

Software Engineering Projekt "Dopplereffekt"

7. April 2015

Autoren: Danilo Santagata, Samuel Loepfe
Abgabetermin: 15.04.2015
Dozent: Ulrich Hauser

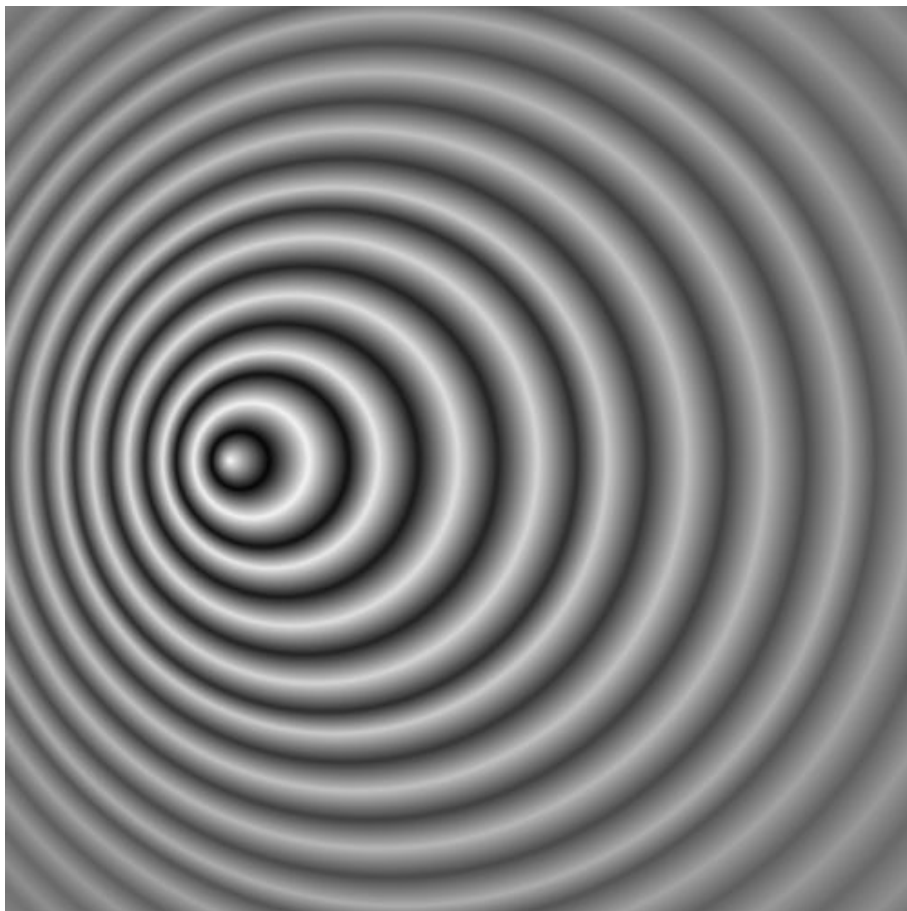


Abbildung 1: Titelbild

Inhaltsverzeichnis

1	Kurzeinleitung	4
2	Team	4
2.1	Aufgaben	4
3	Ausgangssituation	4
3.1	Printscreen Warnungen	5
4	Aufgabe	6
4.1	Anforderungen	6
4.2	Prozess	7
5	Verwendete Programme	8
5.1	Versionsverwaltung	8
5.2	Entwicklungsumgebung	8
5.3	Textsatzprogramm LateX	8
5.4	Editor Brackets	8
5.5	Office Tool VISIO	8
5.6	Office Tool PROJECT	8
6	Use Case	9
7	Sprint/Backlog	10
8	Meilensteine	12
8.1	M1	12
8.1.1	Persönliches	12
8.2	M2	12
8.2.1	Persönliches	12
8.3	M3	13
8.3.1	Persönliches	13
8.4	M4	13
8.4.1	Persönliches	13
8.5	M5	14
8.5.1	Persönliches	14
8.6	M6	14
8.6.1	Persönliches	14
9	Tests	15
9.1	Test 1	15
9.2	Test 2	15
9.3	Test 3	15
9.4	Test 4	16
9.5	Test 5	16
9.6	Test 6	16
9.7	Test 7	16

9.8	Test 8	17
9.9	Test 9	17
9.10	Test 10	17
9.11	Test 11	17
9.12	Test 12	18
9.13	Test 13	18
9.14	Test 13.2	18
10	Gelerntes	19
10.1	Reflexion Programmierung	19
11	Journal	20
12	Glossar	21
13	Anhang	22
13.1	Terminplan	22
13.2	Journal	23
13.3	Klassendiagramm	24
13.4	Sequenzdiagramm	25
14	Daten-CD	26

Abbildungsverzeichnis

1	Titelbild	1
2	Warnung1	5
3	Warnung2	5
4	Scrum Prozess	7
5	Use Case	9
6	Backlog 1	10
7	Backlog 2	10
8	Backlog 3	11
9	Termin Plan	22
10	Journal	23
11	Klassendiagramm	24
12	Sequenzdiagramm	25

1 Kurzeinleitung

Diese Dokumentation soll die Entstehung eines Software Projektes abbilden: Gefordert sind:

1. Dokumentation (enthält Prozessfindung, Entstehung der Applikation, Versionskontrolle, Journal usw.)
2. Code einer Anwendung (Dokumentierter Code)
3. Die persönlichen Erfahrungswerte der Autorenschaft, bezüglich ihrem Projekt und der Dokumentation.

2 Team

Danilo Santagata	Head of Software
Samuel Loepfe	Head of Documentation

2.1 Aufgaben

- Der Software sind prinzipiell die Programmierung und die Dokumentation des Codes unterstellt.
- Der Dokumentation als solches sind die wissenschaftliche Festhaltung des gesamten Projektes, die Zeitgestaltung und Einhaltung, sowie die Zielerfassung und das Erreichen derselben untergeben.

3 Ausgangssituation

Das Software Projekt soll dazu dienen, eine Software zu erstellen und gleichzeitig die Entstehung derselben sauber zu dokumentieren. Wir haben uns für ein Projekt entschieden, welches wir schon früher begonnen hatten, jedoch für sehr passend halten. Die Grundidee ist eine Android-Applikation, welche selbstständig ein PDF-File der Kapo St.Gallen von deren Homepage herunterlädt, die Adressen daraus extrahiert und mit den Geo-Locations des Nutzers abgleicht. Sollte der Nutzer in der Nähe einer dieser Adressen sein, bekommt er eine Meldung zB. (*Achtung, Sie befinden sich in der Nähe einer Geschwindigkeitskontrolle!*), welche auch auf einer allenfalls gepairten Smartwatch angezeigt wird. Nachfolgend werden zwei Ausschnitte solcher Warnungen gezeigt.

