Database Design Document

for

Railway Management System

Version <1.0>

Prepared by

Group Number: 12

Padala Satwik	**B200812CS**	**padala_b200812cs@nitc.ac.in**
Puthalapattu Kushal	**B200796CS**	**puthalapattu_b200796cs@nitc.ac.in**
Kommineni Avinash krishna	**B200834CS**	**kommineni_b200834cs@nitc.ac.in**
Pattipati Manaswi	**B200790CS**	**pattipati_b200790cs@nitc.ac.in**
Nakka Suseel Kumar	**B200858CS**	**nakka_b200858cs@nitc.ac.in**

Instructor	**: Dr . Prabhu M**
Course	**: CS3002D-Database Management Systems**
Date	**: 30-10-22**

Revision History

Date	Version	Author	Change
1-11-22	Version<1.0>	Satwik, Kushal, Manaswi, Avinash, Suseel	This is the first version of this model

Table of Contents

1 Purpose

The Database Design Document maps the logical data model to the target database management system with consideration of the system's performance requirements. The Database Design converts logical or conceptual data constructs to physical storage constructs (e.g., tables, files) of the target Database Management System (DBMS).

1.1 Document Objectives

The Database Design Document has the following objectives:

- To describe the design of a database, that is, a collection of related data stored in one or more computerized files that can be accessed by users or computer developers via a DBMS.
- To serve as a basis for implementing the database and related software units. It provides the acquirer visibility into the design and provides information necessary for software development.

1.2 Intended Audience

This document targets the following target groups:

- Technical reviewers who need to evaluate the quality of this document.
- Developers including:
- Architects whose overall architectural design must meet the requirements described in this document.
- Designers whose design must meet the requirements specified in this document.
- Developers whose software needs to implement the requirements specified in this document.

Acronyms and Abbreviations

Acronym/Abbreviation	Meaning
RDBMS	Relational Database Management System
DBA	Database Administrator
1NF	First Normal Form
2NF	Second Normal Form
3NF	Three Normal Form
BCNF	Boyce Codd Normal Form
OS	Operating System

Key Personnel

Role	Name	Email	PHONE NO
DBA	P.Satwik	padala_b200812cs@nitc.ac.in	9394624898
DBA	P.Kushal	puthalapattu_b200796cs@nitc.ac.in	7013354649
DBA	K.Avinash	kommineni_b200834cs@nitc.ac.in	7995851866
DBA	P.Manaswi	pattipati_b200790cs@nitc.ac.in	7306928669
DBA	N.Suseel	nakka_b200858cs@nitc.ac.in	8919597065

2 Assumptions, Constraints, and Dependencies

2.1 Assumptions

The product needs the following third-party applications for the development of the project:

- XAMPP
- WonderShare Edraw Max

2.2 Constraints & dependencies

Any update regarding the train from the railways is to be recorded to have the correct values.

3 System Overview

This database was developed for web applications used by railway users. This database stores details of trains/users using the railways, as well as administrators who track train whereabouts and updates.

3.1 Database Software Utilities

The database management system used is my SQL as the name suggests SQL is used for creating and handling database and all required functions and before performing any related functions or queries Xampp server should be started.

3.2 Support Software

The software directly related to the database we are using for storing our database is my SQL which basically uses SQL as a query language to perform all the database related queries whereas **MySQL** is written in C and C++. Its **SQL** parser is written in yacc, but it **uses** a home-brewed lexical analyzer.

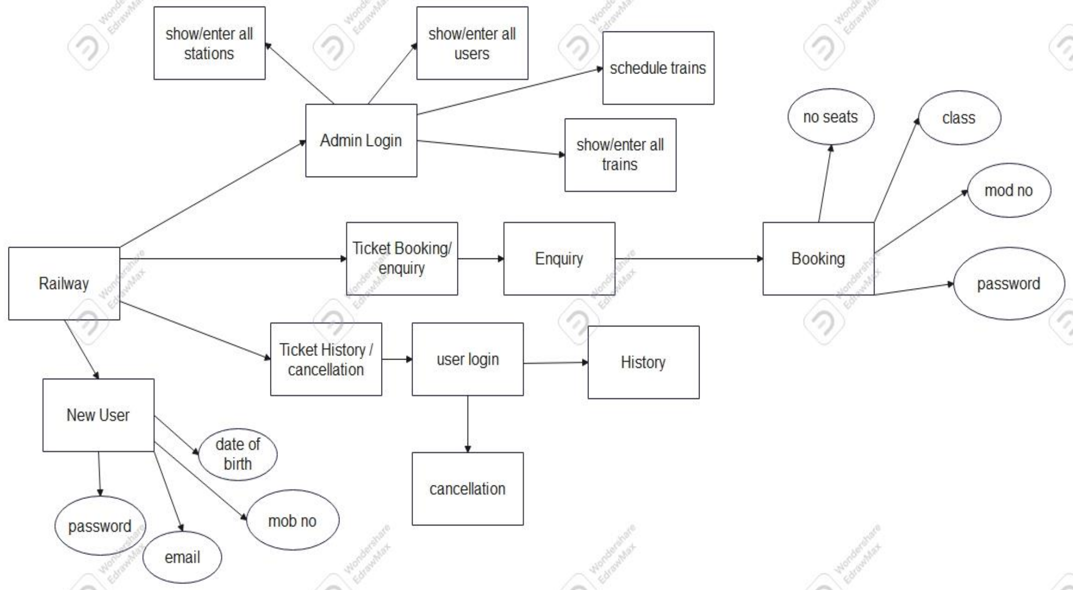
My SQL is a web server database and all the details regarding it and Apache server is the one through which we are going to handle all the files related to the database.

