

Intelligente Wohnungsprotokolle

Durch *ImmoP* effizient und individuell Wohnungs-
übernahmen durchführen!

Studienarbeit HS20

Autoren: Kevin Schweizerhof
Maximilian Marxer

E-Mail: kevin.schweizerhof@ost.ch
maxilian.marxer@ost.ch

Dozent: Prof. Dr. Daniel
Patrick Politze

Themengebiet: Software
Studiengang: Informatik

Erstellt am: 09. Dezember 2020

Letzte Änderung am: 18. Dezember 2020

Abstract

- Aufgabenstellung** Das Ziel dieser Arbeit war es, eine *Android Prototyp Applikation* für eine standardisierte Wohnungsübernahme zu entwickeln, welche die heutige Papierversion ablöst. Unsere Zielgruppen sind Immobilienverwaltungen und Vermieter, welche solche Wohnungsübernahmen durchführen. Es soll ein neuer Workflow konzipiert werden, mit welchem man möglichst effizient und individuell ein auf die jeweilige Wohnung angepasstes digitales Protokoll erstellen kann. Eine Bildaufnahme von allfälligen Schäden vor Ort für eine spätere Beweisführung soll ebenfalls gewährleistet werden.
- Methodik** Zuerst führte das Team eine Analyse über aktuelle *Augmented Reality*-Technologien durch. Durch diese Analyse sollte geprüft werden, ob eine geeignete *Augmented Reality*-Technologie für die Schadensaufnahme am Markt verfügbar ist. Dabei wurden verschiedene *Augmented Reality Frameworks* und *Software Development Kits* genauer angeschaut und deren Funktionen analysiert. Parallel dazu wurde eine Bedarfsermittlung anhand eines Leitfadeninterviews sowie eine Protokollanalyse zur Eruerung der Gemeinsamkeiten durchgeführt, um die Ausgangslage und die Anforderungen der Zielgruppen zu verstehen.
- Anhand der gesammelten Informationen wurde dann vom Team und dem Betreuer entschieden, dass eine Implementierung einer *Augmented Reality*-Funktion nicht Teil der Arbeit sein soll. Nach dieser Entscheidung wurde der Fokus auf eine intelligente Navigation gesetzt, um vorhandene Schäden zu protokollieren. Das Team erstellte ein Konzept für eine effiziente und flexible Navigation einer Schadensaufnahme bei einer Wohnungsübernahme. Nachfolgend wurde das erarbeitete Konzept in einer Android Applikation für Smartphones implementiert.
- Ergebnis** Das Team hat einen funktionsfähigen Prototypen entwickelt, mit welchem man eine effiziente Bestandesaufnahme einer Wohnung erstellen kann. Der Benutzer muss zuerst die Zielwohnung konfigurieren. Es besteht die Möglichkeit, die Wohnung Schritt für Schritt manuell zusammenzustellen oder diese automatisch konfigurieren zu lassen. Bei der automatischen Konfiguration werden vordefinierte Standardräume und Einrichtungsobjekte (Kühlschrank, Backofen etc.) verwendet. Sobald der Benutzer seine Wohnung erstellt und konfiguriert hat, kann er eine Bestandesaufnahme durchführen. Dabei wird er durch eine intelligente Navigation unterstützt, indem er durch alle Räume und deren Objekte durchgeführt wird. Bei Vollendung der Übernahme hat der Benutzer die Möglichkeit, aus den aufgenommenen Schäden ein Protokoll zu generieren. Das automatisch generierte Protokoll steht dem Benutzer als PDF-Datei zur Verfügung. Durch die Anwendung dieser Applikation kann der aktuelle und nicht mehr zeitgemässe Prozess einer Wohnungsübernahme abgelöst werden.

Management Summary

- Ausgangslage** Unsere Zielgruppe sind Immobilienverwaltungen und Vermieter, da Wohnungsübernahmen zu deren täglichen Aufgaben gehören.
- Bei Wohnungsübernahmen kommen nach wie vor mehrheitlich physische Formulare zur Anwendung. Oft haben Immobilienverwaltungen ihre eigenen Formulare. Schäden werden per Hand auf den Formularen aufgenommen und dokumentiert. In seltenen Fällen werden per Handy oder Fotoapparat Bilder für eine Schadensaufnahme gemacht. Die von Hand ausgefüllten Wohnungsübernahme-Protokolle müssen anschliessend von den beteiligten Personen unterzeichnet werden.
- Die erstellten Protokolle werden in der Regel physisch aufbewahrt. Eher selten werden sie von der Verwaltung eingescannt und digital archiviert.
- Methodik** Das Team führte zu Beginn eine Analyse der aktuellen *Augmented Reality*-Technologien durch. Der Fokus dieser Analyse lag bei den einzelnen Funktionen der verschiedenen *Augmented Reality Frameworks* und *Software Development Kits*. Dabei wurde geprüft, ob die einzelnen Funktionen für eine solche Applikation geeignet sind und sie den Anforderungen der Zielgruppe entsprechen. Ein Leitfadenterview mit einem Experten sowie die Analyse verschiedener Übernahmeprotokolle half uns bei der Bedarfsanalyse der Zielgruppe.
- Anhand der gesammelten Informationen wurde dann vom Team und dem Betreuer entschieden, dass eine Implementierung einer *Augmented Reality*-Funktion nicht Teil der Arbeit sein sollte. Der Fokus wurde auf eine Android-Applikation für Smartphones gelegt. Die Kamerafunktion der Smartphones soll entsprechend genutzt werden. Anhand der gesammelten Anforderungen der Zielgruppe wurde ein Konzept einer effizienten und flexiblen Navigation durch ein Wohnungsobjekt mit integrierter Verwaltung der digital erhobenen Daten erstellt. In der darauffolgenden Entwicklungsphase wurde das Konzept implementiert.

Ergebnis

Die fertige Prototyp-Applikation befindet sich auf den Gitlab Repository der OST [1].



Abbildung 1 Startansicht



Abbildung 2 Schadensaufnahme

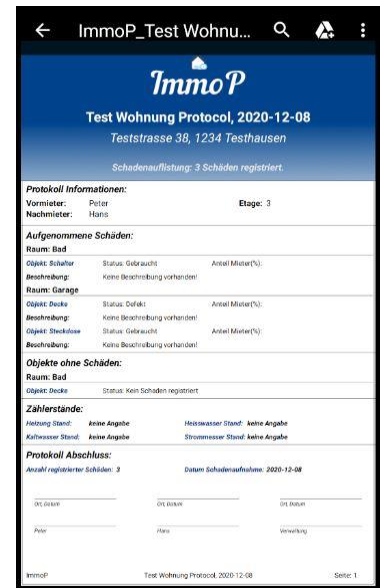


Abbildung 3 Protokoll

Die gewünschte Applikation konnte erfolgreich entwickelt und auf einem Smartphone bereitgestellt werden. Für die Nutzung dieser Android-App wird ein Google Account verlangt. Der User muss sich beim Starten der Applikation mit seinem Google Account einloggen. Nach erfolgreichem Einloggen können Wohnungen durch wenige Klicks individuell erstellt werden. Bei einer Schadensaufnahme werden die jeweiligen Räume und deren Objekte in der konfigurierten Wohnung in die Navigation einfließen und den User durch die Wohnung navigieren. Der User hat jeweils die Möglichkeit, selbst zu bestimmen, in welchem Raum er eine Schadensaufnahme machen möchte. Durch unsere intelligente Navigation muss der User alle hinterlegten Räume und Objekte abarbeiten. Dadurch soll kein Raum oder Objekt vergessen gehen. Die Schadensaufnahme kann erst abgeschlossen werden, wenn durch die konfigurierte Wohnung durchnavigiert wurde. Die aufgenommenen Daten sowie gemachte Bilder werden in der *Google Firebase*-Datenbank gespeichert.

Ein Umstieg auf diese Applikation und die dadurch abgelösten physischen Protokolle ermöglichen es einer Immobilienverwaltung, eine effizientere Wohnungsübernahme durchzuführen. Durch die individuelle Navigation, welche auf jede einzelne Wohnung abgestimmt ist, können die standardisierten Formulare der Immobilienverwaltungen abgelöst werden. Unnötig aufgeführte Objekte oder zu umfangreiche Formulare entfallen mit diesem eigens entwickelten Prototypen.

Die durchgeführten Testläufe mit dieser Anwendung zeigen, dass in gewissen Bereichen der Navigation der Applikation selbst noch Verbesserungspotenzial besteht. Die Anforderungen der intelligenten Navigation für eine Schadensaufnahme der Wohnung konnten gut erfüllt und umgesetzt werden.

Inhalt

Abstract	2
Management Summary	3
1 Technischer Bericht	7
1.1 Ausgangslage & Problembeschreibung	7
1.2 Vision.....	7
1.3 Zielsetzung.....	7
1.4 Abgrenzung.....	8
1.5 Methodik	8
1.6 Vorarbeit	9
1.6.1 Machbarkeitsanalyse Augmented Reality.....	9
1.6.2 Protokollanalyse	12
1.6.3 Interview.....	17
1.7 Lösungskonzept	19
1.7.1 Produkt Perspektive.....	19
1.7.2 Benutzer Charakteristik.....	19
1.7.3 Einschränkungen.....	19
1.7.4 Funktionalitäten	19
1.8 Entscheidungen.....	27
2 Software Engineering	28
2.1 Anforderungsspezifikation.....	28
2.1.1 Funktionale Anforderungen	28
2.1.2 Nichtfunktionale Anforderungen.....	32
2.2 Architektur.....	34
2.2.1 Domainanalyse.....	34
2.2.2 Systemübersicht	35
2.2.3 Logische Architektur	36
2.2.4 Package-Struktur	37
2.2.5 Technologien.....	37
3 Testing	38
4 Umsetzung	40
4.1 Implementierung.....	40
4.1.1 Presentation Layer.....	40
4.1.2 Businesslogik	41
4.2 Fazit Implementierung	46
5 Ergebnisse	47
5.1 Applikation	47
5.2 Review funktionale Anforderungen	55
6 Schlussfolgerung	58
6.1 Ausblick	58
6.2 Reflexion.....	58
7 Literaturverzeichnis	59

8	Verzeichnisse	61
8.1	Abkürzungsverzeichnis (Glossar).....	61
8.2	Abbildungen.....	62
8.3	Tabellen.....	63
9	Erklärung zur Urheberschaft	64
Anhang	65
I	Benutzerhandbuch.....	65
II	Projektplan.....	65
III	ImmoP Protokoll Beispiel	65
IV	Protokoll Wuhrmann Immobilien.....	65
V	Leitfadeninterview.....	65
VI	Testauswertung Protokoll.....	65
VII	Zeitauswertung.....	66

1 Technischer Bericht

Inhalt Im folgenden Kapitel werden die Ausgangslage und die Problemstellung beschrieben. Zudem werden Zielsetzung sowie Abgrenzung definiert und es wird klar dargelegt, wie in dieser Semesterarbeit vorgegangen wurde.
Im zweiten Teil wird das Lösungskonzept vorgestellt.

1.1 Ausgangslage & Problembeschreibung

Einleitung Die meisten Wohnungsabnahmeprotokolle werden auch heute noch physisch auf Papier erstellt. Bei der Begutachtung der Wohnung werden Schäden im Protokoll dokumentiert und oftmals auch mit dem Smartphone fotografiert.
Viele Vermieter verwenden standardisierte Protokolle. Diese enthalten meist auch Räume und Einrichtungen, welche in der spezifischen Wohnung nicht vorhanden sind.
Auch die Aufbewahrung der Protokolle erfolgt meist in physischer Form. Das Foto bleibt nur auf dem eigenen Mobilgerät gespeichert. Dieser Prozess ist sehr umständlich und nicht mehr zeitgemäss.

1.2 Vision

Vision Die Vision dieser Semesterarbeit ist es, den mühsamen Prozess der Wohnungsübernahme zu digitalisieren. Die Mobile-App «ImmoP» bietet Vermietern und Mietern die Möglichkeit, individuelle Protokolle schnell und einfach zu erstellen.
In Zukunft soll die Mobile-App im Play Store von Google oder im direkten Verkauf vertrieben werden. Bei einem guten Marktstart könnte man die App auch noch für iOS-Geräte entwickeln.

1.3 Zielsetzung

Android-App mit Firebase Cloud Das Ziel dieser Semesterarbeit ist es, einen funktionsfähigen Android-Prototypen für die Schadensaufnahme einer Wohnungsübernahme zu entwickeln, um das physische Protokoll abzulösen. Die erstellten Protokolle sollen automatisch in der *Firebase Cloud* gespeichert werden. Das Protokoll soll individualisiert werden können, indem Räume und Objekte für eine Wohnung hinzugefügt oder entfernt werden. Bei der Erstellung eines neuen Protokolls soll der Benutzer durch die verschiedenen Räume und Einrichtungsobjekte navigiert werden. Bei der Protokollierung eines neuen Schadens sollen bereits erfasste Schäden für das jeweilige Einrichtungsobjekt angezeigt werden. Dadurch kann der Benutzer bei der Begutachtung der Schäden unterstützt werden. Bei der Vollendung einer Schadensaufnahme in einer Wohnung soll ein Wohnungsübernahme-Protokoll in Form einer PDF-Datei [2] erstellt und exportiert werden können. Im Rahmen der Arbeit werden mehrere Protokolle analysiert und ausgewertet. Die daraus gezogenen Erkenntnisse sollen aufzeigen, welche Punkte in einem Protokoll am wichtigsten sind.

1.4 Abgrenzung

- Technologie** Die Applikation wird in der Programmiersprache Java und für die Android-Version 10 (API Level 29) [3] entwickelt. Für das Persistieren der Daten wird eine Firebase [4] Cloud Lösung genutzt.
- Applikation** Der Prototyp kann manuell auf einem Smartphone, welches die Android-Version 10 unterstützt, deployt und ausgeführt werden. Alle Funktionalitäten sind nur verfügbar, wenn das Gerät mit dem Internet verbunden ist. Eine Offline-Version wird in dieser Arbeit nicht konzipiert.
- Für die Schadensaufnahme wurden die wichtigsten Elemente aus der Vorarbeit vom Team implementiert.

1.5 Methodik

- Startphase** Der Projektplan wurde in Anlehnung an die *Rational Unified Process*-Methode [5] entwickelt.
- Da beide Teammitglieder noch keine Erfahrungen mit *Augmented Reality*-Technologien hatten, stand an erster Stelle eine Analysephase. Mit der Analyse sollte eruiert werden, inwiefern sich *Augmented Reality*-Funktionen in die Android-App einbinden lassen. Dazu wurden verschiedene Frameworks ausgewählt und zwei davon vertiefter untersucht.
- Um in der Entwicklungsphase möglichst nahe an der Realität zu bleiben, wurde ein Leitfadeninterview mit dem Geschäftsleiter der Wuhrmann Immobilien & Verwaltung GmbH [6] durchgeführt. Neben der Analyse der *Augmented Reality*-Technologien wurde eine Analyse der verschiedenen Wohnungsprotokolle durchgeführt. Das Ziel dieser Protokollanalyse war es, Gemeinsamkeiten der Protokollinhalte für die spätere Konzeptphase zu finden.
- Eine vertiefte Analyse führte zum Entschluss, dass sich die Implementierung einer *Augmented Reality*-Funktionalität als nicht opportun erweist. Der Fokus fiel somit auf eine Android-Applikation für Smartphones.
- Konzeptphase** In einem weiteren Schritt wurde ein Konzept für die Entwicklung einer Android-Applikation für ein Smartphone evaluiert. Dabei standen die gesammelten Anforderungen aus dem Leitfadeninterview und der Protokollanalyse im Fokus. Für das *User Interface* der Applikation wurden Entwürfe zusammengestellt, welche mit dem Betreuer diskutiert und im Anschluss verbessert und erweitert wurden.
- Für die Schadensaufnahme durch die Applikation wurde das Konzept für eine intelligente Navigation durch die Wohnung entworfen. Dabei wurden wichtige Protokollanforderungen aus dem Leitfadeninterview und Ergebnisse aus der Protokollanalyse miteinbezogen.
- Entwicklungsphase** Nachdem das Konzept fertig ausgearbeitet war, begann das Team mit der Entwicklung der Applikation. Da beide Teammitglieder nur geringe Erfahrungen im Bereich der Android-Programmierung mitbrachten, musste dieses Wissen zuerst vertieft werden. Anschliessend erfolgte die Programmierung der Applikation. Der Leitfaden der Entwicklung war das erarbeitete Konzept mit den *Use Cases* und *Requirements*.

Während der Entwicklung wurden für die einzelnen Funktionen laufend Reviews und manuelle Tests durchgeführt. Die Reviews erfolgten über ein gemeinsames Teammeeting. Für die Tests wurde die Applikation entweder auf ein Smartphone oder einen Emulator installiert.

Abschlussphase In der Abschlussphase wurde das Ergebnis nochmals analysiert. Dabei wurden Verbesserungsvorschläge und Ausbaumöglichkeiten ermittelt.
Im letzten Teil dieser Studienarbeit wurde die Dokumentation erfasst.

1.6 Vorarbeit

Inhalt In den folgenden Kapitel werden die Resultate der Vorarbeiten aufgeführt und beschrieben.

1.6.1 Machbarkeitsanalyse Augmented Reality

Einleitung Dieses Kapitel beschreibt die Machbarkeitsanalyse für *Augmented Reality* (AR). In der Analyse werden zwei *Software Development Kits* (SDK) für Augmented Reality genauer untersucht.

Das Ziel der Machbarkeitsanalyse ist es, sich in die *Augmented Reality*-Technologien einzuarbeiten und Erkenntnisse der verschiedenen AR-Technologien zu erhalten. Mit dieser Analyse soll eruiert werden, ob und welche AR-Technologien für die Umsetzung unseres Vorhabens geeignet sind.

Augmented Reality Unter «Augmented Reality», im Deutschen «erweiterte Realität», versteht man den Zusammenschluss der digitalen und der analogen Welt. Er funktioniert entweder über die Kamera eines Mobilgeräts oder mit einer spezialisierten AR-Brille. Im Gegensatz zu einer VR-Brille wird der Benutzer nicht komplett von seiner Umgebung abgeschottet. Stattdessen werden zusätzliche Informationen über sein Umfeld angezeigt.

ARCore ARCore [7] ist ein *Software Development Kit* (SDK) zur Erstellung von *Augmented Reality*-Anwendungen, welches von Google entwickelt wurde. Es ist kostenlos und unterstützt die Entwicklung für Android (7.0 und höher) und iOS-Plattformen (11 und höher). Das SDK wird von den nachfolgenden Plattformen unterstützt:



Abbildung 4 Plattformen für ARCore

ARCore verwendet drei Schlüsseltechnologien, um virtuelle Inhalte in die reale Umgebung zu integrieren:

- **Motion Tracking:** Mit den Bewegungsverfolgungsfunktionen können Entwickler die Position des Telefons relativ zur Umgebung verfolgen.

- **Environmental Understanding:** Dies ermöglicht dem Mobilgerät, die Grösse und Lage von vielen Oberflächen zu erkennen. Flache Oberflächen ohne Struktur, z.B. eine weisse Wand, werden nicht erkannt.
- **Light Estimation:** Das Mobilgerät schätzt die aktuellen Lichtverhältnisse einer Umgebung ab.

Wikitude

Wikitude ist ein *Augmented Reality*-SDK für Mobilgeräte. Sie gehört zu den eher bekannten SDKs, wenn AR zum Einsatz kommt. Sie ist weit verbreitet und kann für verschiedene Plattformen genutzt werden.

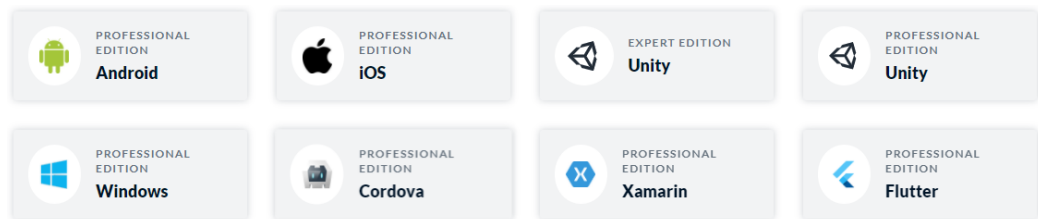


Abbildung 5 Plattformen für Wikitude SDK

Wikitude verfügt über ein schnelles *Tracking Feature*. Das *SLAM markerless AR*-Trackingsystem gehört zu den beliebtesten und anpassungsfähigsten plattformübergreifenden 3D-Trackingsystemen, welche für mobile Plattformen verfügbar sind [8]. Wikitude verfügt über eine Instant-Trackingfunktion, welche eine digitale Nachbildung der Realität erzeugen kann, ohne dass der Benutzer ein Bild scannen muss. Durch die breite Palette an Funktionen kann die Wikitude in verschiedenen Bereichen eingesetzt werden.

