Project Requirements Document

1. Introduction

The purpose of this software project is to develop a super app for travel that provides users with a comprehensive platform for planning and managing all aspects of their travel experience and budget. The app aims to simplify the travel planning process by integrating various services and information into a single, user-friendly platform.

1.1 Purpose of System

The purpose of the super app for travel is to provide users with a one-stop-shop for all their travel needs. The app will enable users to search for and book flights, hotels, and other travel services, as well as provide them with travel information and recommendations. The ultimate goal of the app is to make travel planning easier, more convenient, and more enjoyable for users.

1.2 Scope of System

The super app for travel will include the following features and services:

- Flight booking: This mini-app would allow users to search for and book flights, utilizing the Booking and Destination objects.
- Hotel booking: This mini-app would allow users to search for and book hotels, utilizing the Booking and Destination objects.
- Budget planner: This mini-app would help users plan and manage their travel budget, utilizing the Booking and Payment objects.

2. Actors and Goals

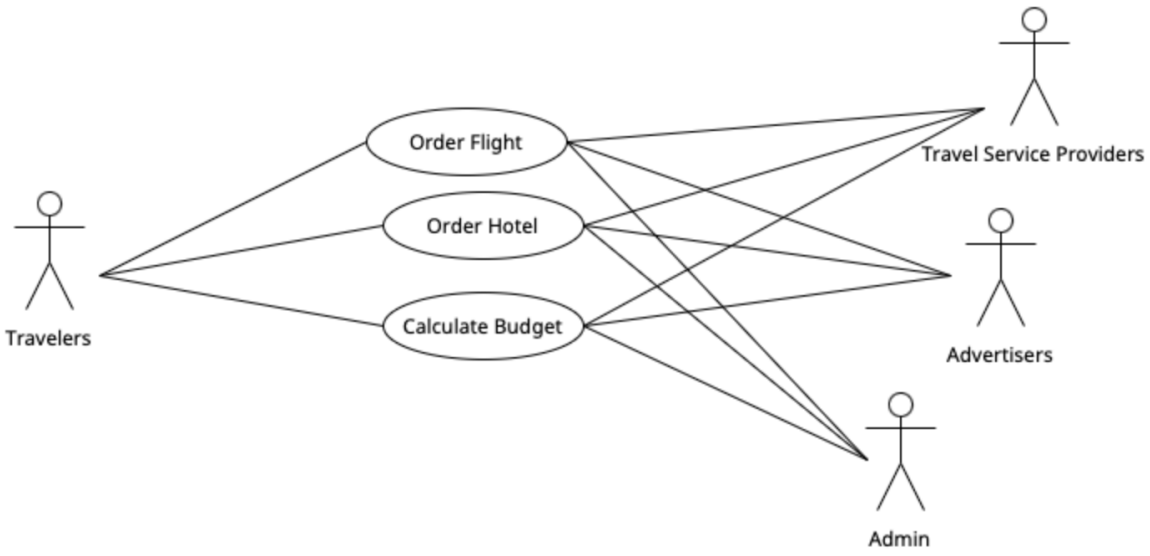
- Travelers: The primary actor of the system, travelers will use the app to search and book flights, hotels and manage their budget.
- Travel service providers: These actors will provide the travel services that are available on the app, such as airlines, hotels. Their goal is to reach a wider audience of potential customers through the app.
- Advertisers: These actors will use the app to advertise their products or services to users, based on their travel preferences and interests. Their goal is to increase brand awareness and drive sales.
- App administrators: These actors will be responsible for maintaining the app, managing user data and transactions, and ensuring the security and privacy of user information. Their goal is to provide a reliable and secure platform for users and travel service providers.

Each of these actors has a different set of goals and motivations for using the super app for travel, and the app's features and services will be designed to meet their respective needs.

3. Functional Requirements

- Order a flights from airlines suggestion list
- Order a hotel rooms from hotels suggestion list (booking , agoda)

3.1 Use Case Diagram



Use Case 1: Order Flight

Actors:

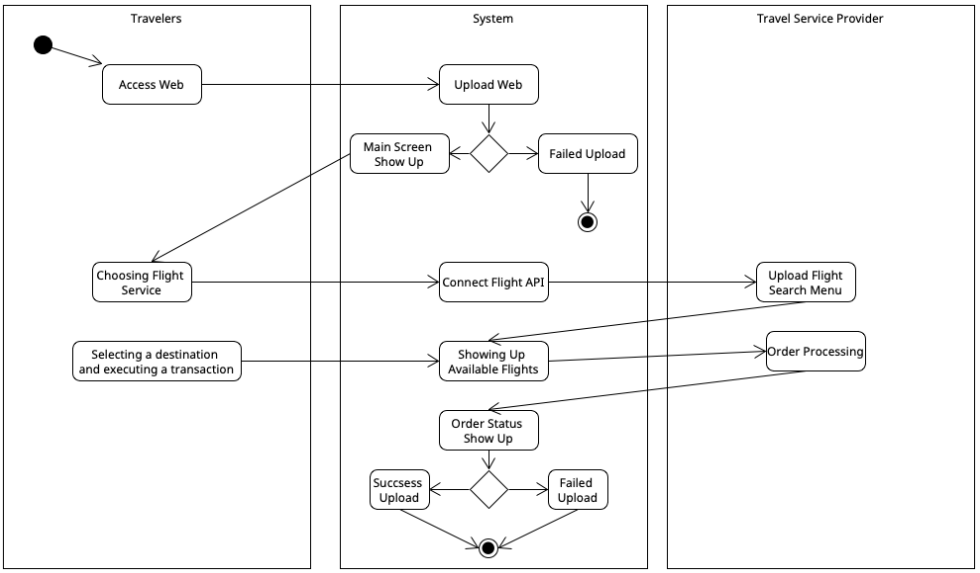
Travelers (Primary)

Travel Service Provider, Advertiser, App admin (Secondary)

Use Case Goal:

An option to search and order flights

Basic Flow :



Alternate Flow :

Travelers	System	Travel Service Provider
Access website		
	Open website	
Enter flight details		
	Check relevant flights	
		Flights details
	Show 0 results	

Use Case 2: Order Hotel

Actors:

