

Dokumentation der Projektaufgabe

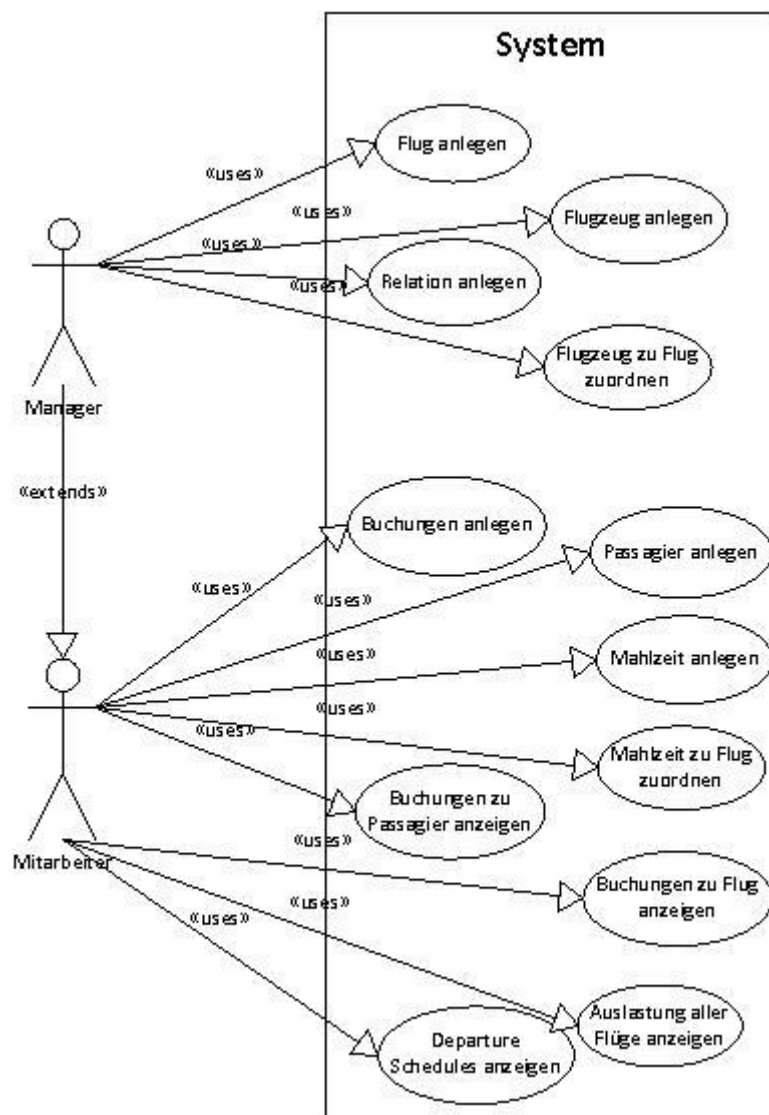
Autoren: Markus Giesbrecht, Halil Özdoğan

Kurzbeschreibung der Anwendung

Im Rahmen der Veranstaltung „JavaEE Seminar“ an der Hochschule Weserbergland soll ein Fluginformationssystem implementiert werden. Dazu wird die Technologie der Java Enterprise Edition verwendet.

User-Story-Diagramme

Ein Gesamtüberblick aller Use-Cases bietet folgende Abbildung :



Name	Relation anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Manager
Stakeholder und Interessen	Manager
Vorbedingungen	Es sind bereits Flughäfen in der Datenbank vorhanden.
Nachbedingungen	Anlage (bzw. Eintrag in der Datenbank) einer Relation mit Start-, Zielflughafen, der Distanz und Flugzeit
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Manager loggt sich ein 2. Der Manager gelangt auf die Maske, in der eine Relation angelegt werden kann. 3. Der Manager wählt aus einer gegebenen Flughafenliste den Start- und Zielflughafen aus. 4. Der Manager gibt eine Flugzeit und eine Distanz für die Relation an.
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	eher seltener
Sonstiges	-

