Software Requirements Specification

for

Bus Management System

Prepared By:

1	**Varun Parashar**	**1701074**
2	**Shivam Anand**	**1701062**
3	**Swapnil Raj**	**1701072**
4	**Rushikesh Shimpi**	**1701055**
5	**Subhash Kumar Suman**	**1701067**

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Indian Institute of Information Technology Guwahati

23.01.2020

Copyright © 1999 by Karl E. Wiegers. Permission is granted to use, modify, and distribute this document.

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Bus Adda	15.01.2020	New	1.0
Bus Adda 1.1	23.01.2020	Update in limitations and app features	1.1

1. Introduction

1.1 Purpose

The Main aim of Bus Management project is to ease the process of ticket booking and keeping records of buses and the drivers or the conductors. We aim to demonstrate the use of create, read, update and delete MySQL operations through this project. This project starts by adding stations then adding details of employee either a driver or a conductor. Once, registration is done user can add bus details with the route. Now, the user can use the ticket counter to book tickets for the passengers.

1.2 Intended Audience and Reading Suggestions

Who are unable to buy bus ticket online at this moment would have to go to the counter to a buy bus ticket. Sometimes, customers' needs to queue up a long queue to buy bus ticket and ask for information and this brings a lot of inconveniences to customers. Online Bus Ticket Reservation System enables the customer to buy bus ticket, make payment, and ask for information online easily.

1.3 Product Scope

The main purpose of this study is to automate the manual procedures of reserving a bus ticket for any journey .This system is said to be an automatic system and customers can select seats by themselves. Specifically, objectives of this project will consist of:

i) Providing a web-based bus ticket reservation function where a customer can buy bus ticket through the online system without a need to queue up at the counter to purchase a bus ticket.

ii) Enabling customers to check the availability and types of buses online. Customer can check the time departure for every bus through the system.

iii) Easing bus ticket payment by obtaining a bank pin after payments is made to our designated banks.

iv) Ability of customers to cancel their reservation.

v) There will be reserved seats for senior citizens.

vi) There will be a panic button in the app which will trigger our alarm system, for case of emergency.

vii) Admin user privileges in updating and canceling payment, route and vehicle records.

viii) A driver or a conductor can be added by a user with admin privileges. On the other hand, a user with roles such as Driver or conductor will login with their user-names and password to find about their current journey.

