### SEP Projektmappe

**Projektmappe des Projektes**

Swat Engagement Pheretima

**Dokumentation des Projektes**

Gruppe L:

Tarik Karaca

Julian Flieter

Jan Voigtländer

Ibrahim Kekec

Steve Nemangou

Inhaltsverzeichnis

[SEP Projektmappe 1](#_Toc513743359)

[Projektbeschreibung 3](#_Toc513743360)

[Iteration I 5](#_Toc513743361)

[User-Stories 5](#_Toc513743362)

[MSCs 10](#_Toc513743363)

[UML Klassendiagramm 15](#_Toc513743364)

[Funktionalitätsplanung 15](#_Toc513743365)

[Iteration II 17](#_Toc513743366)

[User-Stories 17](#_Toc513743367)

[MSCs 17](#_Toc513743368)

[UML Klassendiagramm 17](#_Toc513743369)

[Funktionalitätsplanung 17](#_Toc513743370)

[Unittests 17](#_Toc513743371)

[Systemtests 17](#_Toc513743372)

[Iteration III 19](#_Toc513743373)

[User-Stories 19](#_Toc513743374)

[MSCs 19](#_Toc513743375)

[UML Klassendiagramm 19](#_Toc513743376)

[Funktionalitätsplanung 19](#_Toc513743377)

[Unittests 19](#_Toc513743378)

[Systemtests 19](#_Toc513743379)

[Nutzerhandbuch 21](#_Toc513743380)

[Technische Anforderungen 21](#_Toc513743381)

[Installationsanleitung 21](#_Toc513743382)

[Bedienungsanleitung 21](#_Toc513743383)

## Projektbeschreibung

**Einleitung**

In dem zu entwickelnden Spiel sollen bis zu fünf Spieler gegeneinander spielen können, indem jeweils ein Spieler ein Team aus bewaffneten Würmern steuern. Das Ziel jedes Teams ist es dabei, alle anderen Teamgegner zu besiegen. Ein Team ist besiegt, wenn alle Würmer des Teams kampfunfähig sind, entweder weil sie zu viel Beschuss einstecken mussten, oder weil sie aus der Spielkarte herausgefallen sind. Das Spielgeschehen soll zweidimensional von der Seite gezeigt werden. Außerdem soll das Spiel an einem Computer gespielt werden, an dem die Spieler nacheinander jeweils eine Spielrunde spielen. In einer Spielrunde kann der Spieler, der am Zug ist, einen seiner Würmer bewegen und einmal eine Waffe mit diesem Wurm benutzen.

**Iteration I**

In der ersten Iteration soll eine erste Version des Spiels mit reduziertem Umfang entwickelt werden. Zunächst sollen nur zwei Spieler, die jeweils nur einen Wurm steuern, gegeneinander spielen. Die Landschaft der Spielkarte soll aus einer flachen Ebene bestehen, auf der die Würmer sich nach rechts und links bewegen und springen können. Jedem Spieler steht nur eine Waffe zur Verfügung, die Projektile abschießt. Vor dem Schuss soll der Spieler den Winkel bestimmen können, in dem die Waffe schießt.

**Iteration II**

In dieser Iteration wird das Grundgerüst des Spiels vervollständigt, vor allem sollen Wind und Gravitation für die Spielobjekte simuliert werden. Die Stärke und Richtung des Windes soll sich in jeder Runde zufällig verändern und so die Flugbahn der abgeschossenen Projektile beeinflussen. Außerdem sollen die Projektile und Würmer von einer simulierten Gravitation beeinflusst werden. Fallen Würmer unter eine bestimmte Grenze in der Spielwelt, werden sie kampfunfähig. Diese Grenze soll in jeder Runde leicht steigen. Das Spiel soll auf vier verschiedenen Karten, die sich im Aufbau ihrer Landschaft unterscheiden, gespielt werden können. Diese Landschaften sollen komplexer sein als eine Ebene. Sie sollen außerdem von den Waffen der Würmer zerstört werden können. Den Würmern soll eine Auswahl von drei verschiedenen Waffen zur Verfügung stehen, die jeweils andere Projektile verschießen. Die Projektile sollen sich in dem Schaden unterscheiden, den sie an getroffenen Würmern anrichten, sie sollen verschieden stark von Wind und Gravitation beeinflusst werden, sie sollen die Umgebung auf verschiedene Arten zerstören und sie sollen verschieden aussehen. Außerdem sollen die beiden Spieler jeweils ein Team aus bis zu fünf Würmern steuern. Die Spieler sind weiterhin abwechselnd dran und können weiterhin nur mit einem Wurm in ihrem Zug laufen und schießen. Wann immer ein Wurm stirbt, soll ein Replay der letzten Sekunden angezeigt werden. Zuletzte soll eine weitere Waffe eingeführt werden. Diese Waffe soll Würmern über den Verlauf von mehreren Runden Schaden zufügen. Sofern ein von dieser Waffe getroffener Wurm in die Nähe anderer Würmer gelangt, sollen diese Würmer ebenfalls von dem Effekt der Waffe betroffen werden.

