SEP Projektmappe

**Projektmappe des Projektes**

SemestervErwaltungsPlan

**Dokumentation des Projektes**

Gruppe E:

Johannes Morzeck

Tobias Sieber

Merlin von Rössing

Kaan Erdogan

Dave Makila

Alisan Gündogan

Andreas Häusler

Inhaltsverzeichnis

[Projektbeschreibung 3](#_Toc529038155)

[Iteration I 5](#_Toc529038156)

[User-Stories: 5](#_Toc529038157)

[MSCs 12](#_Toc529038158)

[UML Klassendiagramm 17](#_Toc529038159)

[Funktionalitätsplanung 18](#_Toc529038160)

[Iteration II 20](#_Toc529038161)

[User-Stories 20](#_Toc529038162)

[MSCs 20](#_Toc529038163)

[UML Klassendiagramm 20](#_Toc529038164)

[Funktionalitätsplanung 20](#_Toc529038165)

[Unittests 20](#_Toc529038166)

[Systemtests 20](#_Toc529038167)

[Iteration III 22](#_Toc529038168)

[User-Stories 22](#_Toc529038169)

[MSCs 22](#_Toc529038170)

[UML Klassendiagramm 22](#_Toc529038171)

[Funktionalitätsplanung 22](#_Toc529038172)

[Unittests 22](#_Toc529038173)

[Systemtests 22](#_Toc529038174)

[Nutzerhandbuch 24](#_Toc529038175)

[Technische Anforderungen 24](#_Toc529038176)

[Installationsanleitung 24](#_Toc529038177)

[Bedienungsanleitung 24](#_Toc529038178)

## Projektbeschreibung

Einleitung:  
Ziel ist die Implementierung eines Programms zur Verwaltung sämtlicher organisatorischer Aufgaben, die im Laufe eines Semesters zur Verwaltung des SEP anfallen. Dafür soll ein Programm entwickelt werden, in welchem die einzelnen Gruppen, Studierenden, Betreuer und sonstige organisatorische Pläne eingebettet sind, um eine reibungslose Organisation der Lehrveranstaltung zu gewährleisten.

**Iteration I:**

In der ersten Iteration werden die grundlegenden Strukturen implementiert, auf deren Grundlage das Programm basiert. Wenn das Programm gestartet wird, soll der Betreuer die Möglichkeit haben, ein entsprechendes Semester auswählen zu können, zu welchem er eine Reihe an Details erhalten möchte, dabei können diese jederzeit editierbar sein. Außerdem lassen sich neue Semester anlegen und auch wieder löschen. Zu jedem Semester soll eine entsprechende Struktur von Klassen, den zu den Klassen gehörigen Gruppen, und den zu den Gruppen gehörigen Studierenden existieren. Des Weiteren soll das Programm nach der Schaffung der grundlegenden Struktur seine Daten atomar, konsistent, isoliert und dauerhaft speichern (ACID-Kriterien) und die Anwendung muss auf mehreren Clients gleichzeitig ausgeführt werden können, ohne die genannten Kriterien dabei zu verletzen.

**Iteration II:**

In der zweiten Iteration soll es zu jedem Semester eine organisatorische Übersicht geben, welche den Wochenplan und den Semesterplan beinhaltet, diese beiden Pläne sollen vom Benutzer angelegt und editiert werden können, um zusätzliche Informationen anzureichern. Basierend auf dem Semesterplan soll für jede Klasse ein Anwesenheitsplan erstellt werden, in dem der Benutzer die Fehlzeiten aller Studenten minutengenau eintragen und übersichtlich einsehen kann. Zu jeder der oben genannten Entitäten (s. Iteration 1) soll der Betreuer die Möglichkeit haben, Notizen hinzufügen zu können. Notizen lassen sich in gut, mittel, schlecht und ohne Zuordnung unterscheiden, dabei unterscheiden sich diese visuell. Darüber hinaus soll dem Benutzer die Möglichkeit geboten werden, ein Benutzerkonto anlegen zu können, auf welches er mit einem zuvor selbst festgelegten Passwort zugreifen kann. Zudem sollen die Benutzerkonten und sämtliche Daten des Programms persistent gespeichert werden können.

**Iteration III:**

In der letzten Iteration wird das Programm um Templates und frei wählbare Zusatzfeatures erweitert. Dem Benutzer soll es möglich sein, einen E-Maildienst mit seinem Benutzerkonto verbinden zu können, sodass er in dieser Iteration automatisiert E-Mails verschicken kann. Der Benutzer soll Templates für das Versenden von E-Mails anlegen können. Außerdem soll der Benutzer die Möglichkeit haben, direkt über das Programm einzelnen Studierenden oder ganzen Gruppen auf Grundlage der Templates E-Mails schreiben zu können. Zuletzt sollen die Gruppenmitglieder drei weitere Zusatzfeatures implementieren, welche zum Thema der Aufgabenstellung passen. Diese Zusatzfeatures sollen zuvor mit dem Betreuer abgesprochen werden.

