

Travlr Getaways

# **CS 465 Project Software Design Document**

Version 3.0

## Table of Contents

[**CS 465 Project Software Design Document** 1](#_Toc36198462)

[Table of Contents 2](#_Toc36198463)

[Document Revision History 2](#_Toc36198464)

[Instructions 2](#_Toc36198465)

[Executive Summary 3](#_Toc36198466)

[Design Constraints 3](#_Toc36198467)

[System Architecture View 3](#_Toc36198468)

[Component Diagram 3](#_Toc36198469)

[Sequence Diagram 4](#_Toc36198470)

[Class Diagram 4](#_Toc36198471)

[API Endpoints 4](#_Toc36198472)

[The User Interface 4](#_Toc36198473)

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

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 04/16/2023 | Dani Yosopov | Software Design Document, Travlr Getaways |

## 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)

Travlr Getaways wants to give their customers the ability to have access to vacation information which includes flights, hotels, and tourist sightseeing, on top of that customers, will be able to book it.

This is going to be built using MEAN stack developing web applications.

MEAN stack is a framework for developing web applications.

The Framework will include MongoDB, Express, Angular, and Node.js hence the name MEAN.

MongoDB is the database of the application which stores all the data.  
Express is the application framework for Node.js which is used for web and mobile development.

Angular is a web development platform that provides developers with robust tools for creating the client side of web applications.   
It allows development of single-page web applications where content changes dynamically based on user behavior and preferences.   
Node.js is built for handling asynchronous I/O while JavaScript has an event loop built-in for the client-side this makes Node.js fast compared to other environments.

MEAN is particularly suited to real-time applications like the web application Travlr Getaway needs.

MEAN is only one bundle form Full stack development there are many other bundles like LAMP stack, Ruby on Rails, LEMP stack, Django stack.

When it comes to the full stack development which is an end-to-end development of applications. It includes both the front end and back end of an application. The front end is usually accessed by a client, and the back end forms the core of the application where all the business logic is applied.

This allows better handling of the back and front-end tasks for successful synchronization from both ends.

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

Travlr Getaways wants to be a one-stop shop for your next vacation, this means they need to provide a lot of information about the destination like the prices, photos, attractions, prices, and if it's possible to have it in a package flight and hotel for their customers.

The main implication of the design is adaptability because vacation trends are changing all the time, on top of that in the booking field there are a lot of constant changes in prices and availably dates, so from this, we can understand that it is not feasible to develop a large-scale app.

Another implication of the design is while heavy load scenarios, there may occur a potential loss of records written by MongoDB.

## [System Architecture View](#_heading=h.44sinio)

The first comment is the database using MongoDB which holds all the data for the web application and its users.

This database will feed information to both the server and the client.

In this way, the client will have the basic information like the travel portfolio that includes pictures of the vacation, and the server will act like an authenticator to make sure that the client's username and password are correct.

The second comment is the client will have which will give him access to the web application via a web browser the available data access is the basic vacation information, but in order to book the vacation, the client will have to log in to the server and be verified.

The server gets all the important data information from the database which holds the user's personal information, on top of that the server is getting the user's credentials the server will checks with the data it gets from the database if they match, only then the client will have access to the server session where he can book with his personal information.

