



Ingegneria del software e progettazione web
Progetto A.A. 2023/2024

ADVENTURES

0293919

Margherita di Santo

Table of contents

1. Introduction	3
<i>Aim of the documentation.....</i>	<i>3</i>
<i>Overview of the defined system</i>	<i>3</i>
<i>HW ad SW requirements</i>	<i>4</i>
<i>Related system, pros and cons.....</i>	<i>5</i>
2. User stories	6
3. Functional requirements.....	7
4. Use cases	8
<i>Diagram</i>	<i>8</i>
<i>Internal steps.....</i>	<i>9</i>
5. Storyboards	10
6. Class diagram	11
<i>BCE.....</i>	<i>11</i>
<i>MVC</i>	<i>12</i>
<i>Design pattern.....</i>	<i>13</i>
7. Activity diagram	14
8. Sequence diagram.....	15
9. State diagram.....	16
10. Testing.....	17
11. Code	18
12. Video	19
13. Sonar cloud.....	20

1. Introduction

Aim of the documentation

The aim of the documentation is to provide a full description of the software system. The system has been developed by following a well-defined approach, based on practices of software engineering, toward the satisfaction of the project goals.

Overview of the defined system

Adventures is a comprehensive application that enhances the connection between guides and travelers, by rethinking the organization and participation in adventure trips. This platform improves the accessibility and organization of travel experiences by streamlining the participation and management of adventure trips.

The application has been designed for two distinct user categories: Guides and Travelers. Below are the key features designed for each user type.

Features for Guides:

- **Creation of new trips:** Guides have the opportunity to effortlessly create and publish their available trips. They can specify essential details, such as date, time, departure location, destination, pricing, and itinerary.
- **Editing trip details:** Guides can modify the details of their trips at any time, thereby ensuring that the information remains current and accurate.
- **Viewing other guides' trips:** Guides can access and review trips offered by their colleagues; however, they are unable to participate in these trips. This functionality enables them to keep up with community offerings and activities.
- **Managing participation requests:** Guides receive requests for participation from travelers who are interested in joining their trips. Furthermore, they can decide whether to accept or decline these requests based on their availability and personal preferences.

Features for Travelers:

- **Viewing all available trips:** Travelers can search and view all available trips in the app, using criteria such as country and category.

- **Request for trip participation:** After the identification of a trip of interest, travelers have the opportunity to submit a request for participation to the guide who has organized the trip.
- **Request for Quote with Additional Insurance:** Travelers can also request a personalized quote for the trip. Indeed, the application also provides the option to include an additional insurance coverage in order to strengthen their security of trip.

In the application, users can interact with the system through two different interfaces:

- **Graphical User Interface (GUI):** Users can access the functionality of the system through a minimalist GUI, which provides a clear overview of available trips and applications. Users can search, view trip details, and manage their requests intuitively.
- **Command Line Interface (CLI):** The system also offers a well-structured and user-friendly command-line interface. This interface allows users to perform the same essential tasks using text commands.

Discovering new destinations and organizing trips becomes effortless and enjoyable, With Adventures, a platform tailored to meet the needs of both guides and travelers.

HW ad SW requirements

Software and hardware requirements:

- RAM: 2GB of free RAM
- CPU: any modern CPU
- Disk space: 3.5GB
- Monitor resolution: 1315x810
- Operating system: Microsoft 8 or later, macOS 10.14 or later, any Linux distributions that supports Gnome, KDE or Unity DE

Related system, pros and cons

Adventures may be compared to other travel management platforms, such as GetYourGuide. These systems provide similar features, including the opportunity for users to create and participate in several travel experiences.

Pros of *Adventures*:

- **Customization of Trips:** Guides have the ability to design personalized trips, by adjusting details in order to meet individual preferences and requirements.
- **Direct Interaction between Guides and Travelers:** Guides can manage participation requests directly, therefore increasing communication and overall user satisfaction.

While some cons are:

- **Single-User Limitation:** The system currently supports only single-user interactions, however it does not permit concurrent usage by multiple users, thereby limiting its scalability for larger communities.
- **Absence of a Review Format:** The system lacks a review system with descriptive feedback, which might help users in making more informed decisions.

2. User stories

1. As a traveler, I want to review my travel guide by selecting a grade from 1 to 5, so that I can let the guide know how my experience was.
2. As a traveler, I want to request a quote, so that I can decide whether to go on the trip based on the price.
3. As a guide, I want to see trip organized by other guide, so that I can organize trips that are not already organized.