1.4 References

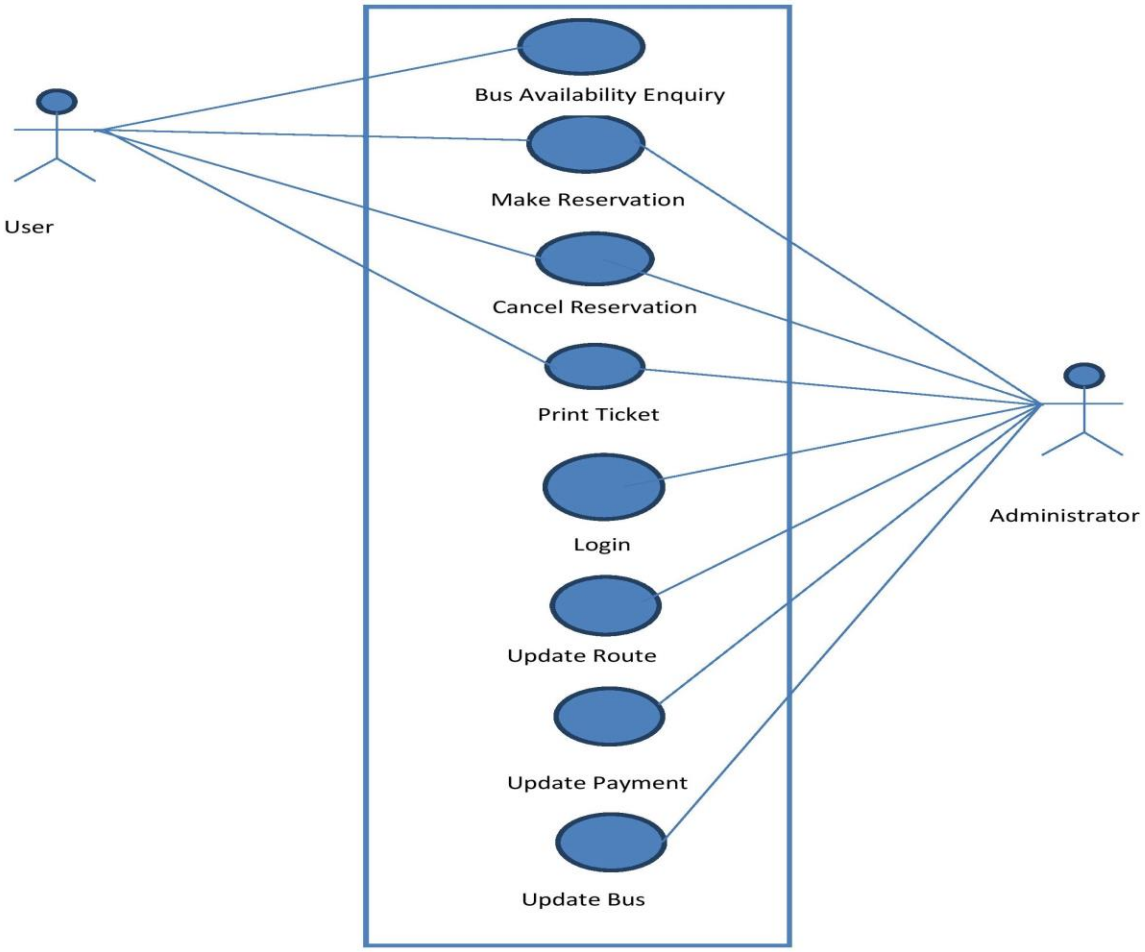
IEEE SRS Format**.**

2. Overall Description

2.1 Product Perspective

This project is a tool to help in managing bus services. It is more useful in current market situation where an organization is not close to a door or a city or a nation. In this case user can book ticket from one corner of the world to another by using internet makes our work for easy.

2.2 Product Function



2.3 User Classes and Characteristics

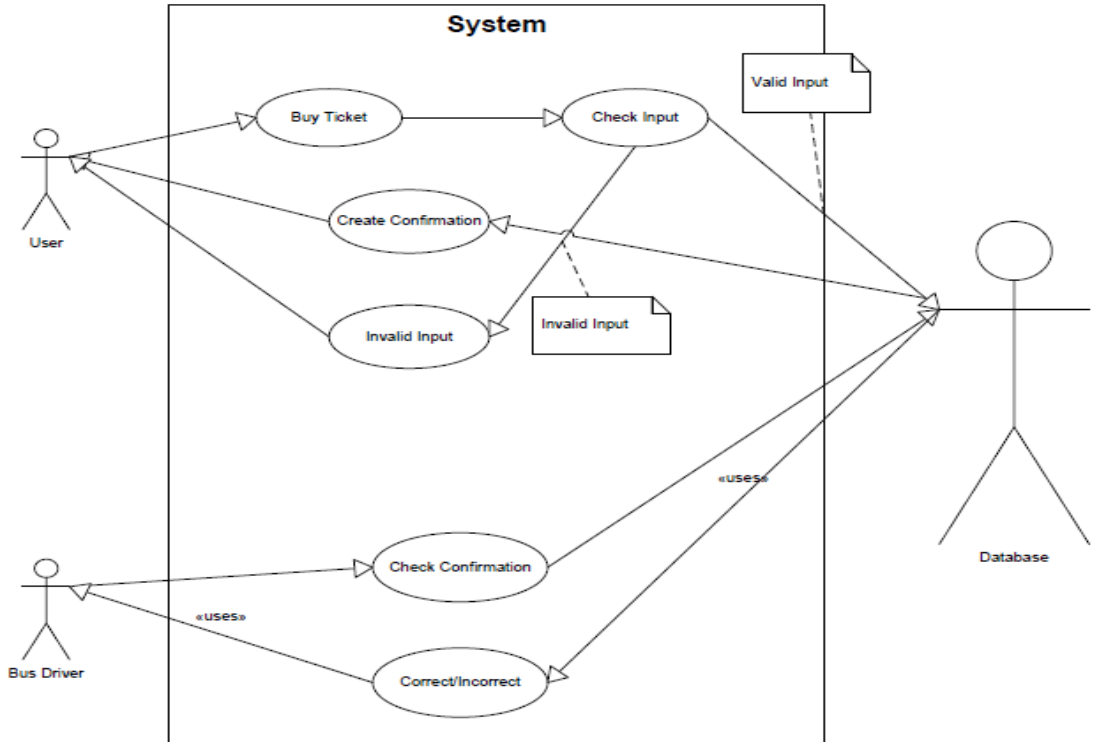
Admin:
This class of users can make reservation, cancel reservation, print ticket of any user ,remove existing user from database ,update Route, update payment mode, add and remove buses.
Customer:
This class of users can reserve ticket for journey by buying ticket or cancel an already booked ticket and can print ticket.

