Realisierungsbericht

|  |  |
| --- | --- |
| **Status** | Abgeschlossen |
| **Projektname** | RentAFlat |
| **Projektleiter** | Victor Hargrave |
| **Auftraggeber** | Andreas Scheidegger |
| **Autoren** | Lars Glatthard, Luca Berger, Victor Hargrave, Felix Neidhart |
| **Verteiler** | Felix Neidhart |

**Änderungskontrolle, Prüfung, Genehmigung**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Datum | Beschreibung, Bemerkung | Name oder Rolle |
| 0.1 | 05.05.15 | Initialversion | Felix Neidhart |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Definitionen und Abkürzungen**

|  |  |
| --- | --- |
| Begriff / Abkürzung | Bedeutung |
|  |  |

**Referenzen**

|  |  |
| --- | --- |
| Referenz | Titel, Quelle |
| [1] |  |
| [2] |  |
| [3] |  |

**Inhaltsverzeichnis**

[1 Zusammenfassung (Felix) 3](#_Toc418584721)

[2 Technische Detailspezifikation (Vic, Luca) 3](#_Toc418584722)

[2.1 Systemdesign 3](#_Toc418584723)

[2.1.1 Struktur 3](#_Toc418584724)

[2.1.2 Dynamik 3](#_Toc418584725)

[2.2 Schnittstellendefinitionen (Luca) 4](#_Toc418584726)

[2.3 Datenmodell (Lars) 4](#_Toc418584727)

[3 Systemdokumentation (Felix) 4](#_Toc418584728)

[3.1 Anwendungshandbuch (Felix) 4](#_Toc418584730)

[3.1.1 Übersicht 4](#_Toc418584731)

[3.1.2 Funktionen und Detailbeschreibung 4](#_Toc418584732)

[3.1.3 Fehlerbehandlung 4](#_Toc418584733)

[3.2 Integrations- und Installationshandbuch (Weglassen, Erklärung Felix) 5](#_Toc418584734)

[3.3 Betriebshandbuch (Weglassen, Erklärung Felix) 5](#_Toc418584735)

[4 Systemtest (Vic, Lars) 5](#_Toc418584736)

[4.1 Unit-Test 5](#_Toc418584738)

[4.2 Systemtest 5](#_Toc418584739)

[4.2.1 Testfälle 5](#_Toc418584740)

[*Testfall 1 „Neuen Benutzer registrieren und anmelden“* 6](#_Toc418584741)

[*Testfall 2 „Bestehenden Benutzer registrieren“* 6](#_Toc418584742)

[5 Testprotokoll (Vic, Lars) 7](#_Toc418584743)

[*5.1* *Systemtest 1* 7](#_Toc418584744)

[*5.1.1* *Testfall 1 „Neuen Benutzer registrieren und anmelden“* 7](#_Toc418584745)

[*5.1.2* *Testfall 2 „Bestehenden Benutzer registrieren“* 7](#_Toc418584746)

[6 Weiterführung der Projektplanung (Felix) 7](#_Toc418584747)

[6.1 Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept 7](#_Toc418584748)

[6.2 Aktualisierung der Risikosituation 7](#_Toc418584749)

[6.3 Planung der nächsten Phase 7](#_Toc418584750)

[Anhang A: Quellcode 8](#_Toc418584751)

[Anhang B: Testcode Unit-Tests 8](#_Toc418584752)

