BIF4
Software Engineering 2

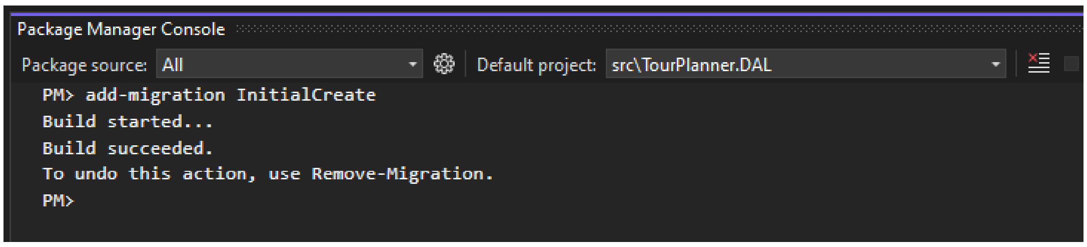
Stefan Kellner
Natalia Guarnizo

# SWEN2 Protocol

## GitHub Link

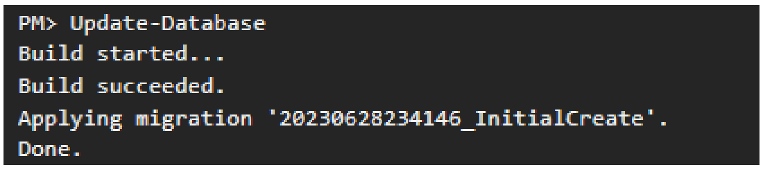
https://www.github.com/StefanK20/SWEN2

After cloning everything, to set up the database one must change directory to TourPlanner.DAL and enter “**Add-Migration InitialCreate**” in the Package Manager Console. If the Folder "Migrations" still exist, you will have to remove it first. And if everything went well it should look like this:



```
Package Manager Console
Package source: All    Default project: src\TourPlanner.DAL
  PM> add-migration InitialCreate
  Build started...
  Build succeeded.
  To undo this action, use Remove-Migration.
  PM>
```

After that, while pgAdmin is open or any other database of your choice, one must enter “**Update-Database**” in the Package Manager console and the database “tourplanner” with the two tables “Tours” and “Logs” should be created:



```
PM> Update-Database
Build started...
Build succeeded.
Applying migration '20230628234146_InitialCreate'.
Done.
```

For more information on the database configuration and more see TourPlanner/Config/settings.json.

