

# Globe

---

PROJECT FOR THE “INGEGNERIA DEL SOFTWARE E PROGETTAZIONE WEB” COURSE

**Teo Miozzi | 0309010 & Simone Tummolo | 0309116**

# INDEX

<b>1. SOFTWARE REQUIREMENT SPECIFICATION .....</b>	<b>2</b>
1.1. INTRODUCTION .....	2
1.1.1. <i>Aim of the document</i> .....	2
1.1.2. <i>Overview</i> .....	2
1.1.3. <i>HWe SW requirements</i> .....	2
1.1.4. <i>Related systems (at least 2), Pros and Cons.</i> .....	2
1.2. USER STORIES .....	2
1.3. FUNCTIONAL REQUIREMENTS .....	3
1.4. USE CASES.....	3
1.4.1. <i>Overview Diagram</i> .....	3
1.4.2. <i>Internal Steps</i> .....	4
<b>2. STORYBOARDS.....</b>	<b>4</b>
<b>3. DESIGN.....</b>	<b>5</b>
3.1. CLASS DIAGRAM.....	8
3.1.1. <i>VOPC</i> .....	8
3.1.2. <i>Design-level diagram</i> .....	8
3.2. ACTIVITY DIAGRAM .....	13
3.3. SEQUENCE DIAGRAM.....	15
3.4. STATE DIAGRAM.....	17
<b>4. TESTING.....</b>	<b>18</b>
<b>5. CODE.....</b>	<b>18</b>
<b>6. VIDEO .....</b>	<b>19</b>

# 1. Software Requirement Specification

## 1.1. Introduction

### 1.1.1. Aim of the document

The main point of the following document is present objectively the fundamental requirements and specific ones of the system *Globe*. So, we will be doing a review of all the important internal and external aspects of the system with the help of some diagrams.

### 1.1.2. Overview

*Globe* is an application designed to facilitate the creation of travel itineraries.

User can request itineraries to travel agency showing what he wants to visit and view agency proposal that he can accept, paying proposal's price, or reject. Moreover, user can generate itineraries by selecting cities and attractions he wants to visit, adding accommodation and flight departure and arrival time (this function is available for guest users too – demo mode).

Agency can view user requests and create a proposal to send to the user and view if its proposals have been accepted or not.

### 1.1.3. HW e SW requirements

The hardware requirements are:

- A computer or a laptop
- An internet connection for city and attraction search
- Some space in the disc to save files if the demo version is on.

The software requirements are:

- Oracle OpenJDK 22 correctly installed
- A server or a local server that can holds the DBMS to obtain information by queries in SQL

### 1.1.4. Related systems Pros and Cons.

- Sygic Travel
  - Pros: it offers additional features such as booking tour packages, sharing itineraries and the multi-device account access support.
  - Cons: users must create a new itinerary for every city when planning multi-city trips and it includes a smaller number of attractions.
- SiVolà
  - Pros: it includes an intuitive and user-friendly interface and offers the possibility to purchase a pre-tested itinerary from the agency for a unique experience.
  - Cons: it doesn't offer the option to create itineraries independently or to request a custom-made itinerary.

## 1.2. User Stories

### Teo Miozzi | 0309010:

- As a user, I want to get the shortest route, so that I can visit all the attractions on my itinerary.
- As an agency, I want to see all attractions and cities user want to visit, so that I can create a complete itinerary.
- As a user, I want to see the details of the agency's proposal and the related itinerary, so that I can decide to accept or reject the proposal.

### Simone Tummolo | 0309116:

- As a user, I want to request an itinerary indicating its type, so that the agency can create the best itinerary for me.
- As an agency, I want to see the history and status of the proposal made to the user, so that the agency offers a more efficient service to the costumers.
- As a user, I want to rate the agency after payment, so that I can help other users choose the best agency.

### 1.3. Functional Requirements

**Teo Miozzi | 0309010:**

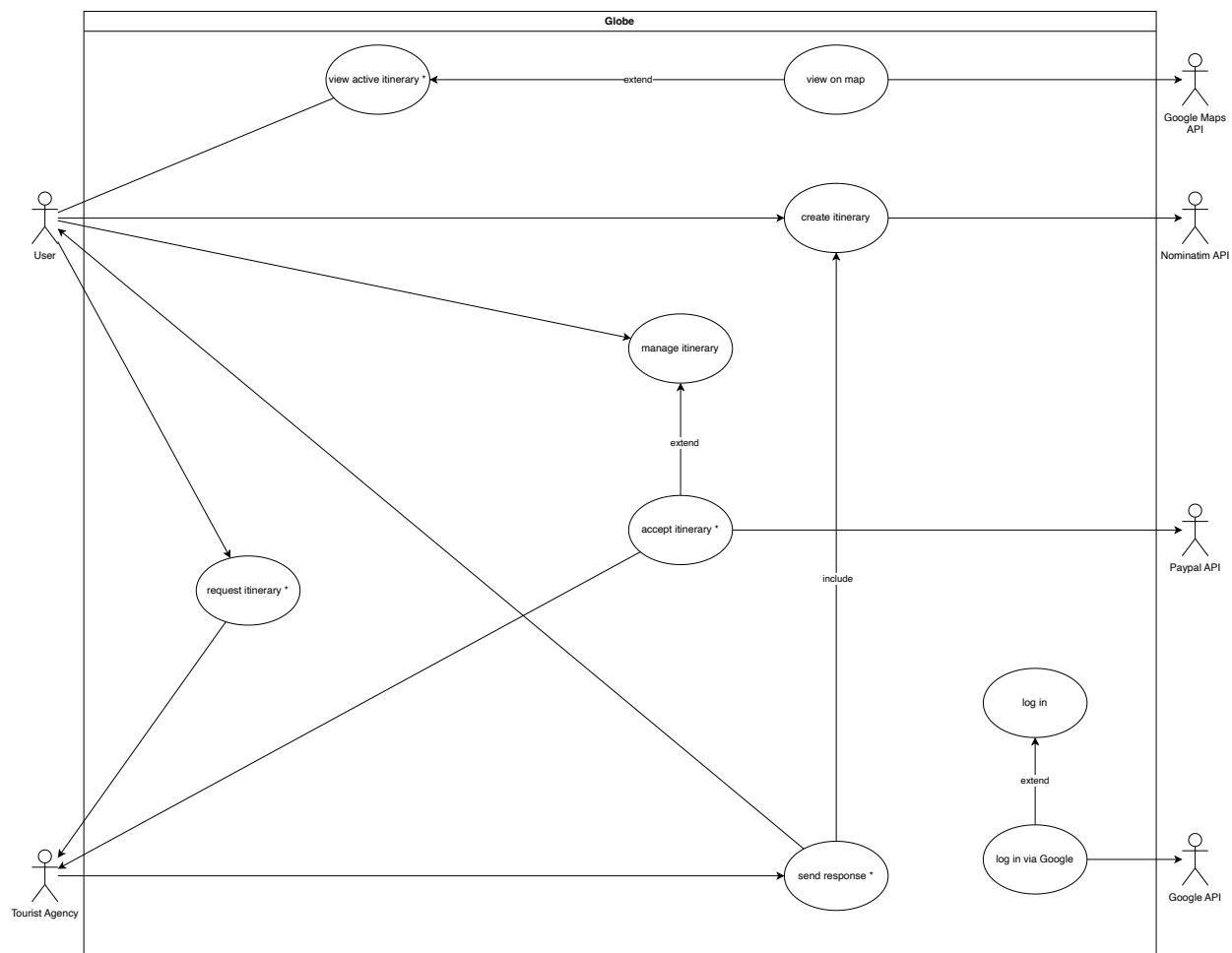
- The system shall provide a real-time updated map to see the user's active itinerary.
- The system shall find the available agencies according to the itinerary's types requested by the user.
- The system shall calculate the shortest route between the attractions selected by the user.

**Simone Tummolo | 0309116:**

- The system shall provide a secure payment method for the user to use.
- The system shall save and display to the user the proposal's itinerary after payment is completed.
- The system shall notify users when their request is accepted by an agency.

### 1.4. Use Cases

#### 1.4.1. Overview



Note: the use cases marked with "\*" include the Log In use case

### 1.4.2. Internal Steps

#### Teo Miozzi | 0309010:

Name: Create Itinerary

1. The user Logs In.
2. The user fills the creation form with itinerary details
3. The user searches cities and attractions he wants to add.
4. The system finds cities and attractions from the API.
5. The user selects correct cities and attractions.
6. The user submits the form to confirm the itinerary.
7. The system verifies itinerary's data.
8. The system calculates the optimal route.
9. The System saves the itinerary.

Extensions:

4a. CITY OR ATTRACTION NOT FOUND

System notifies the user about the city or the attraction it doesn't found.

4b. API NOT RESPONDING

System notifies the user about the problem.

7a. DATA NOT VALID

System notifies the user about the incorrect field.

#### Simone Tummolo | 0309116:

Name: Request itinerary

1. The user Logs In.
2. The user searches for cities and attractions to include in the itinerary request.
3. The system retrieves matching cities and attractions from the API.
4. The user selects the desired cities and attractions from the results and compile the request form.
5. The system validates the data in the request.
6. The user searches for agencies.
7. The system finds the agencies based on the entered criteria.
8. The user selects the agencies to send the request to.
9. The system sends the validated request to the selected agencies.

Extensions:

5a. DATA NOT VALID

System notifies that the data entered is not valid.