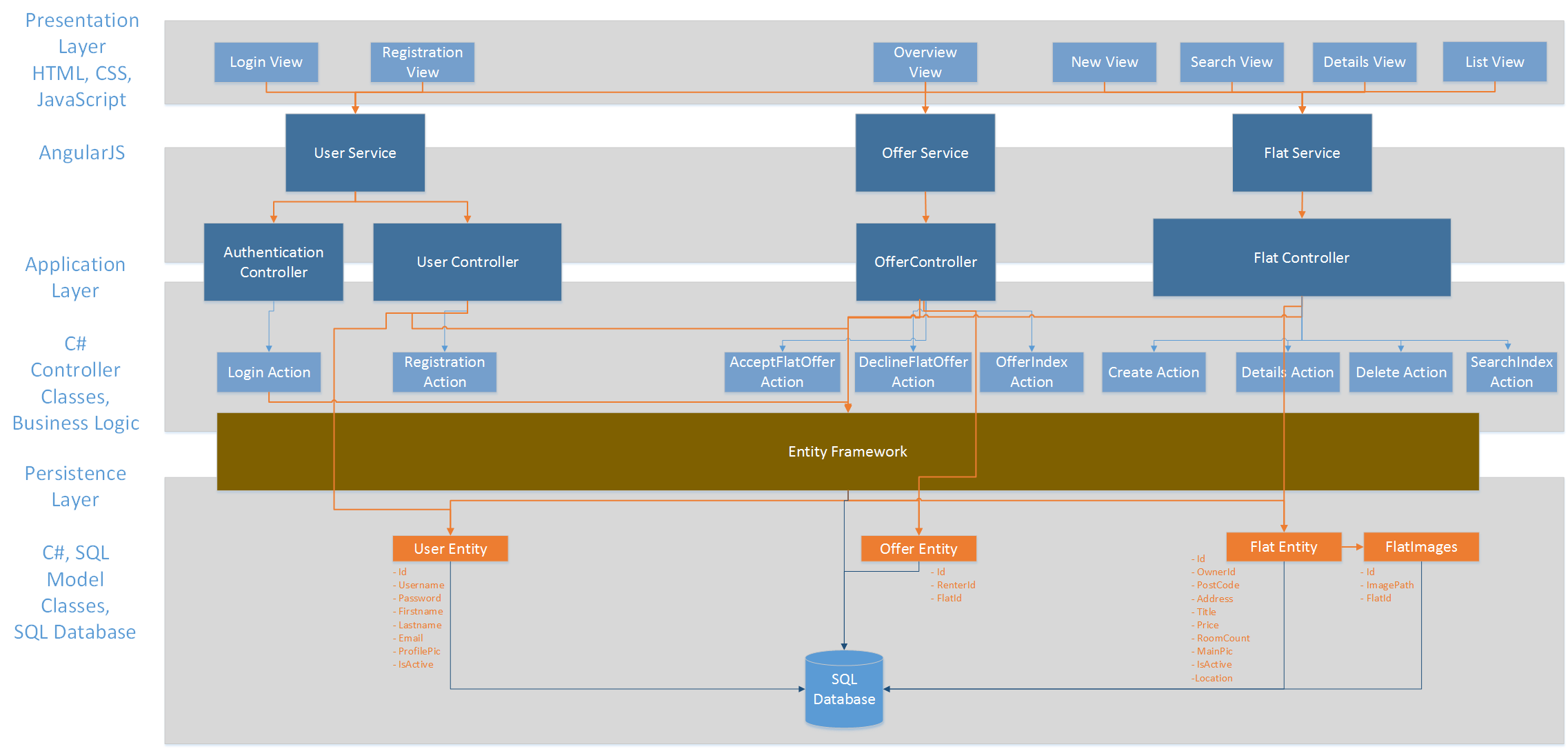
**Abbildungsverzeichnis**

# Zusammenfassung (Felix)

Geben Sie hier eine kurze Zusammenfassung des Inhalts dieses Dokumentes. Wozu dient das Dokument (Zweck) und welche Informationen enthält es?

