**Software Requirements**

**Specification**

**For**

**Bus Tracking System**

.

**INSTRUCTORS: Dr. Pranesh Das**

**COURSE:CS3002D - DATABASE MANAGEMENT SYSTEMS**

**Contents**

1. **INTRODUCTION**

Document purpose

Product scope

Intended audience and Document overview

Definitions, acronyms and Abbreviations

Document Conventions

References and Acknowledgment

1. **OVERALL DESCRIPTION**

Product Overview

Product Functionality

DESIGN AND IMPLEMENTATION CONSTRAINTS

ASSUMPTIONS AND DEPENDENCIES

1. **SPECIFIC REQUIREMENTS**

EXTERNAL INTERFACE REQUIREMENTS

FUNCTIONAL REQUIREMENTS

USE CASE MODEL

1. **OTHER NON-FUNCTIONAL REQUIREMENTS**

PERFORMANCE REQUIREMENTS

SAFETY AND SECURITY REQUIREMENTS

Software Quality Attributes

1. **OTHER REQUIREMENTS**

BUSINESS RULES

OTHER EXPECTED REQUIREMENTS

**1 Introduction**

# Document Purpose

This document is a **Software Requirements Specifications (SRS)** for the bus tracking app**.** It lays out the functional, non-functional and behavioral requirements. The purpose of this source is to describe the bus location which also provides the bus timing details. The SRS is organized into several sections to help and assist the development of the system in the user perspective. This enables the users in the long run with ease in terms of time management.The SRS will be periodically updated based on the incorporation of new features and the feedback received from the users.

**1.2 Product scope**

A bus tracking system is an application intended to convey the users about various aspects like number of buses available, timings of the buses and other important aspects which could enable the growth of bus management

1. This project involves information about the number of buses between two particular bus-stops.
2. This enables the users to know the exact time reaches the particular stop.
3. Users can view the bus details and can easily locate the bus.

# Intended Audience and Document Overview

This SRS is intended for several people including the citizens of the particular city, people looking for the buses as well as the system design developers.

1. The user can check the buses availability in a particular route between two bus-stops and also they can select the desired bus they wish to travel according to their convenience.
2. The developers can use the SRS to design the system in such a way that it meets the requirements of the client.

# The users can give the feedback of their experience with the facilities provided in the bus tracking system application.

# 1.4 Definitions, Acronyms and Abbreviations

BTS: Bus tracking system

*SRS: Software Requirements System*

*ER: Entity Relationship diagram*

# Document Conventions

Formatting Conventions are followed:

ITALICS

BOLD

Headings & Subheadings

Font color

Font size

Line

Spacing

Header and footer

# 1.6 References and Acknowledgments

<https://ieeexplore.ieee.org/document/278253> - IEEE Guide for Software Requirements Specifications

**2 Overall Description**

# Product Overview

This software is an android based application and it is not a component of any other program and is basically intended for the BTS. Our application’s data is stored in MySQL Server which can be accessed by the developer and can be retrieved in the required format by the user.

# Product Functionality

* Provides a secure database of the bus management which register in the management system and containing the login credentials and personal information.
* Facilitates a process of entering, retrieving, modifying and deleting the data of both the ends.
* Provide necessary access to the users with simple interface documentation.
* Provide a list of buses that are connecting cities and their services.
* Provides a query section to the developers such that users can contact the system developers any time
* The system also provides features like the availability of buses, distance measurement between source and destination, journey time.
* Users can find the bus details by entering the bus number or from\_location and to\_location details.
* The system tells if the specialized buses corresponding to two cities are available or not.

# Design and Implementation Constraints

* Users are given access to the website so that they can register into the application by entering their details.
* **Security:** Data stored in database can be accessed by unauthorized person because of using access database which is weak and it cannot prevent unauthorized people from accessing it. This will lead to information stored in their database being interfered and damaged
* Implementation of the database using a centralized database system.
* The challenges in developing the system are the required number of users. The expected number of users once launched will be around 5000 at the first point of implementation

# Assumptions and Dependencies

* + 1. The System assumes that the users who register have mobile phone access to them every time so that they can avail the features of the system.
    2. The System depends on the bus number or from\_location and to\_location.
    3. The System is not required to save generated reports.

**3.Specific Requirements**

# This section contains all the software requirements in a more detailed manner, that when combined with the system context diagram, use cases and use case descriptions is sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies all the requirements.

