Praktikum Software-Engineering 1

Sommersemester 2025 – Blatt 3

Prof. Dr. Stefan Sarstedt <stefan.sarstedt@haw-hamburg.de>
Enrico Christophers <enrico.christophers@haw-hamburg.de>

Aufgabe 1: Use Case & Wireframes

Formuliert einen Use Case für die User-Story „*Ich als BenutzerIn möchte ein Fährticket buchen*.“ (du hast hier freie Hand, wie dies genau aussehen soll) und erstellt zugehörige Wireframes für eine mobile App. Verzichtet beim Use Case auf den „Erweiterungen“-Abschnitt, sondern fokussiert auf den Erfolgsfall, sowie die Fehlerfälle. Geht davon aus, der die BenutzerIn bereits ein Profil mit Bezahldaten besitzt und eingeloggt ist.

Aufgabe 2: Fachliches Datenmodell in UML-Notation

Erstelle ein fachliches Datenmodell in UML-Klassendiagramm-Notation (siehe Vorlesungsfolien) für den folgenden Sachverhalt:

- Jeder Benutzer kann ein oder mehrere Fährfahrten buchen (Start-Hafen, Ziel-Hafen). Jeder Hafen hat eine GPS-Position und eine Bezeichnung.
- Benutzer werden durch die E-Mail-Adresse eindeutig identifiziert.
- Für die Fahrten gibt es jeweils eine Rechnung in €.
- Es gibt drei Kabinenkategorien: NORMAL_INNEN, NORMAL_BALKON und SUITE. Dies sollst du durch einen Aufzählungstyp darstellen und nicht durch eine Entität.

Aufgabe 3: Implementierung des Fachlichen Datenmodells

Implementiert in dieser Aufgabe Euer Fachliches Datenmodell aus Aufgabe 2.

- Implementiert der Einfachheit halber ausschließlich einfach-gerichtete (=unidirektionale) und keine doppelgerichteten (=bidirektionalen) Beziehungen! D.h. in einer Assoziation "sieht" nur genau eine der beiden beteiligten Klassen die jeweils andere. Dies macht die Implementierung weniger fehleranfällig.
- Orientiert Euch bei der Implementierung der einfachen und komplexen fachlichen Datentypen an denen aus dem Beispielprojekt („Gender“, „PhoneNumber", ...)
- Implementiert Repositories für Eure Entitäten (ähnlich CustomerRepository aus dem Beispiel – bei Bedarf mit eigenen Query-Methoden: siehe Dokumentation unter https://docs.spring.io/spring-data/jpa/reference/repositories/query-methods-details.html).
- **Pusht** Euer neues Projekt auf http://git.haw-hamburg.de und gebt mir Maintainer-Berechtigung darauf. Achtet darauf, dass die **Pipeline grün** ist.

Aufgabe 4: Spezifikation von State Charts

Spezifiziert State Charts für das BMS, für einen Rauchwarnmelder, sowie für eine Brandschutztür. Diese State Charts interagieren miteinander durch entsprechendes Auslösen von Events (Rauchwarnmelder triggert das BMS, BMS triggert eine Brandschutztür, etc.)

Rauchwarnmelder

Falls der Rauchwarnmelder ein Feuer erkennt, löst er eine im Melder enthaltene Sirene aus. Durch Drücken eines Reset-Tasters kann der Melder wieder in seinen ursprünglichen Bereitzustand zurückversetzt werden. Alle 30 Sekunden führt der Melder einen Selbsttest durch. Falls dieser erfolgreich ist, beginnt er wieder mit dem Erkennen von Rauch. Bei Fehlschlagen ist der Sensor als „defekt“ markiert.

BMS

Zu Beginn soll eine Meldung auf der Konsole erfolgen, dass der Systemstart durchgeführt wurde. Das BMS befindet sich dann im Zustand „Bereit“. Falls einer der (zahlreichen!) Rauchwarnmelder einen Alarm auslöst, wird sofort ein stiller Alarm auf der Brücke des Schiffes ausgelöst. Nach dieser Auslösung hat die Crew auf der Brücke des Schiffes zwei Minuten Zeit auf diesen Alarm zu reagieren. Erfolgt eine Reaktion der Crew mittels der Quittung-Taste wird der Brückenalarm deaktiviert. Die Crew hat nun die Aufgabe zu prüfen, ob es sich um einen Ernstfall oder einen Fehlalarm handelt. Ist es ein Fehlalarm, kann die Crew über eine AlarmDeaktivieren-Taste alle Alarmmerkmale deaktivieren und anschließend mittels der Start-Taste das System wieder neu starten. Ist der Alarmfall jedoch ein Ernstfall, so kann die Crew über einen separaten Alarm-Taster manuell den Hauptalarm auslösen (dieser Zustandsübergang erfolgt ebenfalls, falls die zwei Minuten ohne Reaktion der Crew vergangen sind). Daraufhin findet die externe Alarmierung der Reederei und benachbarter Schiffe, sowie der interne Alarm auf dem Schiff statt. Zusätzlich werden die Brandschutztüren geschlossen. Wenn der Schiffsalarm aktiv ist, kann dieser auch wieder deaktiviert werden. Hierzu müssen zwei Taster gedrückt werden (die Reihenfolge der Tasteneingabe ist hierbei beliebig). Danach kann durch einen weiteren Taster wieder in den Startzustand des BMS gewechselt werden.

Brandschutztür

Der Kunde hat noch keine Idee zur Funktion einer Brandschutztüre. Hier hast du freie Hand, wie diese funktionieren soll. Überlege dir die Funktion und erstelle auch hierfür ein State Chart!

Aufgabe 5: Modellierung in itemis CREATE

a) Setze nur das State Chart des Rauchwarnmelder in itemis CREATE um und führe eine Simulation durch. Hierzu benötigst du eine Academic License, die du kostenfrei unter https://info.itemis.com/products/itemis-create/academic/order/ beantragen kannst.

b) **Optional**:
   - Recherchiere, wie man „von Hand“ ein State Chart in C oder C++ Code umsetzen (d.h. selber programmieren!) kann.
   - Erzeuge C++Code in itemis CREATE für eine Implementierung deines Modells (du musst den erzeugten Code nicht verstehen können).

Viel Spaß!

Stefan & Enrico