Software Requirements Specification

for

Ohio Transit System

**Version 1.0 approved**

**Prepared by Team 3**

**May 13, 2021**

**Table of Contents**

**Table of Contents [ii](#_3o7alnk)**

**Revision History [ii](#_30j0zll)**

**1. Introduction [1](#_23ckvvd)**

1.1 Purpose [1](#_ihv636)

1.2 Document Conventions [1](#_32hioqz)

1.3 Intended Audience and Reading Suggestions [1](#_1hmsyys)

1.4 Product Scope [1](#_41mghml)

1.5 References [1](#_2grqrue)

**2. Overall Description [2](#_vx1227)**

2.1 Product Perspective [2](#_3fwokq0)

2.2 Product Functions [2](#_1v1yuxt)

2.3 User Classes and Characteristics [2](#_4f1mdlm)

2.4 Operating Environment [2](#_2u6wntf)

2.5 Design and Implementation Constraints [2](#_19c6y18)

2.6 User Documentation [2](#_3tbugp1)

2.7 Assumptions and Dependencies [3](#_28h4qwu)

**3. External Interface Requirements [3](#_nmf14n)**

3.1 User Interfaces [3](#_44sinio)

3.2 Hardware Interfaces [3](#_37m2jsg)

3.3 Software Interfaces [3](#_1mrcu09)

3.4 Communications Interfaces [3](#_46r0co2)

**4. System Features [4](#_2lwamvv)**

4.1 System Feature 1 [4](#_111kx3o)

4.2 System Feature 2 (and so on) [4](#_3l18frh)

**5. Other Nonfunctional Requirements [4](#_206ipza)**

5.1 Performance Requirements [4](#_1ci93xb)

5.2 Safety Requirements [5](#_4k668n3)

5.3 Security Requirements [5](#_2zbgiuw)

5.4 Software Quality Attributes [5](#_1egqt2p)

5.5 Business Rules [5](#_3ygebqi)

**6. Other Requirements [5](#_2dlolyb)**

**Appendix A: Glossary [5](#_sqyw64)**

**Appendix B: Analysis Models [5](#_3cqmetx)**

**Appendix C: To Be Determined List 6**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

*The purpose of this document is to present a detailed description of The Ohio Transit System or “OTS”. This document will explain the product and its features, the interface, what the system will do, and how the system should react to user interactions. This document is intended for both the stakeholders and the developers of said system.*

## Document Conventions

*<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>*

## Intended Audience and Reading Suggestions

*This document is intended for both the stakeholders and the developers of the Ohio Transit System. Stakeholder information is in sections 1 and 2 of this documentation. Developers shall read the entire documentation.*

## Product Scope

The software system will be a customer centric web application in facilitating seamless scheduling between arrivals and departure times in Ohio transit stations with support for multiple devices. The system is designed to be used by locals and visitors of the wonderful state of Ohio. The system will provide users with accurate predictions of their bus arrival and departure time. The buses arrival and departure times are vital to both Bus operation control and passenger information systems. The system also allows for other clients to use. If a client provides GTFS data, the system can be changed to support the new client’s transit system. The Ohio Transit System web application will allow its users to locate their bus, find their desired bus schedules and routes, and pay for the bus fare all in one place. ​

## References

For more information about GTFS data: <https://developers.google.com/transit/gtfs>

# Overall Description

## Product Perspective

*The product is new, self-contained software based on the web and implements a client-server model. The system provides a simple and easy way for users to acquire knowledge on departure and arrival times on routes of Ohio.*

*These are the main features of the system:*

* *Cross platform support: Allows user to access the system through most known devices and commercial operating systems.*
* *User account: The system allows the user to create an account using Google’s Sign-In integration.*
* *Search: It’s possible to search for specific stops based on stop names.*
* *Map/GPS: The user is given a visual of their transportation route, as well as the additional option to pick any two points to see directions (driving, walking, or cycling) for the specified locations.*

## Product Functions

* Schedules
* Trips
* Fares
* Login
* GPS Navigation

## User Classes and Characteristics

*The architectural layout of the application is based on components which they are exported from component to component to use their functional attributes created. Developers shall be comfortable with components and their structural layout.*

## Operating Environment

The software will work on all laptops, desktops, mobile devices, and other devices with access to the internet and web browsers.

## Design and Implementation Constraints

*Developers shall have knowledge of React.js and some of the libraries used along with this programming language such, redux, axios, sass, react-router-dom, materual-UI, mapbox-gl, node and firebase along with google authentication.*

## User Documentation

N/A

## Assumptions and Dependencies

The following are the dependencies used to build the website:

* material-ui
* axios
* firebase
* mapbox-gl@1.13.0
* node-sass
* react-map-gl
* react,js
* react-router-dom
* redux
* express
* mysql
* cors
* google authentication

# External Interface Requirements

## User Interfaces

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## Hardware Interfaces

*The application shall be able to operate in 80% of devices in the market.*

## Software Interfaces

The Ohio Transit System will use axios, express, and cors to connect to a mySQL database containing the GTFS data provided by the client.

## Communications Interfaces

*The application operates data using (async await). HTTP is used in the application as a standard communication.*

# System Features

*4.1 Schedule*

*4.2 Fares*

*4.3 Trips*

*4.4 Login*

*4.5 Map*

## Schedules

4.1.1 Description and Priority

*Provides the user with departure and arrival times for each stop input. The user shall have the option to locate each stop in the map gathering distance and traffic situations on the route.*

4.1.2 Functional Requirements

*The application shall be accessed by devices bigger than mobile devices to use the map feature.*

## Fares

4.2.1 Description and Priority

*Provides the user with fares cost and information about each fare.*

4.2.2 Functional Requirements

*The user shall log in to be able to purchase a bus ride in this section of the application.*

## Trips

4.3.1 Description and Priority

The application provides and estimates arrival time based on the input entered by the user.

4.3.2 Functional Requirements

The user shall be familiar with stop name it stop id. However, this function could be enhanced to provide a better experience to any users not familiar with the area.

## Login

4.4.1 Description and Priority

The user shall have the option to login with google accounts if 1. They have a google account, 2. They do not want to create an account.

4.4.2 Functional Requirements

The user must have a google account to use this functionality

## Map

4.5.1 Description and Priority

The map locates each stop for better user location. However, the map can be used by the user to obtain another location not related to the bus stations.

4.4.2 Functional Requirements

The map shall be used in devices greater than mobile devices for better visibility.

# Other Nonfunctional Requirements

## Performance Requirements

*The application must run in modern browsers and internet faster than 2G.*

## Safety Requirements

*The user shall use a trusted browser to provide extra layer of protection. However, the application shall have SSL certification to verify that it is a trusted website.*

## Security Requirements

*The application shall be deployed from a trusted and secure server to protect the integrity of the code.*

## Software Quality Attributes

*The application can be upgraded with little to nothing down time based on the architectural component structure. The application would be easy to study to any developer from the standpoint of knowledge about the programming language.*

## Business Rules

*The application shall be able to process payments for bus rides to users.*

# Other Requirements

*The application cannot process payments outside of the United Sates of America. However, the page can be accessed from anywhere in the world.*

**Appendix A: Glossary**

*<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>*

**Appendix B: Analysis Models**

*<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams*.>

**Appendix C: To Be Determined List**

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*