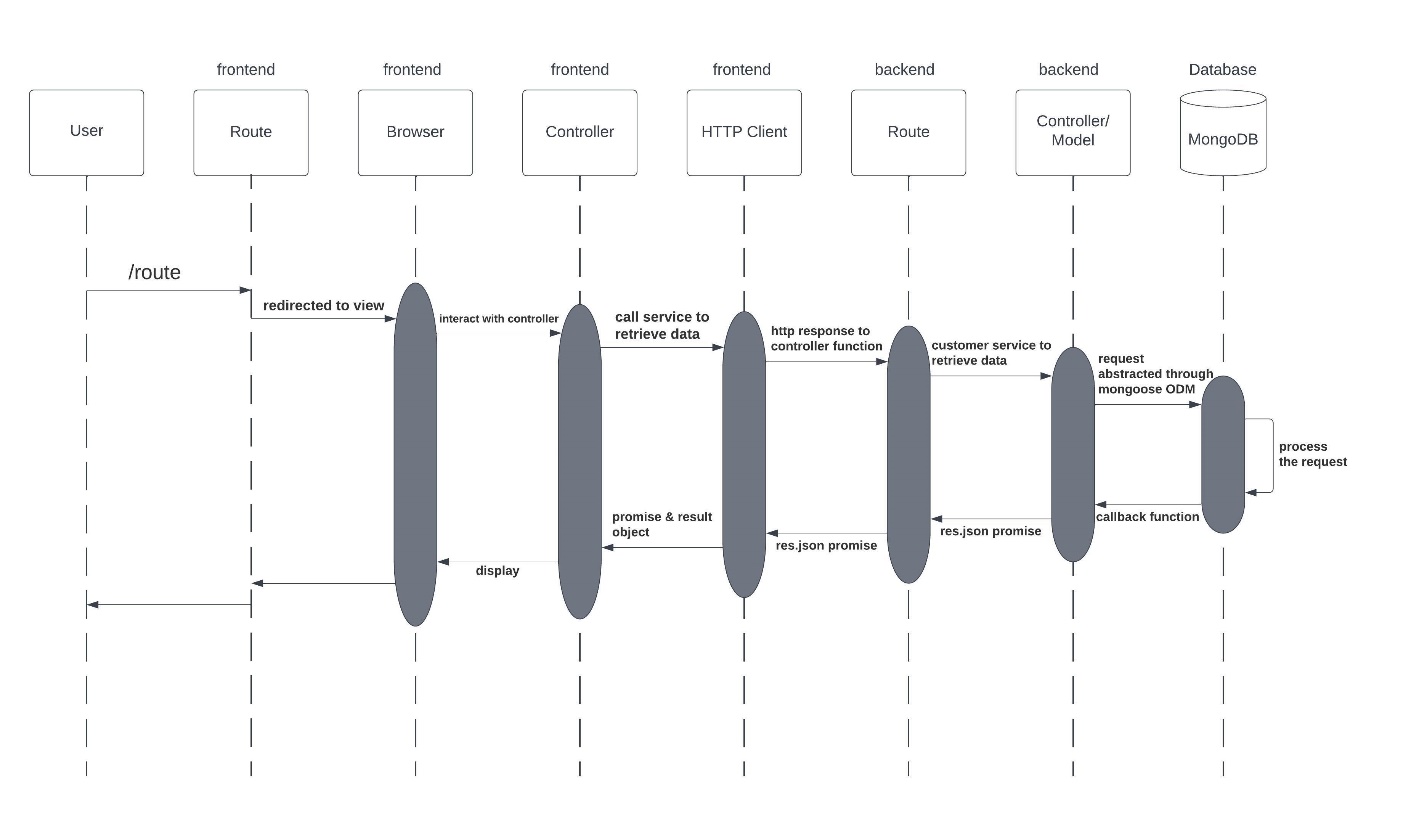
### Component Diagram



A text version of the component diagram is available: [CS 465 Full Stack Component Diagram Text Version](https://learn.snhu.edu/d2l/lor/viewer/view.d2l?ou=6606&loIdentId=24342).

<Describe the overall system architecture of the web application by referring to the component diagram above. Identify the significant components that will be used and their relationships to one another.>

### Sequence Diagram



The user enters the application by using a web browser.