Note: the first user story is NOT implemented.

3. Functional requirements

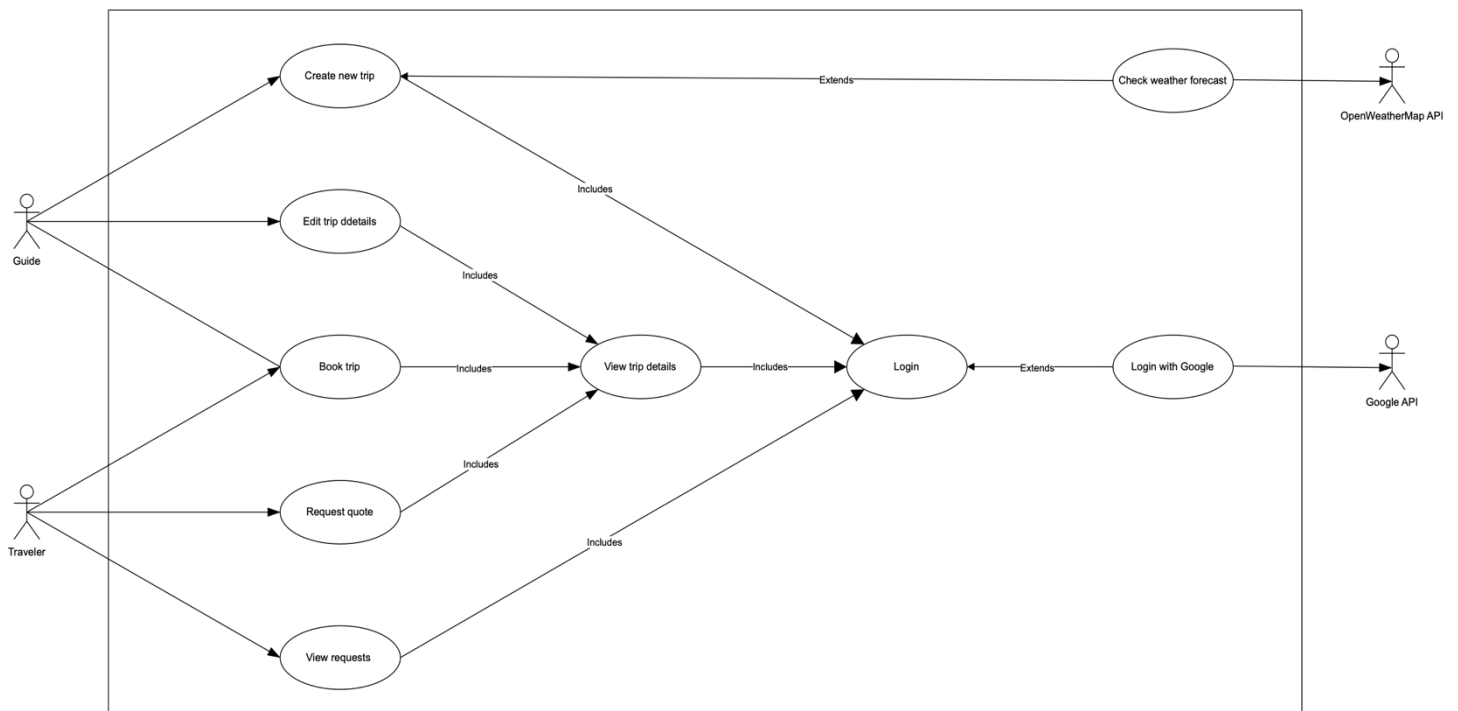
1. The system shall provide the form to request a quote with the possibility to select the type of insurance*.
2. The system shall provide a blank form to enter the information** about a new trip.
3. The system shall show to the guide the requests of the travelers with the traveler's name and surname.

*type of insurance = healthcare insurance, luggage insurance, cancellation insurance.

**information = name, departure city, outbound date, return date, price, category (safari, family, food tasting, sport, fun, relax).

4. Use cases

Diagram



Note: Use cases *Login with Google*, *Check weather forecast* and *Edit trip details* are NOT implemented.

Internal steps

Use case: *Add trip (Guide)*

1. The system authenticates the guide via the use case Login
2. The guide requests to add a trip.
3. The system provides a blank form for the insertion of trip information.
4. The guide enters trip information.
5. The system provides a blank form for the insertion of the stops.
6. The guide enters the stops.
7. The guide selects to save the trip.
8. The system fetches the information.
9. The system saves the trip.