1.6.1.1 Resultat Vorstudie

Erkenntnisse

Die beiden untersuchten *Software Development Kits* bieten eine Vielzahl von Funktionalitäten. Für unsere Aufgabenstellung wurden nur diejenigen Funktionen ausgewählt, welche uns bei der Problemlösung unterstützen.

Es wurde festgelegt, dass ein zuvor angelegtes 3D-Modell einer Wohnung keine Option ist. Dies schränkt die Auswahl der möglichen AR-Funktionalitäten für unser Vorhaben wesentlich ein.

Die Analyse von ARCore und Wikitude hat gezeigt, dass es grundsätzlich zwei AR-Funktionen gibt, welche für die angestrebte Mobile-App nützlich sind. Wir beschreiben im Nachfolgenden diese Funktionalitäten und gehen auf ihre Einschränkungen ein.

Image Recognition

Der Benutzer der App fotografiert bei der Besichtigung der Wohnung alle Schäden und lädt diese in eine Cloud-Datenbank hoch. Bei der nächsten Besichtigung wird mit *Image Recognition* geprüft, ob ein vorhandener Schaden bereits fotografiert wurde. Wenn in der Bilderdatenbank ein Bild gefunden wurde, wird der Benutzer über den bereits vorhandenen Schaden informiert.

Einschränkungen:

Die *Image Recognition* ist sowohl in ARCore als auch in Wikitude auf 2D-Objekte beschränkt. Beim Fotografieren eines Objektes muss dieses als 2D-Bild in der Datenbank abgespeichert werden. Bei der Aufnahme eines 2D-Bildes spielt der Blickwinkel der Kamera eine essenzielle Rolle. Wenn bei einem weiteren Durchgang das Objekt von einem anderen Blickwinkel aufgenommen wird, kann es durch *Image Recognition* nicht erkannt werden.

Bei der *Image Recognition* besteht noch Verbesserungspotenzial. Die Farbinformationen von Bildern werden nicht genutzt. Bilder mit wenig Unterscheidungsmerkmalen oder mit sich wiederholenden Strukturen werden nicht erkannt. Dies bedeutet, dass beispielsweise Flecken auf einer weissen Wand für die *Image Recognition* keine Unterscheidungsmerkmale sind.

Bei ARCore können ausserdem nur maximal 1000 Bilder in eine *Augmented Image*-Datenbank geladen werden.

Folgerung:

Aufgrund der oben erwähnten Einschränkungen eignet sich die *Image Recognition*-Funktion der AR-Technologien leider nicht für die Wiedererkennung eines Schadens. Ein schwarzer Strich an einer Wand weist beispielsweise zu wenig Unterscheidungsmerkmale auf. Ein Tutorial hat gezeigt, dass Kratzer im Parkett aufgrund ihrer sich wiederholenden Strukturen im Boden nicht erkannt werden. Wenn sich ein Schaden im Laufe der Zeit verändert, sich beispielsweise ein Riss in der Scheibe vergrössert, wird dieser nicht mehr als registrierter Schaden erkannt. *Image Recognition* kann unsere Anforderungen nicht erfüllen. Sie hat viel Potenzial, ist aber noch zu wenig ausgereift.

Geo AR

Mittels der Geo AR-Funktionalität ist es möglich, virtuelle Notizen für vorhandene Schäden in einer Wohnung zu platzieren. Die Position und Ausrichtung der Notizen werden mit sogenannten Anchor Clouds [9] in der Cloud gespeichert. Gespeicherte Notizen werden beim Betreten eines Raumes automatisch geladen. Anhand der gespeicherten Informationen werden die Notes dann wieder an der richtigen Stelle in der Wohnung platziert und sind für den Benutzer einsehbar.

Mit dieser Funktionalität können bereits registrierte Schäden einfach erkannt werden. Bei neuen Schäden werden neue Markierungspunkte gesetzt.

Einschränkung:

Die grösste Einschränkung ist hier, dass Ebenen ohne Struktur, beispielsweise eine weisse Wand, nicht als Oberflächen erkannt werden [10]. Eine Objektplatzierung an der weissen Wand funktioniert deshalb nicht.

Bei einem durchgeführten Tutorial wurde eine weitere Einschränkung festgelegt. Die Erkennung von Ebenen und Markierungspunkten bei kleinen Räumen liefert keine guten Ergebnisse. Es wird ein gewisser Abstand benötigt, damit eine Ebene erkannt wird. In engen Gängen und Räumen ist somit eine präzise Platzierung von virtuellen Notizen nicht möglich. Aus dem Tutorial konnten wir auch die Erkenntnis gewinnen, dass die gesetzten Notizen nicht immer erfolgreich von der Cloud geladen werden. Die Kamera des

Mobilgeräts muss sich beim Laden der Objekte an der richtigen Stelle im Raum befinden. Befindet sich das Gerät zu weit entfernt vom Zielobjekt, können Notizen nicht geladen werden.

Ein weiteres Problem hat sich beim Laden der Notizen gezeigt. Diese wurden aufgrund von fehlenden Positions- und Ausrichtungspunkten im Raum und beim Objekt nicht immer korrekt platziert,.

Folgerung:

Die Geo AR-Funktionalität ist für unser Vorhaben besser geeignet als die *Image Recognition*-Funktion. Leider hat die Analyse aber auch hier viele Einschränkungen und Probleme aufgezeigt. Beim Wikitude SDK ist es nicht möglich, die Punkte exakt zu platzieren, da die Abstände zu klein sind. Man muss genaue Punkte definieren, damit die Daten beim Laden wieder korrekt platziert werden. Bei ARCore ist es nicht möglich, Objekte auf weisse Wände zu platzieren. Weiter werden Ebenen in kleinen Räumen und Gängen nicht korrekt erkannt. Beim Selbstversuch wurden ausserdem die Positionen der Objekte oft nicht präzise genug geladen, was zu Verschiebungen der *Notes* im Raum geführt hat.

1.6.1.2 Schlussfolgerung Machbarkeitsanalyse Augmented Reality

Fazit

Aus den gewonnenen Erkenntnissen geht hervor, dass die beiden AR-Funktionalitäten für unseren Anwendungsfall zwar sehr interessant sind, jedoch haben die Tutorials gezeigt, dass es viele Einschränkungen und Fehlerquellen gibt. Für eine Wohnungsübernahme werden genaue und klare Daten benötigt. Die Geo AR-Funktionalität von ARCore wäre geeignet, wenn man Objekte auf strukturlosen Ebenen platzieren könnte und die Erkennung von Ebenen auch in kleinen Räumen oder Gängen funktionieren würde.

Es wurde deshalb mit dem Betreuer zusammen entschieden, keine AR-Funktionen in die Android-App einzubauen. Stattdessen wurde ein neues Lösungskonzept mit einer smarten Navigation erarbeitet. Die smarte Navigation soll den Benutzer durch die Wohnung führen und ihn so bei der Wohnungsabnahme unterstützen.

1.6.2 Protokollanalyse

Einleitung

In diesem Abschnitt wird die Protokollanalyse beschrieben. Wir haben im Rahmen der Arbeit mehrere Wohnungsprotokolle untersucht und deren Inhalte analysiert. Das Ziel der Protokollanalyse war es, die vielen unterschiedlichen Informationen der verschiedenen Protokolle zu evaluieren und auf ein Minimum zu reduzieren.

Fragestellungen

Die nachfolgenden konkreten Fragen wurden zu Beginn der Analyse festgelegt. Sie sollten durch die Analyse beantwortet werden.

1. Welche Informationen sind in jedem Abnahmeprotokoll vorhanden?
2. Wie wird ein Schaden protokolliert?
3. Welche Räume sind in den meisten Protokollen vorhanden?
4. Welche Einrichtungsobjekte kommen meistens vor?

Protokolle

In der nachfolgenden Tabelle sind alle verwendeten Protokolle aufgelistet.

Name	Quelle
ImmobilienScout24.de	https://www.immobilienscout24.de/content/dam/is24/umzug/dokumente/wohnunguebergabeprotokoll.pdf
Mieterverband Schweiz	https://www.mieterverband.ch/mv/mietrechtberatung/ratgeber-mietrecht/top-themen/wohnungsabgabe-protokoll.html
Homegate.ch	https://www.homegate.ch/neutral/img/cms/Wohnungs%C3%BCbergabeprotokoll-Muster.pdf
Wohnungsboerse.net	https://www.wohnungsboerse.net/wohnungsuebergabeprotokoll
Conviva-plus.ch	https://www.conviva-plus.ch/?page=141
Immowelt.de	https://content.cdn.immowelt.de/ratgeber_iw_de/images/Ratgeber/PDF/muster_wohnungsuebergabeprotokoll_immowelt.pdf?v=1594967997
Mietrecht.ch	https://www.mietrecht.ch/fileadmin/files/Formulare/form_w_abnahmeprotokoll_2spaltig.pdf
Deutscher Mieterbund	https://www.mieterbund.de/index.php?eID=tx_naw-securedl&u=0&g=0&t=1607803301&hash=84f8b48625bd665f02bfe3d588e80dfe7363&file=fileadmin/pdf/mietvertrag/Wohnungsuebergabe-Protokoll_neu.pdf
Wuhrmann Immobilien	Exemplar im Anhang

Tabelle 1 Basis Protokollanalyse

Fragestellung 1
Welche Informationen sind in jedem Abnahmeprotokoll vorhanden?

Jedes Protokoll besitzt eine Mängelliste und enthält Informationen zur Wohnung und den Mietparteien. Wir beschränken uns hier auf die Informationen zu den Mietparteien und der Wohnung, da wir die Mängelliste bei den anderen Fragestellungen analysieren. Die nachfolgende Grafik zeigt auf, welche Input-Felder in den einzelnen Protokollen vorkommen.

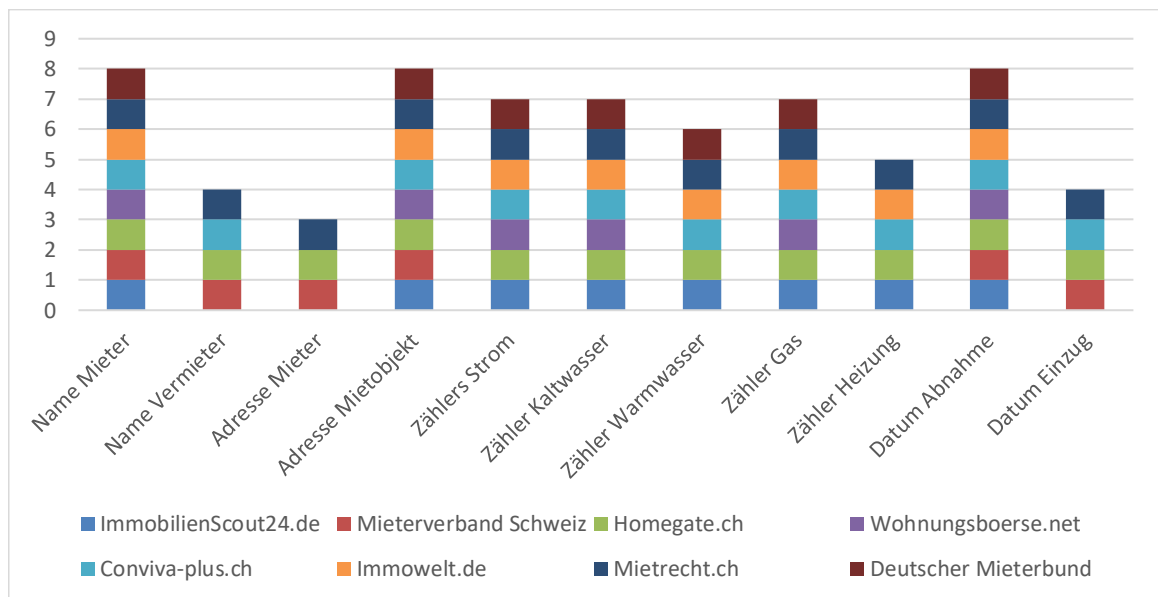


Abbildung 6 Auswertung Felder in Protokollen

- Der Name des Mieters, das Datum der Abnahme und die Adresse des Mietobjekts sind in jedem einzelnen Protokoll vorhanden.
- Die Zählerstände für Strom, Kaltwasser und Gas kommen in sieben von acht Protokollen vor.
- Der Zählerstand Warmwasser kommt in sechs von acht Protokollen vor.
- Der Zählerstand Heizung kommt in fünf von acht Protokollen vor.

Die aufgenommenen Daten werden von allen Parteien durch eine Unterschrift bestätigt.

Ort/Datum:	VermieterIn, VerwalterIn:	Mieter/Mieterin:	Experte/Expertin:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Abbildung 7: Unterschrift Protokoll Mieterverband Schweiz

Konsequenzen:

Informationen, welche in mindestens vier Protokollen vorkommen, sollen in das Lösungskonzept aufgenommen werden. Für die «ImmoP»-PDF-Datei soll am Ende ein Unterschriftenblock eingefügt werden.

Fragestellung 2 Wie wird ein Schaden protokolliert?

Die Hälfte der analysierten Protokolle enthält eine sehr triviale Mängelliste von nachfolgender Struktur:

Bei der Wohnungsbesichtigung am _____ wurden keine folgende Mängel festgestellt:

	In Ordnung	Mängel	Bemerkungen
Diele/Flur	ja		
Küche	ja		
Bad/WC	ja		

Abbildung 8: Ausschnitt Abnahmeprotokoll ImmoScout24.de

Es werden einige vordefinierten Räume aufgelistet, für welche ein Schaden protokolliert werden kann. Wenn in einem Raum kein Schaden vorhanden ist, wird dies im Protokoll explizit angegeben. Dabei werden Textfelder und Checkboxen verwendet. Schäden können in Textfeldern beschrieben werden.

Die andere Hälfte der Protokolle enthält eine wesentlich detailliertere Schadensbeschreibung in nachfolgender oder ähnlicher Struktur:

Küche Beim Einzug frisch gestrichen: Wände Holzwerk Bodenbelag: neu

<input type="checkbox"/> 1. Boden <input type="checkbox"/> 2. Wände <input type="checkbox"/> 3. Decke <input type="checkbox"/> 4. Schränke oben <input type="checkbox"/> 5. Schränke unten <input type="checkbox"/> 6. Plättli <input type="checkbox"/> 7. Türen	<input type="checkbox"/> 8. Schloss/Schlüssel <input type="checkbox"/> 9. Fenster DV/IV <input type="checkbox"/> 10. Rollläden <input type="checkbox"/> 11. Gurten/Kurbeln <input type="checkbox"/> 12. Vorhangbrett/-schienen <input type="checkbox"/> 13. Heizkörper/-ventil <input type="checkbox"/> 14. Backofen	<input type="checkbox"/> Backofenzubehör: <input type="checkbox"/> 15. Blech <input type="checkbox"/> 16. Grill <input type="checkbox"/> 17. Rost <input type="checkbox"/> 18. Herd <input type="checkbox"/> 19. Dunstabzugshaube <input type="checkbox"/> 20. Schüttstein/Chromstahl	<input type="checkbox"/> 21. Batterie <input type="checkbox"/> 22. Kühlschrank/Tiefkühler <input type="checkbox"/> 23. Elektr./Schalter/Stecker <input type="checkbox"/> 24. Geschirrspüler <input type="checkbox"/> 25. <input type="checkbox"/> 26. <input type="checkbox"/> 27. Schlüssel (Anzahl)
--	--	---	---

Bad/Dusche/WC Beim Einzug frisch gestrichen: Wände Holzwerk Bodenbelag: neu

<input type="checkbox"/> 28. Boden <input type="checkbox"/> 29. Plättli <input type="checkbox"/> 30. Wände <input type="checkbox"/> 31. Decke	<input type="checkbox"/> 36. Rollläden <input type="checkbox"/> 37. Gurten/Kurbeln <input type="checkbox"/> 38. Elektr./Schalter/Stecker <input type="checkbox"/> 39. Wanne/Dusche	<input type="checkbox"/> 44. Klosett: <input type="checkbox"/> Spülkasten <input type="checkbox"/> WC-Brille <input type="checkbox"/> 45. Papierhalter	<input type="checkbox"/> 50. Tablare <input type="checkbox"/> 51. Wandschränke <input type="checkbox"/> 52. Heizkörper/-ventil <input type="checkbox"/> 53.
--	---	---	--

Abbildung 9: Ausschnitt Abnahmeprotokoll Mieterverband Schweiz

Für jeden einzelnen Raum sind vordefinierte Einrichtungsobjekte festgelegt. Der Benutzer geht jedes dieser Objekte durch und notiert, ob ein Schaden vorhanden ist. Für einige wenige Objekte gibt es die Möglichkeit, einen Status zu setzen. So kann beispielsweise für Wände angegeben werden, ob diese frisch gestrichen sind.

Die Analyse hat gezeigt, dass ein Zustand auf verschiedene Arten beschrieben werden kann. Im Protokoll von mietrecht.ch werden Zustände wie folgt festgehalten:

Pos.		i. O.	norm. Abnütz.	KA %/Fr.
Küche				
1.	Boden	<input type="checkbox"/>	<input type="checkbox"/>	
2.	Wände	<input type="checkbox"/>	<input type="checkbox"/>	
3.	Decke	<input type="checkbox"/>	<input type="checkbox"/>	
4.	Schränke	<input type="checkbox"/>	<input type="checkbox"/>	

Abbildung 10:Ausschnitt Protokoll mietrecht.ch

Es gibt zwei verschiedene Zustände, und zwar «in Ordnung» und «normale Abnutzung». Dabei kann der Kostenanteil des Mieters in % angegeben werden.

Konsequenzen:

Die vordefinierten Zustände der Protokolle unterscheiden sich, beschreiben aber schlussendlich immer dasselbe. Es wird entschieden, ob ein Objekt neuwertig, gebraucht oder beschädigt ist. Die Angabe «normale Abnutzung» entspricht dem Zustand «gebraucht». Darunter fallen normale Gebrauchsspuren, für welche der Mieter nicht aufkommen muss.

Fragestellung 3 Welche Räume sind in den meisten Protokollen vorhanden?

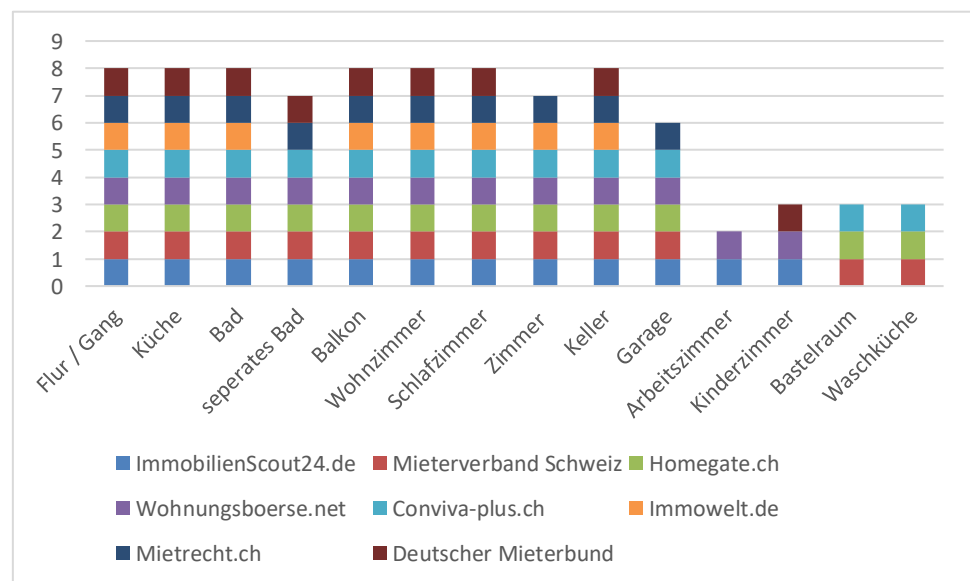


Abbildung 11 Vorhandene Räume in Protokollen

Die Räume Flur/Gang, Küche, Bad, Balkon, Wohnzimmer, Schlafzimmer und Keller sind in allen Protokollen vorhanden. Individuelle Zimmer sowie ein zweites Badezimmer sind bei sieben Protokollen vorhanden.

Konsequenzen:

Alle Räume, welche in mindestens drei Protokollen vorkommen, sollen in der App bereits vordefiniert werden. Damit werden die meisten Räume, welche ein Benutzer benötigt, vorhanden sein.

Fragestellung 4 Welche Einrichtungsobjekte kommen meistens vor?

Die vordefinierten Listen der analysierten Protokolle sind fast identisch. Da sich die Listen nur minimal unterscheiden, haben wir uns hier auf die Liste vom Schweizer Mietverband fokussiert. Die Liste ist sehr gross und umfasst teilweise über 200 Einrichtungsobjekte. Dabei gibt es bestimmte Einrichtungsobjekte, welche für jeden Raum aufgelistet werden.