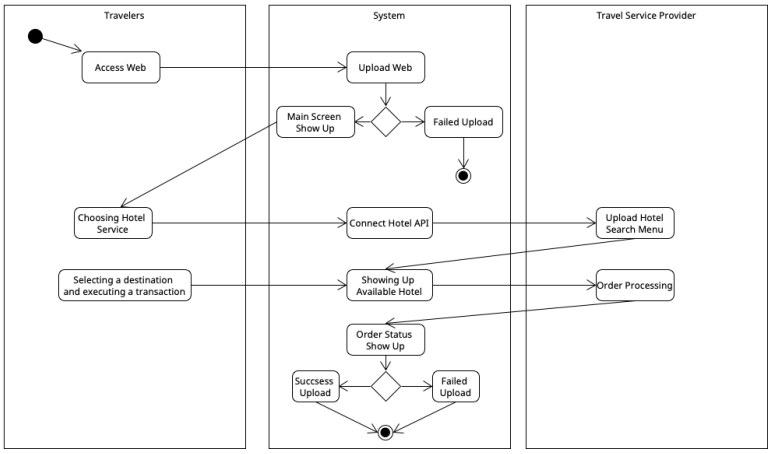
Travelers (Primary)

Travel Service Provider, Advertiser, App admin (Secondary)

Use Case Goal:

An option to search and order hotels

Basic Flow :



Alternate Flow :

Travelers	System	Travel Service Provider
Access website		
	Open website	
Enter hotels details		
	Check relevant hotels in destination	
		Hotels details
	Show 0 results	

Use Case 4: Calculate Budget

Actors:

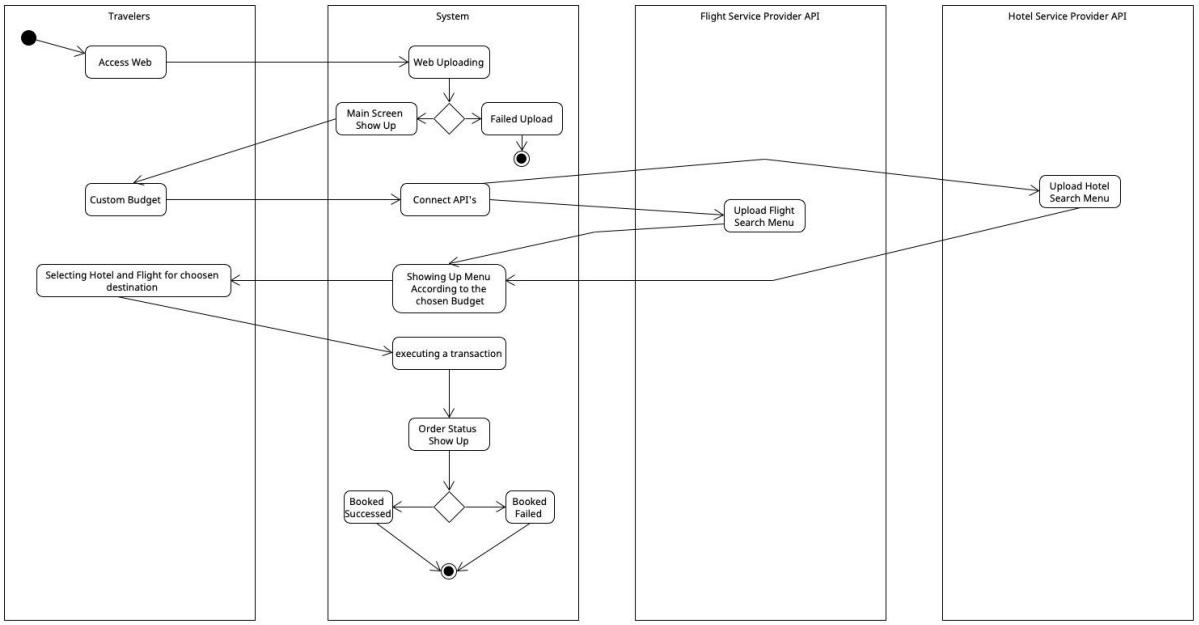
Travelers (Primary)

Travel Service Provider, Advertiser, App admin (Secondary)

Use Case Goal:

An option to set a budget for a trip

Basic Flow :



Alternate Flow :

Travelers	System	Travel Service Provider
Access website		
	Open website	
Enter flight details		
	Check relevant flights	
		Flights details
	Show 0 results	

4. Non Functional Requirements

Requirement Description	Requirement Type	Requirement Number
The app must have a simple and intuitive user interface to ensure ease of use for users with varying levels of technical proficiency (clear buttons actions , big font and more).	**Usability**	**1**
The app must provide clear and concise error messages to users when an error occurs	**Usability**	**2**
The app must be able to handle unexpected user inputs and edge cases without crashing or causing data loss.	**Reliability**	**3**
The searching and loaded of Flights and Hotels will not take more the 10 seconds	**Performance**	**4**
The app must have comprehensive documentation and online support available to assist users with any technical issues or questions they may have.	**Supportability**	**5**

Technologies

- Spring Boot
  The Spring Framework played a vital role in developing the backend infrastructure. It facilitated handling user requests, processing business logic, and integrating with various services such as flight and hotel providers, payment gateways, and customer management systems.
- MongoDB
  Was employed as the primary data storage solution for the flight and hotel booking system. It offers high scalability, flexibility, and performance, making it suitable for managing large volumes of data. MongoDB was used to store booking details, customer information, flight schedules, hotel availability, and other related data. Its document-oriented approach allowed for easy retrieval and manipulation of data in a flexible manner.
- ActiveMQ Artemis
  We integrated Artemis to facilitate asynchronous messaging and event-driven communication within our system. This allowed us to efficiently handle notifications, updates, and other real-time operations.
- JMS
  In our project, JMS, along with Artemis, enabled reliable and asynchronous communication between different modules, ensuring smooth data flow and event handling.
- Tomcat
  Apache Tomcat is a widely-used web server and servlet container for Java applications. We utilized Tomcat to deploy our backend services, providing the necessary environment for running and managing our Java-based server components.
- React
  Was utilized for building the user interface of the flight and hotel booking system. With React, developers were able to create reusable UI components and efficiently manage the application state. The dynamic nature of React made it easier to handle real-time updates, search functionality, and user interactions, providing a smooth and interactive experience for customers when searching and booking flights and hotels.
- Junit
  We employed JUnit to write and execute automated tests for our system's backend components, ensuring the correctness and reliability of critical functionalities.
- Postman
  Postman is a collaboration platform used for testing and documenting APIs. In the flight and hotel booking system, Postman was employed during the

development phase to test and validate the API endpoints responsible for handling flight and hotel bookings. It allowed us to send requests, examine responses, and ensure the smooth functioning of the API services.

