



SOMMERSEMESTER 2022

SWEN2 – TOUR PLANNER

CARMEN KRATZER – if20b111

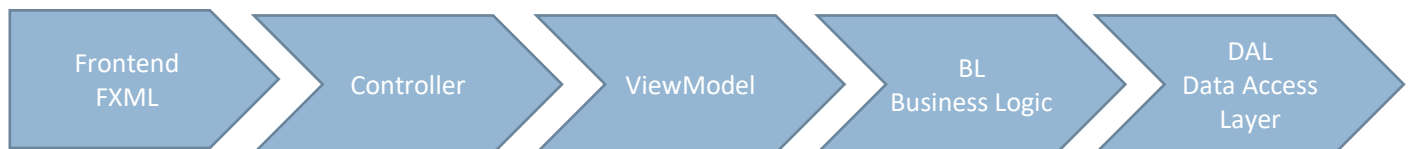
STEFANIE GLATZER – if20b092



SWEN2 – TOUR PLANNER

DESIGNS, FAILURES, SELECTED SOLUTIONS

MVVM



- Im Frontend verwenden wir FXML.
- Hier wird auf die Controller zugegriffen, welche über eine Controller-Factory verwaltet werden.
- Die Controller greifen auf die Funktionen der ViewModel zu.
- Diese wiederum können über die Business Logic auf die Data Access Layer zugreifen.