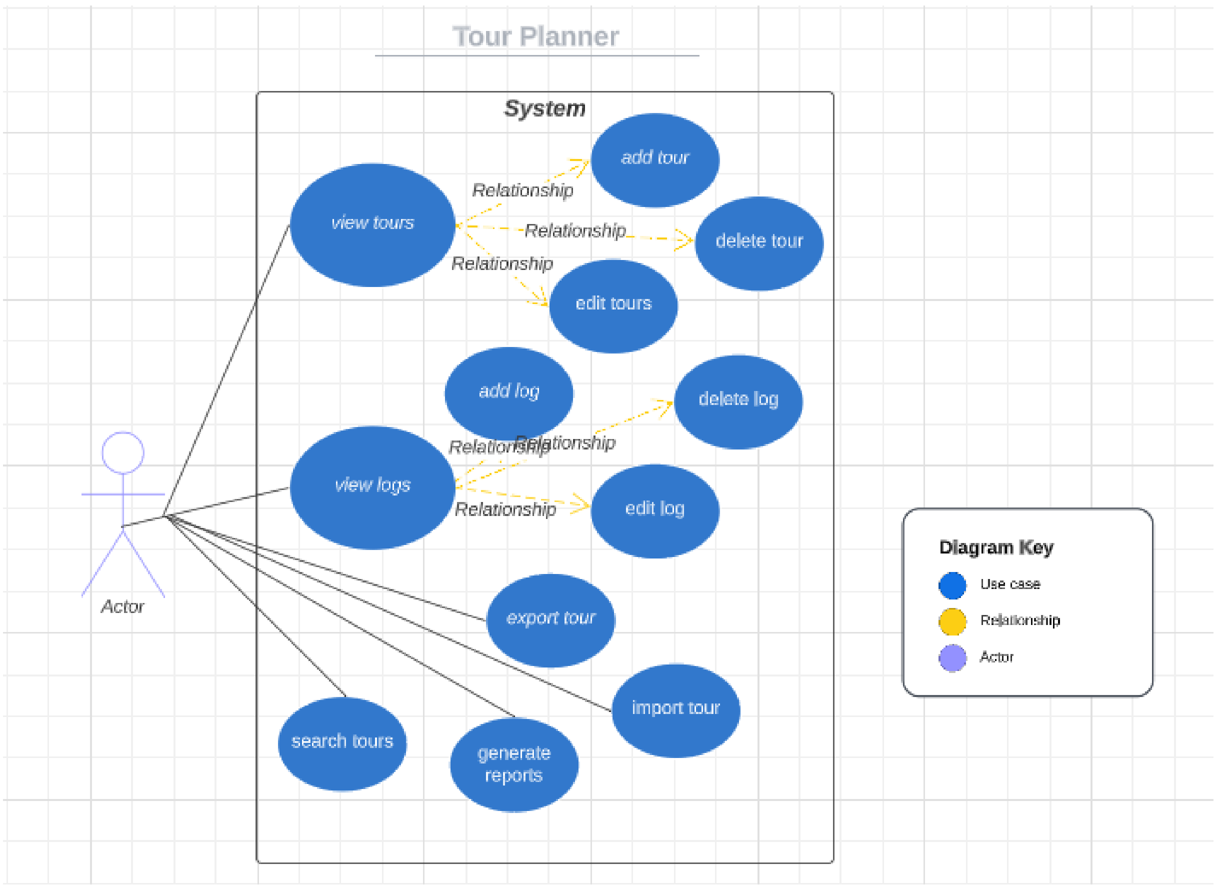
## Architecture

This C# WPF application was developed with the MVVM Design Pattern. All the logic of View components is defined in the corresponding ViewModels.

The project is structured in 3 layers: a Presentation Layer (TourPlanner), a Business Logic Layer (Tourplanner.BL) and a Data Access Layer (TourPlanner.DAL).

The Unit Tests are in a project called “TourPlanner.Tests” and the Models in “TourPlanner.Models”.

## Use case Diagramm



## Implemented Design patterns

The implemented patterns were:

