

TQS: Product specification report

Maria Rafaela [107658], Marta Oliveira [107826], Gabriel Teixeira [107876], Fabio Matias [108011]

v2024-06-03

1 Introduction.....	2
1.1 Overview of the project.....	2
1.2 Limitations.....	2
2 Product concept and requirements.....	3
2.1 Vision statement.....	3
2.2 Actors and Requirements.....	3
2.3 Personas and Main Scenarios.....	3
2.4 Project epics and priorities.....	6
3 Domain model.....	8
4 Architecture notebook.....	9
4.1 Key requirements and constraints.....	9
Key Requirements:.....	9
Constraints:.....	9
4.2 Architecture view.....	9
4.3 Deployment architecture.....	10
5 API for developers.....	12
6 References and resources.....	13

1 Introduction

1.1 Overview of the project

This project assignment has as its main goal to implement a medium-sized project developing a Minimum viable product (MVP) with functional specification, system architecture and the project implementation.

The project should be implemented in partnership with a Software Quality Assurance (SQA) strategy. With this, the project itself should all be tested to ensure its quality and should also apply engineering practices of Continuous Testing, Continuous Integration and Continuous Delivery.

With this in mind we are developing and testing the Airport Organizer, a system that allows the management of an airport from the ticket reservation process to the boarding.

This assignment and the intended approach is the appliance of the knowledge acquired in TQS classes during the semester. The importance of testing while developing a product, in its different phases is the main idea of that knowledge.

1.2 Limitations

Overall the project had the desired and planned features implemented for the MVP (Minimum Valuable Product). Although there were some little features that were not implemented, such as the seats confirmation. Having in mind that he had other difficulties like unjustified errors creating the database tables, that stole us a lot of time. With the short time that we had we defined our priorities and left little features to future work. That way, features only to make the systems more user friendly were left unimplemented.

2 Product concept and requirements

2.1 Vision statement

The Airport Organizer is a digital system that implements a solution to handle flight reservations, the clients check-in at the airline companies so they catch their flight and to give useful information to guide the clients during their stay at the airport.

The main focus are the flight reservations by an user and administration services that check-in a passenger and change the state of a flight in the boards.

To address all these questions, the platform must include three main services. The first one is a personal portal in which users can reserve a flight, check their reservations and search for different flights with different prices.

The second service should be developed for the administration staff and staff that verifies the passengers and their luggage. This one needs to handle the registration of flight passengers, their luggage and their check-in. Staff can access waiting lines, access delayed or canceled flights and call for passengers to check luggage or to board a flight.

The last service is a simple digital signage solution to handle the call screens at the airport. The calls will translate the boarding calls for the flights.

2.2 Actors and Requirements

As actors interacting with this system we have passengers, who want to travel, and the administration staff.

The passengers should be able to register, log in and log out at the personal platform. The passengers should be able to search for a flight and reserve his seat and the flight. The passenger should be able to pay for his flight through the application. The passenger should be able to check-in before he enters his flight. The passenger should be able to have informations about his flight while at the airport. The passenger should be able to consult his reservations.

The administration staff should be able to check-in passengers and their luggage. The administration staff should be able to access waiting lines and delayed or canceled flights. The administration staff, should be able to verify the passenger before they enter their flights. The check-in staff should be able to check-in the passengers.

2.3 Personas and Main Scenarios

Personas



Passengers:

- Marie

Marie is a twenty five year old business woman. She is a content creator who travels a lot to meet new cities and new brands. Marie is a famous content creator and for that reason many brands look for her and invite her for many events around the world. Marie loves this life and loves music, fashion and all sports involving water and sea.

During the initial phase of being a content creator, Marie always took care of herself including all the preparation to attempt all these events and for that reason she continued to take care of her flights reservations, luggage and hotel reservations.

To simplify all the preparations, Marie wants a user-friendly platform where she can easily search for flights based on her preferences such as dates, times, and prices. She also wants the ability to manage her reservations and make changes if necessary due to his busy life. She needs a seamless booking experience with clear information about available flights, including prices and schedules. She appreciates features like seat selection.

Marie becomes frustrated when the booking process is complicated or when it takes a lot of time to book a flight since her life is extremely busy and unstable.

- Peter



Peter is a twenty one year old student that never traveled alone. This summer he decided to do an interrail and the first stop was in his country's airport. Peter loves the beach, sports and adventure and is studying biology at Aveiro's University. As he never traveled alone he was extremely lost at the airport and didn't know where his flight was and at what time he should be on the plane.