- Spring Tool Suite
  We used Spring Tool Suite to write, debug, and manage the source code of our system, facilitating efficient development and collaboration among team members.

- Git
  Git is a distributed version control system that allows multiple developers to collaborate on a project efficiently. It is commonly used in software development to track changes made to source code and manage different versions of the codebase.

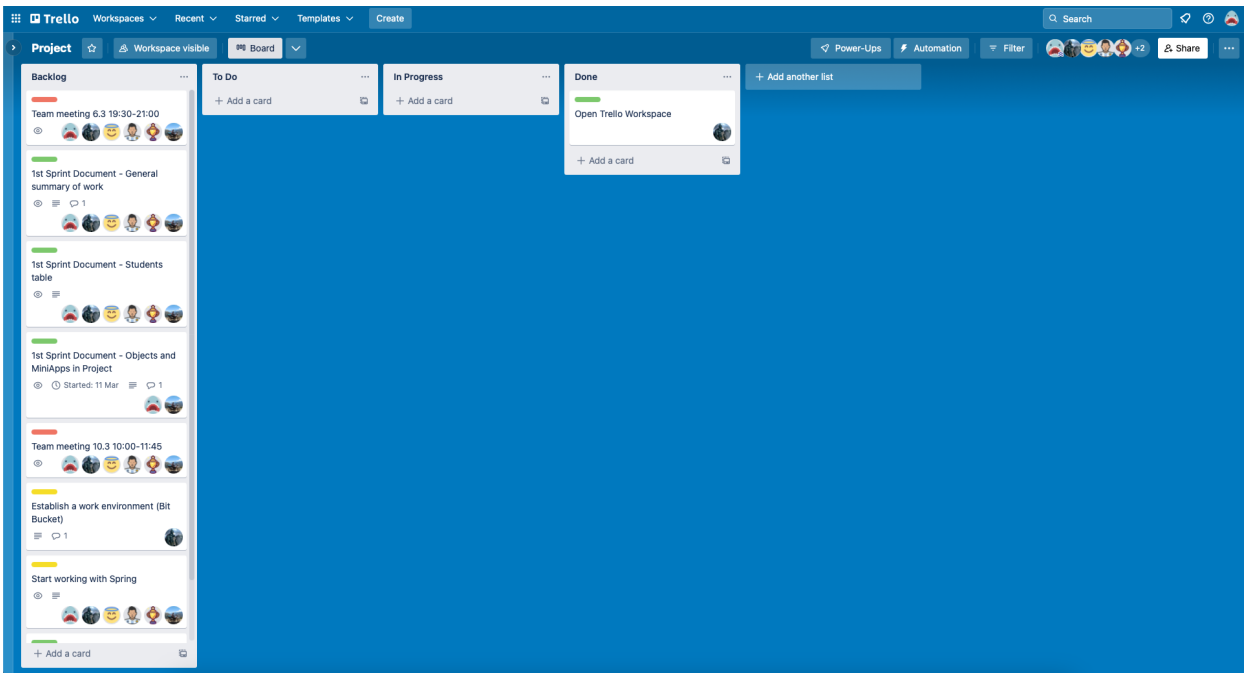
- RapidAPI
  RapidAPI is a platform that allows developers to discover, use, and manage APIs (Application Programming Interfaces) from various providers.
  We used the following API: Tripadvisor, Booking, ExchangeRate API

- Axios
  Axios is a JavaScript library that simplifies sending HTTP requests from a browser. We employed Axios to facilitate seamless communication between the frontend and backend of our system, enabling data retrieval and updating.

