**High Level Design - Bus Management System**

Table of Contents

[1.](#_heading=h.gjdgxs) Revision History 2

[2.](#_heading=h.30j0zll) Approval 2

[3.](#_heading=h.1fob9te) Target Audience/Distribution List 2

[4.](#_heading=h.3znysh7) Reference Documents 2

[5.](#_heading=h.2et92p0) Introduction 2

[6.](#_heading=h.tyjcwt) Requirement 2

[7.](#_heading=h.3dy6vkm) Purpose 3

[8.](#_heading=h.1t3h5sf) Scope 3

[9.](#_heading=h.4d34og8) Exclusions 3

[10.](#_heading=h.2s8eyo1) Architecture 3

[11.](#_heading=h.17dp8vu) ER Diagram 3

[12.](#_heading=h.3rdcrjn) Use case 3

[13.](#_heading=h.26in1rg) Constraints 3

[14.](#_heading=h.lnxbz9) Technologies used 3

[15.](#_heading=h.35nkun2) Tools and Software required to develop 3

[16.](#_heading=h.1ksv4uv) S/w and H/w sizing 3

[17.](#_heading=h.44sinio) Acronyms and Abbreviations 3

# Revision History

| Revision Number | Change By | Changes Incorporated |
| --- | --- | --- |
| PA1 |  | Initial Draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Approval

| Approved By | Designation | Approval Type |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Target Audience/Distribution List

| Name | Designation |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Reference Documents

| Document Name | Location  Physical Location/Website |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Introduction

This High-Level Design Document is intended to provide a high-level overview of the project and establish a shared agreement between the customer and the solution provider to complete a design architecture so that there are no misunderstandings or design gaps during the project's development phase. The Bus Management System is a web-based application that may be used by anyone with an internet connection anywhere on the internet. This application will automate the booking of tickets as well as inquiries about ticket availability. The system will keep track of numerous routes and the buses that run along with them. The system will be built as a web-based application with the main user interface. The primary screen should have a consistent and adaptable format. The system must be built to be user-friendly. Revenue reports can be generated by the admin, which will be useful in determining whether any operational changes are required.

# Requirement

Buses have been a widely used means of transportation since their inception. The current system isn't fully automated. Many procedures, such as bus scheduling and ticket booking, are performed manually, which results in inaccurate entries and leaves room for errors since the data is out of sync. Seat availability is not centrally managed, and the travel operator is unaware of the availability and occupancy of his buses' seats. The main bottleneck is here. As a result, in this modern era, where people prefer to purchase tickets online, a management system that allows passengers to book tickets for long bus travels without bother becomes important.

Therefore, a managed system is required which can overcome the difficulties and faults in the old queue management system.

# Purpose

A proper bus management system provides the below advantages which are mostly not possible for any unstructured and file-based legacy systems:

* Removes data duplication
* Avoid data loss
* Ease the process of ticket booking
* Keeping records of buses and routes in one place
* Determining the performance of routes over a certain period of time.
* Provides data authenticity.
* Hide sensitive data.
* Improve Profitability

# Scope

**Bus Management System** must have the facility to do the following:

* commuters can book a bus ticket from starting point to the endpoint for a day
* there is predefined bus information in the system (code, regn-number, route code, start, end, facilities, seat count, fare\_per\_km)
* route code, start and end must be coming from route table
* there is a predefined route table (route code, stopno, stop name, dist\_km)
* booking information for a bus can be retrieved for any day
* during booking, commuters can see the total fare depending on fare\_per\_km and dist\_km
* no waitlist ticket will be issued
* commuters can book at max 6 seats at a time
* after booking a PNR (unique 10digit alphanumeric data) will be generated for each ticket
* commuters can cancel its ticket as a whole, no partial cancellation
* Revenue reports can be generated per route.