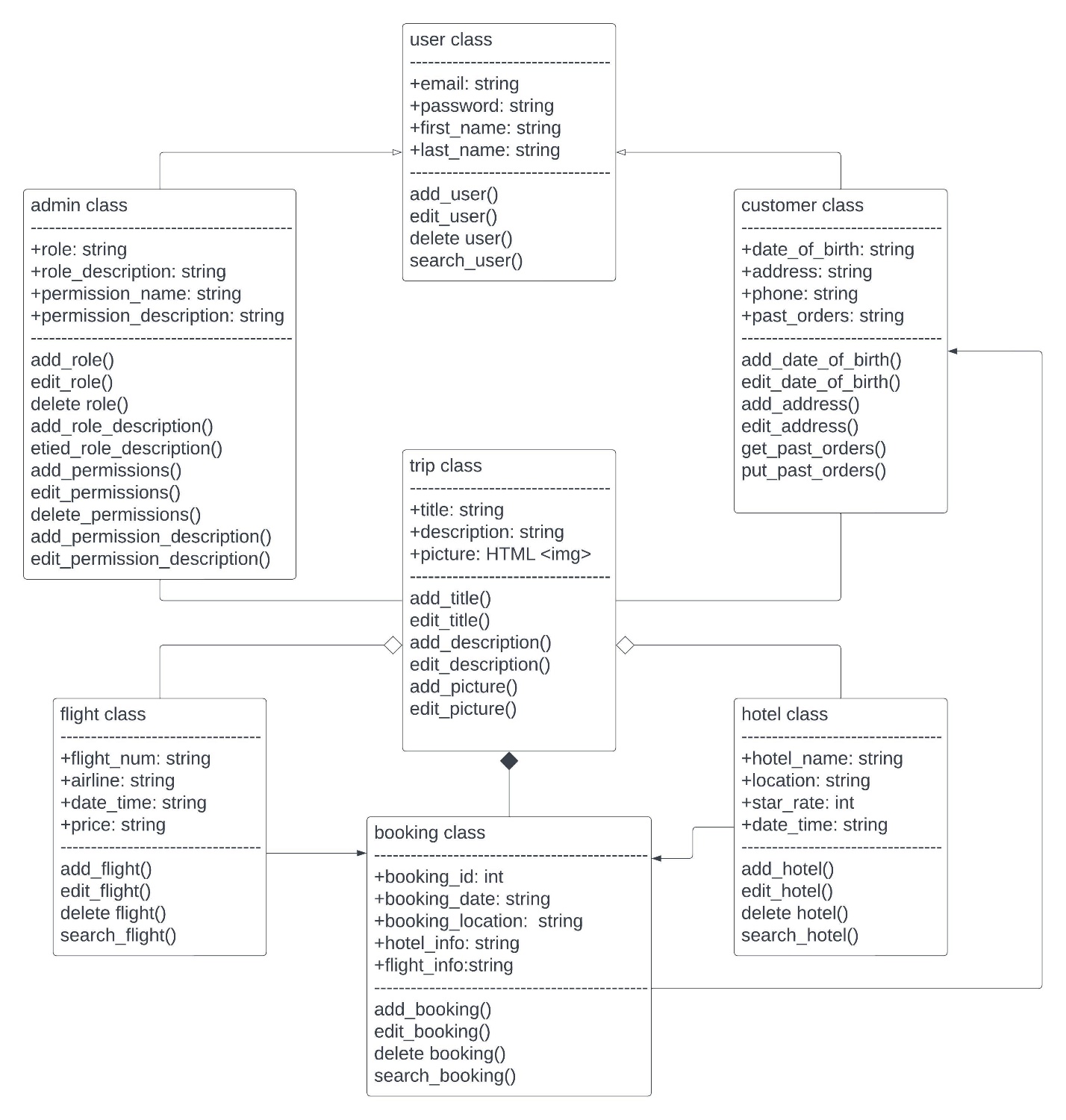
The controller manages the user's request and connects to the HTTP client to send it to the route in the backend and responds to each HTTP request that comes into your application.

In the next step, depending on the route the data is entering the controller/ model is the component that contains all the data logic of the web application, which passes it on to the database to process and verify the data.

After being cleared and verified from the database the data and the request are returned to the controller,

The controller sends back the feedback from the database to the HTTP client and from there it passes the feedback to the controller in the front-end side to display the application in the browser.

## Class Diagram



The master class user which holds all the basic information of all the users in the application which means that it inherits the customers class and the admin class both customers and admins have access to the application, but the customers don’t have the admin's permissions and privileges.

Each customer has his own profile which holds his personal information and even past trips that he acquired with this web application.

From there the customer is able to see access the trip class which contains all the information from possible trips that are listed in the web application to get even deeper information and the ability to check availability we have the hotel and flight classes, in those classes you can see the dates that fit you and the pricing.

Users have the option to choose only hotels, only flights, or a combination of both.

The final step will be to have the desired vacation to the booking class which has all the selected information the users have chosen to confirm and pay.