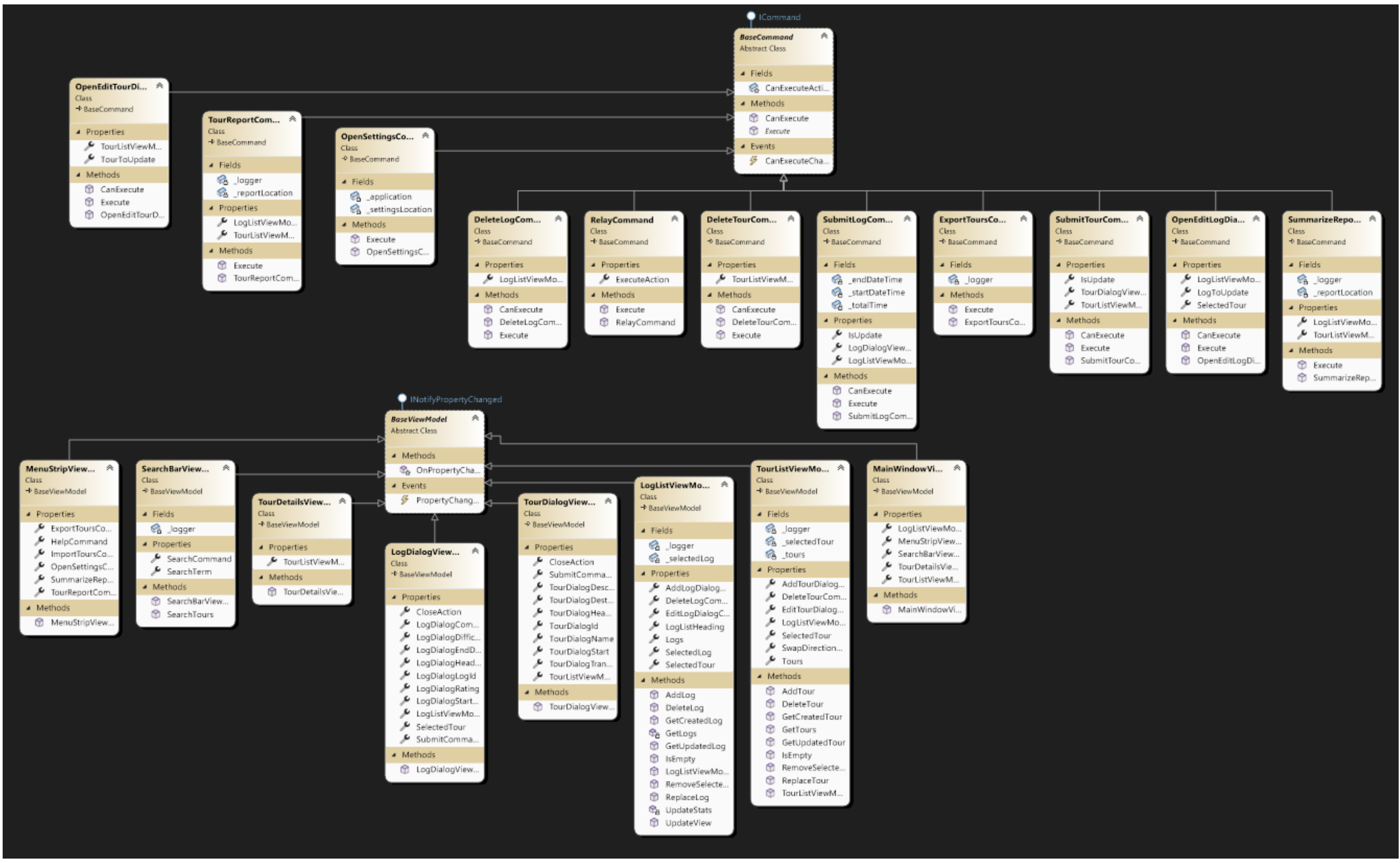
- Command Pattern
- Factory Pattern

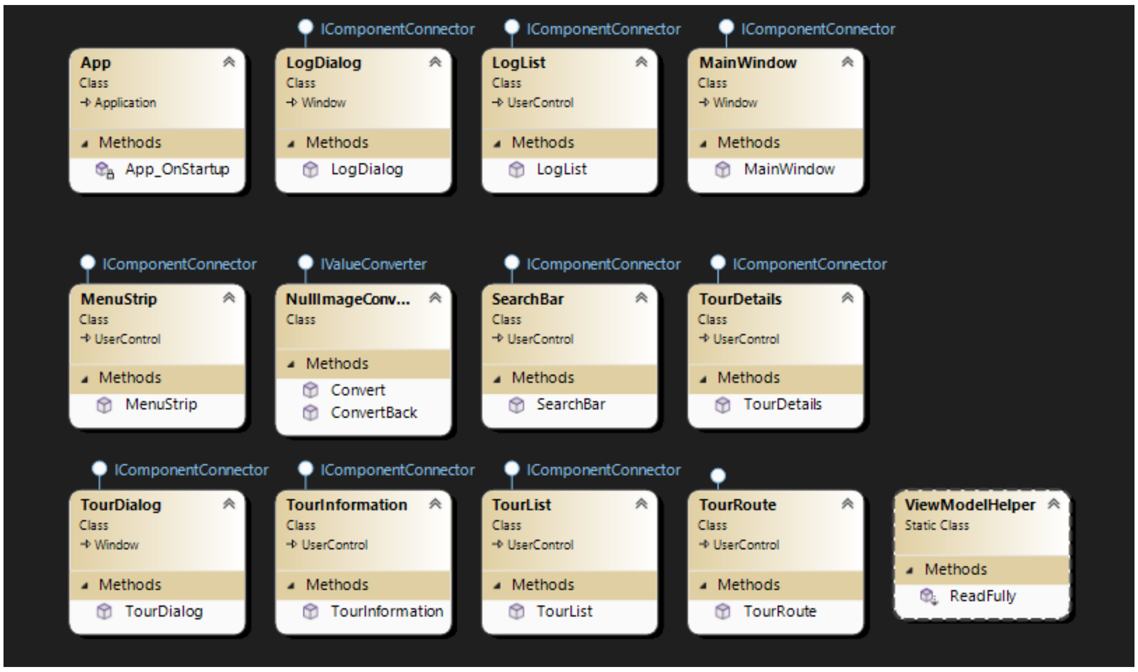
## Unique feature

Our unique feature is a button that swaps the start and destination of the selected tour, since we thought that this could be a good feature to have.

