**SEP Projektmappe**

* ***Projektmappe des Projektes***

**Super Airline Management Programm**

**Dokumentation des Projektes**

*Gruppe O:*

Pascal Berg (3052905)  
Fabian Haarmann (3069576)  
Nkechinyere Pearl Ishieka (3013274)  
El Mouhcine Rachid (2236324)  
Brenda Linda Weber (3027085)

**Inhalt**

Projektbeschreibung

Einleitung:

Ziel ist die Implementierung eines Flugverwaltungsprogramms. Hierzu gehört die Erstellung und Verwaltung von Fluggesellschaften inkl. des Anlegens von neuen Fluglinien und der Generierung von Flügen. Kunden sollen entsprechend Flüge buchen können. Zuletzt sollen Kunden die Möglichkeit haben, ihre Buchungen einzusehen und durch Zusatzfunktionen ihre Reisen weiter planen zu können.

Iteration I:

In der ersten Iteration werden die grundlegenden Strukturen implementiert, auf deren Grundlage das Programm basiert. Hierzu gehört das Einlesen der Datensätze für die Flugzeuge und die Flughäfen sowie die persistente Speicherung aller Daten.

Zunächst soll ein Administratorkonto angelegt werden, über das es möglich ist, Fluggesellschafts-Manager anzulegen. Jedes Konto soll hierbei über einen Benutzernamen und ein Passwort verfügen. Passwörter sollen grundsätzlich gesalted und gehasht gespeichert werden.

Im Folgenden werden die zu implementierenden Funktionen des Fluggesellschafts-Managers beschrieben. Ein Fluggesellschafts-Manager soll genau eine Fluggesellschaft anlegen können. Eine Fluggesellschaft verfügt hierbei über einen Namen, ein Land (Hauptstandort) und ein Budget. Das Budget soll genutzt werden, um Flugzeuge (siehe Datensatz [1]) kaufen zu können. Dabei kann davon ausgegangen werden, dass jedes Flugzeug in beliebiger Zahl zur Verfügung steht. Der Fluggesellschafts-Manager soll Fluglinien anlegen können, die im späteren Verlauf als Vorlage für die Instanziierung von Flügen genutzt werden. Eine Fluglinie verfügt hierbei über einen Start- und Zielflughafen, ein Startdatum (Jungfernflug), ein Intervall (täglich, alle drei Tage, wöchentlich), ein Flugzeug (welches natürlich im Besitz der Fluggesellschaft sein muss), eine Sitzplatzverteilung (Anzahl Economy/Business, wobei ein Business-Sitz zwei Sitzen gemäß Datensatz [1] entspricht und nur maximal 25% aller Sitze als Business-Sitze deklariert werden können) und konkrete Preise für die Sitze in den verschiedenen Klassen. Bei der Zuweisung eines Flugzeuges soll berücksichtigt werden, ob dieses für die gewählte Strecke geeignet ist (z.B. anhand der Reichweite).

Ein Fluggesellschafts-Manager soll sich zudem eine Gesamtübersicht über alle Fluglinien seiner Fluggesellschaft anzeigen lassen können, sowie einzelne Fluglinien auswählen und bearbeiten (Änderung des Flugzeugs, Sitzplatzverteilung, Preise, Intervall) können.