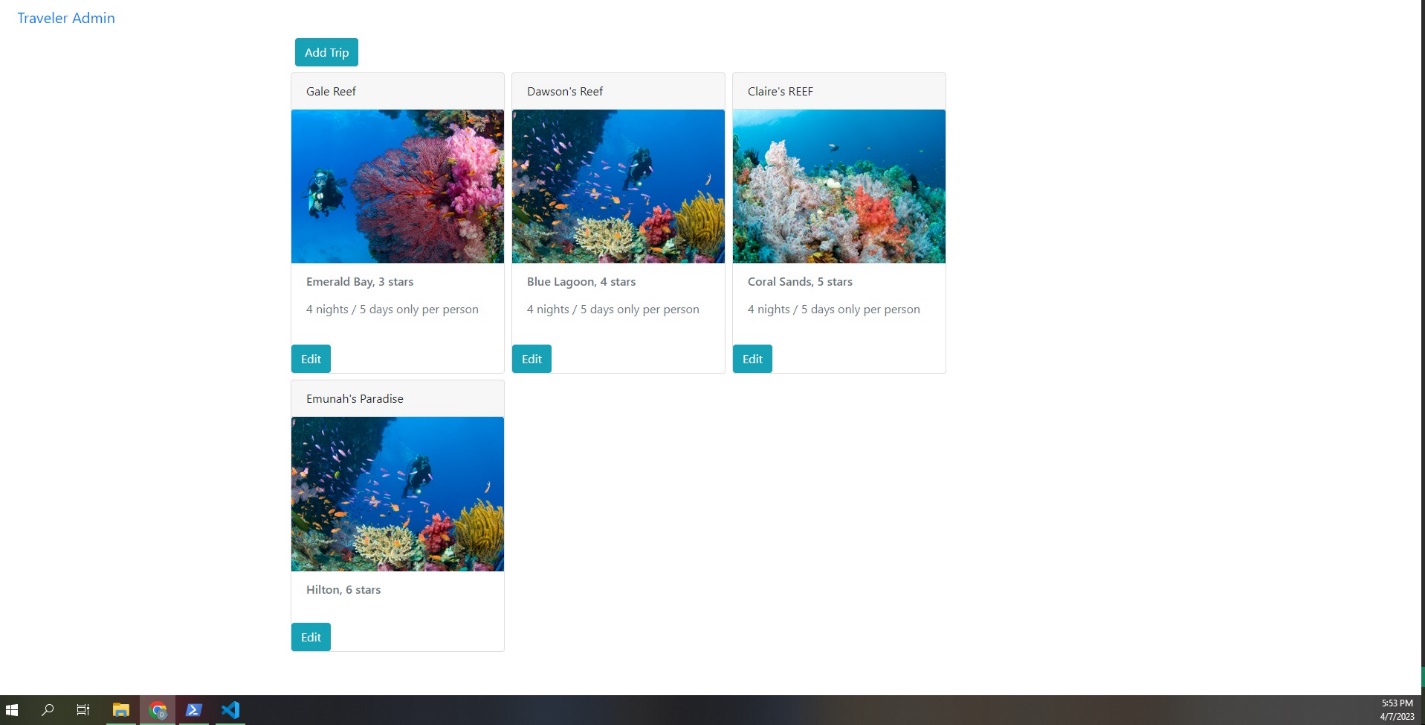
## [API](#_heading=h.2jxsxqh) Endpoints

| **Method** | **Purpose** | **URL** | **Notes** |
| --- | --- | --- | --- |
| .get(tripsController.tripsList); | Retrieve list of all the trips | </trips> | Returns all trips |
| .get(tripsController.tripsFindCode); | Retrieve single trip | </trips/:tripCode'> | Returns single trip by its trip code |

## The User Interface

Graphical user interface, application

Description automatically generatedGraphical user interface, application

Description automatically generated

It is very important to understand the basic structures utilized in this design.

The express and angular structures were chosen based on their purpose.

Express is more minimalistic in its approach, unlike Angular is dynamic, but both have model view controller patterns at their cores.

Express is primarily used for backend purposes, while Angular is known for frontend capabilities.

A single-page application is a website or web application that dynamically rewrites a current web page with new data from the web server, instead of the default method of a web browser loading entire new pages.

SPA advantages are fast load and more flexibility to changes, continuous user experience, and an excellent interactive experience.

SPAs can load more quickly, fetching data in the background, and individual user actions are more responsive since full-page reloads are rare.

SPAs can support incremental updates, saving partially completed forms or documents without the user having to click a button to submit a form.

On top of that SPA offers a richer user interface with lots of features.

This can generate a fluid framework that has an impact on operational efficiency decisions.

There are a lot of ways to test to make sure the SPA is working with the API to GET and PUT data in the database for example I would test the sign-up function which is basically putting data into the database.

The next step will be to get the data from the database by the login function which means the database receives data from the user and the database needs to get the stored data that the user set up in the sign-in prosses for verification.