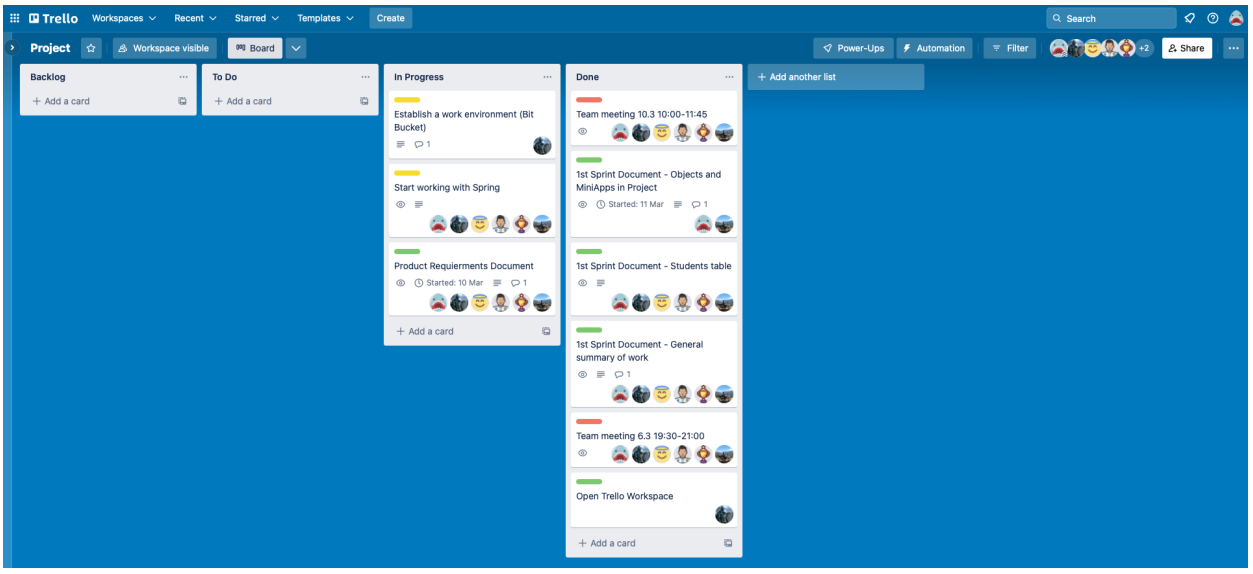
Kanban Boards

1st Sprint

- Beginning of this sprint - 28/02/2023

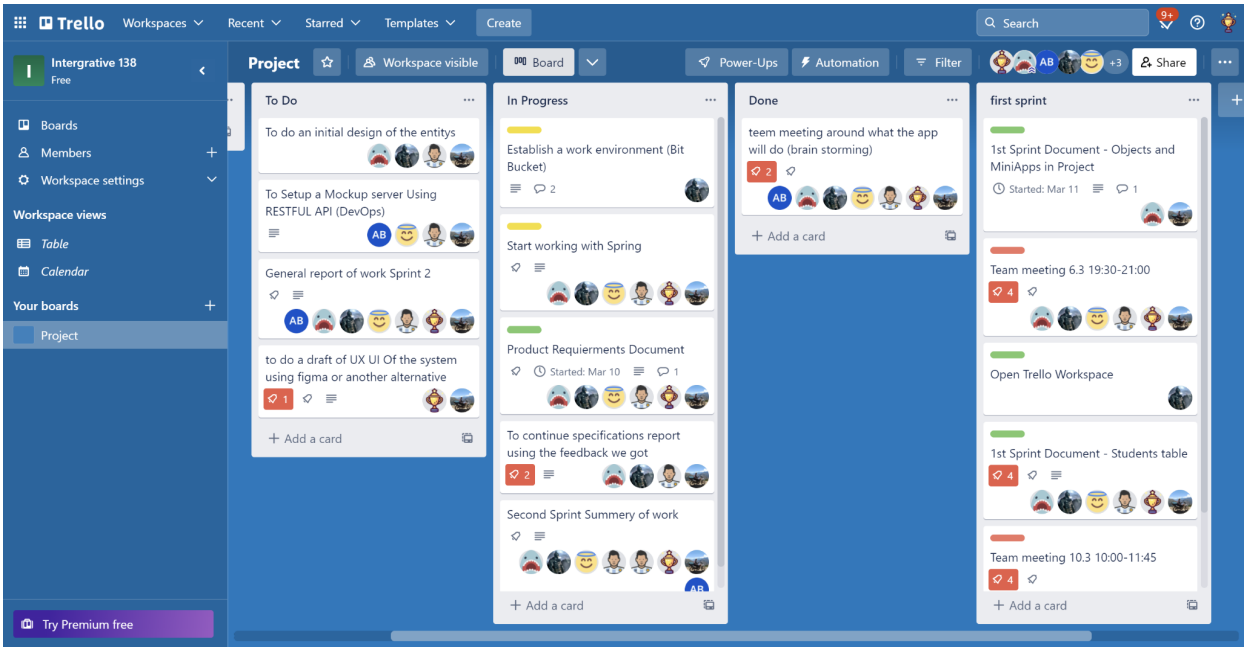


- End of this sprint - 14/03/2023

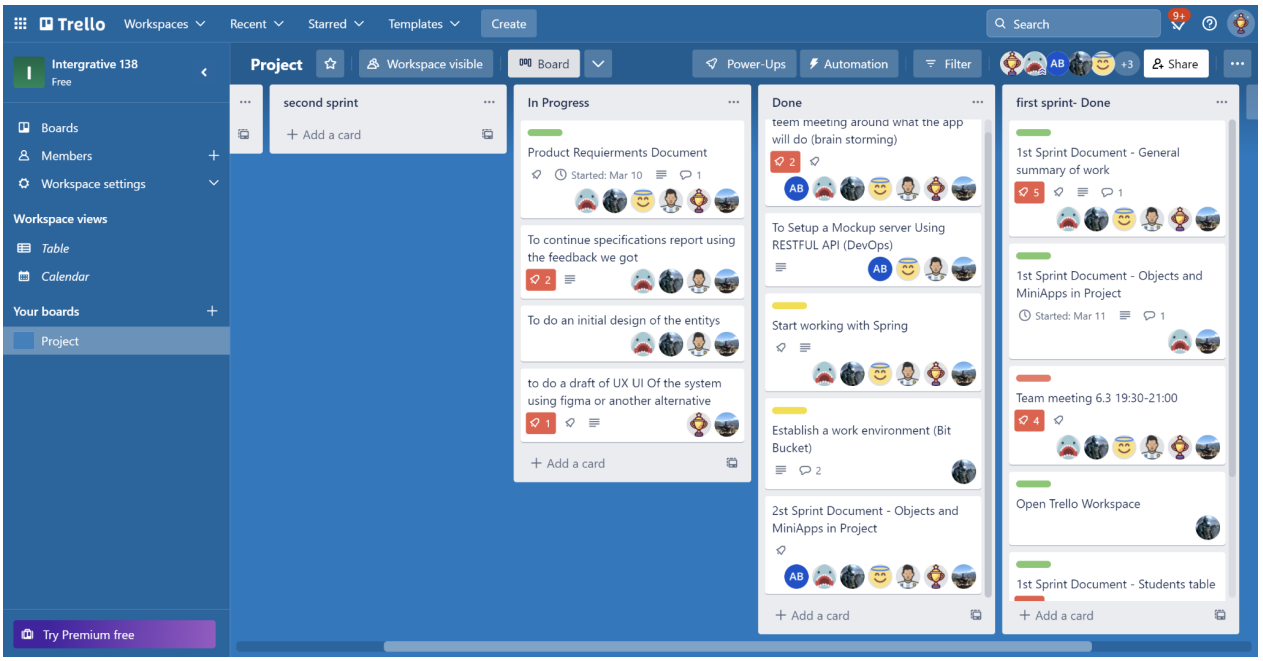


2nd Sprint

- Beginning of this sprint - 14/03/2023

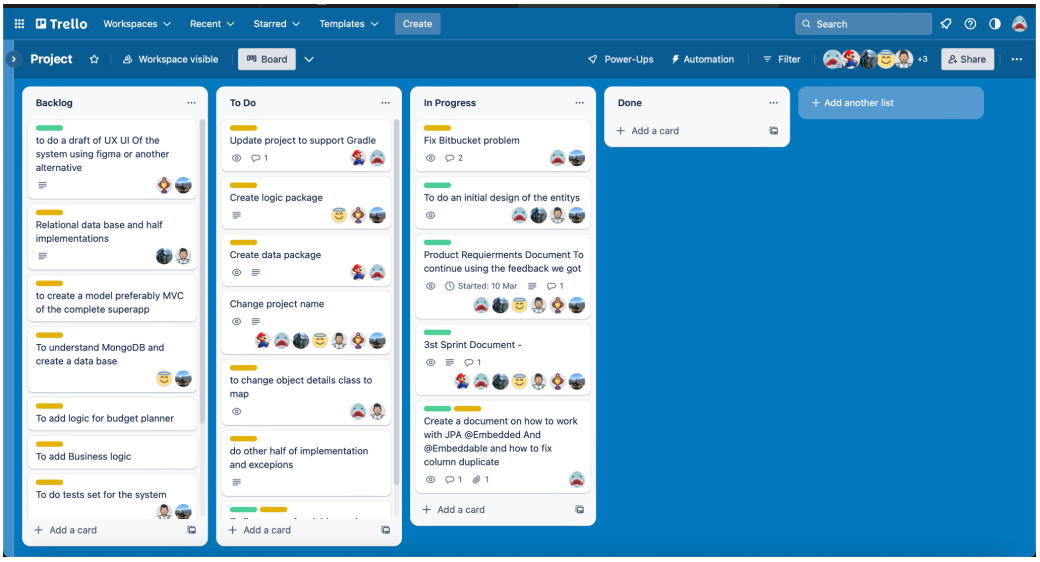


- End of this sprint - 28/03/2023

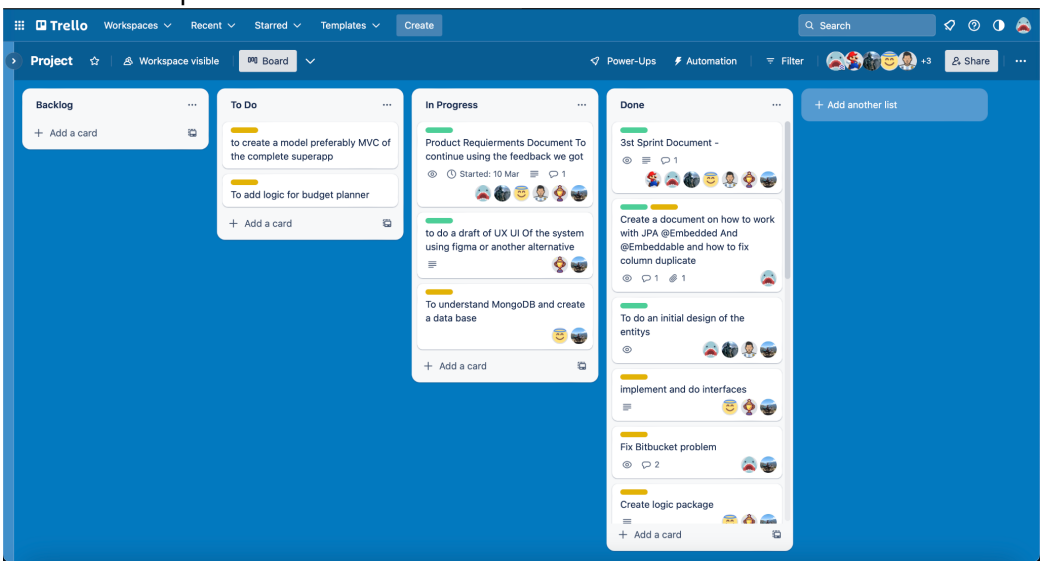


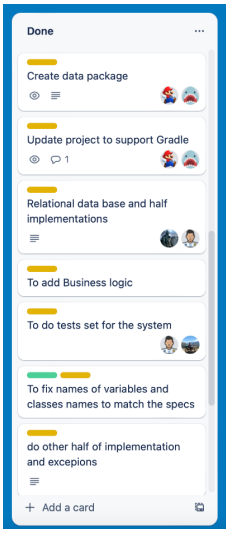
3rd Sprint

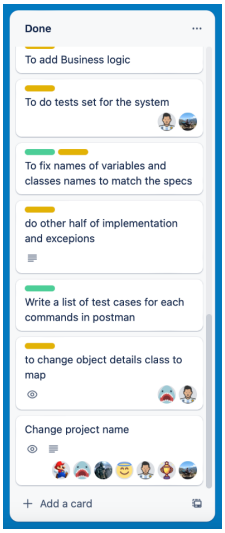
- Beginning of this sprint - 28/03/2023



- End of this sprint - 18/04/2023

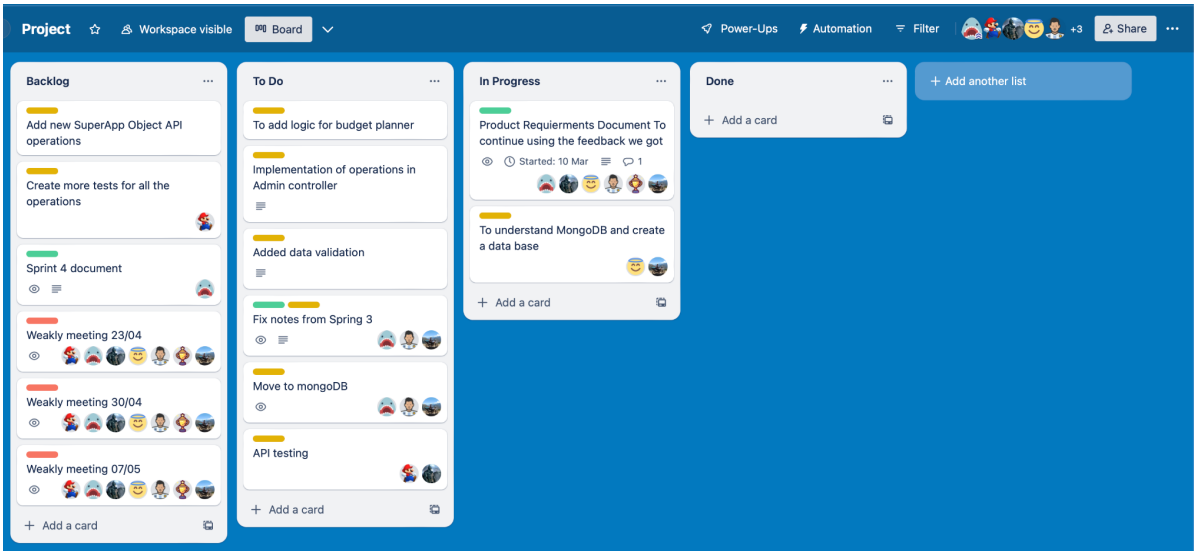




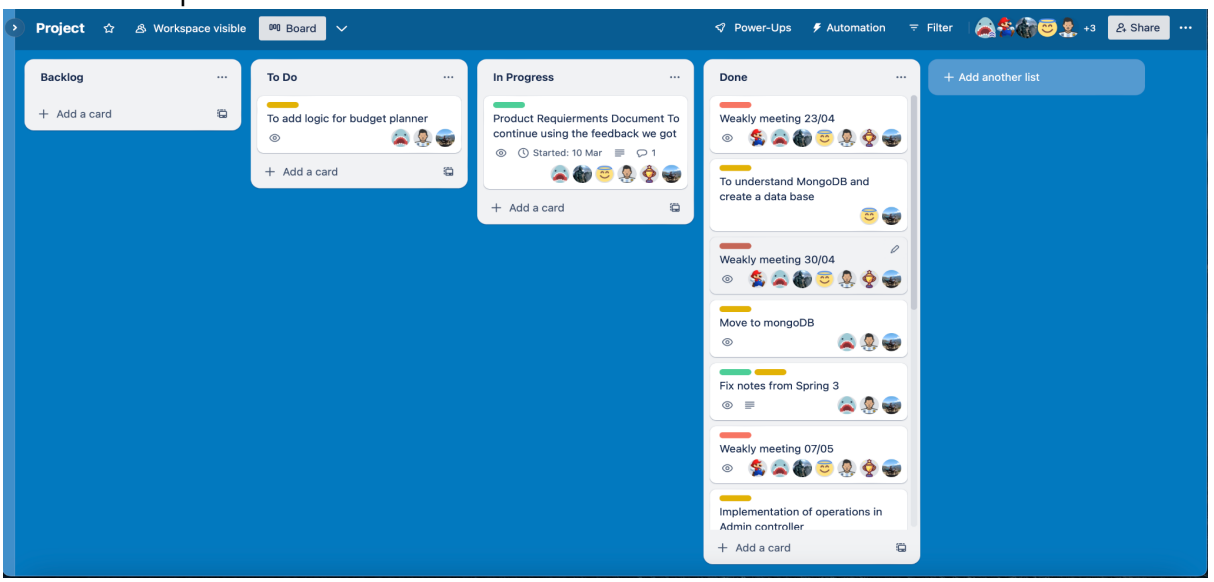


4th Sprint

- Beginning of this sprint - 18/04/2023

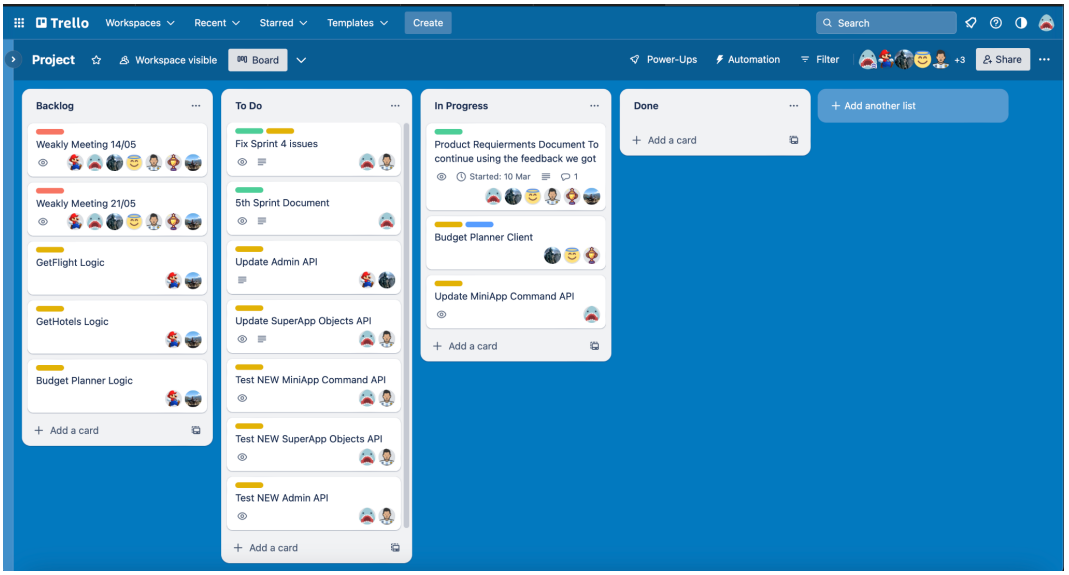


- End of this sprint - 09/05/2023

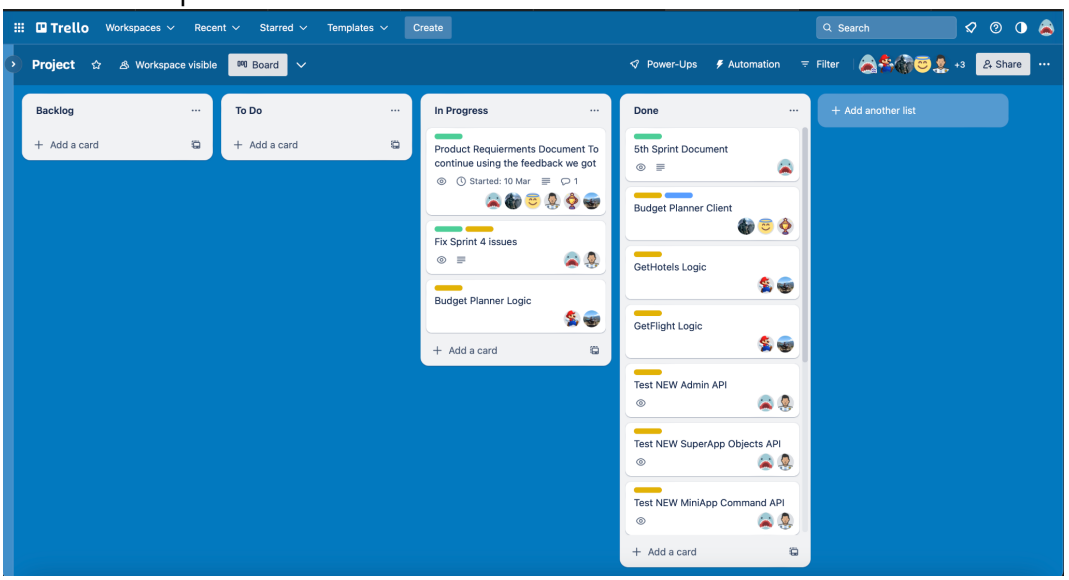


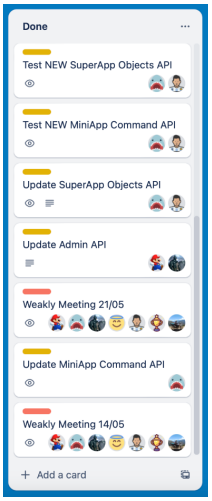
5th Sprint

- Beginning of this sprint - 09/05/2023