**Iteration III**

In der letzten Iteration wird der Umfang des Spiels erweitert. Bis zu fünf Spieler sollen jeweils mit einem Team gegeneinander spielen könne. Jeder Spieler kann seinem Team und den Würmern aus seinem Team individuelle Namen geben, die dann im Spiel angezeigt werden. Dazu soll der Schaden, den die einzelnen Würmer schon genommen haben, visualisiert werden. Anstatt zu schießen soll jeder Spieler, wenn er am Zug ist, auch mit einem seiner Würmer eine Sonderaktion ausführen können. Die Anzahl

Sonderaktionen, die jeder Spieler pro Spiel ausführen darf, soll dabei begrenzt sein. Die möglichen Sonderaktionen sind einen Wurm Teleportieren, eine Miene legen, einen Luftschlag anfordern und ein Standgeschütz aufstellen. Das Standgeschütz soll in jeder Runde automatisch auf andere Teams feuern. Zuletzt soll die bearbeitende Gruppe sich selber drei weitere Features für das Spiel ausdenken. Diese Features sollen zum Thema der Aufgabenstellung passen, und vor der Umsetzung mit dem Gruppenbetreuer abgesprochen werden.

# Iteration I

User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | Bewegung |
| **User Story-Beschreibung** | Als Spieler möchte ich meinen Charakter steuern können, sodass er sich auf der Landschaft frei bewegen kann. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Hoch |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel aktualisiert Charakterposition * 5. Spiel gibt visuelle Rückmeldung an Spieler   Alternativszenarien:  Ausnahmeszenarien:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel stellt Ungültigkeit der Bewegung fest * 5. Spiel aktualisiert Charakterposition nicht |

|  |  |
| --- | --- |
| **User Story-ID** | Springen |
| **User Story-Beschreibung** | Als Spieler möchte ich mit meinem Charakter springen können, um Hindernisse zu überwinden. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | Bewegung |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel aktualisiert Charakterposition * 5. Spiel gibt visuelle Rückmeldung an Spieler   Alternativszenarien:  Ausnahmeszenarien:   * 1. Spieler drückt Bewegungstaste * 2. Spiel erhält Tastencode * 3. Spiel wertet Bewegung aus * 4. Spiel stellt Ungültigkeit der Bewegung fest * 5. Spiel aktualisiert Charakterposition nicht |

