

Traveler

# **CS 465 Project Software Design Document**

Version 1.0

## Table of Contents

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

[Table of Contents](#bookmark1) 2

[Document Revision History](#bookmark2) 2

[Instructions](#bookmark3) 2

[Executive Summary](#bookmark4) 3

[Design Constraints](#bookmark5) 3

[System Architecture View](#bookmark6) 3

[Component Diagram](#bookmark7) 3

[Sequence Diagram](#bookmark8) 4

[Class Diagram](#bookmark9) 4

[API Endpoints](#bookmark10) 4

[The User Interface](#bookmark11) 4

## [Document Revision History](#bookmark12)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/15/23 | Nick Schiffman | Module 3 |

## [Executive Summary](#bookmark13)

This web application is built using the MEAN stack (MongoDB, Express, Angular, Node) during development. Angular would be used for the customer-facing side, providing a dynamic and interactive user interface. Express would handle RESTful APIs for communication between the front-end and back-end. Node would serve as the server-side runtime environment, ensuring scalability and high performance. MongoDB would be used as the NoSQL database for efficient data storage for user data. The administrator single-page application would follow a similar architecture, with Angular used for the interface and Express and Node for handling API requests and server-side applications.

## [Design Constraints](#bookmark14)

Design constraints for the Travlr Getaways web application impact development with cross-platform compatibility that requires thorough testing. Performance and scalability involve code optimization and caching(not reading JSON for each request). Security measures to protect user data. Usability needs user research. Integration with APIs requires planning and unknown error handling. Addressing these constraints will ensure a reliable application.

## [System Architecture View](#bookmark15)

### Component Diagram



The web application's overall system architecture comprises several significant components that interact with each other. The Client component serves as the customer-facing side of the application. The Server handles client requests and interacts as the middleman between the Database for the users and company data. The Database component serves as the storage for all company and user data, keeping it well organized for the server’s requests. Together, they form a cohesive system that enables quick and efficient development that enhances the user experience.

### 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](#bookmark16) 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.>