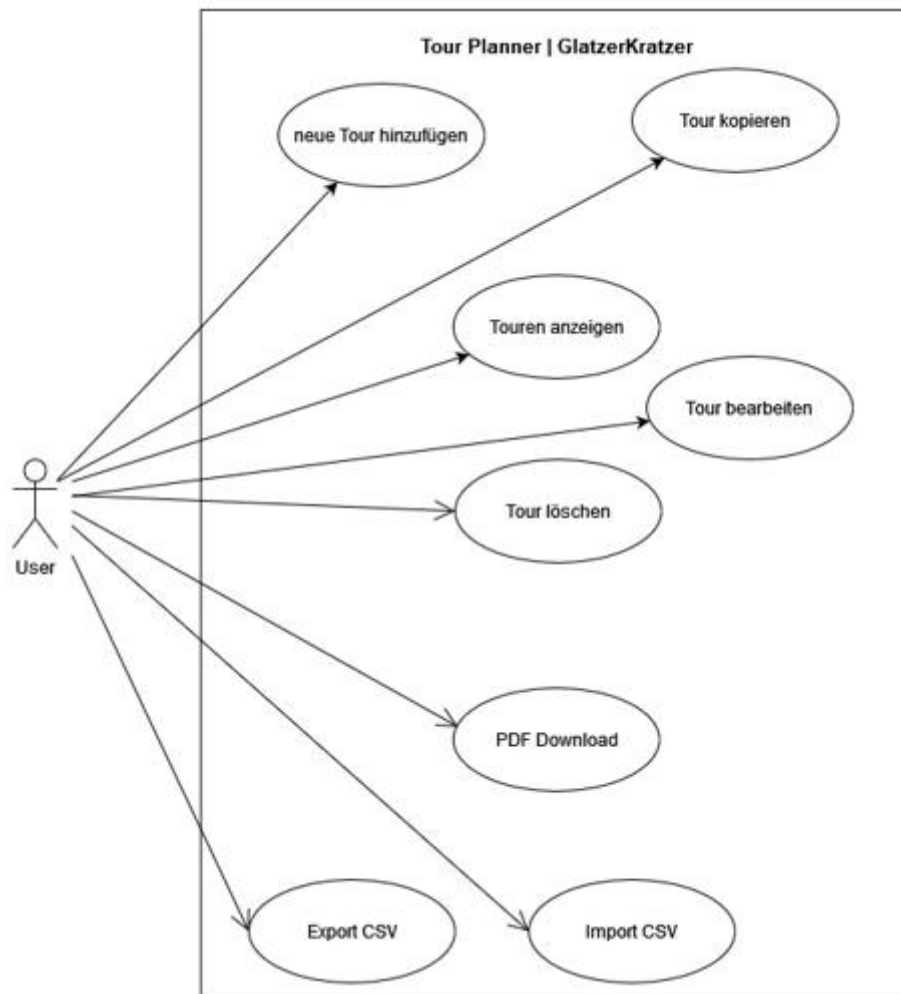
FAILURES & SELECTED SOLUTIONS

Failures	Selected Solutions
MVVM umsetzen	Mit Vorlage arbeiten
Fehlende Controller-Factory	YouTube-Tutorials, Internetsuche, Stackoverflow
Komplizierter Zugriff auf die einzelnen Controller in der Controller-Factory	Controller-Factory als Singleton Zugriffe über Getter
Hohe Ladezeiten für das Map-Quest	Mittels Threads theoretisch lösbar
Threading um MQ Ladezeiten zu umgehen: [failed] Threading von Grund auf nochmal lernen. Zeitlich leider nicht möglich.	
Probleme (vermutlich) mit PowerMockito bei den Unit-Tests	Konnte leider nicht gelöst werden
Probleme bei der Implementierung der Tour Logs TableView: falsche oder keine Anzeige, fehlerhafter Zugriff auf Datenbank	In letzter Minute gelöst: DB Zugriff angepasst, TourLog Klasse angepasst (mit Properties gearbeitet)

