**Protocol – Tour\_Planner**

App architecture:

The App is split in 3 layers: User Interface (UI), Business Layer (BL) and Data Access Layer (DAL).  
The User Interface includes the MVVM pattern: Model – View – ViewModel. When starting the application, the App.xaml.cs creates the ViewModels and the MainViewModel. The ViewModels are the bound to the Views (Databinding) and get/show the data.  
In the BL the logic is implemented. The Controllers are called from the UI, handling requests like adding and deleting tours/logs, create the map for the tours through the MapQuest calls and then they send the data to the DAL. The two DataHandlers in the DAL handle the incoming requests from the BL, get the SQL Statements from the SqlCommand classes and return the data from the postgre sql database.

Use Cases:

Use Case 1: Add Tour  
The User adds a new tour via the User Interface. In the UI the user enters the data of the tour and this data is sent to the Tour Controller in the Business Layer. The Tour Controller creates the map with the MapQuest included, and sends this data to the TourDataHandler in the Data Access Layer. There the TourDataHandler saves the data in the postgresql database with the provided SQLCommands.

Use Case 2: Delete Tour  
The User selects a tour from the Shown Tours in the UI. The data is sent to the Tour Controller which calls the TourDataHandler to delete the selected Tour with the provided SQLCommands.

Use Case 3: Add Log  
The User selects a tour from the Shown Tours and adds a new Log to it. The selected tour and the data input from the user is collected in the UI and sent to the LogController. The LogController in the Business Layer calls the LogDataHandler in the Data Access Layer which saves the new log in the database with the provided SQLCommands.

UX, library decisions, lessons learned:

Design pattern:

We implemented the Singleton design pattern for the logger, database.

Unit Testing Decisions:

Tracked time:

Git Link:

https://github.com/BeccaDreams/SWEN2\_Tour\_Planner