Design Document for

Booking platform

-Semester 3 Individual Project-

Plamen Peev

Student number: 4179080

Contents

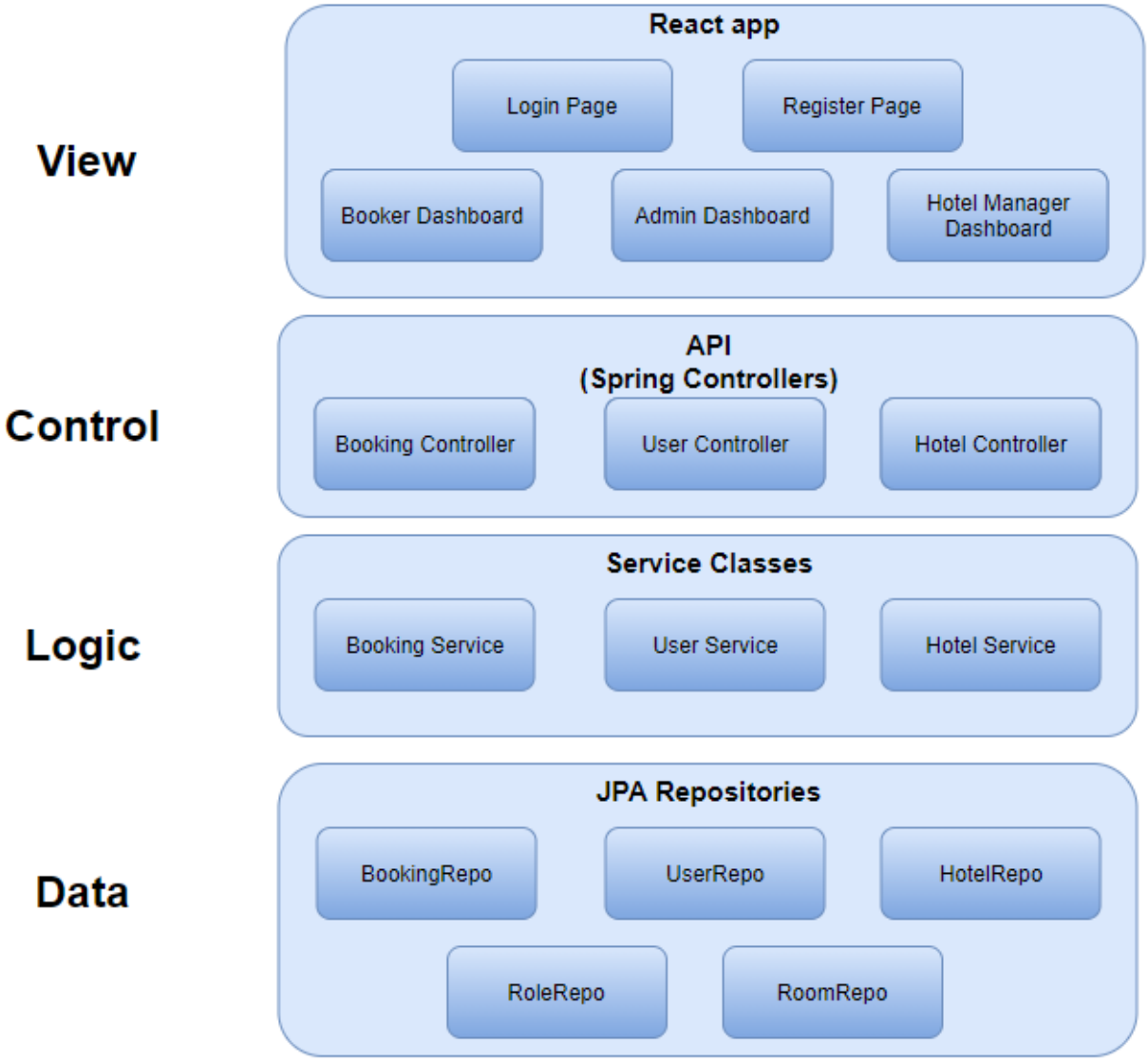
Introduction

The purpose of this document is to outline the design for iTrips. This will include a view of the high level architecture. C4 architecture will be provided to show how the system will be put together and how data will flow through the system. There will also be discussion on the technologies that I will be using throughout this project. This document provides an outline of the application workflow to demonstrate how it will be formatted

System Overview

1.High Level Description

The architecture consists of 4 general layers through which the flow is being processed:



The View sends http request to the Control. If authorization/authentication is successful, the Control, calls the necessary functions from Logic. Logic then gets the necessary data from the Data layer, performs the operations and returns a DTO with the requested information to the Control. The Control then sends the data back to the View in JSON format, and finally the data is being rendered in the UI.

