      {
        "name":"Test Sonnenaufgang",
        "id":"5",
```

```

        "scope": "system-addon-timed-events",
        "lastExecuted": "2013-12-04 09:01:01",
        "5": []
    }
}
}

```

Nach dem Key „lastExecuted“ folgt ein Key welcher seine Bezeichnung der Id des Eintrags anpasst. Auf diese Weise kann das JSON nicht von einem herkömmlichen JSON Converter herausgelesen werden. Hier wird die Klasse EventResultConverter.cs benötigt. Dieser liest die Einträge, welcher sich hinter dem Key mit der Zahl befindet, aus. Das folgende Beispiel zeigt einen solchen Eintrag hinter der Nummer:

```

"2": [
    {
        "name": [
        ],
        {
            "id": [
            ],
            {
                "enabled": true,
                "conditions": [
                    {}
                ]
            },
            {
                "actions": [
                    {}
                ]
            },
            {
                "timeBase": "daily",
                "offset": 600,
                "recurrenceBase": "weekly",
                "time": [
                ],
            },
            {
                "scope": [
                ],
            },
            {
                "deleteCounter": [
                ],
            },
            {
                "lastExecuted": [
                ]
            }
        ]
    ]
}

```

Beim Versuch aus dem EventResult die einzelnen Entries zu konvertieren, kommt folgender Code zum Zuge:

```

1  [DataMember]
2  [JsonConverter(typeof(EventResultConverter))]
3  public List<Event> Entries { get; private set; }

```

So wird der EventResultConverter aufgerufen, welcher die benötigten Key/Value Paare herausliest und der jeweiligen EventDescription zuweist.

Der nächste Spezialfall folgt bei der EventTime, dort sind die Wochentage aufgelistet, an welchen der Event durchgeführt werden soll, also braucht man am Schluss eine List mit DayOfWeek Einträgen. Hier ein Beispiel für einen solchen Time Eintrag:

```
"time": [
  {
    "timeBase": [
  ]
  },
  {
    "offset": [
  ]
  },
  {
    "recurrenceBase": [
  ]
  },
  {
    "0": "MO",
    "1": "TH",
    "2": "FR",
    "3": "SA",
    "4": "SU",
    "recurrence": [
  ]
  }
]
```

Auch hier sind die Keys wieder Nummern, ihre Anzahl hängt von den Anzahl Tagen ab, an welchen der Event durchgeführt werden soll.

Die EventTime Klasse sieht folgendermassen aus:

```
1 [DataContract, JsonSerializer(typeof(EventTimeConverter))]
2 public class EventTime
3 {
4     [DataMember]
5     public DayOfWeek[] Weekdays { get; set; }
6 }
```

Es wird also der EventTimeConverter aufgerufen, der die Wochentage ausliest und dem Array von DayOfWeek hinzufügt. Um herauszufinden, wie so ein Converter selbst geschrieben werden kann, wurde im Internet recherchiert, ein gutes Beispiel fand sich auf einer Forumsseite. [10]

6.6.2. Anzeigen der Timed Events

Die Anzeige der Timed Events geschieht ähnlich wie auf dem Webinterface in Form einer Tabelle. In der ersten Spalte erscheint der Name des Events, der beim Erstellen auch akzeptiert wird, wenn er nicht eindeutig ist.

In der zweiten Spalte erscheint die Zeit, wann der Event durchgeführt werden soll. Die Zeit wird als eine ganze Zahl (int offset) gespeichert. Dieser offset beinhaltet die Anzahl Sekunden nach Mitternacht. Die Klasse EventItem.cs besitzt die Methode GetTimeString(), wo der Integer offset in die Zeit umgerechnet wird und schliesslich im Format „HH:mm“ zurückgegeben wird.

```
4 var timeBase = new DateTime(DateTime.Now.Year, DateTime.Now.Month, DateTime.Now.Day);
5     var time = timeBase.AddSeconds(Offset);
6     return time.ToString("HH:mm");
```

Die dritte Spalte listet die Wochentage auf, an welchen der Event durchgeführt wird. Jeder Event muss auf mindestens einen Wochentag eingestellt sein.

Die vierte Spalte zeigt die Aktivitäten die bei diesem Event durchgeführt werden. Es gibt verschiedene Typen von Aktivitäten welche über das Webinterface einem Event zugeordnet werden können:

- Typ device-scene: ein einzelnes Gerät schaltet sich ein.
- Typ change-state: eine Handlung welche das gesamte Haus betrifft wird durchgeführt. Ist es eine Handlung die eine Id hat für globale Handlungen gemäss der Dokumentation von aizo, so wird der Name der Handlung angezeigt.
- Typ custom-event: eine benutzerdefinierte Handlung, welche ebenfalls auf dem Webinterface definiert werden kann, wird durchgeführt.
- Typ zone-scene: bei diesen Aktivitäten wird angezeigt welche Stimmung in welchem Raum ausgeführt wird.

In der letzten Spalte wird angezeigt, ob der Event zurzeit aktiv ist oder nicht. Dieser Wert kann dort geändert werden.

6.6.3. Hinzufügen / Bearbeiten / Löschen der Timed Events

Auf der Seite der Zeitschaltuhr gibt es einen Button, womit neue Timed Events erzeugt werden können. Einen Event kann man erst speichern, wenn alle Felder ausgefüllt sind, da alle diese Parameter erforderlich sind. Als durchzuführende Handlung kann man auf der App nur den Handlungstyp „device-scene“, also eine Stimmung die einem Raum zugeordnet ist, auswählen.

Wird ein Event auf der Tabelle ausgewählt, so kann man diesen löschen oder bearbeiten. Für die Bearbeitung öffnet sich dasselbe Fenster wie beim Hinzufügen, nur dass hier die Felder schon vorausgefüllt sind mit den existierenden Parametern. Beim Löschen wird der Event auf dem Server gelöscht und verschwindet von der Liste.

6.7. namespace digitalSTROM.Storage

Die Klasse Storage, die das Abspeichern und Laden von Bildern und Favoriten übernimmt, ist ein Singleton. Diese enthält eine private Member Variable mit einer Liste von den Dateinamen der Bilder.

6.7.1. Speichern der Favoriten

Die Favoriten werden im Verzeichnis LocalState abgespeichert mit dem Dateinamen „favorites.xml“. Alle Dateien in diesem Verzeichnis werden solange aufbewahrt, wie die App auf dem Tablet installiert ist.

Es wird eine Liste mit ArrayOfString im XML Format abgelegt. Hier ein Beispiel:

```
1 <ArrayOfArrayOfstring xmlns="http://schemas.microsoft.com/2003/10/Serialization/Arrays"
2   xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
3   <ArrayOfstring><string>339</string><string>30</string></ArrayOfstring>
4 </ArrayOfArrayOfstring>
```

Darin befindet sich pro Eintrag ein Array mit zwei strings, die die RoomId und SceneId repräsentieren. Die ganze Liste wird bei einem neu hinzugefügten Favoriten in die Datei geschrieben. Ist diese Datei schon vorhanden, so wird sie überschrieben.

6.7.2. Laden der Favoriten

Die Datei „favorites.xml“ wird aus dem LocalState Ordner geladen. Falls die Datei nicht gefunden werden kann, wird eine FileNotFoundException geworfen.

Die Liste wird verglichen mit den empfangenen Stimmungen vom DSS-Server. Stimmt die RoomId und die SceneId überein, können diese Stimmungen in einer ObservableCollection hinzugefügt werden.

6.7.3. Speichern von Bildern

Jedes Bild wird als WriteableBitmap abgespeichert. Der Dateiname wird nach dieser Konvention benannt. Zuerst steht die UniqueId der Gruppe. Wenn es sich um einen Favoriten handelt, wird die Nummer 1000 oder bei einem Raum die Nummer 2000 vorangestellt. Danach folgt der Buchstabe „r“ und die RoomId und wenn es sich um eine Stimmung handelt, dann der Buchstabe „s“ und SceneId.

Beispiel eines Dateinamens einer Stimmung: 1000r339s30

Beispiel eines Dateinamens eines Raumes: 2000r339.

Falls ein Bild schon existiert, wird ihm beim erneuten Speichern der Datei eine eindeutige Nummer zugewiesen.

```
1 storageFile = await ApplicationData.Current.LocalFolder.CreateFileAsync(name,
  CreationCollisionOption.GenerateUniqueName);
```

Dieser Dateiname, welcher zum Speichern benötigt wurde, wird in eine Liste eingetragen. Diese Liste wird ebenfalls als XML unter dem Namen „images.xml“ gespeichert.

6.7.4. Laden von Bildern

Für das Laden der Bilder müssen die Dateinamen aus der Liste mit strings geladen werden.

7. digitalSTROM.View

In diesem Paket befinden sich die einzelnen Seiten der digitalSTROM App. Für jede Seite existiert ein XAML-Dokument und eine Code-behind-Datei in C Sharp.

7.1. namespace digitalStrom.DataSource

In diesem Namespace befinden sich die einzelnen Elemente (Items), die auf der View erscheinen. Alle diese Items erben dabei von der Klasse DataCommon. Diese beinhaltet die basic Attribute wie eine UniqueID, einen Titel, und einen Pfad zu einem Bild. Alle Attribute lösen einen INotifyChanged Event aus, wenn eine Änderung stattgefunden hat. Die CommonDataSource enthält alle Referenzen der verschiedenen Items in mehreren ObservableCollection.

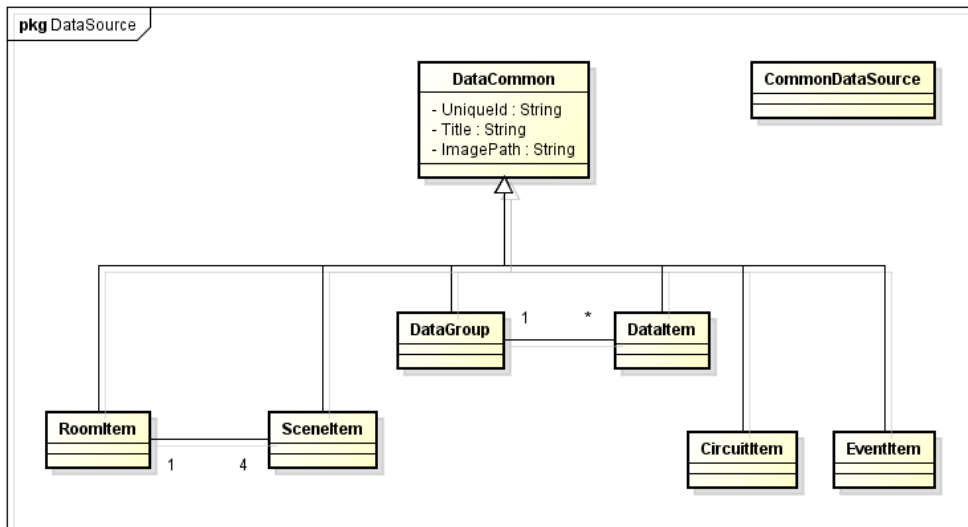


Abbildung 19 namespace DataSource

7.2. Lokalisierung

Die Lokalisierung läuft über eine sogenannte Resources.resw-Datei. Es gibt im Projekt zwei solche Dateien. Eine befindet sich im Ordner Strings\de-De für die Deutsche Übersetzung und die andere im Ordner Strings\en-Us in Englisch. Das Laden erfolgt über einen ResourceLoader.