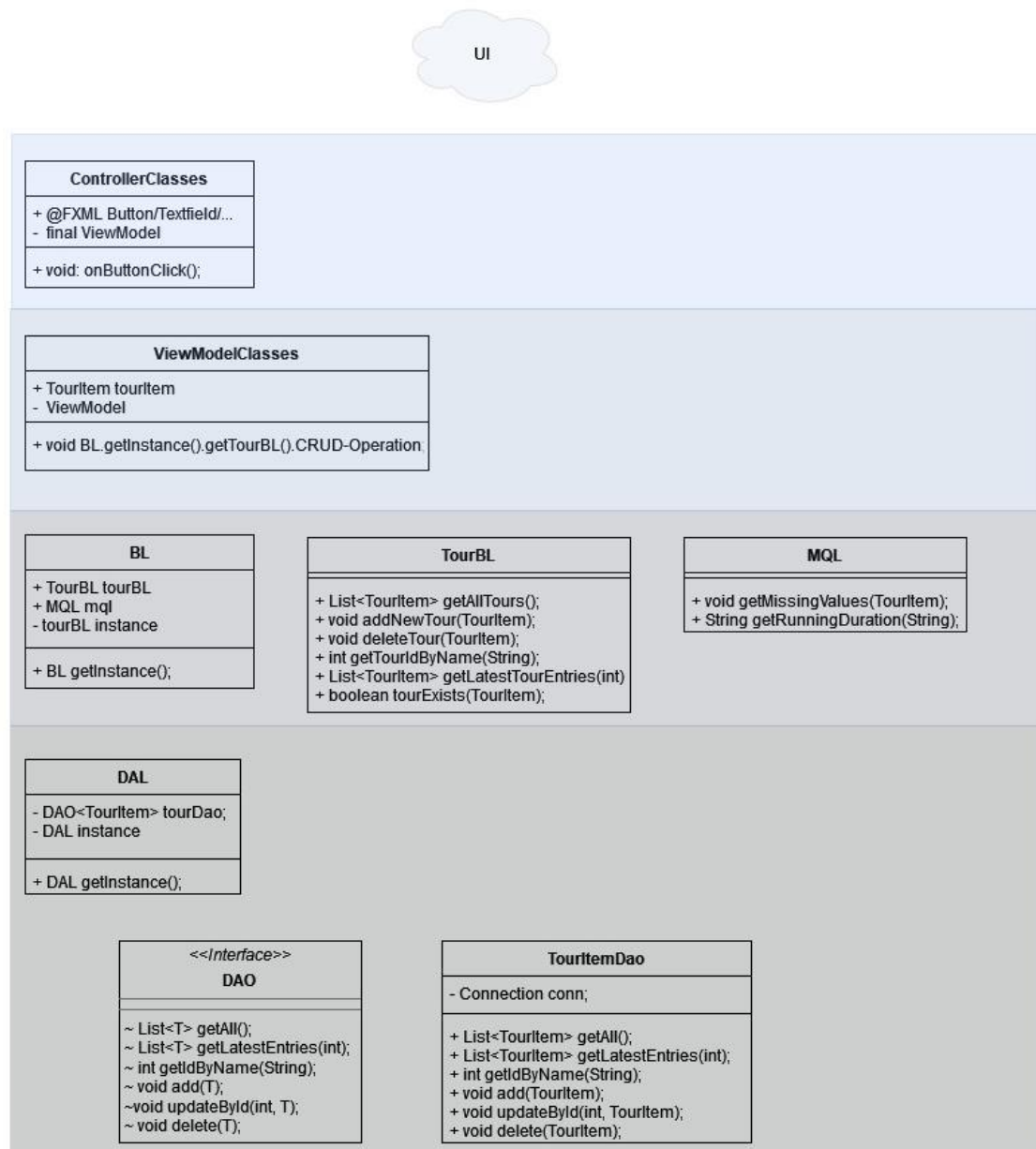
APPLICATION UI – WIREFRAMES

Tour Planner App						
File Edit Options Help						
Tours		<div>+ -</div>				
Search		<div>Description Map</div>				
<div></div>		<div></div>				
						<div></div>
<div>Search</div>						
created	difficulty	rating	duration	comment		