3.1 Printscreen Warnungen

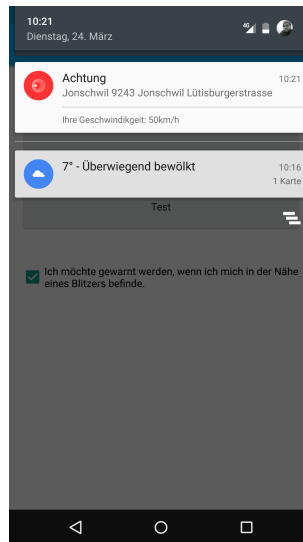


Abbildung 2: Warnung1

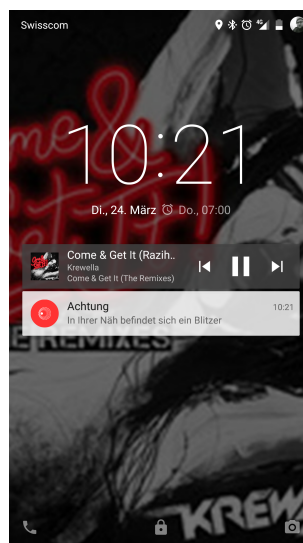


Abbildung 3: Warnung2

4 Aufgabe

Die Aufgabe besteht aus 2 Hauptaspekten: Zum Einen, ein Softwareprojekt im Bereich Engineering, auf welchem das Hauptaugenmerk liegt und zum Anderen, eine möglichst lauffähige Software zu entwickeln. Untergruppen sind:

- Anforderungen
- Prozess
- Versionsverwaltung
- Verschiedene Tests

4.1 Anforderungen

Die Applikation ist fähig, folgende Anforderungen zu erfüllen:

1. Interaktives GUI (User Interface)
2. PDF auf Device herunterladen
3. PDF Inhalte in einen String umwandeln
4. PDF Inhalt auf Device anzeigen
5. GEO-Positionen der Adressen in DB mit HTTP speichern
6. GPS Dialog mit Antwort (soll GPS aktiviert werden?)
7. GPS Daten erhalten
8. Thread für GPS Location erstellen
9. Notification für Device und gepairte Devices erstellen
10. Notification für Device und gepairte Devices ausgeben
11. GPS Service einzeln beenden
12. Simple Navigation durch App ermöglichen (beenden usw.)
13. Warnung ausgeben, falls Geo Position mit Adresse aus PDF übereinstimmt

4.2 Prozess

Wir haben uns für den Scrum-Prozess entschieden, welcher aufgrund des iterativen Charakters des Projektes, die wöchentlichen 'Sitzungen', optimal schien. Allerdings mussten wir einzelne Aspekte etwas anpassen, so ist zum Beispiel die Sprintdauer nicht wie in den meisten Fällen 30 sondern nur 7 Tage lang. Dies lässt sich allerdings auf den stark verkürzten Zeitraum zurückführen. Die entsprechenden Rollen wurden wie folgt vergeben:

- Scrum Master » Danilo Santagata
- Management » Samuel Loepfe
- User, Entwickler, Product Owner » Danilo Santagata und Samuel Loepfe
- Customer » Rijad Zuzo

In den vorgesehenen 2 Lektionen pro Woche fanden die Stand-up Meetings statt, welche allerdings durch die interaktive To Do Liste auf Trello.com an Bedeutung verloren. Trotzdem wurde jede Woche ein Sitzungsprotokoll geschrieben, in welchem die Pendenzen, erledigte sowie künftige Aufgaben und Termine festgehalten wurden. Da wir allerdings einen wesentlichen Teil der Applikation in der Freizeit erstellt haben, wurde die Kontrolle der einzelnen User Stories schwierig, da man nicht nachträglich die Tasks verteilen kann. Wir haben uns deshalb so vor allem mit den Arbeiten beschäftigt, welche noch nicht implementiert waren. Dazu zählten vor allem die Geo-Lokalisation und Überprüfung der Nutzerdaten.

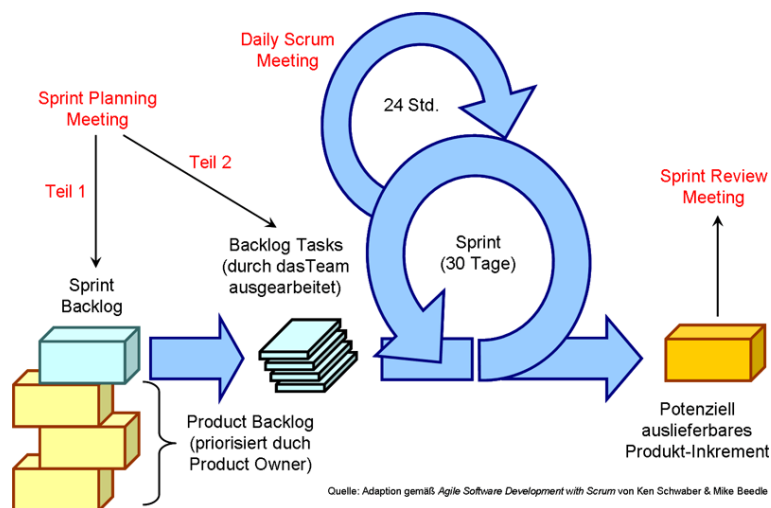


Abbildung 4: Scrum Prozess

5 Verwendete Programme

5.1 Versionsverwaltung

Als Kontrolle, und um die Möglichkeit zu haben, auf einen früheren Projektstand zu wechseln, braucht es eine Versionsverwaltung.

Als Hauptprogramme stehen GIT oder SVN zur Verfügung. Da wir schon beim Roboterprojekt der NTB mit dem Explorerplugin TortoiseSVN gearbeitet hatten, war für uns die Entscheidung naheliegend, auf Bewährtes zu setzen. Als Datenpool für die serverseitige Speicherung benutzten wir Assembla.com, ein kostenloses Web-Tool, welches 5 GB Speicherplatz zur Verfügung stellt. Damit ist stets gewährleistet, dass nach einem erfolgreichen Commit/Update die aktuellste Version verfügbar ist.

