

Juergen Pfau Melih

# **CS 465 Project Software Design Document**

Version 1.0

## Table of Contents

[**CS 465 Project Software Design Document** 1](#_heading=h.gjdgxs)

[Table of Contents 2](#_heading=h.30j0zll)

[Document Revision History 2](#_heading=h.1fob9te)

[Instructions 2](#_heading=h.1fob9te)

[Executive Summary 3](#_heading=h.3znysh7)

[Design Constraints 3](#_heading=h.2et92p0)

[System Architecture View 3](#_heading=h.tyjcwt)

[Component Diagram 3](#_heading=h.3dy6vkm)

[Sequence Diagram 4](#_heading=h.1t3h5sf)

[Class Diagram 4](#_heading=h.4d34og8)

[API Endpoints 4](#_heading=h.4d34og8)

[The User Interface 4](#_heading=h.4h5b3qv007h8)

## [Document Revision History](#_heading=h.lnxbz9)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 |  | Juergen Pfau |  |

## Instructions

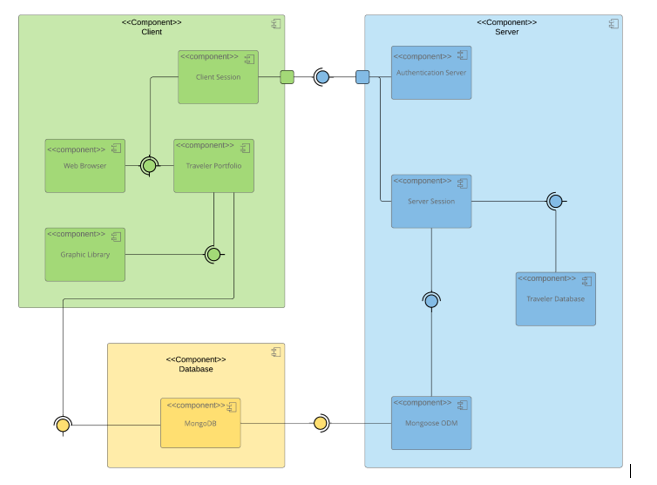
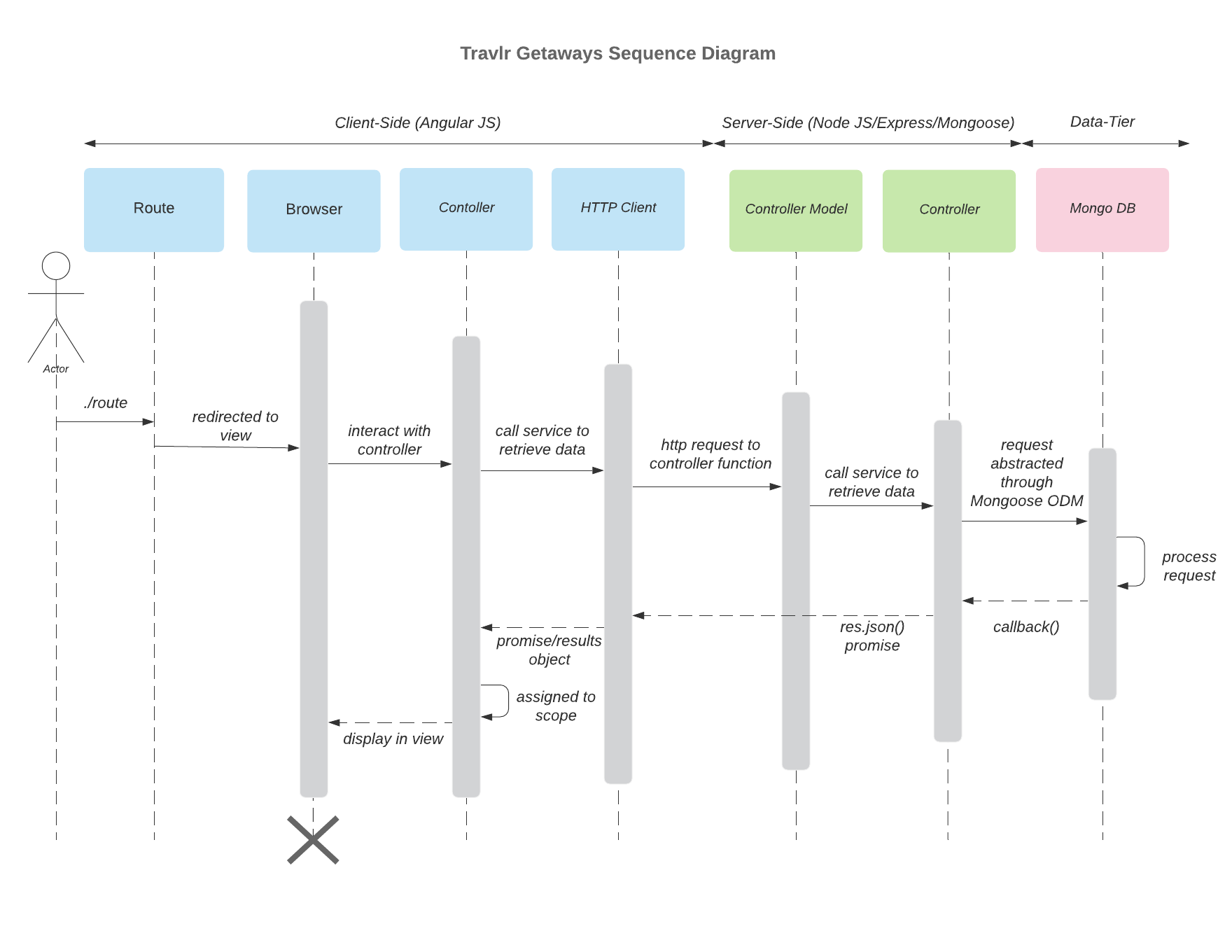
Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