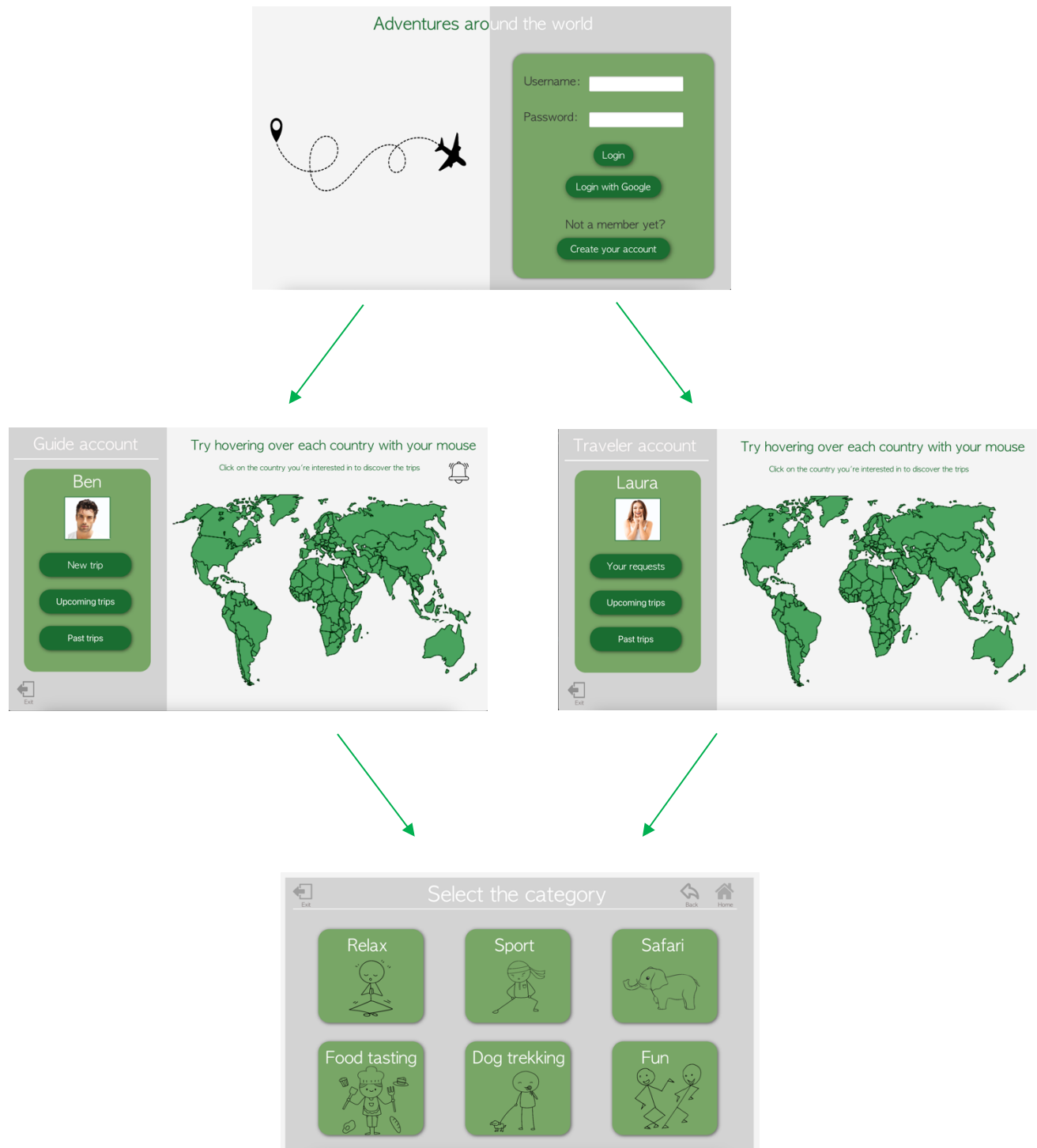
Extensions:

- 1.a *The user does not exist*: the system shows the error message “Wrong username or password”.
- 7.a *The form is not complete*: the system shows the error message “Please complete the field”.
- 8.a *The trip already exists*: the system notifies the guide and terminates the use case.