5.2 Entwicklungsumgebung

Die prädestinierte Umgebung für die Androidprogrammierung ist das hauseigene Android Studio. Damit lässt sich einfach und schnell vieles bewerkstelligen. Der Emulator ermöglicht eine zeitgleiche Vorschau der Applikation, ohne dass Treiber für Endgeräte installiert werden müssten. Mit der Wahl des API Levels lässt sich einstellen, für welche Androidversion die App konzipiert werden soll.

5.3 Textsatzprogramm LaTeX

Wegen Unverträglichkeit gegenüber herkömmlichen Schreibeditoren seitens der Autorenschaft, wurde bewusst nur auf LaTeX gesetzt. Somit war jederzeit eine saubere Darstellung der einzelnen Abschnitte möglich und jeder Eintrag konnte beliebig editiert, redigiert oder gelöscht werden, ohne Einfluss auf das Gesamtwerk zu haben. Mit den einzelnen Makros zur Texteingabe oder Bildeinbindung war ein flüssiges, konstruktives Arbeiten möglich.

5.4 Editor Brackets

Damit wurde das PHP File geschrieben. Dieser Editor eignet sich besonders für HTML, JavaScript, PHP und andere Websprachen. Er ermöglicht eine direkte Syntaxhervorhebung und eine Live-Vorschau im Chrome-Browser.

5.5 Office Tool VISIO

Mit diesem Programm werden simple Diagramme einfach erstellt. Es bietet eine Anzahl vordefinierter Symbole und Abläufe. Wir haben es für das Use Case Diagramm verwendet.

5.6 Office Tool PROJECT

Dieses Tool ermöglicht eine absolut unkomplizierte Art einen Terminplan zu zeichnen. Es wird automatisch die entsprechende Ansicht generiert, welche vom Datum und der Dauer erfordert wird.

6 Use Case

Unsere Applikation soll gemäss dem folgenden Use-Case arbeiten:

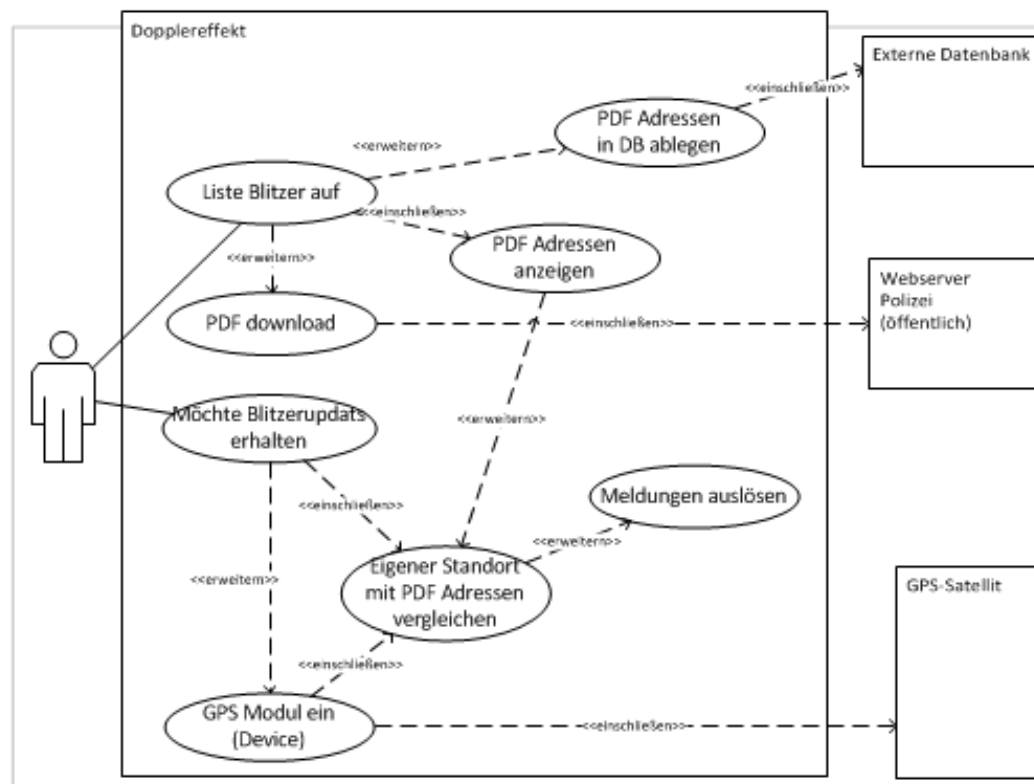


Abbildung 5: Use Case

7 Sprint/Backlog

Für den gesamten Sprint sowie für den Product Backlog haben wir Trello.com verwendet. Dieses Online-Tool ermöglicht eine Interaktive TO-DO Liste, in welcher alle Developer Tasks erstellen, jemandem zuweisen, sowie die Tasks erledigen und als erledigt markieren können. Damit haben wir einen Weg gefunden auch ohne Daily Sprint miteinander zu kontrollieren, wer was wann macht. So ist auch immer jeder genau über die aktuellen Aufgaben aller informiert.