**Mögliche Zusatzfeatures:**

* Nachrichten versenden (Chat) Registrierte Nutzer können sich gegenseitig Nachrichten senden, diese werden persistent gespeichert. Außerdem kriegt der Empfänger eine visuelle Rückmeldung, falls er eine neue Nachricht erhalten hat.
* E-Mails können mit Anhang versendet werden Beim Versenden von E-Mails können Dateien angehangen werden.
* Verknüpfung mit GitLab, um Metriken auf Gruppen-, Klassen- und Personen-Ebene zu erhalten Metriken sollen sowohl grafisch, als auch textuell angezeigt werden können.
* Favorisieren von Gruppen und Personen Es können ausgewählte Gruppen und/oder Personen favorisiert werden, welche beim Starten der Anwendung, als erstes angezeigt werden sollen.
* Datenbestände können als PDF exportieren werden Alle Datenbestände eines Semesters lassen sich als PDF-Datei exportieren.
* Zurücksetzen der Daten Dieses Zusatzfeature erlaubt es alle Datenbestände, die in der Anwendung gespeichert sind zurückzusetzen.

# Iteration I

## User-Stories:

|  |  |
| --- | --- |
| **User Story-ID** | 001 |
| **User Story-Beschreibung** | Als Betreuer möchte ich mich im System anmelden müssen, um Fremdzugriff auf das System zu vermeiden. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Merlin von Rössing, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | bMSC\_Login |

|  |  |
| --- | --- |
| **User Story-ID** | 002 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine Benachrichtigung bekommen, wenn die von mir eingegebenen Login-Daten nicht korrekt sind, damit weiß, dass der Login-Vorgang fehlgeschlagen ist. |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Merlin von Rössing |
| **Abhängigkeiten zu anderen User Stories** | 001 |
| **Zugehörige Szenarien** | bMSC\_Login\_Feedback |

|  |  |
| --- | --- |
| **User Story-ID** | 003 |
| **User Story-Beschreibung** | Als Betreuer möchte ich ein zentrales Fenster haben, in welchen ich alle nötigen Details von Semestern sehen kann, um eine möglichst hohe Übersichtlichkeit vorzufinden |
| **Geschätzter Realisierungsaufwand** | 12 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | bMSC\_HomeView |

|  |  |
| --- | --- |
| **User Story-ID** | 004 |
| **User Story-Beschreibung** | Als Betreuer möchte ich existierende Semester sehen können, um einen Überblick über die Semesterplanung zu bekommen. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 003 |
| **Zugehörige Szenarien** | bMSC\_ShowSemester |

|  |  |
| --- | --- |
| **User Story-ID** | 005 |
| **User Story-Beschreibung** | Als Betreuer möchte ein Semester auswählen können, um dieses Semester bearbeiten zu können. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 004 |
| **Zugehörige Szenarien** | bMSC\_EditSemester |

|  |  |
| --- | --- |
| **User Story-ID** | 006 |
| **User Story-Beschreibung** | Als Betreuer möchte ich ein neues Semester erstellen können, um noch nichtexistierende Semester anlegen zu können. |
| **Geschätzter Realisierungsaufwand** | 8 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 004,005 |
| **Zugehörige Szenarien** | bMSC\_CreateSemester |

|  |  |
| --- | --- |
| **User Story-ID** | 007 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Semester löschen können, um Duplikate oder falsch erstellte Semester aus dem System entfernen zu können |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | mittel |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 004 |
| **Zugehörige Szenarien** | bMSC\_DeleteSemester |

|  |  |
| --- | --- |
| **User Story-ID** | 008 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine Bestätigung geben müssen, um ein Semester zu löschen, um ein versehentliches löschen zu verhindern. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 007 |
| **Zugehörige Szenarien** | bMSC\_DeleteSemesterFeedback |

|  |  |
| --- | --- |
| **User Story-ID** | 009 |
| **User Story-Beschreibung** | Als Betreuer möchte ich existierende Klassen sehen können, um einen Überblick über die Klassenplanung zu bekommen. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 003 |
| **Zugehörige Szenarien** | bMSC\_ShowGroupage |

|  |  |
| --- | --- |
| **User Story-ID** | 010 |
| **User Story-Beschreibung** | Als Betreuer möchte eine Klasse auswählen können, um diese Klassen weiter zu bearbeiten. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 009 |
| **Zugehörige Szenarien** | bMSC\_EditGroupage |

