

<Name of Software Application>

# **CS 465 Project Software Design Document**

Version 1.0

## Table of Contents

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

[Table of Contents 2](#_30j0zll)

[Document Revision History 2](#_1fob9te)

[Instructions 2](#_3znysh7)

[Executive Summary 3](#_2et92p0)

[Design Constraints 3](#_tyjcwt)

[System Architecture View 3](#_3dy6vkm)

[Component Diagram 3](#_1t3h5sf)

[Sequence Diagram 4](#_17dp8vu)

[Class Diagram 4](#_3rdcrjn)

[API Endpoints 4](#_26in1rg)

[The User Interface 4](#_1ksv4uv)

## [Document Revision History](#_44sinio)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 03.20.2022 | <Alexander Mock> | <Last update - Static to Templates with JSON> |

## 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](#_2jxsxqh)

<Describe the appropriate architecture of the web application based on your client’s software requirements. Be sure to reference your use of the MEAN stack for development. Explain both the customer-facing side of the application and the administrator single-page application (SPA).>

Our Clients software requirements are to bring together a website, a database, and an admin single-page application and have them work seamlessly while supporting network traffic.

The Customer facing side of things will be a website that they can navigate through a variety of pages. These pages are Home,Travel, Rooms, Meals, News, About, Contact.

On the customer-facing side, we created HTML documents with JavaScript and JSON to display the data. Using this MVC architecture, multiple pages will be loaded one at a time as the user requested them.

Looking ahead at the admin page we will implement the use of angular. We can preload the pages so it also provides a seamless experience and pages dont have to load and be cached. This method will alow the code to be sitting on the client ready to go.

## [Design Constraints](#_z337ya)

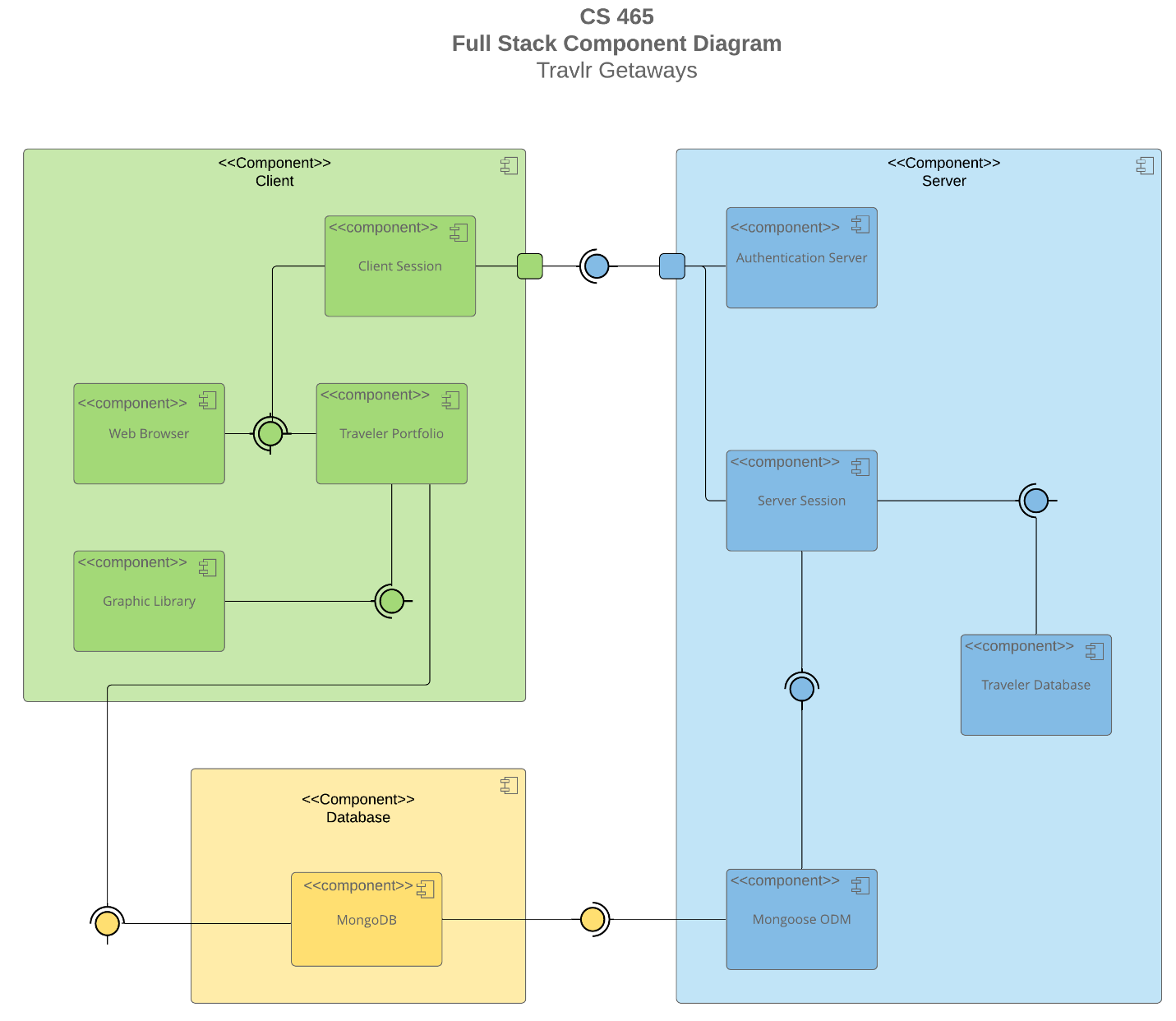
<Identify the design constraints for developing the web-based Travlr Getaways application. Explain the implications of the design constraints on the application development.>