Peter's objective is to navigate through the airport and find the departure gate for his flight. As a first-time solo traveler, he seeks guidance and assistance to ensure a smooth and stress-free journey. Peter needs clear and easily accessible information to help him navigate the airport and locate his departure gate. He requires a system, in this case a digital signage, that provides real-time updates on flight departures and gate assignments.

One of Peter's pain points is feeling overwhelmed and lost in the airport environment, especially as a first-time solo traveler. He may experience anxiety and uncertainty about where to go and how to reach his destination within the terminal. Without clear signage and guidance, he may struggle to find his way and risk missing his flight.

Peter is motivated to use the digital signage system because it offers him the guidance and direction he needs to navigate the airport effectively. The system's ability to display real-time flight information, including departure times and gate assignments, provides Peter with the assurance that he's heading in the right direction. By relying on the digital signage, Peter can focus on enjoying his travel experience and successfully reaching his destination.

Administration Staff :

- Mark



Mark is a thirty year old experienced airline employee who works in the administration department. He loves classic music, organization and loves his jog. He's responsible for managing passenger registrations, luggage, and ensuring smooth operations at the airport, accessing waiting lines and exposing information like delayed or canceled flights.

Mark's primary goal is to efficiently handle passengers traffic inside the airport and luggage procedures while adhering to safety and security protocols. He needs to manage flight manifests, handle

changes to flight statuses, and communicate effectively with passengers. Mark requires a robust system that allows him to access flight information in real-time, manage passenger lists, and handle unexpected situations such as flight delays or cancellations. He also needs tools to communicate with passengers and coordinate boarding procedures.

Mark faces challenges when dealing with outdated or inefficient systems that slow down the check-in process. He also finds it difficult to manage communication with passengers during busy periods or emergencies without adequate tools or support like digital signage.

- Sophie



Sophie worked as a flight attendant for five years at an international airline before transitioning to customer service in a busy airport. She is familiar with check-in, security, and boarding procedures. Sophie is 35 years old and holds a degree in Tourism and Hospitality with a specialization in Aviation. She likes books, animals and to know more about the countries around the world. Furthermore, Sophie has excellent communication skills, she is fluent in Portuguese, English, Spanish and French, enabling her to effectively interact with passengers from different nationalities. She is adept at

handling challenging situations and resolving issues efficiently, ensuring customer satisfaction. Sophie puts passengers' needs first and strives to provide a pleasant and smooth travel experience and has familiarity with technology so she is comfortable with electronic check-in systems and other technologies used in the aviation industry.

Sophie's objective is to efficiently verify passengers and facilitate the check-in process to ensure smooth and timely departures for flights. She needs a reliable system that enables her to quickly and accurately verify passengers' identities, check them in for their flights, and handle any special requests or situations that may arise.

One of Sophie's pain points is dealing with long queues and delays during the check-in process, especially during peak travel times. This can lead to frustrated passengers, increased stress levels for staff, and potential disruptions to flight schedules. Additionally, manual verification procedures can be time-consuming and prone to errors, resulting in inefficiencies and customer dissatisfaction.

Sophie is motivated to use the system because it offers a solution to her pain points by automating and expediting the check-in process. The system's ability to quickly verify passengers, handle special requests efficiently, and reduce wait times aligns with her goal of providing exceptional customer service and ensuring on-time departures.

Main Scenarios

Scenario: Flight Reservation

Marie accesses the Airport Organizer system to make a flight reservation for a fashion event in Chaves. She uses the platform to browse flights, filtering by dates, times, and prices. After finding the ideal option for her schedule, she selects it and proceeds to confirm her preference. At the end of the booking process, she securely makes the payment.

Scenario: Passenger Baggage Management

Mark starts his shift at the airport and accesses the Airport Organizer system to check the day's flights. He updates the passenger list and their respective baggage, ensuring that everything is in its proper place and secure. During a delayed flight, he uses the system to inform passengers about the new flight schedule and organizes the waiting queue to be more efficient.

Scenario: Check-in Process

Sophie at the check-in counter uses the system to quickly verify the identity of all passengers and process their data, ensuring that they receive the necessary assistance and support for their boarding.

Scenario: Airport Navigation

Peter, lost and alone on his first airplane trip, observes the digital signs that show real-time information about flights and boarding gates. By following clear and updated instructions, Peter easily finds his boarding gate to check-in and feels more confident about his trip.