```

1  DataItem leave = new DataItem();
2  leave.Title = _loader.GetString("Leave/Content");

```

Wie aus Zeile 2 hervorgeht, ladet die Instanz _loader den Titel aus der Resources.resw-Datei. In dieser Datei ist für den Namen Leave.Content als Wert „Verlassen“ gespeichert.

In einer XAML-Datei wird ein Tag mit der Bezeichnung x:Uid="Leave" eingesetzt.

7.3. Hubpage

Die Hubpage ist die oberste Ebene des Apps. Dort werden alle Hauptinhalte als Hub des Apps angezeigt.

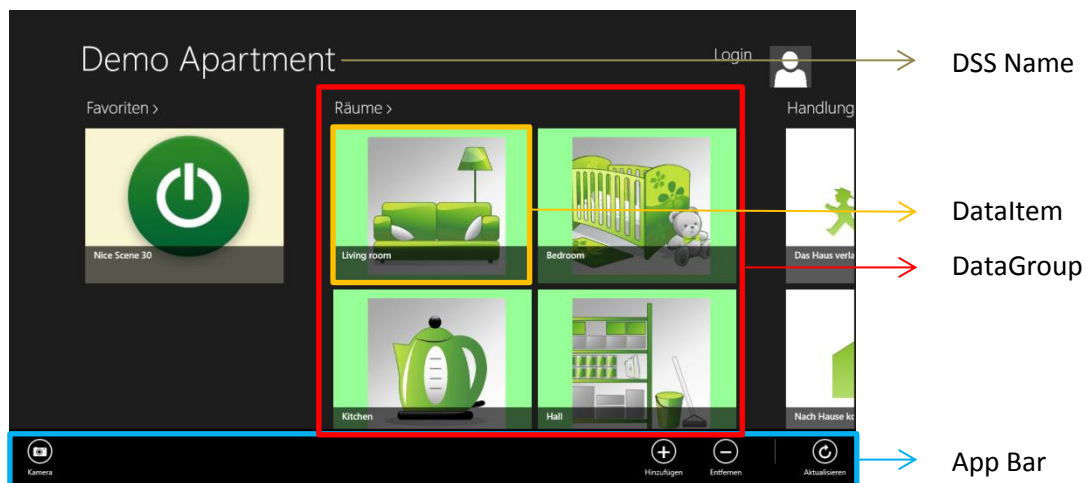


Abbildung 20 Hubpage

Die Hubs werden in verschiedene Abschnitte unterteilt. Diese sind als DataGroup Objekte im Paket digitalSTROM.DataSource gespeichert. Diese wiederum beinhalten eine ObservableCollection von Dataltem. Ein Dataltem wird mit einem Bild und einem Titel dargestellt.

Die Hubpage hat oben einen Namen, welcher vom digitalSTROM Server übertragen wird und unten besitzt sie eine App Bar um Aktionen auszuführen. Rechts oben ist ein Button integriert, der auf die Loginpage verweist.

Die Page besitzt ein HubPageDesignViewModel und ein HubPageViewModel. Das HubPageDesignViewModel wird für die CollectionViewSource gebraucht. Zur Design Zeit wird davon eine Instanz erzeugt und angezeigt.

```

1      <CollectionViewSource
2          x:Name="groupedItemsViewSource"
3          Source="{Binding Groups}"
4          IsSourceGrouped="true"
5          ItemsPath="TopItems"
6          d:DesignSource="{Binding AllGroups, Source={d:DesignInstance
designViewModels:HubPageDesignViewModel, IsDesignTimeCreatable=True}}"/>
    
```

Eine Instanz des HubPageViewModel wurde in die Code Behind als DataContext angehängt.

7.4. HubPageViewModel

Das HubPageViewModel ist ein Teil des MVVM-Patterns. Es bindet Daten als Inhalt oder Attribute der Steuerelemente. Ändern sich die Daten im ViewModel, so wird die Funktion der Oberklasse SetProperty<T>, die INotifyPropertyChanged implementiert, aufgerufen. Dabei wird ein Event ausgelöst und die Steuerelemente reagieren und updaten sich.

Commands werden ebenfalls an das HubpageViewModel gebunden. Diese sind im ViewModel RelayCommand, die eine Action und eine Bedingung oder nur eine Action enthalten. Diese Klasse RelayCommand ist aus dem MVVM light Framework und befindet sich im Paket digitalSTROM.Common.

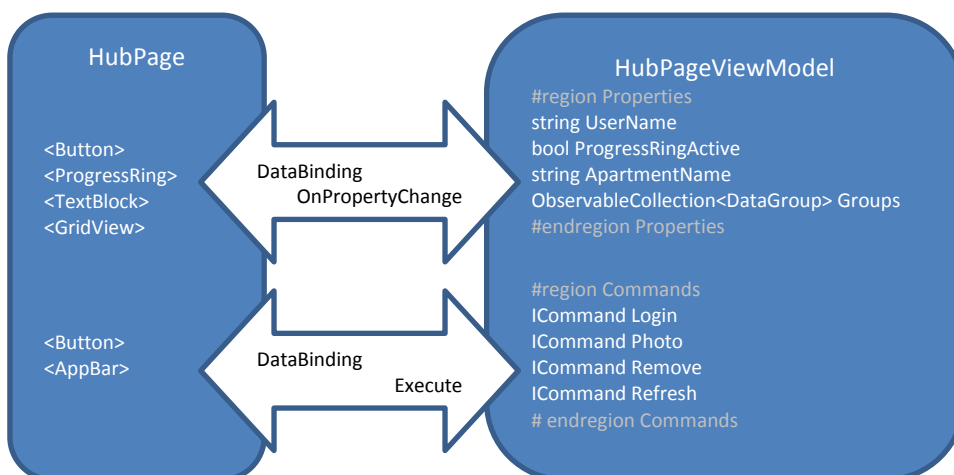


Abbildung 21 DataBinding auf der Hubpage

An dieser Stelle folgen zwei Beschreibungen zu wichtigen Commands:

7.5. Photo Command

Dieses Command ist an den AppBarButton auf der AppBar gebunden. Der AppBarButton ist aktiviert, sobald ein DataItem von der Gruppe GroupFavorites oder GroupRooms ausgewählt wurde. Nach dem Anklicken des Buttons wird die Action ClickPhoto ausgeführt. Das heisst es wird zuerst geprüft, ob wirklich ein DataItem selektiert wurde und reicht dieses weiter an die PhotoPage.

