SEP Projektmappe

**Projektmappe des Projektes**

<IMDB\_Klon >

**Dokumentation des Projektes**

Gruppe <H >:

<Lorenz Wollring>

<Felix Bernardi>

<Tobias Zettelmeyer>

<Raphael Kruska>

<David Rumpf>

<Thanansaiyan Pushparajah>

**Hinweis**

An vielen Stellen findet Ihr im Dokument folgendes Kästchen:

*Dies ist eine Hilfestellung.*

Diese Kästen dienen dazu, Euch kurze Informationen über Ziele und Inhalte der jeweiligen Abschnitte zu geben. Auch die Beispiele und Templates dienen dazu, euch bei der Dokumentation eures Projektes zu unterstützen. **Sowohl die Kästchen als auch die Beispiele und Templates sind spätestens zur finalen Abgabe der Projektmappe vollständig zu entfernen.** Betrachtet dieses Dokument bitte nicht als Aufgabe, die man von oben nach unten abarbeiten soll; es soll vielmehr als durchgängige Dokumentation eurer Projektarbeit dienen und fortlaufend erweitert bzw. angepasst werden, sodass am Ende des SEPs der Entwicklungsprozess Eurer Software vollständig dokumentiert ist.

Das SEP-Team wünscht Euch  
**Viel Erfolg**

Inhalt

[Projektbeschreibung 4](#_Toc85022905)

[Zyklus I 5](#_Toc85022906)

[Spezifikationsplanung 5](#_Toc85022907)

[User-Stories 5](#_Toc85022908)

[Papierprototypen 6](#_Toc85022909)

[Szenarien (MSCs) 6](#_Toc85022910)

[Strukturdiagramm (Klassendiagramm) 7](#_Toc85022911)

[Funktionalitätsplanung 7](#_Toc85022912)

[Systemtests 7](#_Toc85022913)

[Zyklus II 10](#_Toc85022914)

[Spezifikationsplanung 10](#_Toc85022915)

[User-Stories 10](#_Toc85022916)

[Papierprototypen 10](#_Toc85022917)

[Szenarien (MSCs) 10](#_Toc85022918)

[Strukturdiagramm (Klassendiagramm) 10](#_Toc85022919)

[Funktionalitätsplanung 10](#_Toc85022920)

[Unittests 11](#_Toc85022921)

[Systemtests 11](#_Toc85022922)

[Zyklus III 14](#_Toc85022923)

[Spezifikationsplanung 14](#_Toc85022924)

[User-Stories 14](#_Toc85022925)

[Papierprototypen 14](#_Toc85022926)

[Szenarien (MSCs) 14](#_Toc85022927)

[Strukturdiagramm (Klassendiagramm) 14](#_Toc85022928)

[Funktionalitätsplanung 15](#_Toc85022929)

[Unittests 15](#_Toc85022930)

[Systemtests 15](#_Toc85022931)

[Nutzerhandbuch 16](#_Toc85022932)

[Technische Anforderungen 16](#_Toc85022933)

[Installationsanleitung 16](#_Toc85022934)

[Bedienungsanleitung 16](#_Toc85022935)

# Projektbeschreibung

In diesem Abschnitt soll die Projektbeschreibung abgedruckt werden, die ihr als Aufgabenbeschreibung von eurem Betreuer erhalten habt. Sie dient als initiales Anforderungsdokument für eure Spezifikationsaktivitäten.

Die SEP-Filmdatenbank soll ein Programm werden, das ein soziales Netzwerk zum Austausch über  
Filme darstellt. Nutzer des Netzwerkes können Filme suchen, eine Watchlist verwalten, Filme  
bewerten und sich mit anderen Nutzern über die Filme austauschen. Nutzer können außerdem ihre  
eigene Statistik einsehen und erhalten Filmvorschläge auf der Grundlage ihrer Bewertungen.  
Systemadministratoren stellen die Daten für die Filme bereit und können Statistiken verwalten.  
Das Programm soll auf einer Client-Server-Architektur aufbauen; der Server soll von mehreren Clients  
gleichzeitig erreichbar sein.