# Technische Detailspezifikation (Vic, Luca)

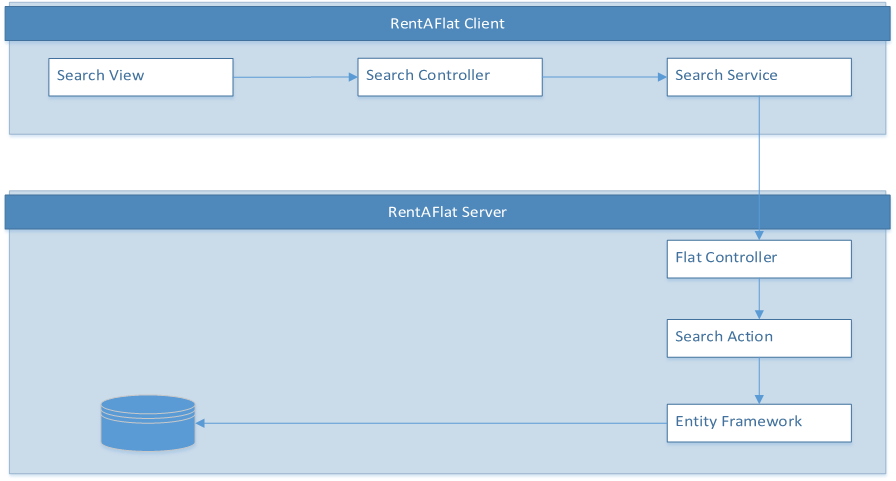
## Systemdesign



### Struktur

Der Benutzer öffnet den Browser und navigiert zur Website „RentAFlat“. Die Seite besteht aus HTML, CSS und AngularJS (JavaScript Framework). Wenn der Benutzer ein Request zum Server senden möchte, kann er dies über ein Suchfeld machen. (Z. B. das Suchen nach einer Wohnung). Von der „Search“-View wird eine Anfrage zum Controller geschickt, der dies danach zum Service weiterleitet. Dieser Service kümmert sich danach um die Verbindung zum Server und nimmt die gegebenen Parameter entgegen.   
Sofern der Service alle Parameter beisammen hat, sendet er ein http-Request zum Server. Dieser kümmert sich dann um die Authentifizierung, sofern diese benötigt ist und schickt anschliessend die Daten im JSON Format an den AngularJS Service zurück.

### Dynamik



## Schnittstellendefinitionen (Luca)

In unserem Projekt haben wir zwei Schnittstellen. Das User Interface, wo dem Benutzer die Möglichkeit gegeben wird, mit dem Tool zu interagieren. Hierbei gibt es keine Regeln.

Die zweite Schnittstelle ist ein von uns programmierter Webservice, der die Requests vom Client entgegen nimmt und dann eine Antwort zurücksendet. Wenn der Service nicht erreichbar ist, wird dem Benutzer eine Meldung auf dem GUI angezeigt.

Die Meldungen und die Daten sind immer im JSON Format und werden als Parameter im http-Request, resp. In der http-Response mitgeschickt. Falls die Daten vom Client nicht korrekt kommen, wird der Webservice nicht aufgerufen und es wird eine Meldung im GUI angezeigt.

## Datenmodell (Lars)

Wenn Sie in Ihrem System eine Datenbank haben, fügen Sie hier eine ERD ein. Beschreiben Sie die einzelnen Tabellen wo nötig kurz. Aus dem ERD müssen die Primary- und Foreign-Keys, sowie die Multiplizitäten hervorgehen

Wenn Sie objektorientiert arbeiten und ein Mapping-Framework (z.B. JPA/Hibernate oder ADO.Net Entity-Framework) verwenden, können Sie auch ein Klassendiagramm mit den Entitätsklassen einfügen. Auch hier beschreiben Sie die einzelnen Klassen, wo nötig, kurz.

# Systemdokumentation (Felix)

Hermes 5 definiert unter anderem das Anwendungshandbuch, die Integrations- und Installationsanleitung, sowie das Betriebshandbuch als Resultate der Realisierungsphase. Hier werden diese Ergebnisse unter dem Oberbegriff Systemdokumentation zusammengefasst.



## Anwendungshandbuch (Felix)

Das Anwendungshandbuch richtet sich an den Anwender oder Benutzer des Systems. Es „enthält alle Informationen, die der Anwender eines Produkts / Systems braucht, um es ordnungsgemäss zu bedienen und im Fall von Problemen richtig reagieren zu können“ (siehe HERMES 5 Handbuch Seite 127). Dazu gehören Übersicht, Funktionen, Detailbeschreibungen zur Anwendung und Fehlerbehandlung.

Denken Sie daran, dass die Anwender in der Regel keine IT-Fachpersonen sind. Entsprechend muss das Anwendungshandbuch in einer für das Zielpublikum verständlichen Sprache geschrieben sein.

### Übersicht

Hier folgt ein kurzer Überblick des Systems für die Anwender. Beschreiben Sie den Zweck und die Ziele des Systems.

### Funktionen und Detailbeschreibung

Beschreiben Sie hier die einzelnen Funktionen des Systems soweit, dass der Anwender diese erfolgreich anwenden kann. Stellen Sie sich also die Frage: „Was muss dem Anwender gesagt werden, damit er das System erfolgreich benutzen kann?“

Wenn Sie mit Anwendungsfällen gearbeitet haben, können Sie diese als Basis verwenden. Arbeiten Sie mit Screenshots. Denken Sie auch daran, dass es unter Umständen verschiedene Anwender gibt, wie registrierte Benutzer, Administratoren, etc.