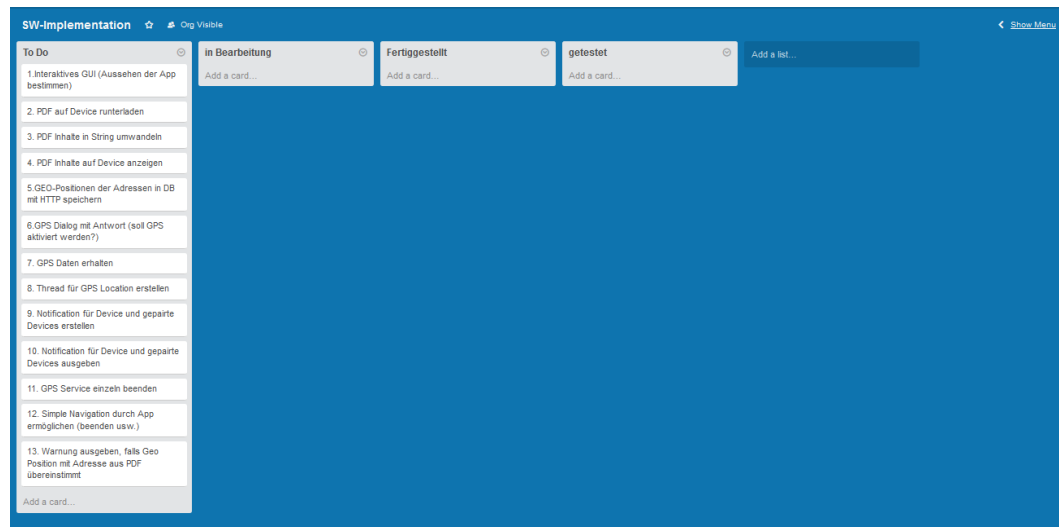


Abbildung 6: Backlog 1

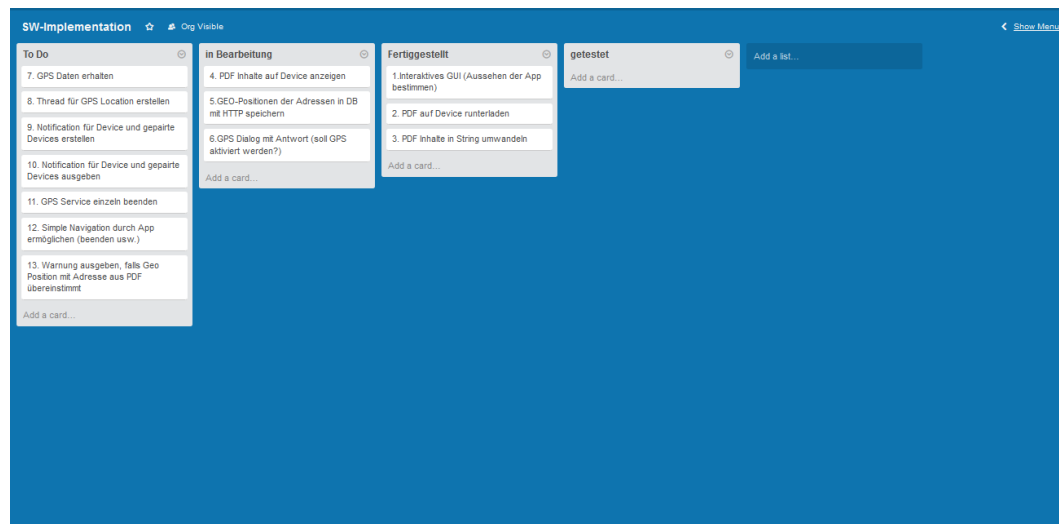


Abbildung 7: Backlog 2

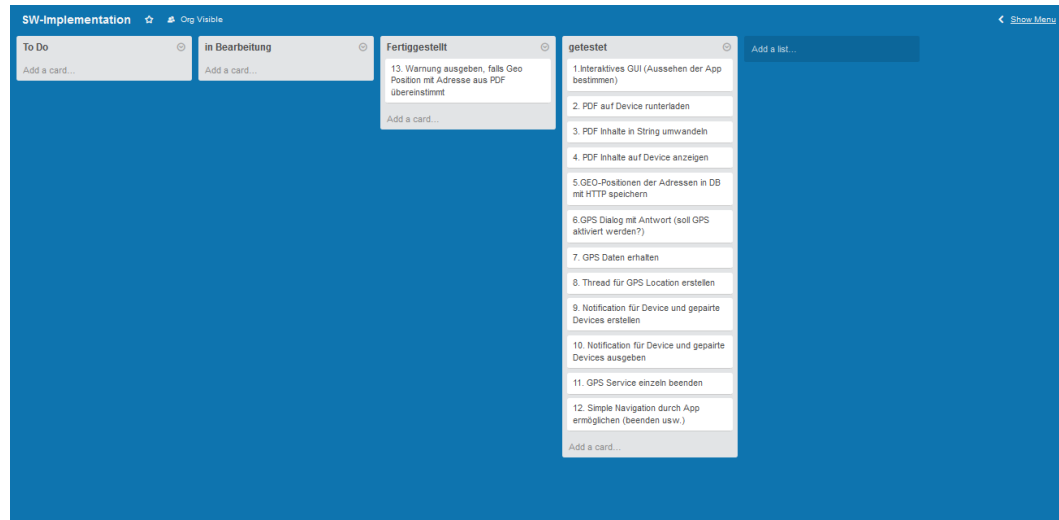


Abbildung 8: Backlog 3

Abhängig von diesen TO-DO Listen konnten wir ermitteln, was im Moment gerade erledigt wurde, was noch ausstand und welche Funktionen bereits getestet wurden. Dies hat uns eine gute Grundlage geboten, um auch alleine speditiv und fokussiert arbeiten zu können. Dank dieser Methode werden auch die Gefahren kleiner, etwas zu vergessen oder falsch zu priorisieren. Diese Prioritäten wurden in unserem Product Backlog festgelegt. Nur in diesem Backlog durften nachträglich irgendwelche Änderungen vorgenommen werden.

8 Meilensteine

Die definierte Zeitspanne eines Sprints beträgt 7 Tage. Dies aufgrund der kürzeren Zeitspanne unseres Projektes. Davon sind jeweils 2 Tage für die Tests und allfällige Anpassungen reserviert. Als Abschluss eines Sprints gilt der Meilenstein, an dem die genannten Anforderungen implementiert und erfüllt sein müssen.

8.1 M1

Der Erste Meilenstein beinhaltet die Anforderungs-ID 1: Die komplette Erstellung eines GUI's, ohne Funktionen zu implementieren. Quasi eine Rohversion der Applikation ohne Verknüpfung der Elemente.

8.1.1 Persönliches

Als es um die Aufgabenverteilung ging, hatten wir kleinere Differenzen. Da dies unsere erste Dokumentation einer Softwareentwicklung war, waren wir uns nicht ganz sicher, wie wir vorgehen wollten. Da wir jedoch genau wussten, wie unsere Applikation auszusehen hatte und auch die Funktionen bekannt waren, machten wir uns als Erstes an das GUI. Android Studio bietet hierfür einen grafischen Editor. Damit lassen sich sämtliche Elemente einfach positionieren und mit den einzelnen Labels beschriften. So konnten wir den Grundstein für unser Projekt legen.

8.2 M2

Als zweiter Meilenstein wurde die Fertigstellung der Anforderungen 6,9 und 10 festgelegt. Darunter gehören die Notifications erstellen, und der GPS-Dialog.

8.2.1 Persönliches

Wir waren sehr an der Funktion *Notification* interessiert. Da wir über eine Smartwatch verfügten, wollten wir unbedingt die Warnung auch über diese anzeigen können. Dies wurde uns von Google noch erleichtert, indem die Smartwatch automatisch angesprochen wurde, sobald sie *gepairt* worden war. Dieses Problem war somit erledigt. Die Checkbox für den GPS-Dialog liess sich einfach implementieren, die verknüpfte Methode wollte jedoch gründlich gesucht werden. Die Anpassung nahm noch etwas Zeit in Anspruch, es blieb jedoch im Rahmen.