Note: these internal steps do not correspond to the final version of the application. Steps 3 and 5 are written individually just to meet the requirements of the deliverable, but in the application, they are implemented on the same page. In addition, the controls described in "extensions" have not been implemented in the final version.

5. Storyboards

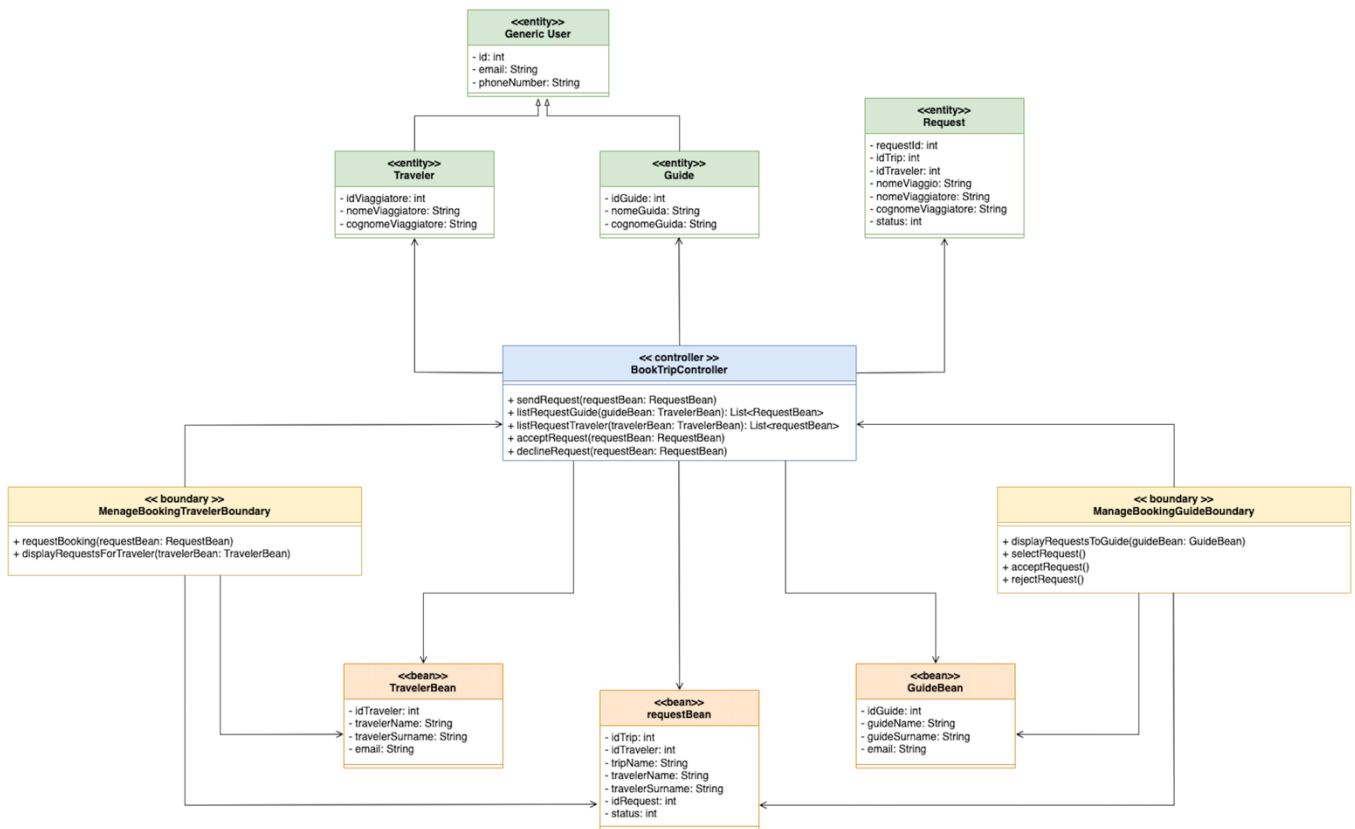
Below are illustrated a series of storyboards that correspond respectively to the login screen, the two home screens (depending on the type of user) and the screen for choosing the category of the trip.