|  |  |
| --- | --- |
| **User Story-ID** | 011 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine neue Klasse erstellen können, um noch nichtexistierende Klassen anlegen zu können. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 009,010 |
| **Zugehörige Szenarien** | bMSC\_CreateGroupage |

|  |  |
| --- | --- |
| **User Story-ID** | 012 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Klassen löschen können, um Duplikate oder falsch erstellte Klassen aus dem System entfernen zu können |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 009 |
| **Zugehörige Szenarien** | bMSC\_DeleteGroupage |

|  |  |
| --- | --- |
| **User Story-ID** | 013 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine Bestätigung geben müssen, um eine Klasse zu löschen, um ein versehentliches löschen zu verhindern. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 012 |
| **Zugehörige Szenarien** | bMSC\_DeleteGroupageFeedback |

|  |  |
| --- | --- |
| **User Story-ID** | 014 |
| **User Story-Beschreibung** | Als Betreuer möchte ein Gruppen auswählen können, um diese Gruppen weiter zu bearbeiten. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 003 |
| **Zugehörige Szenarien** | bMSC\_EditGroup |

|  |  |
| --- | --- |
| **User Story-ID** | 015 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine neue Gruppe erstellen können, um noch nichtexistierende Gruppe anlegen zu können. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 014 |
| **Zugehörige Szenarien** | bMSC\_CreateGroup |

|  |  |
| --- | --- |
| **User Story-ID** | 016 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Gruppe löschen können, um Duplikate oder falsch erstellte Gruppen aus dem System entfernen zu können |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | mittel |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 014 |
| **Zugehörige Szenarien** | bMSC\_DeleteGroup |

|  |  |
| --- | --- |
| **User Story-ID** | 017 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine Bestätigung geben müssen, um eine Gruppe zu löschen, um ein versehentliches löschen zu verhindern. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 016 |
| **Zugehörige Szenarien** | bMSC\_DeleteGroupFeedback |

|  |  |
| --- | --- |
| **User Story-ID** | 018 |
| **User Story-Beschreibung** | Als Betreuer möchte einen Studenten auswählen können, um diese Studenten weiter zu bearbeiten. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 003 |
| **Zugehörige Szenarien** | bMSC\_EditStudent |

|  |  |
| --- | --- |
| **User Story-ID** | 019 |
| **User Story-Beschreibung** | Als Betreuer möchte ich einen neuen Studenten erstellen können, um noch nichtexistierende Studenten anlegen zu können. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 018 |
| **Zugehörige Szenarien** | bMSC\_CreateStudent |

|  |  |
| --- | --- |
| **User Story-ID** | 020 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Studenten löschen können, um Duplikate oder falsch erstellte Studenten aus dem System entfernen zu können |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | mittel |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 018 |
| **Zugehörige Szenarien** | bMSC\_DeleteStudent |

|  |  |
| --- | --- |
| **User Story-ID** | 021 |
| **User Story-Beschreibung** | Als Betreuer möchte ich eine Bestätigung geben müssen, um einen Studenten zu löschen, um ein versehentliches löschen zu verhindern. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Niedrig |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 020 |
| **Zugehörige Szenarien** | bMSC\_DeleteStudentFeedback |

|  |  |
| --- | --- |
| **User Story-ID** | 022 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Studenten zu bestimmten Gruppen hinzufügen können, damit der Student der zugehörigen Gruppe zugeordnet werden kann. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | bMSC\_StudentInGroup |

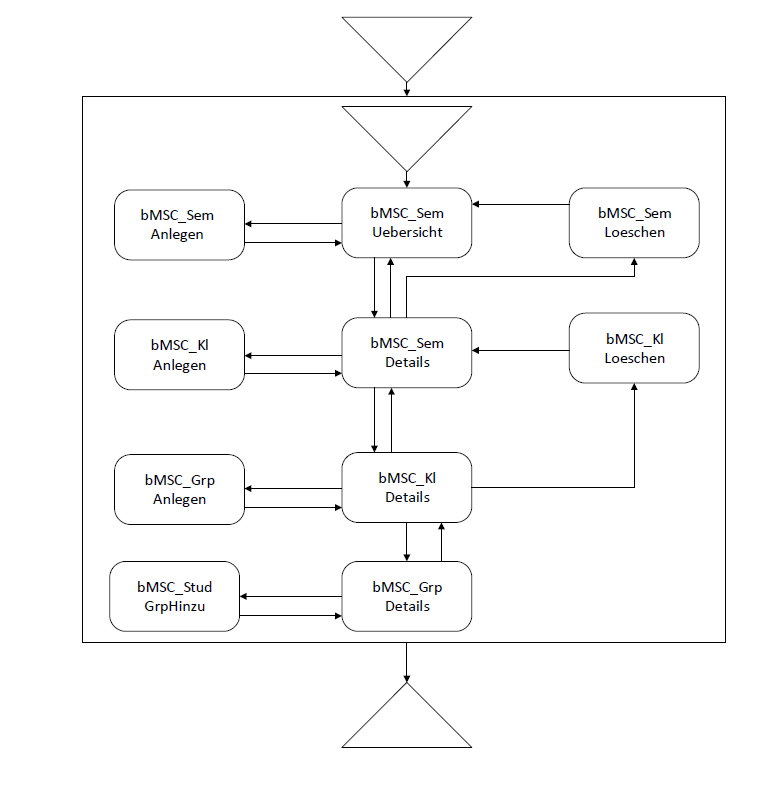
|  |  |
| --- | --- |
| **User Story-ID** | 023 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Gruppen zu bestimmten Klassen hinzufügen können, damit eine Gruppe der zugehörigen Klasse zugeordnet werden kann. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | mittel |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | bMSC\_GroupeInCGroupage |