8.3 M3

M3 beinhaltet die ID's 2,3 und 4, namentlich die Implementierung der PDF-Funktionen wie Download, Verarbeitung und Anzeige.

8.3.1 Persönliches

PDF lassen sich sehr einfach mit einer vorgefertigten Google-Methode herunterladen. Dazu muss nur ein URL übergeben werden. Damit wir den Inhalt der PDF-Files auslesen konnten, mussten wir eine eigene Methode schreiben, welche aus dem Inhalt einen grossen, zusammenhängenden String generiert. Wir haben verschiedene Methoden implementiert um mit dem PDF-File zu arbeiten. Zum Beispiel kann eine Methode die Anzahl Adressen angeben, dies aufgrund von Schlüsselwörter im String. Eine Andere generiert ein Array aus allen Adressen um sie später mit dem GPS-Signal des Nutzers abzugleichen. Damit die aktuellen Adressen angezeigt werden können, wurde eine Liste generiert, in welche die Adressen mithilfe eines *ListAdapter* geschrieben wurden.

8.4 M4

Als Meilenstein 4 wurde nur die ID 5 definiert. Dies aufgrund der erhöhten Komplexität der Implementierung.

8.4.1 Persönliches

Um einen umfassenden Überblick von den gesammelten Adressen zu bekommen, entschieden wir uns, diese in eine externe Datenbank zu speichern. Dort sind sie jederzeit abrufbar und ermöglichen uns, ein eventuelles Muster von *Hotspots* zu erkennen.

8.5 M5

Der fünfte Meilenstein widmet sich den Anforderungen 7,8 und 11. Hier werden die GPS-Funktionen erstellt und eingebunden. Dazu gehört die Abfrage der Nutzerposition, die Erstellung eines eigenen Threads für den GPS-Service und die Möglichkeit diesen Service einzeln zu beenden.

8.5.1 Persönliches

Mit diesem Meilenstein hatten wir zu kämpfen, da die Beendigung einer App nicht zwangsläufig die Beendigung eines Services bedeutet. Damit man das GPS-Signal ausschalten kann, ohne dass die App geschlossen werden muss, kam ein Thread zum Tragen. Dieser kann nun mit der Checkbox einfach gestartet oder beendet werden.

8.6 M6

Zum Schluss werden im Meilenstein 6 die ID's 12 und 13 festgelegt. Diese regeln die In-App Navigation, sowie die Warnung, welche auch auf gepairten Geräten ausgegeben wird.

8.6.1 Persönliches

Um ein interaktives Erlebnis mit unserer App gewährleisten zu können, haben wir einen etwas speziellen Test durchgeführt. Unser *Customer* Rijad Zuzo musste uns Informationen aus der App angeben können, ohne dass wir ihm diese erklärt hätten. Damit war für uns hinreichend gezeigt, dass man mit der App auch ohne Entwicklerwissen gut umgehen kann.

9 Tests

Für jede Anforderung muss gemäss Definition ein Test bestehen. Da wir eine Applikation entwickeln, welche stark auf Interaktivität ausgelegt ist, macht es keinen Sinn, eigene Testprogramme zu schreiben. Viel wichtiger sind sogenannte angewandte Tests: Im Voraus wird definiert welche Art wie geprüft, und welche Reaktion erwartet wird. Aufgrund der Ergebnisse wird der Test als erfüllt oder nicht erfüllt gewertet.

9.1 Test 1

Anforderung	Erfüllungskriterium
ID 1: Interaktives GUI	Oberfläche mit Button und Check-Box
Testablauf	Erwartete Reaktion
Sichtkontrolle	Alle Elemente sind vorhanden und es sieht ansprechend aus
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 01.03.2015

9.2 Test 2

Anforderung	Erfüllungskriterium
ID 2:PDF Download	Verbindung zu URL wird aufgebaut und PDF lokal gespeichert
Testablauf	Erwartete Reaktion
Testbutton wird gedrückt	PDF wird heruntergeladen
Tatsächliche Reaktion	Test erfüllt?
Lädt PDF	JA 17.03.2015

9.3 Test 3

Anforderung	Erfüllungskriterium
ID 3: PDF in String wandeln	Ein grosser String wird generiert
Testablauf	Erwartete Reaktion
Ausgabe in Methode	komplettes PDF als String dargestellt
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 17.03.2015

9.4 Test 4

Anforderung	Erfüllungskriterium
ID 4: PDF Inhalt anzeigen	Zeigt vollständige Adressen in Liste an
Testablauf	Erwartete Reaktion
Testbutton wird gedrückt	saubere Liste mit Adressen wird angezeigt
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 17.03.2015

9.5 Test 5

Anforderung	Erfüllungskriterium
ID 5: Adressen in DB speichern	Adressen sind in DB ersichtlich
Testablauf	Erwartete Reaktion
Testbutton wird gedrückt	Adressen werden vollständig in DB gespeichert
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 19.03.2015

9.6 Test 6

Anforderung	Erfüllungskriterium
ID 6: GPS Dialog anwenden	Durch drücken von Check-Box startet Dialog, durch Bestätigung wird GPS gestartet
Testablauf	Erwartete Reaktion
Betätigen Check-Box	Dialog öffnet, GPS startet
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 09.03.2015

9.7 Test 7

Anforderung	Erfüllungskriterium
ID 7: GPS Daten von Nutzer einlesen	location wird angezeigt
Testablauf	Erwartete Reaktion
Testbutton drücken	Android.locationlistener gibt daten
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 01.04.2015

9.8 Test 8

Anforderung	Erfüllungskriterium
ID 8: GPS-Thread erstellen	Thread existiert
Testablauf	Erwartete Reaktion
Testbutton drücken	Thread schreibt Location in jedem Durchgang
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 01.04.2015

9.9 Test 9

Anforderung	Erfüllungskriterium
ID 9: Notification erstellen	Code existiert
Testablauf	Erwartete Reaktion
Sichtkontrolle Code	Methode existiert
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 09.03.2015

9.10 Test 10

Anforderung	Erfüllungskriterium
ID 10: Notifications ausgeben	Meldungen werden auf Device und z.B. SmartWatch angezeigt
Testablauf	Erwartete Reaktion
Testbutton drücken	Notification werden angezeigt und durch Vibration signalisiert
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 09.03.2015

9.11 Test 11

Anforderung	Erfüllungskriterium
ID 11: GPS Service beenden	Nur GPS-Service wird beendet, nicht ganze App!
Testablauf	Erwartete Reaktion
Check-Box abwählen	Service wird beendet
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 01.04.2015