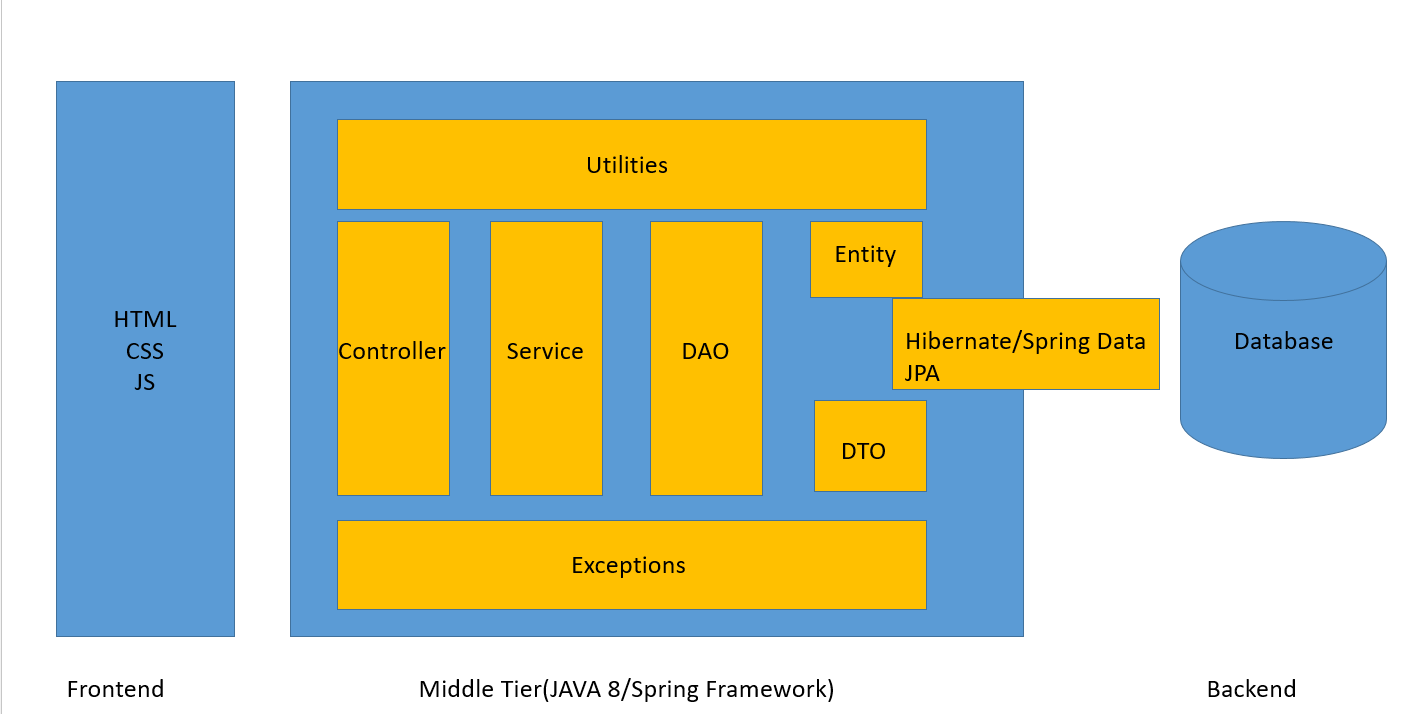
# Exclusions

This system will not consider the below functionalities as part of development:

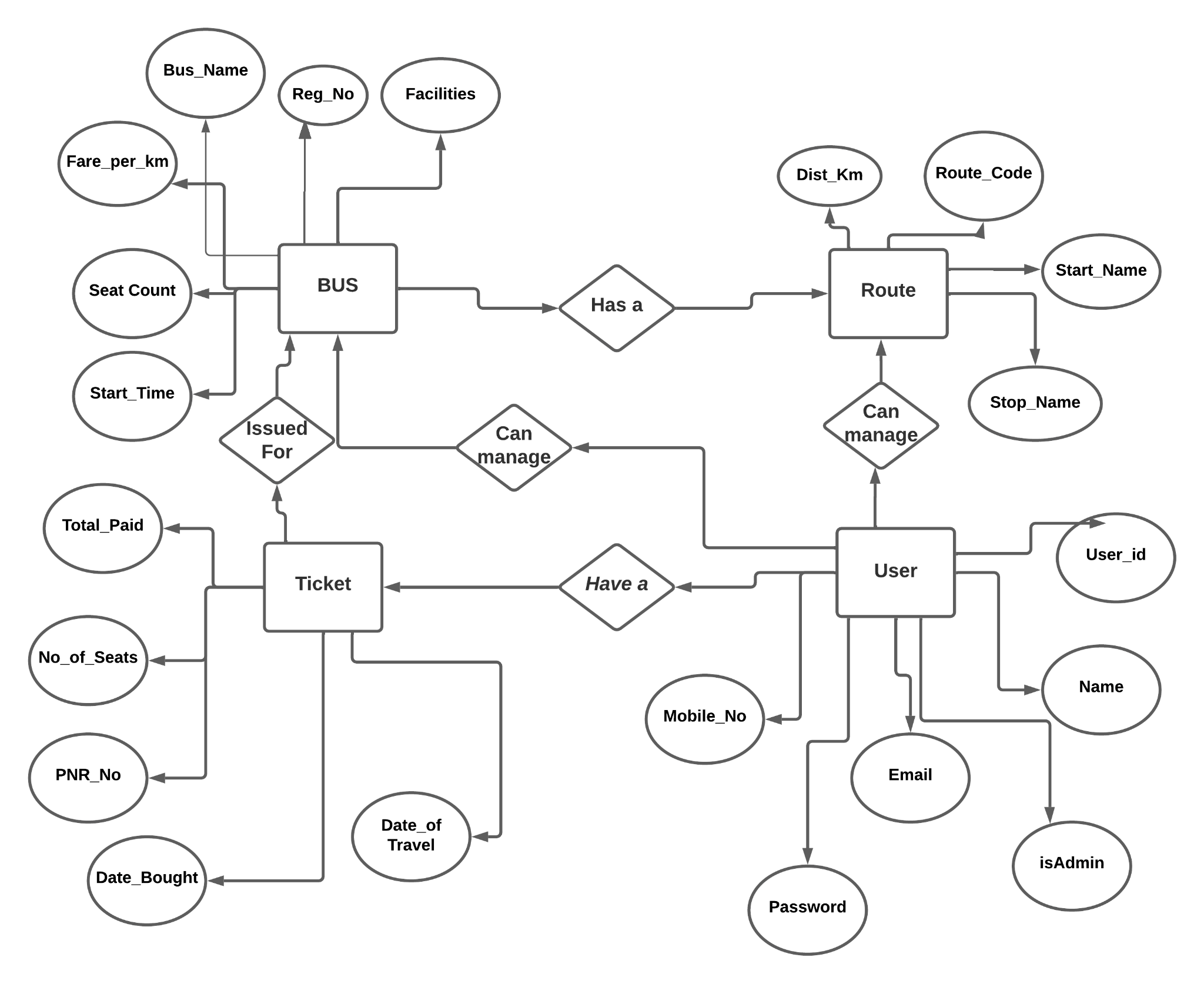
* This system will not accept any payment.
* No waitlist ticket can be generated.
* No partial cancellation of a ticket can be done.
* No registration of new admin can be done from the frontend