Name	Flug anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Manager
Stakeholder und Interessen	Manager
Vorbedingungen	Es sind bereits Relationen angelegt.
Nachbedingungen	Anlage (bzw. Eintrag in der Datenbank) eines Flugs mit Abhängigkeit zu einer Relation und mit Abflugs-, Ankunftszeitpunkt und Preis.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Manager loggt sich ein 2. Der Manager gelangt auf die Maske, in der ein Flug angelegt werden kann. 3. Der Manager wählt aus einer gegebenen Relationenliste die zugehörige Relation aus. 4. Der Manager gibt den Abflugszeitpunkt und den Preis an.
Erweiterungen	-
Besondere Anforderungen	Der Ankunftszeitpunkt wird mit Hilfe der Flugzeit der Relation und dem Abflugszeitpunkt errechnet.
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	durchschnittlich
Sonstiges	-

Name	Flugzeug anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Manager
Stakeholder und Interessen	Manager
Vorbedingungen	-
Nachbedingungen	Anlage (bzw. Eintrag in der Datenbank) eines Flugzeugs mit Angabe des Herstellers, des Typs und der Anzahl Sitzplätze.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Manager loggt sich ein 2. Der Manager gelangt auf die Maske, in der ein Flugzeug angelegt werden kann. 3. Der Manager gibt den Hersteller, den Typ und die Anzahl der Sitzplätze an.
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	-
Sonstiges	Die Anzahl Sitzplätze wird für die Berechnung der Kapazität eines Flugs benötigt.

Name	Flugzeug zu Flug zuordnen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Manager
Stakeholder und Interessen	Manager
Vorbedingungen	Es müssen mindestens ein Flug und ein Flugzeug angelegt sein.
Nachbedingungen	Das Flugzeug ist erfolgreich dem Flug zugeordnet.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Manager loggt sich ein 2. Der Manager gelangt auf die Maske, in der ein Flugzeug einem Flug zugeordnet werden kann. 3. Der Manager wählt aus einer Flugliste den Flug aus 4. Der Manager wählt aus einer Flugzeugliste das Flugzeug aus
Erweiterungen	-
Besondere Anforderungen	Ein Flugzeug kann nur genau einem Flug zugeordnet werden.
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Eher selten
Sonstiges	-

Name	Mahlzeit anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	-
Nachbedingungen	Die Mahlzeit ist erfolgreich mit einem Namen, einer Mahlzeitart und der Angabe, ob die Mahlzeit vegetarisch ist, in der Datenbank angelegt.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der eine Mahlzeit angelegt werden kann. 3. Der Akteur gibt den Namen, die Art und, ob die Mahlzeit vegetarisch ist, an.
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Durchschnittlich
Sonstiges	-

Name	Mahlzeit zu Flug zuordnen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es müssen mindestens ein Flug und eine Mahlzeit angelegt sein.
Nachbedingungen	Die Mahlzeit ist erfolgreich dem Flug zugeordnet.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der eine Mahlzeit einem Flug zugeordnet werden kann. 3. Der Akteur wählt aus einer Flugliste den Flug aus 4. Der Akteur wählt aus einer Mahlzeitliste die Mahlzeit aus
Erweiterungen	-
Besondere Anforderungen	Ein Flug hat genau eine zugeordnete Mahlzeit.
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	durchschnittlich
Sonstiges	-

Name	Passagier anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	-
Nachbedingungen	Der Passagier ist erfolgreich mit seinem Vornamen, Nachnamen, seiner Anschrift, seinem Geburtsdatum und seiner Nationalität, in der Datenbank angelegt.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der eine Mahlzeit angelegt werden kann. 3. Der Akteur gibt den Vornamen, Nachnamen, die Anschrift, das Geburtsdatum und die Nationalität des Passagiers an.
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Durchschnittlich
Sonstiges	-