- End of this sprint - 23/05/2023





General Summary of Project

- **What worked well for your team throughout the semester and you will want to preserve in the next projects you participate in in the industry**
    - Effective utilization of Postman for API testing and documentation.
    - Utilizing the Spring Framework for developing a robust backend infrastructure.
    - Leveraging MongoDB for efficient data storage and management.
    - Utilizing React for building a dynamic and interactive user interface.

- **How you can improve your work in other projects that you will participate in in the future**
    - Enhance project planning and requirement gathering to ensure clearer objectives and expectations.
    - Implement regular code reviews and collaboration to improve code quality and maintainability.
    - Utilize comprehensive automated testing strategies to identify and address potential issues early on.
    - Foster more effective communication and coordination among team members.

- **What did you enjoy most about working on the project?**
    - Discovering new techniques, acquiring knowledge in specific areas, and developing new skills.
    - The challenge of solving complex problems or overcoming obstacles within a project is exciting and engaging

- **What would you do differently if you were to start the project now, after the knowledge and experience you have gained this semester**
    - It is important to establish clear project goals and ensure that all team members understand and are aligned with them. This helps create a sense of purpose and direction, and enables the team to work towards a common objective.
    - Implement more regular progress tracking and milestone evaluation to monitor project development effectively.
    - Emphasize the importance of documentation and code commenting for improved maintainability and knowledge transfer.