# Zyklus I

## Spezifikationsplanung

Jedes Artefakt, das im Rahmen des SEP erstellt wird, muss mit dem Namen genau einer Verantwortlichen/eines Verantwortlichen versehen werden. Das bedeutet, dass jede User Story, jeder Papierprototyp, jedes MSC, das Klassendiagramm und sämtliche Tests mit dem Namen der/des Verantwortlichen versehen und hier in der untenstehenden Tabelle entsprechend eingetragen werden muss. Natürlich kann die Gruppe gemeinsam an einem Artefakt arbeiten, als Verantwortlicher sollte aber genau eine Person eingetragen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Artefakte | Art des  Artefakts | Verantwortlicher | Status |
|  | Registrierung der Nutzer |  |  |  |
| 1.1 | User Story 1 | User Story | Lorenz Wollring |  |
| 1.3 | Registrierungs-Fenster | Papierprototyp | Lorenz Wollring |  |
| 1.4 | Nutzerkontoerstellung | MSC | Lorenz Wollring |  |
| 2. | Filme Automatisiert anlegen |  |  |  |
| 2.1 | User Story 2 | User Story | Felix Bernardi |  |
| 2.2 | Login-Fenster | Papierprototyp | Felix Bernardi |  |
| 2.3 | Nutzerkontologin | MSC | Felix Bernardi |  |
| 3. | Klassendiagramm Registrierung/Login |  |  |  |
| 3.1 | User Story 3 | User Story | Raphael Kruska |  |
| 3.2 | Klassendiagramm Login | UML | Raphael Kruska |  |
| 3.3 | Klassendiagramm Registrierung | UML | Raphael Kruska |  |
| 4. | Klassendiagramm Anlegen von Filmen/Automatisierung |  |  |  |
| 4.1 | User Story 4 | User Story | Thanansaiyan Pushparajah |  |
| 4.2 | Klassendiagramm anlegen von Filmen | UML | Thanansaiyan Pushparajah |  |
| 4.3 | Klassendiagramm automatisierte anlegen von Filmen | UML | Thanansaiyan Pushparajah |  |
| 5 | Anlegen von Filmen |  |  |  |
| 5.1 | User Story 5 | User Story | David Rumpf |  |
| 5.2 | Anlege-Fenster | Papierprototyp | David Rumpf |  |
| 5.3 | Anlegen eines Filmes | MSC | David Rumpf |  |
| 6. | Filme Automatisiert anlegen |  |  |  |
| 6.1 | User Story 6 | User Story | Tobias Zettelmeyer |  |
| 6.2 | Maske für das Automatisierte anlegen | Papierprototyp | Tobias Zettelmeyer |  |
| 6.3 | Der Prozess dahinter | MSC | Tobias Zettelmeyer |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Artefakte | Art des Artefakts | Verantwortlicher | Status |
| 1. | Projektorganisation |  |  |  |
| 1.1 | Übernahme einer agilen Softwareentwicklungsmethode 'Scrum' | Projektmanagement | Felix Bernardi |  |
| 1.2 | Prüfung auf Einhaltung dieser Methodik | Projektmanagement | Felix Bernardi |  |
| 2. | Erstellung einer Client Server Umgebung |  |  |  |
| 2.1 | Aufsetzten eines Servers für Server Client Kommunikation | Serverarchitektur |  |  |
| 2.2 | Datenbankserver aufsetzten | Datenbank |  |  |
| 3 | Registrierung der Systemadministratoren |  |  |  |
| 3.1 | Registrierungsmaske | Papierprototyp |  |  |
| 3.3 | Tabellenerstellung | Tabellen |  |  |
| 3.4 | Übermittlung der Daten an die Datenbank (Check auf Persistenz) | Funktion |  |  |
| 4. | Login eines Systemadministrators |  |  |  |
| 4.1 | Login Interface | Papierprototyp |  |  |
| 4.2 | Datenabgleich mit der Datenbank | Funktionalität |  |  |
| 5. | Filme Manuell anlegen |  |  |  |
| 5.1 | Anlegungsmaske | Papierprototyp |  |  |
| 5.2 | Rechtesystem | Funktionalität |  |  |
| 5.2.1 | Tabellenerstellung | Tabellen |  |  |
| 5.2.2 | Anlegen der Rechte | Datenpflege |  |  |
| 5.3 | Tabellenerstellung | Tabellen |  |  |
| 5.4 | Editierungsfunktionalität | Funktion |  |  |
| 5.5 | Löschfunktionalität | Funktion |  |  |
| 6. | Main Page |  |  |  |
| 6.1 | Indexmaske | Papierprototyp |  |  |
| 6.2 | Funktionalität zum laden der Daten aus der Datenbank | Funktion |  |  |
| 7.0 | Filme automatisiert anlegen |  |  |  |
| 7.1 | Frontendmaske zum ausführen des Webscrapers | Papierprototyp |  |  |
| 7.2 | Entwicklung des Webscrapers | Funktion |  |  |
| 7.3 | Daten in die Datenbank übermitteln | Tabellenpflege |  |  |