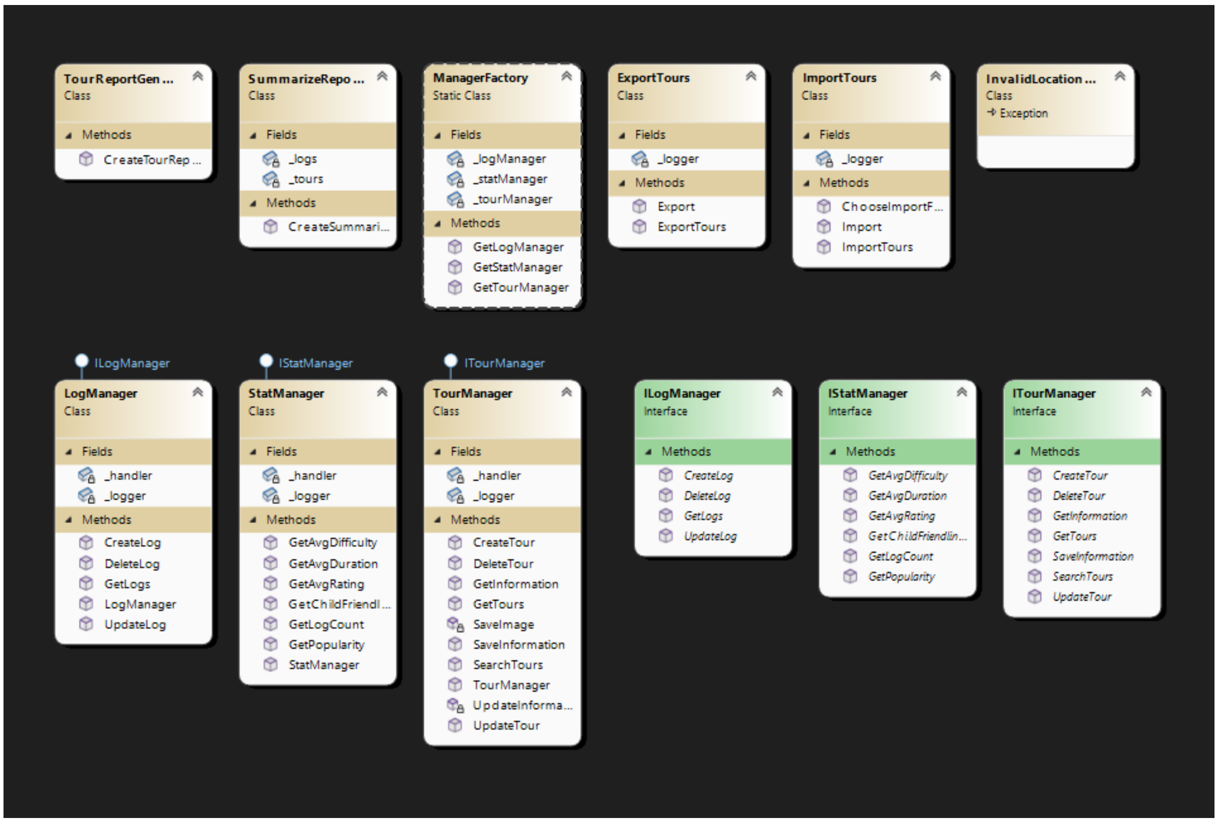
## UML Diagram

### *Presentation Layer*

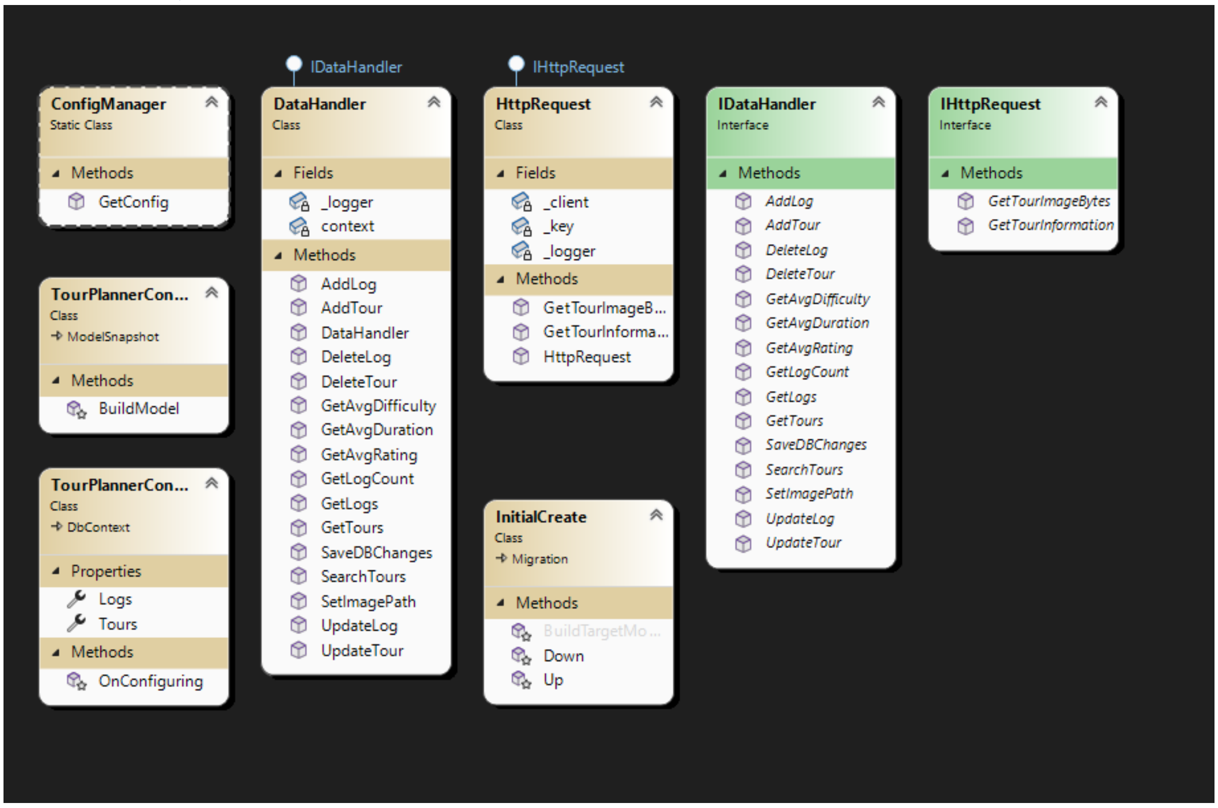