### Fehlerbehandlung

Listen Sie hier mögliche Fehler, mit denen der Anwender konfrontierte werden kann auf und beschreiben Sie, was mögliche Gründe sein können und wie beim Auftraten dieser Fehler zu verfahren ist.

## Integrations- und Installationshandbuch (Weglassen, Erklärung Felix)

Fügen Sie hier eine Installationsanleitung für Ihr System ein. Je nach Art des Systems, muss eine unabhängige Fachperson oder gar der Anwender selber in der Lage sein, mit dieser Anleitung Ihr System erfolgreich zu installieren. Beschreiben Sie die einzelnen Schritte, allfällige Konfigurationsdaten und Tests zur Überprüfung, ob die Installation erfolgreich war.

## Betriebshandbuch (Weglassen, Erklärung Felix)

Wird der Betrieb des Systems nicht durch den Anwender selbst wahrgenommen, so braucht es ein Betriebshandbuch, welches „alle Informationen, die der Betreiber benötigt, um das

System ordnungsgemäss betreiben und im Fall von Problemen richtig reagieren zu können“ enthält (siehe HERMES 5 Handbuch Seite 128). Dazu gehören im Minimum Angaben über den Start / Neustart des Systems, Überwachung des Systems (z.B. Logs), Sicherung des Systems (z.B. Backup) sowie zu ergreifende Massnahmen bei Fehlern

# Systemtest (Vic, Lars)

Im Modul 226 haben Sie die verschiedenen Teststufen und Vorgehensweisen zum Testen kennengelernt. Zur Repetition: Das Ziel beim Testen ist es, Fehler zu finden. Es geht nicht darum zu zeigen, dass das System keine Fehler hat.

Getestet wird auf verschiedenen Ebenen vom Einzeltest bis zum Abnahmetest (Akzeptanztest beim Kunden). Wir beschränken uns auf die beiden Ebenen Einzeltest (Unit-Test) und Systemtest/Abnahmetest.

Tests müssen reproduzierbar sein. Die Ausgangslage, der Ablauf und die erwarteten Resultate eines jeden Testfalls müssen detailliert beschrieben sein. Dies ist der Zweck der Testspezifikation.



## Unit-Test

Auf Unit Tests haben wir verzichtet, da wir unter grossem Zeitdruck standen und das Produkt im Vordergrund steht.

## Systemtest

Auf System Tests haben wir verzichtet, da wir unter grossem Zeitdruck standen und das Produkt im Vordergrund steht.

### Testfälle

Nachfolgend beschreiben Sie die einzelnen Testfälle für den Systemtest Ihres Systems. Als Ausgangspunkt nehmen Sie die Ziele aus der Voranalyse und die Anwendungsfälle aus Ihrem Konzept. Jeder Anwendungsfall führt jetzt zu mindestens einem separaten Testfall.

Das Ziel ist die vollständige Abdeckung der Funktionalität Ihres Systems durch die Testfälle. Es wird deshalb nötig sein auch weitere Testfälle zu beschreiben.

Spezifizieren Sie jeden Testfall separat mit einer Tabelle.

Beispiele:

#### *Testfall 1 „Neuen Benutzer registrieren und anmelden“*

|  |  |  |
| --- | --- | --- |
| ***Beschreibung*** | *Ein neuer Benutzer registriert sich und meldet sich beim System an.* | |
| ***Abgedeckte Anwendungsfälle*** | *Benutzerregistrierung* | |
| ***Ausgangssituation*** | *Es besteht noch kein solcher Benutzer.* | |
| ***Vorbereitungsschritte*** | 1. *Die Datenbank des Systems wird über PhpMyAdmin mit dem Script initTF1.sql in den Initialzustand versetzt.* 2. *Starten des Firefox auf der Testclient-Maschine.* | |
| ***Testschritte*** | | ***Erwartetes Resultat*** |
| 1. *Auf dem Testclient im Browser die URL* [*http://www.budget.ch*](http://www.budget.ch) *eingeben* | | *Die Startseite der Budgetplaner-Anwendung wird angezeigt.* |
| 1. *Den Link „Registrieren“ wählen.* | | *Das Registrierungsformular wird angezeigt.* |
| 1. *Eingeben der folgenden Informationen:*    1. *Benutzername:*  *Test1*    2. *Passwort:*  *testABC1*    3. *Passwortwiederholung:*  *testABC2* | |  |
| 1. *Den Button „Registrieren“ wählen* | | *Das Registrierungsformular wir erneut angezeigt. Die Felder für die Passworteingabe sind markiert.* |
| 1. *…* | | *…* |
| 1. *…* | | *In der Datenbank ist in der Tabelle Benutzer ein neuer Record mit dem Namen Test1 vorhanden. Überprüfung mit PhpMyAdmin.* |