# Architecture

The details of the system and data flow are given in the architecture section. The bus management system is made up of information about buses and routes, as well as their management. It also provides passenger-related data and its management, which includes the ability to create, amend, and delete records. A three-layer system will be used. HTML, CSS, and JS will be used to implement the front-end layer. The middle layer will be built with Java 8/Spring Framework, while the database layer will be built with MYSQL.



# ER Diagram



# Use Case

Below are the Use cases for the development of the Bus management system:

1. Passengers can register/login into the system.
2. Passengers can see all bus information between two stops.
3. Passengers can book a bus ticket with a maximum of 6 seats.
4. Passengers can cancel a ticket as a whole.
5. On Booking a ticket, a unique 10 digit PNR will be generated.
6. Admin can add a new bus.
7. Admin can disable a bus.
8. Admin can generate Revenue reports for a specified time period.
9. Admin can generate a ticket list of a bus for a particular date.

# Constraints

Constraints of this software development are defined here. Some constraints of this development are described here.

* Passenger data is the sole property of the customer.
* Access to Customer premises must be between morning 10.00 am to 5p.m and no extended period of work is allowed inside customer premises.
* Customer side desktops, to be used for development, are quite old and use core-i3 CPU with only 2GB RAM.

# Technologies used

Here is the list of technologies used for this application:

* Frontend Development – HTML, CSS, JavaScript, jQuery
* Middle tier business logic – Spring Framework, Hibernate ORM, Spring Data JPA
* Backend database – MySQL database
* Reporting(Excel) – Apache POI

# Tools and Software required to develop

A bus management system is a multiuser web-based software that keeps track of bus information as it travels along various routes. Data will be stored in a database called MySQL, and interaction between the user and the database will be handled by middle-tier software written in Java. Spring Tool Suite 4 (4.13.1) will be used for Java/JSTL/HTML/JSP development. MySQL Workbench will be used for database development.

# S/w and H/w sizing

Software and Hardware must be specified to cater to the SLA (Service Level Agreement) defined between Customer and Software supplier. To meet the pre-defined SLA, it is required to define the correct sizing of hardware to provide service within the stipulated time. For multiuser systems and huge transaction rates, software and hardware must be designed and tuned to the appropriate level.

S/w requirement:

* OS – Windows 10/Linux
* Java – version 8
* Spring – 4. x
* Hibernate – 5.x
* MySQL – 8.x

H/w requirement:

* RAM – 4GB
* Storage – 512 MB

# Acronyms and Abbreviations

* HLD – High-Level Design
* SLA – Service Level Agreement
* JSTL – Java Standard Tag Libraries
* S/W – Software (Java version, Database version, Reporting tool software)
* H/W – Hardware (Server-side CPU/RAM/LAN and Network architecture)
* ER – Entity Relationship