Name	Buchung anlegen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es müssen mindestens ein Passagier und ein Flug angelegt sein.
Nachbedingungen	Die Buchung ist erfolgreich mit ihrem Buchungsdatum, mit Verweisen auf den zugehörigen Flug und Passagier, in der Datenbank angelegt.
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der eine Buchung angelegt werden kann. 3. Der Akteur wählt aus einer Flugliste einen Flug aus 4. Der Akteur wählt aus einer Passagierliste einen Flug aus
Erweiterungen	-
Besondere Anforderungen	Das Buchungsdatum wird automatisch auf den Zeitpunkt der Ausführung gesetzt.
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	häufig
Sonstiges	-

Name	Buchungen zu Passagier anzeigen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es muss mindestens ein Passagier eine Buchung durchgeführt haben.
Nachbedingungen	Anzeige aller Buchungen zu einem Passagier
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der alle Passagiere aufgelistet sind. 3. Der Akteur klickt auf einen Passagier
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Durchschnittlich
Sonstiges	-

Name	Buchungen zu Flug anzeigen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es muss mindestens zu einem Flug eine Buchung durchgeführt sein.
Nachbedingungen	Anzeige aller Buchungen zu einem Flug
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein 2. Der Akteur gelangt auf die Maske, in der alle Flüge aufgelistet sind. 3. Der Akteur klickt auf einen Flug
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Durchschnittlich
Sonstiges	-

Name	Auslastung aller Flüge anzeigen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es gibt Flüge, zu denen Buchungen durchgeführt wurden und ein Flugzeug zugeordnet wurde.
Nachbedingungen	Anzeige der freien Plätze des Flugs
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein. 2. Der Akteur gelangt auf die Maske, auf der er einen Flug auswählen kann 3. Der Akteur wählt einen Flug aus.
Erweiterungen	-
Besondere Anforderungen	-
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Häufig
Sonstiges	-

Name	Departure Schedules anzeigen
Scope	Fluginformationssystem
Ebene	Nutzerziel
Hauptakteur	Mitarbeiter, Manager
Stakeholder und Interessen	Mitarbeiter, Manager
Vorbedingungen	Es sind Flüge angelegt.
Nachbedingungen	Anzeige der Status aller Flüge
Haupterfolgsszenario	<ol style="list-style-type: none"> 1. Der Akteur loggt sich ein. 2. Der Akteur navigiert auf die Maske Departure Schedules
Erweiterungen	<p>2a. Der Akteur wählt einen Flug aus und der Flug ist im Status „scheduled“:</p> <ol style="list-style-type: none"> 1. Der Akteur kann einen Passagier auswählen und bucht für diesen den ausgewählten Flug.
Besondere Anforderungen	Die Status „canceled“ und „delayed“ werden zufällig gesetzt.
Technologie- und Datenvarianz	-
Auftretenshäufigkeit	Häufig
Sonstiges	-

Darstellung der technischen Architektur

Die Anwendung ist in drei Schichten aufgeteilt. Das Frontend, das Backend und die Datenbank¹.

Im Frontend wird hauptsächlich das Komponentenframework Java Server Faces verwendet. Im Backend wird zum Großen Teil mit Enterprise Java Beans und dem Java Persistence API gearbeitet. Die Datenbank ist eine MySQL Datenbank.

Das Frontend besteht zum einen aus mehreren Komponenten. Darunter fällt der Ordner Web Content, die Packages Frontend Beans und Controller und zuletzt einem Servlet.

Der Ordner Web Content beinhaltet sämtliche xhtml Seiten für den Benutzer. Weiter ist ein Hintergrundbild im Ordner img enthalten, sowie die bootsfaces.css im Ordner css. Neben den xhtml wird noch eine jsp Seite für den Logout des Users verwendet.

Die Frontend Packages enthalten sämtliche Beans, welche frei von Logik sind und nur Methoden und Attribute aus dem Backend ins Frontend weiterleiten. Dabei werden verschiedene Rückmeldungen deklariert und zurückgegeben. Somit wird erkennbar, ob es sich um Fehler oder um Erfolgsmeldungen handelt. Ebenso wird die UserInfoBean im Frontend Package abgebildet. Diese gibt zustände über die Session wieder und erhält diese Informationen aus dem Frontend Controller.