#### *Testfall 2 „Bestehenden Benutzer registrieren“*

|  |  |  |
| --- | --- | --- |
| ***Beschreibung*** | *Ein Benutzer versucht sich mit einem bereits bestehenden Benutzernamen zu registrieren.* | |
| ***Abgedeckte Anwendungsfälle*** | *Benutzerregistrierung* | |
| ***Ausgangssituation*** | *Es besteht bereits ein Benutzer mit dem gleichen Namen.* | |
| ***Vorbereitungsschritte*** | 1. *Die Datenbank des Systems wird über PhpMyAdmin mit dem Script initTF2.sql in den Initialzustand versetzt.* 2. *Starten des Firefox auf der Testclient-Maschine.* | |
| ***Testschritte*** | | ***Erwartetes Resultat*** |
| 1. *Auf dem Testclient im Browser die URL* [*http://www.budget.ch*](http://www.budget.ch) *eingeben* | | *Die Startseite der Budgetplaner-Anwendung wird angezeigt.* |
| 1. *Den Link „Registrieren“ wählen.* | | *Das Registrierungsformular wird angezeigt.* |
| 1. *Eingeben der folgenden Informationen:*    1. *Benutzername:*  *Test1*    2. *Passwort:*  *testABC1*    3. *Passwortwiederholung:*  *testABC1* | |  |
| 1. *Den Button „Registrieren“ wählen* | | *Das Registrierungsformular wir erneut angezeigt. Das Feld für die Eingabe des Benutzernamens ist markiert.* |
| 1. *…* | |  |

# Testprotokoll (Vic, Lars)

Das Testprotokoll gibt darüber Auskunft, wer, wann, was getestet hat und welches die Befunde des betreffenden Testdurchlaufes waren. Führen Sie die einzelnen Testdurchläufe in separaten Unterkapiteln auf.

Beispiel:

## *Systemtest 1*

*Getestete Version: 0.5, Iteration 1*

*Tester: Bill Geitsnoguet*

*Datum, Zeit: 13.5.2011, 14.00 – 15.30*

### *Testfall 1 „Neuen Benutzer registrieren und anmelden“*

|  |  |  |
| --- | --- | --- |
| ***Testschritt*** | ***Erfüllt*** | ***Bemerkung*** |
|  | 🗹 |  |
|  | 🗹 |  |
|  | 🗹 |  |
|  | 🗹 |  |
|  | 🗹 |  |
|  | 🗷 | *In der Tabelle Benutzer ist keine Record mit dem Namen Test1 eingetragen worden.* |

### *Testfall 2 „Bestehenden Benutzer registrieren“*

*Ganzer Testfall erfüllt.* (Tabelle mit Testschritten erübrigt sich.)

# Weiterführung der Projektplanung (Felix)

## Abgleich von Planung und tatsächlichem Verlauf der Phase Konzept

Soll/Ist-Vergleich bezüglich der Zeitplanung für die Phase Realisierung.

Welche Risiken sind während der Phase Realisierung Realität geworden? Wie wurde diese Situation gemeistert? Welche Konsequenzen ergeben sich daraus für das gesamte Projekt?

## Aktualisierung der Risikosituation

Haben sich an der Risikosituation Änderungen (Verschärfung oder Entschärfung, neue Risiken) ergeben? Dann beschreiben Sie hier die Risikosituation neu. Führen Sie wiederum konkrete Massnahmen auf, wie Sie den Risiken in der nächsten Phase begegnen wollen.

## Planung der nächsten Phase

Aktualisieren und verfeinern Sie hier Ihren Projektplan, vor allem in Bezug auf die nächste Phase.

# Anhang A: Quellcode

Fügen Sie zuerst eine **Übersicht mit allen Dateien** des Codes ein. Zum Code gehören auch HTML, CSS, JS, JSP, SQL, etc.

Anschliessend folgt der Code.

**Unbedingt beachten**:

* Formatierung und Einrückung
* Neue Seite bei neuer Datei

# Anhang B: Testcode Unit-Tests

Da wir keine Unit Test gemacht haben, gibt es keinen Code dazu.