# External Interface Requirements

* + 1. **System Interface**

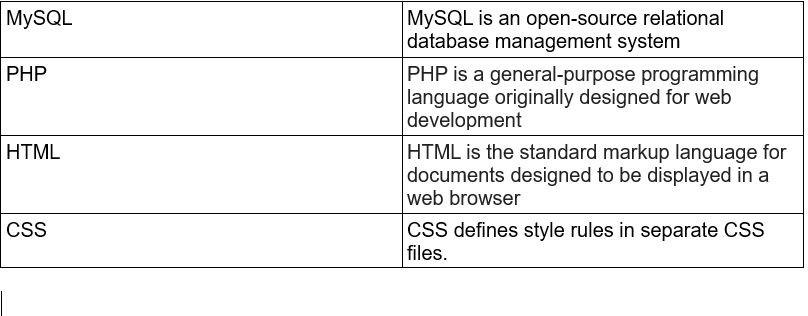
The User Inputs data via the web server using HTML forms. The actual program that performs the operation will be written in PHP.

### **3.1.2 User Interfaces**

The new system shall provide a very intuitive and simple interface to the user and the administrator, so that the user can easily navigate through and find the best bus.

#### **Software Interfaces**

* Front end development requires react native
* Back end Interfacing using javascript
* Database Setup using SQLITE
  + 1. **Hardware Interfaces**



# Functional Requirements

* + 1. **Use Case Scenario**

# Use Case Scenario 1- user login

|  |  |
| --- | --- |
| **Purpose** | User logs in to the system using email and password |
| **User** | A user can create a profile with a unique email and password. |
| **Input Data** | Profile email and password. |
| **Output Data** | Corresponding page data. |
| **Invariants** | Profile table data and user information. |
| **Pre-conditions** | User is not logged in to a profile, input profile exists in database, user password matches profile. |
| **Post-conditions** | User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile |
| **Basic Flow:** | Webpage looks up profile data and returns the matching cookie. Web Page is updated to match new user data. |
| **Alternative Flow(s):** | Invalid password, invalid username, or mismatched username and password redirect to error message and previous page. |

* + - 1. **Use Case Scenario 2 –Buses available from the source and destination**

A user logs into the system and can check the buses scheduled and from the given source and destination.

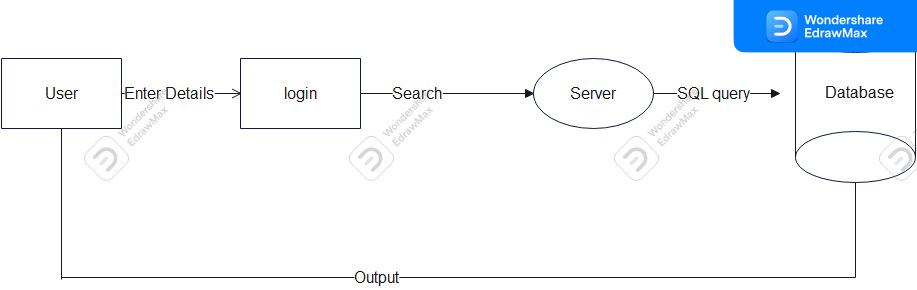
**To check the scheduled buses:**

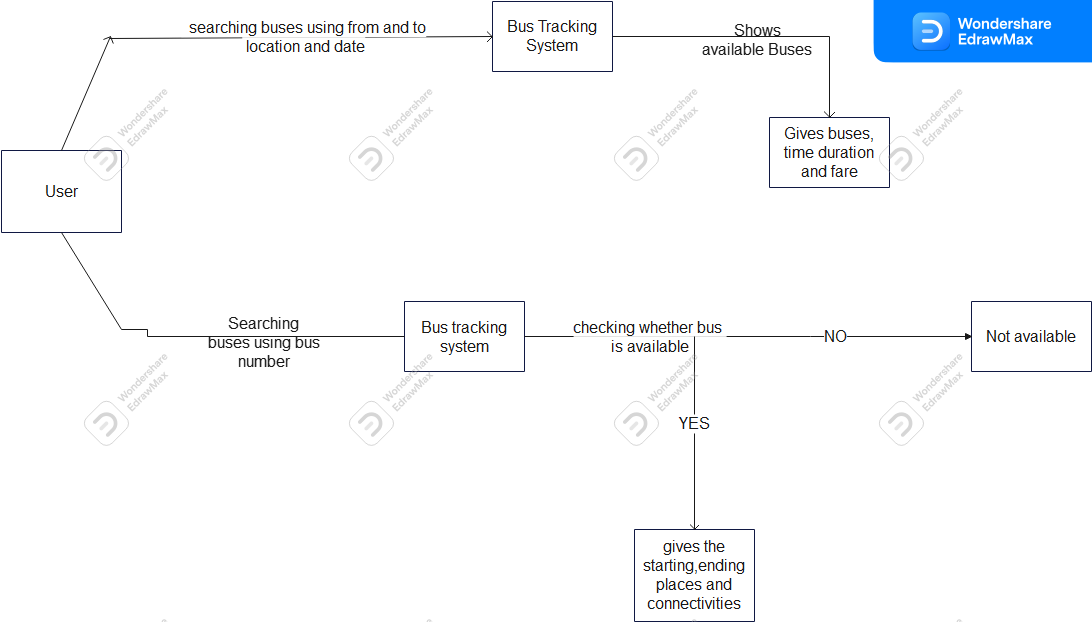
|  |  |
| --- | --- |
| **Purpose** | A user wants to check the buses and vacancy availability in them. |
| **User** | Any user can check the bus details. |
| **Input Data** | Choose source and the destination. |
| **Output Data** | Buses are displayed in the ascending order of the scheduled time, location and review of the bus will also be displayed. |
| **Invariants** | Permission for location access has to be enabled by the user. |
| **Pre-conditions** | User is not compulsory to Logged in. |
| **Post-conditions** | Users could be able to get the list of buses. |
| **Basic Flow:** | After that buses in the ascending order of the time reached from the source will be displayed on the screen. Among them, the user has to choose one to display the schedule of that bus |