9.12 Test 12

Anforderung	Erfüllungskriterium
ID 12: simple Navigation	nicht instruierte Personen können App intuitiv steuern
Testablauf	Erwartete Reaktion
Customer benutzt App	Customer kann die App ohne fremde Hilfe bedienen
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 08.04.2015

9.13 Test 13

Anforderung	Erfüllungskriterium
ID 13: Warnung ausgeben bei Kontakt	Device (und evt. Uhr) meldet Warnung
Testablauf	Erwartete Reaktion
Testbutton drücken / mit App zur Adresse	Warnung erscheint auf allen aktiven Devices
Tatsächliche Reaktion	Test erfüllt?
Distanz 0.0 = Fehler	Nein 08.04.2015

9.14 Test 13.2

Anforderung	Erfüllungskriterium
ID 13: Warnung ausgeben bei Kontakt	Device (und evt. Uhr) meldet Warnung
Testablauf	Erwartete Reaktion
Testbutton drücken / mit App zur Adresse	Warnung erscheint auf allen aktiven Devices
Tatsächliche Reaktion	Test erfüllt?
Entspricht Erwartung	JA 11.04.2015

10 Gelerntes

Ein wichtiger Aspekt eines jeden Software-Projektes ist, was man daraus mitnimmt. Für zukünftige Projekte sind uns ein paar Punkte aufgefallen:

- Von Anfang an genaue Anforderungen erstellen
- Den Terminplan im Voraus erstellen
- Den wöchentlichen Meetings die nötige Aufmerksamkeit schenken

Im Grossen und Ganzen finden wir unsere Art, ein derartiges Projekt durchzuführen, sehr Gut. Wir wussten immer wer was macht, hatten nie mit persönlichen Differenzen zu kämpfen, was natürlich auch an der geringen Anzahl Teammitglieder lag, und konnten immer vorausschauend weiterarbeiten. Wichtig ist für uns, dass man immer weiss wo man steht, dass man genau vor Augen hat, was erledigt werden muss und in welcher Zeit. Dies haben wir dank den elektronischen Helferlein sehr gut geschafft.

10.1 Reflexion Programmierung

Dieses Projekt hat uns aufgezeigt, wie undurchsichtig und komplex Software sein kann. Beim Erstellen der Software war ich extrem froh, dass wir die Planung schon grösstenteils fertig hatten. Mit abgeschlossener Planung und Diagrammen, lässt es sich viel einfacher codieren und man kann sich als Entwickler auf das Wesentliche konzentrieren. Man wird nicht abgelenkt von neuen Ideen und Features die noch „cool“ wären und unbedingt eingebaut werden müssen. Die Planung hat mir vorgegeben, welche Funktionen die App haben muss und sobald es diese Fähigkeit hatte, war dieser Punkt für mich abgeschlossen. Des Weiteren war ich erfahrener da ich schon mehrere Applikationen gemacht habe. Android-Entwicklung war kein völliges Neuland mehr für mich. Nicht das ich ein Profi Entwickler wäre, auf gewisse Stolpersteine bin ich schon zu oft reingefallen. Eines das sehr oft vorkam, ist das deklarieren einer neuen *Activity* in der *Manifest.xml* Datei. Ohne diese Deklaration werden lauter Fehler ausgegeben, obwohl der Code fehlerfrei ist. Mit jeder weiteren Applikation, lernt man neues hinzu. In diesem Projekt habe ich sehr viel über Services, Threads und parallele Klassen gelernt. Bis zu diesem Zeitpunkt wusste ich, dass diese Themen existieren, ich konnte ihnen jedoch meistens aus dem Weg gehen. Nun war es an der Zeit, sich mit diesem Thema auseinanderzusetzen. Am Anfang schien es sehr schwierig zu sein. Je mehr ich damit experimentierte, desto einfacher schien es zu werden. Ich beherrsche diese Thematik immer noch nicht zu 100 Prozent, jedoch kann ich behaupten, soviel zu verstehen, dass ich mit diesen Technologien eine lauffähige Applikation zustande bekomme. Ein weiteres Feature von Android, was ich neu kennengelernt habe, ist die `startActivityForResult()` Methode. Sie gibt nach dem Abarbeiten einer Aufgabe, ein Resultat zurück. Ohne diese Kenntnis wäre die Aufgabe fast nicht zu bewältigen gewesen.

Probleme habe ich immer mit demselben Muster gelöst. Sobald Fehler aufgetaucht sind, die ich mir nicht erklären konnte, wurde zuerst die Theorie nachgeschlagen. Anhand deren habe ich den fehlerhaften Code in einem Testprojekt laufengelassen um den Fehler einzugrenzen. Meistens habe ich so die „Arbeitsweise“ des Codes verstanden und somit das Problem lösen können. Es gab auch Bugs, die ich mit diesem Verfahren nicht zum fehlerfreien Run bringen konnte. Wenn das der Fall war, habe ich mein Ziel gegoogelt. Auf www.stackoverflow.com fand ich immer eine Lösung, jedoch wurden meistens andere Ansätze verwendet. Zu erwähnen ist, dass nie eine

„pfannenfertige“ Lösung gefunden bzw. verwendet wurde. Anpassungen und Änderungen des Programmierkonzepts, waren immer von Nöten, um schwerwiegende Probleme zu lösen. Da ich nach meinem Studium als Mobile-Application-Developer arbeiten will, möchte ich jetzt so viele Konzepte und Lösungsansätze verstehen wie nur möglich. Darum kamen für mich fertige Lösungen und Codestücke nicht in Frage.

In einem weiteren Projekt würde ich nicht viel anders machen. Wir haben als Team sehr gut funktioniert. Samuel konnte sehr gut meine Schwächen ausbügeln und ich denke, dass ich für ihn auch eine Hilfe war. Im Grossen und Ganzen haben wir uns hervorragend ergänzt. Programmiertechnisch gibt es nicht viel was geändert werden muss. Anhand des Terminplans und den wöchentlichen Sitzungen, konnte ich mich hervorragend orientieren. Wenn mal ein fast unmögliches Problem auftauchte, habe ich das Konzept von „Teile und Herrsche“ angewandt. Bis zum Schluss hat es mich erfolgreich durch das Projekt begleitet. Ganz am Schluss hat es trotzdem nicht gereicht um termingerecht die Tests abzuschliessen. Zum Glück hatten wir noch Pufferzeit eingeplant, für den Fall einer Verzögerung.