|  |  |
| --- | --- |
| **User Story-ID** | 024 |
| **User Story-Beschreibung** | Als Betreuer möchte ich Klassen zu bestimmten Semestern hinzufügen können, damit eine Klasse dem zugehörigen Semester zugeordnet werden kann. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | mittel |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** |  |
| **Zugehörige Szenarien** | bMSC\_GroupageInSemester |

|  |  |
| --- | --- |
| **User Story-ID** | 025 |
| **User Story-Beschreibung** | Als Benutzer möchte ich, dass die Anwendung auf mehreren Clients ausgeführt werden kann, ohne dass dabei Dateninkonsistenzen auftreten, damit bei Parallelbetrieb keine Dateninkonsistenzen auftreten. |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Johannes Morzeck, Tobias Sieber |
| **Abhängigkeiten zu anderen User Stories** | 012,002 |
| **Zugehörige Szenarien** | bMSC\_Concurrency |

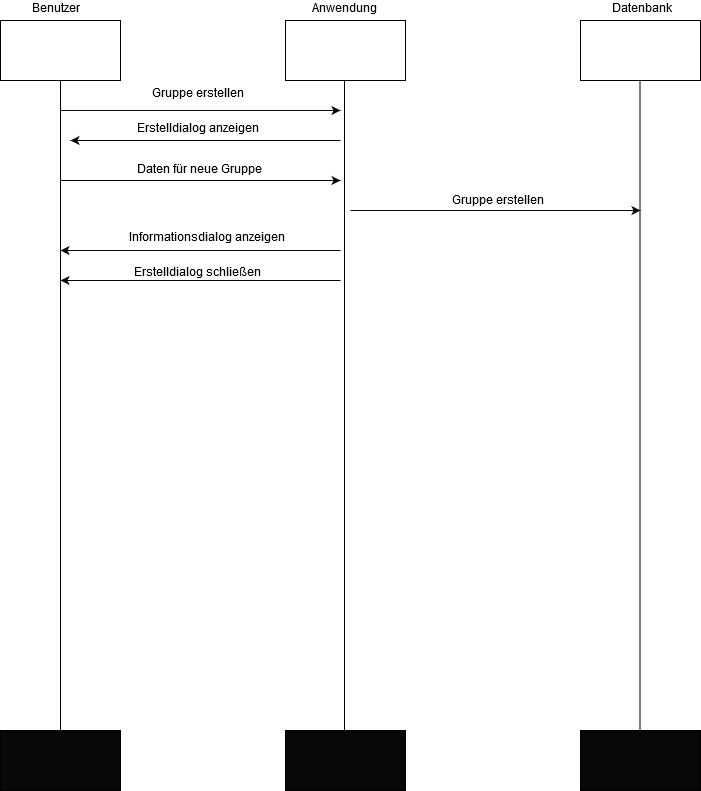
## MSCs

hMSC:



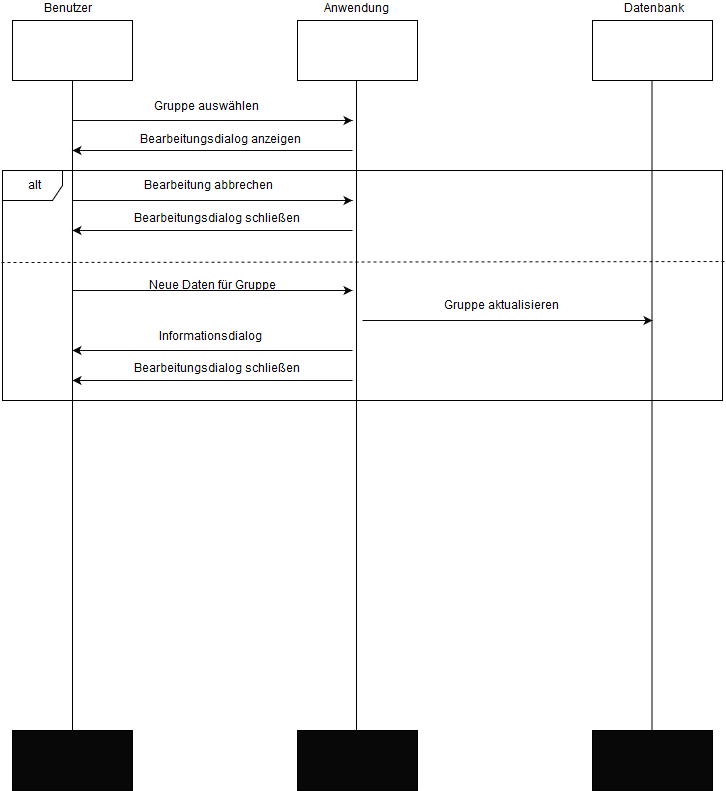
bMSC’s:

Erstellen einer Gruppe  
Dies ist repräsentativ für das Erstellen eines Studenten, einer Klasse und eines Semesters, nur die jeweiligen Attribute ändern sich etwas.

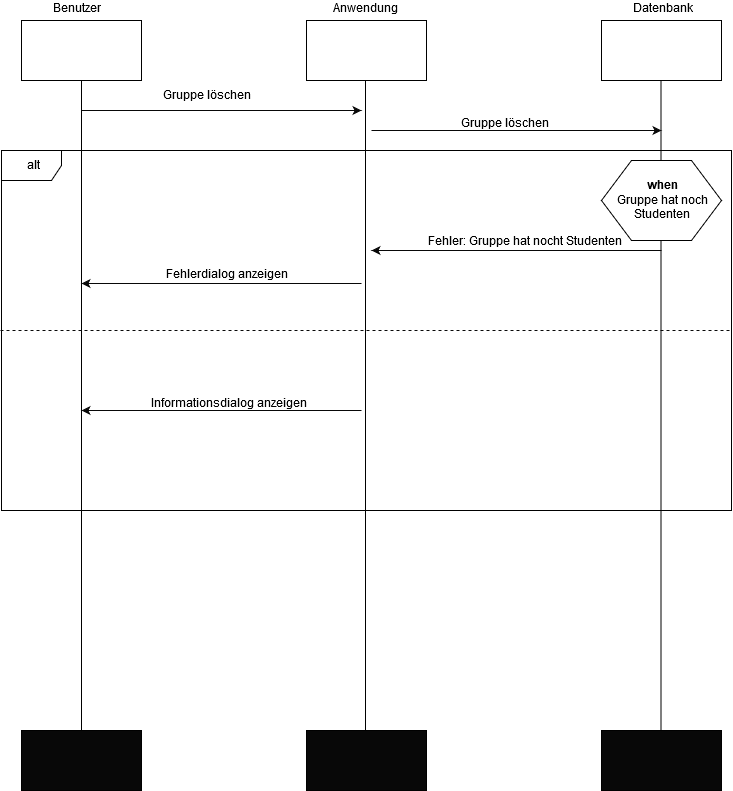


Editieren einer Gruppe

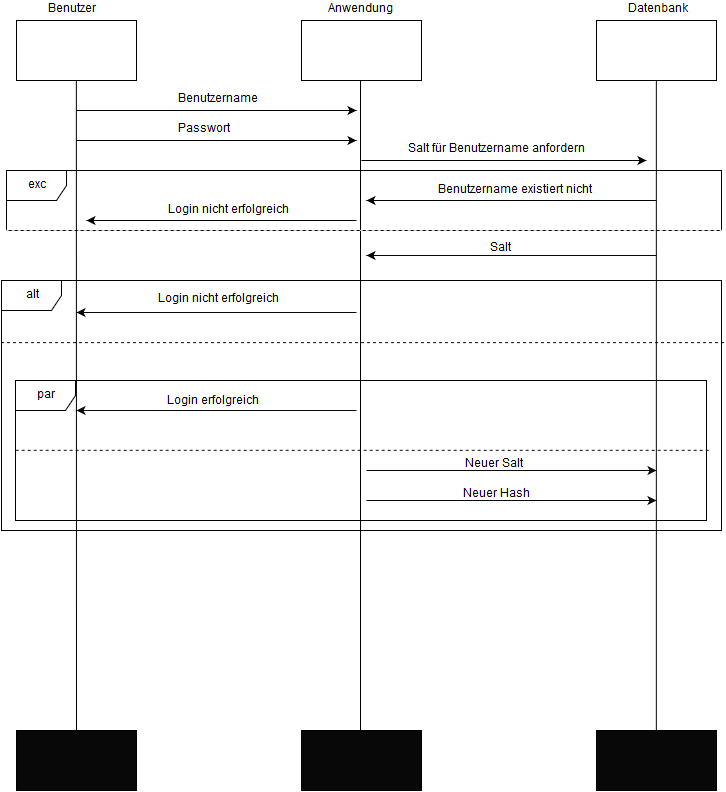
Dies ist repräsentativ für das Editieren eines Studenten, einer Klasse und eines Semesters, nur die jeweiligen Attribute ändern sich etwas.