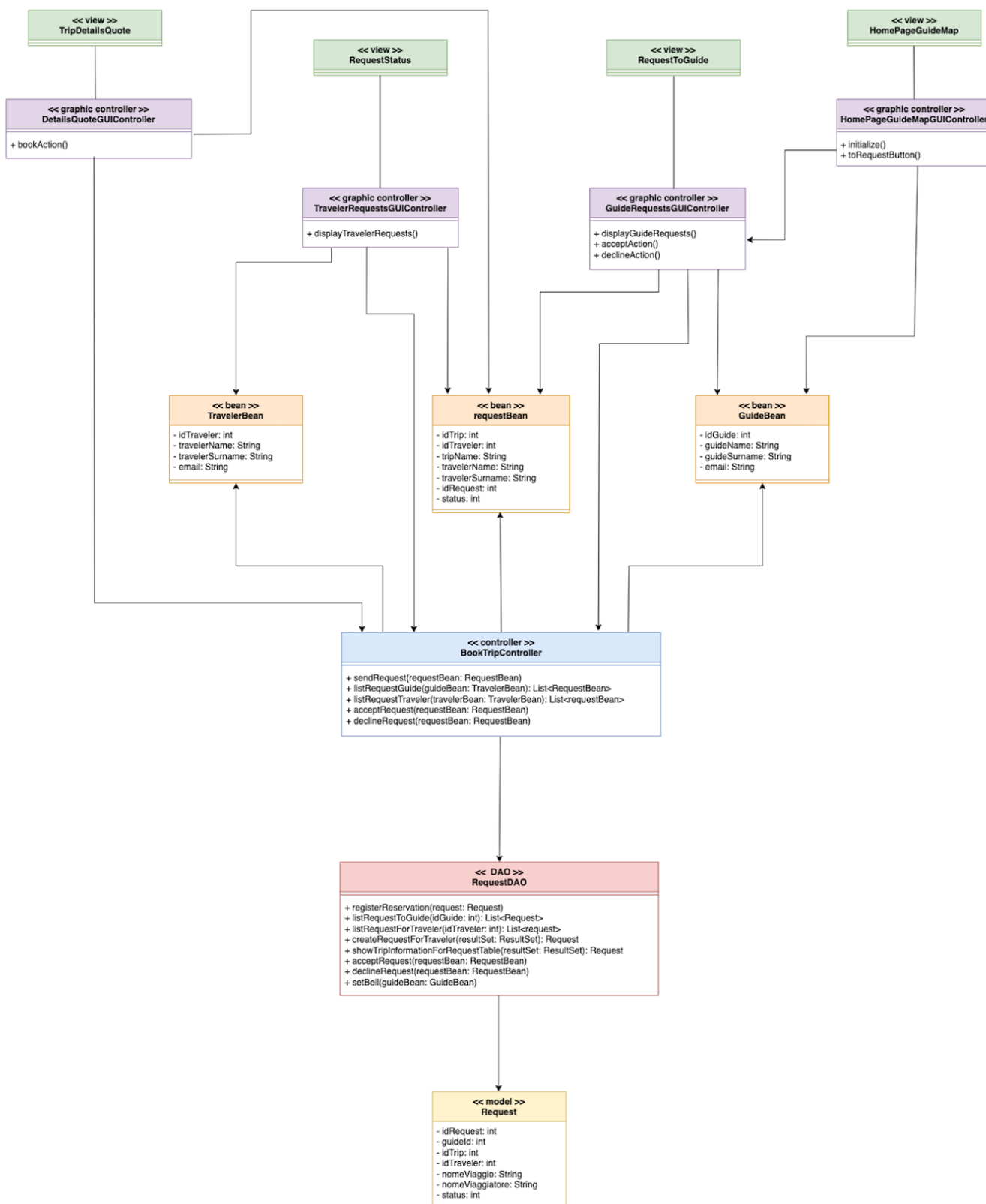
Note: Some features have not been implemented, thus some graphics have been added to give you a preview of how they will be available in the future.