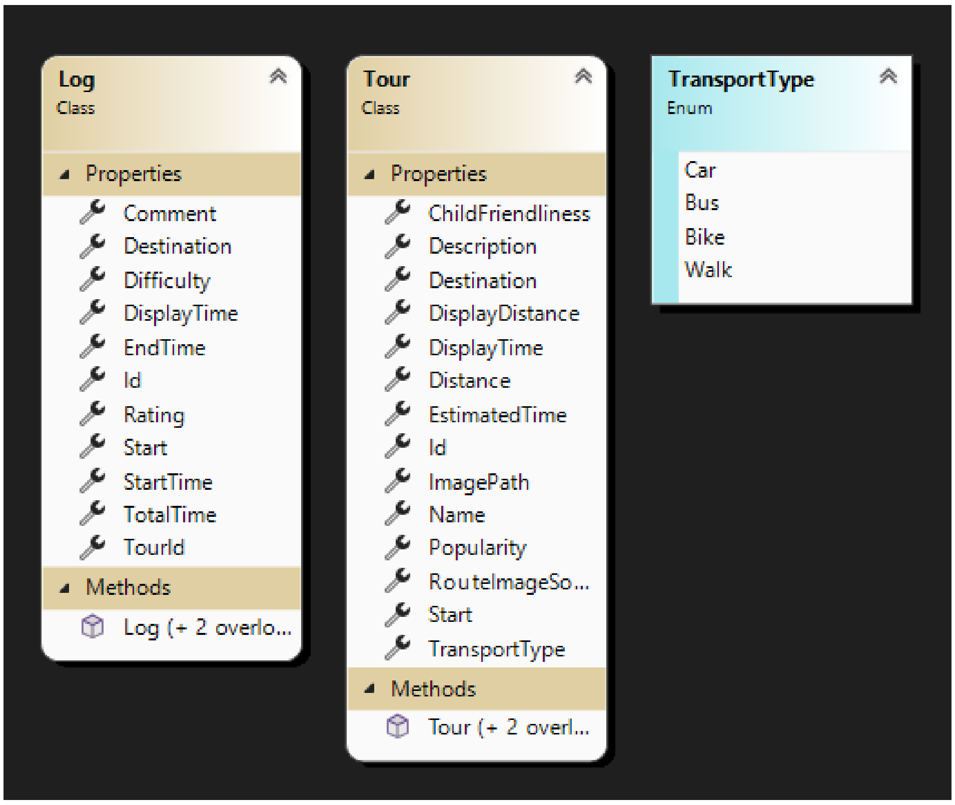
## *Business Layer*



## *Data Access Layer*

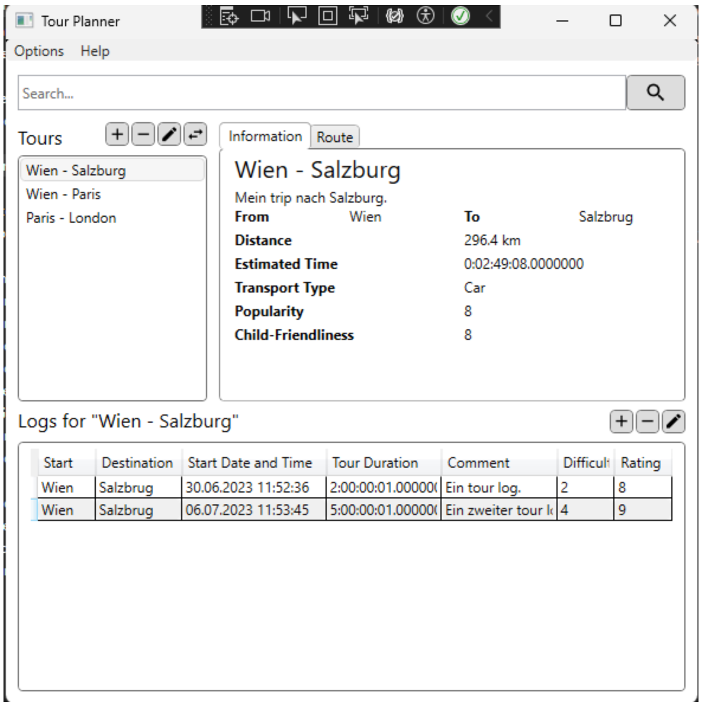


### *Models*



## UX, library decisions and lessons learned

### *Tour list and log view*



Wien - Salzburg
Wien - Paris
Paris - London

Wien - Salzburg
Mein trip nach Salzburg.