2.4 Project epics and priorities

To develop our solution to the assigned project we will use the agile method with iterations of one week, with a total of 5 iterations.

Week 2 (25/04 to 09/05)

1. System Modeling:
 - Configure the backend data model to support flight reservations, baggage management, and check-in functionalities.
2. Prototype Development:
 - Develop prototypes of the web and mobile applications, focusing on the user interface for flight search and reservation, and user profile access.

Week 3 (09/05 to 16/05)

3. (Epic) Flight Search:
 - Implement the flight search process.

4. (Epic) Passenger Experience Enhancement:

- Implement additional features such as seat selection and reservation management to enhance the user experience.

Week 4 (16/05 to 23/05)

5. (Epic) Account Setup:

- Implement the login and registration process.

6.(Epic) Create a Reservation:

- Implement the reservation process after the flight search process.

Week 5 (23/05 to 30/05)

7. (Epic) Check-in Process:

- Implement the check-in process, including passenger identity verification and data processing.

8.(Epic) Luggage Registration:

- Implement the luggage registration process in the administration interface.

9. (Epic) Airport Navigation:

- Develop a digital signage system to provide real-time flight and boarding gate information.

10. (Epic) System Reliability Assurance:

- Finalize development, ensuring the system is reliable and secure for all users, including security checks.

3 Domain model

We created several entities in our database that relate to each other, namely airlines, flights, users, passengers, reservations, luggage and seats.

Flights are dependent on the airlines and the seats table stores the seats occupied on each flight and is dependent on the flights. Passengers are dependent on users, reservations are dependent on passengers and luggage is dependent on the reservations.

The user and passenger tables were created separately, since a person with an account can book a trip for someone else, for example a family member, making the reservation on behalf of that passenger and not with their details as a user.

The occupied seats table had to be created because initially, when entering data, the database conflicted with the flights entity having an array of occupied seats.

