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**

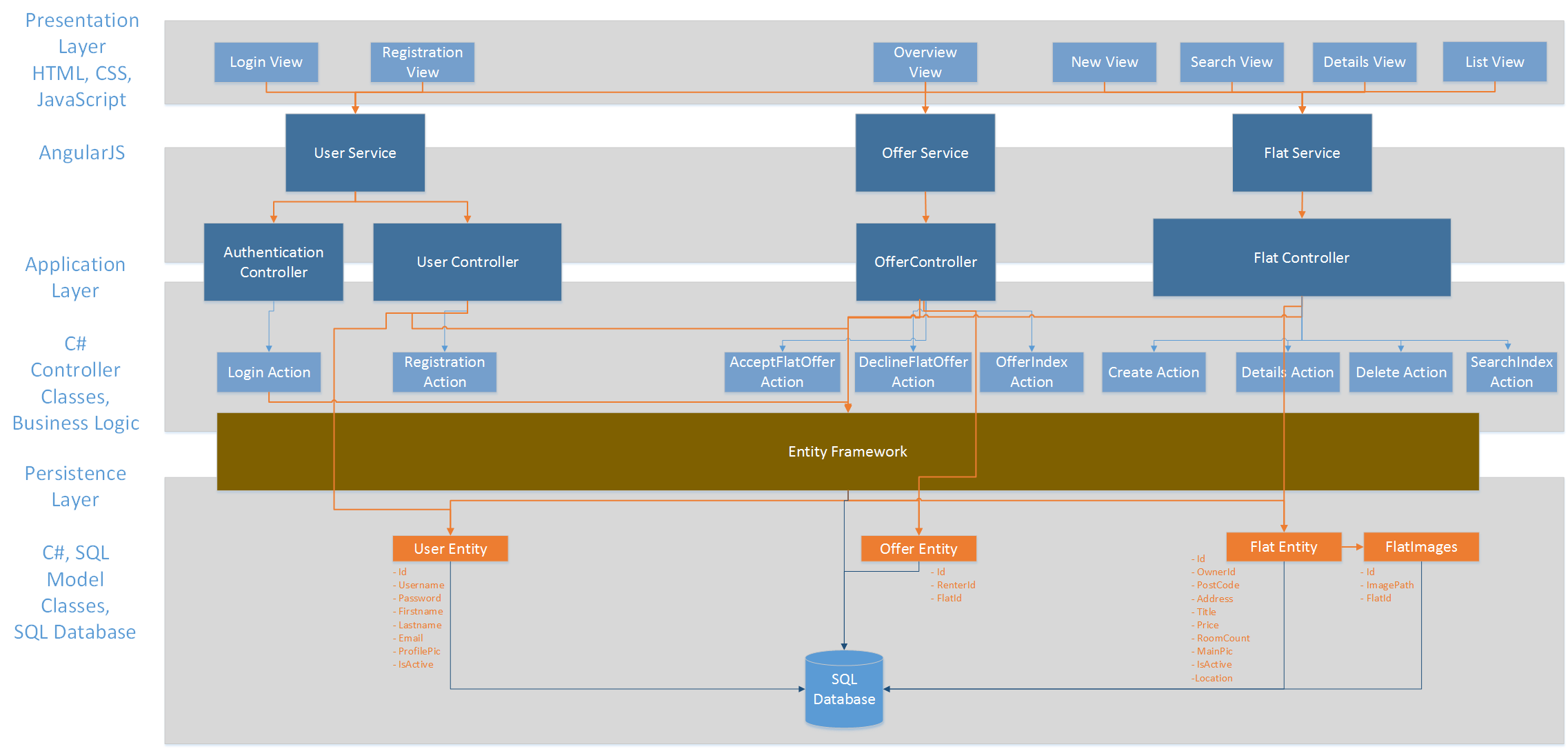
# Zusammenfassung (Felix)

Dieses Dokument hält die abgeschlossenen Arbeiten im Projekt fest. Erklärt werden alle technischen Aspekte, alle Tests und die Bedienung der Webseite (als Anwender). Zum Schluss wird noch ein Soll/Ist Vergleich mit der vorangegangenen Planung aus den Initialisierungs-, Studien- und Konzeptberichten.