2.4 Operating Environment

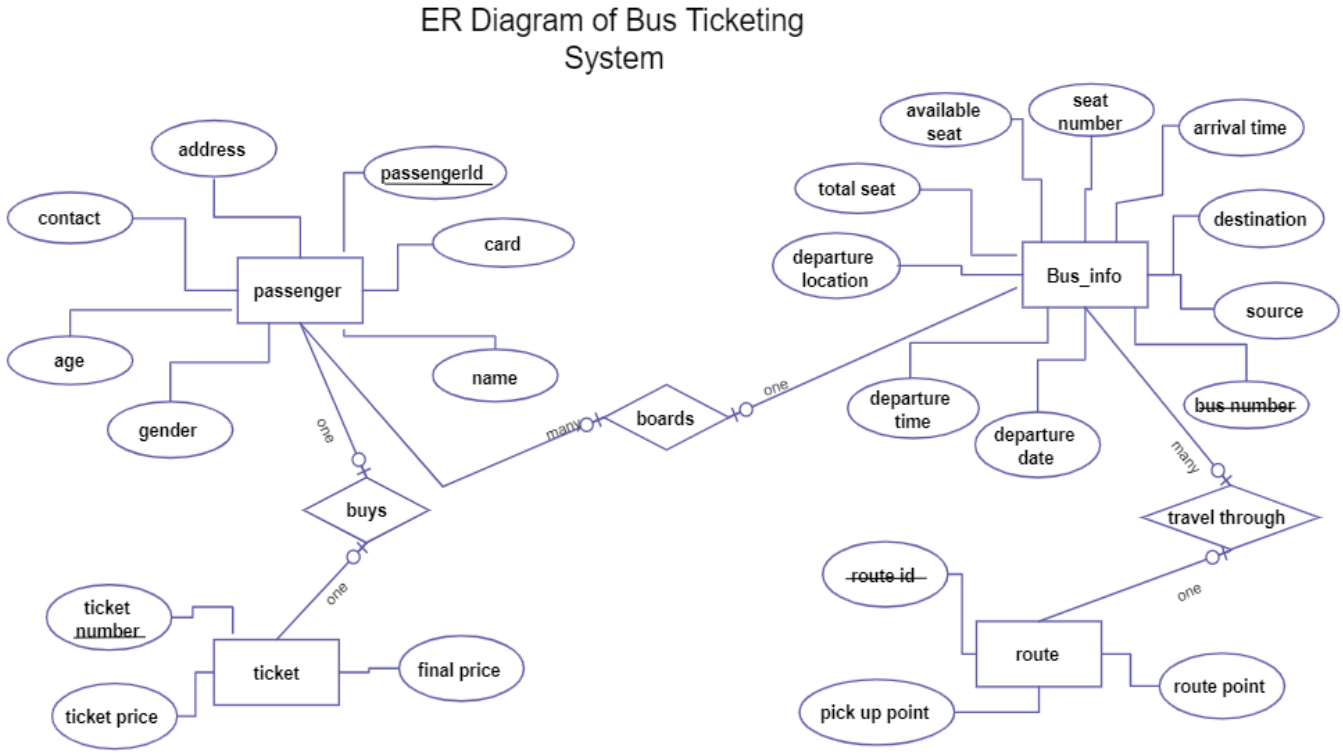
Operating environment for the bus management system is as listed below.
I. client/server system
II. Operating system: Windows.
III. database: SQL database (MySql)
IV. platform: Python /Django(Web Frameworks) /Java /PHP

2.5 Design and Implementation Constraints

I. SQL commands for above queries/applications
II. How the response for application 1 and 2 will be generated. Assuming these are global queries. Explain how various fragments will be combined to do so.



III. Implement the database at least using a centralized database management system.



2.6 Assumptions and Dependencies

Let us assume that this is a distributed bus management system and it is used in the following application:

I. A request for booking/cancellation of a bus from any source to any destination, giving connected buses in case no direct bus between the specified Source-Destination pair exist.

II. Calculation of high travellers (most frequent travellers) and calculating appropriate reward points for these travellers.