7a. AGENCY NOT FOUND

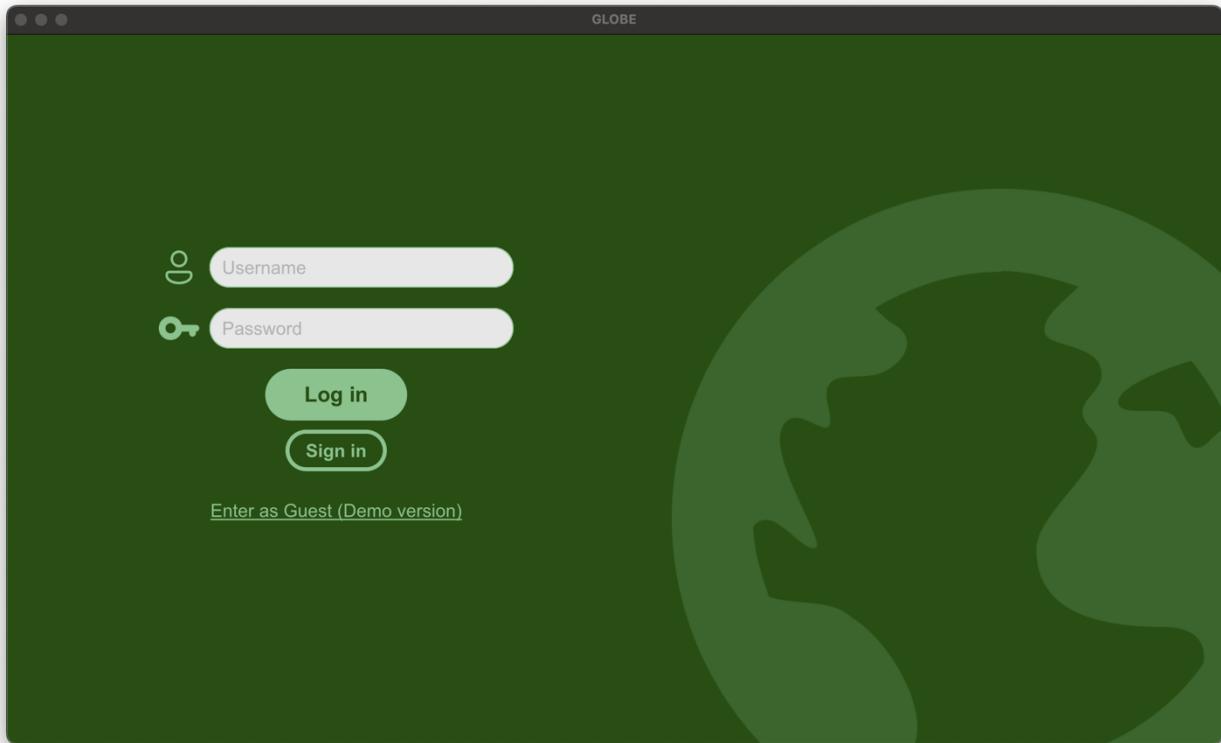
System notifies the user that there aren't agencies available based on the user's input.

8a. NO AGENCY SELECTED

System notifies the user that he has not selected any agency.

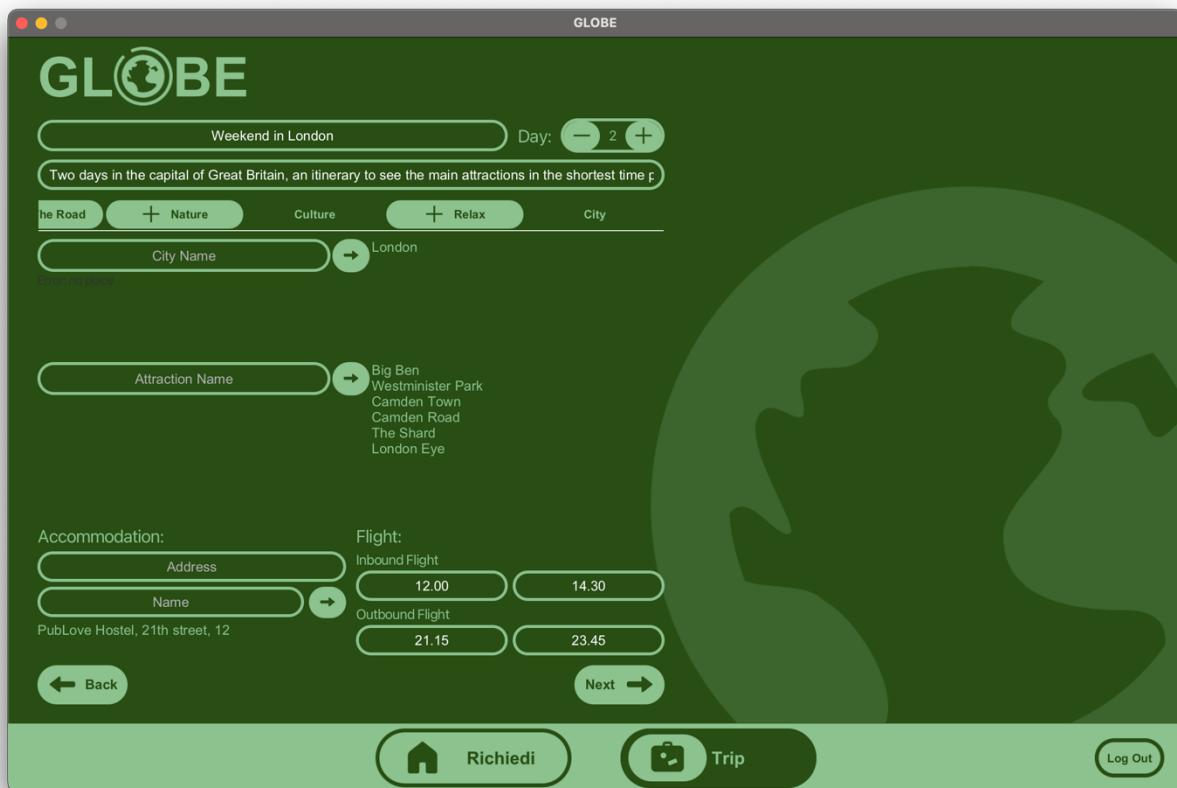
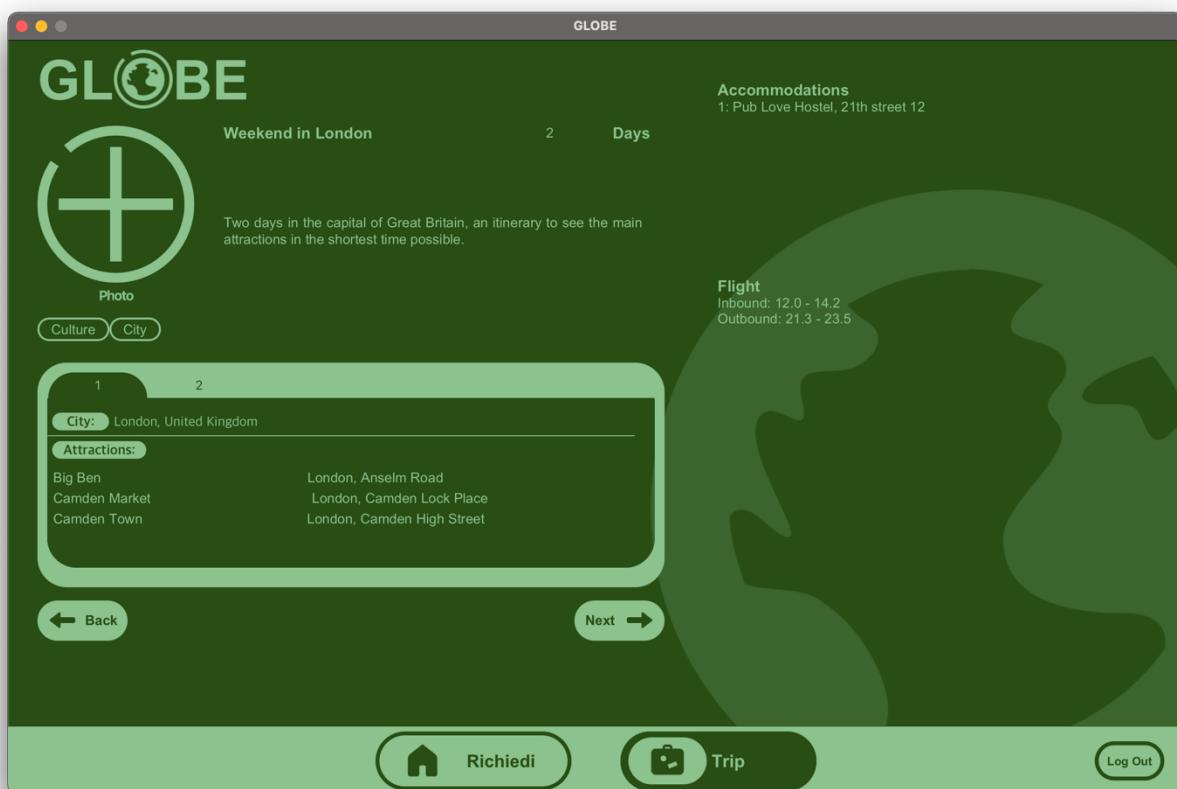
## 2. Storyboards