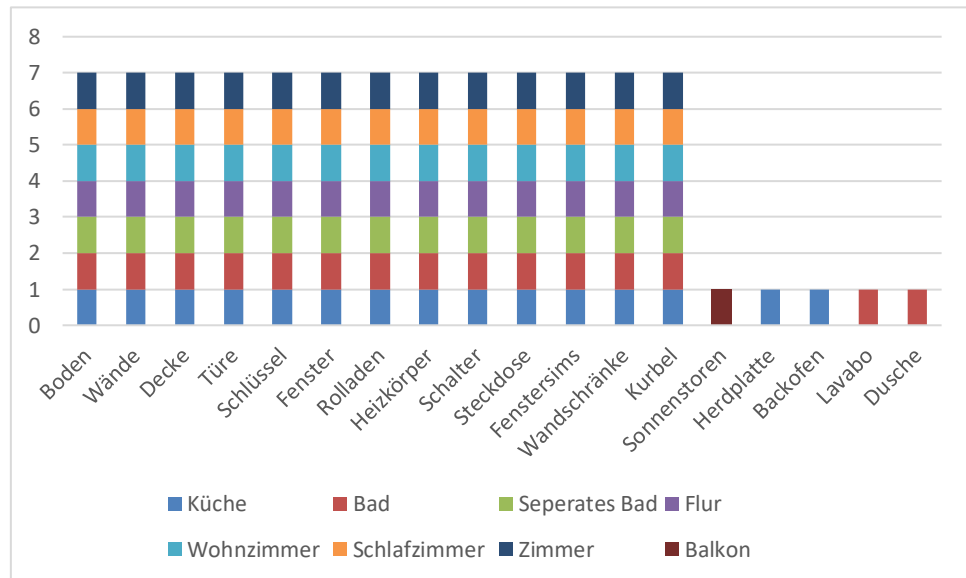


Abbildung 12 Vorhandene Einrichtungsobjekte in Protokollen

Die obige Grafik zeigt auf, welche Objekte in jedem Raum vorhanden sind. Individuelle Einrichtungsobjekte wie ein Backofen sind nur einzeln vorhanden.

Konsequenzen:

Die Objekte, welche in jedem Raum vorhanden sind, sollen in der App zur Auswahl stehen.

1.6.3 Interview

Vorhaben / Ziel Damit der Prototyp schlussendlich möglichst benutzerfreundlich ist und die Funktionalitäten dem User einen hohen Nutzen versprechen, haben wir ein Leitfadeninterview mit einer Immobilienverwaltung durchgeführt. Das Ziel des Interviews war die Eruierung des Interesses sowie der funktionalen Anforderungen aus Sicht des Nutzers.

Das Interview wurde mit Herrn Bruno Wuhrmann durchgeführt. Er ist seit 1995 in der Immobilienbranche tätig. Seine langjährige Erfahrung in diesem Bereich gab uns einen wertvollen Einblick in die Branche.

Es ist zu erwähnen, dass sich ein Teil des Interviews auf Fragen zu *Augmented Reality*-Funktionalitäten bezogen hat. Da schlussendlich keine solchen Funktionen implementiert wurden, hatten die diesbezüglichen Anforderungen keinen direkten Einfluss auf die darauffolgende Konzeptphase. Das Team hat versucht, die wichtigsten Anforderungen daraus für eine Android-Applikation zu adaptieren.

Kernpunkte

Im Interview wurden folgende Punkte diskutiert:

Organisation und Administration:

- Personenbeteiligung bei einer Wohnungsübernahme
- Zeitaufwand einer Wohnungsübernahme
- Verwendung von Geräten
- Probleme bei einer Wohnungsübernahme

Schadenaufnahme:

- Form der Schadenaufnahme
- Beschreibung eines Schadens
- neuralgische Punkte
- Aufbewahrung der gesammelten Informationen

Applikation:

- Prozess Schadenaufnahme
- essenzielle Funktionen einer Applikation
- Anwendungsgerät

Abschlussfragen:

- Interesse an einer solchen Applikation
- Bemerkungen des Interviewpartners

Auswertung

Das Interview hat dem Team klar gezeigt, dass die aktuelle Methode einer Schadenaufnahme veraltet ist. Die Protokolle werden für eine Wohnungsübernahme in Papierform genutzt und auch aufbewahrt. Schäden werden in der Regel von Hand auf dem Protokoll notiert. Wenn nötig, wird ein Foto des Schadens mittels Handys oder Fotoapparats erstellt.

Für den Interviewpartner wäre bei einer solchen Applikation eine Kamerafunktion, mit welcher man die Schäden direkt aufnehmen und in das Protokoll integrieren könnte, am wichtigsten. Zusätzlich müsste es die Möglichkeit geben, den Schaden zu dokumentieren. Das Endprodukt sollte ein Protokoll sein, welches umgehend und automatisch generiert werden kann.

Die Kernaussage in diesem Interview war für das Team, dass nicht zwingend der Prozess der Schadenaufnahme verbessert werden muss, sondern das Aufbewahren und Verwalten des physischen Protokolls. Vom Interviewpartner wurde aber auch klar vermittelt, dass eine neue Art der Schadenaufnahme wünschenswert sei, wenn dies durch ein effizientes und benutzerfreundliches Verfahren umgesetzt würde. Der Interviewpartner war aber skeptisch bei der Annahme, dass eine solche Applikation den Inhalt eines physischen Protokolls ganz ablösen würde. Seiner Ansicht nach sind die aktuellen physischen Protokolle zu umfangreich, um diese durch eine Applikation ersetzen zu können.

Ein weiterer Anhaltspunkt war das Verwalten der Wohnungen über eine Applikation. Eine solche Funktionalität sei sehr wünschenswert. Nach Aussage des Interviewpartners gibt es keine vernünftige und auch kostengünstige Anwendung, welche diesen Use Case abdecken kann.

Herausforderung Aus der Voranalyse ergaben sich die zu erarbeiteten Funktionalitäten, welche eine Applikation erfüllen muss. Dabei stehen das konzeptionelle Layout, der Inhalt des Protokolls sowie ein effizientes Navigieren durch die Bestandesaufnahme im Zentrum.

1.7 Lösungskonzept

Konzeptphase Im Lösungskonzept werden die Funktionalitäten der Applikation genauer beschrieben. Für das Lösungskonzept wurden gesammelte Informationen aus der Protokollanalyse und dem Leitfadeninterview miteinbezogen. Diese Informationen waren die Grundlage für die wichtigsten zu treffenden Entscheidungen für das Team.

1.7.1 Produkt Perspektive

Zweck Mit dieser Prototyp-Applikation soll eine Alternative für die heute zeitaufwändigen, oft noch physischen Prozesse bei einer Wohnungsübernahme zur Verfügung gestellt werden.

1.7.2 Benutzer Charakteristik

Zielgruppen Zur Zielgruppe gehören Immobilienverwaltungen und Vermieter, welche das Wohnungsübernahmeprotokoll effizient und digital erstellen möchten. Die Funktionalitäten sollten dabei eine hohe Benutzerfreundlichkeit aufweisen.

1.7.3 Einschränkungen

Abhängigkeit Aufgrund der Google Authentifizierung und *Google Firebase Cloud*-Anbindung wird der Nutzer mit seinem mobilen Gerät mit dem Internet verbunden sein müssen, um die Funktionalitäten nutzen zu können.

1.7.4 Funktionalitäten

User Interface Entwürfe Für das Lösungskonzept wurden vom Team *User Interface*-Entwürfe erstellt. Diese Entwürfe dienen dazu, die einzelnen Funktionen genauer zu beschreiben. Die Wahl der Farben und Anordnungen der Elemente wurden bei der definitiven Umsetzung zum Teil nochmals abgeändert. Aus Zeitgründen wurden die *User Interface*-Entwürfe nach einem zweiten Entwurf nicht mehr weiter ausgearbeitet.

Nachfolgende Bilder in den Unterkapiteln sind vom Team designte *User Interface*-Entwürfe und dienen der Erläuterung des Lösungskonzepts.

1.7.4.1 ImmoP-Applikation Übersicht

Idee / Konzept Beim Start der Applikation soll der User eine Übersicht über alle bereits im System erfassten Wohnungen erhalten. Für eine einfache Navigation steht dem Nutzer im Header ein Menü zur Verfügung. Die Anwendung einer Liste gibt dem User eine gute Übersicht seiner erstellten Wohnungen.

Design Nachfolgende Bilder zeigen das Navigationsmenü und die Übersicht der Wohnungen.

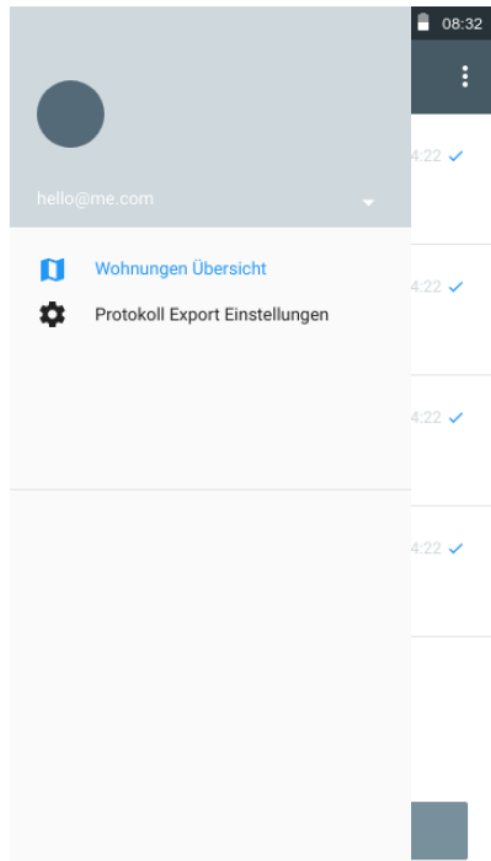


Abbildung 13 ImmoP Menü

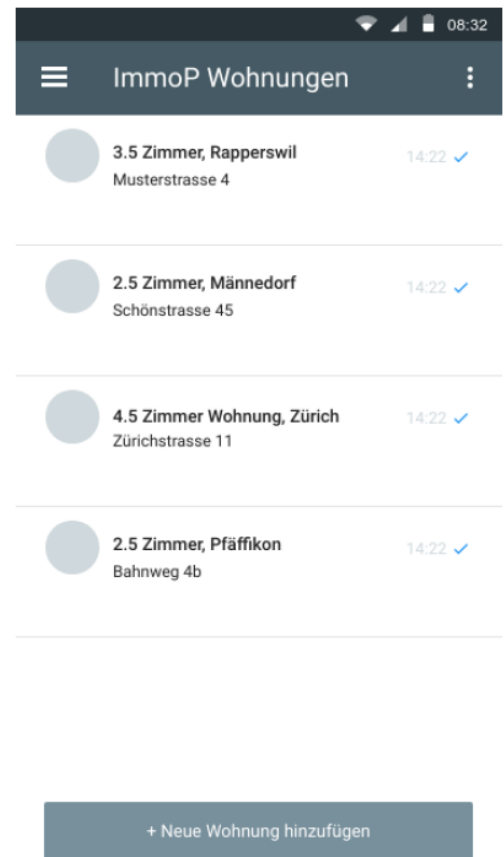


Abbildung 14 ImmoP Wohnungsübersicht

1.7.4.2 Wohnung Verwaltung

Idee / Konzept Der User soll die Möglichkeit haben, eine neue Wohnung auf einfache Art zu erstellen.

Die wichtigsten Informationen für ein Protokoll sind dabei folgende Inhalte:

- *Name der Wohnung*
- *Strasse und Hausnummer*
- *PLZ und Ort*
- *Etage der Wohnung*

Um dem User die Möglichkeit zu geben, noch weitere Informationen anzugeben, wurde ein zusätzliches Inputfeld «Beschreibung» definiert.

Der User soll zusätzlich die Möglichkeit haben, Standardräume und -objekte automatisch erstellen zu lassen. Die Liste der Standardräume und -objekte wurde mithilfe der Protokollanalyse zusammengestellt.

Für die Detailansicht einer individuellen Wohnung wurde ein Dashboard konzipiert. In diesem Dashboard kann der User einfach und schnell auf die einzelnen Räume der ausgewählten Wohnung zugreifen, Protokolle verwalten und die wichtigsten Informationen der Wohnung einsehen. Das Dashboard wurde entworfen, um einen guten Überblick über den Inhalt der einzelnen Wohnungen zu erhalten.

Design

Nachfolgende Bilder zeigen die Erstellansicht einer Wohnung und zwei Dashboard-Ansichten einer Wohnung.

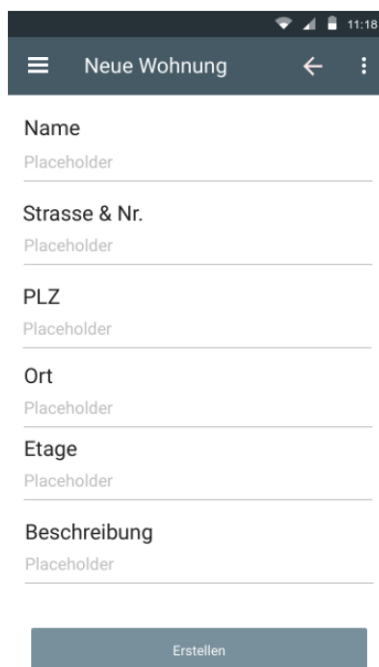


Abbildung 15 Wohnung erstellen

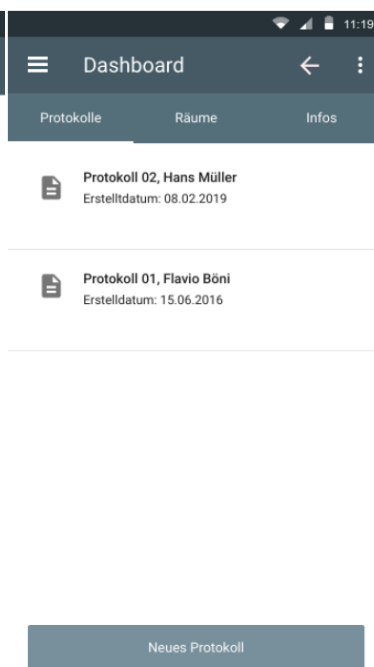


Abbildung 16 Dashboard Protokolle

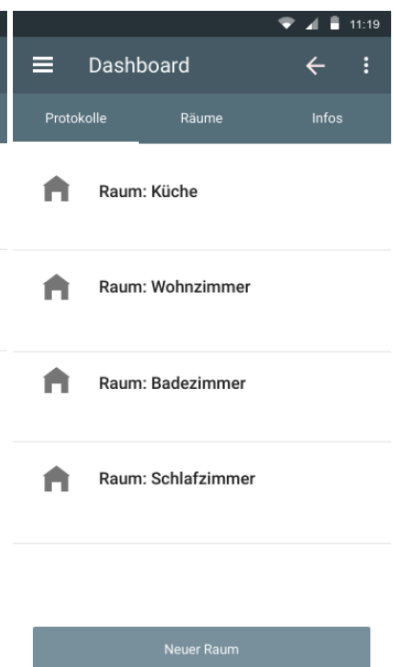


Abbildung 17 Dashboard Räume

1.7.4.3 Raum- und Objektverwaltung

Idee / Konzept

Ähnlich wie bei der Verwaltung einer Wohnung, soll der User die Möglichkeit haben, seine Wohnung mit Räumen und Objekten einzurichten. Das Erstellen von Räumen und Objekten sollte möglichst einfach und effizient gemacht werden können. Auch wenn Standardräume und -objekte bereits erstellt wurden, soll der User immer noch die Möglichkeit haben, weitere Räume und Objekte manuell hinzuzufügen.

Für eine effiziente Einrichtung der Wohnung wird dem User eine Liste von häufig vorkommenden Räumen und Objekten zur Verfügung gestellt. Diese Liste wurde mithilfe der Protokollanalyse zusammengestellt und soll als Dropdown-Menü implementiert werden.

Raum:

Beim Erstellen eines neuen Raums soll nur ein Name als Input verlangt werden. Dies entspricht den minimalen Anforderungen eines physischen Protokolls.

Objekt:

Für das Erstellen eines Objektes in einem Raum sollen folgende Angaben gemacht werden können:

- Objektname
- individueller Name (optional)
- Zustand des Objektes (Auswahlmöglichkeit zwischen Zuständen)
- Einbaujahr des Objektes (wenn nötig)
- Seriennummer des Objektes (wenn nötig)

Objektzustand:

Der User kann einem Objekt beim Erstellen einen Zustand zuweisen. Für die Wahl der verschiedenen Möglichkeiten an Zuständen wurde wiederum die Protokollanalyse miteinbezogen. Weitere Erläuterungen zum Objektzustand werden im Kapitel [1.7.5.4 Schadenaufnahme](#) gemacht.

Navigation:

Objekte können erstellt bzw. erreicht werden, indem ein Raum ausgewählt wird.

Design

Nachfolgende Bilder zeigen die Erstellansicht eines Objektes und die Objektübersicht in einem Raum.

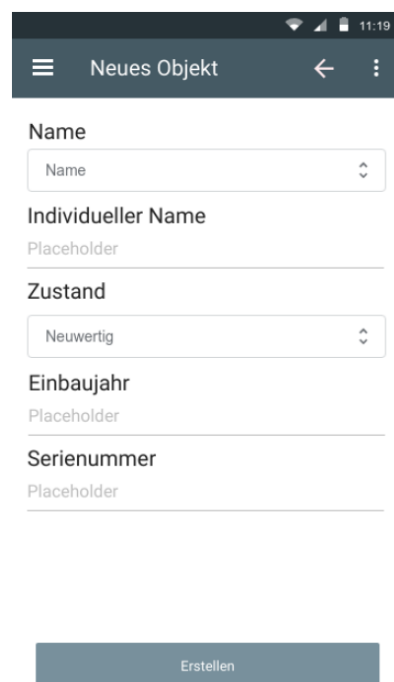


Abbildung 18 Objekt erstellen

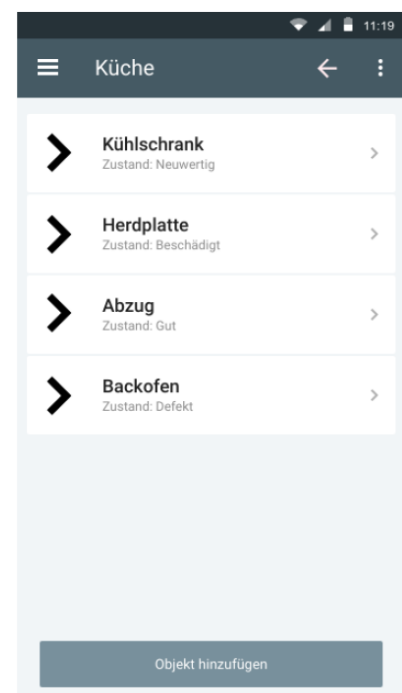


Abbildung 19 Objektübersicht

1.7.4.4 Intelligente Navigation

Idee / Konzept

Eine der grössten Herausforderungen war der Entwurf eines Prozesses für eine Schadensaufnahme. Das Team hat sich entschlossen, eine sogenannte «intelligente Navigation» für die Schadensaufnahme zu entwickeln. Durch die Erkenntnisse aus dem Leitfadeninterview wusste das Team, dass eine Überarbeitung der Schadensaufnahme wünschenswert ist. Die Protokollanalyse hat gezeigt, dass sich Wohnungsprotokolle je nach Anwender unterscheiden. Aus diesem Grund wurden die wichtigsten gesammelten Gemeinsamkeiten aus der Protokollanalyse und die wichtigsten Anforderungen aus der Bedarfsanalyse (Leitfadeninterview) ausgewählt und für das Konzept der «intelligenten Navigation» miteinbezogen.

Einer der wichtigsten Punkte bei einer Wohnungsübernahme ist, dass alle Räume und Objekte kontrolliert werden. Um zu verhindern, dass ein Raum oder ein Objekt vergessen geht, verlangt unsere konzipierte Navigation, dass der User jeden Raum und dessen Objekte auf dem Interface angezeigt bekommt. Dem Team war wichtig, dass der Benutzer die Freiheit hat, selbst zu entscheiden, wie er durch die Wohnung navigieren möchte. Aus diesem Grund soll die Reihenfolge der Räume vom Benutzer selbst bestimmt werden können. Dafür wird jeweils eine Liste der noch verbleibenden Räume angezeigt. Per Klick auf einen Raum springt die Navigation in den gewünschten Raum und die zugewiesenen Einrichtungsobjekte werden geladen.

Der Workflow, um ein vollständiges Protokoll zu erstellen, soll folgendermassen aussehen:

1. *Protokoll erstellen und nötige Informationen hinterlegen*
2. *Schadensaufnahme durch die «intelligente Navigation» durchführen*
3. *Schadensaufnahme abschliessen*
4. *Protokoll exportieren*

Beim Erstellen eines neuen Protokolls für eine Wohnung soll ein Algorithmus im Hintergrund alle Räume und deren Objekte aufbereiten. Der Algorithmus soll die Flexibilität bei der Raumauswahl gewährleisten. Zusätzlich soll sichergestellt werden, dass eine Schadensaufnahme erst abgeschlossen werden kann, wenn alle Räume und deren Objekte abgearbeitet wurden.