```
1 private async void ClickPhoto()
2     {
3         if (SelectedItem != null)
4         {
5             var rootFrame = Window.Current.Content as Frame;
6             if (rootFrame != null)
7             {
8                 rootFrame.Navigate(typeof(PhotoPage), SelectedItem);
9             }
10        }
11    }
```

7.6. Remove ICommand

Dieses Command reagiert auf den Remove AppBarButton. Sobald eine Stimmung auf der Hubpage ausgewählt wurde, wird der Button mit dem Command aktiv. Wird dieser ausgelöst, wird die UniqueId der Stimmung ausgelesen und an die CommonDataSource weitergereicht. Diese löscht die ausgewählte Stimmung aus der Liste mit den Favoriten. Diese Liste wird dann über die Storageinstanz als XML-Datei gespeichert.

7.7. ZonePage

Die ZonePage ist eine von vier Seiten, die eine DataGroup repräsentiert. In einer GridView sind alle Räume im Haus mit Foto und Namen visualisiert. Mit einem Klick kommt man in den entsprechenden Raum.

Auch ist es möglich, auf dieser Seite den Stromverbrauch abzulesen.

8. User Interface

In diesem Abschnitt wird auf die Design Entscheidungen eingegangen. Dabei wird begründet, warum gewisse Designentscheidungen gefällt worden sind.

Das User Interface ist mit dem Windows 8 typischen Metro Look ausgestattet. Nach dem Login erscheinen die verschiedenen Kacheln, die in Gruppen sortiert sind. Die Einstiegsseite HubPage ist eine Panoramadarstellung. Die Navigations Hierarchie ist so flach wie möglich gehalten.

8.1. Papierprototyp

Das aizo Webinterface wurde benutzt, um einen Papierprototyp zu entwerfen und um herauszufinden, welche Informationen dargestellt werden können.

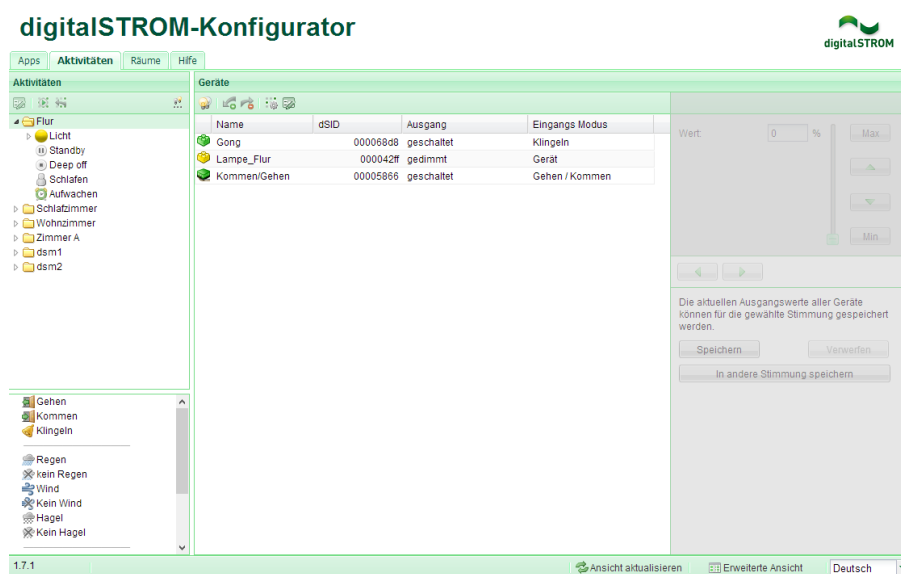


Abbildung 22 Webinterface von digitalSTROM

8.2. Navigationsdesign

Für die Navigation wurde das Hub Style Navigation Pattern des Windows-Store verwendet. Dieses Muster eignet sich bestens für Apps mit einer visuellen Navigation.

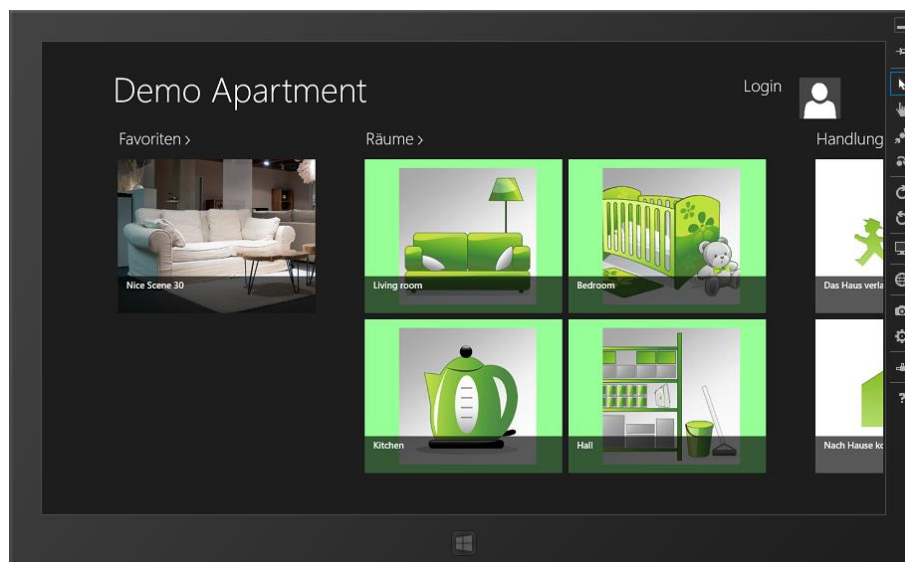


Abbildung 23 Hauptseite mit einem Favorit

Als Einstieg bekommt der Anwender eine Abbildung einer Demo Wohnung zu sehen. Möchte dieser sich mit seiner Wohnung verbinden, muss der Benutzer sich zuerst am System anmelden.

Auf der Login Seite wird er aufgefordert, die Login- und Verbindungsdaten in die Maske einzutragen.

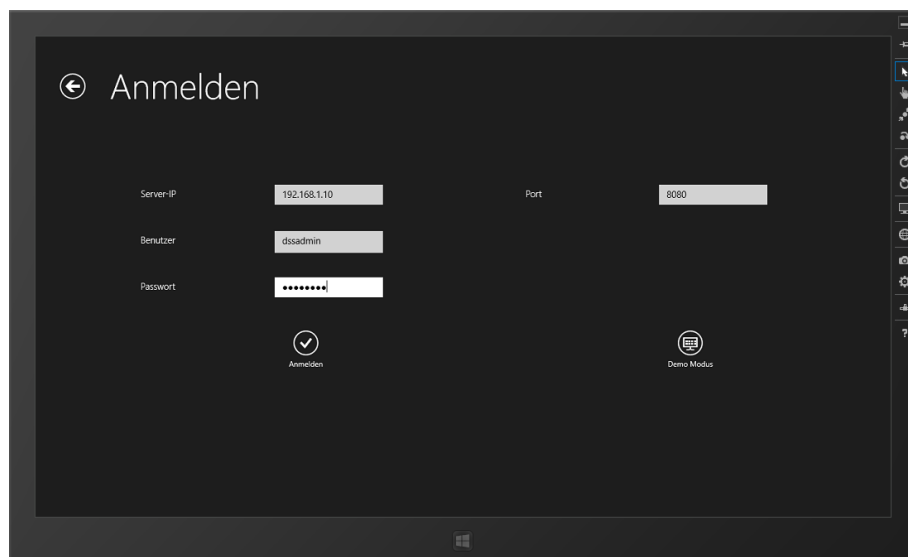


Abbildung 24 Anmelden

Nach dem erfolgreichen Login befindet man sich wieder auf der Hauptseite mit vier verschiedenen Kategorien.

8.2.1. Favoriten

Die erste Gruppierung sind die favorisierten Stimmungen. Diese werden beim Antippen des Bildes aktiviert. Möchte man diese verwalten, klickt man auf den Titel Favoriten.

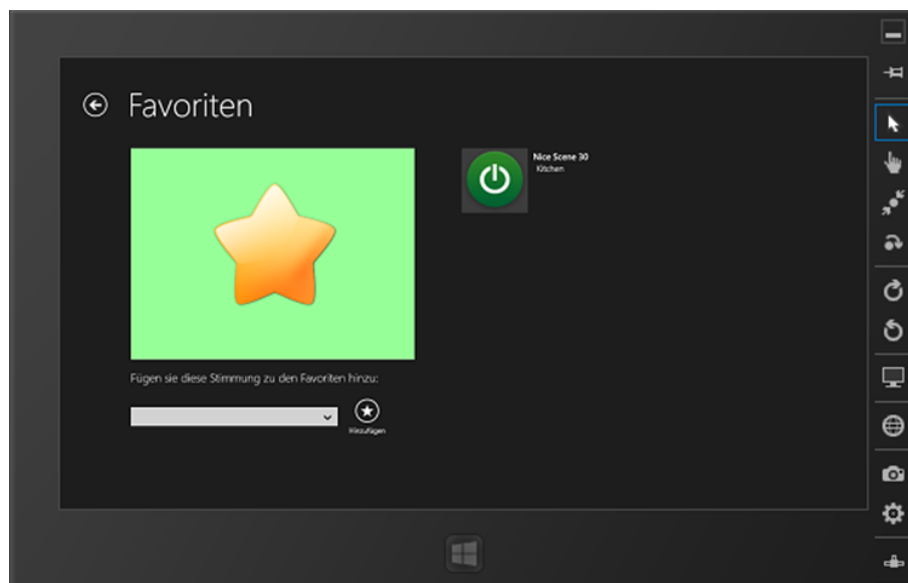


Abbildung 25 Gruppe Favoriten

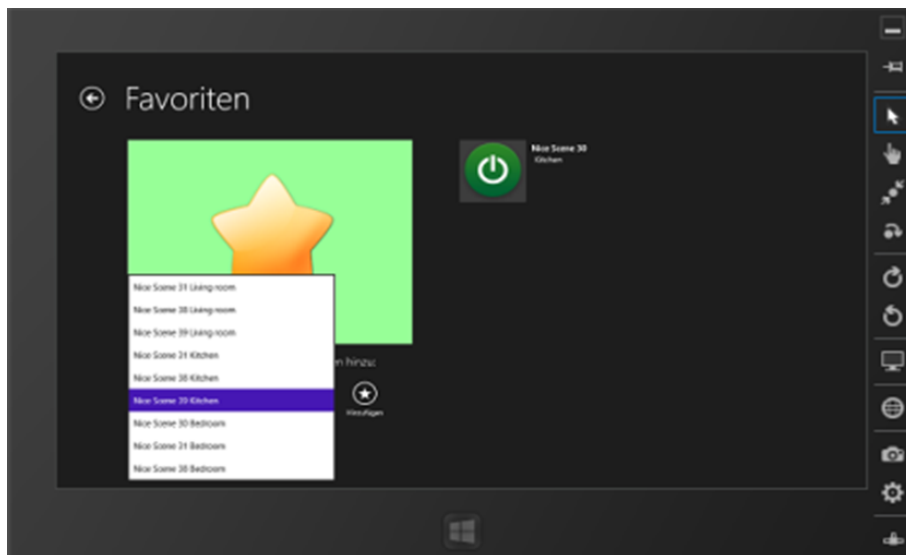


Abbildung 26 Gruppe Favoriten mit der Möglichkeit Favoriten hinzuzufügen

8.2.2. Räume

In den einzelnen Räumen befinden sich die steuerbaren Elemente einer digitalSTROM Wohnung. Ein Raum besteht aus einem Set von vier verschiedenen Stimmungen. Es kann nur eine Stimmung auf einmal aktiviert werden. Im Hauptmenü Räume kann zudem der Stromverbrauch einer Stromschaltung abgelesen werden.

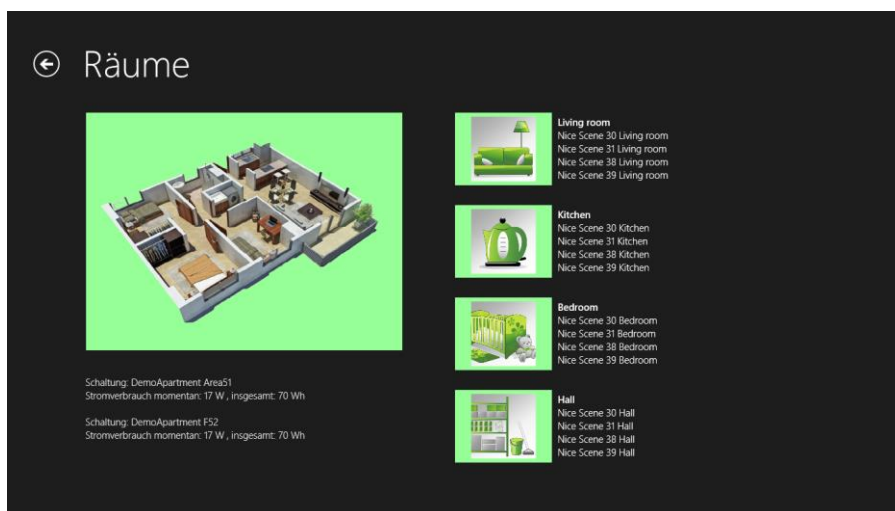


Abbildung 27 Seite mit der Gruppe von Räume



Abbildung 28 Raum Detailansicht

8.2.3. Handlungen

Auch die Gruppe Handlungen sind auf Knopfdruck bedienbar. So lässt sich das ganze Haus in einen bestimmten Zustand versetzen. Die einzelnen Handlungen sind schon vorgegeben und müssen nicht konfiguriert werden.

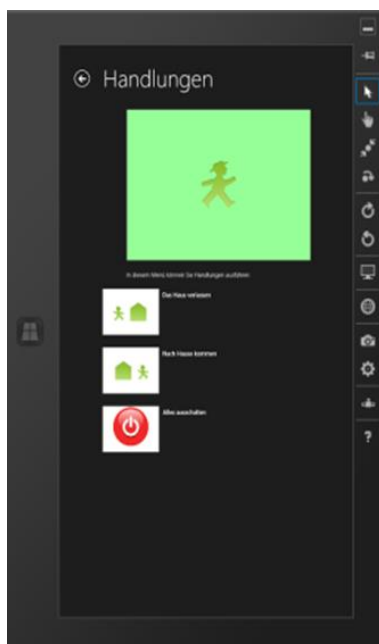


Abbildung 29 Gruppe Handlungen

8.2.4. Einstellungen

8.2.4.1. Zeitschaltuhr

Mit der Zeitschaltuhr können Aktivitäten programmiert werden. Eine Aktivität ist zum Beispiel, dass in der Küche die Stimmung „Frühstücken“ jeden Montag um 7.00 Uhr angeschaltet wird.

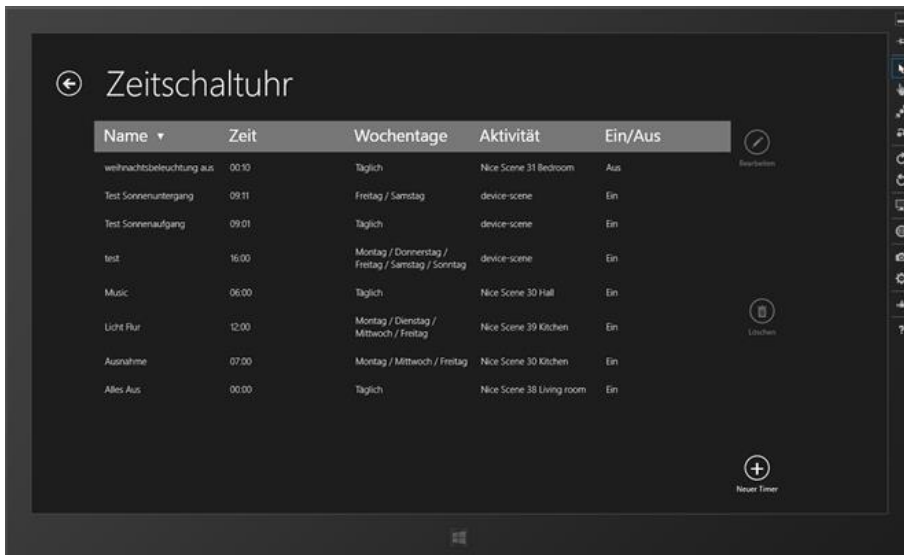


Abbildung 30 Zeitschaltuhr mit Events

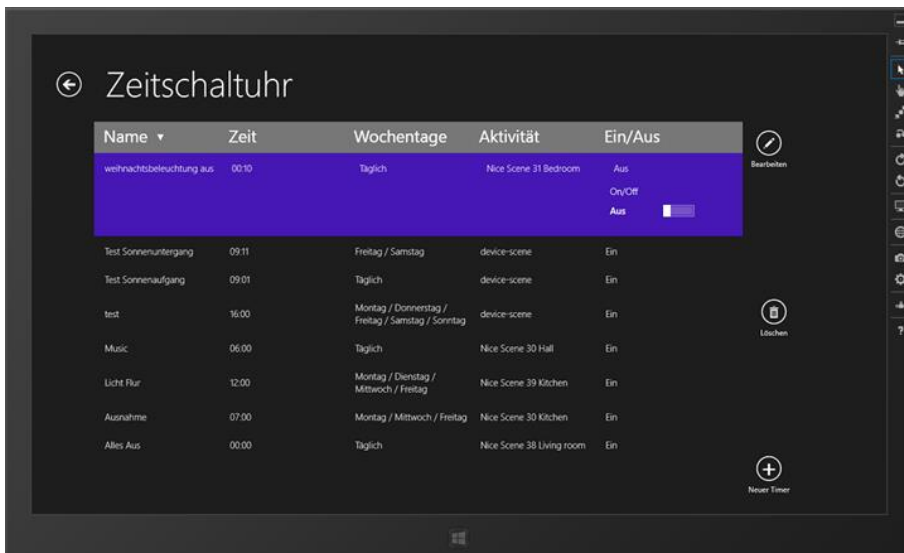


Abbildung 31 Aufklappen eines Events um ihn aktiv oder passiv zu stellen

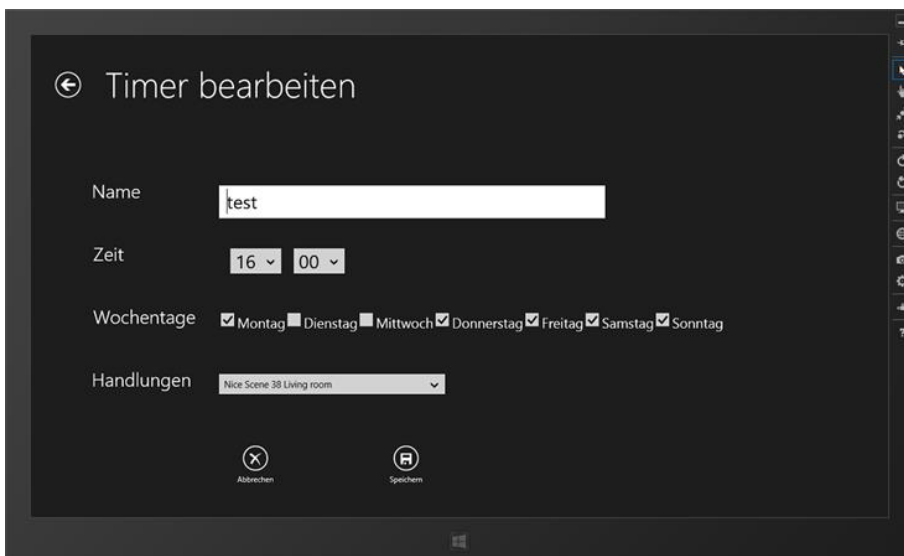


Abbildung 32 Bearbeiten eines Events

8.2.5. Foto Seite

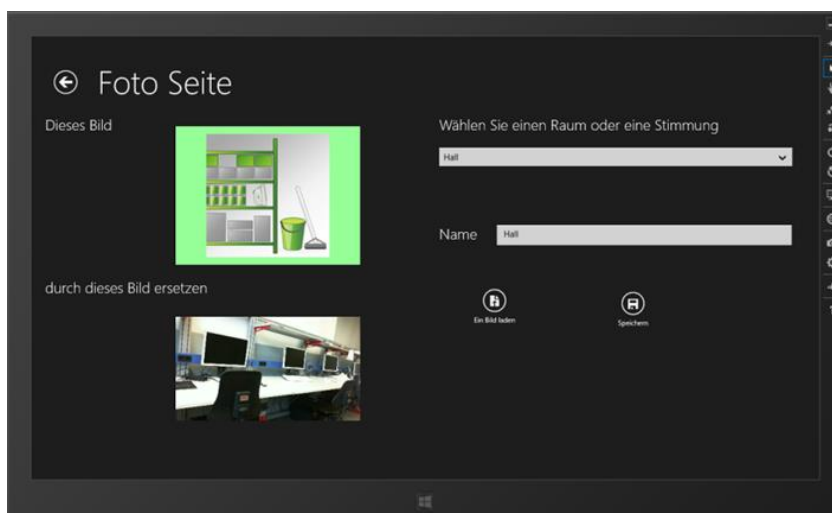
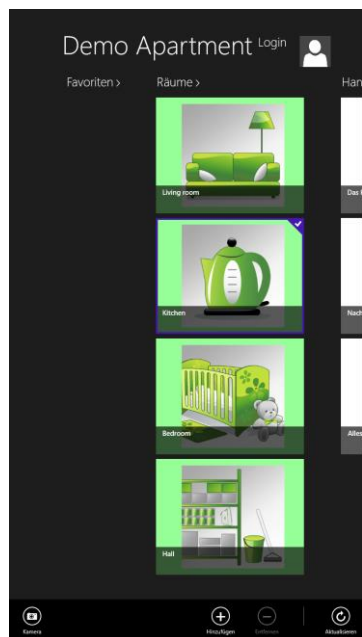


Abbildung 33 Foto Seite

Abbildung 34 AppBar

Man kann einen Raum auswählen und aufs Fotopiktogramm klicken. Danach erscheint die Kamera, die ein Foto schießt. Wenn es gefällt, kann man es zuschneiden oder einfach auf weiter klicken

Dann wird das alte Foto durch das neue beim Abspeichern ersetzt. Ein Raum oder eine Stimmung ist so individuell gestaltbar.

Projektmanagement

9. Projektmethodik

Als Projektmethodik verwendeten wir Scrum. Am Anfang der Arbeit teilten wir die Entwicklung der App in fünf Sprints ein und wiesen jedem Sprint seine Aufgaben zu. Gegen Ende des Projekts gab es eine Verzögerung mit dem Upload in dem Windows Store, da sich einige Aufgaben komplexer erwiesen als vermutet. Schlussendlich konnte der Projektplan, dank unserer intensiven Einarbeitung, doch noch gut eingehalten werden.

9.1. Sitzungsprotokolle

Während der gesamten Projektphase gab es einmal pro Woche eine Fortschrittsbesprechung mit dem Betreuer Prof. Hansjörg Huser und seiner Assistentin Franziska Schäpper. Die Protokolle befinden sich im Anhang.

9.2. Code Review

Am Donnerstag, 7. November 2013 wurde bei der Firma Noser AG ein Code Review durchgeführt. Das Protokoll und die Auswertung davon befinden sich ebenfalls im Anhang.

9.3. Team Foundation Server

Als Team Foundation Server (TFS) wurde derjenige von Noser AG zur Verfügung gestellt. Die History befindet sich im Anhang. (Kapitel 18)

9.4. Sprints

9.4.1. Sprint 1

Ziele

Entwicklungsumgebung eingerichtet

Projektplan erstellt

Hauptaufgaben

Einrichten der Entwicklungsumgebung, Installation von Windows 8.1 und Visual Studio 2013.

Selbststudium über Windows 8.1 App. [11] [12]

Erstellen von Projektplan, Anforderungsanalyse, Risikoanalyse.

Studium API Demogerät (JSON Interface Dokumentation) [15] [16]

Besuch bei der Firma aizo, abholen des Entwicklungsgerätes / Demogerät und Ansicht des Demo Apartments mit digitalSTROM.

9.4.2. Sprint 2

Ziele

Design fertig geplant

Hauptaufgaben / Fokussierung im Sprint

Webinterface und Testgerät kennenlernen

Domain Model, Architektur

Use Cases

Paper prototype und UI Login

Implementation Login

Probleme

Der digitalSTROM Server im Developer Kit fand die Geräte mit den Klemmen nicht. Wir führten verschiedene Tests durch, in verschiedenen Stromkreisen mit verschiedenen Einstellungen, aber die Geräte wurden nicht gefunden. Schliesslich blieb nur noch die Lösung, dass ein Teil aus dem Development Kit fehlt. Netterweise durften wir einen Stecker IEC320 <-> T13, von welchem wir vermuteten, dass er benötigt wird, von einem Elektrogeschäft ausleihen, um erneut das Gerät zu testen. Mit diesem Stecker direkt am Testgerät angeschlossen, fand der Server schliesslich die Geräte, worauf wir einen solchen Stecker kauften. Später wurde uns von Miguel Rodriguez von der Firma aizo bestätigt, dass wir ein solches Verbindungsstück benötigen und wir dieses mit einem Windows Surface Tablet abholen konnten.

9.4.3. Sprint 3

14.10.2013 – 08.11.2013

Ziele

Implementation Stimmungssteuerung

Hauptaufgaben / Fokussierung im Sprint

Installieren und konfigurieren des Surface Tablets von Windows 8.0 auf Windows 8.1

An- und Abschaltknopf für eine ausgewählte Stimmung

Über das Netzwerk und mit Hilfe von JSON Befehle absetzen

Designndokumentation

9.4.4. Sprint 4

11.11.2013 – 29.11.2013

Ziele

Implementation Stromverbrauchsanzeige

Favoriten verwalten