Die Dokumentation richtet sich an alle Benutzergruppen, die mit der Anwendung in Berührung kommen (könnten): Anwender, Entwickler, Auftraggeber, Projektleiter und Tester.

# Technische Detailspezifikation (Vic, Luca)

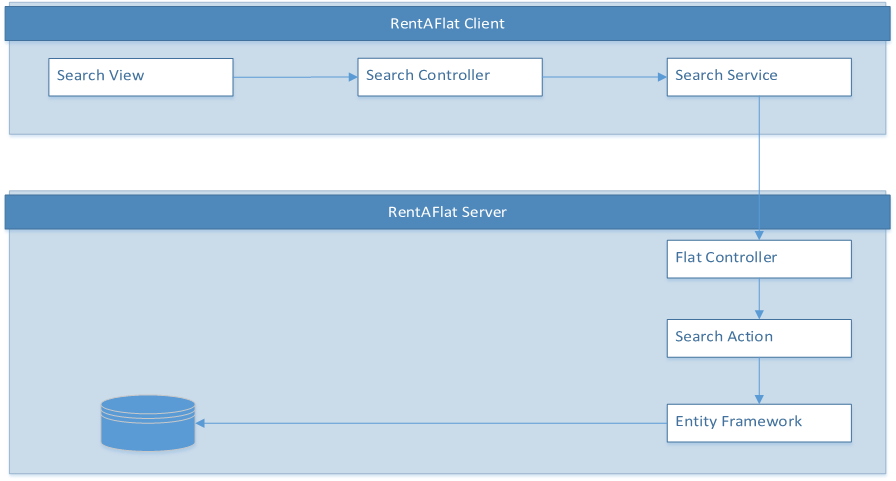
## Systemdesign



### 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)

## Anwendungshandbuch

|  |  |
| --- | --- |
| Die Startseite beinhaltet die Suchfunktion. Die Suchfunktion wurde stark vereinfacht, um ein angenehmes und schnelles Suchen zu ermöglichen. |  |
| Nach dem Klick auf „Search flat“ werden alle verfügbaren Wohnungen angezeigt.  Wählt man eine der Wohnungen aus, so kommt man auf die Detailansicht. |  |
| Dies ist die Detailansicht. |  |
| Wenn man im Header auf „Login“ klickt, erscheint dieses Fenster. Hat man bereits einen Account, so kann man sich hier einloggen. |  |
| Hat man jedoch noch keinen Account, so kann man sich über „Register“ registrieren.  Die Registrationsvorgaben sind:  Username: Mind. 3 Zeichen  Password: 1 Gross- & Kleinbuchstaben und 1 Zahl  Email: Gültige Mail Adresse  Prename: Mind. 5 Buchstaben  Lastname: Mind. 5 Buchstaben. |  |
| Hat man sich registriert / eingeloggt, so erscheint im Header-Balken der Benutzername. Klickt man darauf, so öffnet sich ein Dropdown wie nebenan abgebildet. Mit „Logout“ wird man ausgeloggt, mit „Notifications“ sieht man seine Benachrichtigungen und mit „Overview“ wird man auf seine persönliche Übersichtsseite weitergeleitet. |  |
| Dies ist die persönliche Übersichtsseite. Hier werden die Wohnungen des Benutzers aufgelistet, wie auch die Bewerbungen darauf. Auch zu sehen sind die Wohnungen, die man selbst gemietet hat. |  |

### Übersicht



### 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

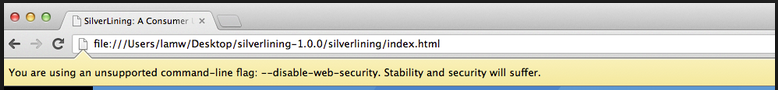
Listen Sie hier mögliche Fehler, mit denen der Anwender konfrontierte werden kann auf und beschreiben Sie, was mögliche Gründe sein können und wie beim Auftraten dieser Fehler zu verfahren ist.