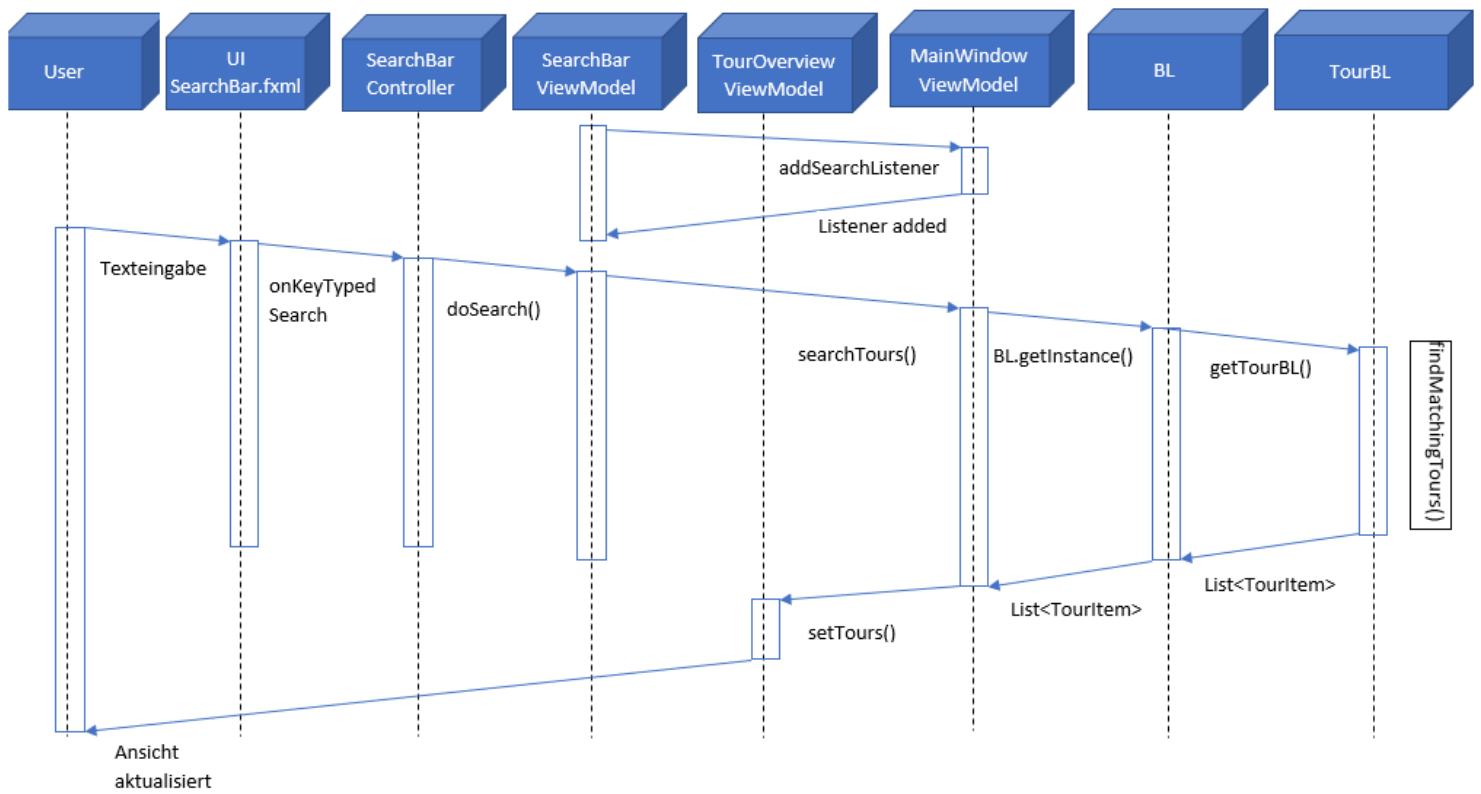
APPLICATION FEATURES – USE CASE DIAGRAMM



APPLICATION ARCHITECTURE – CLASS DIAGRAMM



APPLICATION ARCHITECTURE – SEQUENCE DIAGRAMM FÜR VOLLTEXT-SUCHE



(NOT SO) MANDATORY UNIQUE FEATURE

Geplant, nicht implementiert:

- **Tour Highlights**
User können Bilder zur ausgewählten Tour hochladen (Aussichtspunkt, Restaurant etc) und eine zugehörige kleine Map sehen.
- **Filter Funktion für Touren und Tour Logs**
Beim Klicken des Filter-Buttons öffnet sich ein Fenster und Filteroptionen, wie Transporttyp, Minimal-/Maximaldistanz, Minimal-/Maximaldauer können, können ausgewählt werden und die Liste wird danach gefiltert.

Kleinere „Bonus-Features“ implementiert:

- **Deutsche/Englische App Version**
Unter „Optionen“ hat man die Möglichkeit zwischen deutscher und englischer Version der App zu wählen. Alle Komponenten (bis auf die in der Datenbank gespeicherten Eingaben) werden in die jeweilige Sprache übersetzt per Zugriff auf das ResourceBundle und Zuweisung der ausgewählten Sprache.
- **Refresh Buttons für Tourliste und Logliste**
Ein kleineres Feature um die Liste zu aktualisieren. Sinn dahinter soll sein, dass bei eventuell zukünftiger Multi-User Anwendung (sofern zukünftig ein REST Server implementiert würde) neu eingetragene Elemente in die aktuelle Liste „gezogen“ werden.

UNIT TESTS

Wir haben versucht, die Datenbank zu mocken, und um den Constructor zu umgehen, wollten wir Powermockito verwenden. Es ist uns allerdings nicht gelungen, weshalb wir nur einfache Unit-Tests implementiert haben.

Sinnvolle Unit-Tests (halbwegs):

- Die Überprüfung der Datenbank Zugriffsdaten (Datenbank soll nicht null sein)
- Überprüfung der Datenbank Connection nach erfolgreichen Zugriff (soll ebenfalls nicht null sein)
- Die Überprüfung der MapQuest Zugriffsdaten

ZEITAUFZEICHNUNG

Pro Person haben wir etwa 70 Stunden für das Projekt aufwenden müssen.

(Eine genaue Auflistung ist dem fokussierten Arbeiten am Projekt zum Opfer gefallen.)

GIT REPOSITORY LINK

https://github.com/Leyaera/SWEN2_TourPlanner.git