Hauptaufgaben / Fokussierung im Sprint

Die verschiedenen Seiten mit den dazugehörigen ViewModels gestalten

Den Stromverbrauch abfragen

Favoriten hinzufügen / löschen

Code Review bei Noser AG

Probleme

Verspäteter Beginn als Folge des verzögerten Abschlusses von Sprints 3. Damit das App trotzdem am Ende des Projektes mit allen, von uns geplanten Aufgaben abgeschlossen ist, verschoben wir den Upload in den Windows Store auf den Projektabschluss.

Auf den Code Review bei Noser AG folgte der Versuch, das Core Projekt mit einer PCL zu schreiben. In Rücksprache mit aizo AG wurde der Punkt einer PCL schliesslich verworfen, wie unter Kapitel 6.1.1 beschrieben ist.

9.4.5. Sprint 5

02.12.2013 – 13.12.2013

Ziele

Implementation der Zeitschaltuhr

Fotos abspeichern und mit Räume und Stimmungen verknüpfen

App fertig stellen

Hauptaufgaben / Fokussierung im Sprint

Zeitschaltuhrseite mit einer Tabelle, die sortiert werden kann und editierbar ist, erstellen

Zeitschaltuhr implementieren

Eigene Fotos auf LocalStorage abspeichern und laden

Probleme

Zeitschaltuhr Webservice funktionierte nicht zuverlässig. Als Antwort kam „ok“ zurück, bevor dieser seine Arbeit durchgeführt hatte. Somit konnte nicht überprüft werden, ob der Event tatsächlich gesetzt worden war.

9.4.6. Projektabschluss

16.12.2013 – 20.12.2013

Ziele

Upload der App in den Windows Store

Plakat

Abschluss Dokumentation

Abgabe der Bachelorarbeit

Anpassung der App an Vorgaben für Windows Store

Hauptaufgaben / Fokussierung im Sprint

Beim Testen der App mit dem Windows App Certification Kit, erhielten wir das folgende positive Ergebnis. Somit stand einem Upload in den Windows Store nichts mehr im Wege.

Windows App Certification Kit – Testergebnisse

Gesamtergebnis: **BESTANDEN**

Test von Abstürzen oder Hängen

Bachelorarbeit HS 2013

Datum: 20.12.2013

BESTANDEN Abstürze und Hängen

Test der Einhaltung des App-Manifests

BESTANDEN App-Manifest

Test der Windows-Sicherheitsfunktionen

BESTANDEN Analyser für Binärdateien

BESTANDEN Analyser für gesperrte Dateien

BESTANDEN Private Codesignatur

Test der unterstützten APIs

BESTANDEN Unterstützte APIs

Test der Leistung

BESTANDEN Generierung von Bytecode

BESTANDEN Optimierte Bindungsverweise

BESTANDEN Leistung beim Start

BESTANDEN Leistung beim Anhalten

Test der App-Manifestressourcen

BESTANDEN App-Ressourcenüberprüfung

BESTANDEN Brandingvalidierung

Test der Debugkonfiguration

BESTANDEN Debugkonfiguration

Dateicodierungstest

BESTANDEN UTF-8-Dateicodierung

Direct3D-Featuretest

BESTANDEN Direct3D-Unterstützung auf Funktionsebene

BESTANDEN Direct3D-Trimmen nach dem Anhalten

App-Funktionstest

BESTANDEN Sonderfunktionen

Windows Runtime-Metadatenvalidierung

BESTANDEN Test auf Attribut "ExclusiveTo"

BESTANDEN Test auf Typstandort

BESTANDEN Test auf richtige Groß-/Kleinschreibung von Typnamen

BESTANDEN Test auf Richtigkeit von Typnamen

BESTANDEN Test auf allgemeine Richtigkeit der Metadaten

BESTANDEN Eigenschaftentest

Paketintegritätsprüfung

BESTANDEN Der Plattform entsprechende Dateien

BESTANDEN Überprüfung der unterstützten Verzeichnisstruktur

Ressourcenverwendungstest

BESTANDEN Test für WinJS-Hintergrundaufgaben

10. Anforderungsanalyse

10.1. Einführung

10.1.1. Zweck

Die Firma aizo AG stellt elektronische Steuerungen für den Haushalt her. Diese können über das angeschlossene Stromnetzwerk mit einer Schaltung (Klemme) oder einem Server kommunizieren. Die Firma Noser Engineering AG in Zusammenarbeit mit aizo AG möchte eine Mobile App für Windows 8, welche alle Geräte im Haus zentral steuert.

Zuerst kann der Bediener der App das Layout der Räume aufnehmen und im Tablet Computer oder Telefon digitalisieren. Danach wird die Position der zu steuernden Geräte im Plan eingezeichnet. Die Geräte, wie zum Beispiel eine Tischlampe, kann mit einem eindeutigen Punktemuster markiert werden, welches die Kamera des mobilen Gerätes erkennt und inventarisiert.

Nachdem das Haus aufgenommen ist, dient die App als Steuerung und Programmierung der angeschlossenen Geräte. Zusätzlich wird eine Statistik über den Stromverbrauch erstellt. Viel benutzte Geräte mit hohem Verbrauch können dann identifiziert werden und der Benutzer hat die Möglichkeit, geeignete Massnahmen zu treffen.

10.2. Allgemeine Beschreibung

10.2.1. Produktperspektive

Die Windows 8 App wird in einem ersten Schritt für die Bedürfnisse der Heimbewohner entwickelt. Diese können mit der App ihre Wohnung steuern und Statistiken über den Verbrauch abfragen.

Für die Zukunft sind weitere Funktionen geplant, wie zum Beispiel einen zentralen Server, welcher mehrere Haushalte steuert. Dieser kann sich zum Beispiel bei einer Versicherung befinden, der bei einer schlechten Wetterlage die Jalousien herunterlässt.

Auch wäre vorstellbar einen Anschluss an eine Cloud für das Archivieren von Messwerten und das Steuern über eine Remote Verbindung.

10.2.2. Produkt Funktionen

10.2.2.1. Hauptfunktionen der App

- Registration
- Schnellanzeige der Favoriten um häufige Handlungen zu steuern.
- Anzeige von Räumen mit ihren Geräten und Steuern von deren Stimmungen.
- Anzeigen des Stromverbrauches
- Mit dem Start der App werden alle Konfigurationen und Einstellungen vom Server gelesen. Danach sind die Einstellungen statisch auf der App vorhanden.

10.2.2.2. Priorität 2 Funktionen der App

- Der User kann mit einem Sprachbefehl ein Licht ein- oder ausschalten. Die App zeigt an was sie Verstanden hat und welche Befehle ausgeführt werden.
- Mit Hilfe der Zeitschaltuhr können die Stimmungen zu einer bestimmten Zeit ein- und ausgeschaltet werden.

10.2.2.3. Priorität 3 Funktionen der App

- Über die App können Handlungen definiert werden. Der User kann zum Beispiel einstellen, welche Geräte bei der Handlung „Aufstehen“ eingeschaltet werden sollen.

10.2.3. Benutzer Charakteristiken

Die Zielgruppe dieser App sind Eigenheimbesitzer, die eine intelligente Wohnung mit DigitalSTROM besitzen. Durch die Installation der Steuerungsmodule haben diese einen Grundstein für die Applikation gelegt. Durch die einfach bedienbare App für das Windows Tablet ist es möglich, mehrere Geräte auf einen Knopfdruck zu bedienen. Zum Beispiel, dass beim Verlassen des Hauses alle Lichter ablöschen.

Es wird angenommen, dass der Benutzer Freude an der modernen Technik hat und dadurch ein Grundverständnis mitbringt.

10.2.4. Abhängigkeiten

Der Benutzer hat eine Installation von der Firma aizo AG im Haushalt und er besitzt Windows 8 fähige Hardware. Auch braucht es einen WLAN Empfang zum Accesspoint.

11. Persönliche Berichte

11.1. Annrita Egli

Aufgabenstellung

Die gestellte Aufgabe umfasst die Entwicklung einer komfortablen, einfach verständlichen Bedienung einer Anlage, welche in einem Haushalt sämtliche Verbraucher elektrischer Energie gemäss vorzugebenden bzw. vorgegebenen Parametern ein- oder ausschaltet. Diese, „App“ genannte Bedienung soll auf Windows 8-Betriebssystem basierten Tablet-Computer lauffähig sein. Mit dieser rudimentären Beschreibung ist die zu lösende Aufgabe jedoch nur sehr unvollständig umrissen. Es galt deshalb in einer ersten Phase, den Funktionsumfang der zu entwickelnden App in Zusammenarbeit mit dem Auftraggeber exakt zu definieren.

Schwierigkeitsgrad

Dank vorliegender Beschreibungen der Anlagefunktionen war die Einarbeitung in die Problematik relativ schnell möglich. Hilfreich war auch die Tatsache, dass bereits Android und iOS basierte Lösungen existieren, welche quasi als Messlatte für das zu erzielende Resultat dienen können. Allerdings zeigte sich während der Realisierung unserer App, dass die zu steuernde Anlage eine sehr hohe Anzahl von Funktionen aufweist und so konzipiert ist, dass die Erweiterung von bestehenden Funktionen sowie die Implementation neuer Funktionen jederzeit möglich sind. Die auszubereitende App muss deshalb bezüglich ihrer Erweiterbarkeit sehr offen sein und Ergänzungen jederzeit erlauben.

