

# **SEP Projektmappe**

***PROJEKTMAPPE DES PROJEKTES***

## **SEP-Drive**

### **Dokumentation des Projektes**

Gruppe C:

Sara Akgüvercin

Aya Almallal

Rawan Almallal

Bashar Herzallah

Ibrahim Nagi Ahmed Mohamed Sayed

Rias Safi

Melike Ünver

## 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
Zyklus I.....	5
Spezifikationsplanung.....	5
User-Stories .....	7
Papierprototypen .....	8
Strukturdiagramme (Komponenten- und Klassendiagramme).....	8
Verhaltensdiagramme (Kommunikationsdiagramme).....	8
Funktionalitätsplanung.....	9
Systemtests .....	10
Zyklus II.....	12
Spezifikationsplanung.....	12
User-Stories .....	12
Papierprototypen .....	13
Strukturdiagramme (Komponenten- und Klassendiagramme).....	13
Verhaltensdiagramme (Kommunikationsdiagramme).....	13
Funktionalitätsplanung.....	14
Modultests .....	15
Systemtests .....	15
Zyklus III.....	16
Spezifikationsplanung.....	16
User-Stories .....	16
Papierprototypen .....	17
Strukturdiagramme (Komponenten- und Klassendiagramme).....	17
Verhaltensdiagramme (Kommunikationsdiagramme).....	17
Funktionalitätsplanung.....	18
Modultests .....	19
Systemtests .....	19
Nutzerhandbuch.....	20
Technische Anforderungen .....	20
Installationsanleitung .....	20
Bedienungsanleitung.....	20

# 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.*

# 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 Komponentendiagramm, jedes Klassendiagramm, jedes Kommunikationsdiagramm 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	Artefakt	Art des Artefakts	Verantwortlicher	Status
<b>1.</b>	Registrierung der Nutzer			
1.1	Registrierung (Kunde/Fahrer)	User Story	Sara Akgüvercin	offen
1.2	Registrierungs-Fenster	Papierprototyp	Sara Akgüvercin	offen
<b>2.</b>	Login der Benutzer			
2.1	Login	User Story	Aya Almallal	offen
2.2	Zwei-Faktor-Authentifizierung	User Story	Aya Almallal	offen
2.3	Login-Fenster	Papierprototyp	Aya Almallal	offen
2.4	Zwei-Faktor-Authentifizierung- Fenster	Papierprototyp	Aya Almallal	offen
<b>3.</b>	Benutzerprofile			
3.1	Benutzerprofil (Kunde/Fahrer)	User Story	Rawan Almallal	offen
3.2	Profilsuche	User Story	Ibrahim Sayed	offen
3.3	Profilsuche-Fenster	Papierprototyp	Ibrahim Sayed	offen
3.4	Benutzerprofil-Fenster (Kunde)	Papierprototyp	Rawan Almallal	offen
3.5	Benutzerprofil-Fenster (Fahrer)	Papierprototyp	Rawan Almallal	offen
<b>4.</b>	Kartenvisualisierung von Routen			
4.1	Kartenanzeige & Interaktion	User Story	Sara Akgüvercin	offen
4.2	Karten-Fenster	Papierprototyp	Sara Akgüvercin	offen

<b>5.</b>	Fahranfragen			
5.1	Fahranfrage erstellen	User Story	Melike Ünver	offen
5.2	Aktive Fahranfragen sehen und löschen	User Story	Melike Ünver	offen
5.3	Aktive Fahranfragen-Fenster	Papierprototyp	Melike Ünver	offen
5.4	Fahranfrage-Fenster	Papierprototyp	Melike Ünver	offen
<b>6.</b>	Backend Architektur			
2.1	Komponentendiagramm Backend	Komponentendiagramm	Bashar Herzallah	offen
2.2	Klassendiagramm Backend	Klassendiagramm	Rias Safi	offen
2.3	Kommunikationsdiagramm Backend	Kommunikationsdiagramm	Ibrahim Sayed	offen

# User-Stories

## Template:

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

## Schlechtes Beispiel:

<b>User Story-ID</b>	
<b>User Story-Beschreibung</b>	Ich möchte ich rechtzeitig informiert werden, wenn ein Patient einen Termin nicht wahrnimmt.
<b>Geschätzter Realisierungsaufwand</b>	1337
<b>Priorität</b>	-
<b>Autor</b>	Emmett Brown, Rick Sanchez, Amelia Pond
<b>Abhängigkeiten zu anderen User Stories</b>	