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

Da unsere Applikation webbasiert ist, wurde dieser Teil zu einem Grossteil weggelassen. Eine Änderung, die lokal vorzunehmen ist, haben wir jedoch hier wie folgt beschrieben:

Unglücklicherweise erhalten wir von allen Browsern die Message „Same-origin Policy wurde verletzt.“  
Dies geschieht, weil der Server auf einem anderen Port läuft als der Client.  
Die Folge davon ist, dass der Browser die Requests zum Server nicht abschickt, sondern blockiert.

Als Workaround, kann man im Chrome diese Einstellung verhindern:

1. Navigieren sie im cmd zu folgendem Pfad:  
   „C:\Program Files (x86)\Google\Chrome\Application“
2. Geben sie im cmd folgenden Befehl ein:  
   „chrome.exe --disable-web-security“
3. Warten sie bis der Browser folgende Nachricht anzeigt:  
   
4. Navigieren sie anschliessend in **diesem Tab** auf die Seite RentAFlat!

## Betriebshandbuch (Weglassen, Erklärung Felix)

Dieser Teil wurde weggelassen, da unsere Applikation im Betrieb nicht betreut werden muss. Es wird kein Administrator benötigt.

# 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.

Testfall 4 „Wohnung erfassen“

|  |  |  |
| --- | --- | --- |
| **Beschreibung** | Ein eingeloggter Benutzer erfasst eine neue Wohnung im System. | |
| **Abgedeckte Anwendungsfälle** | Als Benutzer möchte ich eine Wohnung ausschreiben können, um die Wohnung zu vermieten. | |
| **Ausgangssituation** | Der Benutzer ist eingeloggt | |
| **Vorbereitungsschritte** | Der Benutzer hat sich eingeloggt und ist auf der Flat „My Flats“ Page | |
| **Testschritte** | | **Erwartetes Resultat** |
| Der Benutzer klickt auf dem Button „Wohnung erfassen“ | | Der Benutzer wird auf er /Flat/Create Page weitergeleitet |
|  | | Das Wohnungerfassen Formular wird angezeigt. |
| Der Benutzer gibt folgende Daten ein :  Post Code: 1111  Address: Bernstrasse 59  Title: Cheap Wohnung  Price: 400$ / Night  Room Count: 3.5  Main Pic: none  Location: Bern | |  |
| Den Button „Speichern“ wählen | | Der Benutzer wird auf der „My Flats“ Page weitergeleitet und sieht seine neu erfasste Wohnung |
| … | | … |
| … | | In der Datenbank ist in der Tabelle Flats ein neuer Record mit dem Namen Cheap Wohnung vorhanden. Überprüfung mit SQL Abfrage in VS. |

Testfall 5 „Wohnung löschen“

|  |  |  |
| --- | --- | --- |
| **Beschreibung** | Ein eingeloggter Benutzer löscht eine von seiner Wohnungen im System. | |
| **Abgedeckte Anwendungsfälle** | Als Benutzer möchte ich meine Wohnungsinserate löschen können, damit sie nicht mehr im System gefunden werden. | |
| **Ausgangssituation** | Der Benutzer ist eingeloggt  Testfall 4 wurde ausgeführt | |
| **Vorbereitungsschritte** | Der Benutzer hat sich eingeloggt und ist auf der Flat „My Flats“ Page | |
| **Testschritte** | | **Erwartetes Resultat** |
| Der Benutzer klickt auf dem Link „Löschen“ beim Eintrag mit dem Namen „Cheap Wohnung“ | | Das System fragt ob der Benutzer den Eintrag wirklich löschen will |
| Der Benutzer klickt auf ja | | Die „My Flats“ Page wird neu geladen und den Eintrag „Cheap Wohnung“ wird nicht mehr gefunden |
| … | | In der Datenbank ist in der Tabelle Flats kein Record mehr mit dem Namen „Cheap Wohnung“ vorhanden. |

