

Department of Computer Science
Technical University of Cluj-Napoca

Travel Planner

Name: Zdrenghea Iulia
Group: 30433

Contents

1	Introduction	3
2	Technical Information	4
3	Software Architecture	5
4	Diagrams	6
4.1	Use Case Diagram	6
4.1.1	Client	6
4.1.2	Agency	6
4.2	Package Diagram	7
4.3	Class Diagram	7

Chapter 1

Introduction

The aim of this project is to develop a web application using a database. For now I implemented the backend with the connection to the database. I am creating a Travel Planner which can be used to have better access to different offers from different agencies. A user can login to the account and buy a trip from an agency.

Chapter 2

Technical Information

The database that I am using is written in MySQL. The backend is written in Java Spring Boot and is developed using JetBrains IntelliJ IDEA. The backend is designed using the layered architecture.

Chapter 3

Software Architecture

Layered architecture is used to develop the application. These are the meanings of the packages:

- **Controller:** This package serves as the entry point for incoming requests from the frontend. Contains classes annotated with `@Controller` or `@RestController` annotations, which define request handling methods. It uses the Service layer to execute business logic and retrieve necessary data.
- **DTO:** This package facilitates the transfer of data between the backend and frontend. Contains classes representing data objects tailored to the needs of the frontend. It's used by Controller classes to send and receive data to and from the frontend.
- **Entity:** This package holds the entity classes that represent database tables or documents. Contains POJOs (Plain Old Java Objects) annotated with database mapping annotations. It's used by both the Repository and Service layers for mapping database records to Java objects and vice versa.
- **Repository:** This package is responsible for establishing the connection between our application and the database. Contains interfaces or classes that define the data access methods, such as CRUD operations. It's used by the Service layer to fetch and persist data.
- **Service:** This package implements the business logic of our application. Contains classes that encapsulate application-specific logic and operations. It relies on the Repository layer for data access and manipulation.

Chapter 4

Diagrams

4.1 Use Case Diagram

4.1.1 Client

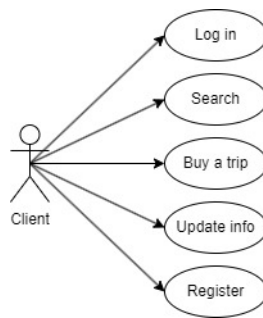


Figure 4.1: Client Use case diagram

4.1.2 Agency



Figure 4.2: Agency use case diagram

4.2 Package Diagram

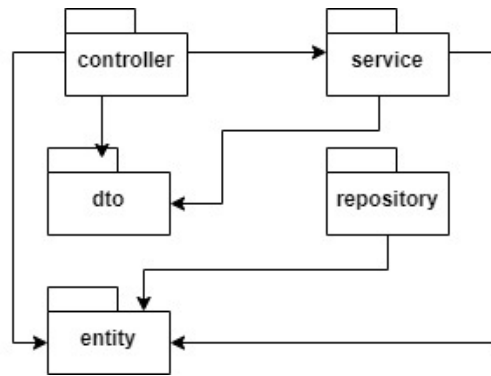


Figure 4.3: Package diagram

4.3 Class Diagram

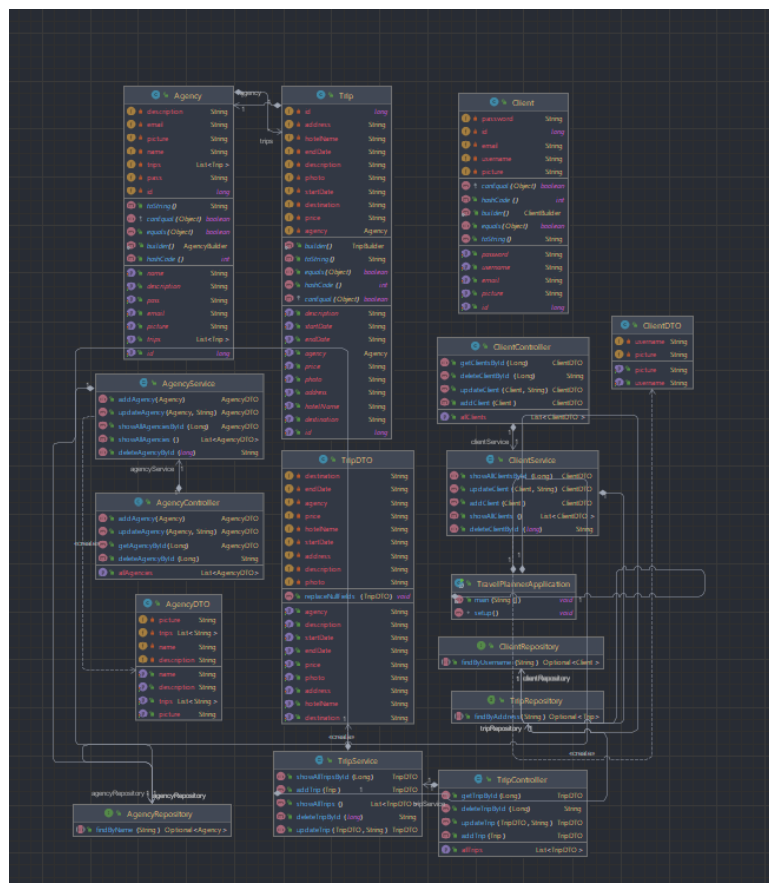


Figure 4.4: Class Diagram