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone:  Als <Rolle> möchte ich <Ziel> [, um/sodass <Nutzen>]  (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** | <Einschätzung der Zeit, die benötigt wird, um die Userstory zu implementieren> |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung z.B. hoch, mittel niedrig> |
| **Autor** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

**Schlechtes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** | Ich möchte, dass ich rechtzeitig informiert werde, wenn ein Patient einen Termin nicht wahrnimmt. |
| **Geschätzter Realisierungsaufwand** | 1337 |
| **Priorität** | - |
| **Autor** | Emmett Brown, Rick Sanchez, Amelia Pond |
| **Abhängigkeiten zu anderen User Stories** |  |

**Gutes Beispiel:**

|  |  |
| --- | --- |
| **User Story-ID** | 1.6 |
| **User Story-Beschreibung** | Als Arzt möchte ich mindesten fünf Minuten vor dem Termin informiert werden, wenn ein Patient einen Termin nicht wahrnimmt, sodass ich andere Patienten vorziehen kann. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor** | Emmett Brown |
| **Abhängigkeiten zu anderen User Stories** | 1.3, 1.5 |

## 

Papierprototypen

Das Erstellen eines Papierprototypen dient als Methode des Brainstormings, Designs, Herstellens, Testens und des Kommunizierens von Benutzer Interfaces.

## Szenarien (MSCs)

MSC-Diagramme inkl. Bezeichner

**Hauptszenario:** Typische Folge von Interaktionsschritten zur Erreichung des Ziels bzw. der Ziele einer oder mehrerer User Stories

**Alternativszenario:** Alternative Interaktionsschritte, die das Hauptszenario (oder Teile davon) ersetzen und ebenfalls zur Erfüllung der Ziele hinter den User Stories führen

**Ausnahmeszenario:** Interaktionen in Ausnahmefällen, die dazu führen, dass nicht alle Ziele hinter den abgebildeten User Stories erreicht werden

Szenarien dienen als Grundlage zur Definition von Testfällen

Erläuterung der MSCs (beispielsweise getroffene Annahmen)

## Strukturdiagramm (Klassendiagramm)

Ein Strukturdiagramm dient der grafischen Darstellung von Klassen, Schnittstellen und deren Beziehungen. Es hilft dabei, Quellcode und Implementierungsarbeiten zu strukturieren bevor diese starten und ermöglicht eine Aufteilung der Programmieraufgaben.