Der Frontend Controller enthält die Klasse SessionUtils, welche sich um die Nutzer Session kümmert. Darunter fällt das Login aber auch die Unterscheidung von Managern oder normalen Mitarbeitern.

Das Servlet im Package Frontend Servlet wird über das logout.jsp angesprochen und sorgt dafür, dass die aktuelle Session aufgelöst und der User ausgeloggt wird.

Im Backend gibt insgesamt vier Packages.

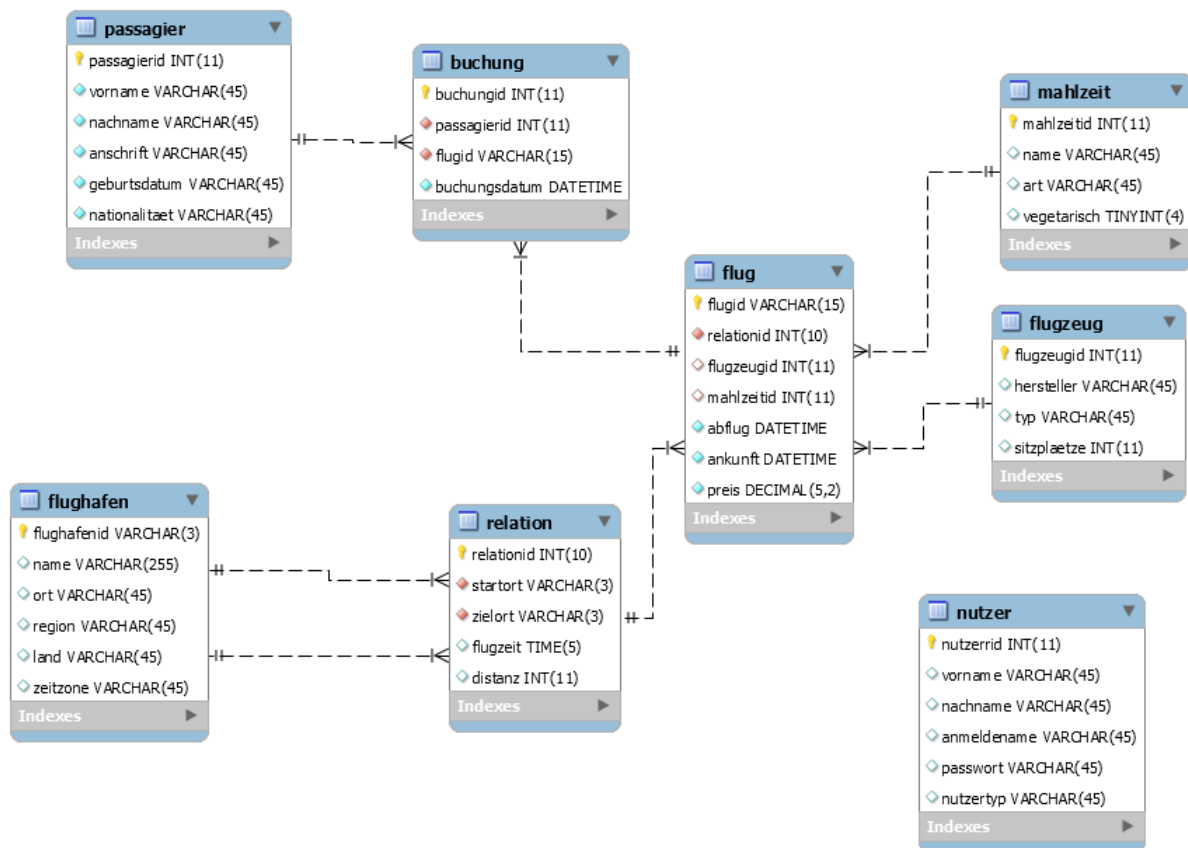
Das Package backend.enterpriseLogic beinhaltet die Klasse DatabaseHandler. In dieser werden Attribute und der Konstruktor für alle anderen Handler-Klassen, mit Ausnahme der SuccessHandler und ErrorHandler, gesetzt. Daraus folgend ist die Klasse DatabaseHandler die Superklasse und alle anderen Handler-Klassen die Subklassen. Die Handler-Klassen beinhalten verschiedenen Methoden, die die Entitäten im Package backend.entities verwalten bzw. handeln (siehe API_Methoden.pdf)

Im Package backend.entities befinden sich alle Entitätsklassen, die die Tabellenstrukturen der Datenbank abbildet. Neben den vorgegebenen NamedQueries wurden an mehreren Stellen eigene NamedQueries angelegt.