Lösungsfindung

Die Lösungsfindung im Team und in Zusammenarbeit mit dem Auftraggeber glich dem Zusammensetzen eines Mosaiks, wobei die einzelnen Steine nicht von Anfang an definiert waren. Da ein detailliertes technisches Pflichtenheft fehlte, musste dieses quasi parallel zur Lösung erarbeitet werden. Dies muss aber nicht schlecht sein, fand doch dadurch ein intensiver Austausch mit dem Auftraggeber statt, der oft zu klareren Definitionen der Funktionen führte oder die Aufnahme zusätzlicher Eigenschaften in den Funktionsumfang sinnvoll erscheinen liess. Dies erhöhte zwar den Schwierigkeitsgrad und die Komplexität der Aufgabe, führte jedoch auch zu einer umfassenderen und vollständigeren Lösung.

Zusammenfassung

Die Aufgabe hat sich umfassender und komplexer herausgestellt als bei Beginn angenommen. Dazu beigetragen hat der relativ offen definierte Funktionsumfang der Anlage. Damit wurde aber die Aufgabe anspruchsvoller und erlaubte einen tiefen Einblick in die Programmierung Hardware-naher Software. Das Vorhandensein der Anlage in Form eines Prototyps gab uns die Möglichkeit zum vollständigen Testen der Programmfunktionen und somit einen reellen Praxisbezug. Die direkten Kontakte zum Auftraggeber waren eine Bereicherung und vereinfachten die laufende Überprüfung des Arbeitsfortschritts.

11.2. Richard Schulthess

Themenwahl

Das Thema dieses Projektes hat mich von Anfang an sehr interessiert, da ich bereits über ein gewisses Grundwissen diesbezüglich verfügte. Als Technikfreak hatte das Entwickeln einer eigenen App einen hohen Stellenwert und war mir sehr nahe gewesen.

Projekt

Beim ersten Kickoff- Meeting wurde mir bewusst, wie viele Personen in diesem Projekt involviert und wie viele unterschiedliche Vorstellungen über die Realisation vorhanden waren.

Ich hielt mir stets vor Augen, dass das App für jedermann einfach bedienbar sein sollte. Ein besonderes Interesse galt der Philosophie von Microsoft den Desktop PC mit einem Tablet Computer verschmelzen zu lassen, um ein klares und einfaches App als Window 8.1 zu entwickeln und zu präsentieren.

Am Anfang des Projektes war die Fülle der Beispiele von Anwendungen, die Microsoft zur Verfügung gestellt hat, sehr hilfreich. Auch die Dokumentation über die Testinstallation war verständlich.

Mit einem kleinen Konsolenprogramm konnte sehr rasch eine Verbindung aufgebaut und schon erste Abfragen gemacht werden.

Probleme / Herausforderungen

Als der erste Prototyp vom User Interface stand, machte ich den Fehler und implementierte ein altes Konzept von Windows 8. Dies hat mich dann wertvolle Zeit gekostet es für ein modernen Ansatz wieder umzugestalten.

Eine andere Herausforderung war, das App nach dem MVVM-Model zu bearbeiten, da die kleineren Beispielprojekte im Internet oder die Templates von Microsoft einen anderen Ansatz hatten.

Rückblick / Ausblick

Das Arbeiten im Team sowie mit den Projektpartnern war für mich eine sehr lehrreiche Erfahrung. Geschätzt habe ich insbesondere die Hilfestellungen aller Beteiligten.

Der Umgang des Visual Studios war sehr schnell zu erlernen und ich habe eine effiziente Arbeitsweise entdeckt.

Ich denke, dass wir mit dem entwickelten App in der Zeit sind und das Bedürfnis der Benutzer nach solchen Apps noch weiter zunehmen wird. Momentan sind wir mit dem entstandenen Produkt zufrieden, kann mir aber vorstellen, dass mit neuen Technologien die Anwendungen noch optimiert werden könnten.

Verzeichnisse

12. Glossar

Abkürzung / Begriff	Beschreibung
Aktivität	Die Aktivitäten beinhalten alle Stimmungen und Handlungen
Apartment	Ein Apartment nennt man das gesamte Gebäude, welches mit digitalSTROM eingerichtet wurde.
Chip	Ein Chip ist ein kleiner Minicomputer, der vor dem Gerät oder im Gerät eingebaut sein kann.
Circuit	Ein Apartment kann mehrere Circuits haben. Dort werden die digitalSTROM Geräte angeschlossen und der Stromverbrauch gemessen.
Device	Siehe Gerät
digitalSTROM Meter	Ein eingebauter Strom Zähler.
Ethernet	Ist ein Computernetzwerk.
Event	Siehe Timer
FI Schalter	Ist ein Fehlstromschutzschalter, der als Sicherung des Stromkreises fungiert.
Gerät	Mit Geräten sind die elektrischen Verbraucher, welche an die digitalSTROM Einrichtung angeschlossen sind, gemeint. Jedes Gerät ist einem Raum und einer Gruppe zugeordnet.
Gruppe	In einer digitalSTROM Einrichtung werden die Geräte in Gruppen unterteilt. Es gibt folgende Gerätegruppen: <ul style="list-style-type: none"> 1 Lichter 2 Rollläden 3 Klimaanlage, Heizung 4 Audio 5 Video, TV 6 Sicherheit, Alarm 7 Zugang, Klingel 8 Joker
Handlung	Auf dem App gibt es die drei Handlungen: „Das Haus verlassen“, „Nach Hause kommen“ und „Alles ausschalten“. Wird eine von diesen Handlungen ausgewählt, so werden die elektrischen Verbraucher im gesamten Haus auf den voreingestellten Status für diese Handlung eingestellt.
Klemme	Durch die Klemme fließt der Strom, welcher Daten und Energie liefert. In der Klemme ist ein Chip integriert. An einer Klemme kann man jedes Gerät anschliessen.
Raum	Siehe Zone
Scene	Siehe Stimmung
Schaltung	Siehe Circuit
SSL	Ein Netzwerkprotokoll zum Verschlüsseln von Netzwerkdaten.
Stimmung	Jeder Zone im Haus sind vier Stimmungen (Scenes) zugeordnet. Wird eine Stimmung ausgewählt, so stellen sich alle Geräte auf den voreingestellten Status für diese Stimmung ein. Man kann zum Beispiel eine Stimmung „Lesen“ einstellen, wo sich alle Leselampen einschalten.

Timer	Ein Timer ist eine Aktivität, welche zu einer bestimmten Zeit ausgeführt werden soll. Ein solcher Eintrag in der Zeitschaltuhr wird auch als Event bezeichnet.
Webinterface	Mit dem Browser kann die digitalSTROM Website abgerufen werden.
Windows Store	Der Microsoft Marktplatz ist für den Vertrieb von Windows Apps verantwortlich.
Zone	Ein Apartment ist in verschiedene Zonen unterteilt. Wie diese Zonen aussehen, ist den Bewohnern des digitalSTROM Hauses überlassen, meistens ist jeder Raum eine eigene Zone. Man kann aber auch grosse Räume in mehrere Zonen unterteilen.

13. Literaturverzeichnis

13.1. Bücher

- [1] C# 5.0 in a Nutshell, Fifth Edition by Joseph Albahari and Ben Albahari

13.2. Internet

digitalSTROM:

- [2] <http://www.digitalstrom.com/?gclid=CNOI49OwubsCFQIb3godxmUAMg>

Login:

- [3] <http://blogs.msdn.com/b/wsdevsol/archive/2013/10/17/how-to-ignore-self-signed-certificate-errors-in-windows-store-apps-8-1.aspx>

PCL für Windows Phone 8 Login:

- [4] <http://social.msdn.microsoft.com/Forums/windowsapps/en-US/4a776e8c-0e10-4f03-908f-7f765d914080/how-to-connect-to-a-http-rest-service-with-self-signed-certificatein-the-server-side?forum=winappswithcsharp>
- [5] <http://social.msdn.microsoft.com/Forums/windowsapps/en-US/b6887fa1-1b29-4747-a0f3-573e77cd10ea/can-i-tell-windows-8-apps-to-ignore-security-certificate-errors>

Lokalisierung:

- [6] <http://msdn.microsoft.com/en-us/library/windows/apps/hh965328.aspx>
- [7] <http://code.msdn.microsoft.com/windowsapps/Application-resources-and-cd0c6eaa>

Linq:

- [8] <http://msdn.microsoft.com/en-us/library/bb397926.aspx>

PCL:

- [9] <http://msdn.microsoft.com/en-us/library/vstudio/gg597391%28v=vs.100%29.aspx>

Manueller JSON Converter:

- [10] <http://stackoverflow.com/questions/17745814/deserialize-the-json-where-the-values-are-field-names-with-json-net>

Windows App Entwicklung für Einsteiger:

- [11] <http://msdn.microsoft.com/de-de/library/windows/apps/hh974581.aspx>
- [12] <http://channel9.msdn.com/Series/Windows-Store-apps-for-Absolute-Beginners-with-C->

13.3. Dokumente

digitalSTROM Installationshandbücher:

- [13] http://www.aizo.com/de/support/documents/digitalSTROMInstallationshandbuch_A1121D002V009_DE_2013-05-22.pdf
- [14] http://dsid.org/de/support/documents/digitalSTROM_Installationshandbuch.pdf

API digitalSTROM Server:

- [15] <http://developer.digitalstrom.org/Architecture/>
- [16] <http://developer.digitalstrom.org/download/dss/1.0/dss-1.0.0-doc/>

[17] <http://redmine.digitalstrom.org/projects/dss/wiki/TimedEventsInternals>

14. Tabellenverzeichnis

Tabelle 1 Codemetrik Berechnung	17
Tabelle 2 Login Methode	21
Tabelle 3 Webservice GetApartmentName	24
Tabelle 4 Webservice SetApartmentName	25
Tabelle 5 Webservice GetStructure	25
Tabelle 6 Webservice GetReachableGroups	26
Tabelle 7 Webservice GetCircuits	27
Tabelle 8 Webservice CallSceneBySceneNumber	28
Tabelle 9 Webservice GetConsumption	29
Tabelle 10 Webservice GetEnergyMeterValue	29
Tabelle 11 Webservice ZoneCallScene	30
Tabelle 12 Webservice GetReachableScenes	30
Tabelle 13 Webservice GetSceneName	31
Tabelle 14 Webservice SetSceneName	31
Tabelle 15 Webservice SetRoomName	32
Tabelle 16 Webservice GetOutputValue	32
Tabelle 17 Webservice GetTimedEvents	33
Tabelle 18 Webservice SetTimedEvents	34
Tabelle 19 Webservice DisableTimedEvent	35
Tabelle 20 Webservice EnableTimedEvent	35
Tabelle 21 Webservice DeleteTimedEvent	36
Tabelle 22 DataCache GetApartmentAsync	38
Tabelle 23 DataCache SetApartmentNameAsync	38
Tabelle 24 DataCache CallActivity	38
Tabelle 25 DataCache ZoneCallSceneAsync	38
Tabelle 26 DataCache SetRoomNameByld	39
Tabelle 27 DataCache SetSceneNameAsync	39
Tabelle 28 DataCache GetTimedEventsAsync	39
Tabelle 29 DataCache SetTimedEventAsync	40
Tabelle 30 DataCache DisableTimedEventAsync	40
Tabelle 31 DataCache EnableTimedEventAsync	40
Tabelle 32 DataCache DeleteTimedEventAsync	40
Tabelle 33 DataCache IsActiveScene	41

15. Abbildungsverzeichnis

Abbildung 1 digitalSTROM bisher	11
Abbildung 2 Demogerät von aizo AG	12
Abbildung 3 Development Kit von aizo.....	13
Abbildung 4 Use Case Diagramm	16
Abbildung 5 Zertifikat von digitalSTROM	18
Abbildung 6 Architektur MVVM.....	19
Abbildung 7 Aufruf am Beispiel der Methode GetApartmentNameAsync	20
Abbildung 8 Klassen im namespace digitalSTROM.Core.Network.....	21
Abbildung 9 Ablauf Login.....	24
Abbildung 10 Klasse im namespace digitalSTROM.Core	37
Abbildung 11 DataCache - CommonDataSource.....	37
Abbildung 12 Überblick namespaces im DataModel.....	41
Abbildung 13 Datenmodell für die Struktur des Apartments	42
Abbildung 14 Datenmodell um die Circuits des Apartments herauszulesen	43
Abbildung 15 Datenmodell um den EnergyMeterValue herauszulesen.....	43
Abbildung 16 Datenmodell um Consumption herauszulesen	44
Abbildung 17 Datenmodell Zone	44
Abbildung 18 Klassen im namespace digitalSTROM.Core.JsonConverters	45
Abbildung 19 namespace DataSource	50
Abbildung 20 Hubpage	50
Abbildung 21 DataBinding auf der Hubpage	51
Abbildung 22 Webinterface von digitalSTROM	53
Abbildung 23 Hauptseite mit einem Favorit.....	53
Abbildung 24 Anmelden	54
Abbildung 25 Gruppe Favoriten.....	54
Abbildung 26 Gruppe Favoriten mit der Möglichkeit Favoriten hinzuzufügen.....	55
Abbildung 27 Seite mit der Gruppe von Räume.....	55
Abbildung 28 Raum Detailansicht	56
Abbildung 29 Gruppe Handlungen	56
Abbildung 30 Zeitschaltuhr mit Events	57
Abbildung 31 Aufklappen eines Events um ihn aktiv oder passiv zu stellen.....	57
Abbildung 32 Bearbeiten eines Events	57
Abbildung 33 Foto Seite	58
Abbildung 34 AppBar	58
Abbildung 36 CommonDataSource beim Starten der App.....	74
Abbildung 37 Navigation Sequenz Diagramm.....	75
Abbildung 38 Ablauf Login.....	75
Abbildung 39 Ablauf beim Aufruf einer Stimmung.....	76

Anhang

16. Risikomanagement

Risiko Analyse							
Projektname: Windows 8.1 App digitalSTROM							
Projektteam: Richard Schulthess, Annrita Egli							
Risiko Bewertungen							
Risk ID	Risiko	Auswirkung	Massnahme	Aufwand Massnahmen in h	Max. Schaden in h	Wahrsch'keit des Eintreffens	Priorität
R01	Einarbeitung in neue Technologien dauert länger als erwartet	Sprints benötigen mehr Arbeitszeit als geplant	Genug Einarbeitungszeit einplanen	40	100	50%	1
R02	Zerstörung der Hardware-Komponenten des Live-Systems durch Software-Fehlfunktionen	Demogerät kann nicht mehr gebraucht werden	API Demogerät studieren	5	20	15%	2
R03	Schlechte Qualität der Informationen, die der Kunde liefert (z.B. über das spätere Einsatzumfeld der App)	Die Architektur des Systems wird unbrauchbar spezifiziert	Kunde und End-User/Experten in den Entwicklungsprozess einbeziehen und ständigen Kontakt halten	20	N/A	10%	2
R04	End-User finden sich im späteren Produktiv-Einsatz nicht zurecht mit der App	Kunde unzufrieden und arbeitet nicht mit dem System.	Kunde und End-User/Experten in den Entwicklungsprozess einbeziehen und ständigen Kontakt halten	20	N/A	10%	2
R05	Lange Lieferfristen und hohe Wiederbeschaffungskosten für defekte Teile des Test-Systems, die während der Entwicklungsphase ersetzt werden müssen	Es fallen Wartezeiten für Reparaturen an	API Demogerät studieren	5	20	20%	3
R06	Personal-Ausfall während Entwicklungsphase durch Krankheit	Unzugängliches Know-how. Projektabschluss kann nur durch Nachtschicht sichergestellt werden.	Knowhowtransfer sicherstellen	5	20	30%	3
R07	Fehleinschätzung in der Zeitplanung durch Einarbeitung in neue Technologie	Langsamere Projekt-Fortschritt und zusätzliche Schulungs-Kosten	Genug Einarbeitungszeit einplanen und Experten hinzuziehen	40	80	40%	1
R08	Stabilität der Kunden-Anforderungen im Projekt-Verlauf (z.B. Kunde ändert immer wieder die Anforderungen)	Projekt-Abschluss verzögert sich.	Anforderungsanalyse ab einem bestimmten Zeitpunkt fixieren (alles was danach kommt, muss später umgesetzt werden)	8	120	5%	1
Total Kosten in Arbeitspaketen enthalten				143	360		

17. Diagramme

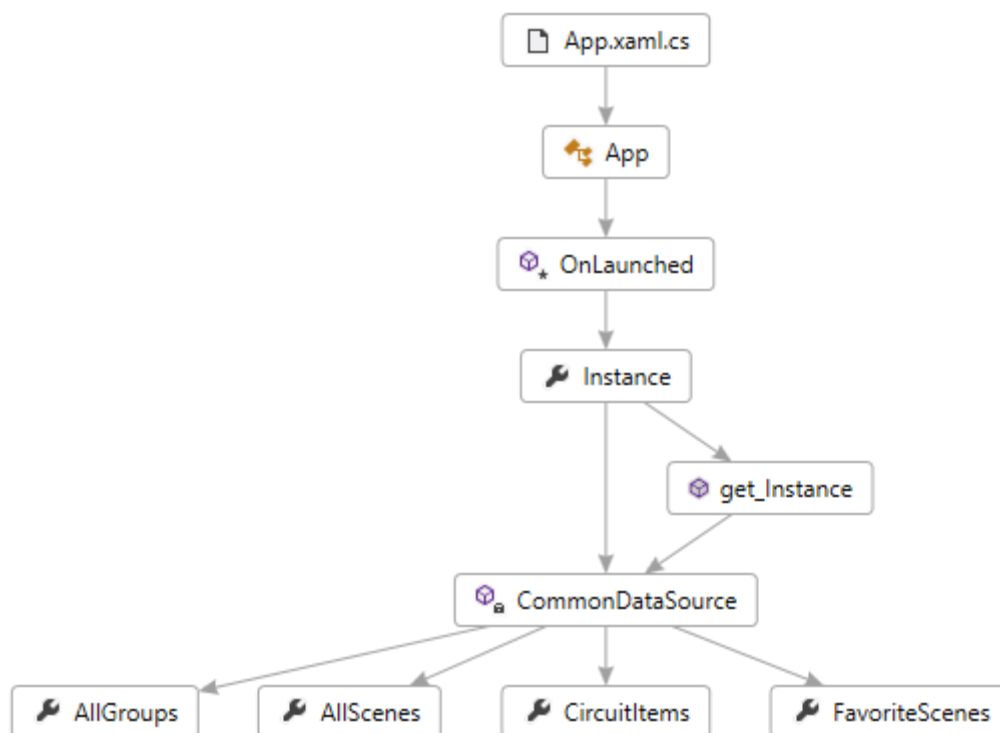


Abbildung 35 CommonDataSource beim Starten der App

sd NavigationSequenceDiagram

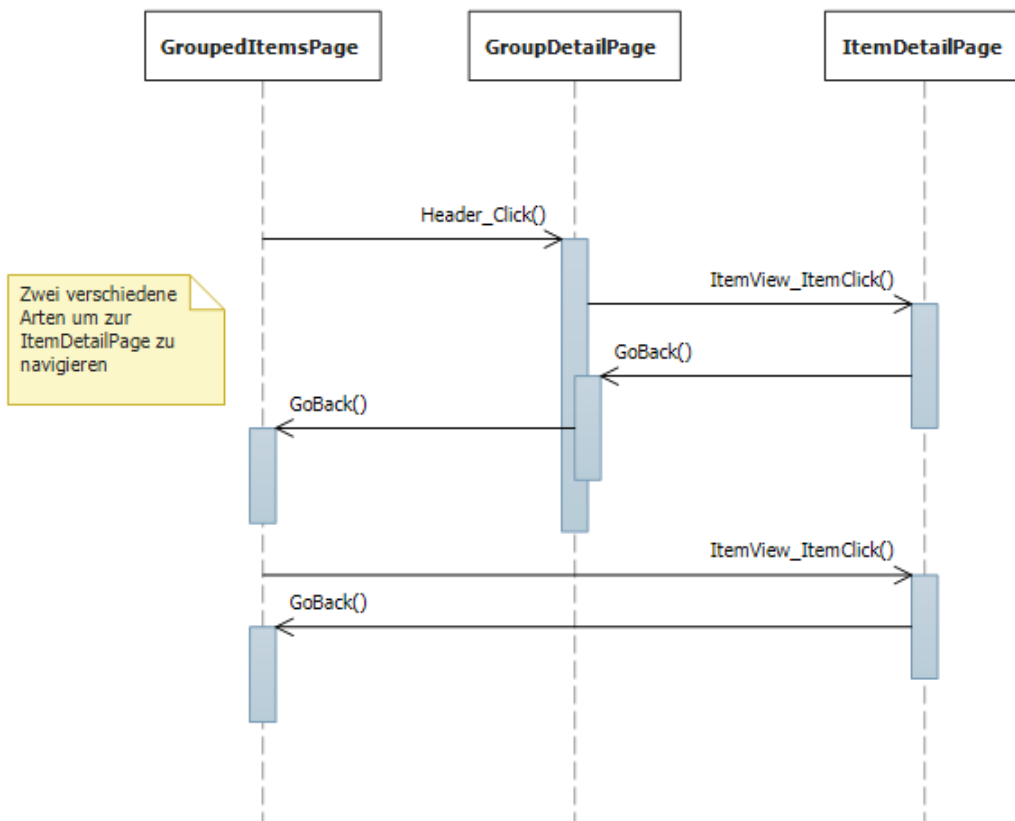


Abbildung 36 Navigation Sequenz Diagramm

sd Login

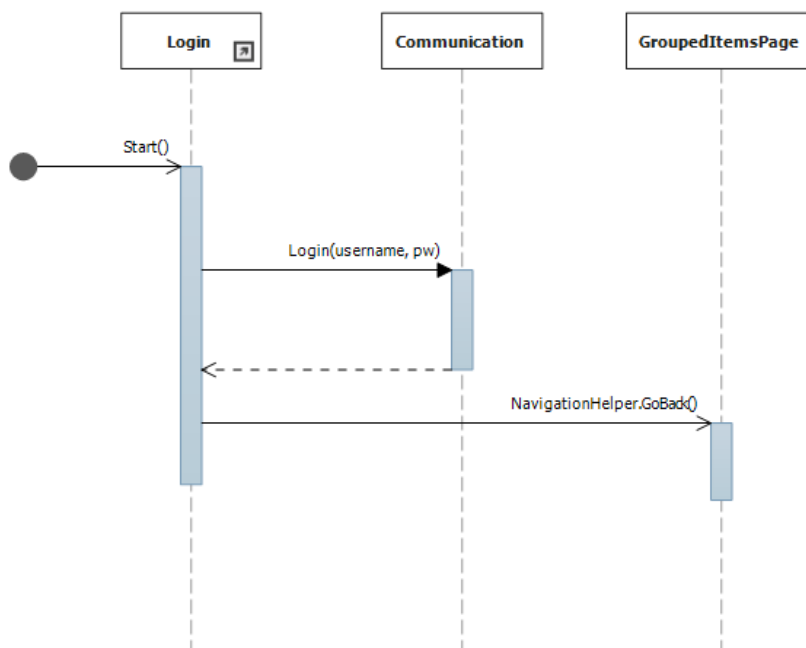


Abbildung 37 Ablauf Login

sd ActionSequenceDiagram

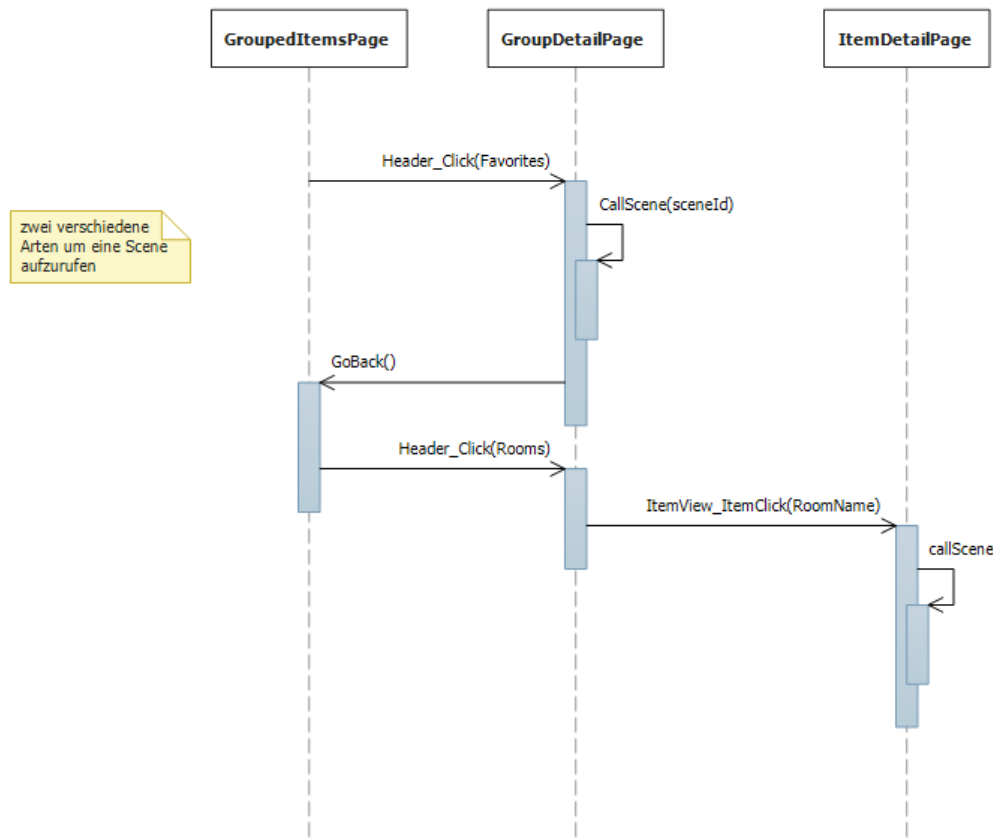


Abbildung 38 Ablauf beim Aufruf einer Stimmung

18. TFS History

Change-set	Benutzer	Datum	Kommentar
21929	DMZ\annrita.egli	Freitag, 19. Dezember 2013 00:30:49	fix method FindsSceneltem, final version
21917	DMZ\annrita.egli	Donnerstag, 19. Dezember 2013 13:55:58	rename 1 parameter, remove unused this
21914	DMZ\annrita.egli	Mittwoch, 18. Dezember 2013 22:57:49	fix exception on photopage
21902	DMZ\annrita.egli	Mittwoch, 18. Dezember 2013 13:25:11	handle exception
21895	DMZ\annrita.egli	Mittwoch, 18. Dezember 2013 10:39:12	fix merge
21888	DMZ\annrita.egli	Mittwoch, 18. Dezember 2013 08:07:28	digitalStrom => digitalSTROM
21864	DMZ\annrita.egli	Montag, 16. Dezember 2013 22:57:05	insert storage for login data, add and implement function setApartmentName
21848	DMZ\annrita.egli	Sonntag, 15. Dezember 2013 21:11:07	unit test & login refactoring
21846	DMZ\annrita.egli	Samstag, 14. Dezember 2013 14:55:25	handlung => aktivität on timer page. version app store
21845	DMZ\annrita.egli	Samstag, 14. Dezember 2013 10:00:46	refactoring
21844	DMZ\annrita.egli	Freitag, 13. Dezember 2013 20:02:36	complete resource file
21843	DMZ\annrita.egli	Freitag, 13. Dezember 2013 19:09:47	add file

21842	DMZ\annrita.egli	Freitag, 13. Dezember 2013 19:05:39	insert demodata class