|  |  |
| --- | --- |
| **User Story-ID** | Landschaft-IT1 |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass die Landschaft aus einer flachen Ebene besteht, damit die Charaktersteuerung einfach ist. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Hoch |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spielansicht |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass das Spielgeschehen zweidimensional von der Seite dargestellt wird, um die Spielkomplexität gering zu halten. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Hoch |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Spieleranzahl |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass zwei Spieler gegeneinander spielen, um das Spiel übersichtlich und den Spielfluss hoch zu halten. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Hoch |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Rundenbasiert |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Spieler jeweils nacheinander am Zug sind, sodass ein rundenbasiertes Spiel entsteht, damit faire Abwechslung herrscht. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Hoch |
| **Autor(en)** | Ibrahim Kekec |
| **Abhängigkeiten zu anderen User Stories** | Spieleranzahl |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler ist am Zug * 2. Spieler bewegt Charakter * 3. Spieler greift an * 4. Anderer Spieler ist am Zug   Alternativszenarien:   * 1. Spieler ist am Zug * 2. Spieler greift an * 3. Anderer Spieler ist am Zug   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Kampfunfähigkeit |
| **User Story-Beschreibung** | Als Spieler möchte ich, dass Charaktere kampfunfähig werden, wenn sie zu viel Schaden einstecken oder aus der Spielkarte herausfallen. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Hoch |
| **Autor(en)** | Jan Voigtländer |
| **Abhängigkeiten zu anderen User Stories** | Bewegung, Landschaft-IT1, Angreifen |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Projektil trifft Charakter * 2. Spiel wertet Schaden aus * 3. Spiel überprüft verbleibendes Leben * 4. Spiel erklärt Charakter als kampfunfähig   Alternativszenarien:   * 1. Charakter bewegt sich * 2. Spiel überprüft Position * 3. Spiel stellt fest, dass Charakter aus der Karte gefallen ist * 4. Spiel erklärt Charakter als kampfunfähig   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Gewinnen |
| **User Story-Beschreibung** | Als Spieler möchte ich die Partie gewinnen, wenn alle gegnerischen Charaktere kampfunfähig sind. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Mittel |
| **Autor(en)** | Ibrahim Kekec |
| **Abhängigkeiten zu anderen User Stories** | Kampfunfähigkeit |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spiel stellt Kampfunfähigkeit eines Charakters fest * 2. Spiel bestimmt Gewinner * 3. Spiel beendet Runde und zeigt Gewinner an   Alternativszenarien:  Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Rundenablauf |
| **User Story-Beschreibung** | Als Spieler möchte ich in meinem Zug meinen Charakter bewegen und mit einer Waffe einmal schießen können, um mehr Möglichkeiten des Spielens zu haben. |
| **Geschätzter Realisierungsaufwand** | Mittel |
| **Priorität** | Mittel |
| **Autor(en)** | Julian Flieter |
| **Abhängigkeiten zu anderen User Stories** | Bewegung, Angreifen, Rundenbasiert |
| **Zugehörige Szenarien** | Hauptszenario:  Alternativszenarien:  Ausnahmeszenarien: |

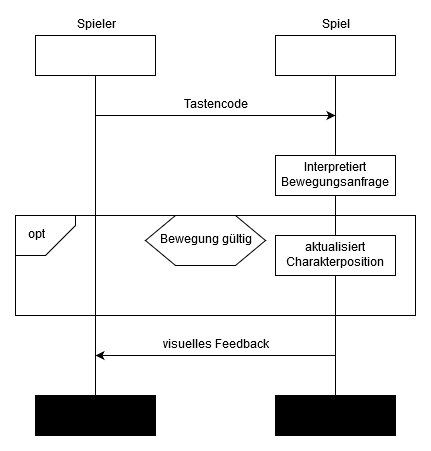
|  |  |
| --- | --- |
| **User Story-ID** | Angreifen |
| **User Story-Beschreibung** | Als Spieler möchte ich mit einer Waffe zunächst den Abschusswinkel bestimmen und danach Projektile abschießen, um gegnerischen Charakteren gezielt Schaden zufügen zu können. |
| **Geschätzter Realisierungsaufwand** | Hoch |
| **Priorität** | Mittel |
| **Autor(en)** | Tarik Karaca |
| **Abhängigkeiten zu anderen User Stories** | keine |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Spieler wählt angreifen * 2. Spiel zeigt Zielrichtung an * 3. Spieler wählt Zielrichtung * 4. Spieler schießt * 5. Spiel erzeugt Projektil   Alternativszenarien:   * 1. Spieler wählt angreifen * 2. Spiel zeigt Zielrichtung an * 3. Spieler schießt * 4. Spiel erzeugt Projektil   Ausnahmeszenarien: |

|  |  |
| --- | --- |
| **User Story-ID** | Neuer Spieler am Zug |
| **User Story-Beschreibung** | Als Spieler möchte ich deutlich darüber informiert werden welcher Spieler am Zug ist, um leichter den Überblick behalten zu können. |
| **Geschätzter Realisierungsaufwand** | Gering |
| **Priorität** | Mittel |
| **Autor(en)** | Steve Nemangou |
| **Abhängigkeiten zu anderen User Stories** | Rundenbasiert |
| **Zugehörige Szenarien** | Hauptszenario:   * 1. Neuer Spieler ist am Zug * 2. Spiel gibt visuelle Rückmeldung in Form von Text * 3. Spiel ändert Aussehen des am Zug befindlichen Charakters   Alternativszenarien:  Ausnahmeszenarien: |