Das Package backend.models enthält Klassen, die rein zur Anzeige für JSF bzw. das Frontend genutzt werden. Das Package backend.test enthält Testfälle zu den Klassen im Package backend.enterpriseLogic.

¹ Die Datenbank wird im Kapitel Darstellung des Datenmodells beschrieben.

Darstellung des Datenmodells



In der Tabelle flughafen sind alle Flughäfen enthalten, zu denen eine Relation angelegt werden kann. Eine Relation besteht aus einem startort, zielort (beide Fremdschlüssel auf flughafenid in der Tabelle flughafen), aus der Flugzeit und der Distanz. In der Tabelle Flug werden Flüge zu Relationen hinterlegt, dabei kann es auf einer Relation mehrere Flüge geben. Zudem kann einem Flug ein Flugzeug, sowie eine Mahlzeit zugeordnet werden (jeweils Fremdschlüssel auf mahlzeitid in mahlzeit und flugzeugid in flugzeug). Wichtig ist in der Tabelle flugzeug die Spalte sitzplaetze. Diese ist die Basis, um die Auslastung eines Flugs anzuzeigen. Wichtig ist außerdem zu wissen, dass ein Flugzeug nur genau einem Flug zugeordnet werden kann. Im Gegensatz dazu kann eine Mahlzeit auf unterschiedlichen bzw. mehreren Flügen angeboten werden. Weitere Spalten sind abflug (Format: yyyy-dd-mm hh:mm:ss) und ankunft (Format: yyyy-dd-mm hh:mm:ss), sowie der preis des Flugs. Bei der Anlage eines Flugs wird von der Anwendung in den Spalten flugzeugid und mahlzeitid auf den ersten Eintrag in den jeweiligen Tabellen verwiesen (diese sind gesondert gekennzeichnet, durch ihre Werte in den jeweiligen Spalten). Um eine Buchung durchführen zu können, muss zunächst ein Passagier in der Tabelle passagier angelegt sein. In der Tabelle buchungen ist hinterlegt, welcher Passagier welchen Flug gebucht hat.

Einrichtungshinweise

1. Für die Datenbankverbindung muss in Eclipse unter Window/Preferences/Data Management/Connectivity/Driver Definitions eine neue Driver Definition vom Typ MySQL in der Version 5.1 angelegt werden. Die MySQL-Connector.jar ist im Workspace unter dem Ordner /libs zu finden. Die Zugangsdaten zum Server lauten:

Connection URL: jdbc:mysql://giesbrechthosting.de:3306/fis-db

Database name: fis-db

Driver Class: com.mysql.jdbc.Driver

Password: start123

User ID: root

Falls die hier genannte Datenbank nicht verwendet werden soll, kann unter dem Ordner docs/ das Skript „database.sql“ auf die gewünschte Datenbank ausgeführt werden.

Als nächstes in Eclipse die View JPA öffnen und mit Hilfe der angelegten Driver Definition ein neues Connection Profile anlegen.

2. Des Weiteren wurden einige Bibliotheken mit dem Tool Maven automatisiert hinzugefügt. Falls die Bibliotheken noch nicht gedownloadet sind, muss auf das Projekt per Rechtsklick auf den Punkt Maven/Update Project... navigiert werden. Im anschließenden Fenster auf OK klicken.
3. Wichtig ist außerdem, dass die Glassfish Version, die in der Installation.JEE.WI.HSW.HM.WS1718. pdf aus dem 5. Semester verwendet wird, da diese Version einige wichtige Bibliotheken beinhaltet.