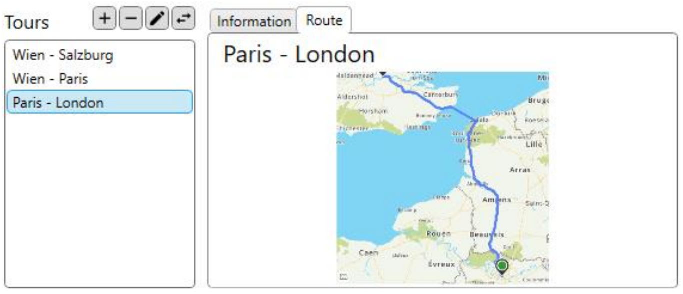
| | | | |
|---|---|---|---|
| **From** | Wien | **To** | Salzbrug |
| **Distance** | | 296.4 km | |
| **Estimated Time** | | 0:02:49:08.0000000 | |
| **Transport Type** | | Car | |
| **Popularity** | | 8 | |
| **Child-Friendliness** | | 8 | |

Logs for "Wien - Salzburg"

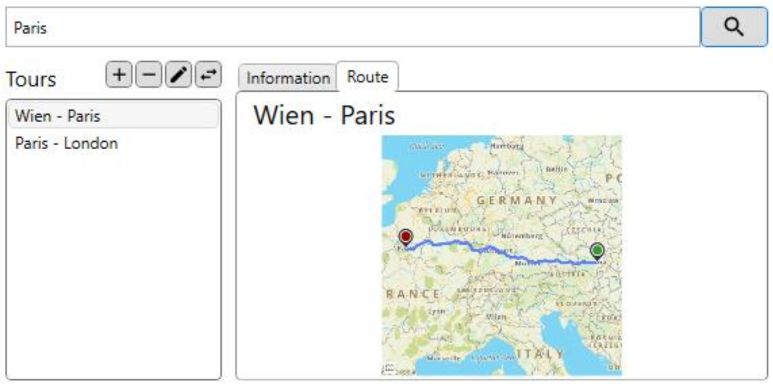
| Start | Destination | Start Date and Time | Tour Duration | Comment | Difficul | Rating |
|---|---|---|---|---|---|---|
| Wien | Salzbrug | 30.06.2023 11:52:36 | 2:00:00:01.00000( | Ein tour log. | 2 | 8 |
| Wien | Salzbrug | 06.07.2023 11:53:45 | 5:00:00:01.00000( | Ein zweiter tour lc | 4 | 9 |

Stefan Kellner
Natalia Guarnizo

### *Map view*



### *Search Tours*



We decided to hold everything very clear and simple, so that users do not get overwhelmed when using the Application. The UI technology we used was based on markup language with XAML files and we used a MVVM Model. We used the MapQuest API to get our map images, distance, time and more. For the Logs we used the Microsoft Logging library in our Log Manager.

When we were almost done with the project, we noticed that we were missing a Must-Have from the List. The problem was that we did not implement an OR-Mapping library.

So, lessons learned were that we should have read the Specification more thoroughly and should have taken more time to do this project. If we had done that, we would have noticed the missing OR-Mapping earlier and we would not have to stress the way we did. We lost a lot of precious time that we could have spent searching for bugs and other things.

## Unit Tests

We chose to do our Unit Tests about various things. First, we did some for our View Models, since it is an important part of our MVVM Model, and it is important that they load. We also did some Command Unit Tests, as well as Data Handler Tests. We also Tested our Models, to see if everything saves correctly. We tried to focus on doing Unit Tests, without doing any Integration Tests.

## Time spent

Stefan: around 47 hours

Natalia: around 41 hours