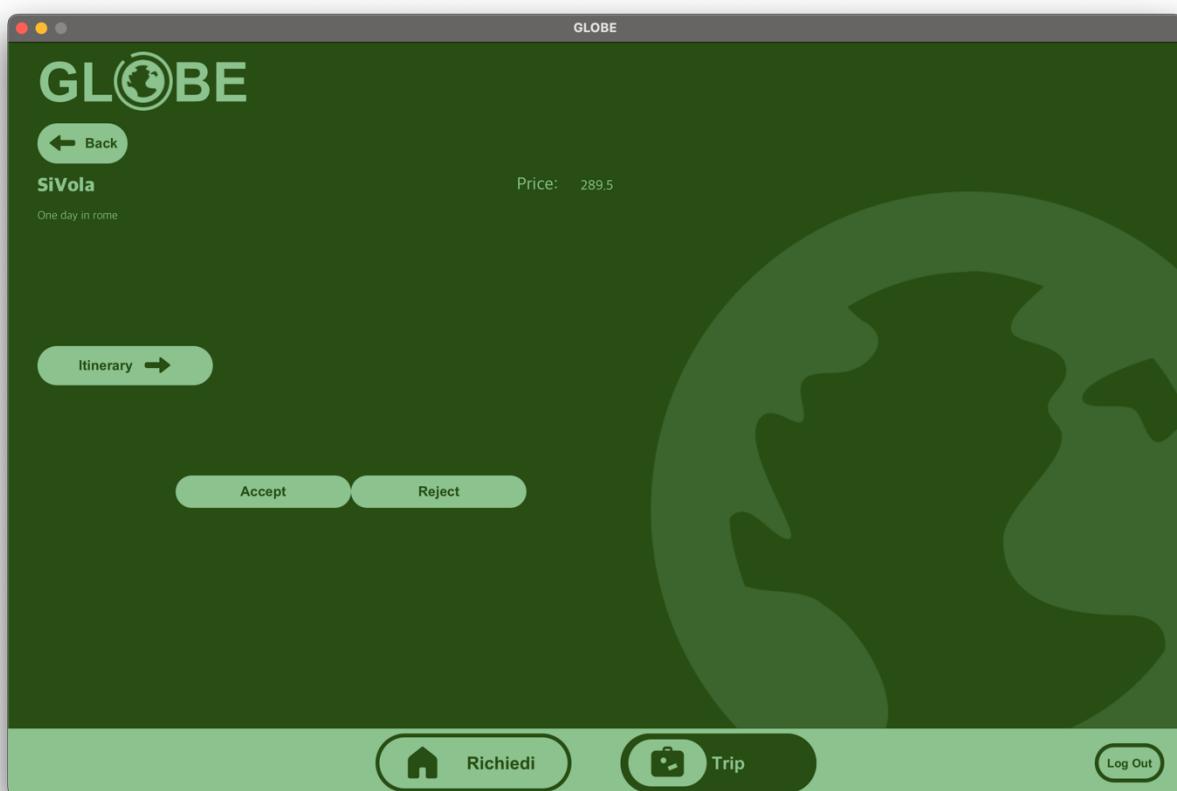
Log In:



Manage Itinerary:

The itinerary management screen shows the 'GLOBE' logo at the top. Below it, there are two tabs: 'My Itinerary' (selected) and 'Agency Proposal'. A large '+' button is on the left. Two itinerary cards are displayed: 'Trip to Italy' (15 Days) featuring a Roman Colosseum image, and 'Weekend in London' (2 Days) featuring a London skyline image. At the bottom, there are three buttons: 'Richiedi' (with a house icon), 'Trip' (with a suitcase icon), and 'Log Out'.

**Create Itinerary:****Display Itinerary:**

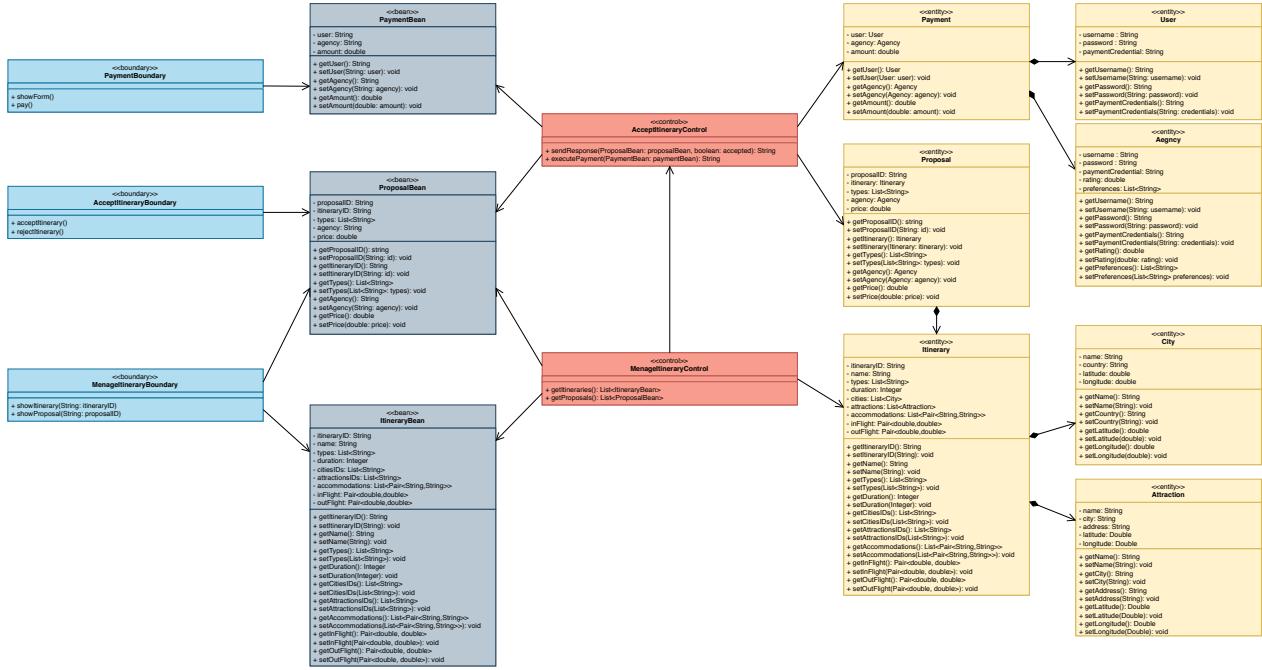
**Create Request:****Accept Proposal:**

## 3. Design

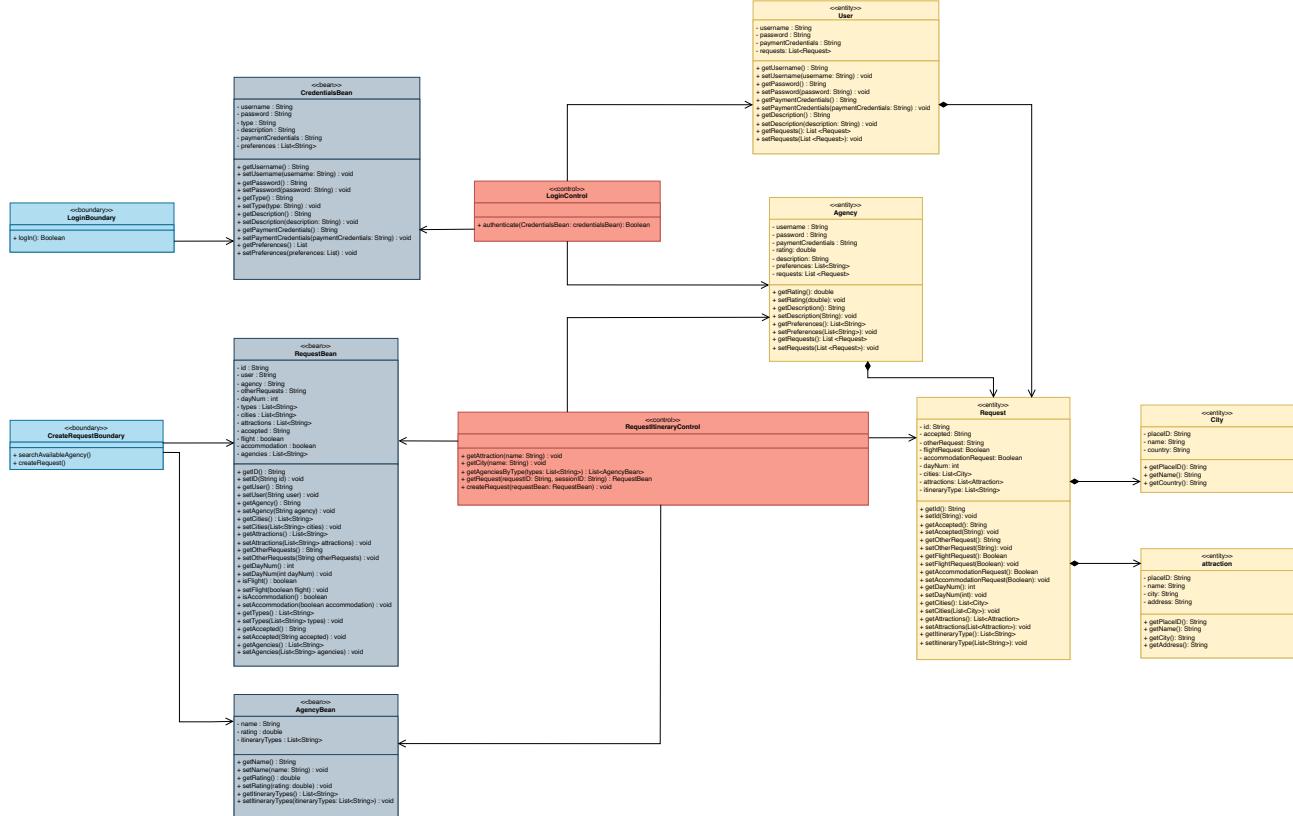
### 3.1. Class diagram

### 3.1.1. VOPC

Teo Miozzi | 0309010:



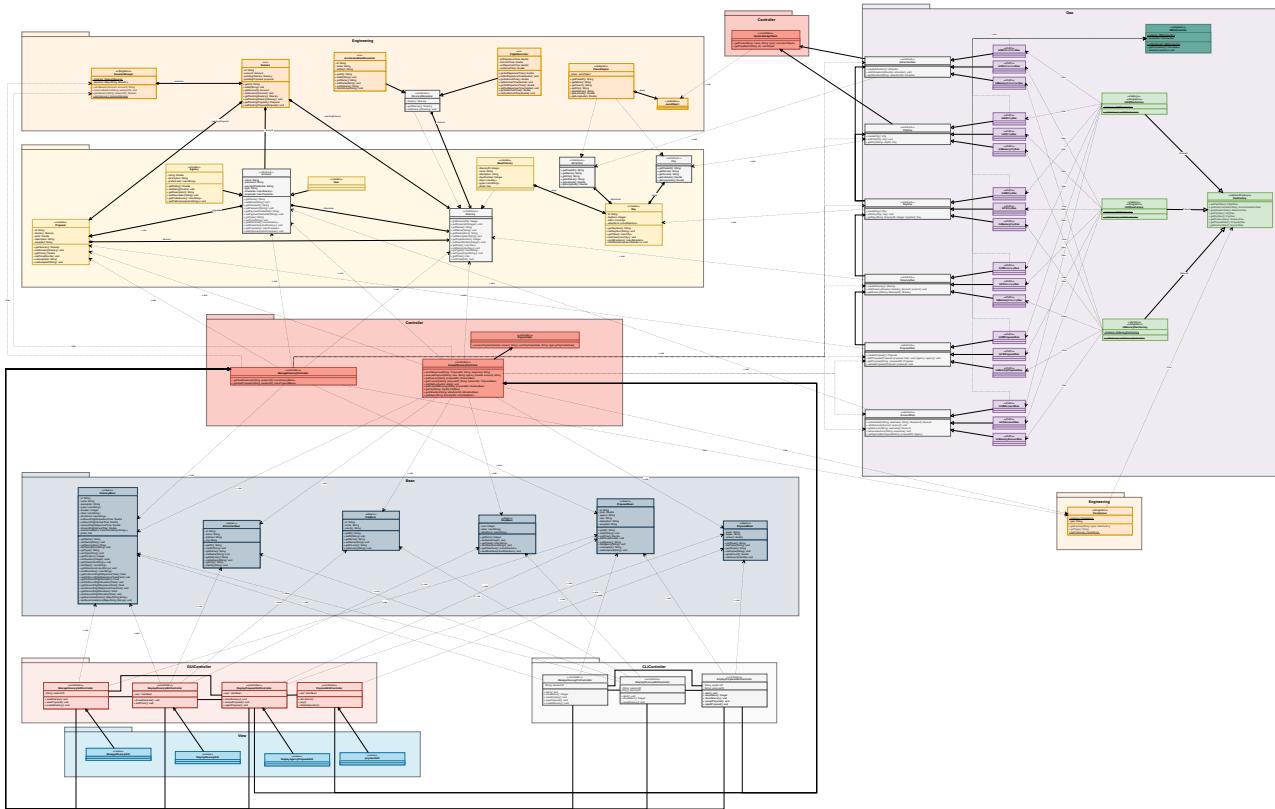
**Simone Tummolo | 0309116:**



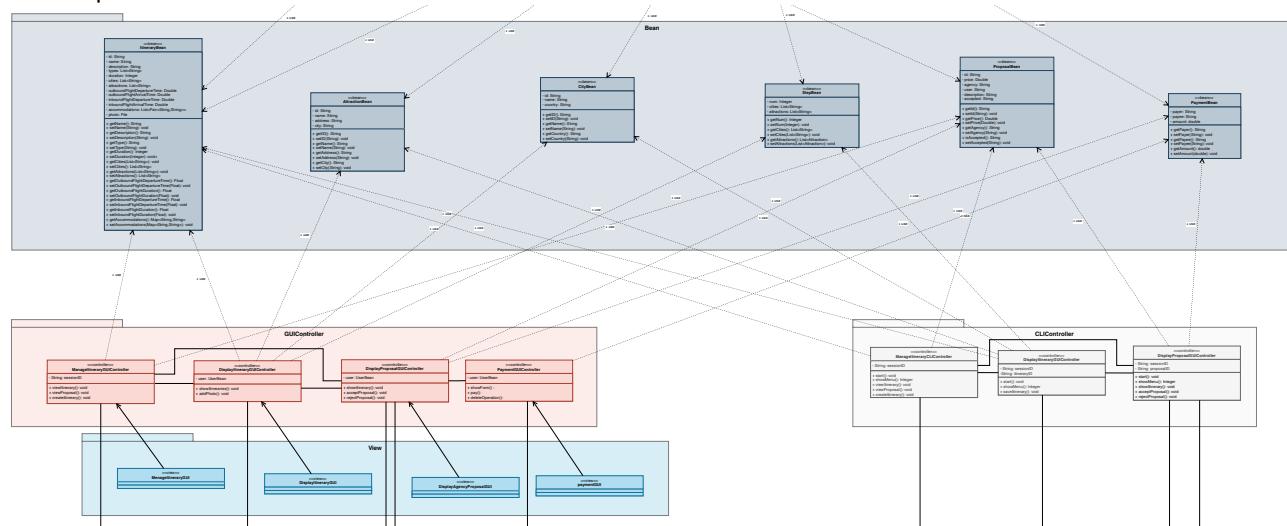
### 3.1.2. Design-level diagram

**Teo Miozzi | 0309010:**

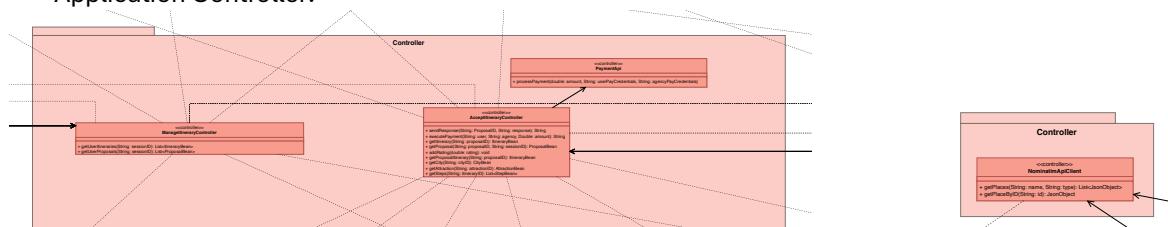
Complete Diagram:



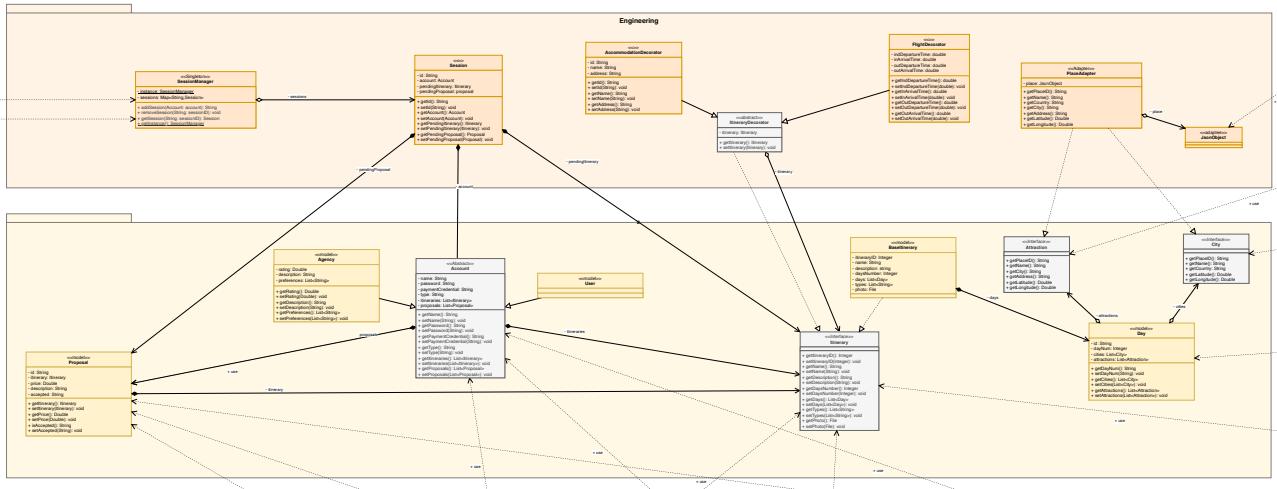
Graphic Controller and Bean:



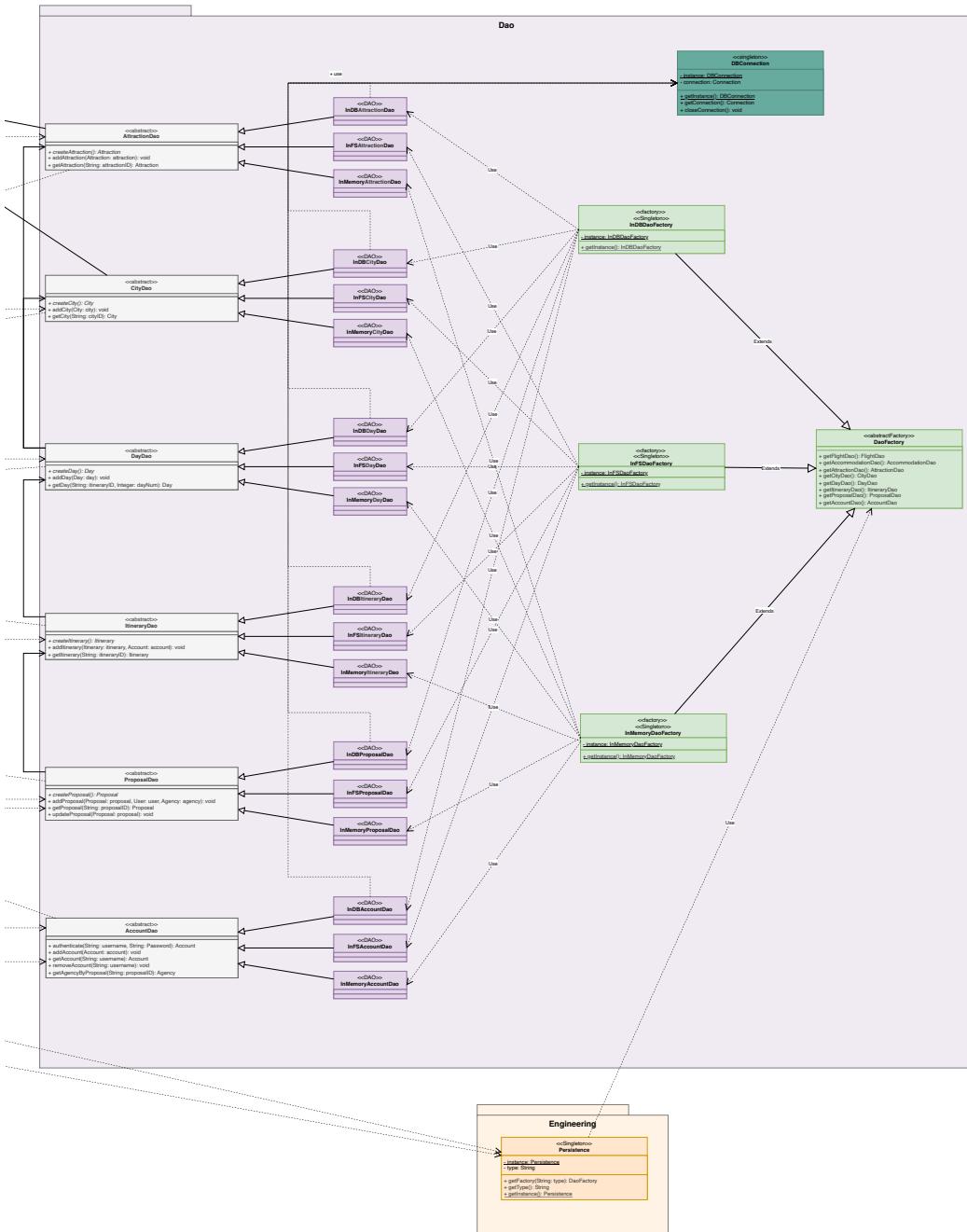
Application Controller:



### Model and Engineering:

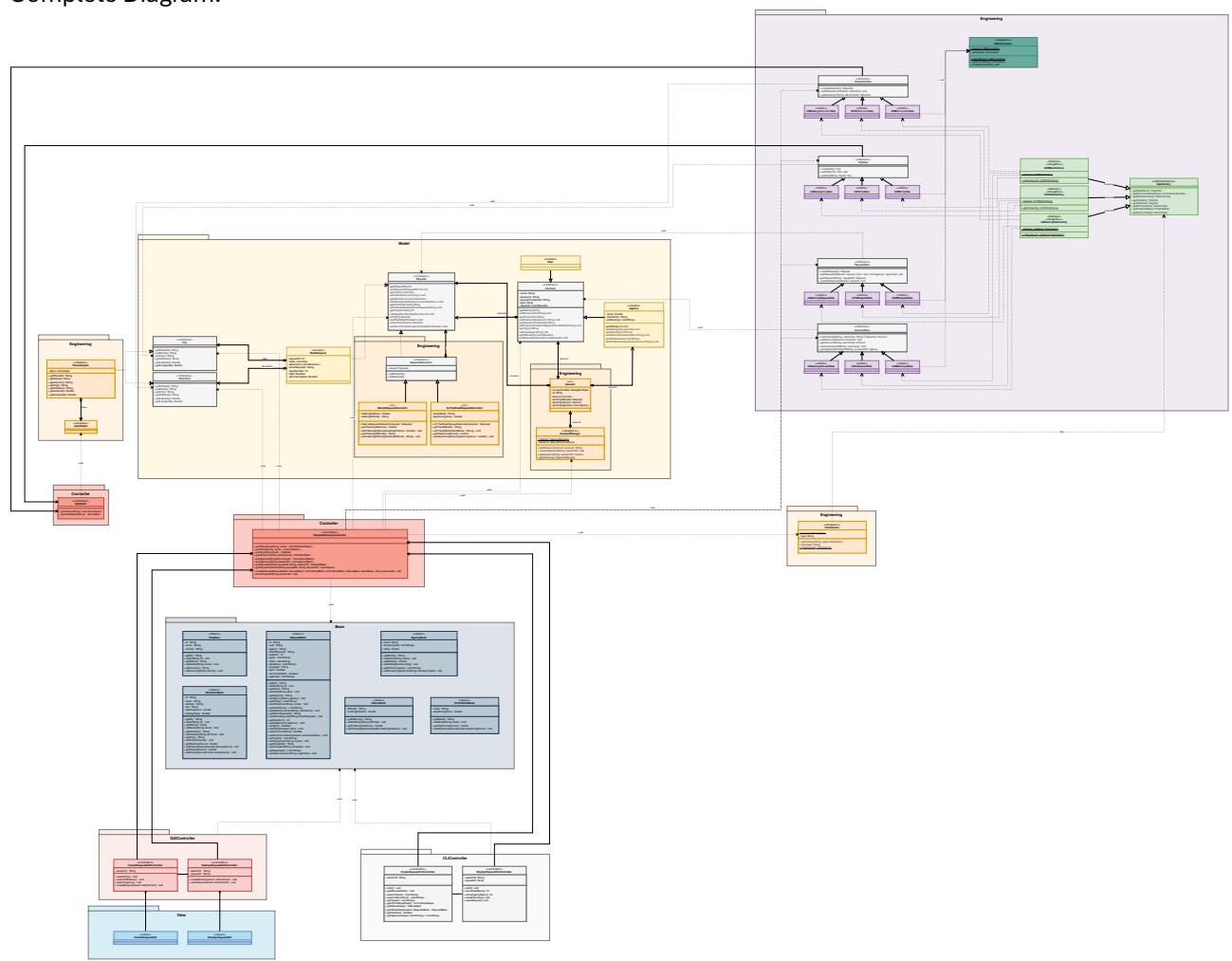
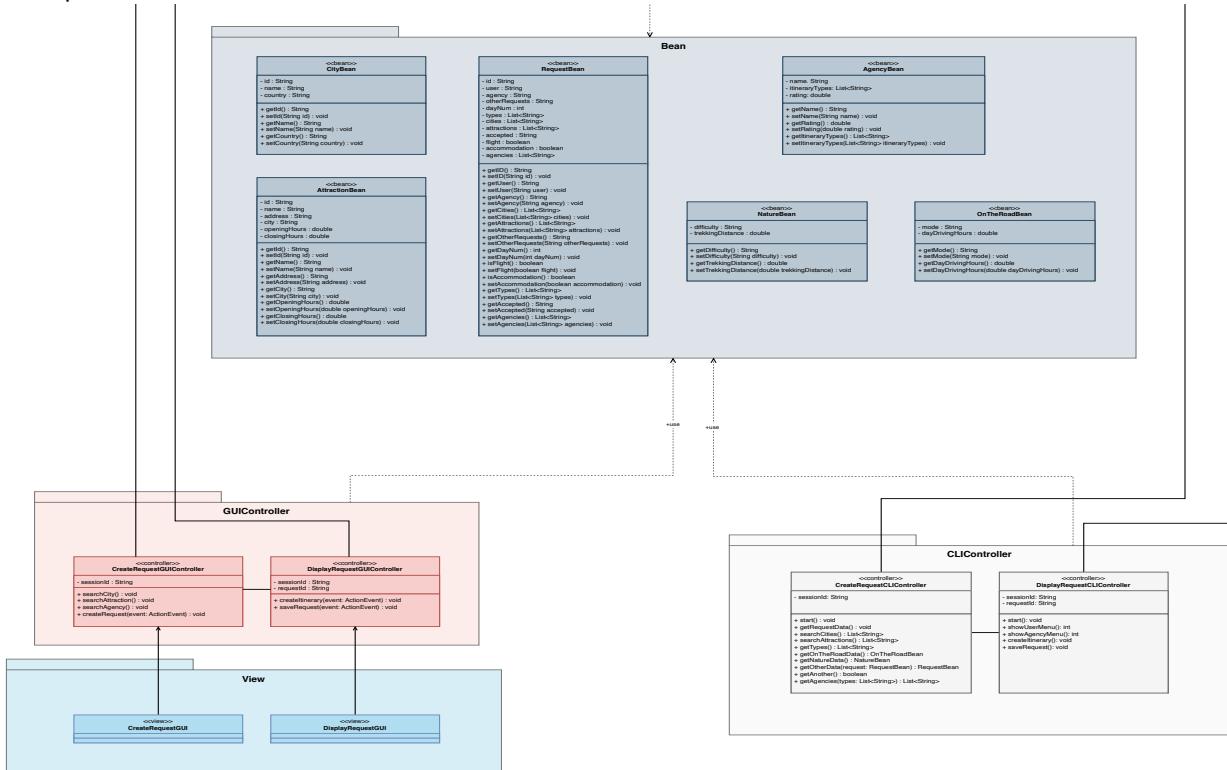