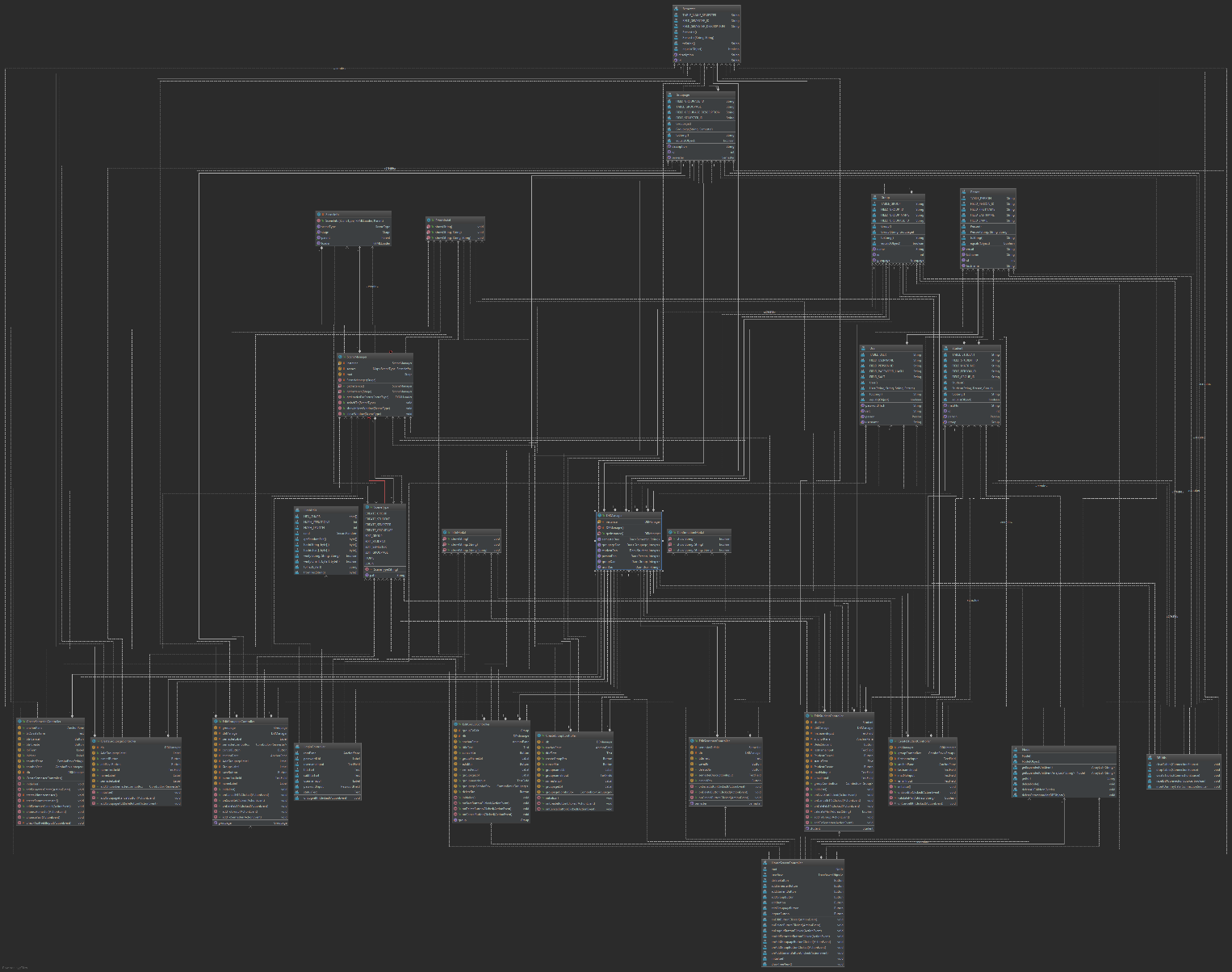
Löschen einer Gruppe  
Dies ist repräsentativ für das Löschen eines Studenten, einer Klasse und eines Semesters.