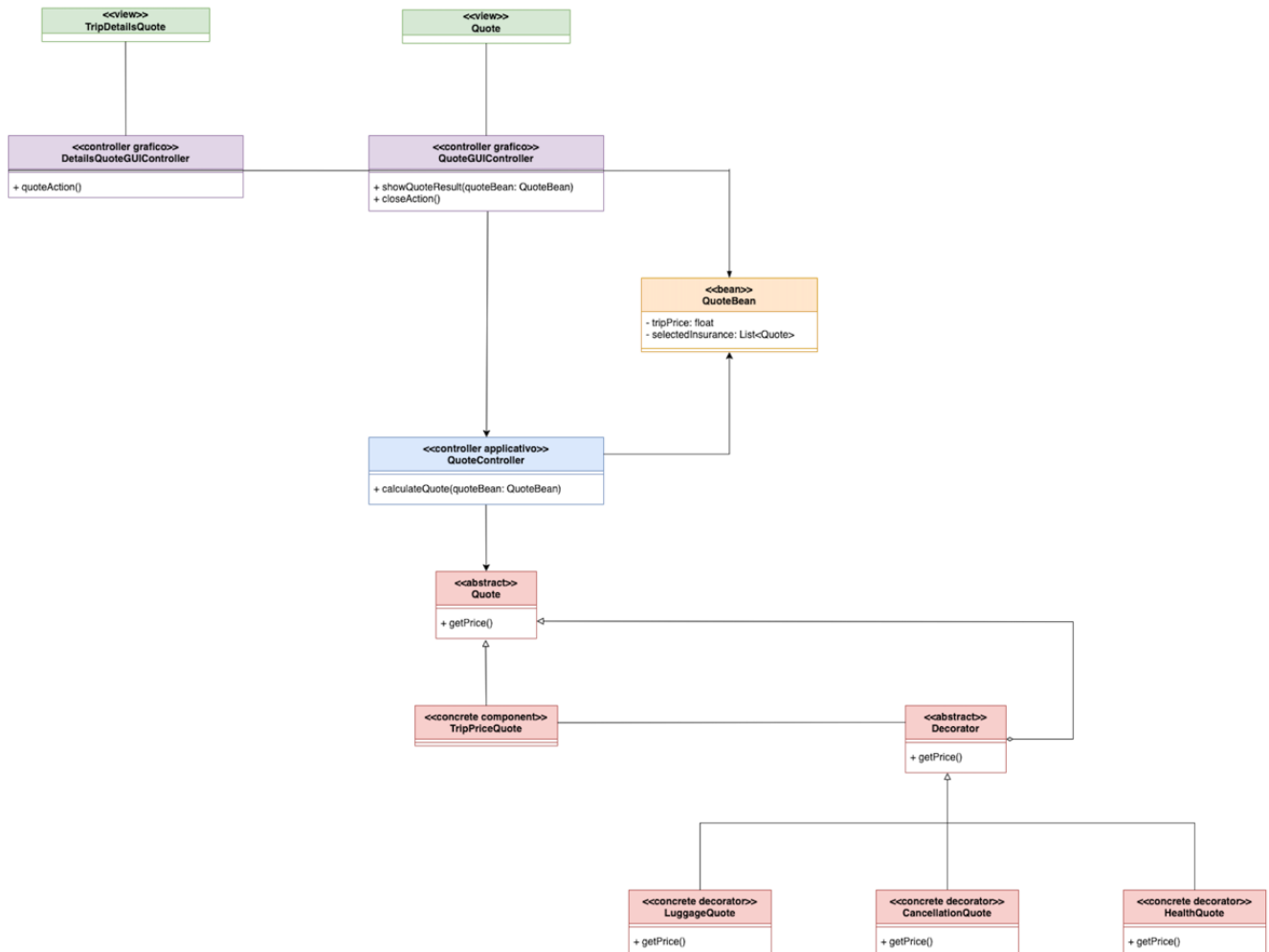
6. Class diagram

BCE - Use case: Book Trip (Traveler)