3. External Interface Requirements

3.1 User Interfaces

Screen Name	Description
Login	Log in into the system as a CSR or Manager
Reservation	Retrieve button, update/save reservation, cancel reservation, modify reservation, change reservation, adjust seat number,accept payment type
Travel	Modify date to travel, place of destination
Bus Payment	Accept Payment
Passenger Record	Add or update passenger records
Administration seats	Availability
Administration User	Create, modify, and delete users; change password
Reports	Select view, add, save

3.2 Hardware Interfaces

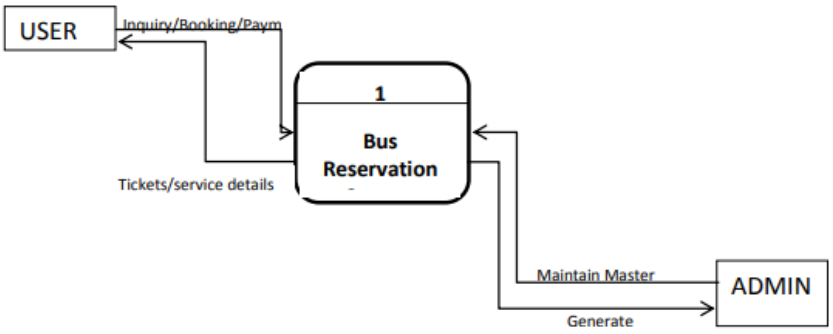
I. A smart devide with browser which support CGI ,HTML and Java-script.

3.3 Software Interfaces

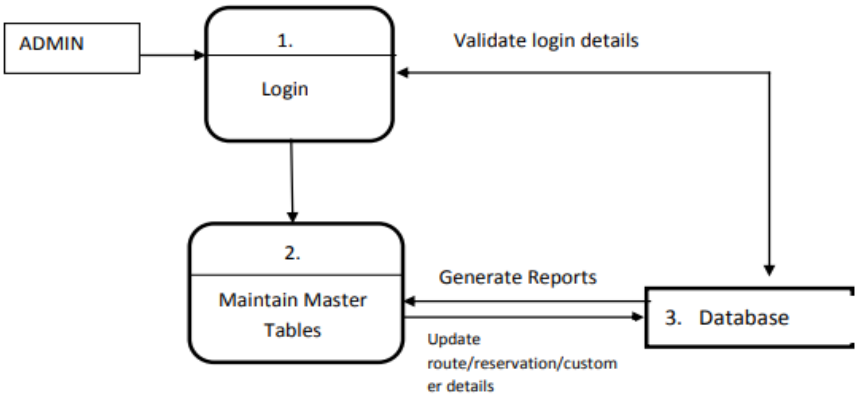
The system shall interface with access database.

3.4 Communications Interfaces

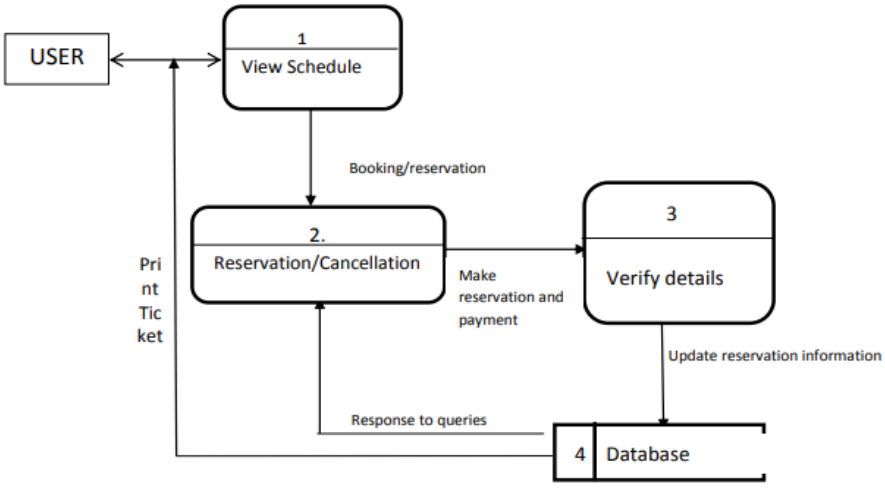
Level 0



LEVEL 1



Level 2



System Features

3.5 System Feature

3.5.1 Stimulus/Response Sequences

I. Web application interfaces with user

Using input, users select a text field, the OS provides data entry methods

Inputs:

First Name - String, 30 characters max, not case sensitive

Last Name - String, 30 characters max, not case sensitive

Email - String, 30 characters max, not case sensitive, requires exactly 1 "@" sign and at least 1 "." symbol

Ticket Type - Drop down menu with "Single Use," "1 Day Pass," "3 Day Pass,"Week Pass." Defaults to "Single Use"

Credit card type - Drop down menu with "Visa," "Discover," "Master-card," "American Express." Defaults to "Select type."