Angaben Protokoll:

Zusätzlich zu den aufgenommenen Schäden gehören auch allgemeine Informationen zum Protokoll. Das Team hat sich dabei wieder der Erkenntnisse aus der Protokollanalyse bedient.

Die wichtigsten Informationen für ein brauchbares Protokoll sind:

- *Datum Schadensaufnahme/Wohnungsübernahme*
- *Name Vormieter*
- *Name Nachmieter*
- *Zählerstand Heizung*
- *Zählerstand Warmwasser*

- Zählerstand Kaltwasser
- Zählerstand Stromzähler

Dem Team ist bewusst, dass die Zählerstände nicht immer einsehbar sind. Entsprechend wurden diese Felder optional definiert.

Die Voraussetzung für eine vollständige Bestandesaufnahme ist die vollständige Konfiguration der Wohnung in der Applikation.

Design

Nachfolgende Bilder zeigen einen Entwurf der «intelligenten Navigation».

Abbildung 20 Protokollangaben

Abbildung 21 Navigation Raumauswahl

1.7.4.5 Schadensaufnahme

Idee / Konzept

Die eigentliche Schadensaufnahme eines Objektes ist Hauptbestandteil der «intelligenten Navigation» und wird daher separat konzipiert. Die Schadensaufnahme soll das Schreiben von Hand und das separate Fotografieren ersetzen können. Es soll eine Ansicht konzipiert werden, welche eine umfassende Schadensaufnahme eines Objektes ermöglicht.

Ermittelte Funktionen:

Folgende Funktionalitäten, welche aus der Protokollanalyse und dem Leitfadeninterview evaluiert wurden, sollen implementiert werden:

- Der User soll einen Schaden fotografieren können. Für den Prototypen hat das Team definiert, dass es möglich sein soll, pro Objekt ein Foto aufzunehmen. Die Fotofunktion dient dem Vergleich der Schäden. Jedes erstellte Foto wird in die Schadenshistorie aufgenommen.
- Der Schaden soll mittels Textfeldes dokumentiert werden können.
- Der Mieteranteil in Prozent soll hinterlegt werden können.

Zusätzliche Funktionen:

Zusätzliche Funktionalitäten sollen dem User zur Verfügung gestellt werden. Diese zusätzlichen Funktionalitäten dienen der Effizienz einer Schadensaufnahme und den zukünftigen Auswertungen.

- Dem Objekt kann ein Zustand zugewiesen werden. Ein Zustand soll aussagen, in welcher Beschaffenheit sich ein Objekt aktuell befindet. Wenn bereits ein Zustand eines Objektes vorhanden ist, wird dieser bei der Auswertung überschrieben.
- Es soll eine Schadenshistorie verfügbar sein. Der Zweck dieser Historie ist, dass der User besser ermitteln kann, welche Schäden bereits vorhanden waren. Alle in der Vergangenheit fotografierten Schäden sollen in der Historie ersichtlich sein.

Zustände:

Bei einer Schadensaufnahme soll der User definieren können, in welchem Zustand sich ein Objekt befindet. Die Protokollanalyse hat gezeigt, dass es verschiedene Arten gibt, wie man Schäden beschreiben kann. Oft werden Schäden mit Worten beschrieben. Neben einer Beschreibung soll der Benutzer drei Auswahlmöglichkeiten haben, um einen Zustand für ein Einrichtungsobjekt anzugeben.

Folgende Zustände sollen dem Benutzer zur Auswahl stehen:

- *neuwertig*
- *gebraucht*
- *beschädigt*

Mit diesen drei Zuständen kann ein Lebenszyklus eines Objektes abgedeckt werden. Da Wohnungen renoviert und Objekte ersetzt werden können, werden alle erfassten Zustände bei Vollendung einer Schadensaufnahme überschrieben. So kann sichergestellt werden, dass die gespeicherten Zustandsdaten der Objekte nach einer Schadensaufnahme immer aktuell sind.

Navigation:

Um zum nächsten Objekt im Raum zu gelangen, werden dem Benutzer drei Möglichkeiten zur Auswahl stehen:

- **Weiter** → Ein Schaden ist vorhanden und wurde aufgenommen. Es wird zum nächsten Objekt navigiert. Das Objekt wird ins Schlussprotokoll aufgenommen.
- **Überspringen** → Das Objekt soll nicht ins Schlussprotokoll aufgenommen werden. Dieser Button kann betätigt werden, wenn ein Objekt gar nicht vorhanden ist oder keinen Einfluss mehr auf das Schlussprotokoll hat.
- **Kein Schaden vorhanden** → Wenn kein Schaden vorhanden ist, kann der User diesen Button betätigen. Die Navigation leitet den User automatisch zum nächsten Objekt weiter. Das Objekt wird ins Schlussprotokoll aufgenommen.

Sobald alle Objekte in einem Raum abgearbeitet wurden, erscheint die Liste mit den verbleibenden Räumen. Der Benutzer hat dann die Möglichkeit, den nächsten Raum auszuwählen.

Design

Nachfolgendes Bild zeigt die Ansicht einer Schadensaufnahme.



Abbildung 22 Entwurf Schadensaufnahme

1.7.4.6 Protokollauswertung & Export

- Idee / Konzept** Sobald eine Schadensaufnahme abgeschlossen wurde, soll der User die Möglichkeit haben, die erfassten Daten im Protokollformat als PDF-Datei zu exportieren. Der Zeitpunkt des Exportes soll unabhängig sein. Ein Algorithmus soll die individuellen Protokolldaten zusammenstellen und daraus eine PDF-Datei generieren. Für die Generierung der PDF-Datei soll eine bereits vorhandene PDF-Bibliothek genutzt werden, um unnötigen Mehraufwand zu verhindern. Die Wahl der PDF-Bibliothek wird in der Entwicklungsphase durchgeführt und ist nicht Teil der Konzeptphase.
- Design** Für das Design der PDF-Datei orientiert sich das Team an den verschiedenen Protokollen. Dabei ist es nicht das Ziel, bereits existierende Protokolle eins zu eins nachzubauen. Vorhandene Protokolle dienen lediglich der Orientierung für eine Struktur. Kernpunkt ist, dass alle erfassten Daten einer Schadensaufnahme strukturiert in einer PDF-Datei dargestellt werden. Es sollen alle Hauptinformationen, welche beim Erstellen eines Protokolls erfasst wurden, sowie eine Übersicht aller Räume mit dessen Objekten und Schäden ersichtlich sein.

1.7.4.7 Cloud-Lösung

- Idee / Konzept** Das Ziel ist es, alle erfassten Daten jederzeit abrufen zu können. Die Daten sollen in einer übersichtlichen Struktur in einer Datenbank abgelegt werden können. Die genaue Struktur wird in der Entwicklungsphase definiert und ist nicht Teil der Konzeptphase.
- Google Firebase Cloud** Als Datenbank für die Daten und Bilder soll die Firebase [4] Cloud-Lösung von Google genutzt werden. Das Team hat bereits Erfahrung mit der Firebase [4] Cloud-Lösung von Google gemacht.

1.8 Entscheidungen

- Begründung** Wie die Protokollanalyse gezeigt hat, unterscheiden sich die Wohnungsübernahmeprotokolle. Um mit dem Ergebnis arbeiten zu können, sollen für den Prototypen die wichtigsten Gemeinsamkeiten aus der Protokollanalyse implementiert werden. Das Leitfadenterview soll dem Team zusätzlichen einen Input für den Prozess der Schadensaufnahme geben.
- Während der Konzeptphase wurden in den Meetings mit dem Betreuer die erarbeiteten Teilkonzepte angeschaut und diskutiert.
- Es gibt verschiedene Möglichkeiten, wie Protokolle erstellt und designt werden können. Der Fokus wurde auf einen funktionsfähigen Prototypen gelegt, welcher die minimalen Anforderungen einer Schadensaufnahme und des daraus resultierenden Protokolls erfüllt.

2 Software Engineering

Inhalt Im folgenden Kapitel werden die Software Engineering technischen Aspekte beschrieben. Dabei werden die funktionalen und nichtfunktionalen Anforderungen klar definiert und die Architektur anhand UML [11] Diagrammen dargestellt.

2.1 Anforderungsspezifikation

2.1.1 Funktionale Anforderungen

Inhalt Im folgenden Kapitel werden alle funktionalen Anforderungen definiert.

2.1.1.1 Aktoren

ImmoP User Benutzer, welcher sich mit seinem Google Account angemeldet hat
Er kann Wohnungen erstellen, Räume und Objekte hinzufügen, eine Schadensaufnahme durchführen und Protokolle als PDF-Datei exportieren.

Nicht registrierter User nicht angemeldeter Benutzer

2.1.1.2 Use Cases Übersicht

Diagramm In der folgenden Abbildung sind die Use Cases, welche sich auf die Applikation beziehen, ersichtlich. Der Fokus dieser Semesterarbeit liegt auf der Umsetzung einer Android-Applikation. Das Augenmerk wird daher auf die entsprechenden Anwendungsfälle gelegt. Die einzelnen Use Cases werden in einem weiteren Schritt detaillierter beschrieben und priorisiert.

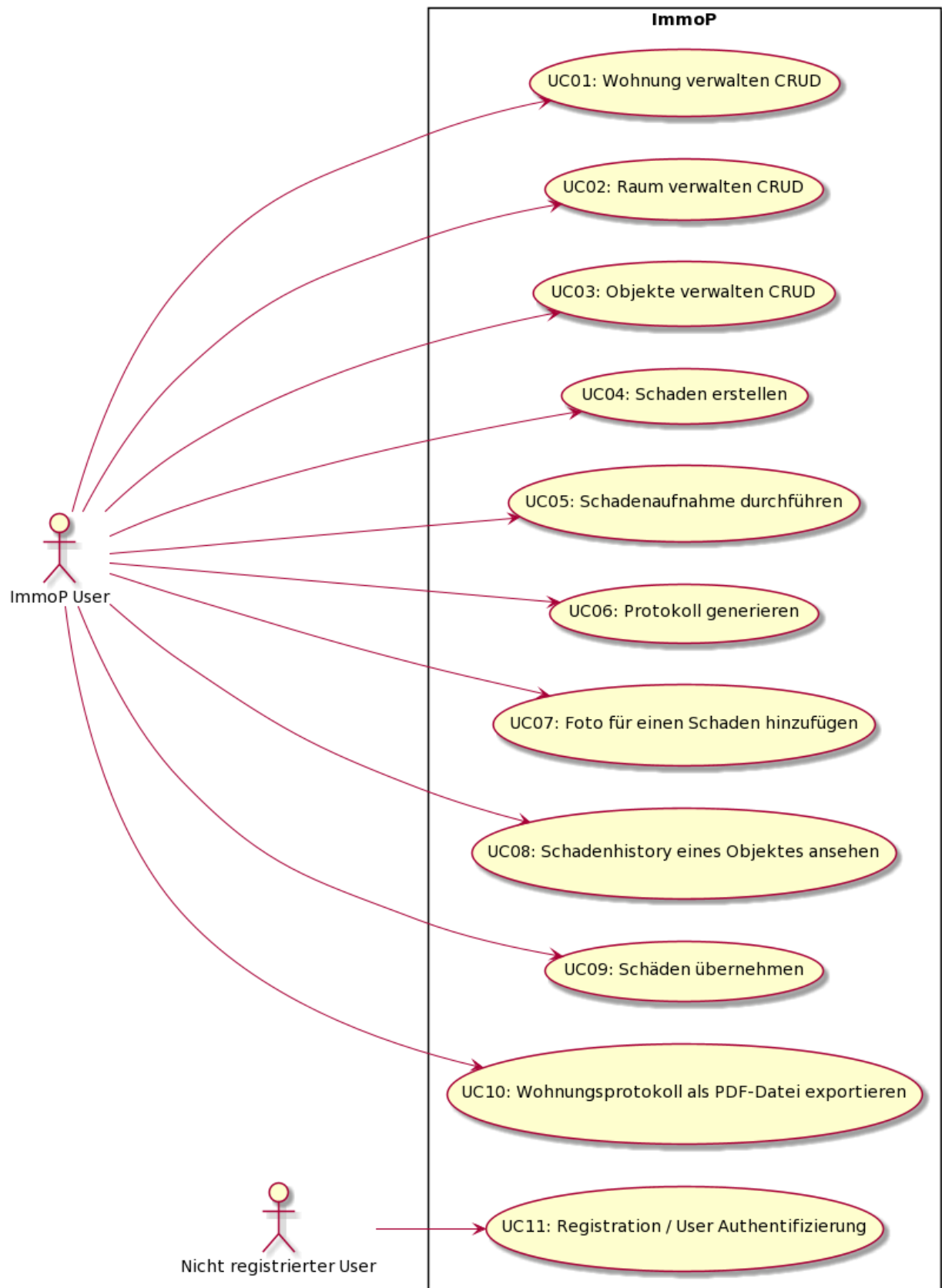


Abbildung 23 ImmoP Use Case Diagramm

2.1.1.3 Use Cases Beschreibung

UC01: Wohnung verwalten CRUD	Aktor: ImmoP User Ein Benutzer erstellt eine neue Wohnung oder lässt sich seine Wohnung anzeigen. Der Benutzer hat die Möglichkeit, Standardräume und -objekte automatisch erstellen zu lassen. Eine Wohnung kann modifiziert und gelöscht werden.
UC02: Raum verwalten CRUD	Aktor: ImmoP User Ein Benutzer erstellt für seine Wohnungen einen neuen Raum oder kann bestehende Räume modifizieren und löschen.
UC03: Objekte verwalten CRUD	Aktor: ImmoP User Ein Benutzer erstellt für einen Raum ein Einrichtungsobjekt oder kann bestehende Objekte modifizieren und löschen.
UC04: Schaden erstellen	Aktor: ImmoP User Ein Benutzer kann einen vorhandenen Schaden an einem Objekt aufnehmen und erstellen.
UC05: Schadensaufnahme durchführen	Aktor: ImmoP User Der Benutzer kann eine Schadensaufnahme durchführen. Dabei sollen alle Räume und Objekte in der Wohnung miteinbezogen werden. Durch eine intelligente Navigation wird der Benutzer durch die Wohnung geführt und kann so eine Schadensaufnahme machen.
UC06: Protokoll generieren	Aktor: ImmoP User Nachdem ein Benutzer die Schadensaufnahme beendet hat, wird von der Applikation eine Protokollstruktur generiert und persistiert.
UC07: Foto für einen Schaden hinzufügen	Aktor: ImmoP User Bei einer Schadensaufnahme hat der Benutzer die Möglichkeit, einem Objekt ein Foto hinzuzufügen. Dieses Foto wird dann der Schadenshistorie hinzugefügt.
UC08: Schadenshistorie eines Objektes ansehen	Aktor: ImmoP User Der Benutzer soll zwei Möglichkeiten haben, die Schadenshistorie von Objekten anzuschauen, und zwar beim Verwalten des einzelnen Objektes und bei der Schadensaufnahme, um neue Schäden mit alten zu vergleichen.
UC09: Schäden übernehmen	Aktor: ImmoP User Schäden, welche zwischen einem Wohnungswechsel nicht behoben wurden und daher bei einer erneuten Schadensaufnahme immer noch vorhanden sind, können von der Schadenshistorie durch den Benutzer übernommen werden.
UC10: Wohnungsprotokoll als PDF-Datei exportieren	Aktor: ImmoP User Der Benutzer kann vorhandene Protokollstrukturen in eine PDF-Datei exportieren und anzeigen lassen.

UC11: Registration **Aktor:** nicht registrierter User

/ User Authentifizierung Ein Benutzer registriert sich mit seinem Google Account. Nach erfolgreicher Anmeldung können alle Funktionalitäten genutzt werden.

2.1.1.4 Use Cases Priorisierung

Priorisierung

- Priorität 3 → Kernfunktionen, essenziell für die Applikation
- Priorität 2 → Zusatzfunktionen, kleine Erweiterungen für die Applikation
- Priorität 1 → Zusatzfunktionen, Ausbaumöglichkeiten für die Applikation

Wir haben für die *Use Case*-Priorisierung eine Gewichtung von 1 bis 3 durchgeführt. Die Zahl 3 hat die höchste Priorität, ist also essenziell für die Applikation. Die Zahl 2 hat die mittlere Priorität und wird vom Team als «nice to have» eingestuft. Priorität 2 wird implementiert, sobald alle Use Cases mit der Priorität 3 umgesetzt wurden. Alle Use Cases mit Priorität 1 können am Schluss, wenn noch genügend Zeit zur Verfügung steht, umgesetzt werden.

Use Cases mit der Priorität 1 sind relevant, falls sich das Team mit der Zeit verschätzt und daher zu früh mit der Entwicklung fertig wird.

Use Cases mit Priorität 3:

UC01, UC02, UC03, UC04, UC05, UC06, UC07, UC10

Use Cases mit Priorität 2:

UC08, UC11

Use Cases mit Priorität 1:

UC09 und die Modifikation aus den CRUD-Operationen aus den Use Cases UC01, UC02 und UC03.

Begründung

Die meisten Use Cases enthalten Kernfunktionen für die Applikation. Deshalb haben die meisten die Priorität 3. Das Team hat für eine brauchbare Applikation die wichtigsten Use Cases mit der Priorität 3 versehen. Dabei wurden die gesammelten Informationen aus der Vorarbeit miteinbezogen. Damit eine Schadensaufnahme (**UC05**) durchgeführt werden kann, sind **UC01** bis **UC04** notwendig. Für die beiden Use Cases **UC06** und **UC07** muss der **UC05** zuerst umgesetzt werden.

Diese Abhängigkeiten der Use Cases gibt dem Team eine klare Vorgabe, in welcher Reihenfolge die Funktionen implementiert werden müssen.

Um möglichst schnell ein interessantes Ergebnis zu erhalten, hat sich das Team entschieden, **UC08** und **UC11** mit der Priorität 2 zu kategorisieren, weil die Schadenshistorie eine nützliche Zusatzfunktion und keine Kernfunktion ist. Dasselbe gilt für **UC11**. Ein Login ist für unseren Fall nicht mit der höchsten Priorität zu versehen, da es sich um keine Kernfunktionalität handelt.

Um nicht zu viel Zeit für die CRUD-Operationen zu verbrauchen, hat sich das Team entschieden, die Modifikation für eine Wohnung, Räume und Objekte mit der Priorität 1 zu

versehen. Damit falsch angelegte Wohnungen entfernt werden können, soll die LösCHFunktion jedoch höher priorisiert werden.

UC09 wurde ebenfalls als eine Ausbaumöglichkeit definiert. Das Team erachtet diese Funktionalität als nicht zwingend notwendig. Sie wurde deshalb niedrig priorisiert.

2.1.2 Nichtfunktionale Anforderungen

Methodik Für die nichtfunktionalen Anforderungen werden die Qualitätsmerkmale von FURPS [12] verwendet. Für eine zielführende Definition der nichtfunktionalen Anforderungen wird der SMART [13] Ansatz gewählt.

2.1.2.1 Funktionalität

Angemessenheit Die Google Authentifizierung soll möglichst simpel gehalten werden. Der User muss lediglich einen Google Account besitzen. Ist dieser Google Account bereits auf dem Smartphone registriert, wird kein Login benötigt.

Richtigkeit Erfasste Daten, welche in der Google Firebase Cloud persistiert werden, werden dem User bei erfolgreichem Login nach bestem Gewissen zur Verfügung gestellt. Für die Erstellung von Wohnung, Raum, Objekt oder Schaden sind nicht zwingend alle Angaben erforderlich.

Sicherheit Nicht angemeldete User dürfen keinen Zugriff auf die Daten aus der Google Firebase Cloud haben.

2.1.2.2 Benutzbarkeit

Input Alle Inputfelder sind mit einem Label beschriftet, damit dem User klar ist, welcher Input verlangt wird.
Sensible Inputdaten werden entsprechend validiert.

Bedienbarkeit Die Applikation soll so konzipiert werden, dass der User die Applikation nach 5 Minuten versteht. Für eine benutzerfreundliche Ansicht werden Standard-Bibliotheken von Google für das Design verwendet. Dabei wird die Farbwahl des Designs einheitlich verwendet.

Navigation Der User kann einfach durch die Applikation durchnavigieren.

2.1.2.3 Zuverlässigkeit

Reife Die Funktionalitäten der Applikation sollten in 90% der Fälle wie vorgesehen funktionieren. Bei Systemfehlern sollte die Applikation nach einem Neustart wieder voll funktionsfähig sein. Ein Konzept für das Logging wurde noch nicht erarbeitet. Für den Prototypen werden Systemfehler lediglich auf der Developer-Konsole ausgegeben. Ein Konzept für ein Logging von Systemfehlern könnte für die Zukunft geplant werden.

Fehlertoleranz Sensible Inputdaten müssen validiert werden. Bei Fehleingaben wird der User informiert.