## Gutes Beispiel:

<b>User Story-ID</b>	1.6
<b>User Story-Beschreibung</b>	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.
<b>Geschätzter Realisierungsaufwand</b>	2 Tage
<b>Priorität</b>	Hoch
<b>Autor</b>	Emmett Brown
<b>Abhängigkeiten zu anderen User Stories</b>	1.3, 1.5

## Papierprototypen

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

## Strukturdiagramme (Komponenten- und Klassendiagramme)

Im SEP soll die statische Struktur des Systems mittels Komponenten- und Klassendiagramme modelliert werden. Ein Komponenten- und Klassendiagramme dienen der grafischen Darstellung von Komponenten/Klassen, Schnittstellen und deren Beziehungen. Die Diagrammtypen helfen dabei, Quellcode und Implementierungsarbeiten zu strukturieren, bevor diese starten und ermöglicht eine Aufteilung der Programmieraufgaben.

## Verhaltensdiagramme (Kommunikationsdiagramme)

Im SEP soll das dynamische Verhalten des Systems mittels Kommunikationsdiagramme modelliert werden. Ein Kommunikationsdiagramme ermöglicht die grafische Darstellung des Nachrichtenaustausches zwischen Systemobjekten. Systemobjekte können Komponenten im Komponentendiagramm und Klassen im Klassendiagramm sein. Kommunikationsdiagramme zielen darauf ab, die Zusammenarbeit der Systemobjekte darzustellen

## 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	Verantwortlicher	Abhängige Funktionalitäten	Verknüpfte User-Stories	Quellcode-referenz	Status
<b>1.</b>	<b>Registrierung von Admins</b>					
1.1	Registrierungs view im Frontend	Max Muster	1.2			Fertig
1.2	Datenbank-modell des Admins	Max Muster	1.3, 1.4			WIP
1.2.1	....					
<b>2.</b>	<b>Anliegen von Liegen</b>					
2.1	Ligaübersicht im Frontend	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.

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

<b>Datum</b>	03.03.2019		
<b>Tester</b>	Martina Musterfrau		
<b>SW-Version</b>	V 0.1.2		
<b>Vorbedin-gung(en)</b>	Nutzer „Max Mustermann“ ist am System mit Passwort „geheim“ registriert		
<b>Schritt</b>	<b>Aktion (User)</b>	<b>Erwartete Reaktion (System)</b>	<b>V / X</b>
1	Der Benutzer gibt den Benutzernamen „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.	✓
<b>Nachbe-dingung(en)</b>	Nutzer ist am System angemeldet, Anmeldezeitpunkt ist im System gespeichert.		
<b>Testurteil</b>	<b>Test bestanden.</b>		

## Zyklus II

### Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
...				
...				
...				

### User-Stories

Template:

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

Papierprototypen

Strukturdiagramme (Komponenten- und Klassendiagramme)

Verhaltensdiagramme (Kommunikationsdiagramme)

## Funktionalitätsplanung

ID	Funktionalität	Verantwortlich er	Abhängige Funktionalitä ten	Verknüpfte User-Stories	Quellcode-referenz	Status
<b>1.</b>						
1.1						
1.2						
1.2.1	....					
<b>2.</b>						
2.1						
...						

## Modultests

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)	V / X
1			
2			
3			
Nachbe-dingung(en)			
Testurteil			

## Zyklus III

### Spezifikationsplanung

ID	Artefakt	Art des Artefakts	Verantwortlicher	Status
...				
...				
...				

### User-Stories

Template:

<b>User Story-ID</b>	
<b>User Story-Beschreibung</b>	
<b>Geschätzter Realisierungsaufwand</b>	
<b>Priorität</b>	
<b>Autor</b>	
<b>Abhängigkeiten zu anderen User Stories</b>	

Papierprototypen

Strukturdiagramme (Komponenten- und Klassendiagramme)

Verhaltensdiagramme (Kommunikationsdiagramme)

## Funktionalitätsplanung

ID	Funktionalität	Verantwortlich er	Abhängige Funktionalitä ten	Verknüpfte User-Stories	Quellcode-referenz	Status
<b>1.</b>						
1.1						
1.2						
1.2.1	....					
<b>2.</b>						
2.1						
...						

## Modultests

ID	Getestete Funktionalität	Quellcode Referenz	Status

## Systemtests

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

# Nutzerhandbuch

## Technische Anforderungen

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

## Installationsanleitung

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

## Bedienungsanleitung

Genaue Erläuterung, wie das entwickelte System zu bedienen ist.