Testfall 6 „Mich für eine Wohnung bewerben“

|  |  |  |
| --- | --- | --- |
| **Beschreibung** | Ein eingeloggter Benutzer möchte ich mich für eine Wohnung bewerben | |
| **Abgedeckte Anwendungsfälle** | Als Benutzer möchte ich mich für eine Wohnung bewerben, damit ich die Wohnung mieten kann. | |
| **Ausgangssituation** | Der Benutzer ist eingeloggt | |
| **Vorbereitungsschritte** | Der Benutzer hat sich eingeloggt und ist auf der Flat „Details“ Page von einer Wohnung | |
| **Testschritte** | | **Erwartetes Resultat** |
| Der Benutzer klickt auf dem Button „mich bewerben“ | | Ein kleines Offer Formular öffnet sich |
| Der Benutzer gibt folgende Daten ein :  From : 02.11.2015  Until : 04.11.2015 | |  |
| Der Benutzer klickt auf auf dem Button „Speichern“ | | Die Page lädt sich neu. |
|  | | Ein Offer würde am Vermieter der Wohnung geschickt |
| … | | In der Datenbank ist in der Tabelle Flats ein neuer Offer für die Wohnung vorhanden. Überprüfung mit SQL Abfrage in VS. |

# 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.

Systemtest 1

*Getestete Version: 0.5, Iteration 1*

*Tester: Victor Hargrave*

*Datum, Zeit: 10.5.2015, 14.00 – 15.30*

Testfall 4 „Wohnung erfassen“

|  |  |  |
| --- | --- | --- |
| ***Testschritt*** | ***Erfüllt*** | ***Bemerkung*** |
| *1* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
| *2* | 🗷 | *-* |
| *3* | 🗷 | *-* |
|  |  | *Ganzer Testfall nicht erfüllt.* |

Testfall 5 „Wohnung löschen“

|  |  |  |
| --- | --- | --- |
| ***Testschritt*** | ***Erfüllt*** | ***Bemerkung*** |
| *1* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
| *2* | 🗷 | *-* |
|  |  | *Ganzer Testfall nicht erfüllt.* |

Testfall 6 „Mich für eine Wohnung bewerben“

|  |  |  |
| --- | --- | --- |
| ***Testschritt*** | ***Erfüllt*** | ***Bemerkung*** |
| *1* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
| *2* | 🗷 | *-* |
| *3* | 🗷 | *-* |
|  |  | *Ganzer Testfall nicht erfüllt.* |

|  |  |  |
| --- | --- | --- |
| ***Testschritt*** | ***Erfüllt*** | ***Bemerkung*** |
| *1* | 🗹 |  |
| *2* | 🗹 |  |
| *3* | 🗹 |  |
| *4* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
| *5* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
| *6* | 🗷 | *Die Pages sind bis jetzt gar nicht erstellt worden.* |
|  |  | *Ganzer Testfall nicht erfüllt.* |

# Weiterführung der Projektplanung

## Abgleich von Planung und tatsächlichem Verlauf der Phase Realisierung

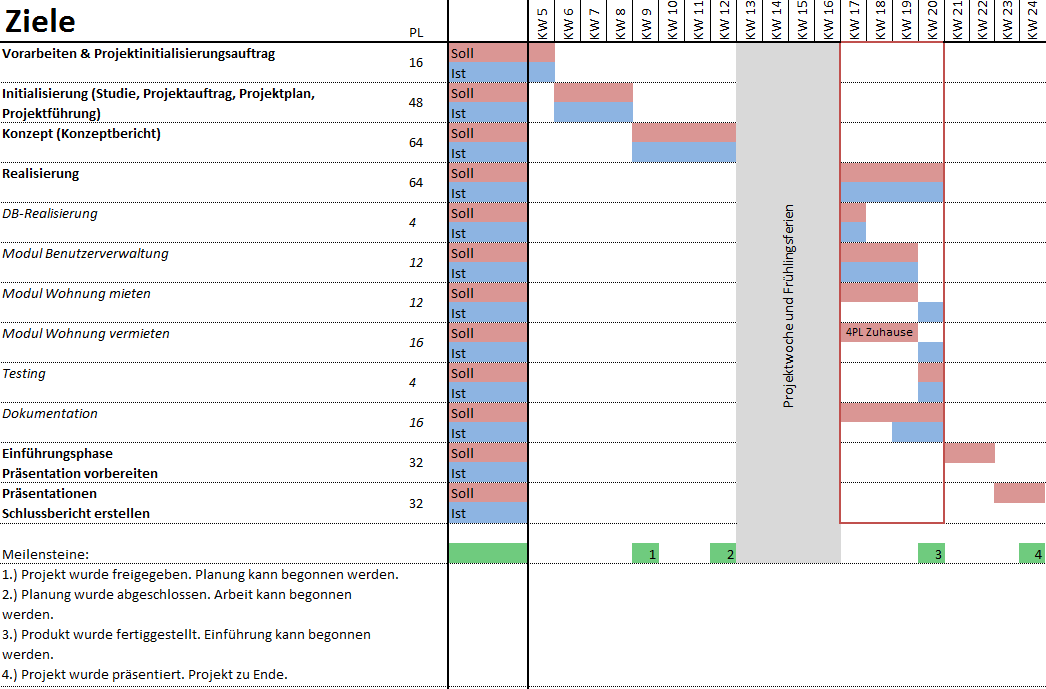
### Soll / Ist Vergleich der einzelnen Punkte der Realisierungsphase