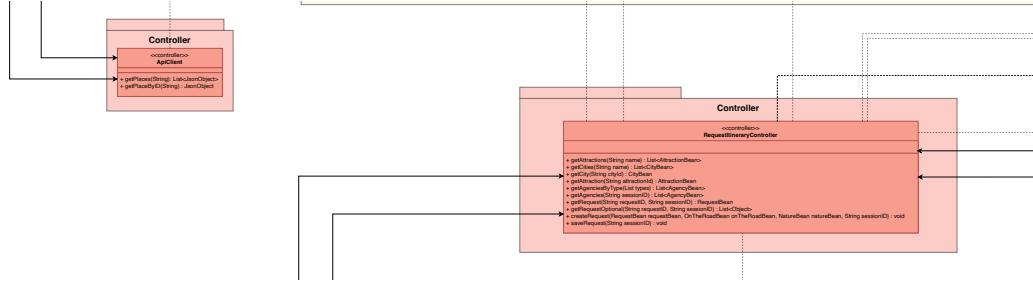
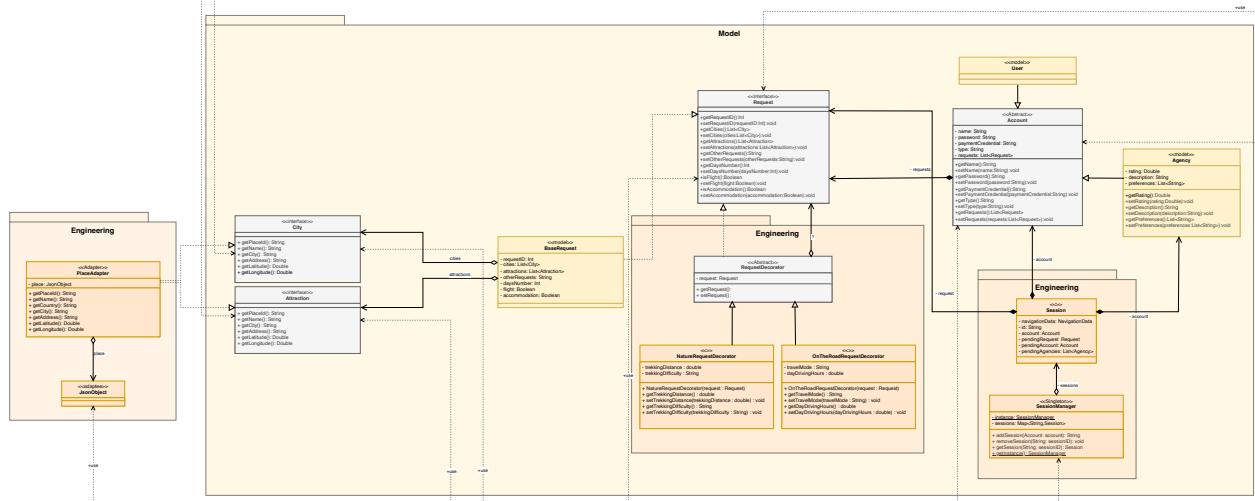
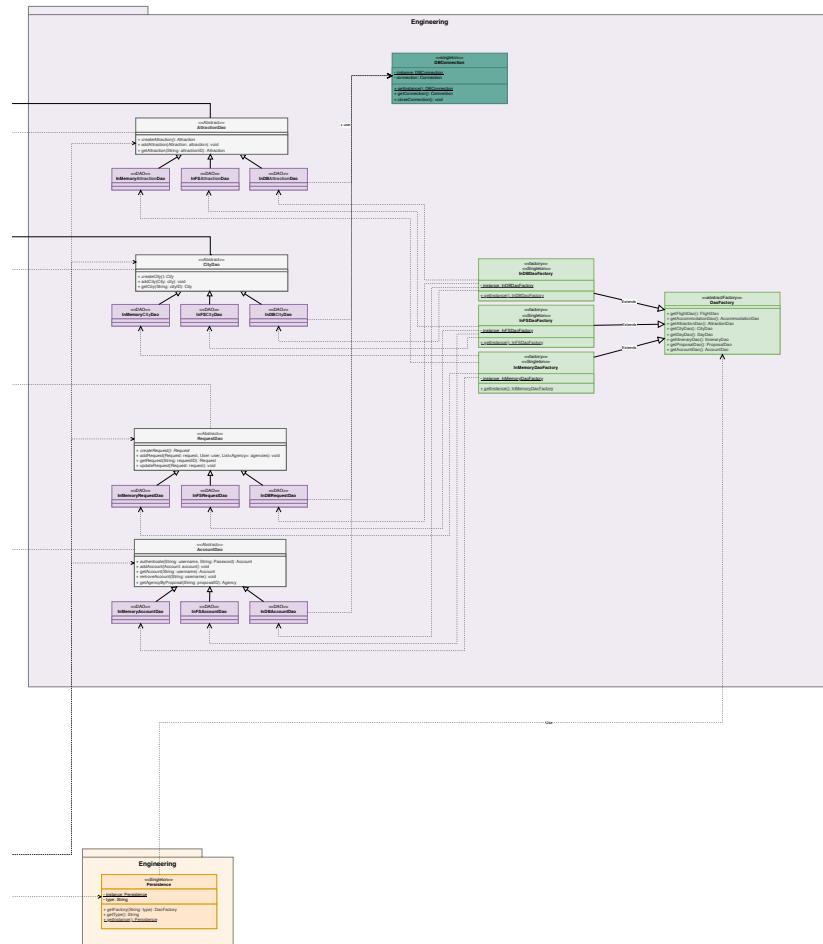
### DAO:



**Simone Tummolo | 0309116:**

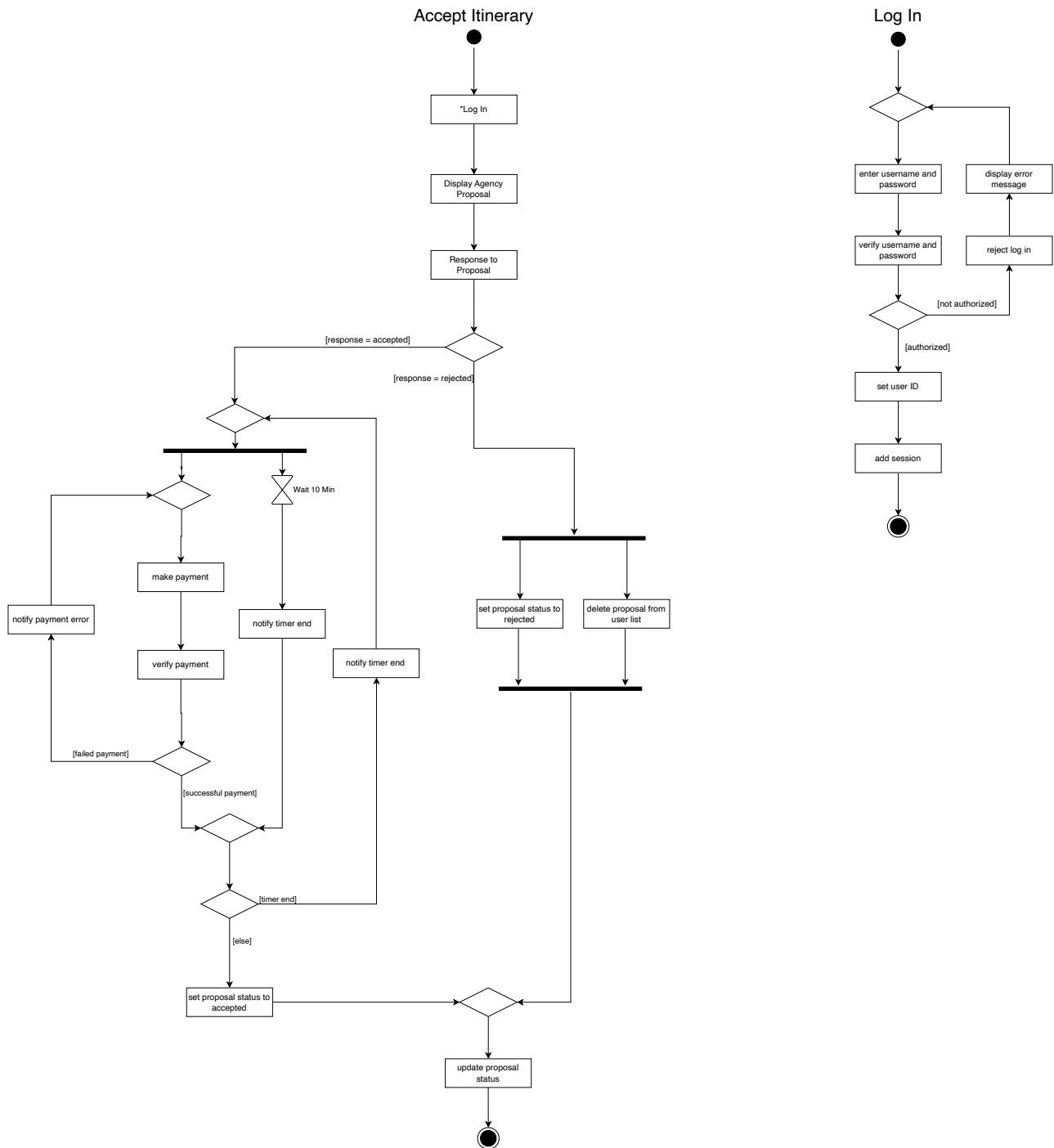
Complete Diagram:

**Graphic Controller and Bean:**

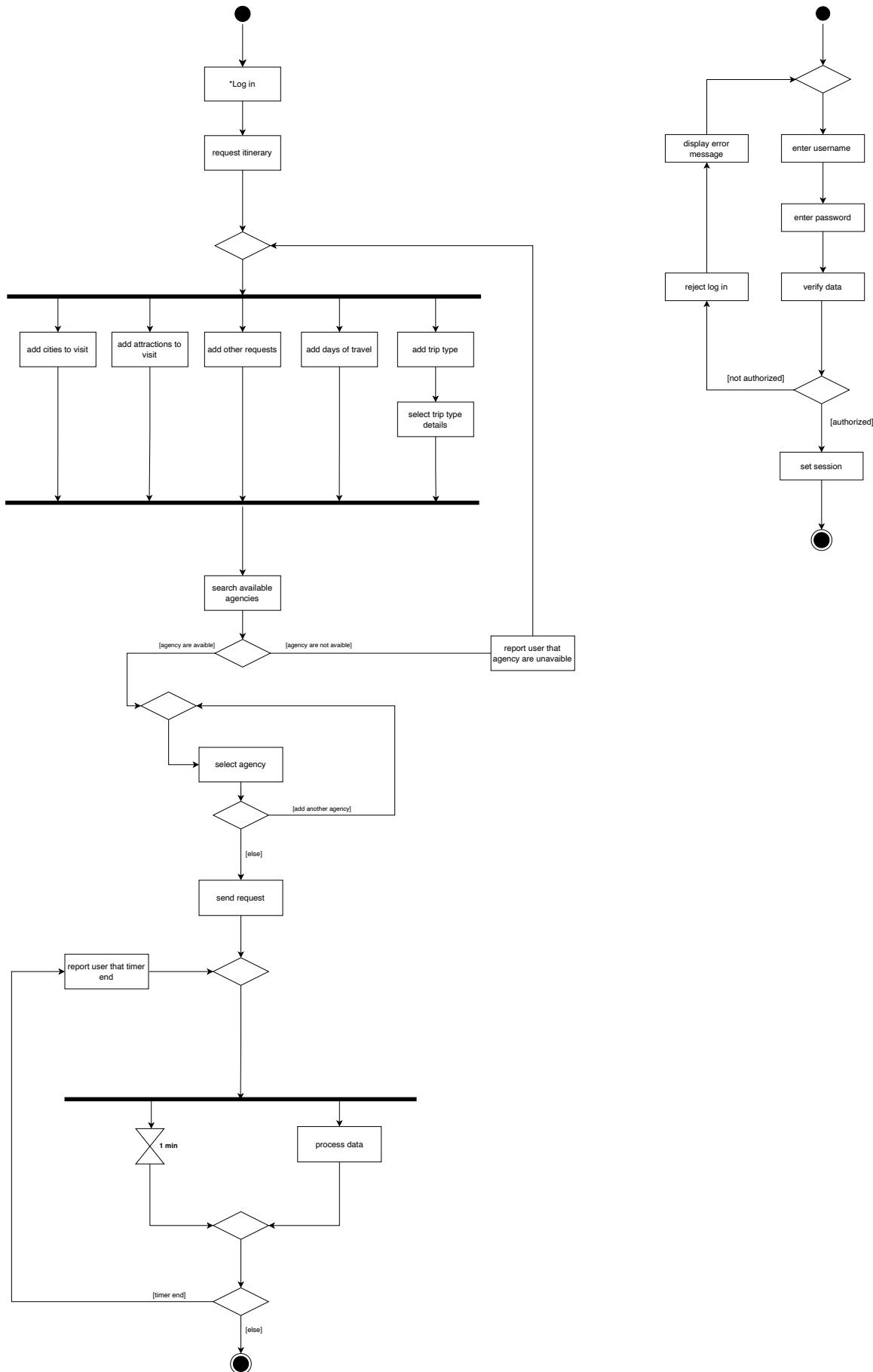
**Application Controller:****Model and Engineering:****DAO:**

### 3.2. Activity diagram

**Teo Miozzi | 0309010:**



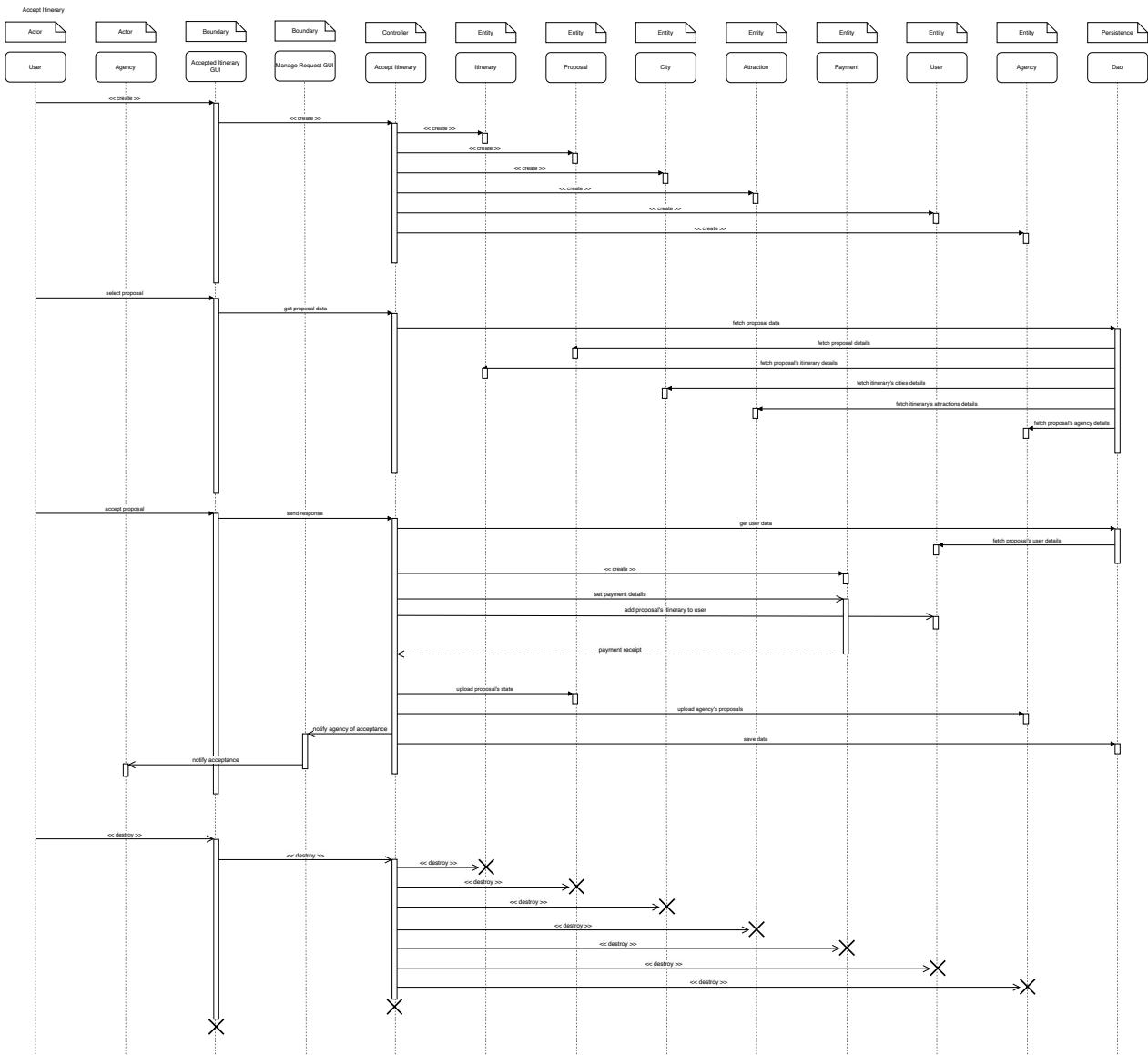
**Simone Tummolo | 0309116:**  
Request Itinerary