Bugs sind im Laufe des codieren immer aufgetaucht. Schätzungsweise waren 90 % der Bugs in weniger als 10min behoben. Weitere 9% verlangten tiefere Kenntnisse in der Android-Entwicklung. Für das letzte Prozent konnten wir nichts dafür: Der Host, wo unser PHP File für die Datenbankspeicherung abgelegt war, war für mehrere Tage down und die Firma kommunizierte es nicht. Das File schien auf dem Server zu liegen, trotzdem funktionierte der Speichervorgang nicht. Das Problem habe ich nach hinten verlegt. Als ich mich dem widmen wollte, war der Host wieder online und die Speicherung funktionierte tadellos.

Ich hoffe, ich konnte einen kleinen Einblick aus meiner Sicht geben und man kann sich nun besser in meine Situation hineinversetzen.

Das Projekt hat uns neben der Arbeit auch viel Spass bereitet und als Entwicklungsteam ist es der beste Lohn, wenn man miterlebt, wie eine Idee sich zu einer kompletten Anwendung entwickelt. Mein grösster Erfolgsmoment war, als die App das erste Mal die Arbeit aufnahm und so reagierte wie sie soll. Es war sehr erfreulich zu sehen, wie die App auf den User reagiert und alle Teilsysteme miteinander arbeiten.

11 Journal

Das Journal soll einen Überblick bieten, was wann erledigt wurde. Zusammen mit den Teamsitzungsprotokollen sowie der Terminplanung sollte eine nachträgliche Rekapitulation der Ereignisse möglich sein. Das vollständige Journal findet sich im Anhang.

12 Glossar

PDF » Portable Document Format

Kapo » Kantonspolizei

GUI » Graphical User Interface

DB » DataBase

HTTP » HyperText Transfer Protocol

GPS » Global Positioning System

APP » Application

GIT » Git (engl.: "Blödmann")

SVN » SubVersioN

URL » Uniform Resource Locator

13 Anhang

13.1 Terminplan

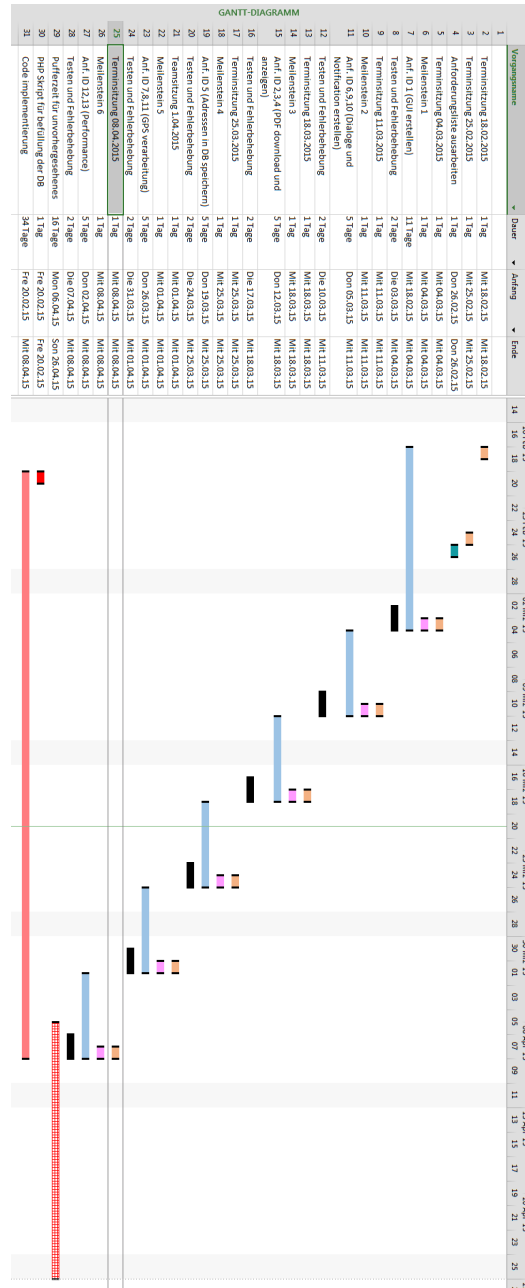


Abbildung 9: Termin Plan

13.2 Journal

ID	Was	Wann	Zeitaufwand [h]	Kürzel
1	Teamsitzung 1 (Aufgabe erhalten)	18.02.2015	0.5	Team
2	Brainstorming bezüglich Aufgabe und Prozessfindung	18.02.2015	1	Team
3	GUI (Aussehen der App bestimmen)	21.02.2015	0.5	ds
4	MainActivity den Forderungen anpassen	"immer wieder"	7	ds
5	Dokumentation beginnen mit LaTeX Vorlage	21.02.2015	1	sl
6	GUI implementieren	23.02.2015	1	Team
7	Teamsitzung 2	25.02.2015	0.25	Team
8	Test ID 1	01.03.2015	0.25	Team
9	Teamsitzung 3	04.03.2015	0.25	Team
10	Meilenstein 1	04.03.2015	0.5	Team
11	Dokumentation nachführen mit Test und Prozess	05.03.2015	1.5	sl
12	Dialoge für verschiedene Situationen erstellen	04.03.2015	1	ds
13	Notification erstellen (Warnen Blitzer in der Nähe)	07.03.2015	1	ds
14	GPS Dialog mit Antwort	08.03.2015	1.5	ds
15	Test ID 6,9,10	09.03.2015	0.5	Team
16	Teamsitzung 4	11.03.2015	0.25	Team
17	Meilenstein 2	11.03.2015	0.5	Team
18	PDF runterladen von Webserver	12.03.2015	0.5	ds
19	Inhalt vom PDF in String verarbeiten	14.03.2015	2	ds
20	PDF Inhalte anzeigen	16.03.2015	2	ds
21	Test ID 2,3,4	17.03.2015	0.5	Team
22	Teamsitzung 5	18.03.2015	0.25	Team
23	Meilenstein 3	18.03.2015	0.5	Team
24	Adressen des PDF's in externe DB speichern	19.03.2015	0.75	ds
25	Test ID 5	19.03.2015	0.5	Team
26	Dokumentation nachführen (Test, Aufgabe)	19.03.2015	3	sl
27	Teamsitzung 6	25.03.2015	0.25	Team
28	Meilenstein 4	25.03.2015	0.5	Team
29	GPS Thread erstellen	25.03.2015	3.5	ds
30	GPS Service abbrechen	26.03.2015	1.5	ds
31	GPS Daten von Nutzer einbinden	26.03.2015	0.5	ds
32	Dokumentation nachführen (Anforderungen, Prog.)	26.03.2015	3	sl
33	Test ID 7,8,11	01.04.2015	0.5	Team
34	Teamsitzung 7	01.04.2015	0.25	Team
35	Meilenstein 5	01.04.2015	0.5	Team
36	Position mit Adressen vergleichen	02.04.2015	2	ds
37	AusschaltDialog mit Antwort	03.04.2015	3	ds
38	Dokumentation nachführen (Tests, Meilensteine)	03.04.2015	2	sl
39	Dokumentation soweit fertigstellen (Journal)	04.04.2015	4	sl
40	Test ID 12,13	08.04.2015	0.5	Team
41	Teamsitzung 8	08.04.2015	0.25	Team
42	Meilenstein 6	08.04.2015	0.5	Team
43	Methode GPS2Meter überarbeiten	09.04.2015	2.5	Team
44	Test ID 13 wiederholt	11.04.2015	0.5	Team
	Total Zeit		54.25	