Wiederherstellbarkeit Daten werden zurzeit nur auf einer Cloud persistiert. Daten, welche einmal gelöscht werden, können nicht mehr wiederhergestellt werden. Eine Backuplösung könnte für die Zukunft geplant werden.

2.1.2.4 Effizienz

Laden der Daten Die Applikation sollte innerhalb von 2-5 Sekunden gestartet sein (Login-Screen). Nach erfolgreichem Login werden die Daten in 90% der Fälle innerhalb von 5 Sekunden geladen. Der Export einer PDF-Datei sollte in 90% der Fälle innerhalb von 5 Sekunden erfolgen.

2.1.2.5 Änderbarkeit

Modifizierbarkeit Funktionale Änderungen am Prototypen sollten innerhalb von wenigen Sekunden vom Entwickler erneut auf dem Smartphone bereitgestellt werden können.

Stabilität Der Prototyp sollte in 90% der Fälle stabil anwendbar sein.

2.1.2.6 Übertragbarkeit

Installation Der Prototyp kann über die Entwicklungsumgebung innerhalb von 10 Minuten auf ein Android Smartphone installiert werden, welches die Android Version 10 (API Level 29) [3] installiert hat.

2.2 Architektur

Inhalt In diesem Kapitel wird die Architektur der zu entwickelnden Android-Applikation beschrieben. Architekturentscheide werden begründet und verwendete Technologien aufgezeigt.

2.2.1 Domainanalyse

Domainmodell Nachfolgendes Bild zeigt das Domainmodell der Applikation.

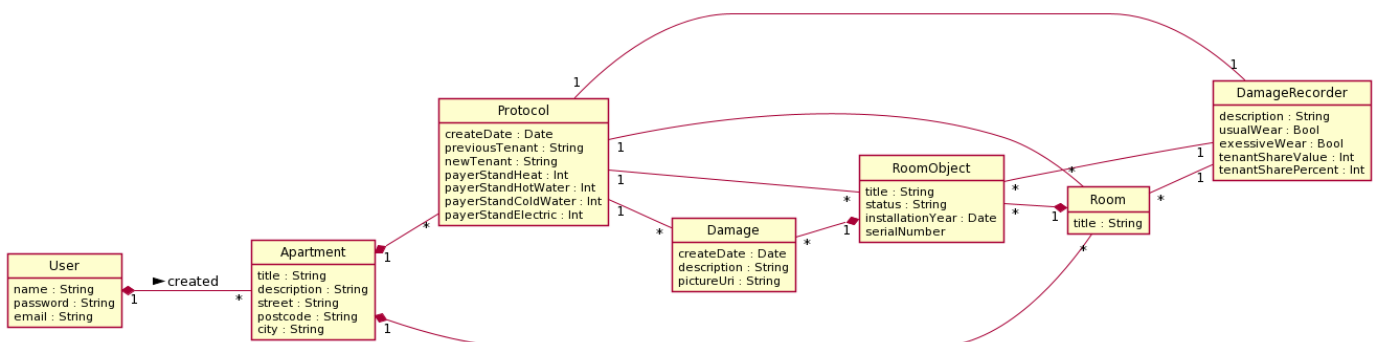


Abbildung 24 ImmoP Domainmodell

- User** Der User spielt die Hauptfigur. Jeder User mit einem gültigen Google Account kann die Applikation nutzen. Ein User kann erstellte Elemente verwalten.
- Apartment** Ein *Apartment* wird von einem authentifizierten User erzeugt. Jedes erstellte *Apartment* wird einem User zugewiesen. Ein *Apartment* enthält Informationen über die Wohnung. *Protocol* und *Room* sind Kinderelemente eines Apartments.
- Room** Ein *Room* wird von einem authentifizierten User erzeugt. Jeder erstellte *Room* wird einem *Apartment* zugewiesen. Ein erstellter *Room* enthält einen Namen. *RoomObjects* sind Kinderelemente eines *Rooms*.
- RoomObject** Ein *RoomObject* wird von einem authentifizierten User erzeugt. Jedes erstellte *RoomObject* wird einem *Room* zugewiesen. Ein erstelltes *RoomObject* enthält einen Namen, einen Status, ein Einbaujahr und eine Seriennummer. *Damages* sind Kinderelemente eines *Rooms*.
- Damage** Ein *Damage* wird von einem authentifizierten User während einer Schadensaufnahme erzeugt. Jeder erstellte *Damage* wird einem *RoomObject* zugewiesen. Ein erstellter *Damage* hat ein Erstellungsdatum, eine Beschreibung und eine Foto URI.
- Protocol** Ein *Protocol* wird von einem authentifizierten User erstellt. Durch das Erstellen eines *Protocols* wird eine Schadensaufnahme gestartet. Jedes *Protocol* wird einem *Apartment* zugewiesen. Ein *Protocol* enthält alle Informationen über eine durchgeführte Schadensaufnahme.

DamageRecorder Ein *DamageRecorder* wird bei einer Schadensaufnahme benötigt. Dieses Objekt soll den Kern der «intelligenten Navigation» darstellen. Der *DamageRecorder* wird für die Navigation bei einer Schadensaufnahme verantwortlich sein.

2.2.2 Systemübersicht

Inhalt Nachfolgendes Bild stellt das Deployment Diagramm der ImmoP-Applikation dar. Die Applikation wird auf einem Android-Gerät installiert und alle Daten werden auf der Google Firebase [4] Cloud persistiert.

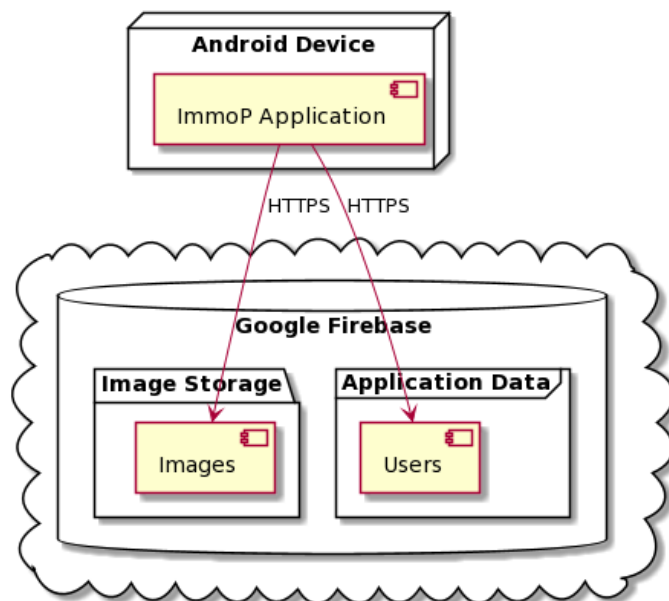


Abbildung 25 ImmoP Deployment Diagramm

2.2.3 Logische Architektur

Schichtenmodell Für die logische Architektur der Applikation hat sich das Team für ein *Model View Controller*-Architekturmuster entschieden. Zwischen *Controller* und *Model Layer* wurde noch ein *Service Layer* hinzugefügt. Das Team hat das MVC [14] Muster gewählt, da es für die Entwicklung einer Android-Applikation bevorzugt wird. Dieses Muster erleichtert es, die Applikation zukünftig einfach zu erweitern. Folgendes Bild zeigt die logische Architektur der ImmoP-Applikation.

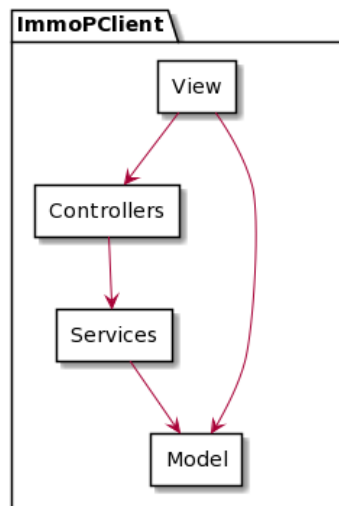


Abbildung 26 Schichtenmodell

- View** Die *View* repräsentiert den *Presentatin Layer* respektive das User Interface. In der View werden alle User-Interaktionen abgefangen und weiter an die Controller geleitet.
- Controllers** Die *Controller* sind das Routing für die Datenverarbeitung. Die *Controller* nehmen User-Anfragen entgegen und delegieren diese zu den entsprechenden Services, welche für die Datenverarbeitung und Datenaufbereitung zuständig sind.
- Services** Die *Services* umfassen die Businesslogik der Problemdomäne. Die Service-Schicht wird im Diagramm separat dargestellt, ist aber ein Teil des Modells.
- Model** Das *Model* repräsentiert den *Data Access Layer*. Auf dieser Schicht werden Zugriffe auf gespeicherte Daten und die Manipulation von diesen möglich.

2.2.4 Package-Struktur

Packages

Nachfolgendes Diagramm zeigt die Package-Struktur der ImmoP-Applikation und deren Abhängigkeiten.

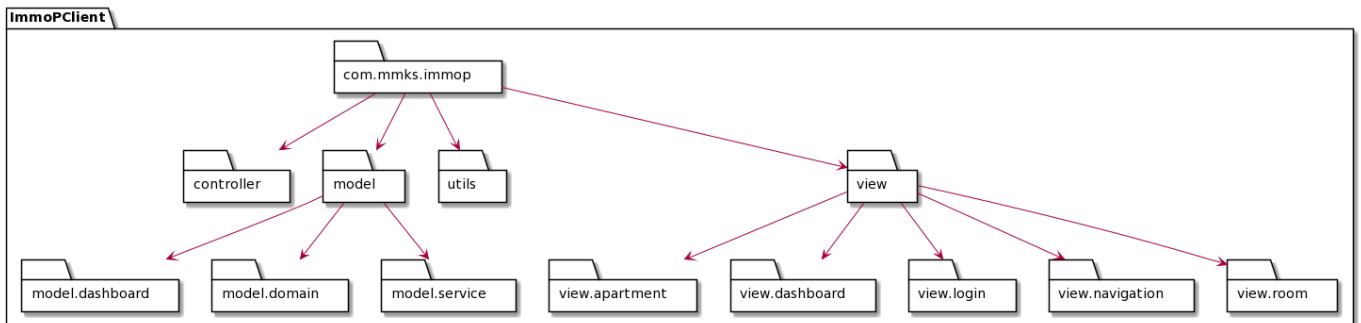


Abbildung 27 Package Diagramm

2.2.5 Technologien

Komponente	Technologie
Mobile-App (Android)	<ul style="list-style-type: none"> • XML • Java
Datenbank	<ul style="list-style-type: none"> • Google Firebase Cloud

Tabelle 2 Technologien

Begründung

Begründet wird die Verwendung der aufgeführten Technologien mit Erfahrungen in der Vergangenheit. Zwar muss erwähnt werden, dass sich die Erfahrungen in Grenzen halten, was die allgemeine Android-Entwicklung angeht, jedoch wurden schon viele Erfahrungen mit der Programmiersprache Java in diversen Modulen gemacht.

Firebase ist eine Cloud-Lösung von Google und bietet diverse Features. Für das Speichern von Daten wird eine Realtime-Datenbank [15] verwendet, welche Daten in JSON [16] Format speichert. Für die Bilderablage wird die Storage [17] Lösung verwendet. Die Storage-Lösung bietet eine strukturierte Ablage für Dateien.

3 Testing

Unittests Für die Android-Applikation wurden Unit-Tests erstellt. Diese überprüfen die Controller und Services der Businesslogik auf ihre Korrektheit.

Systemtests Die Systemtests der ImmoP-App wurde zum Grössten Teil manuell durchgeführt. Für die Systemtests wurden folgende Android-Geräte verwendet:

Smartphone:

- *Huawei P30 Pro*
- *Google Pixel 3*
- *Samsung Galaxy 20 FE*

Tablet:

- *Huawei AGS-W09*

Die Systemtests wurden anhand der Use Cases aufgebaut. Die Use-Cases wurden in mehrere Testfälle unterteilt und getestet. Das Testprotokoll ist im **Anhang** einsehbar.

Auswertung / Erkenntnisse Von den 45 Systemtests konnten 41 Tests erfüllt werden. Die Applikation macht was sie soll. Die Protokolle werden mit den zugewiesenen Schäden erfolgreich in der Cloud gespeichert und geladen.

Nicht erfüllte Testfälle:

Zwei Testfälle konnten nicht erfüllt werden. Diese wurden als Bug klassifiziert und bestehen aktuell noch.

- **Test 5.7:** Der User kann mit dem Rückwärts-Pfeil zurück zur Raumsicht und einen anderen Raum selektieren.
- **Test 5.6:** Der User löscht ein Objekt oder einen Raum welche bereits in einem Protokoll referenziert sind. Das Protokoll kann weiterhin exportiert werden.

Erfüllte Testfälle mit Einschränkungen:

- **Test 6.1:** Der Testfall konnte erfüllt werden, jedoch wurde bei einer schlechten Internet Verbindung eine lange Ladezeit beobachtet. Bei einer Protokollerstellung werden die erfassten Schäden, inklusive Bilder, in die Cloud gespeichert. Dieser Speichervorgang dauert bei einer schlechten Internetverbindung und grossen Bilddateien länger als erwartet. Da dieses Problem erst am Ende der Entwicklungsphase entdeckt wurde, konnte es aus Zeitgründen leider nicht behoben werden.

Nicht durchgeführte Testfälle:

Folgende Testfälle konnten nicht durchgeführt werden, da die entsprechenden Funktionalitäten aus Zeitgründen nicht implementiert werden konnten.

- **Test 8.4:** In der Objekt-Ansicht kann der Benutzer ein Einrichtungsobjekt auswählen und die Schadenshistorie einsehen.
- **Test 9.1:** Der User kann einen bestehenden Schaden für das Einrichtungsobjekt übernehmen.

Testing mit dem Tablet:

Bei den manuellen Tests mit dem Huawei Tablet wurde festgestellt, dass das Tab-Layout und die Raumansicht nicht korrekt dargestellt werden. Da die Anforderungen an ein Smartphone gerichtet waren und nicht an ein Tablet, sehen wir diese Einschränkung nicht als gravierend. Für die Zukunft müsste man das Layout-Konzept für eine Tablet Version umbauen.

4 Umsetzung

Inhalt Im folgenden Kapitel werden die Implementierungen der primären Komponenten mit ihren wichtigsten Funktionalitäten beschrieben. Für die Beschreibung werden der *Presentation Layer* und die Businesslogik separat betrachtet.

4.1 Implementierung

4.1.1 Presentation Layer

Design Für das *User Interface*-Design der Android-Applikation wurde das Google Material Design [18] angewendet. Es wurden ausschliesslich Komponenten aus der Google Material Designsprache verwendet. Das gesamte resultierende User Interface wird im Kapitel [Ergebnisse](#) genauer beschrieben.

Das User Interface besteht primär aus zwei Teilen.

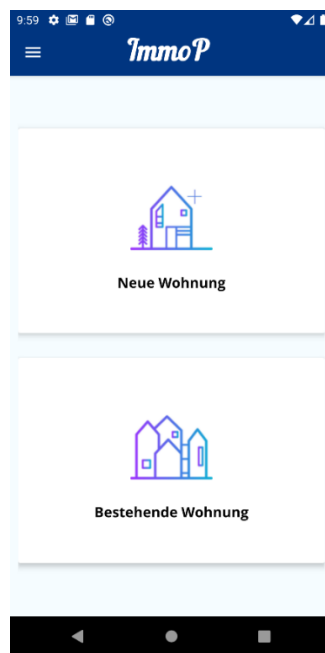


Abbildung 28 Teil Hauptansicht

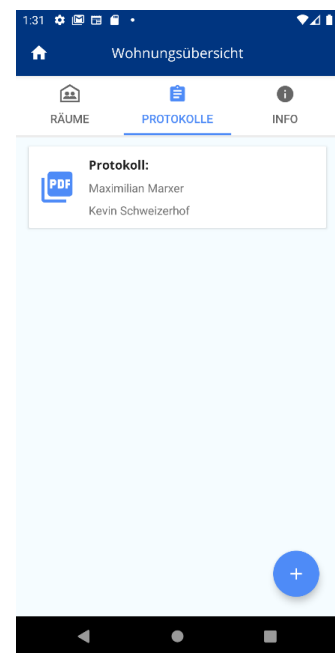


Abbildung 29 Teil Wohnungsübersicht

Teil Hauptansicht Nachdem sich der Benutzer erfolgreich angemeldet hat, gelangt er zur Hauptansicht. Die Hauptansicht wurde mit einer Toolbar ausgestattet, welche oben rechts in der Abbildung *Teil Hauptansicht* ein *Hamburger-Menu-Item* hat. Über dieses Menü kann sich der Benutzer wieder abmelden.

Teil Wohnungsübersicht Die Abbildung *Teil Wohnungsübersicht* zeigt das Dashboard einer Wohnung. Dort wird anstelle eines *Hamburger-Menu-Items* ein *Haus-Icon* angezeigt, welches den Benutzer zur Hauptansicht zurückführt. Für eine hohe Usability beim Wechsel zwischen den Listen *Räume*, *Protokolle* und *Infos* wurden Material-Tabs von Google Material Designs verwendet. Der Erstellen-Button für eine neues Protokoll wurde mittels Floating-Buttons mit Plus-Symbol implementiert.

4.1.2 Businesslogik

4.1.2.1 Universally Unique Identifier

UUID Ein Universally Unique Identifier [19] (kurz UUID) repräsentiert eine eindeutige Zeichenfolge. Für unsere Applikation bekommt jedes Objekt, welches in der Firebase-Cloud gespeichert wird, eine solche UUID. Diese UUID dient als eindeutige Identifikation eines Objektes in der Firebase-Datenbank. Durch diesen Identifier können einfach und effizient Referenzen und Beziehungen abgebildet werden.

Beispiel:

Nachfolgendes Bild zeigt einen Ausschnitt aus der Firebase-Datenbank eines Objektes und dessen UUID. Vor der UUID wurde der Objektname eingefügt, um die UUIDs besser auseinander zu halten.



Abbildung 30 Beispiel UUID

Implementation Folgende Objekte erhalten beim Erstellen eine UUID:

- *Apartment (erstellte Wohnung)*
- *Room (erstellter Raum)*
- *RoomObject (erstelltes Objekt in einem Raum)*
- *Damage (erstellter Schaden)*
- *Protocol (Protokoll bei einer Schadensaufnahme)*

Eine UUID kann über den UUIDController erstellt werden. Folgender Code repräsentiert die Generierung einer UUID:

```
public String generateUUID(){
    UUID = UUID.randomUUID();
    return uuid.toString();
}
```

4.1.2.2 Kontext

ImmopContext Der ImmopContext repräsentiert den aktuellen Kontext der Applikation. Über den ImmopContext können alle geladenen Daten abgefragt werden. Der ImmopContext ist Teil vom Model und wird von den Services verwendet. Der ImmopContext wird beim Starten der Applikation mit den geladenen Daten der Firebase-Datenbank abgefüllt. Für das Laden der Daten wurde der Eager Loading [20] Ansatz gewählt. Beim Start der Applikation werden alle Daten mit Ausnahme der Bilder geladen. Geladene Daten werden im ImmopContext gespeichert.

Implementation Folgender Codeausschnitt zeigt die Verwaltung der *Apartments* im ImmopContext.

```
//Apartment List
private ArrayList<Apartment> apartments = new ArrayList<>();
public void setApartments(ArrayList<Apartment> apartments){
    this.apartments = apartments;
}
public ArrayList<Apartment> getApartments(){
    return apartments;
}

public void addNewApartment(Apartment apartment){
    apartments.add(apartment);
}
public void deleteApartment(Apartment apartment){
    apartments.remove(apartment);
}
```

4.1.2.3 Firebase-Anbindung

Laden

Für das Laden der Daten wird die Firebase API [21] genutzt.

Zur Laufzeit gibt es grundsätzlich zwei Zeitpunkte, an welchem Daten von der Firebase-Cloud geladen werden. Beim Start der Applikation werden alle Wohnungsdaten geladen und in den ImmoPContext gemappt. Dieser Vorgang wird ausgeführt, sobald sich ein User erfolgreich mit seinem Google Account angemeldet hat. Anhand der Login-Informationen kann ermittelt werden, welche Daten zu welchem Benutzer gehören. Die Bilder für die Schadenshistorie werden geladen, wenn diese benötigt werden. Im Endeffekt bedeutet das, dass bei einer Schadensaufnahme eines Einrichtungsobjektes referenzierte Bilder in der Schadenshistorie geladen werden. Dieser Ansatz wurde vom Team gewählt, damit das Starten der Applikation nicht zu lange dauert. Es werden lediglich Daten in Form einer JSON [16] Struktur geladen.

Speichern

Für das Speichern der Daten wird die Firebase API [21] genutzt.

Daten können zu einem beliebigen Zeitpunkt vom User gespeichert werden. Ein Speichervorgang wird durch folgende Aktivitäten ausgeführt:

- Erstellen einer Wohnung
- Erstellen eines Raums
- Erstellen eines Objektes
- Bei der Durchführung einer Schadensaufnahme werden gleichzeitig Bilder, Schäden und Protokollinformationen gespeichert.