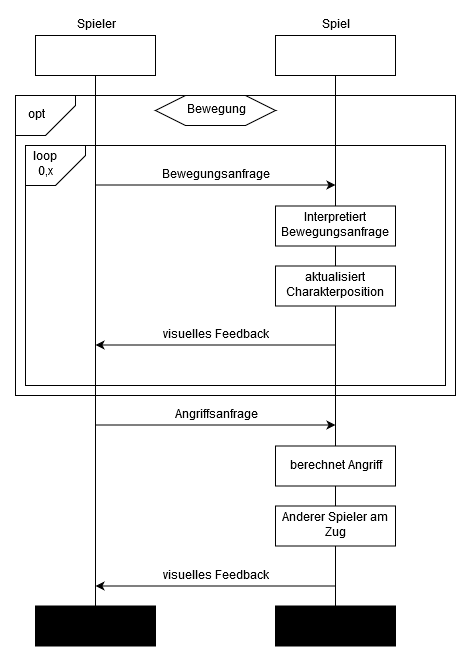
## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

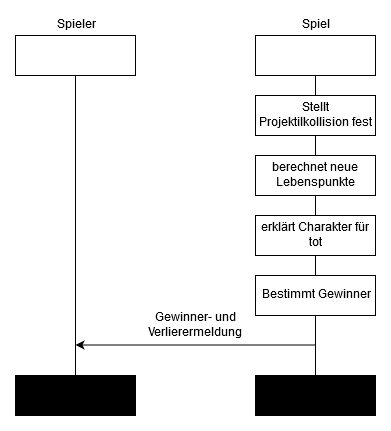
Bewegung: (*Jan Voigtländer*)



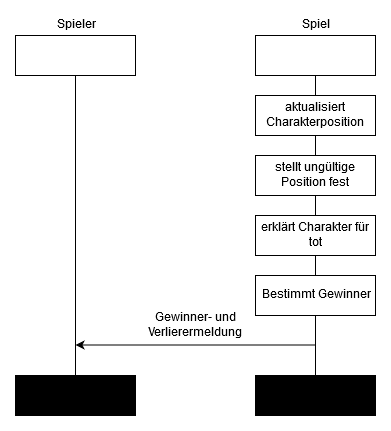
Rundenbasiert: (*Ibrahim Kekec*)



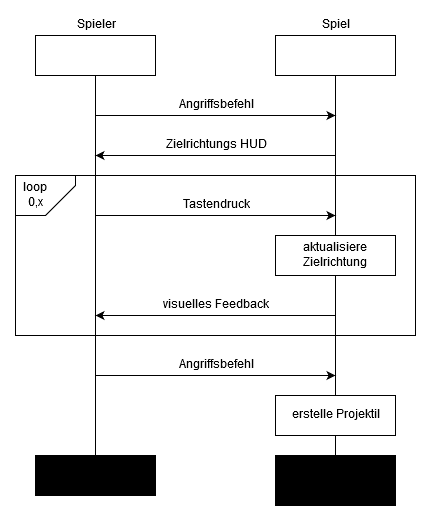
KIA Angriff: (*Julian Flieter*)



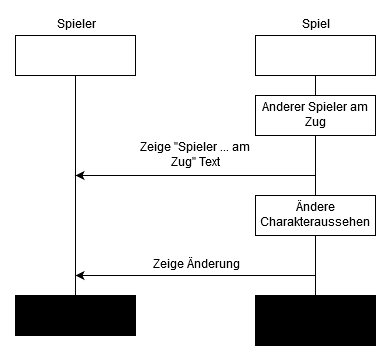
KIA Angriff (Alternativ): (*Ibrahim Kekec)*



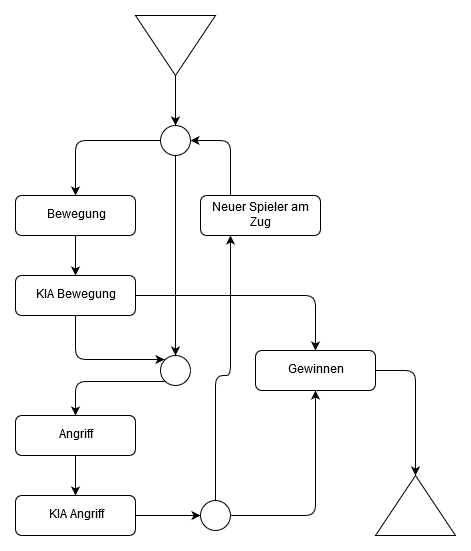
Angriff: (*Tarik Karaca*)