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Soll | Ist | Erfüllt |
| DB-Realisierung | KW17- KW1. | KW17 abgeschlossen. Kleinere Änderungen vorbehalten. | 1 |
| Benutzerverwaltung | KW17- KW19. | KW19 Login & Registration abgeschlossen. Profilbearbeitung wurde weggelassen. Anfängliche Komplikationen mit der Verbindung zum Server. | 1 |
| Wohnung mieten | KW17- KW19. | KW20 durch Arbeit zuhause fertiggestellt. Verzug durch Komplikationen in Phase Benutzerverwaltung. | 0 |
| Wohnung vermieten | KW17- KW19. | KW20 durch Arbeit zuhause fertiggestellt. Verzug durch Komplikationen in Phase Benutzerverwaltung. | 0 |
| Dokumentation | KW17- KW20. | KW20 abgeschlossen. | 1 |

Auf dem Gantt Diagramm ist sehr deutlich zu sehen, dass unsere Planung nur sehr schlecht aufgegangen ist. Folgende Gründe wurden nachträglich festgestellt:

* Für 75% der Entwickler war grosser Lernaufwand notwendig, um überhaupt erst arbeiten zu können. Dabei ging viel Zeit verloren.
* Der Realisierungsaufwand wurde grob unterschätzt.
* Die verwendeten Werkzeuge (Programmiersprachen, Frameworks, Libraries, etc.) sind für grössere Projekte als das unsere besser geeignet.
* Der Aufbau und die Zusammenarbeit sind sehr effizient und übersichtlich. Verlangt wird jedoch ein Grundgerüst, das im Verhältnis zu unserem Zeitbudget, zu aufwändig aufzubauen und einzuführen war.

Als Folge davon wurde eine übersichtliche, leicht erweiterbare, performante und liebevoll designte Anwendung aufgebaut, die jedoch aufgrund des Zeitmankos an vielen zuvor definierten Funktionen mangelt.



## Aktualisierung der Risikosituation

Vorherige Risiken:

1. Andauernde, schwerwiegende Krankheit von Lars Glatthard.
2. Kurzzeitige Krankheit eines Teammitglieds
3. Datenverlust

Vorherige Risikomatrix:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Schadensausmass | Katastrophal | 3 |  |  |  |
| Hoch |  |  |  |  |
| Mittel |  | 1, 2 |  |  |
| Gering |  |  |  |  |
|  | Unwahrscheinlich | Gering | Wahrscheinlich | Sehr Wahrscheinlich |
|  | Eintretens Wahrscheinlichkeit | | | | |

Aktuelle Risikomatrix:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Schadensausmass | Katastrophal | 3 |  |  |  |
| Hoch |  |  |  |  |
| Mittel |  |  |  |  |
| Gering | 1, 2 |  |  |  |
|  | Unwahrscheinlich | Gering | Wahrscheinlich | Sehr Wahrscheinlich |
|  | Eintretens Wahrscheinlichkeit | | | | |

Die Risiken 1 und 2 wurden milder bewertet, da die Temperaturen bereits sehr hoch sind und keine Krankheitswelle im Umlauf ist. Dass Lars wieder krank wird, kann ausgeschlossen werden, da er seit nun 2 Monaten wieder gesund ist, arbeitet und vom Arzt offiziell wieder als gesund erklärt wurde.