Implementation **Laden:**

Für das Laden der Daten wurde die Firebase API verwendet. Nachfolgender Codeausschnitt zeigt das Laden der Daten beim Start der Applikation.

```
//Create Database connection
database = FirebaseDatabase.getInstance()
    .getReference()
    .child(ImmoContext.getInstance().getCurrentUser().getFirebaseId())
    .child("ApartmentList");

//EventListener for first connection to read the data
database.addValueEventListener(readEvent);

//EventListener who loads and process data from firebase
public ValueEventListener readEvent = new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        //Process DataSnapshot

    }
}
```

Für das Laden der Bilder wird eine URI [22] verwendet, welche von Firebase nach dem Speichervorgang ausgestellt wurde. Für die Darstellung in der View wurde die Picasso [23] Bibliothek verwendet. Mit der *Picasso Bibliothek* können vereinfacht externe Bilder angezeigt werden. Da unsere Bilder im Firebase Storage gespeichert werden, benötigen wir eine Bibliothek, welche eine solche Funktion anbietet.

Nachfolgender Codeausschnitt zeigt das Laden eines Bildes von der Firebase-Cloud:

```
Picasso.with(getContext())
    .load(damageHistoryUriList.get(0)).into(objectImage);
```

Speichern:

Um ein Objekt zu speichern, müssen die Informationen in eine geordnete Struktur gebracht werden. Die Firebase API bietet dabei Unterstützung, indem beim Speichern alle Werte in einer Hashmap abgelegt werden können. Die JSON-Struktur wird dabei von Firebase selbst erstellt.

Nachfolgender Codeausschnitt zeigt, wie eine Apartment-Objekt in der Firebase-Cloud gespeichert werden kann. Der «setValue()» Methode wird dabei eine Hashmap mit Apartment-Attributen übergeben.

```
//Safe to firebase
FirebaseDatabase.getInstance().getReference()
    .child(currentUser)
    .child(ImmopConfig.APARTMENTLIST)
    .child(apartment.getObjectId()).setValue(apartmentEntity);
```

Beim Speichern eines Bildes wird eine Instanz des Firebase Storage genutzt. Dieser Instanz kann dann das erstellte Bild übergeben werden. Über eine Callback-Methode wird bei erfolgreichem Upload eine Download URI zurückgegeben. Mit dieser neuen URI kann das Bild wieder heruntergeladen werden.

Nachfolgender Codeausschnitt zeigt, wie ein Bild auf dem Firebase Storage gespeichert werden kann.

```
StorageReference imageStorage = FirebaseStorage.getInstance().getReference();
StorageReference imagePath = imageStorage.child(ImmopConfig.APARTMENTLIST)
    .child(apartmentId)
    .child(roomId)
    .child("Objects")
    .child(objectId)
    .child(UUIDController.getInstance().generateUUID());

imagePath.putFile(damageUri)
    .addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            taskSnapshot.getMetadata().getReference()
                .getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                    @Override
                    public void onSuccess(Uri uri) {
                        finishSavingDamage(uri, protocolDamageItem);
                    }
                });
        }
    });
```

4.1.2.4 Intelligente Navigation

Navigierung

Damit der User flexibel durch die Wohnung navigieren kann, wurden drei zentrale Funktionalitäten eingebaut. Mit diesen drei Funktionen, welche im *DamageRecorderController* der View als Routing zur Verfügung stehen, können Räume und Objekte effizient durchnavigiert werden. Während der Navigation wird der aktuelle Zustand der Navigation immer zentral festgehalten. Dieser Zustand sagt aus, welche Räume und Objekte noch nicht abgearbeitet wurden. Beim Start einer Schadensaufnahme sowie bei einem Raum- oder Objektwechsel wird der View ein Objekt mitgegeben, in welchem dieser aktuelle Zustand abgefragt werden kann. Durch dieses sogenannte *DamageRecordingObjectInfoModel*

Objekt kann evaluiert werden, ob ein Raumwechsel oder ein Objektwechsel der nächste Schritt in der Navigation sein wird. Eine funktionsfähige Backnavigation konnte noch nicht umgesetzt werden. Es wurden jedoch bereits Vorkehrungen getroffen, um diese Funktionalität zukünftig einfach zu implementieren.

Nachfolgender Codeausschnitt zeigt die Funktion, welche über den *DamageRecorder-Controller* aufgerufen werden kann, um in einen neuen *Raum* zu navigieren. Eine essenzielle Rolle spielt dabei eine Queue, welche zu Beginn mit allen Räumen befüllt wurde:

```
public DamageRecorder nextRoom(Room room){
    currentRoom = room;
    initRemainingRoomObjects(remainingRooms.get(room.getObjectId()));
    if(remainingRoomObjectsOfCurrentRoomQueue != null
        && remainingRoomObjectsOfCurrentRoomQueue.size() > 0){
        currentRoomObject = remainingRoomObjectsOfCurrentRoomQueue.poll();
        objectDetailInfoModel = new ObjectDetailInfoModel
            (currentRoom, currentRoomObject);
    }else{
        objectDetailInfoModel = null;
        currentRoomObject = null;
    }
    remainingRooms.remove(room.getObjectId());
    return this;
}
```

Nachfolgender Codeausschnitt zeigt die Funktion, welche über den *DamageRecorder-Controller* aufgerufen werden kann, um das nächste *Objekt im Raum* zu erhalten:

```
public DamageRecorder nextObject(){
    if(remainingRoomObjectsOfCurrentRoomQueue != null
        && remainingRoomObjectsOfCurrentRoomQueue.size() > 0){
        currentRoomObject = remainingRoomObjectsOfCurrentRoomQueue.poll();
        objectDetailInfoModel = new ObjectDetailInfoModel
            (currentRoom, currentRoomObject);
    }else{
        objectDetailInfoModel = null;
        currentRoomObject = null;
    }
    return this;
}
```

Die dritte erwähnte Funktion ist die Zurücknavigation in einer Schadensaufnahme. Diese Funktion wurde geplant, um bei fehlerhaften Aufnahmen in einer Wohnung zurücknavigieren zu können. Diese Funktionalität konnte jedoch aus zeitlichen Gründen nicht fertig implementiert werden. Einen Teil der Logik wurde im *DamageRecorderService* bereits implementiert.

Nachfolgender Codeausschnitt zeigt einen Teil der Backnavigation, welcher im *DamageRecorderService* bereits implementiert werden konnte:

```
public DamageRecordingObjectInfoModel undoNavigation(){
    DamageRecordingObjectInfoModel item = ImmopContext.getInstance()
        .getNavigationHistory().pop();
    ImmopContext.getInstance().setDamageRecorder(item.getDamageRecorder());
    return item;
}
```

Das Prinzip der Backnavigation ist, dass bei jedem Raum- oder Objektwechsel der aktuelle Stand auf einem Stack gespeichert wird. Wenn in einer Schadensaufnahme zurücknavigiert werden muss, kann der Stack von oben nach unten abgebaut werden.

4.1.2.5 PDF-Generierung

- Bibliothek** Für die Generierung einer PDF-Datei wurde eine bereits vorhandene Klasse von Android genutzt. Mit der PdfDocument [24] Klasse von Android können einfach und effizient PDF-Dateien erstellt werden.
- Export** Für den Export einer PDF-Datei wurde ein Algorithmus gebaut. Dieser Algorithmus erstellt eine individuelle PDF-Datei anhand der Protokolldaten, welche bei einer Schadensaufnahme gesammelt wurden. Es musste eine Logik eingebaut werden, welche für eine saubere Darstellung und Seitenverwaltung zuständig ist. Da dieser Algorithmus sehr umfangreich ist, wird an dieser Stelle kein Codeausschnitt gezeigt. Der PDF-Algorithmus, welcher für das Erstellen einer PDF-Datei zuständig ist, kann im Repository im Package «com.mmks.immop.model.service» in der Klasse PdfService angeschaut werden.

4.2 Fazit Implementierung

- Fazit Implementierung** Der *Presenation Layer* sowie der *Business Logic Layer* konnten gut unabhängig voneinander entwickelt werden. Als Schnittstelle der beiden Layer dienten die Controller, welche dem *Presenation Layer* alle nötigen Daten lieferten. Dadurch konnte das Team effizient arbeiten. Die Aufgaben wurden meistens so aufgeteilt, dass ein Teammitglied den *Presenation Layer* entwickelte und das andere Mitglied die Funktionalitäten in der Businesslogik implementierte.
- Aufgrund der geringen Android-Programmiererfahrungen, welche das Team zu Beginn mitbrachte, gab es eine steile Lernkurve. Dies führte dazu, dass gewisse Implementierungen zum Teil noch einmal überarbeitet werden mussten. Aus zeitlichen Gründen wurden ab der Hälfte der Entwicklungsphase gewisse Codeteile nicht mehr gross überarbeitet, da sonst die Gefahr bestand, dass nicht mehr alle Funktionen bis zur Deadline implementiert werden können.

5 Ergebnisse

5.1 Applikation



Abbildung 31: Splash Screen

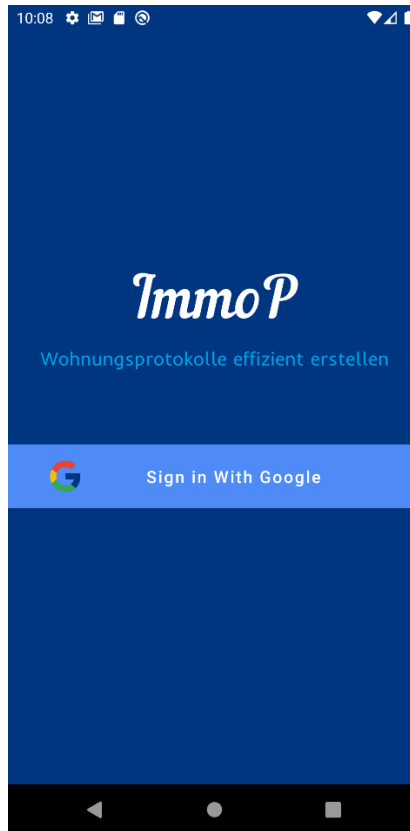


Abbildung 32: Login Screen

Beim Start von ImmoP erscheint für 2.5 Sekunden ein animierter Splash Screen, welcher unser Logo zeigt.

Danach wird der Benutzer automatisch zum Login-Screen weitergeleitet. Für das Login wird Firebase Authentication verwendet. Die Firebase Authentifizierung nutzt die Industriestandards OAuth 2.0 und OpenID Connect.

Als Identity Provider wird aktuell nur Google angeboten. Dadurch wird ein Google Account für die Anmeldung vorausgesetzt.

In der Zukunft können noch weitere Provider wie Facebook oder Twitter angeboten werden.

Nach der erfolgreichen Anmeldung wird dem Benutzer der Home-Screen angezeigt.

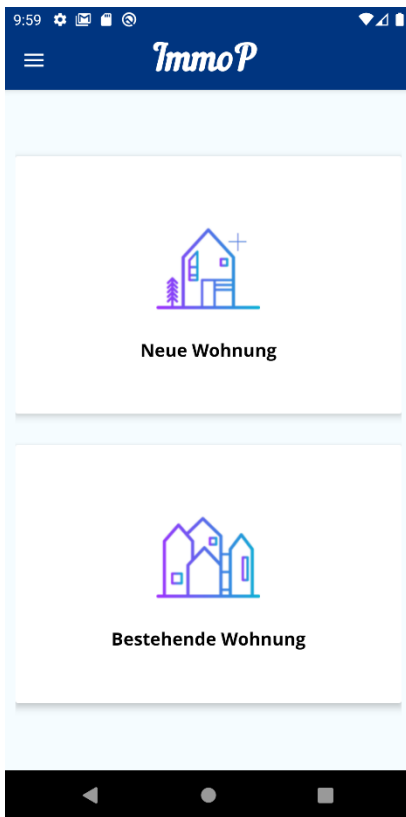


Abbildung 33: Home Screen

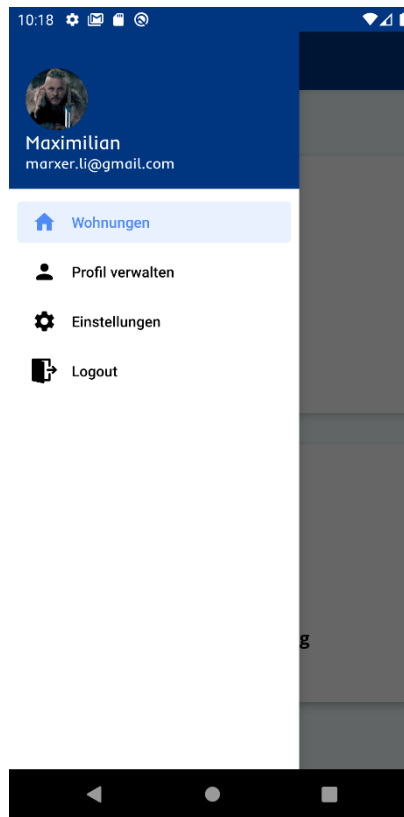


Abbildung 34: Navigation Drawer

Im Home-Screen stehen dem Benutzer mehrere Möglichkeiten zur Auswahl:

1. Der Benutzer möchte eine neue Wohnung erstellen. Dafür kann der Button **Neue Wohnung** angesteuert werden.
2. Vorhandene Wohnungen können über den Button **Bestehende Wohnung** angeschaut werden.
3. Über das Navigation Drawer Menu (oben links in der Abbildung Home Screen) kann der User sich wieder aus der App ausloggen, sein Profil betrachten oder die Einstellungen der Applikation ändern.

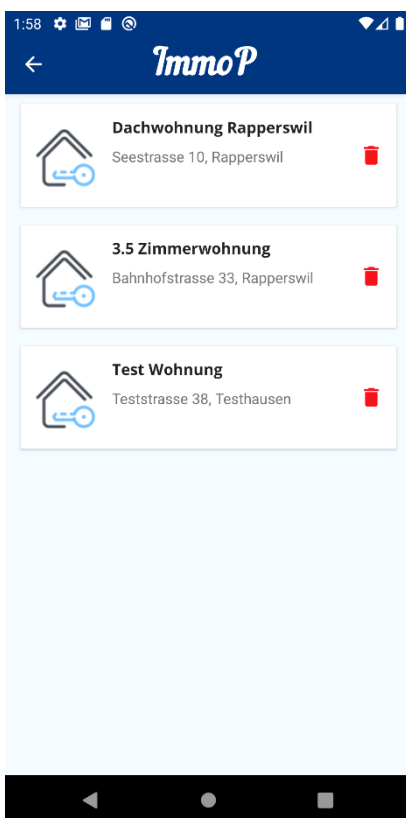


Abbildung 35: Bestehende Wohnungen

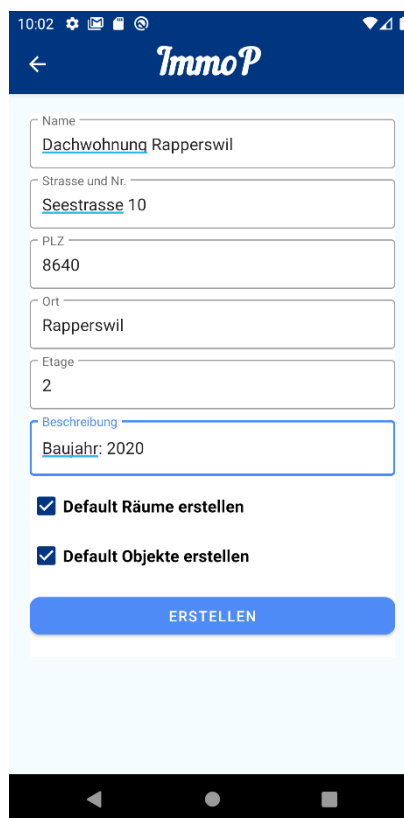


Abbildung 36: Neue Wohnung

Per Klick auf eine bestehende Wohnung gelangt der Benutzer zur Wohnungsübersicht.

Beim Erstellen einer neuen Wohnung stehen dem Benutzer die nachfolgenden Eingabefelder zur Verfügung:

- Name
- Strasse und Nr.
- PLZ
- Ort
- Etage
- Beschreibung

Der Benutzer hat die Möglichkeit, automatisch vordefinierte Räume und Objekte zu erstellen. Damit soll die Usability erhöht werden.

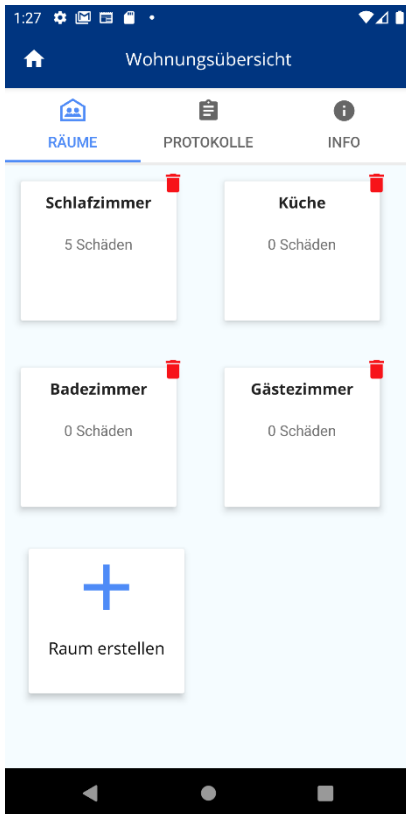


Abbildung 37: Räume

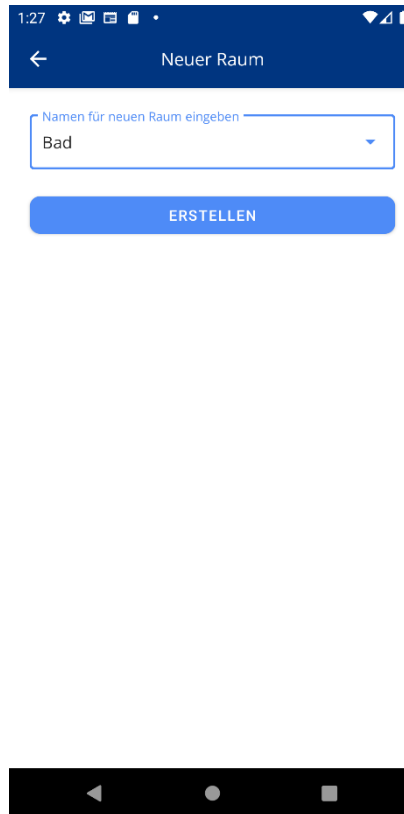
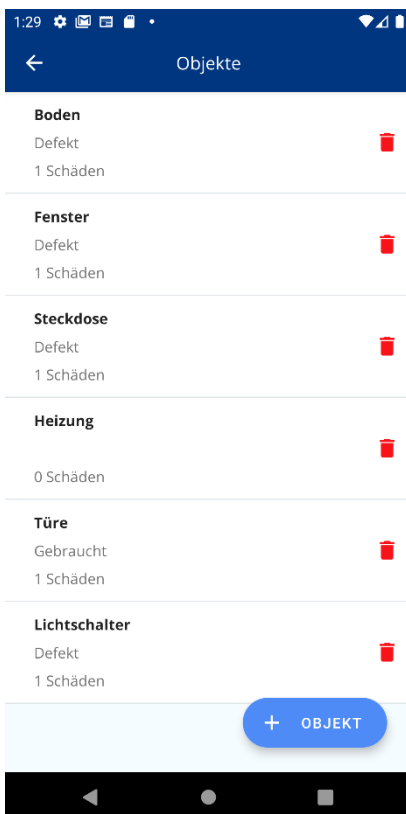


Abbildung 38: Neuer Raum



In der Wohnungsübersicht verwenden wir eine Toolbar und eine Tab-Navigation mit den drei Tabs: Räume, Protokolle und Info.

Standardmässig wird dem Benutzer die Ansicht **Räume** angezeigt.

Räume können gelöscht und neu hinzugefügt werden.

Um einen neuen Raum zu erstellen, muss der Benutzer auf die Karte **Raum erstellen** klicken.

Es öffnet sich eine neue Ansicht mit einer Dropdownbox für Namensvorschläge. Die Vorschläge wurden aus der Protokollanalyse übernommen. Der Benutzer ist dabei frei in der Namenswahl.

Die Ansicht **Objekte** werden alle Einrichtungsobjekte angezeigt. Die Objekte werden dabei in einer Liste dargestellt. Einrichtungsobjekte können gelöscht oder hinzugefügt werden.

Objekte werden mit einem Klick auf den roten Eimer gelöscht.