Iteration I

* User-Stories

|  |  |
| --- | --- |
| **User Story-ID** | 1.1 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich ein Administratorkonto haben, damit die Kunden nur Zugriff auf die vorgesehenen Funktionen haben. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2, 1.5 |
| **User Story-ID** | 1.2 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich Fluggesellschaftsmanager anlegen können, um neue Kunden in das Programm einbinden zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.1, 1.5 |
| **User Story-ID** | 1.3 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich Daten über Flugzeuge einlesen können, um sie zum System hinzufügen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Gering |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.4, 1.5 |
| **User Story-ID** | 1.4 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich Daten über Flughäfen einlesen können, um sie zum System hinzufügen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Gering |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.3, 1.5 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.5 |
| **User Story-Beschreibung** | Als Systemadministrator möchte ich, dass alle Daten gespeichert werden, um das Programm über das Internet zur Verfügung zu stellen. |
| **Geschätzter Realisierungsaufwand** | 3 Tage |
| **Priorität** | Sehr Hoch |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.1, 1.2, 1.3, 1.4 |
| **User Story-ID** | 1.6 |
| **User Story-Beschreibung** | Als Sicherheitsbeauftragter möchte ich, dass die Passwörter gesaltet werden, um schwache Passwörter zu schützen. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Medium |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.5, 1.7 |
| **User Story-ID** | 1.7 |
| **User Story-Beschreibung** | Als Sicherheitsbeauftragter möchte ich, dass die Passwörter gehasht gespeichert werden, um Sicherheitsstandards einzuhalten. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Medium |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.5, 1.6 |
| **User Story-ID** | 1.8 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich eine Fluggesellschaft anlegen können, um das Programm benutzen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Hoch |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2, 1.5 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.9 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich mein Budget eintragen können, um damit neue Flugzeuge kaufen zu können. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Gering |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2 |
| **User Story-ID** | 1.10 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich neue Fluglinien anlegen können, um neue Routen ins System einzugliedern. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Hoch |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2, 1.5 |
| **User Story-ID** | 1.11 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich, dass das Programm überprüft, ob das Flugzeug für die Strecke geeignet ist, um Unfälle zu vermeiden. |
| **Geschätzter Realisierungsaufwand** | 1 Tag |
| **Priorität** | Gering |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2 |
| **User Story-ID** | 1.12 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich eine Übersicht aller meiner Fluglinien haben, um alle Fluglinien auf Fehler untersuchen kann. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Gering |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2, 1.5, 1.10 |

|  |  |
| --- | --- |
| **User Story-ID** | 1.13 |
| **User Story-Beschreibung** | Als Fluggesellschaftsmanager möchte ich Fluglinien ändern können, um Änderungen in das System einzugliedern. |
| **Geschätzter Realisierungsaufwand** | 2 Tage |
| **Priorität** | Medium |
| **Autor(en)** | Fabian Haarmann (3069576) |
| **Abhängigkeiten zu anderen User Stories** | 1.2, 1.10 |

Papierprototypen

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

Erläuterung des Papierprototypen:

* Aktivitätsdiagramm

Ein Aktivitätsdiagramm dokumentiert/spezifiziert eine Aktivität, indem es die Knoten und Flüsse modelliert. Aktivitätsdiagramme eignen sich zur Visualisierung von Abläufen zur Erreichung eines bestimmten Verhaltens, z.B. für: Operationen, Do/activity eines Zustandsdiagramms, Szenarien eines Use Cases, Geschäftsprozesse, Algorithmen, Ein Aktivitätsdiagramm modelliert eine Menge konkreter Abläufe (Ablaufinstanzen).



Erläuterung des Aktivitätsdiagramms:

* Gestartet wird das Programm im Login Fenster. Hier werden die Login Daten überprüft. Außerdem wird hier der Typ des Accounts abgefragt um so entweder mit dem laden der Admin Seite oder der Manager Seite fortzufahren.  
  Angenommen der Login erfolgt als Admin lädt die Admin Seite welche die Manager erstellen funktionalität bereitstellt. In der man einen Account vom Typen Manager erstellen kann. Die so erstellten Daten werden gehasht und gesalted in der Datenbank aufbewahrt.  
  Sollte sich nun ein Manager anmelden öffnet sich ein Manager Fenster in dem die folgenden Aktivitäten zu finden sind: Fluggesellschaft anlegen, Fluglinie anlegen, Fluglinie anzeigen und Fluglinie ändern. "Fluggesellschaft" anlegen erstellt eine Fluggesellschaft sollte keine vorhanden sein. "Fluglinie anlegen" erstellt eine Fluglinie.  
  "Fluglinie anzeigen" zeigt alle Fluglinien an und bieten die möglichkeit diese zu verändern. "fluglinie ändern" bietet die selbe funktionalität.
* Strukturdiagramm (Klassendiagramm oder Objektdiagramm)

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.