4 Architecture

4.1 Hardware Architecture:

In hardware architecture we will mostly see how the data is stored in databases and how they are connected. It mostly comprises internal level. So, it deals with how the information is stored.

Flow chart:



Register table:

Register table stores username, email, password and mobile number.

Enquiry and ticket booking table:

This table stores starting point, destination point and date of journey and once the user submits then it will show all the trains that are scheduled at that date from these two cities and proceeds with train booking and entering user credentials.

Ticket history and Cancellation:

In this table , by entering registered mobile number and password and logging in, it displays the tickets booked by the user in the history and the tickets booked for future can be cancelled by entering their PNR number.

Admin Login:

In this table, by entering userID and password of admin,we have access to show and modify stations,trains,users and train schedule. This table also contains all booked and cancelled tickets.

Requests and Reviews table:

In this table,the requests and reviews which are entered and user credentials
are stored.

4.2 Software Architecture:

Software architecture mostly deals with external level. Software architecture deals with interfaces. In Railway Management System, we have many interfaces and now let us see how they work.

INTERFACES:

1.Register interface
2.Enquiry and ticket booking

3.Ticket History and Cancellation interface

4.Admin interface

5.Requests and Reviews Interface

1.*Register interface:* In this interface the user can register by entering details like email, password, mobile number, dob etc. And when they completed their registration, they can save and return back to home page.

2.*Enquiry and ticket booking interface:* In this interface, the user can enter their starting and destination points, date of journey and get the trains that are scheduled in that date. The user can login to the portal and after login they have to enter their personal details and chose their birth and proceed with the payment. In this Login interface we have many options, if we click on home, we will be redirected to home page.

3.*Ticket History and cancellation interface:* In this interface, the user can enter his mobile number and password and it will redirect him to the page where he can see previously booked tickets, the tickets booked for the future and have an option to cancel by entering the PNR number.

4.Admin interface**:** In the admin interface, admins can enter their email and password and it will redirect them to a admin page where they can show all stations, trains, users, booked and cancelled tickets. It also allows to add new trains scheduled and stations.

5.Requests and Reviews interface**:** In this interface, the users add their requests and reviews regarding their journey by entering their login credentials.

Components: Railway Management System has many components. They are listed below:

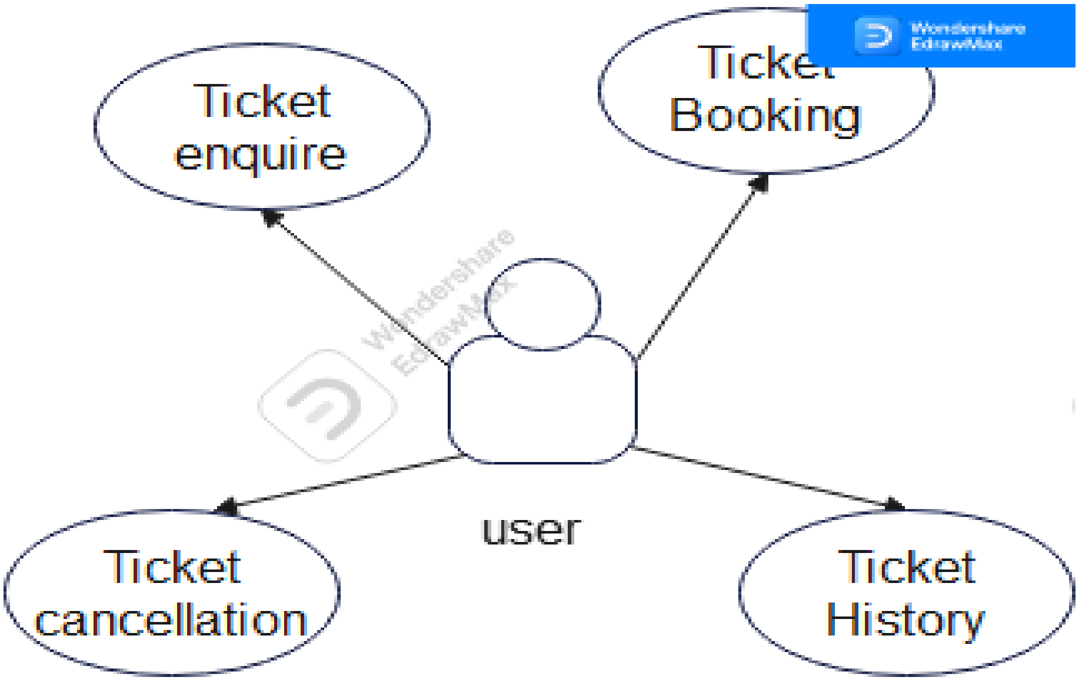
1.User: A user can search trains, put forward a request for a train number and view the trains that are scheduled, view their details, ticket history,reviews ,returns, view responses sent by admin.