Bei der Erstellung eines neuen Objektes stehen dem Benutzer folgende Eingabefelder zur Verfügung:

- Name
- Zustand
- Einbaujahr
- Seriennummer



Abbildung 39 Info Ansicht

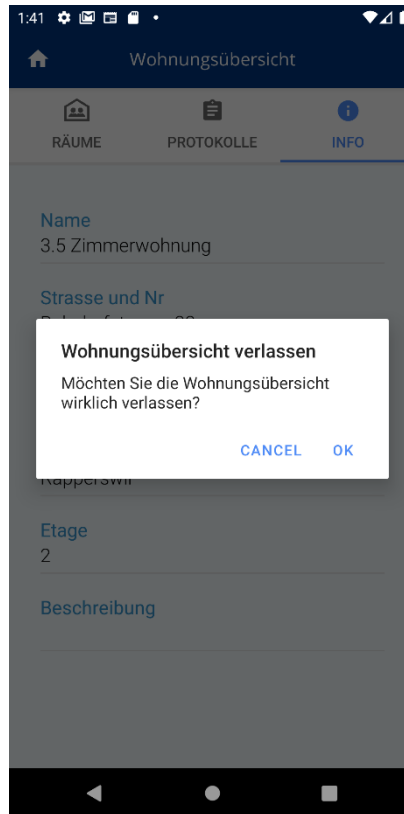


Abbildung 40 Dialog Info

Die Info-Ansicht zeigt alle erfassten Informationen einer Wohnung.

Um das Dashboard einer Wohnung zu verlassen, kann über den Home-Button (oben links Abbildung Info Ansicht) zurücknavigiert werden.

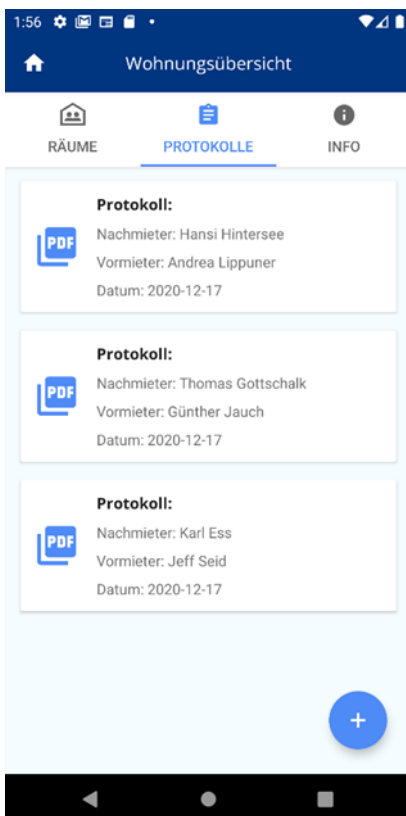


Abbildung 41 Dashboard Protokolle



Abbildung 42 Protokoll erstellen

In der Protokollübersicht werden alle bestehenden Protokolle angezeigt. Pro Protokollelement werden Vor- und Nachmieter angezeigt.

Mit einem Klick auf den «+»-Button kann eine neue Schadensaufnahme erstellt werden. Dabei wird der Benutzer zu Beginn auf die Ansicht **Neues Protokoll** geleitet.

Dem Benutzer stehen die nachfolgenden Eingabefelder zur Verfügung:

- Mieter
- Vormieter
- Zählerstand Heizung
- Zählerstand Warmwasser
- Zählerstand Kaltwasser
- Zählerstand Strom

Durch Selektieren des Buttons **Weiter** wird die Navigation durch die Wohnung gestartet.

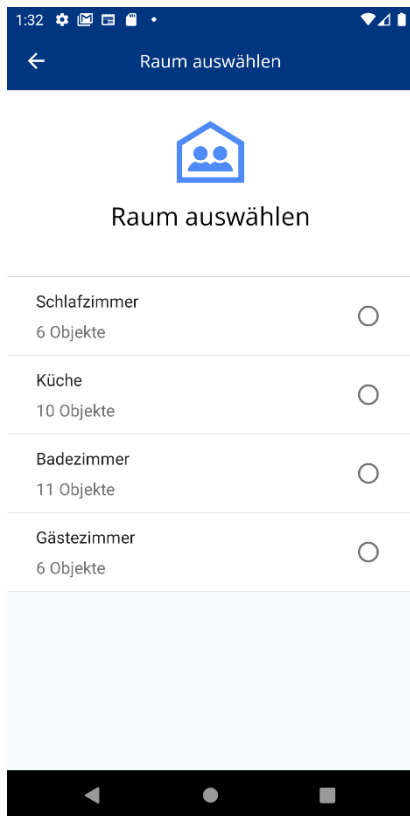


Abbildung 43 Schadensaufnahme Navigation

Bei der Raumauswahl-Ansicht werden dem Benutzer alle Räume der Wohnung mit der Anzahl Objekte angezeigt.

Wenn der Benutzer einen Raum selektiert hat, wird eine neue Ansicht für die Schadensaufnahme eines Einrichtungsobjektes eröffnet.

Sobald der Benutzer durch alle Objekte eines Raumes navigiert ist, gelangt er wieder zurück zur Raumauswahl-Ansicht. Die Liste der Räume verringert sich immer um den jeweiligen davor besuchten Raum.

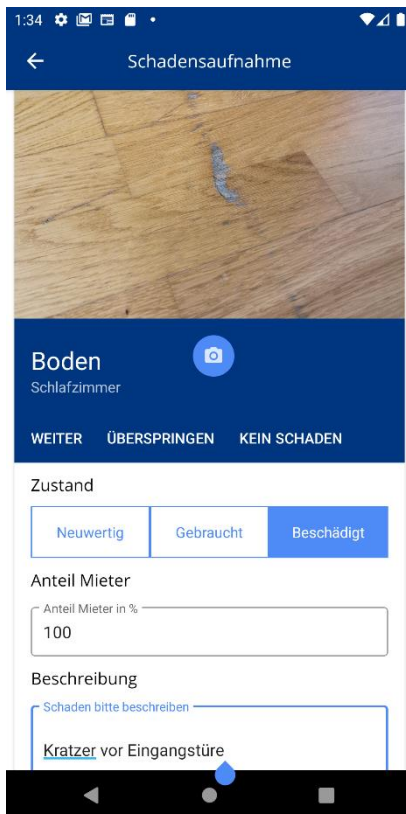


Abbildung 44 Schadensaufnahme

Die Abbildung **Schadensaufnahme** repräsentiert die Ansicht einer Schadensaufnahme für ein Einrichtungsobjekt.

Für eine Schadensaufnahme werden dem Benutzer die nachfolgenden Möglichkeiten gegeben:

- Der blaue Kamera-Button öffnet die Kamera-App des Android-Geräts. Der Benutzer kann dann ein Foto des Schadens aufnehmen. Das erstellte Foto wird oberhalb des Kamera-Buttons angezeigt. Wenn kein Foto für ein Objekt vorhanden ist, wird ein grauer Hintergrund anstelle des Bildes angezeigt.
- Für ein Einrichtungsobjekt kann der Benutzer einen Zustand definieren. Dabei hat er die Auswahl zwischen «neuwertig», «gebraucht» und «beschädigt».
- Weiter kann der prozentuale Anteil des Mieters am Schaden eingetragen werden.
- Die genaue Beschreibung des Schadens erfolgt im Freitextfeld «Beschreibung».

Für die Navigierung zum nächsten Einrichtungsobjekt gibt es drei Möglichkeiten:

- Mit dem Button «Weiter» wird ein Objekt inklusive seiner Schadensaufnahme ins Protokoll aufgenommen.
- Mit dem Button «Überspringen» wird ein Objekt ignoriert und nicht ins Protokoll aufgenommen. Der User wird automatisch zum nächsten Objekt des Raumes weitergeleitet.
- Mit dem Button «Kein Schaden» wird ein Objekt ins Protokoll aufgenommen und als «schadensfrei» klassifiziert.
- Am unteren Ende dieser Ansicht wird die Schadenshistorie angezeigt.



Abbildung 45 Nächstes Objekt

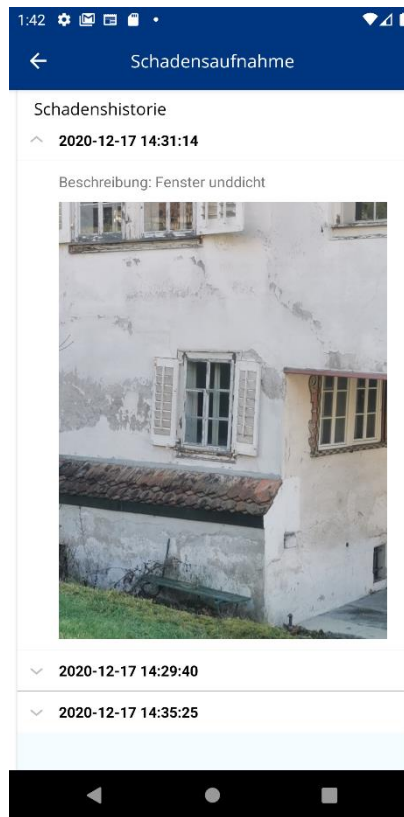


Abbildung 46 Schadenshistorie

Die Schadenshistorie zeigt, nach Datum gruppiert, Schäden an, welche in der Vergangenheit protokolliert wurden. Durch die Auswahl eines Datums werden die Schadenbeschreibung und das zugehörige Bild des Schadens angezeigt. Der Benutzer hat so eine effiziente Möglichkeit, zu prüfen, ob ein Schaden bereits vorhanden war oder nicht.

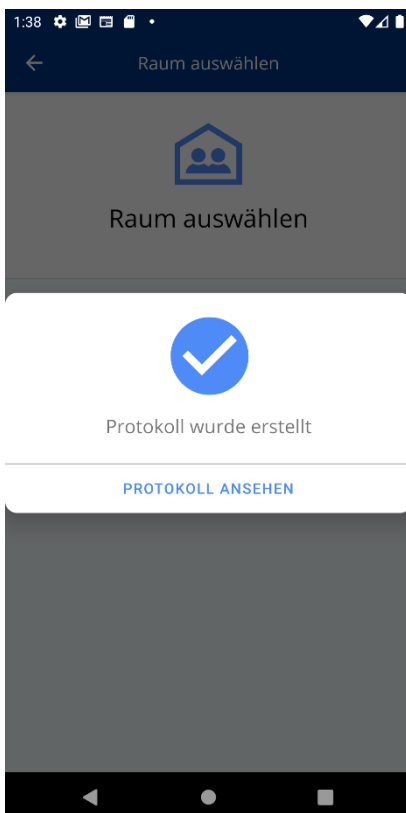


Abbildung 47 Protokoll erstellt Dialog

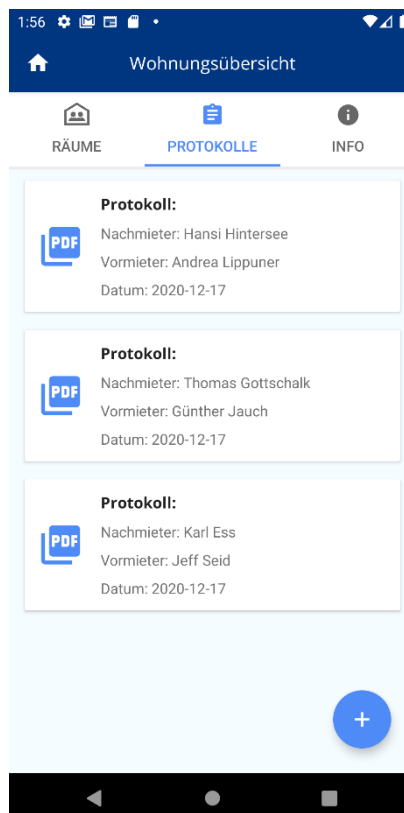


Abbildung 48 Protokollliste

Sobald die Objekte aller Räume durchgearbeitet worden sind, erscheint ein Dialog, welcher den Benutzer über die erfolgreiche Erstellung des Protokolls informiert.

Mit dem Button «Protokoll ansehen» gelangt der Benutzer wieder zur Wohnungsübersicht. Unter den Protokollen wird jetzt das neu erstellte Protokoll angezeigt.

Wenn der Benutzer auf das Protokoll klickt, öffnet sich die Detailansicht des Protokolls.

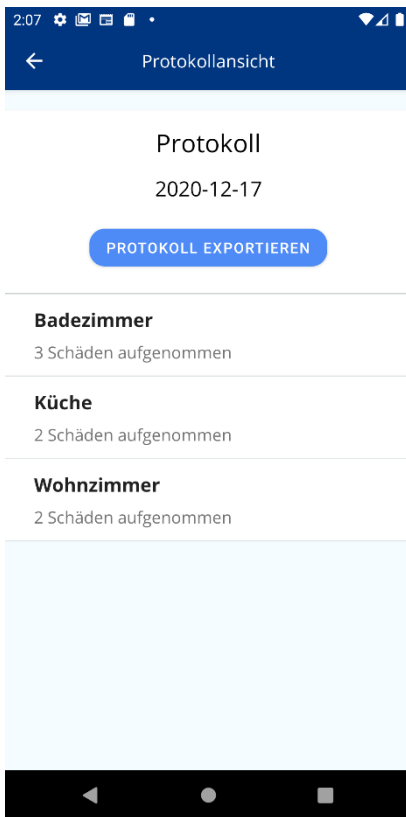


Abbildung 49 Protokoll Detailansicht

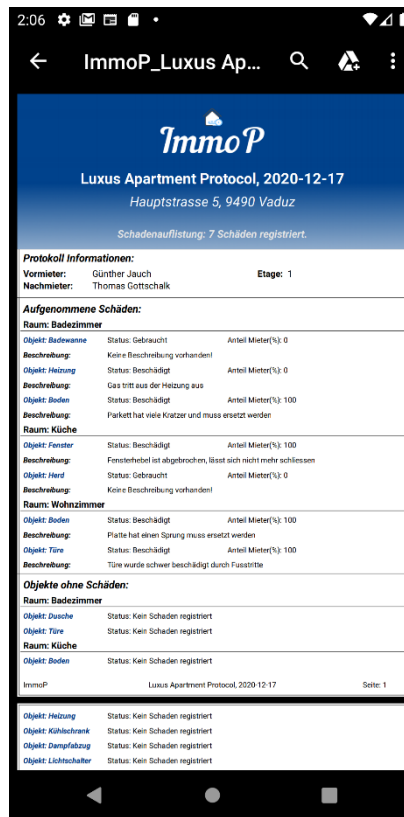


Abbildung 50 PDF-Datei (Export)

In der Detailansicht des Protokolls werden nochmals alle Räume aufgelistet, für welche der Benutzer Schäden protokolliert hat.

Mit dem Button «Protokoll exportieren» wird ein PDF-Dokument erstellt und angezeigt. Für die Ansicht der PDF-Datei wird vom Gerät ein PDF-Reader ausgewählt.

5.2 Review funktionale Anforderungen

UC01: Wohnung verwalten CRUD

Prio: 3**Status:** erledigt (ohne Modifikation)

Der Benutzer kann eine neue Wohnung erstellen und Wohnungen einzeln wieder löschen. Bei der Erstellung einer Wohnung kann der Benutzer die nachfolgenden Informationen angeben:

- Name der Wohnung
- Strasse und Nr.
- Postleitzahl
- Ort
- Etage
- Beschreibung
- Default Räume erstellen ja / nein
- Default Objekte erstellen ja / nein

Die Modifikation einer Wohnung wurde noch nicht umgesetzt. Das Erstellen und Löschen einer Wohnung und die Erledigung der anderen *Use Cases* wurde höher priorisiert.

Beim Erstellen der Wohnung wurden die beiden zusätzlichen Checkboxen «Default Räume erstellen» und «Default Objekte erstellen» eingebaut, damit für den Benutzer vordefinierte Räume und Objekte angelegt werden. Dadurch kann er seine Wohnung schneller konfigurieren.

Die Wohnungen werden dem Benutzer als Liste in Form einer *CardView* angezeigt.

Um eine Wohnung zu löschen, klickt der Benutzer auf das zugehörige Symbol mit dem roten Abfalleimer.

UC02: Raum verwalten CRUD

Prio: 3**Status:** erledigt (ohne Modifikation)

Räume können erstellt und gelöscht, aber nicht modifiziert werden. Um einen Raum zu erstellen, wird dem Benutzer eine Dropdownliste mit den Räumen aus der Protokollanalyse angezeigt. Der Benutzer hat die Möglichkeit, einen eigenen Raum, welcher nicht in der Liste vorhanden ist, zu erstellen.

Räume werden im Gegensatz zu den Wohnungen nicht als vertikale Liste angezeigt. Wir haben uns hier für ein Grid-Layout entschieden, sodass jeweils 2 Räume auf einer Zeile angezeigt werden.

Das Erstellen und Löschen eines Raums und die Implementierung der anderen *Use Cases* wurden höher priorisiert als die Modifikation.

UC03: Objekte verwalten CRUD

Prio: 3**Status:** erledigt (ohne Modifikation)

Für einen Raum können beliebig viele Objekte erstellt und wieder entfernt werden. Bei der Erstellung eines Objektes können die nachfolgenden Daten angegeben werden:

- Name
- Zustand (Neuwertig / Gebrauchte / Beschädigt)
- Einbaujahr
- Seriennummer

Die Objekte werden als einfache *ListView* angezeigt. Um ein Objekt zu löschen, klickt der Benutzer auf das zugehörige Symbol mit dem roten Abfalleimer.

Die Modifikation ist indirekt durch das Löschen und Erstellen bereits gegeben und wurde aus Zeitgründen nicht mehr umgesetzt.

UC04: Schaden erstellen

Prio: 3

Status: erledigt

Der Benutzer kann für einzelne Objekte einen Schaden protokollieren. Für die Protokollierung eines Schadens stehen dem Benutzer die folgenden Möglichkeiten zur Auswahl:

- Zustand des Objekts beschreiben (neuwertig / gebraucht / beschädigt)
- Anteil des Mieters in % angeben
- Beschreibung des Schadens erfassen

Ein erfasster Schaden soll und kann nicht modifiziert werden.

UC05: Schadenaufnahme durchführen

Prio: 3

Status: erledigt

In der Businesslogik wurde eine intelligente Navigation entwickelt, welche den Benutzer durch die verschiedenen Objekte eines Raumes leitet. Zu Beginn wird dem Benutzer eine Liste aller verfügbaren Räume angezeigt. Der Benutzer entscheidet selbst, mit welchem Raum er die Schadenaufnahme beginnen möchte. Aus dem Interview haben wir erfahren, dass bei einer Wohnungsübernahme jedes Einrichtungsobjekt eines Raumes geprüft werden muss. Um sicherzustellen, dass kein Objekt übersehen oder vergessen wird, wurde die intelligente Navigation so konzipiert, dass durch alle Räume und Objekte durchnavigiert werden muss.

Der Benutzer wird gezwungen, bei jedem Objekt eine der drei nachfolgenden Aktionen durchzuführen:

- **Weiter:** Wenn der User auf den Button «Weiter» drückt, werden das aktuelle Einrichtungsobjekt und der erfasste Schaden ins Protokoll aufgenommen und erscheinen somit später in der PDF-Datei.
- **Überspringen:** Der User kann ein Objekt überspringen. Wenn bei der Schadenaufnahme ein Einrichtungsobjekt von der Businesslogik vorgeschlagen wird, welches im Raum nicht existiert, kann dieses übersprungen werden. Das Objekt wird dann nicht ins Protokoll aufgenommen und ist somit später in der PDF-Datei nicht ersichtlich. Vor dem nächsten Rundgang sollte der Benutzer das entsprechende Objekt entfernen.
- **Kein Schaden:** Mit dem Button «Kein Schaden» wird das entsprechende Einrichtungsobjekt ohne Schaden mit ins Protokoll aufgenommen.

Sobald der Benutzer durch alle Objekte des Raumes navigiert wurde, wird er wieder zur *Raumauswahl* weitergeleitet. Die Schadenaufnahme für den aktuellen Raum ist damit beendet und der User entscheidet, mit welchem Raum er als nächstes fortfahren möchte.

UC06: Protokoll generieren**Prio: 3****Status:** erledigt

Sobald der Benutzer sich durch alle Räume einer Wohnung durchgearbeitet hat, erstellt die Businesslogik im Hintergrund das entsprechende Protokoll. Der Benutzer wird über ein Dialogfenster informiert, dass das Protokoll erfolgreich erstellt wurde. Das generierte Protokoll ist im Dashboard der Wohnung einsehbar.

UC07: Foto für einen Schaden hinzufügen**Prio: 3****Status:** erledigt

Der Benutzer hat die Möglichkeit, ein Foto bei der Schadensaufnahme zu erstellen. Das erstellte Foto wird zusammen mit den anderen Informationen in der Firebase-Datenbank abgespeichert und gehört so eindeutig zu einem bestimmten Schaden.

UC08: Schadenshistorie eines Objektes ansehen**Prio: 2****Status:** teilweise erledigt

Bei der Schadensaufnahme hat der Benutzer die Möglichkeit, die Schadenshistorie des Objektes anzuschauen. Die Schadenshistorie zeigt, nach Datum gruppiert, Schäden an, welche in der Vergangenheit protokolliert wurden.