Erläuterung des Strukturdiagramms:

* 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. Folgend erhalten sie ein Template.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Funktionalität** | **Geschätzter**  **Aufwand** | **Verantwortlicher** | **Abhängige**  **Funktionalitäten** | **Quellcode-**  **referenz** | **Status** |
| **1.** | **Anzeige** |  |  |  |  |  |
| 1.1 | GUI / css support | 5 Tage | Pascal Berg (3052905) |  |  | fertig |
| 1.2 | SQL Planung? |  |  |  |  |  |
| 1.2.1 | …. |  |  |  |  |  |
| **2.** | **Allgemeine Funktionen** |  |  |  |  |  |
| 2.1 | Speicherung der Daten auf Server | 3 Tage | Brenda Linda Weber (3027085) |  |  | fertig |
| 2.2 | Login / salting / hashing | 1 Tag | El Mouhcine Rachid (2236324) |  |  | fertig |
| 2.3 |  |  |  |  |  |  |
| 2.4 | Manager anlegen | 2 Tage | Brenda Linda Weber (3027085) |  |  |  |
| 2.5 | Manager ändern | 1 Tag |  | 2.4 |  | fertig |
| 2.6 | Manager löschen | 1 Tag |  | 2.4 |  | fertig |
| 2.7 | Fluglinie erstellen | 2 Tage | Nkechinyere Pearl Ishieka (3013274) | 2.10, 2.11 |  | fertig |
| 2.8 | Fluglinien anzeigen | 1 Tag | Nkechinyere Pearl Ishieka (3013274) | 2.7 |  | fertig |
| 2.9 | Fluglinien ändern | 1 Tag | Nkechinyere Pearl Ishieka (3013274) | 2.7, 2.8 |  | fertig |
|  | Sitzplatzverteilung |  | Fabian Haarmann (3069576) |  |  |  |
|  | Reichweite-Check |  | Fabian Haarmann (3069576) |  |  |  |
| 2.10 | Flugzeuge einlesen | 2 Tage | Brenda Linda Weber (3027085) |  |  | fertig |
| 2.11 | Flughäfen einlesen | 2 Tage | Brenda Linda Weber (3027085) |  |  |  |
| 2.12 | Flugzeuge kaufen | 2 Tage | Fabian Haarmann (3069576) | 2.9 |  |  |



Iteration II

* 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(en)** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |

Papierprototypen

Erläuterung des Papierprototypen:

LoginStart stellt das Login Fenster da.  
AdminMenu dient links neben dem eigentlich fenster als Orientierungs Menu als Admin.  
ManagerMenu dient links neben dem eigentlich fenster als Orientierungs Menu als Manager.  
AdminManagerErstellen, AdminManagerÄndern, AdminManagerLöschen stellen die Aktivitäten als Admin dar.  
ManagerFluggesellschaftAnlegen, ManagerFluglinieÄndern, ManagerFluglinieAnlegen, ManagerFluglinieAnsehen, ManagerFlugzeugeAnsehen stellen die Aktivitäten als Manager dar.

Aktivitätsdiagramm

Erläuterung des Aktivitätsdiagramms:

* Strukturdiagramm (Klassendiagramm oder Objektdiagramm)

Erläuterung des Strukturdiagramms:

* Funktionalitätsplanung

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

Unittests

Unittest oder 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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **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)** | |
| 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 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)** | |
| 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. | |  |

Iteration III

* 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(en)** | <Hier bitte nur einen Zuständigen eintragen z.B. Max Mustermann> |
| **Abhängigkeiten zu anderen User Stories** | <Auflistung verwandter User Stories> |
| **Zugehörige Szenarien** | <Zuordnung zu Szenarien>   * Hauptszenario: * Alternativszenarien: * Ausnahmeszenarien: |

Papierprototypen

Erläuterung des Papierprototypen:

Aktivitätsdiagramm

Erläuterung des Aktivitätsdiagramms:

* Strukturdiagramm (Klassendiagramm oder Objektdiagramm)

Erläuterung des Strukturdiagramms:

* Funktionalitätsplanung

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

* Unittests

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

* Systemtests

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** |  | |  |
| **Tester** |  | |  |
| **SW-Version** |  | |  |
| **Vorbedin-gung(en)** |  | |  |
| **Schritt** | **Aktion (User)** | **Erwartete Reaktion (System)** | |
| 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.