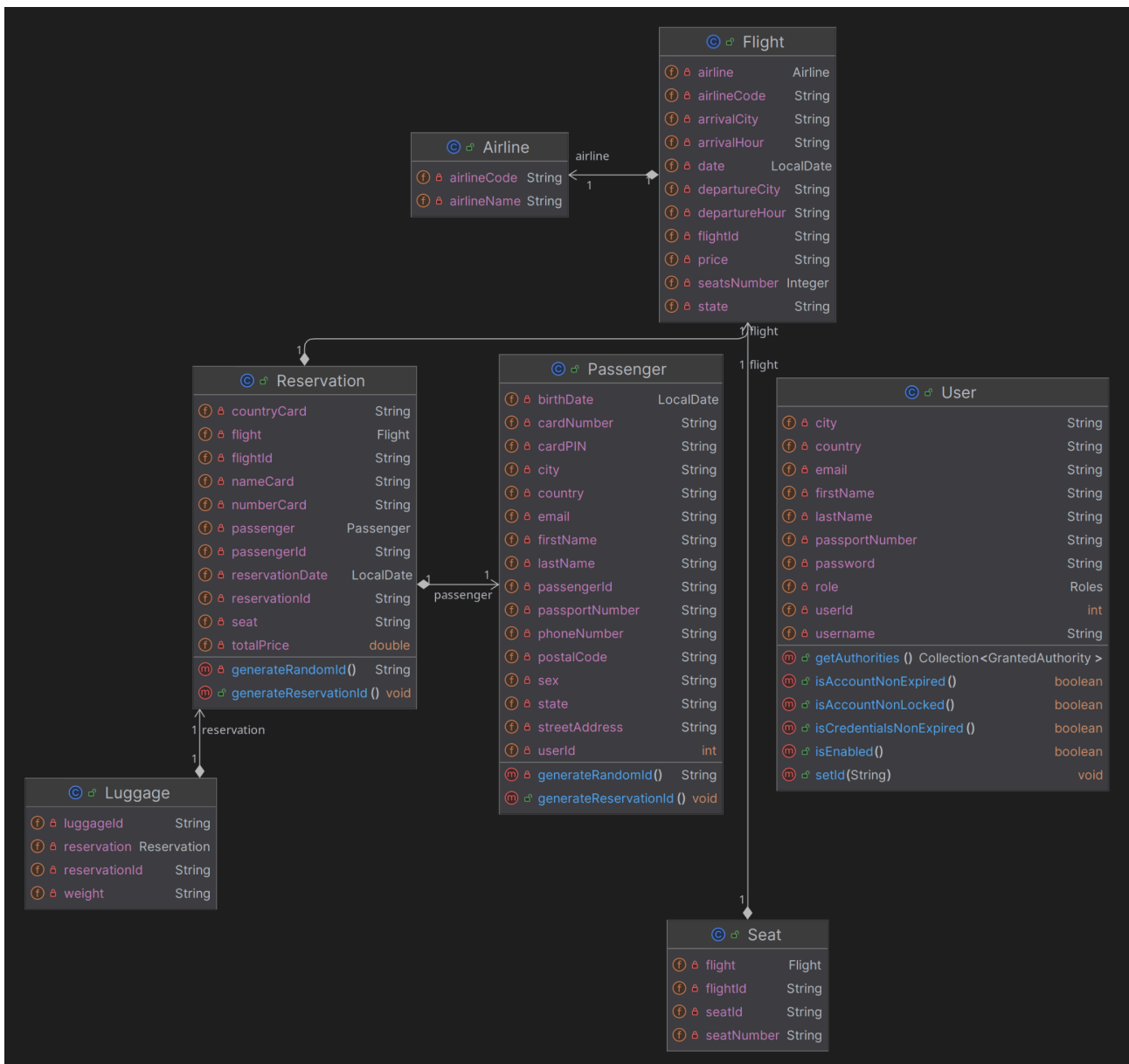


Figure 5: Domain Model Diagram

4 Architecture notebook

4.1 Key requirements and constraints

Key Requirements:

- **System Scalability:** The Airport Organizer must efficiently scale to handle varying numbers of concurrent users, from individual travelers to large groups, without degradation in performance.
- **Real-Time Data Processing:** Given the dynamic nature of flight statuses, the system must process and reflect updates in real-time to ensure accurate dissemination of flight information.

- **Integration with Legacy Systems:** The system must integrate seamlessly with existing airport infrastructure systems for flight scheduling, passenger management, and more.
- **Multiple Platform Support:** The Airport Organizer should be accessible via various platforms, including web interfaces, mobile devices, and informational kiosks within the airport.

Constraints:

Hardware Independence: The system architecture should not rely on specific hardware, making it adaptable across various devices used within the airport ecosystem.

High Availability and Disaster Recovery: The system must be designed for high availability and include strategies for disaster recovery to maintain service continuity.

4.2 Architecture view

To correspond the project expectations for a user interface, an administration interface and a digital signage, we developed three main interfaces connected by the same backend. All the interfaces were developed using React and CSS.

All the backend works due to SpringBoot. We developed 6 main controllers which have all the endpoints needed to make the system work. All the controllers call the methods defined in the services and the services make calls to the repositories that communicate with our MySQL Database.