2.Technology Stack

Server

- **Spring Boot** - An open-source java micro-framework with auto configurable production-grade Spring application.

- **Gradle** - A build automation tool that controls the development process in the tasks of compilation and packaging to testing, deployment and publishing.

Spring Framework makes programming Java quicker, easier, and safer for everybody. Spring is the world's most popular Java framework. It focuses mostly on speed, simplicity, and productivity.
One of the many reasons to choose Spring Boot as a tool, is that it makes developing web applications with Spring Framework even faster and easier through three main capabilities:

1. Autoconfiguration
2. An opinionated approach to configuration
3. The ability to create standalone applications

Client

- **React** - A javascript library for building user interfaces

The main reason I chose React for web development, is because it is a library, and not a framework. Working with framework, you have to fit your code in the framework, whereas with library, it is vice-versa: you have to fit the library in your code. This is my preferred way of working.
A great benefit of React is that it is remarkably flexible. It may be harder to master it than, for example Angular, but once you've learned it, you can use it on vast variety of platforms to build quality user interfaces.

3. Tools

There are a number of tools and technologies that this project will utilize for development.

Server Side

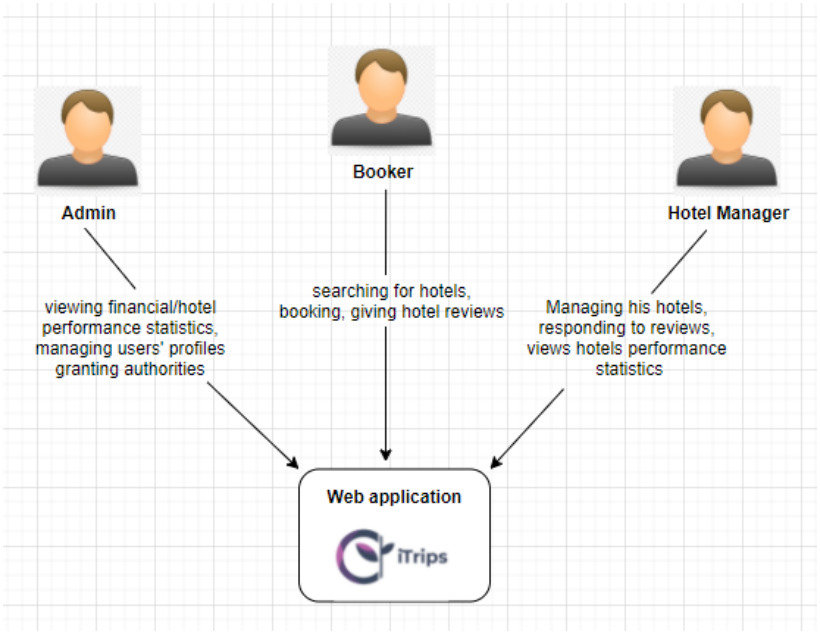
- **JDK 11** - A set of tools for developing and testing programs written in Java and running on the Java platform
- **Spring Data JPA** - abstraction over the Data Access Layer using Java Persistence API and ORM implementations like Hibernate
- **Spring Security** - A framework that handles authentication and authorization at the Web request level as well as the method invocation level
- **Mapstruct** – A code generator tool that greatly simplifies the implementation of mappings between Java bean types based on a convention over configuration approach
- **Auth0 Java-JWT** - An open standard (RFC 7519) that defines a compact and self-contained way for securely transmitting information between parties as a JSON object
- **Joda-Time** – Standard date and time library for java. Provides replacement for the Java date and time classes