21837	DMZ\annrita.egli	Freitag, 13. Dezember 2013 16:49:18	insert different activity types
21834	DMZ\annrita.egli	Freitag, 13. Dezember 2013 11:51:33	insert labels in resourcefiles
21832	DMZ\annrita.egli	Donnerstag, 12. Dezember 2013 22:40:54	progress ring on addnewtimeevent
21831	DMZ\annrita.egli	Donnerstag, 12. Dezember 2013 18:26:26	textfield addnewtimer name
21823	DMZ\annrita.egli	Donnerstag, 12. Dezember 2013 12:55:29	delete empty timed event entries
21820	DMZ\annrita.egli	Donnerstag, 12. Dezember 2013 10:31:26	ignore empty entries in timedevents
21812	DMZ\annrita.egli	Mittwoch, 11. Dezember 2013 23:24:57	call scenes via favorites, check connectivity
21801	DMZ\annrita.egli	Mittwoch, 11. Dezember 2013 14:05:35	insert check IsConnected
21799	DMZ\annrita.egli	Mittwoch, 11. Dezember 2013 10:26:14	insert and implement timer-remove button
21797	DMZ\annrita.egli	Mittwoch, 11. Dezember 2013 09:22:06	catch WebException
21796	DMZ\annrita.egli	Mittwoch, 11. Dezember 2013 00:25:45	refresh timer view
21781	DMZ\annrita.egli	Dienstag, 10. Dezember 2013 16:26:08	sceneld = string in parameter
21780	DMZ\annrita.egli	Dienstag, 10. Dezember 2013 16:00:17	completed unittest
21777	DMZ\annrita.egli	Dienstag, 10. Dezember 2013 14:38:32	insert login unit tests
21769	DMZ\annrita.egli	Dienstag, 10. Dezember 2013 13:38:20	add method SetRoomName for photopage
21766	DMZ\annrita.egli	Dienstag, 10. Dezember 2013 10:58:05	update timerview after edit
21761	DMZ\annrita.egli	Montag, 9. Dezember 2013 23:32:50	ignore empty scenesId
21760	DMZ\annrita.egli	Montag, 9. Dezember 2013 22:17:19	Ws => Wh
21758	DMZ\annrita.egli	Montag, 9. Dezember 2013 16:57:23	fix login - replace "" with null
21751	DMZ\annrita.egli	Montag, 9. Dezember 2013 14:03:12	rename constantes
21746	DMZ\annrita.egli	Montag, 9. Dezember 2013 07:51:47	bind toggleswitch
21745	DMZ\annrita.egli	Sonntag, 8. Dezember 2013 23:07:30	disable edit button in timer page when no event is selected
21743	DMZ\annrita.egli	Sonntag, 8. Dezember 2013 13:18:12	add new or edit timedevent, fix navigation
21737	DMZ\annrita.egli	Freitag, 6. Dezember 2013 13:24:52	fix method EnableTimeEvents
21731	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 22:09:28	insert unit test
21730	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 21:06:33	delete string resource in unit tests
21729	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 17:56:26	insert unit test, fix method in datacache
21726	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 14:48:34	add sceneld to demodata. setEventItem with groupId parameter
21725	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 11:57:12	fix unit test & typo
21724	DMZ\annrita.egli	Donnerstag, 5. Dezember 2013 11:22:16	show weekdays for current culture
21722	DMZ\annrita.egli	Mittwoch, 4. Dezember 2013 20:52:41	insert ReachableGroup json answer

21717	DMZ\annrita.egli	Mittwoch, 4. Dezember 2013 14:22:27	fix on/off view, correct typo
21716	DMZ\annrita.egli	Mittwoch, 4. Dezember 2013 13:46:34	copy string folder in unit tests
21709	DMZ\annrita.egli	Dienstag, 3. Dezember 2013 16:35:19	refactoring with resharper
21684	DMZ\annrita.egli	Sonntag, 1. Dezember 2013 23:03:20	fix command binding in timepage
21682	DMZ\annrita.egli	Sonntag, 1. Dezember 2013 19:54:38	delete empty testclass
21680	DMZ\annrita.egli	Sonntag, 1. Dezember 2013 15:58:56	add unit test example for view model
21679	DMZ\annrita.egli	Sonntag, 1. Dezember 2013 11:51:58	fix login, insert eventtime converter
21677	DMZ\annrita.egli	Sonntag, 1. Dezember 2013 09:37:00	reorder unit tests
21674	DMZ\annrita.egli	Freitag, 29. November 2013 16:35:49	adjust EventItem, fix setEventItemMethod
21669	DMZ\annrita.egli	Freitag, 29. November 2013 13:53:04	catch empty eventaction exception
21665	DMZ\annrita.egli	Freitag, 29. November 2013 11:19:41	add unit tests with login, refresh page after reload
21664	DMZ\annrita.egli	Freitag, 29. November 2013 09:31:07	insert and implement refresh button
21662	DMZ\annrita.egli	Donnerstag, 28. November 2013 16:32:46	add method DeleteEvent
21661	DMZ\annrita.egli	Donnerstag, 28. November 2013 14:38:38	add zoneId & SceneId to TimeEvent
21654	DMZ\annrita.egli	Donnerstag, 28. November 2013 11:32:47	instanziate timerviewmodel
21653	DMZ\annrita.egli	Donnerstag, 28. November 2013 11:11:56	add TimerViewModel
21652	DMZ\annrita.egli	Donnerstag, 28. November 2013 11:02:02	add loadEventData
21645	DMZ\annrita.egli	Mittwoch, 27. November 2013 23:04:57	deserialize manual EventDescription
21643	DMZ\annrita.egli	Mittwoch, 27. November 2013 13:55:21	add resource in unittest, add unittest
21642	DMZ\annrita.egli	Mittwoch, 27. November 2013 10:35:35	add methods EnableTimeEvent, DisableTimeEvent
21641	DMZ\annrita.egli	Mittwoch, 27. November 2013 08:51:10	add timed events
21631	DMZ\annrita.egli	Dienstag, 26. November 2013 14:05:22	fix navigation to rooms and favorites
21628	DMZ\annrita.egli	Montag, 25. November 2013 23:58:08	fix and refactor login
21625	DMZ\annrita.egli	Montag, 25. November 2013 16:55:00	fix errortext loginpage
21623	DMZ\annrita.egli	Montag, 25. November 2013 15:34:36	Lokalization CircuitList
21618	DMZ\annrita.egli	Sonntag, 24. November 2013 23:19:48	implement localization, refactor
21617	DMZ\annrita.egli	Sonntag, 24. November 2013 20:00:05	show only rooms with devices
21616	DMZ\annrita.egli	Sonntag, 24. November 2013 13:55:43	insert set roomid
21615	DMZ\annrita.egli	Sonntag, 24. November 2013 13:37:52	add zoneId to callScene method
21612	DMZ\annrita.egli	Freitag, 22. November 2013 16:47:42	remove unused usings
21605	DMZ\annrita.egli	Freitag, 22. November 2013 13:59:04	work on datacache
21599	DMZ\annrita.egli	Freitag, 22. November 2013 10:50:53	add classes for user activities

21598	DMZ\annrita.egli	Freitag, 22. November 2013 10:49:57	add method GetUserActivites
21597	DMZ\annrita.egli	Freitag, 22. November 2013 09:16:26	insert datacontainers
21596	DMZ\annrita.egli	Freitag, 22. November 2013 09:13:44	add method GetOutputValue, rename nameresult => generalresult
21595	DMZ\annrita.egli	Donnerstag, 21. November 2013 22:30:38	list circuits
21594	DMZ\annrita.egli	Donnerstag, 21. November 2013 21:01:25	fix data model
21584	DMZ\annrita.egli	Donnerstag, 21. November 2013 13:08:57	consumption view change
21583	DMZ\annrita.egli	Donnerstag, 21. November 2013 12:20:00	remove unused code
21581	DMZ\annrita.egli	Donnerstag, 21. November 2013 11:18:14	remove not reachable code
21580	DMZ\annrita.egli	Donnerstag, 21. November 2013 10:55:55	add Method SetSceneName
21579	DMZ\annrita.egli	Donnerstag, 21. November 2013 10:54:00	add method SetSceneName
21576	DMZ\annrita.egli	Mittwoch, 20. November 2013 21:24:52	refactoring
21563	DMZ\annrita.egli	Mittwoch, 20. November 2013 09:53:22	refactor login page
21561	DMZ\annrita.egli	Mittwoch, 20. November 2013 09:32:07	ProgressRing for Login
21560	DMZ\annrita.egli	Dienstag, 19. November 2013 23:13:31	fix method ZoneCallScene and GetConsumption
21545	DMZ\annrita.egli	Dienstag, 19. November 2013 11:29:03	demo data in english
21544	DMZ\annrita.egli	Dienstag, 19. November 2013 09:59:26	dataModel
21539	DMZ\annrita.egli	Montag, 18. November 2013 13:25:46	4 scenes per zone
21529	DMZ\annrita.egli	Sonntag, 17. November 2013 22:06:36	add datamodel
21528	DMZ\annrita.egli	Sonntag, 17. November 2013 11:40:48	implement consumption
21525	DMZ\annrita.egli	Freitag, 15. November 2013 16:28:27	prepare zonepageviewmodel for consumption
21524	DMZ\annrita.egli	Freitag, 15. November 2013 15:53:54	prepare view for consumption
21523	DMZ\annrita.egli	Freitag, 15. November 2013 15:52:56	prepare view for consumption