Login repräsentiert das einloggen in das Programm.



## UML Klassendiagramm



## Funktionalitätsplanung

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Dokumentation** |  |  |  |  |  |
| 1.1 | Erstellen der User Stories | 3h | Johannes, Tobias | bMSCs, hMSC | N.A. | fertig |
| 1.2 | Erstellen des hMSCs | 4h | Andreas | bMSCs | N.A. | fertig |
| 1.3 | Erstellen der bMSCs | 5h | Kaan, Alisan, Merlin, Tobias, Dave | Klassendiagramm | N.A. | fertig |
| 1.4 | Erstellen des Klassendiagramms | 2h | Merlin, Tobias | N.A. | N.A. | fertig |
| 1.5 | Funktionalitätsplanung | 1h | Merlin, Tobias | N.A. | N.A. | fertig |
| 1.6 | Erstellen der Präsentation | 2h | Johannes, Tobias | N.A. | N.A. |  |
| **2.** | **Datenbank** |  |  |  |  |  |
| 2.1 | Zugriff auf die Datenbank | 3h | Merlin, Alisan | Modelle | DBManager | fertig |
| 2.2 | Erstellen einer Datenbank | 1h | Merlin | DBManager, Modelle | DBUtils | fertig |
| **3.** | **Controller** |  |  |  |  |  |
| 3.1 | Erstellen eines Semesters | 5h | Dave, Merlin, Johannes, Tobias | SemesterModel | CreateSemesterController | fertig |
| 3.2 | Editieren eines Semesters | 1h | Johannes | SemesterModel | EditSemesterController | fertig |
| 3.3 | Erstellen einer Klasse | 5h | Kaan, Merlin, Johannes, Tobias | GroupageModel | CreateGroupageController | fertig |
| 3.4 | Editieren einer Klasse | 1h | Tobias | GroupageModel | EditGroupageController | fertig |
| 3.5 | Erstellen einer Gruppe | 4h | Johannes | GroupModel | CreateGroupController | fertig |
| 3.6 | Editieren einer Gruppe | 4h | Johannes | GroupModel | EditGroupController | fertig |
| 3.7 | Erstellen eines Studenten | 5h | Tobias | StudentModel | CreateStudentController | fertig |
| 3.8 | Editieren eines Studenten | 4h | Tobias | StudentModel | EditStudentController | fertig |
| 3.9 | Einloggen in das Programm | 2h | Merlin | HashUtils, UserModel | LoginController | fertig |
| 3.10 | Übersicht und dient zur Öffnung der anderen Controller | 30h | Andreas | Alle Models | HomeViewController | fertig |
| **4.** | **Models** |  |  |  |  |  |
| 4.1 | Schnittstelle Java/Datenbank | 1,5h | Merlin | N.A. | SemesterModel | fertig |
| 4.2 | Repräsentation der Klasse aus der Datenbank | 1,5h | Merlin | N.A. | GroupageModel | fertig |
| 4.3 | Schnittstelle Java/Datenbank | 1,5h | Merlin | N.A. | GroupModel | fertig |
| 4.4 | Schnittstelle Java/Datenbank | 1,5h | Merlin | N.A. | StudentModel | fertig |
| 4.5 | Schnittstelle Java/Datenbank | 1,5h | Merlin | N.A. | PersonModel | fertig |
| 4.6 | Schnittstelle Java/Datenbank | 1,5h | Merlin | N.A. | UserModel | fertig |
| **5.** | **Utils** |  |  |  |  |  |
| 5.1 | Zugriff auf die Datenbank | 0,5h | Merlin | DBManager | DBUtils | fertig |
| 5.2 | Verschlüsselung der Passwörter | 2,5h | Merlin | DBManager | HashUtils | fertig |
| 5.3 | Fensterverwaltung | 1,5h | Merlin | N.A. | SceneManager | fertig |
| 5.4 | Start des Programms | 0,1h | Merlin | SEP | Launcher | fertig |
| 5.5 | Main Klasse | 0,1h | Merlin | N.A. | SEP | fertig |