    - Allocate more time for user testing and feedback to refine the system based on user needs.

- **Was the teamwork also done remotely, without physical meetings?**
  Yes, we used a WA group for quick updates and messaging. For our weekly meetings and working groups we used the Discord platform. And to manage our assignments we used Trello.

6th Sprint

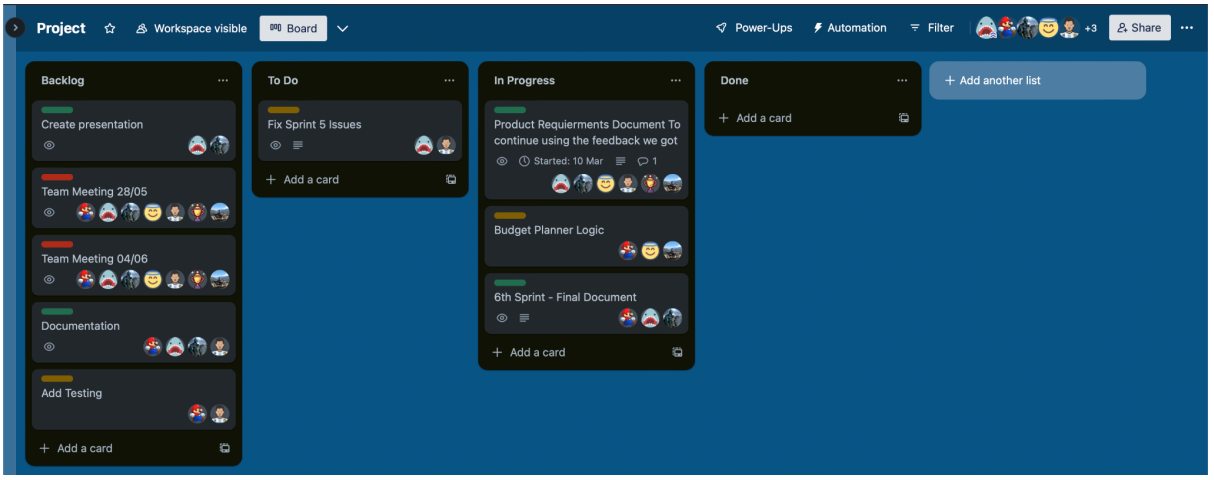
06/06/2023 - Integrative Software Engineering - Semester B 2023 - Travel Genius

List of Students detailing for each student

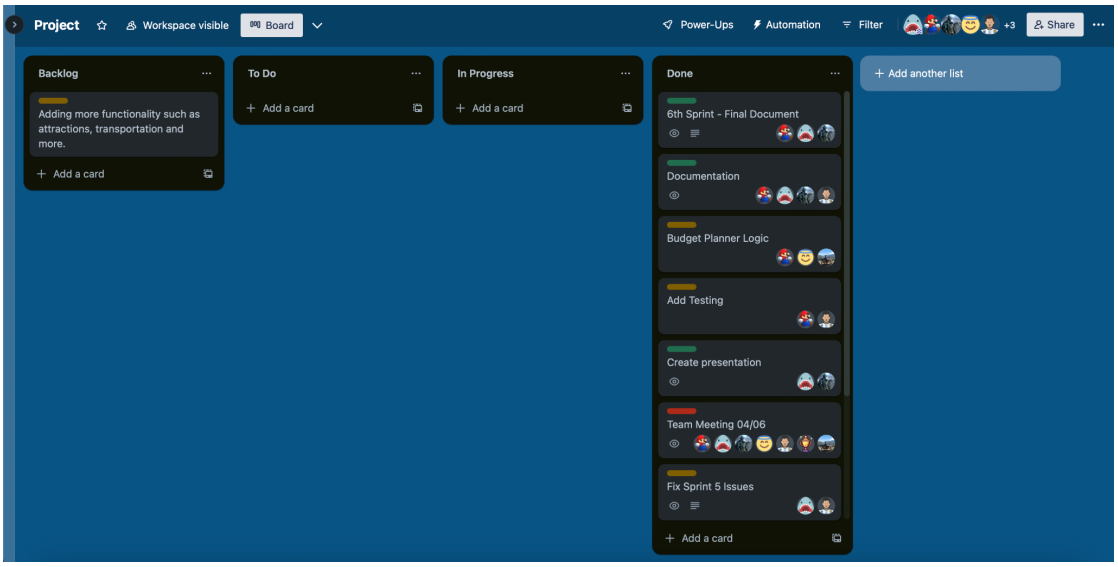
Student Name	Student ID	Student Avatar	Roles in Team
Omer Iny	206571739		Team Leader, Team
Ido Oriane	318420361		Product Owner, System Architecture, Team
Tamer Taji	209274166		QA Engineer, Technical Writer, Team
Yuval Shaib	207209792		UI/UX, Team
Matan Elad	206272700		Scrum Master, Team
Noy Ben Moshe	315783738		DBA, Team
Avi Baazov	318603271		DevOps, Team