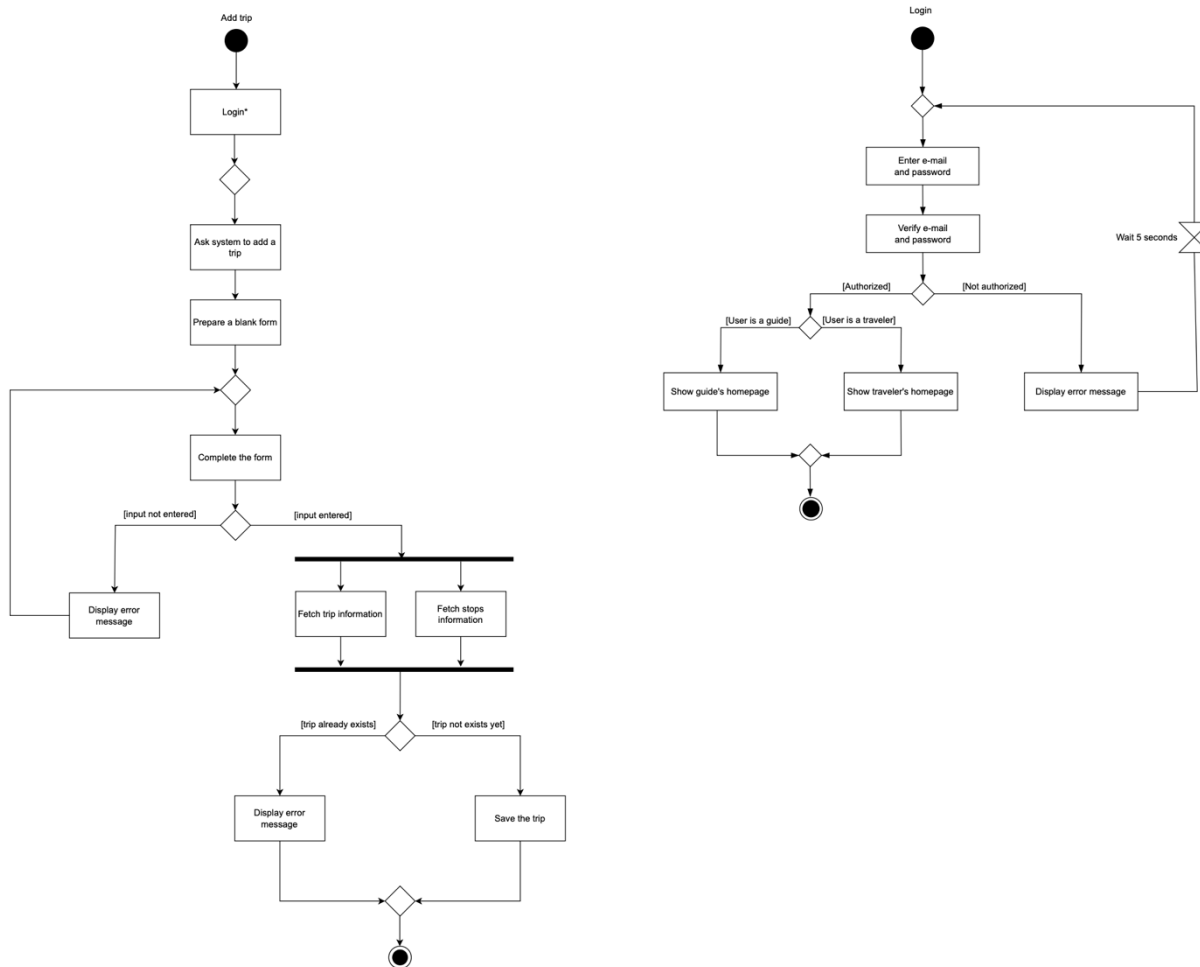
MVC - Use case: Book Trip (Traveler)