Client Side

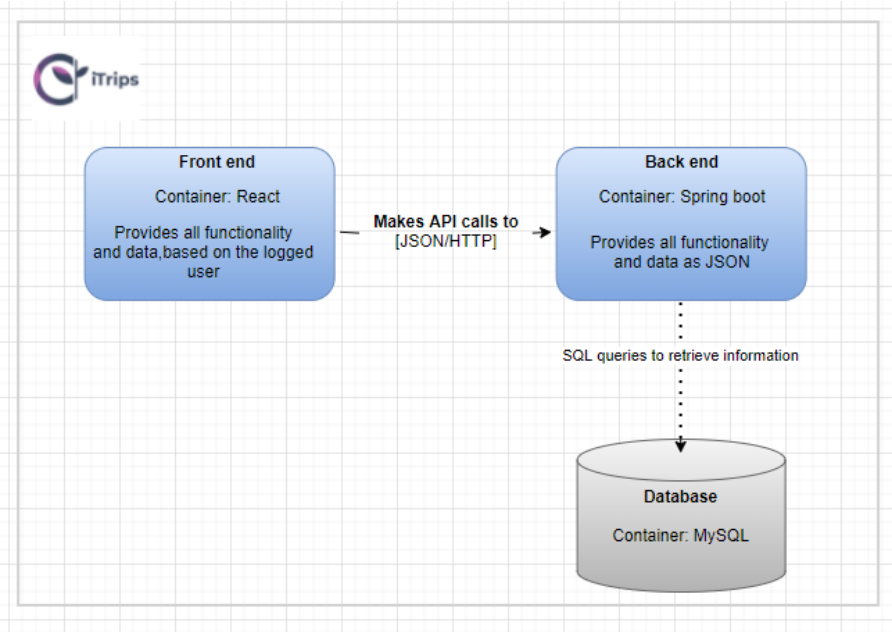
- **Material UI** - A React library that allows easy templating of websites and includes built in responsive styled components
- **Axios** – A Javascript library used to make HTTP requests from node. js or XMLHttpRequests from the browse
- **ApexCharts** – A Javascript charting library that helps creating interactive visualizations for web pages
- **Stomp** – A Javascript library that provides a STOMP over WebSocket client for Web browser and node.js

C4 Architecture

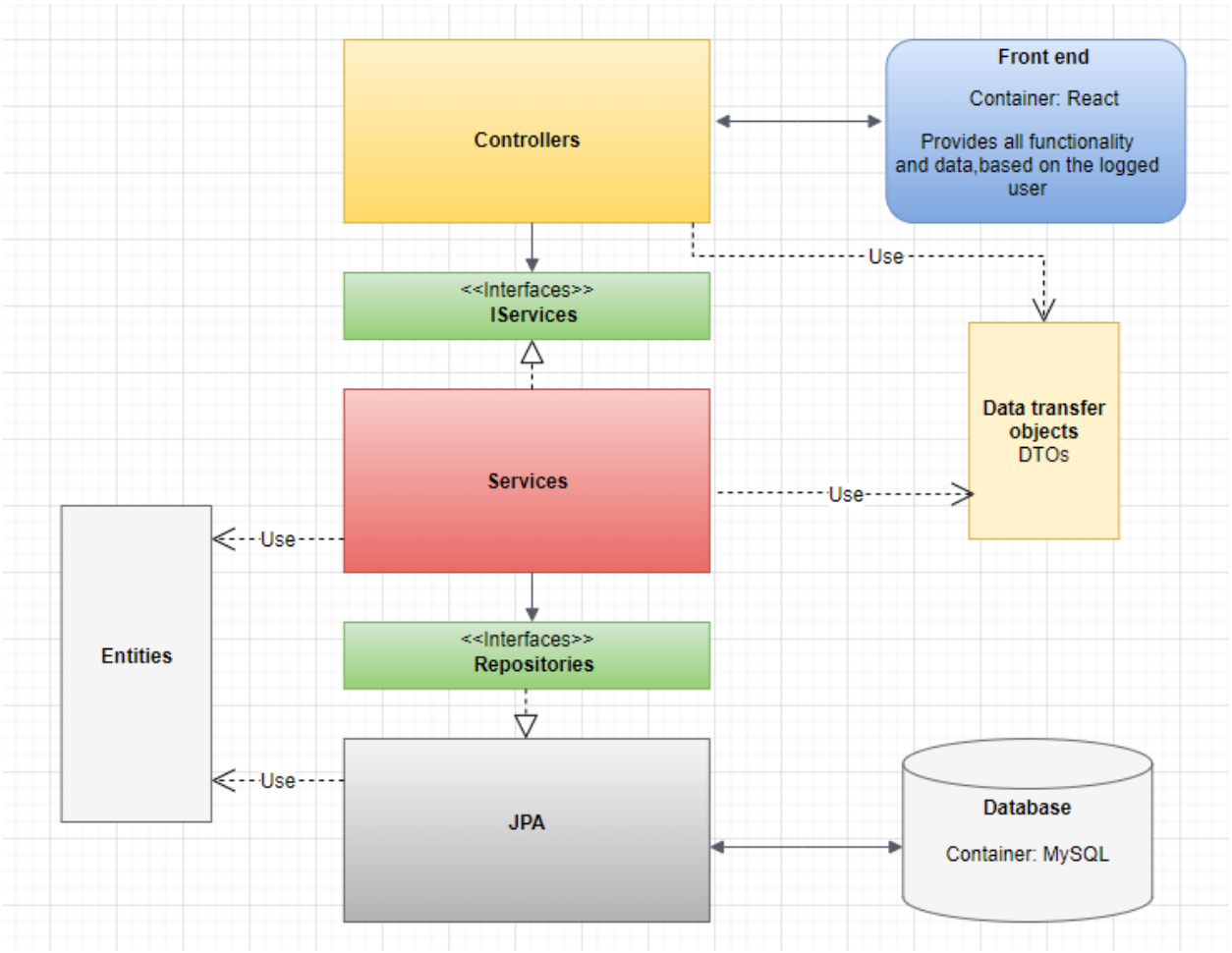
System Context (C1)