# Iteration II

## User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | 026 |
| **User Story-Beschreibung** | Als Benutzer möchte ich Notizen zu den verschiedenen Entitäten erstellen können, um mich an wichtige Dinge zu erinnern. |
| **Geschätzter Realisierungsaufwand** | 4 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Kaan Erdogan |
| **Abhängigkeiten zu anderen User Stories** | 003 |
| **Zugehörige Szenarien** | bMSC\_CreateNote |

|  |  |
| --- | --- |
| **User Story-ID** | 027 |
| **User Story-Beschreibung** | Als Benutzer möchte ich die Notizen farbig hinterlegen können, um die Priorisierung optisch zu erkennen. |
| **Geschätzter Realisierungsaufwand** | 1 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Kaan Erdogan |
| **Abhängigkeiten zu anderen User Stories** | 026 |
| **Zugehörige Szenarien** | bMSC\_CreateNote, bMSC\_EditNote |

|  |  |
| --- | --- |
| **User Story-ID** | 028 |
| **User Story-Beschreibung** | Als Benutzer möchte ich die Notizen speichern und löschen können, um sie zeitunabhängig jederzeit zu bearbeiten/abzurufen. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Hoch |
| **Autor(en)** | Kaan Erdogan |
| **Abhängigkeiten zu anderen User Stories** | 026 |
| **Zugehörige Szenarien** | bMSC\_DeleteNote, bMSC\_CreateNote, bMSC\_EditNote |

|  |  |
| --- | --- |
| **User Story-ID** | 029 |
| **User Story-Beschreibung** | Als Benutzer möchte ich alle meine Notizen aufgelistet bekommen, um per Schnellzugriff auf eine Notiz zu gelangen. |
| **Geschätzter Realisierungsaufwand** | 2 Std. |
| **Priorität** | Mittel |
| **Autor(en)** | Kaan Erdogan |
| **Abhängigkeiten zu anderen User Stories** | 026, 028 |
| **Zugehörige Szenarien** | bMSC\_NotesTab |

## MSCs

bMSCs:

bMSC zum anzeigen der Notes Liste



bMSC zum erstellen einer Notiz



bMSC zum editieren einer Notiz



bMSC zum anzeigen einer Notiz



bMSC zum löschen einer Notiz



## UML Klassendiagramm

Erläuterung des Klassendiagramms:

## Funktionalitätsplanung

Template:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcodereferenz** | **Status** |
| **1.** | **Dokumentation** |  |  |  |  |  |
| 1.1 | Erstellen der User Stories | 3h | Kaan, | bMSCs, hMSC | N.A. |  |
| 1.2 | Erstellen des hMSCs | 4h |  | bMSCs | N.A. |  |
| 1.3 | Erstellen der bMSCs | 5h | Kaan, | Klassendiagramm | N.A. |  |
| 1.4 | Erstellen des Klassendiagramms | 2h |  | N.A. | N.A. |  |
| 1.5 | Funktionalitätsplanung | 1h | Kaan, | N.A. | N.A. |  |
| 1.6 | Erstellen der Präsentation | 2h |  | N.A. | N.A. |  |
| **2.** | **Datenbank** |  |  |  |  |  |
| 2.1 | Zugriff auf die Datenbank | 3h | Kaan, | Modelle | DBManager | fertig |
| 2.2 | Erstellen einer Datenbank | 1h | Kaan, | DBManager, Modelle | DBUtils | fertig |
| **3.** | **Controller** |  |  |  |  |  |
| 3.1 | Erstellen einer Notiz | 4h | Kaan | Notepad, StudentNotepad, GroupNotepad, GroupageNotepad (Models) | CreateNotepadController | fertig |
| 3.2 | Editieren einer Notiz | 4h | Kaan | Notepad, StudentNotepad, GroupNotepad, GroupageNotepad  (Models) | EditNotepadController | fertig |
| 3.3 | Anzeigen der Notizliste | 12h | Kaan | Notepad, StudentNotepad, GroupNotepad, GroupageNotepad, HomeScreen | NotesTabController | teilweise fertig |
| 3.4 | Anzeigen der Notiz | 2h | Kaan | NotesTab | NoteWindowController | fertig |
| **4.** | **Models** |  |  |  |  |  |
| 4.1 | Schnittstelle Java/Datenbank | 1,5h | Kaan | N.A. | Notepad | fertig |
| 4.2 | Schnittstelle Java/Datenbank | 1,5h | Merlin, Kaan | N.A. | GroupageNotepad | fertig |
| 4.3 | Schnittstelle Java/Datenbank | 1,5h | Merlin, Kaan | N.A. | GroupNotepad | fertig |
| 4.4 | Schnittstelle Java/Datenbank | 1,5h | Merlin, Kaan | N.A. | StudentNotepad | fertig |
| **5.** | **Utils** |  |  |  |  |  |

## Unittests

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Getestete Funktionalität** | **Quellcode**  **Referenz** | **Status** |
| 1 | Erstellen einer Notiz | CreateNotepadController | fertig |
| 2 | Editieren einer Notiz | EditNotepadController | Fertig |
| 3 | Anzeigen einer Notiz | NoteWindowController | fertig |

## 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.