Credit card number - String, Exactly 16 numeric characters

Expiration date month - String, Exactly 2 numeric characters

Expiration date year - String, Exactly 4 numeric characters

II. Web application interfaces with database

When sending output "data," database returns 6 digit string "confirmation code"

Input:

Confirmation Code - String, 6 digit numeric code assigned by database as a key when receiving successful "data" input from Phone App. Sent back to phone application to display on screen

Using touch input, users touch an image button "Purchase", "Data" output is only sent once

Outputs:

Data - String, all text fields entered as inputs concatenated together, separated by semicolons

Database interfaces with Web application

database receives "data" string

Input:

Data - String, all text fields entered on phone concatenated together, separated by semicolons database sends "confirmation code" string back to Web application

Output:

Confirmation Code - String, 6 digit numeric code assigned by database as a key when receiving successful "data" input from Web App.

III. Bus application interfaces with user

Using keyboard, users enter "confirmation code" provided upon successful purchase from Web app

Input:

Confirmation Code - String, 6 digit numeric code assigned by database as a key when receiving successful "data" input from Web App.

IV. Bus application interfaces with database

When sending output "code," database will return a Boolean indicating successful removal of given entry from database

Input:

Success - Boolean, returned from database, if "True" then display success message to user. If "False" display fail message to user.

Using mouse, users click an image button "Sign In", "Code" output is only sent once

Output:

Code - String, 6 digit numeric code, exact copy of "Confirmation Code." Sent to database to verify existence of user information

Database interfaces with Bus application

database receives "code" string

Input:

Code - String, 6 digit numeric code, used to check if string already exists in database as a key. When found, that entry in database is removed, and output "success" is returned.

database sends "success" Boolean back to bus application

Output:

Success - Boolean, "true" if entry specified by "code" was found and successfully removed. "False" if entry specified by "code" was not found. Database is unchanged if "Success" is false

V. **Extra Features Of The Bus Management System**

Any empty seat in the middle of the journey (would probably happen due to the user getting off the bus due to the arrival of his destination) would be synchronized as needed.(i.e the passenger that got off at a destination would leave a seat empty and thus another passenger with that place as his boarding point would get that seat.)

3.5.2 Functional Requirements

I. Open GUI on phone

- ✓ create window with all text fields and images and buttons
- ✓ Submit button is pushed

test validity of all user entered fields

verify all fields are entered

verify proper formatting for fields requiring formatting

display output to screen about outcome of verification

Send data to database

connect to database

consolidate all text fields into a string

send string to database

parse string to database fields

send confirmation back to application

display on screen success animation

II. Open Check In App on bus

✓ create window with text field and button

✓ Submit button is pushed

test validity of user entered field

verify field is entered

verify proper formatting for field

display output to screen about outcome of verification

Send data to database

connect to database

search for code entered by user

remove user from database if entry is "Single Use Pass"

send confirmation back to application

display on screen success animation

4. Other Nonfunctional Requirements

4.1 Performance Requirements

I. The system must be interactive and the delays involved must be less . ...

II. Information transmission should be securely transmitted to server without any changes in information.

III. Reliability.

4.2 Data Requirement

All data will be stored as Strings in the database. The only calculations that will be performed is the search implemented to check the "code" with all the entry Keys in the database. The only other methods that are run are the add and remove methods for entries in the database.

4.3 Security Requirements

Security is used to ensure credit card information as confidential and is not stored anywhere. The only location any user data is visible is when users enter in their information. System Administers are trained on customer privacy.

4.4 Safety Requirement

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure**.**

4.5 Software Quality Attributes

I. After each revision the system will be analyzed and compared with the system requirements designated by the client. We will than try to break the program with test cases.

II. We will all be responsible for quality assurance. We will cross-analyze the system implementation with the System Requirements to verify system accuracy.

III. All final documents will be reviewed for spelling errors, broken links, and any other issues that affect the quality of the document. Any errors found in either the system or supporting documents will be fixed at the time errors are found.

Appendix A: Data Dictionary

This SRS document is used to give details regarding Online Bus reservation System. In this all the functional and non-functional requirements are specified in-order to get a clear cut idea to develop a project.

Appendix B:Group Log

A group of five people was made to prepare the project for Software Engineering. A sub topic was decided upon as Online Bus Reservation System. The topic and the group members' names were then given to the Instructor. All the four members then read the example srs document provided and searched for the requisite document for system. The Problem description, the purpose and the scope of the document was finalized.

Appendix C: Limitations

I. A user must be a registered before booking a ticket.

II. An un-registered user can only see our web app in view mode. None of the changes made by them will be updated in our database.

Appendix D: Constraints

I. A user can only be access our web app using internet services.

II. Database requirements will be limited.

Appendix D: Further scope

I. A user can book ticket by just using a feature phone.