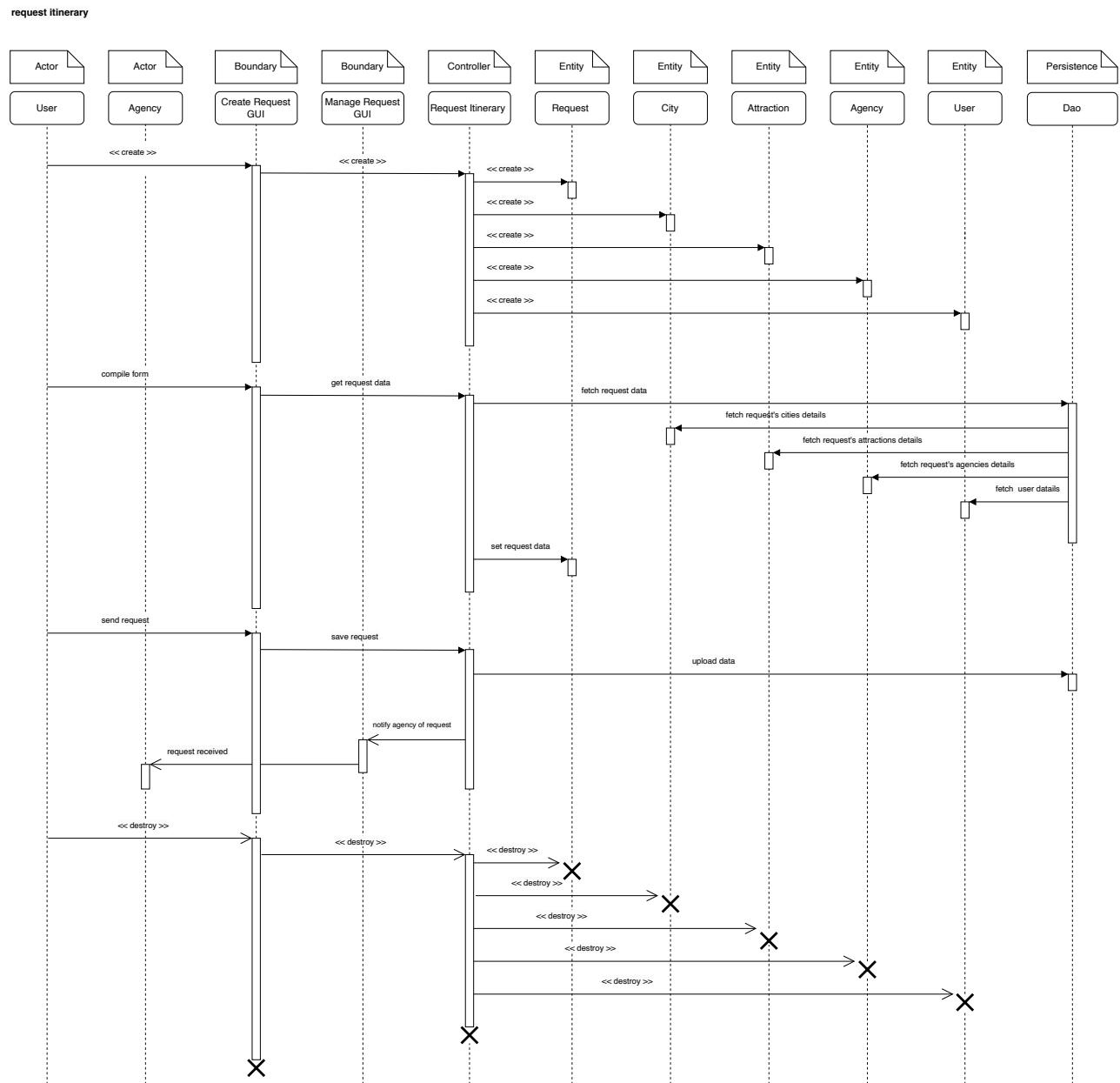
### 3.3. Sequence diagram

**Teo Miozzi | 0309010:**

Accept Itinerary Use Case:



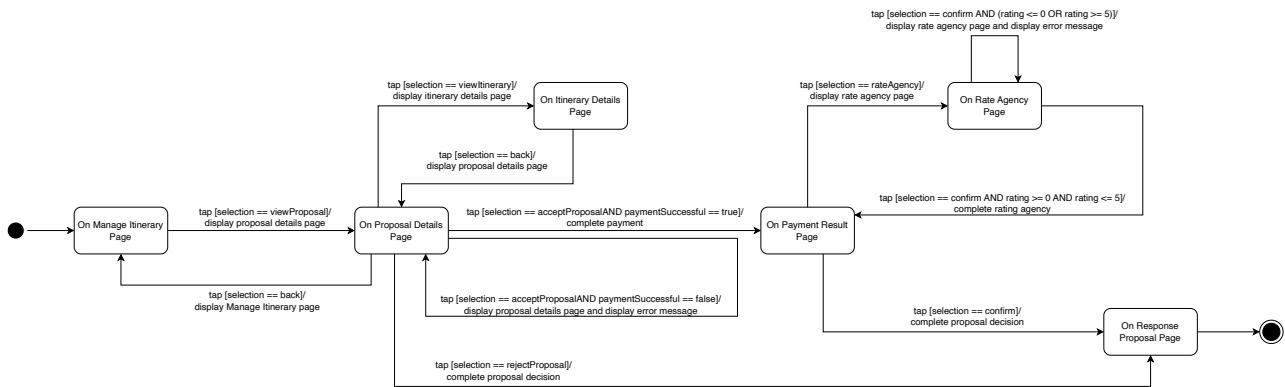
## **Simone Tummolo | 0309116:**



### 3.4. State diagram

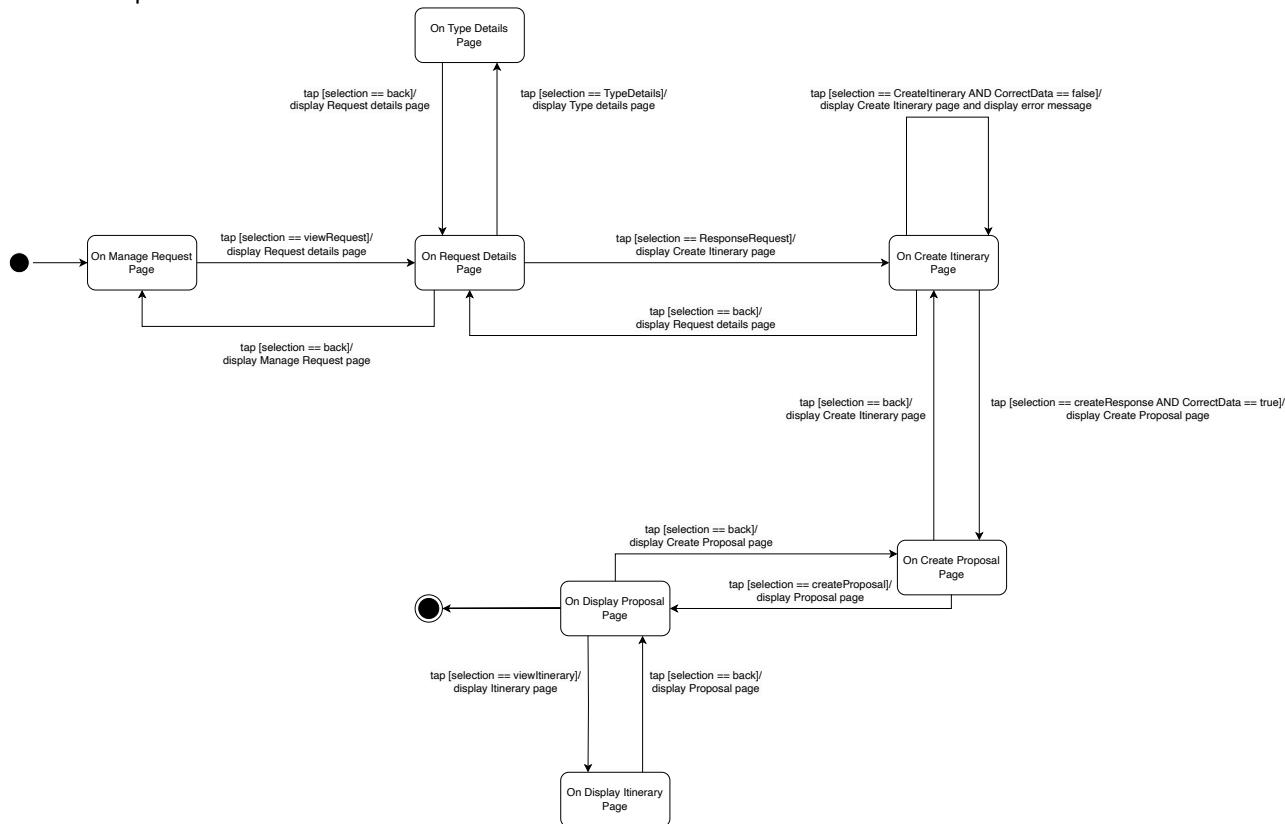
**Teo Miozzi | 0309010:**

Accept Itinerary Use Case:



**Simone Tummolo | 0309116:**

Send Response Use Case:



## 4. Testing

The test classes we have analysed can be found at the following [link](#).

In particular, we tested some methods of classes:

- **Teo Miozzi | 0309010:**
  - ItineraryBean → setDuration(), setOutboundFlightDepartureTime() and setAccommodations().
  - ProposalBean → setPrice(), setAccepted() and setDescription().
  - AcceptItineraryController → sendResponse(), getProposalItinerary() and getAccountType.
  - ManageItineraryController → getUserProposals().
- **Simone Tummolo | 0309116:**
  - RequestBean → setAccepted(), setDayNum() and setOtherRequests().
  - LoginController → logIn(), signIn() and getUserType().
  - RequestItineraryController → createRequest(), getRequest() and saveRequest().

## 5. Code

The source code of the project is available on GitHub at the following link: [GitHub Repository](#).

The code quality analysis is available on SonarCloud at the following link: [SonarCloud Analysis](#).

## 6. Video

A video demonstration showing how the application works is available on YouTube at the following link: [YouTube video](#).