Kanban Boards of this sprint:

- Beginning of this sprint - 23/05/2023



- End of this sprint - 06/06/2023



General summary of work

- **What went well for the team and should be continued in the next phases of work**
    - We finished the work on time, we splitted assignments between 2 groups, Group 1 worked on the API logic implementation, Group 2 worked on documents.
    - The communication between the teams was more effective and we shared info and questions in real time on the WA group.
    - We perform testing in Postman, JSON files are attached to sprint folder and also links to Postman platform:
      Admin | checkRelations | miniApp | Object | User | checkWithPagination | AsyncCheck
- **What should be improved in team work**
    - Communication is improving, we talked about the contribution of each member to the team and everyone agreed that not everyone contributes the sum for the team success.
- **What problems did the team encountered through this phase of work**
    - We wanted to use the annotation of lombok, @toString, @Data, @noArguments, etc. We faced an issue using setters and getters in the DB classes.
- **Why did we not complete all planned work**
    - We push the logic and client in order to delivery the project in time for the client, so we had to dealy sprint 5th issues fixing to next sprint
- **What is expected for the next sprint**
    - Fixing User authorization control, Active query, Pagination for parents and children
    - Adding more functionality such as attractions, transportation and more.

Known Issues

- The date selector chooses the date before the date you choose
- When login if the username doesn't match it will login anyway this is because of the specs of the login in the usercontroller
- When you return back and choose a new flight, and you have already booked a hotel. The budget manager will reserve both.
- Get all parents and children do not support pagination.

ReadMe - How to install React

1. Clone the repository:

    git clone https://YOURUSERNAME@bitbucket.org/travelappdevintegrative/2023b.omer_iny_react.git

2. Navigate to the project directory:

    cd travel-genius

3. Install the dependencies :

    npm install --legacy-peer-deps

4. Make sure that the spring application is up and running

5. Start the development server:

    npm start

TravelGenuis

Team Members

Omer Iny
Team Lead, Team

Ido Oriane
Product Owner, System Architecture, Team

Tamer Tji
QA Engineer, Technical Writer, Team

Yuval Shib
UI/UX, Team

Matan Elad
Scrum Master, Team

Noy Ben Moshe
DBA, Team

Avi Baazov
DevOps, Team

MiniApps

01
Budget Planner

Help users plan and manage their travel budget

02
Flight Booking

Allow users to search for and book flight

03
Hotel Booking

Allow users to search for and book hotels

01

Budget Planner

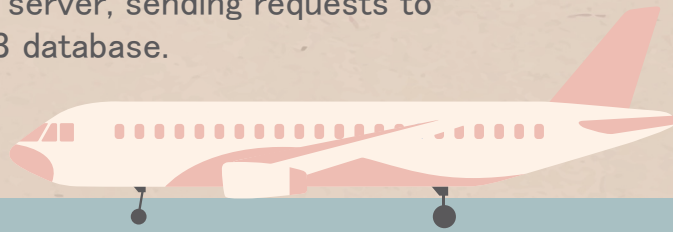
The Budget Planner app is designed to help users manage their finances and plan their expenses. It allows users to track their income, set budget limits for different categories, and monitor their spending.

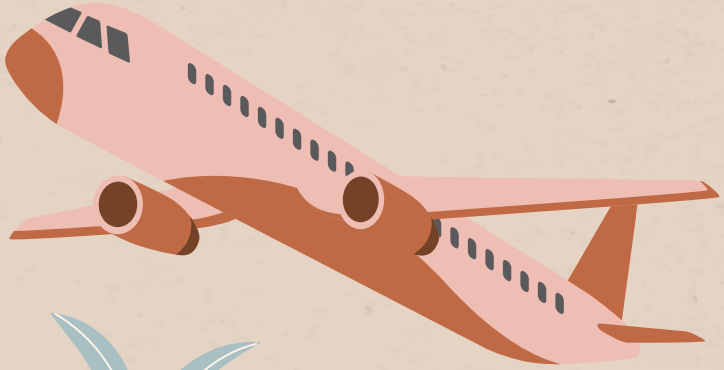
For the budget planner we used some of the following:

$ React: helped us create the user interface, allowing users to interact with the app and view their budget information.

$ MongoDB: used to store user data, including income, expenses, and budget limits. It would provide a persistent storage solution for retrieving and updating budget information.

$ Axios: used to communicate with the backend server, sending requests to retrieve and update budget data in the MongoDB database.





02
Flight Booking

The Flight Booking app enables users to search for available flights, select their desired flights, and make bookings. It provides a convenient way for users to plan and book their flights.

For the flight booking we used some of the following:

✈ Spring framework: was used on the backend to handle flight search requests, process bookings, and communicate with the flight booking database.

✈ MongoDB: store flight data, including flight schedules, availability, and booking information. It would provide a reliable and scalable database for managing flight-related data.

03 Hotel Booking

The Hotel Booking app allows users to search for available hotels, view hotel details, and make bookings. It provides a seamless experience for users to find and reserve accommodations.

For the flight booking we used some of the following:

- MongoDB: store hotel data, including hotel details, availability, and booking information. It would serve as a reliable and scalable database for managing hotel-related data.

- Axios: helped us to send requests from the frontend to the backend API, retrieving hotel information, checking availability, and making bookings.

General Overview

The flight and hotel booking system is a comprehensive platform that allows users to plan, search, and book flights and hotels. It provides a user-friendly interface and efficient functionalities for budget planning, flight booking, and hotel booking. The system incorporates various technologies to handle frontend development, backend processing, data storage, and communication with external services.