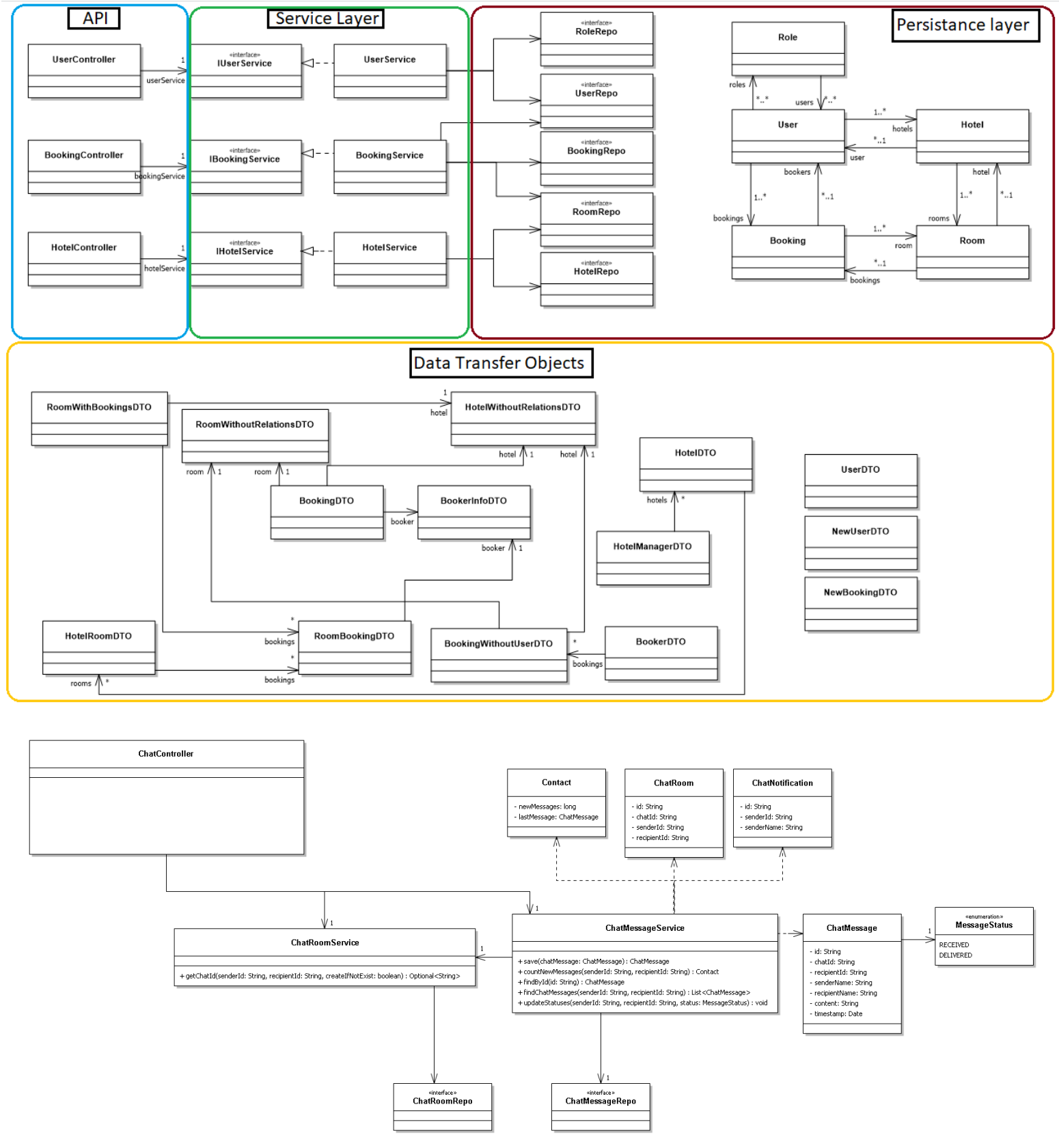
Containers and technology (C2)



Components (C3)


Controllers
Front end
Container: React
Provides all functionality and data,based on the logged user
Use
<<Interfaces>>
IServices
Data transfer objects
DTOs
Services
Use
Use
Entities
<<Interfaces>>
Repositories
Use
JPA
Database
Container: MySQL

Class diagram (C4)



CI Pipeline