In der Konzeptphase hätte man erkennen müssen, dass wir mit den Arbeiten überfordert sein würden und entsprechend Risiken auflisten. Konkret wäre denkbar:

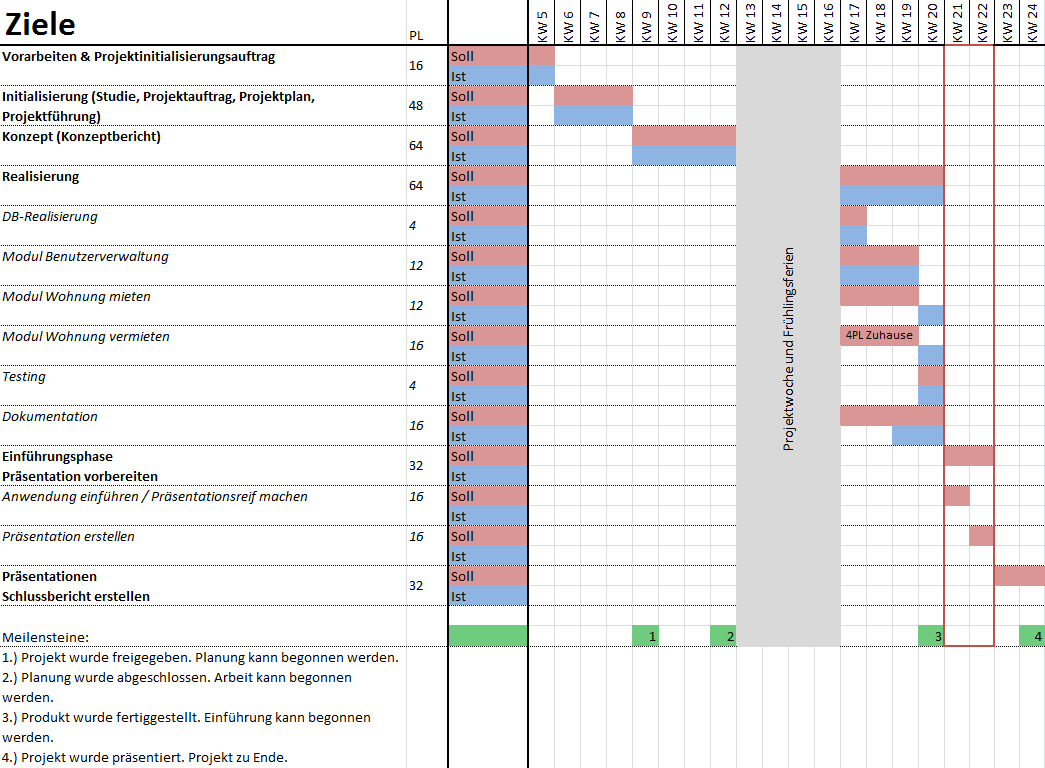
1. Die Mitarbeiter begreifen den neuen Stoff nicht rasch genug
2. Technische Pannen mit der GIBBIX
3. Überschätzung des Arbeitsaufwandes
4. Unvorhersehbare / Restliche Risiken

Dementsprechend hätte die Risikomatrix im Konzept wie folgt ausgesehen:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Schadensausmass | Katastrophal | 3 |  |  |  |
| Hoch |  |  | 4, 6 |  |
| Mittel |  | 1, 2, 5 |  | 7 |
| Gering |  |  |  |  |
|  | Unwahrscheinlich | Gering | Wahrscheinlich | Sehr Wahrscheinlich |
|  | Eintretens Wahrscheinlichkeit | | | | |

Deutlich zu sehen ist hier, wie viele der Risiken sich im roten, daher im nicht tolerierbaren Bereich befinden. Es hätten dementsprechend Massnahmen getroffen werden sollen.

## Planung der nächsten Phase



In der nächsten Phase werden wir uns darum kümmern, dass unser Produkt zumindest präsentationsreif ist, wenn es schon nicht einsatzbereit ist. Wir werden viel mit statischen Daten arbeiten um so den gewünschten Effekt zu erzielen. Zum Schluss werden wir noch eine Präsentation erstellen, mit welcher wir das Produkt dann vorstellen werden.

# 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.