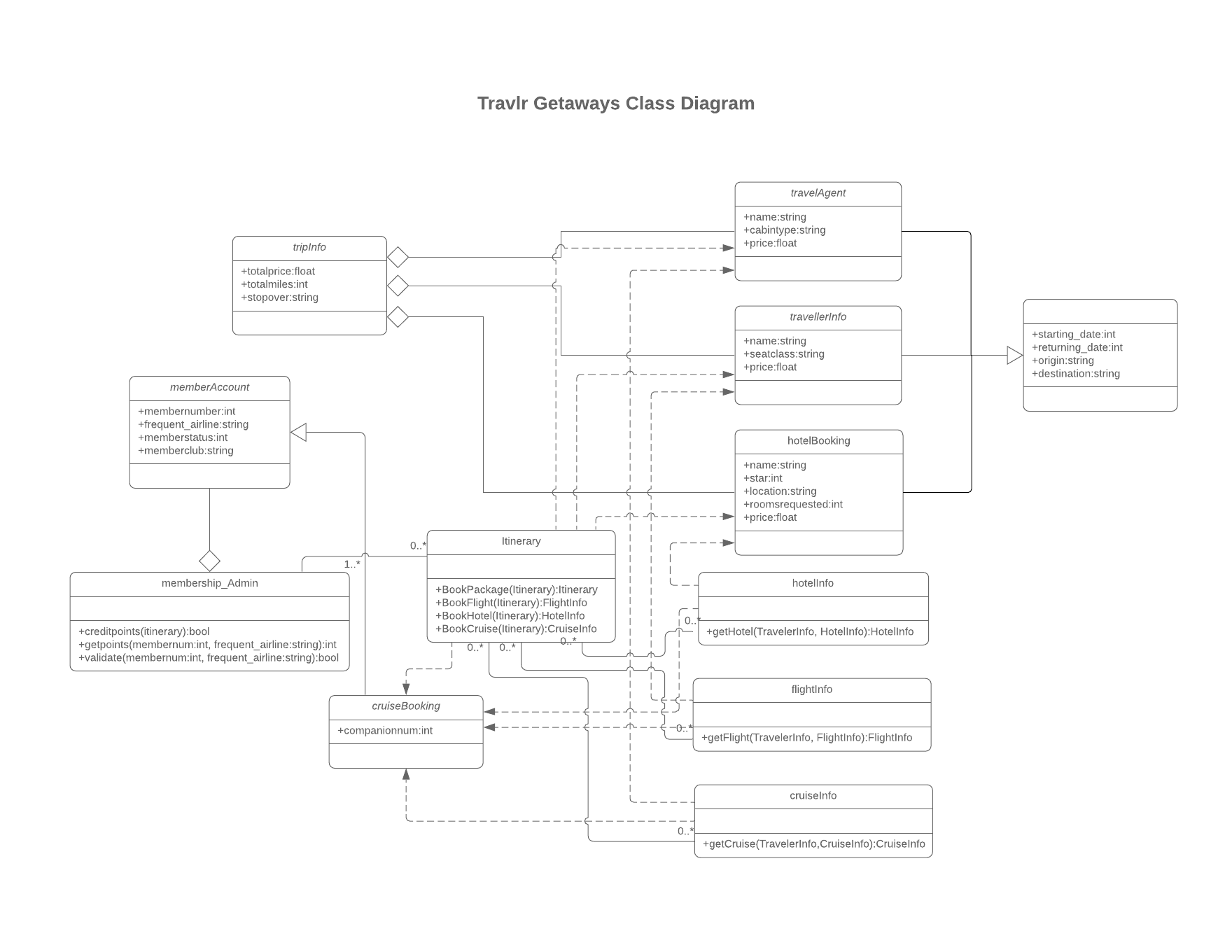
## [Executive Summary](#_heading=h.35nkun2)