Design pattern - Use Case: *Request Quote (Traveler)*

7. Activity diagram

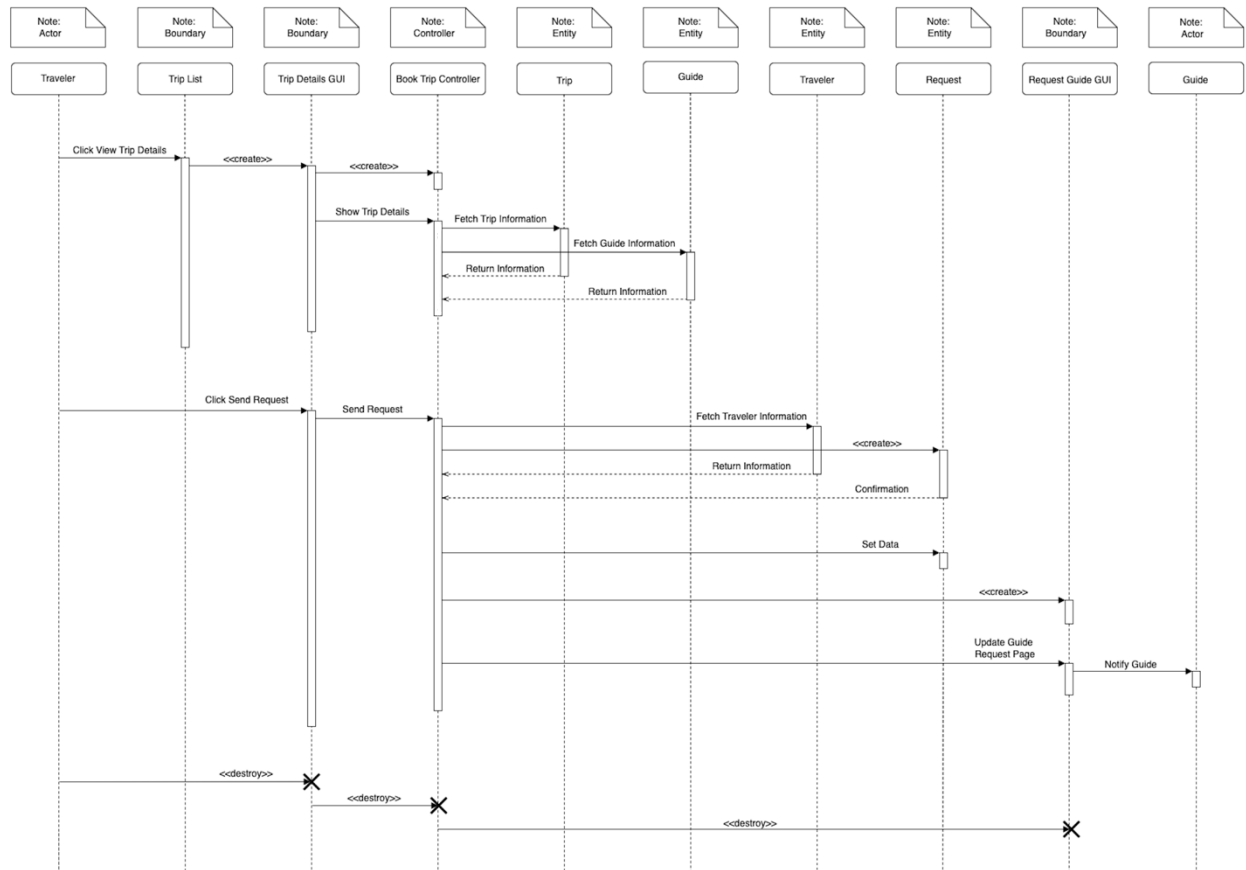
Use Case: *Add trip (Guide) + Login*



Note: these Activity Diagrams do not correspond at all to the final version of the application. Many of the actions are written just to meet the requirements of the deliverable and are not implemented.

8. Sequence diagram

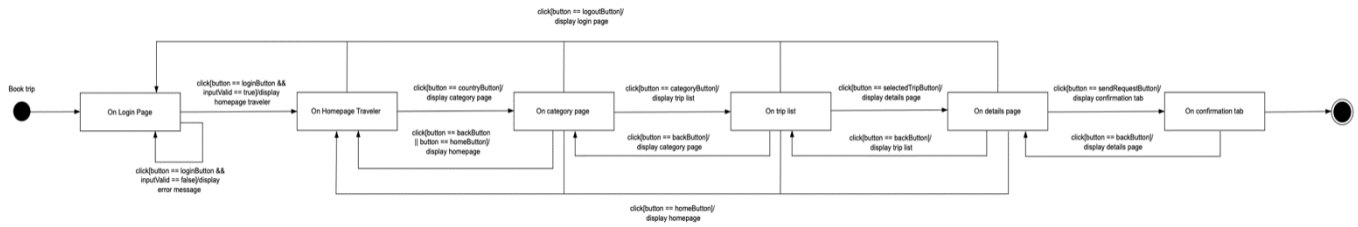
Use Case: *Book trip (Traveler)*



NOTE: in the application all the asynchronous calls are synchronous and the <<destroys>> are not implemented since the Garbage Collector oversees the deallocation in Java.

9. State diagram

Use Case: *Book trip (Traveler)*



Note: this state diagram does not correspond to the final version of the application.

10. Testing

Three tests have been implemented:

- *LoginBeanTest*: it checks that the constructor of the LoginBean class raises an exception of type EmailFormatException when an email address with an invalid format is provided.
- *NewTripTest*: it verifies that an exception is raised when a trip is created with an arrival date earlier than the current date.
- *NumberOfTripTest*: it verifies that after adding a new trip, by calling a method that returns the number of trips, it is incremented by 1.

11.Code

<https://github.com/MargheritaD/AdventuresProject>

12. Video

<https://github.com/MargheritaD/Deliverables/blob/main/videoPresentazioneAdventures.mpeg>

13.Sonar cloud

https://sonarcloud.io/project/overview?id=MargheritaD_AdventuresProject