

SOFTWARE ENTWICKLUNG 2

Wintersemester 2022

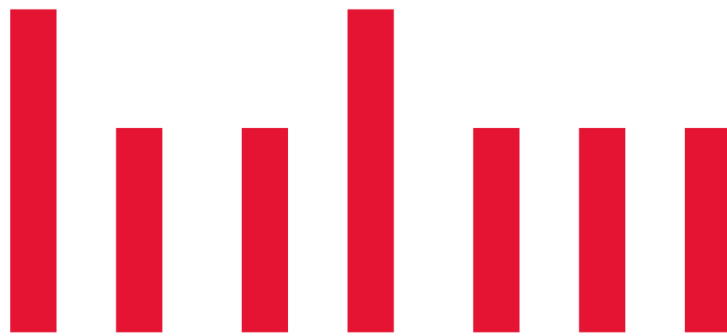
REISEPLANER

Coralie Abbenzeller

Nils Friedrich

Antonio Succurro

<https://gitlab.mi.hdm-stuttgart.de/ca057/reiseplaner>



HOCHSCHULE
DER MEDIEN

1. Kurzbeschreibung des Projekts

Bei unserem Projekt handelt es sich um einen simplen Reiseplaner mit Login und Profilfunktion, um einzelne Reisen Profilen zuzuordnen.

Man registriert ein Profil, loggt sich mit diesem ein und kann nun Trips verschiedener Art erstellen, diese und Profile können bearbeitet werden.

Durch Zeitmangel ging etwas an Realismus verloren, weswegen es sich eher um einen Reiseplanungssimulator handelt.

Durch die schlechte Zeiteinstellung unsererseits, ging auch einiges an Funktionen (hauptsächlich die Tripverwaltung) verloren.

Die Profilfunktionen sind jedoch alle wie geplant umgesetzt worden.

2. Start

Die Main-/Startmethode befindet sich in der Klasse MyTravel

3. Besonderheiten

Wir nutzten JDK 19

Die Create Trip Methoden funktionieren leider nicht, da wir noch nicht herausgefunden haben, wie wir die verschiedenen Trip-Typen mit GSON serialisieren, bzw. deserialisieren sollten.

4. UML-Klassendiagramme

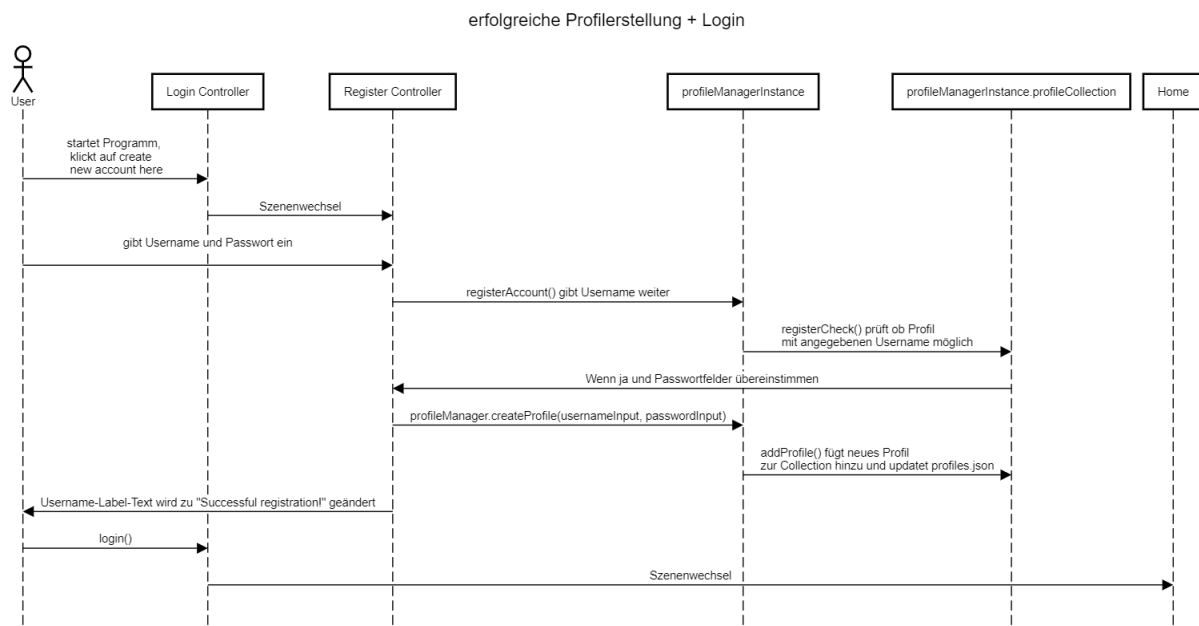
Klassendiagramm:

Wegen schlechter Lesbarkeit ist das Klassendiagramm im Git Repository unter reiseplaner/other

Use Case Diagramm:

Wegen schlechter Lesbarkeit ist das Use Case Diagramm im Git Repository unter reiseplaner/other

SequenceDiagram (Profilerstellung + Login):



5. Stellungnahme

Architektur:

Die MyTravel Klasse ist das Hauptstück des Programms, sie instanziert einen Travel- sowie einen ProfilManager, diese Verwalten von User erstellte Objekte.

"Implementations" beinhaltet die verschiedenen Trip Klassen und deren Parentklasse, Trip.

"Interfaces" beinhaltet das Manager Interface, das später vielleicht dazu gedient haben könnte, einen Destination Manager, der Reiseziele verwaltet, hinzuzufügen und List, welches verwendet worden wäre, um Pack- & Checkliste für Reisen zu implementieren.

Exceptions:

invalidAgeInputException: Eine Exception, die geworfen wird wenn die validateAge() Methode des ProfilManagers einen negativen Wert als Parameter hat, wenn dies bei der Bearbeitung eines Profils geschieht, wird das invalide angegebene Alter eines bearbeiteten Profils bei der Erstellung des neuen Profilobjekts ignoriert.

Tests:

Unser Programm verfügt über zwei Testmethoden, welche sich im Test-Ordner in der Klasse ProfileManagerTests befinden, testInvalidAgeException() testet ob die oben genannte Exception geworfen wird, wenn die validateAge() Methode einen negativen Int als Parameter übergeben bekommt.

beforeLoginGetLoggedInProfilesIsNull testet, ob das LoggedInProfile eines ProfileManagers wie beabsichtigt vor einem Login Nullwerte als Passwort und Username besitzt.

GUI (JavaFX):

controllerClasses enthält die Controller Klassen die für das GUI zuständig sind, die dazugehörigen XML files befinden sich unter resourcen -> fxml

Logging:

Alle Logs werden in ein File logs.log mit einem RollingFileAppender geschrieben, das eine Maximalgröße von 20MB besitzt.

Es werden verschiedene Logstufen verwendet und auch unsere invalidAgeInputException wird geloggt, wenn sie geworfen wird.

Threads:

MusicHandler:

Thread wird beim erstmaligen erfolgreichen Einloggen gestartet und spielt (mehr oder weniger) schöne Möwen- und Meeresgeräusche ab.

Streams und Lambda-Funktionen:

ProfileManager:

Zum Durchsuchen der ProfileCollection, um Login-Daten zu überprüfen (.loginCheck(), wird über LoginController aufgerufen), indem der Stream die Collection nach Profilen mit dem eingegebenem Username und passendem Passwort filtert, und bei Fund true returnt.

Zum Durchsuchen der ProfileCollection, um zu prüfen, ob ein neuregistrierter Account einen einzigartigen Username hat, hierzu wird die Collection nur nach Username gefiltert.

Factories:

Unser Projekt verfügt über zwei Factories, eine sehr simple, die nur Objekte eines Typs erstellt, den Profilmanager, der nur Profil-Instanzen bildet, darüber hinaus auch noch die Aufgabe, diese zu verwalten, übernimmt.

Eine aus technischer Sicht interessantere Factory ist der TripManager, welcher verschiedene Objekte, die das Interface TripType implementieren, erstellt und verwaltet.