End customers will have access to a navigation and booking application through the Travlr Getaways project. To improve accessibility, this will be a web service application. Users will navigate through web pages to book hotel rooms and search tourist attractions. Additional development will provide the ability to book flights to and from destinations. This application is made to be a centralized location for traveling needs and information. The Travlr Getaways application will follow a MEAN design and structure. This process will help with the creation of a complete and fully functioning web application. Frameworks will include Mongo DB for database needs, Express, Angular, and Node.js to help with data manipulation.

## [Design Constraints](#_heading=h.1ksv4uv)

Travlr Getaways is being built as a one-stop shop for all of your vacation needs. Adaptability will be one of the application's design constraints. With both hardware and software, the world is always changing. Instead of using web-only applications, most consumers choose to use a mobile app with simple controls.

  
  
  
The graphic above depicts the components required throughout the application to help the end user achieve their objectives. Clients, a database, and a server are examples of components. Each of these elements is required for the application to be complete. To use the app, you'll need a client, which could be a phone or a PC. The database will serve as a central repository for data and requests. Mongo DB services will be used to support the database. The server will make use of a variety of components, including user authentication, server sessions, Mongoose ODM, and the traveler database. This application's components will all work together to help users achieve their objectives.  
  
The user interactions with the system are depicted in the sequence diagram above. The user first enters the system using a web browser. For authentication, the system consults the database and server. The administrator is in charge of coordinating user interactions with the system, with an emphasis on use settings and authentication. In addition, the administrator is in charge of handling user requests, excursions, and reservations in accordance with available resources. The controller manages the user's request and connects to the HTTP client to send it to the database. The database receives and processes the user clients sent to the Mongoose ODM by the controller model. The request is processed and a callback function is returned. The controller organizes the requests' transformation operations, with the client viewing the replies via the browser. The presentation (client), business, and data access layers all interact with the system. These layers offer the best solution for completing the data coordinating process.

The itinerary module records travel details, connects to cruise information, and stores cabin and route data. Client information is stored in the traveler info table, while facts about the clients' journeys are stored in the flight, hotel, and trip info entities. The system, on the other hand, gathers and saves subscription information in the membership table. Tables for hotel, cruise, and flight bookings incorporate information on restaurants, cruises, and trips based on customer preferences. The travel agent table contains information about the agency. The system also keeps track of customer information in the member account table.  
  
UI:  
Understanding the basic structures utilized in designing and displaying the web application above is critical. The frameworks used in the development process are one factor to consider. The express and angular structures were chosen based on their application areas and ability to achieve the desired results. The express structure has been implemented to help with backend development. The Angular framework, on the other hand, has been used to promote front-end development. The developers were able to display dynamic front end pages while keeping consistency in the back end modules using these methods. A simple web application has only the most basic features and functions. SPA, on the other hand, has a wealth of capabilities that help to build a responsible approach to user and system interaction. This method generates a fluid framework that has an impact on operational efficiency decisions. Users must assess data accuracy when doing API testing.