Abbildung 10: Journal

13.3 Klassendiagramm

Klassendiagramm «Dopplereffekt»

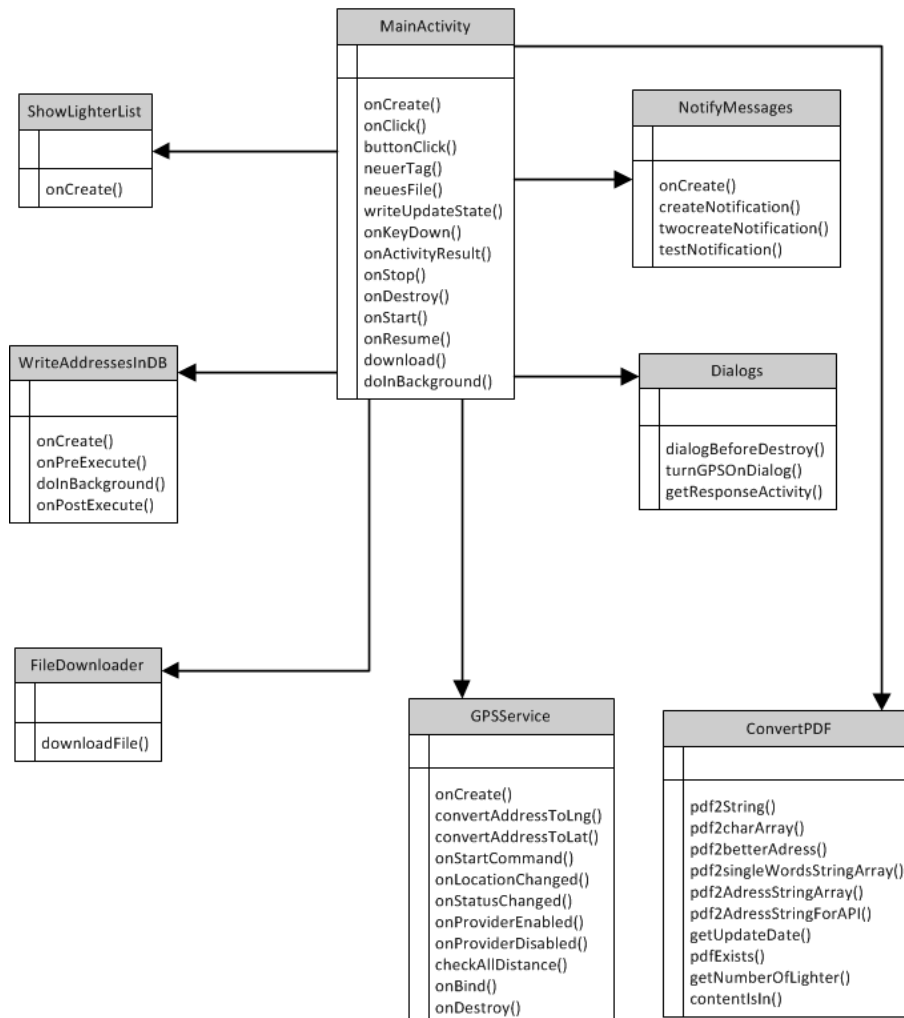


Abbildung 11: Klassendiagramm

13.4 Sequenzdiagramm

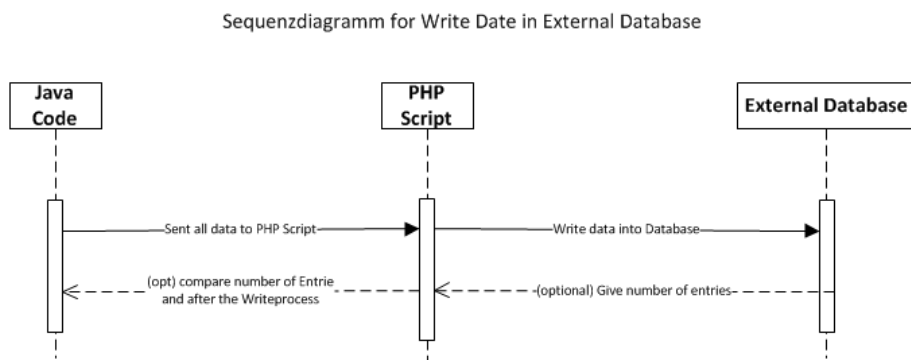
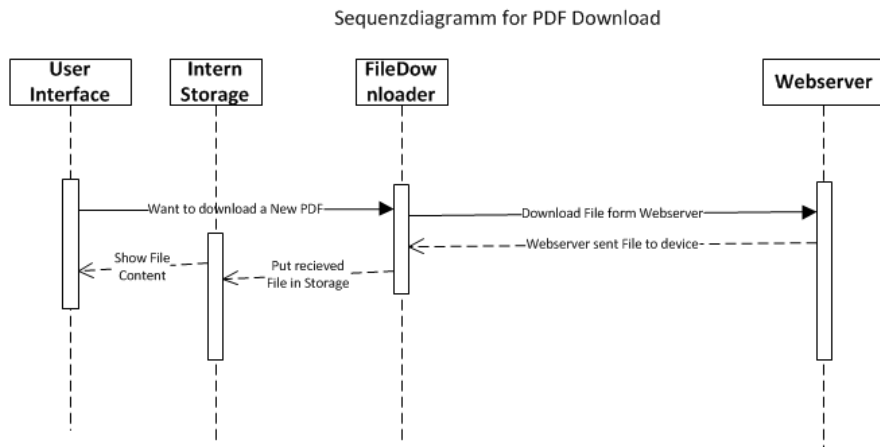


Abbildung 12: Sequenzdiagramm

14 Daten-CD

Hier einkleben...