* + - 1. **Use Case Scenario 5 –Ratings and reviews for the train journey**

A user can give the review and ratings regarding the treatment in the bus.

|  |  |
| --- | --- |
| **Purpose** | To give review or rating for a bus based on experience of journey. |
| **User** | A legitimate user logged into the system |
| **Input Data** | The bus number and from\_location,to\_location |
| **Output Data** | a message saying “Thank you for your response” after finishing the review. |
| **Invariants** | NIL |
| **Pre-conditions** | User is logged in. |
| **Post-conditions** | NIL |
| **Basic Flow** | The registered user should login into the system and should input the bus number in which he/she took travel and should give the review based on the experience and the result and then there comes a message entering “Thank you for your response” after finishing the review. This could be useful for the other users to check how the treatment and the travel is. |





**4.Other Non-functional Requirements**

# Performance Requirements

* The end product is web based and needs a running server to host it.
* The request response time will depend on the latency of the server and the network speed.
* So, an adequate server capable of handling concurrent requests of 750-1000 users at a time is required.
* The run time of the application also depends on the hardware of the server hosted on.

# Safety and Security Requirements

* The data in the server has to be backed up periodically.
* Secure processing and storage of sensitive information like login credentials and personal information.
* Users are classified on various authorization levels based on which they can access, retrieve and modify data in the database.
* Masked password entry will be provided.
* The people who avail the basic functionalities like viewing nearest buses, available vacancies and the pay have the access of read-only display.

# 4.3 Software Quality Attributes

* Reliability
* Operability
* Performance efficiency
* Security
* Compatibility
* Maintainability

### **4.3.1 Reliability**

* Basically any UPDATE is reflected in the database in minimal time.
* The data of the bus can be fetched at any point by the users.

### **Operability**

* The user can easily get accustomed to the developed interface.
* The created web application will ensure smooth user experience and a designated help section will be provided for standard FAQs.
* Appropriate error messages are shown up when errors are encountered.

### **Performance Efficiency**

* An attempt to incorporate the best website and database practices like normalization to ensure faster request response times.
* Since login, user access is real time, an attempt will be made to perform these operations in the least possible time, so that the emergency cases can make the best use of it.

### **4.3.4 Security**

* Users are requested to create a unique user id (error message will be popped up if they enter a username that already presents in the database).
* The web browser will not display the password while entering in the field instead special characters will be displayed to assist the user for example ‘\*’.
* The system’s backend will be accessible to only authorized levels of users i.e the main overlooking system of buses.

### **Maintainability**

* Our SQL Server has a database maintenance routine scheduled every week/ or whenever there’s an issue.
* Periodic testing and bug fixes will be rolled out based on the feedback from the users.

**5 Other Requirements**

* 1. **Business Rules**
* All the registered users are expected to have a unique user id which can be made using mobile number.
* Requires proper input for the fields during registration for an event which will be validated by the already pre-existing data we have
  1. **Other Expected Requirements**
* All the users are expected to be familiar with basic web browsing and web interfaces.
* The users are also expected to raise concerns about the existing bugs and issues so that they can be rectified.
* The language of the entire web interface will be in English. So basic English is assumed to be known by the corresponding users.