Essentially our constraints come down to making every portion work together. We have to take into account the front end, database, and Admin back end. These things can be tough to manage but when in full stack development we can pick a specific stack to work with that will keep us within those parameters to work together.

In our case we will be utilizing JSON, to display and store data, in conjunction with JavaScript. MongoDB for database storage which uses BSON. And Angular for the admin portion.

## [System Architecture View](#_3j2qqm3)

### 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.>

We can break this down into three sections, Client, Server, Database. The Client itself will have to authenticate with the server and database. The Client itself will house the Session data, Web Browser, Portfolio, and Graphics.

The Server will house Session data, Authentication, Traveler database, and help establish MonoDB to Mongoose.

The MonoDB has relations to the client and the server as the data will have to be stored and communicated from the server to the client.

### Sequence Diagram

<Illustrate the flow of logic in a web application by completing a sequence diagram. Insert an image of the sequence diagram here.>

<Describe the flow of logic in the web application based on the sequence diagram. Be sure to describe the interactions between the layers, or tiers, of the full stack application. It will be helpful to include significant processes such as Sign In, Trips, and Admin interactions when referring to the sequence diagram.>

## Class Diagram

<Illustrate the JavaScript classes of the web application by completing a class diagram for the web application. Insert an image of the class diagram here.>

<Describe the JavaScript classes of the web application based on the class diagram.>

## [API](#_1y810tw) Endpoints

<Exposing RESTful endpoints is a design approach to enable an application to participate in a larger ecosystem. Document each endpoint in the table below, including the HTTP method, purpose, URL, and notes.>

| **Method** | **Purpose** | **URL** | **Notes** |
| --- | --- | --- | --- |
| **GET** | <Retrieve list of things> | </api/things> | <Returns all active things> |
| **GET** | <Retrieve single thing> | </api/things/:thingId> | <Returns single thing instance, identified by the thing ID passed on the request URL> |

## The User Interface

<Insert screenshots from the development of the SPA development to show the following: (1) a unique trip, added by you, (2) the Edit screen, and (3) the Update screen.>

<Summarize the Angular project structure and how it compares to the Express project structure. Be sure to describe the rich functionality provided by the SPA compared to a simple web application interaction. Describe the process of testing to make sure the SPA is working with the API to GET and PUT data in the database.>