Neuer Spieler am Zug: (*Julian Flieter)*



hMSC: (*Jan Voigtländer)*



## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **PlayScreen** |  |  |  |  |  |
| 1.1 | Verwaltung des Spielablaufs | 1 Tag | Tarik Karaca |  | PlayScreen | Fertig |
| 1.2 | Darstellung der Spielwelt | 2 Tage | Tarik Karaca |  | PlayScreen | Fertig |
| 1.3 | Eingabeverwaltung | 1 Tag | Tarik Karaca |  | PlayScreen | Fertig |
| 1.4 | Initialisierungen der Spielwelt | 2 Tage | Tarik Karaca |  | PlayScreen | Fertig |
| 1.5 | Verwaltung der Spielwelt | Halber Tag | Tarik Karaca |  | PlayScreen | Fertig |
| **2.** | **GameOverScreen** |  |  |  |  |  |
| 2.1 | Visualisierung des Spielendes | 3 Tage | Ibrahim Kekec |  | GameOverScreen | Fertig |
| **3.** | **PlayUILayer** |  |  |  |  |  |
| 3.1 | Graphische Oberfläche | 2 Tage | Steve Nemangou |  | PlayUILayer | Fertig |
| **4.** | **CollisionHandler** |  |  |  |  |  |
| 4.1 | Verwaltungen von Kollisionen | 5 Tage | Julian Flieter |  | CollisionHandler | Fertig |
| **5.** | **SEPGame** |  |  |  |  |  |
| 5.1 | Initialisierungen des Spiels | 1 Tag | Julian Flieter |  | SEPGame | Fertig |
| **6.** | **AnimatedSprite** |  |  |  |  |  |
| 6.1 | Darstellung der Animationen | 2 Tage | Tarik Karaca |  | AnimatedSprite | Fertig |
| 6.2 | Initialisierung von Animationen | 1 Tag | Tarik Karaca |  | AnimatedSprite | Fertig |
| **7.** | **GameCamera** |  |  |  |  |  |
| 7.1 | Verwaltung der Spielansicht | 1 Tag | Julian Flieter |  | GameCamera | Fertig |
| 7.2 | Verfolgung von Spielobjekten | 2 Std. | Julian Flieter |  | GameCamera | Fertig |
| **8.** | **Worm** |  |  |  |  |  |
| 8.1 | Behandlung von Charakterbewegungen | Halber Tag | Jan Voigtländer |  | Worm | Fertig |
| 8.2 | Verwaltung von Spielercharakteren | 1 Tag | Jan Voigtländer |  | Worm | Fertig |
| 8.3 | Visualisierung von Spielercharakteren | 3 Tage | Jan Voigtländer |  | Worm | Fertig |
| **9.** | **Ground** |  |  |  |  |  |
| 9.1 | Initialisierung und Visualisierung der Landschaft | 6 Tage | Steve Nemangou |  | Ground | Fertig |
| **10.** | **Projectile** |  |  |  |  |  |
| 10.1 | Initialisierung und Visualisierung der Projektile | 6 Tage | Steve Nemangou |  | Projectile | Fertig |
| **11.** | **ShotDirectionIndicator** |  |  |  |  |  |
| 11.1 | Initialisierung und Visualisierung des Abschusswinkels | 4 Tage | Ibrahim Kekec |  | ShotDirectionIndicator | Fertig |
| 11.2 | Berechnen des Abschusswinkels | 2 Tage | Ibrahim Kekec |  | ShotDirectionIndicator | Fertig |
| **12.** | **HealthBar** |  |  |  |  |  |
| 12.1 | Visualisierung der Lebensanzeige | 1 Tag | Jan Voigtländer |  | HealthBar | Fertig |

# Iteration II

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung> |
| **Autor(en)** | Max Mustermann |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

Template:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **…** |  |  |  |  |  |  |

## Unittests

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

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 05.05.2017 | | |
| **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. | | |

# Iteration III

## User-Stories

Template:

|  |  |
| --- | --- |
| **User Story-ID** | <Eindeutiger Identifizierer> |
| **User Story-Beschreibung** | <Text der User Story mittels Satzschablone (s. Foliensatz „Anforderungen“)> |
| **Geschätzter Realisierungsaufwand** |  |
| **Priorität** | <Wichtigkeit der User Story hinsichtlich der Aufgabenstellung> |
| **Autor(en)** | Max Mustermann |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

## MSCs

MSC-Diagramme inkl. Bezeichner und, falls nötig, erläuterndem Text (beispielsweise getroffene Annahmen)

## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

Vorlage:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **…** |  |  |  |  |  |  |

## Unittests

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

## Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | 05.05.2017 | | |
| **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. | | |

# 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.