## Funktionalitätsplanung

Anhand der Funktionalitätsplanung werdet Ihr geprüft. Diese sollte dementsprechend **immer** aktuell gehalten werden und pro Funktionalität **nur einen** Verantwortlichen enthalten. Als Quellcodereferenz solltet Ihr immer euer Package, eure Klasse und die dazugehörige Methode angeben. ggf. könnt Ihr auch Zeilenangaben machen.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Route** |  |  |  |  |  |  |
| 1.1 | Suche nach POI/ Straßen | 1 Tag | Max Muster | 1.2 |  |  | fertig |
| 1.2 | Erstellen von Routen | 2 Tage | Max Muster | 1.3, 1.4 |  |  | In Bearbeitung |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** | **Anzeige** |  |  |  |  |  |  |
| 2.1 | Lokales Speichern der Routen | 4 Tage | Anja Muster | 2.4, 2,7 |  |  | fertig |
| … |  |  |  |  |  |  |  |

## Systemtests

Systemtests sind Tests des Gesamtsystems gegen die Anforderungen nach erfolgreicher Integration. Eingaben und Sollverhalten werden dabei aus der Anforderungs-spezifikation abgeleitet.

Die Systemtests werden von Eurer Parallelgruppe spezifiziert und durchgeführt, daher ist dieser Bereich von den Mitgliedern der Parallelgruppe auszufüllen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **X** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **X** |
| **Testurteil** | Test nicht bestanden. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 03.03.2019 | | |
| **Tester** | Martina Musterfrau | | |
| **SW-Version** | V 0.1.2 | | |
| **Vorbedin-gung(en)** | Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 | Der Benutzer gibt den Benutzername „Max Mustermann“ auf der Tastatur ein. | Das System zeigt „Max Mustermann“ auf dem Display an. | **√** |
| 2 | Der Benutzer gibt das Passwort „geheim“ auf der Tastatur ein. | Das System zeigt das Passwort durch „\*“-Symbole zensiert an. | **√** |
| 3 | Der Benutzer klickt auf „Anmelden“. | Das System zeigt die Meldung „Anmeldung erfolgreich“ auf dem Display an. | **√** |
| **Nachbe-dingung(en)** | Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert. | | **√** |
| **Testurteil** | Test bestanden. | | |

# Zyklus II

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |

Papierprototypen

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** |  |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |

## Unittests

Unittest, auch Modultests, sind Komponententests. Diese werden in der Softwareentwicklung angewendet, um die funktionalen Einzelteile (Units) von Computerprogrammen zu testen.

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Zyklus III

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Artefakt** | **Art des Artefakts** | **Verantwortlicher** | **Status** |
| **1.** |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| **2.** |  |  |  |  |
|  |  |  |  |  |
| … |  |  |  |  |

## Spezifikationsplanung

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** |  |
| **User Story-Beschreibung** |  |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** |  |
| **Autor** |  |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** |  |

Papierprototypen

## Szenarien (MSCs)

Erläuterung der Szenarien:

## Strukturdiagramm (Klassendiagramm)

## Funktionalitätsplanung

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktiona-litäten** | **Verknüpfte**  **User-Stories** | **Quellcode-**  **referenz** | **Status** |
| **1.** |  |  |  |  |  |  |  |
| 1.1 |  |  |  |  |  |  |  |
| 1.2 |  |  |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |  |
| **2.** |  |  |  |  |  |  |  |
| 2.1 |  |  |  |  |  |  |  |
| … |  |  |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| **…** |  |  |  |

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | | |
| **Tester** |  | | |
| **SW-Version** |  | | |
| **Vorbedin-gung(en)** |  | | |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | **√ / X** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| **Nachbe-dingung(en)** |  | |  |
| **Testurteil** |  | | |

# Nutzerhandbuch

## Technische Anforderungen

Technische Mindestanforderungen, welche das Programm benötigt, um wie gewünscht bedienbar zu sein.

## Installationsanleitung

Genaue Erläuterung, wie das entwickelte Programm vollkommen funktionsfähig auf einem Rechner in Betrieb genommen werden kann.

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte Programm zu bedienen ist.