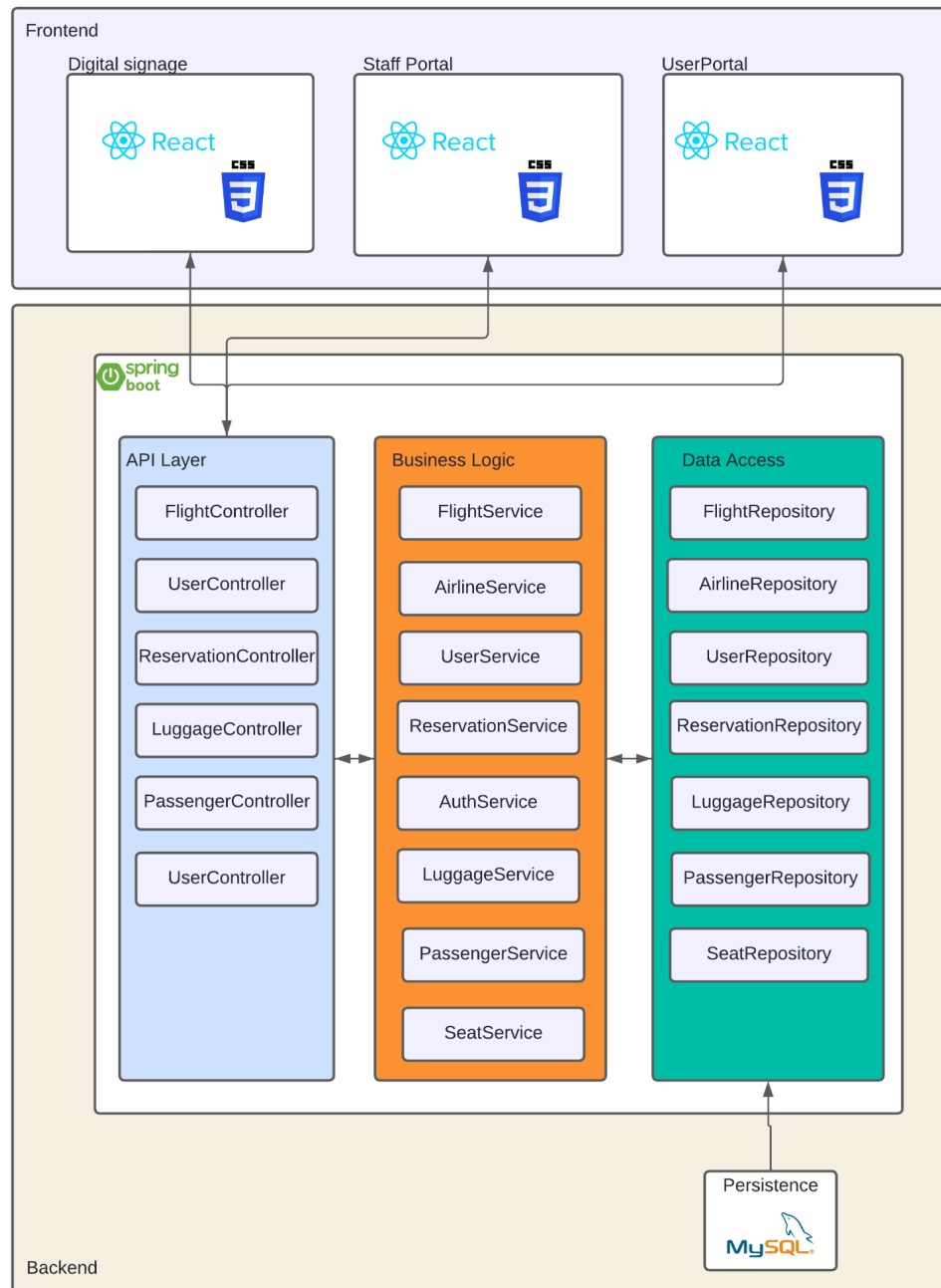


Figure 6: Architecture Diagram

4.3 Deployment architecture

Having in mind the designed architecture, we planned to do the deployment in a known way. All the backend and frontend needed to run our system should be dockerized and then the docker container should be uploaded and published to the virtual machine given by the professor.