Aus Zeitgründen konnte eine Schadenshistorie in der Detailansicht eines Einrichtungsobjektes nicht implementiert werden.

UC09: Schäden übernehmen**Prio: 1****Status:** noch offen

Der Benutzer muss aktuell noch Schäden manuell aus der Schadenshistorie übernehmen. Leider konnten wir aus Zeitgründen keine Funktionalität mehr dafür entwickeln.

UC10: Wohnungsprotokoll als PDF-Datei exportieren**Prio: 3****Status:** erledigt

Nach einer erfolgreichen Schadensaufnahme kann der Benutzer aufgenommene Daten in Protokollform als PDF-Datei exportieren. Im exportierten Protokoll werden alle erfassten Informationen der Schadensaufnahme aufgezeigt. Für die Struktur der PDF-Datei orientierten wir uns an der Protokollanalyse.

UC11: Registration / User Authentifizierung**Prio: 2****Status:** erledigt

Für die Authentifizierung wurde Firebase verwendet. Firebase verwendet die aktuellen Standards wie OAuth 2.0 und OpenID Connect für die Authentifizierung. Es werden viele verschiedene Identity Provider wie Google, Facebook und Instagram unterstützt. Wir haben uns für Google entschieden, da die meisten Besitzer eines Androids-Geräts bereits einen Google Account besitzen und so keinen neuen Account für unsere Applikation erstellen müssen.

6 Schlussfolgerung

6.1 Ausblick

Die erarbeiteten Erkenntnisse zeigen klar das Potenzial der digitalen Protokollführung. Dabei sind die Anwendungsmöglichkeiten beinahe grenzenlos. Die von den Autoren erstellte Applikation ist noch ausbaufähig. So könnte beispielsweise die individuelle Gestaltung der Protokollinhalte zu einem Mehrwert führen. Eine Verwaltung über das Web und eine Anwendung nicht nur bei Smartphones, sondern auch Tablets, könnte dem Nutzer einen zusätzlichen Mehrwert bringen. Nicht zuletzt wäre die Vermarktung der Applikationen über einen App-Store oder im direkten Verkauf ein interessantes Geschäftsmodell.

6.2 Reflexion

In der Anfangsphase der vorliegenden Arbeit waren die Autoren stark gefordert und oft mit grosser Unsicherheit behaftet. Regelmässige Gespräche mit dem betreuenden Dozenten führten dann immer mehr Licht in die Angelegenheit. Aus einer zähen Anfangsphase entwickelte sich eine teilweise euphorische Produktivphase, in welcher die Ideen gesammelt, gebündelt und internalisiert wurden. Rückblickend darf gesagt werden, dass in der Anfangsphase viel Energie und mehr Zeit als geplant investiert wurde. Diese konnte jedoch während der Entwicklungsphase der Arbeit durch tägliche intensive Gespräche zwischen den Autoren und regelmässigen Feedbacks mit dem begleitenden Dozenten weitgehend wettgemacht werden. So hat beispielsweise die Einarbeitung in das Thema der *Augmented Reality*-Technologien viel Zeit in Anspruch genommen. Die erworbenen Erkenntnisse dieser Technologie konnten in der Arbeit nicht verwertet werden, gaben uns jedoch einen zusätzlichen, wertvollen Einblick in die Welt der Digitalisierung.

Trotz der anfänglichen Startschwierigkeiten und Extrarunden, die die Autoren gehen mussten, ist das Ergebnis aus dieser Arbeit trotz teilweise vorhandener Optimierungsmöglichkeiten im Endprodukt sehr zufriedenstellend und erfüllt die Autoren mit Stolz.

Die Zusammenarbeit zwischen den Autoren und der betreuenden Person war fordernd, fördernd und sehr fruchtbar. Sie wurde vom Autorenteam sehr geschätzt.

7 Literaturverzeichnis

- [1] HSR, «GitLab Hsr,» [Online]. Available: <https://gitlab.dev.ifs.hsr.ch/>.
- [2] Wikipedia, «Portable Document Format,» [Online]. Available: https://de.wikipedia.org/wiki/Portable_Document_Format.
- [3] Android, «Android 10 for Developers,» [Online]. Available: <https://developer.android.com/about/versions/10/highlights>.
- [4] Google, «Firebase,» [Online]. Available: <https://firebase.google.com/>.
- [5] D. M. Kuhrmann, «Rational Unified Process (RUP),» [Online]. Available: <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/Rational-Unified-Process-%28RUP%29>.
- [6] W. Immobilien, «Wuhrmann Immobilien,» [Online]. Available: <https://www.wuhrmann-immobilien.ch/>.
- [7] Google, «ARCore,» 2020. [Online]. Available: <https://developers.google.com/ar/discover/concepts>.
- [8] Viscircle.de, «Viscircle.de,» [Online]. Available: <https://viscircle.de/wie-sie-augmented-reality-apps-mit-der-wikitude-sdk-erstellen-koennen/>.
- [9] Google, «Anchor Clouds,» [Online]. Available: <https://developers.google.com/ar/develop/java/cloud-anchors/overview-android>.
- [10] Google, «GeoAR,» [Online]. Available: <https://developers.google.com/ar/discover/concepts>.
- [11] Wikipedia, «Unified Modeling Language,» [Online]. Available: https://de.wikipedia.org/wiki/Unified_Modeling_Language.
- [12] Wikipedia, «FURPS,» [Online]. Available: <https://de.wikipedia.org/wiki/FURPS>.
- [13] Wikipedia, «SMART,» [Online]. Available: [https://de.wikipedia.org/wiki/SMART_\(Projektmanagement\)](https://de.wikipedia.org/wiki/SMART_(Projektmanagement)).
- [14] GeeksForGeeks, «MVC-Architekturmuster,» [Online]. Available: <https://www.geeksforgeeks.org/mvc-model-view-controller-architecture-pattern-in-android-with-example/>.
- [15] Google, «Firebase Realtime Database,» [Online]. Available: <https://firebase.google.com/docs/database>.
- [16] Wikipedia, «JavaScript Object Notation,» [Online]. Available: [https://de.wikipedia.org/wiki/JavaScript_Object_Notation#:~:text=Die%20JavaScript%20Object%20Notation%20\(JSON,existieren%20in%20allen%20verbreiteten%20Sprachen..](https://de.wikipedia.org/wiki/JavaScript_Object_Notation#:~:text=Die%20JavaScript%20Object%20Notation%20(JSON,existieren%20in%20allen%20verbreiteten%20Sprachen..)
- [17] Google, «Firebase Storage,» [Online]. Available: <https://firebase.google.com/docs/storage>.
- [18] Google, «Material Design,» [Online]. Available: <https://material.io/design>.
- [19] Wikipedia, «Universally Unique Identifier,» [Online]. Available: [https://de.wikipedia.org/wiki/Universally_Unique_Identifier#:~:text=Ein%20Universally%20Unique%20Identifier%20\(UUID,Software%2C%20Registry\)..](https://de.wikipedia.org/wiki/Universally_Unique_Identifier#:~:text=Ein%20Universally%20Unique%20Identifier%20(UUID,Software%2C%20Registry)..)
- [20] IT-Visions.de, «Eager Loading,» [Online]. Available: https://www.it-visions.de/glossar/alle/6217/Eager_Loading.aspx.
- [21] Google, «Firebase API,» [Online]. Available: <https://firebase.google.com/docs/reference>.

-
- [22] Wikipedia, «Uniform Resource Identifier,» [Online]. Available: https://de.wikipedia.org/wiki/Uniform_Resource_Identifier.
- [23] Picasso, «Picasso,» [Online]. Available: <https://square.github.io/picasso/>.
- [24] Android, «PdfDocument,» [Online]. Available: <https://developer.android.com/reference/android/graphics/pdf/PdfDocument>.
- [25] [Online].
- [26] Google, «Android Navigation,» [Online]. Available: <https://developer.android.com/guide/navigation>.

8 Verzeichnisse

8.1 Abkürzungsverzeichnis (Glossar)

Abkürzung	Erklärung
<i>AR</i>	Augmented Reality (erweiterte Realität)
<i>Firebase</i>	Google Entwicklungsplattform für Mobile- und Webanwendungen. Cloud-Service, Datenbank
<i>Git</i>	Versionsverwaltungstool
<i>Identity Provider</i>	zentrales Zugangssystem für Anmeldedienste
<i>JSON</i>	JavaScript Object Notation, Datenformat für einfachen Datenaustausch
<i>SDK</i>	Software Development Kit, Sammlung von Programmierwerkzeugen
<i>Tutorial</i>	schriftliche oder filmische Gebrauchsanleitung
<i>URI</i>	Uniform Resource Identifier, Identifikator einer Ressource
<i>ListView</i>	Material Design Layout, um Elemente als Liste anzuzeigen
<i>CardView</i>	Material Design Layout, um Elemente als Karten anzuzeigen
<i>Fragment</i>	wichtige UI-Komponente von Android-Apps; seit der Version 3.0 (Honeycomb) verfügbar
<i>Drawer-Menu</i>	Das Menü, welches von der Material-Design-Komponente <i>Navigation Drawer</i> verwendet wird.

8.2 Abbildungen

Abbildung 1 Startansicht	4
Abbildung 2 Schadensaufnahme.....	4
Abbildung 3 Protokoll.....	4
Abbildung 4 Plattformen für ARCore.....	9
Abbildung 5 Plattformen für Wikitude SDK.....	10
Abbildung 6 Auswertung Felder in Protokollen	14
Abbildung 7: Unterschrift Protokoll Mieterverband Schweiz	14
Abbildung 8: Ausschnitt Abnahmeprotokoll ImmoScout24.de	15
Abbildung 9: Ausschnitt Abnahmeprotokoll Mieterverband Schweiz.....	15
Abbildung 10:Ausschnitt Protokoll mietrecht.ch.....	16
Abbildung 11 Vorhandene Räume in Protokollen	16
Abbildung 12 Vorhandene Einrichtungsobjekte in Protokollen	17
Abbildung 13 ImmoP Menü.....	20
Abbildung 14 ImmoP Wohnungsübersicht	20
Abbildung 15 Wohnung erstellen.....	21
Abbildung 16 Dashboard Protokolle	21
Abbildung 17 Dashboard Räume.....	21
Abbildung 18 Objekt erstellen	22
Abbildung 19 Objektübersicht	22
Abbildung 20 Protokollangaben.....	24
Abbildung 21 Navigation Raumauswahl.....	24
Abbildung 22 Entwurf Schadensaufnahme	26
Abbildung 23 ImmoP Use Case Diagramm.....	29
Abbildung 24 ImmoP Domainmodel	34
Abbildung 25 ImmoP Deployment Diagramm.....	35
Abbildung 26 Schichtenmodell	36
Abbildung 27 Package Diagramm	37
Abbildung 28 Teil Hauptansicht.....	40
Abbildung 29 Teil Wohnungsübersicht.....	40
Abbildung 30 Beispiel UUID.....	41
Abbildung 31: Splash Screen	47
Abbildung 32: Login Screen	47
Abbildung 33: Home Screen	48
Abbildung 34: Navigation Drawer	48
Abbildung 35: Bestehende Wohnungen.....	48
Abbildung 36: Neue Wohnung	48
Abbildung 37: Räume	49
Abbildung 38: Neuer Raum.....	49
Abbildung 39 Info Ansicht	50
Abbildung 40 Dialog Info.....	50
Abbildung 41 Dashboard Protokolle	50
Abbildung 42 Protokoll erstellen	50
Abbildung 43 Schadensaufnahme Navigation.....	51
Abbildung 44 Schadensaufnahme.....	52
Abbildung 45 Nächstes Objekt	53

Abbildung 46 Schadenshistorie	53
Abbildung 47 Protokoll erstellt Dialog	53
Abbildung 48 Protokollliste.....	53
Abbildung 49 Protokoll Detailansicht	54
Abbildung 50 PDF-Datei (Export)	54
Abbildung 51 Zeitauswertung Detail	66
Abbildung 52 Zeitauswertung Projekt-Sprints.....	66
Abbildung 53 Zeitauswertung (Daten) Kevin Schweizerhof.....	66
Abbildung 54 Zeitauswertung Kevin Schweizerhof	67
Abbildung 55 Zeitauswertung (Daten) Maximilian Marxer	67
Abbildung 56 Zeitauswertung Maximilian Marxer	67

8.3 Tabellen

Tabelle 1 Basis Protokollanalyse.....	13
Tabelle 2 Technologien	37

9 Erklärung zur Urheberschaft

Erklärung Wir erklären hiermit, dass wir die vorliegende Arbeit ohne Hilfe Dritter angefertigt habe. Wir habe nur die Hilfsmittel benutzt, die wir angegeben haben. Gedanken, die wir aus fremden Quellen direkt oder indirekt übernommen habe, sind kenntlich gemacht. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort/Datum Rapperswil, 18. Dezember 2020

Unterschrift



Kevin Schweizerhof



Maximilian Marxer

Anhang

I Benutzerhandbuch

Installationsanleitung:

Die Installationsanleitung für Android Smartphones ist als separate Anlage verfügbar.

II Projektplan

Der Projektplan ist als PDF-Datei als separate Anlage verfügbar.

III ImmoP Protokoll Beispiel

Ein Beispiel Protokoll der ImmoP-Applikation ist als separate Anlage einsehbar.

IV Protokoll Wuhrmann Immobilien

Das Wohnungsübernahmeprotokoll des Interviewpartners wurde eingescannt und ist als separate Anlage verfügbar.

V Leitfadeninterview

Das für die Analyse durchgeführte Leitfadeninterview ist als PDF-Datei-Anlage einsehbar.

VI Testauswertung Protokoll

Das Testprotokoll ist als PDF-Datei-Anlage einsehbar.

VII Zeitauswertung

Zeitauswertung nach Sprints (gesamtes Projekt):

Nachfolgende Grafiken zeigen die Auswertungen der geplanten und benötigten Zeiten während der einzelnen Sprints in diesem Projekt.

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Zeit geplant (h)	7,00	60,00	61,00	81,00	65,00	79,50	61,00	69,00
Zeit effektiv (h)	7,00	60,00	61,50	73,50	65,00	77,50	73,50	70,00
Zeit geplant (h)	483,50							
Zeit effektiv (h)	488,00							

Abbildung 51 Zeitauswertung Detail

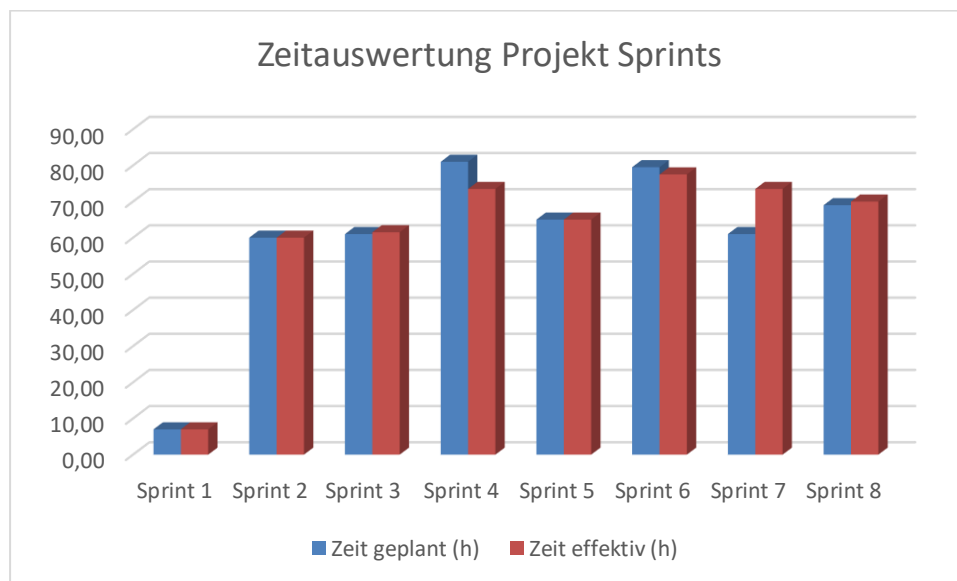


Abbildung 52 Zeitauswertung Projekt-Sprints

Zeitauswertung Teammitglied Kevin Schweizerhof:

Nachfolgende Grafiken zeigen die Zeitauswertung vom Teammitglied Kevin Schweizerhof.

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Zeit geplant	3,50	32,00	29,50	41,00	34,00	34,50	28,50	33,00
Zeit effektiv	3,50	32,50	30,50	38,00	33,50	35,00	35,50	36,00
Zeit geplant (h)	236,00							
Zeit effektiv (h)	244,50							

Abbildung 53 Zeitauswertung (Daten) Kevin Schweizerhof

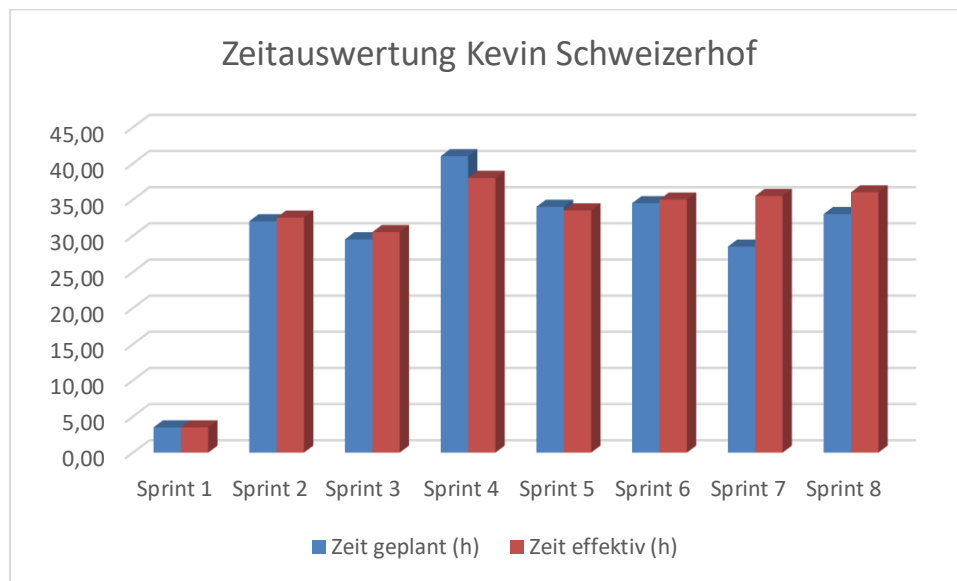


Abbildung 54 Zeitauswertung Kevin Schweizerhof

Zeitauswertung Teammitglied Maximilian Marxer:

Nachfolgende Grafiken zeigen die Zeitauswertung vom Teammitglied Maximilian Marxer.

	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5	Sprint 6	Sprint 7	Sprint 8
Zeit geplant	3,50	28,00	31,50	40,00	31,00	45,00	32,50	36,00
Zeit effektiv	3,50	27,50	31,00	35,50	31,50	42,50	38,00	34,00
Zeit geplant (h)	247,50							
Zeit effektiv (h)	243,50							

Abbildung 55 Zeitauswertung (Daten) Maximilian Marxer

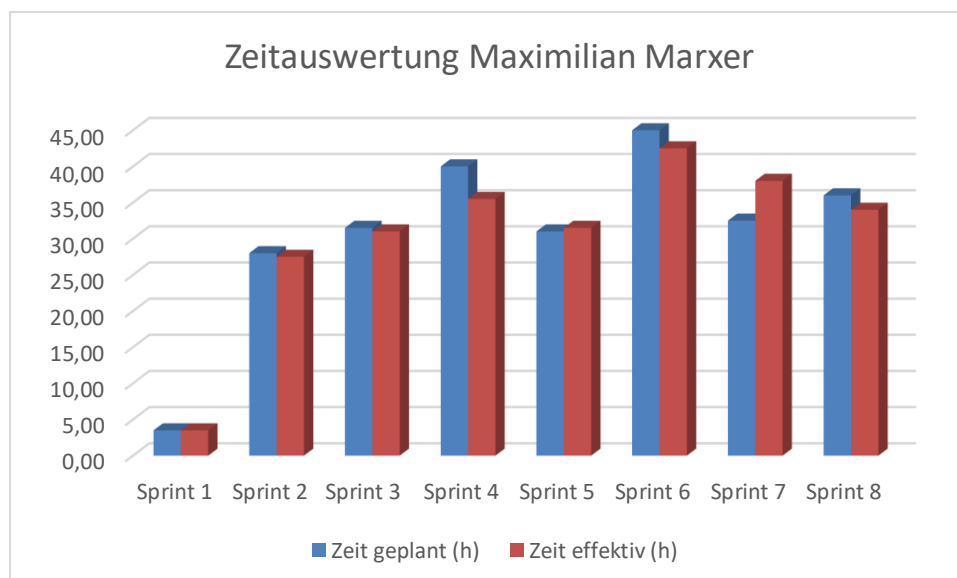


Abbildung 56 Zeitauswertung Maximilian Marxer