21521	DMZ\annrita.egli	Freitag, 15. November 2013 11:30:35	split circuit data files
21519	DMZ\annrita.egli	Freitag, 15. November 2013 11:05:28	insert circuit data in commonDataSource
21518	DMZ\annrita.egli	Freitag, 15. November 2013 08:48:00	delete unused property
21517	DMZ\annrita.egli	Donnerstag, 14. November 2013 22:33:29	remove unused usings
21516	DMZ\annrita.egli	Donnerstag, 14. November 2013 22:32:20	CallScene by Id, refactoring with resharper
21514	DMZ\annrita.egli	Donnerstag, 14. November 2013 15:55:55	refactoring after code review
21509	DMZ\annrita.egli	Donnerstag, 14. November 2013 11:57:22	typo correction
21507	DMZ\annrita.egli	Donnerstag, 14. November 2013 10:35:17	changes after delete project2
21499	DMZ\annrita.egli	Mittwoch, 13. November 2013 21:28:37	add message for no devices
21498	DMZ\annrita.egli	Mittwoch, 13. November 2013 21:22:44	get sceneName in ui

21497	DMZ\annrita.egli	Mittwoch, 13. November 2013 19:41:40	list devices in room
21495	DMZ\annrita.egli	Mittwoch, 13. November 2013 16:59:33	reinsert scenes
21492	DMZ\annrita.egli	Mittwoch, 13. November 2013 10:06:43	merge weekend work
21466	DMZ\annrita.egli	Montag, 11. November 2013 08:31:59	fix list items exception ui
21465	DMZ\annrita.egli	Montag, 11. November 2013 00:00:07	reinsert databinding for loginname, insert property changed in SetZoneltms. implement setZoneltms, remove hardcoded scenes
21464	DMZ\annrita.egli	Sonntag, 10. November 2013 21:59:17	update unit tests
21461	DMZ\annrita.egli	Freitag, 8. November 2013 15:14:19	add method getRoomNameById
21453	DMZ\annrita.egli	Donnerstag, 7. November 2013 17:12:54	refactoring after architecture review
21451	DMZ\annrita.egli	Donnerstag, 7. November 2013 16:41:14	refactoring after architecture review
21450	DMZ\annrita.egli	Donnerstag, 7. November 2013 16:19:22	refactoring after architecture review
21442	DMZ\annrita.egli	Mittwoch, 6. November 2013 21:19:59	Loginobject for Login Method
21429	DMZ\annrita.egli	Dienstag, 5. November 2013 13:36:31	add unit tests, rename nameData
21426	DMZ\annrita.egli	Montag, 4. November 2013 21:57:53	insert circuit data and methods
21420	DMZ\annrita.egli	Montag, 4. November 2013 13:41:31	catch exception in list converter
21417	DMZ\annrita.egli	Sonntag, 3. November 2013 23:12:22	add GroupedItemsViewModel, add Bindings, add Scene in StructureData
21416	DMZ\annrita.egli	Sonntag, 3. November 2013 17:32:21	add ViewModel for GroupDetailView
21415	DMZ\annrita.egli	Sonntag, 3. November 2013 11:27:16	Split class datasource
21412	DMZ\annrita.egli	Freitag, 1. November 2013 15:29:04	add Sequence Diagram
21411	DMZ\annrita.egli	Freitag, 1. November 2013 15:03:37	Add Sequence Diagrams
21409	DMZ\annrita.egli	Freitag, 1. November 2013 11:51:39	add unit test, work on datacache
21406	DMZ\annrita.egli	Donnerstag, 31. Oktober 2013 18:46:01	add methods in DataCache
21405	DMZ\annrita.egli	Donnerstag, 31. Oktober 2013 17:01:58	rename typo
21403	DMZ\annrita.egli	Donnerstag, 31. Oktober 2013 15:52:06	add example from franziska
21398	DMZ\annrita.egli	Donnerstag, 31. Oktober 2013 10:54:42	increase methode getStructure
21392	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 21:38:19	Update of out of date project references
21391	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 21:34:47	Merge of Richards UI work into the main project
21390	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 21:29:01	Merge of Richards UI work into the main project
21389	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 19:26:00	
21388	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 19:17:31	delete project
21386	DMZ\annrita.egli	Mittwoch, 30. Oktober 2013 17:08:11	delete folder
21375	DMZ\annrita.egli	Dienstag, 29. Oktober 2013 16:33:29	work on getSceneName()
21365	DMZ\annrita.egli	Dienstag, 29. Oktober	refactor, upgrading GetDataAsync() method

		2013 08:37:38	
21362	DMZ\annrita.egli	Montag, 28. Oktober 2013 14:44:26	add demodata for getReachableScenes
21353	DMZ\annrita.egli	Sonntag, 27. Oktober 2013 09:07:53	add unittest
21342	DMZ\annrita.egli	Freitag, 25. Oktober 2013 11:07:18	Fallback, no changes
21333	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 23:33:54	Fix von inkorrektem Merge
21332	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 23:17:05	Fix von inkorrektem Merge
21331	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 22:56:54	Fix von inkorrektem Merge
21329	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 17:08:24	add unit tests
21326	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 16:13:42	remove debug
21325	DMZ\annrita.egli	Donnerstag, 24. Oktober 2013 16:12:12	add method getRooms, add Images
21303	DMZ\annrita.egli	Mittwoch, 23. Oktober 2013 08:23:26	DataCache namespace angepasst
21302	DMZ\annrita.egli	Dienstag, 22. Oktober 2013 22:20:33	callScene, add classes for data
21298	DMZ\annrita.egli	Dienstag, 22. Oktober 2013 16:13:28	add Methods: getApartmentName, getReachableScenes
21283	DMZ\annrita.egli	Montag, 21. Oktober 2013 20:40:46	bind image
21279	DMZ\annrita.egli	Montag, 21. Oktober 2013 16:11:08	GroupedItemPage added
21277	DMZ\annrita.egli	Montag, 21. Oktober 2013 14:20:30	renamed methods
21267	DMZ\annrita.egli	Sonntag, 20. Oktober 2013 23:10:40	datasource integration, refactor datasource viewmodel, refactor architecture
21266	DMZ\annrita.egli	Sonntag, 20. Oktober 2013 17:00:04	store real data
21265	DMZ\annrita.egli	Sonntag, 20. Oktober 2013 12:12:37	delete DataContext project
21264	DMZ\annrita.egli	Sonntag, 20. Oktober 2013 11:51:22	add core project
21259	DMZ\annrita.egli	Freitag, 18. Oktober 2013 13:43:12	added DataContracts
21257	DMZ\annrita.egli	Freitag, 18. Oktober 2013 12:24:00	test commit
21253	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 18:50:04	fix
21252	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 18:48:53	demo_en integrate
21250	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 18:32:44	add class for json
21243	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 15:20:49	fix
21236	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 12:01:31	SampleStructureDataSource-> SampleDataSource
21233	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 11:42:13	add callScene
21230	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 11:13:44	device id
21229	DMZ\annrita.egli	Donnerstag, 17. Oktober 2013 10:57:33	split json classes
21226	DMZ\annrita.egli	Mittwoch, 16. Oktober 2013 22:51:13	Trennen von view - viewmodel / Json Klassen
21221	DMZ\annrita.egli	Mittwoch, 16. Oktober 2013 14:56:17	login method ausgelagert
21187	DMZ\annrita.egli	Montag, 14. Oktober	Windows.Storage

		2013 22:33:52	
21182	DMZ\annrita.egli	Montag, 14. Oktober 2013 15:13:47	add new projects
21177	DMZ\annrita.egli	Freitag, 11. Oktober 2013 22:22:48	refactoring
21174	DMZ\annrita.egli	Freitag, 11. Oktober 2013 14:25:10	refactoring
21170	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 23:32:43	Login implementation
21169	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 17:49:48	Login in Bearbeitung
21165	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 16:22:34	Add DataContext
21164	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 16:22:19	added FrontEnd
21162	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 11:42:58	add name to textbox
21161	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 10:32:22	drag & drop enabled
21160	DMZ\annrita.egli	Donnerstag, 10. Oktober 2013 10:29:48	drag & drop enabled
21158	DMZ\annrita.egli	Mittwoch, 9. Oktober 2013 17:25:29	Titel hinzugefügt
21141	DMZ\annrita.egli	Donnerstag, 3. Oktober 2013 18:18:46	Login-GUI