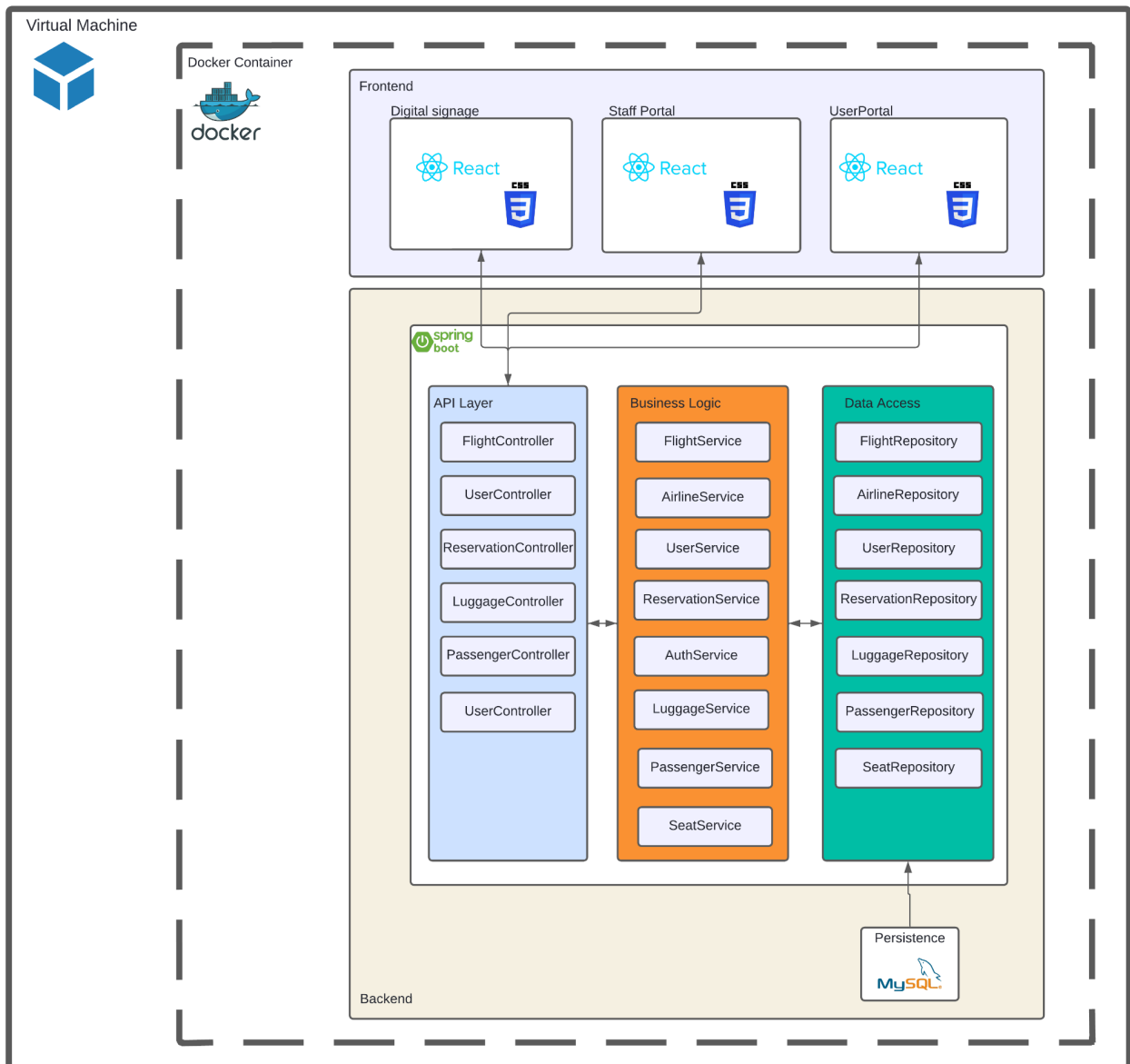


Figure 7: Deployment Architecture

5 API for developers

We have 6 controllers, for flights, users, passengers, reservations, luggage and seats. Which one has their endpoints, GET or POST that can have some request parameters.

Airport Manager API 1.0.0 QAS 3.0

API documentation for the Airport Manager application

Servers
http://localhost:8981

default

GET	/api/flights/airlines	Get all airlines	▼
GET	/api/flights/cities	Get all cities	▼
GET	/api/flights/flights	Get all flights	▼
GET	/api/flights/flights/{state}	Get flights by state	▼
GET	/api/flights/flightCheckout/{flightId}	Get flight details by flight ID	▼
POST	/api/flights/searchFlight	Search for flights	▼
GET	/api/luggage/luggage	Get all luggage	▼
GET	/api/luggage/luggage/{id}	Get luggage by ID	▼
POST	/api/luggage/createLuggage	Create a new luggage entry	▼
GET	/api/passengers/passengers	Get all passengers	▼
GET	/api/passengers/passengerInfo	Get passenger by ID	▼
GET	/api/passengers/passengers/{state}	Get passengers by state	▼
POST	/api/passengers/createPassenger	Create a new passenger entry	▼
GET	/api/passengers/user/{userId}	Get passengers by user ID	▼
POST	/api/reservations/createReservation	Create a new reservation	▼
GET	/api/passengers/user/{userId}	Get passengers by user ID	▼
POST	/api/reservations/createReservation	Create a new reservation	▼
GET	/api/reservations/getAllReservations	Get all reservations	▼
GET	/api/reservations/getReservation/{reservationId}	Get reservation by ID	▼
GET	/api/reservations/getReservationsByPassenger	Get reservations by passenger ID	▼
GET	/api/accounts/userInfo	Get user info by ID	▼
GET	/api/accounts/userInfoByUsername	Get user info by username	▼
GET	/api/accounts/users	Get all users	▼
POST	/api/accounts/register	Register a new user	▼
POST	/api/accounts/login	User login	▼
GET	/api/seats/seats	Get all seats	▼
GET	/api/seats/seatsByFlight	Get seats by flight ID	▼
GET	/api/seats/seatInfo	Get seat info by ID	▼
POST	/api/seats/createSeat	Create a new seat	▼

Figure 8: Swagger Documentation

6 References and resources

<https://swagger.io/solutions/api-documentation/>

<https://spring.io/guides/gs/spring-boot>

https://github.com/RobertoCastro391/IES_Project_TravelingBooking