Ticket enquire: A user can search for books which are added by admin. If the book that is required for the student is present then the student can requestfor the book.

Ticket Booking: A user can book his/her
tickets by entering the login credentials and
paying the required amount

Ticket Cancellation: A user can see the tickets issued to him by clicking on this.By this he can enter the unique PNR number to cancel the tickets that are booked by him.

Ticket History: A user can leave a request and review which he/she feels best for that particular train and the journey.



vv

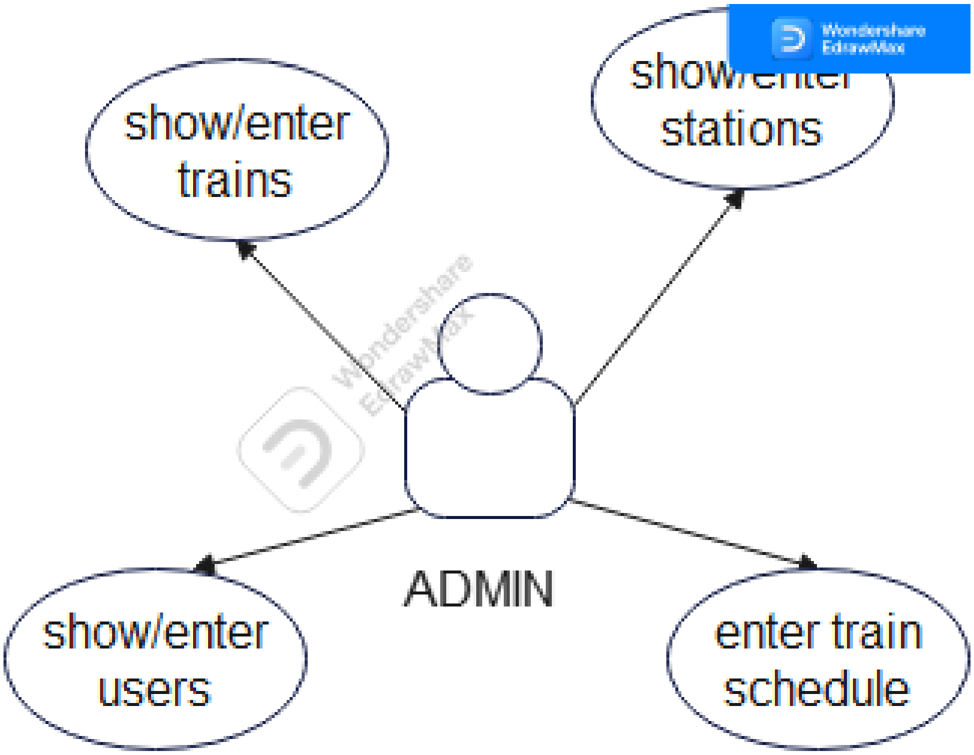
Admin: An admin can add trains,stations, view requests, can view currently issued tickets,can add new trains,stations and previous history of the users.

Show/enter trains: Admin can add trains to the railways,categorical wise so that users can get the trains required for their journey.

Show/enter stations: Admin can add newly added stations ,if the trains are required to stop at the particular station and trains scheduled at that station.

Show/register Users: Admin can view the users who are registerd using their credentials. He cannot alter the user credentials.

Enter Trains scheduled: Admin can add trains scheduled between the two stations according to the train details respectively.



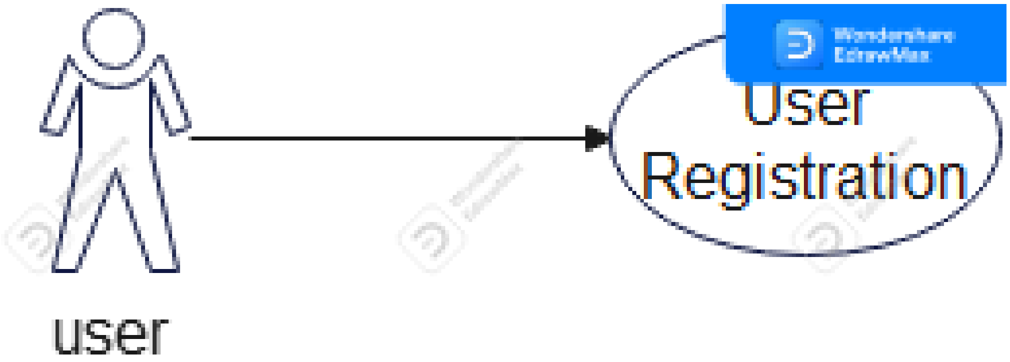
4.3 Datastores:

Data is stored in the library management database and in the tables mentioned above. For file management, we are storing the content in htdocs in the xampp folder. Php files are used for backend and html and CSS files are used for front end.

5 Database-Wide Design Decisions

5.1 Interfaces

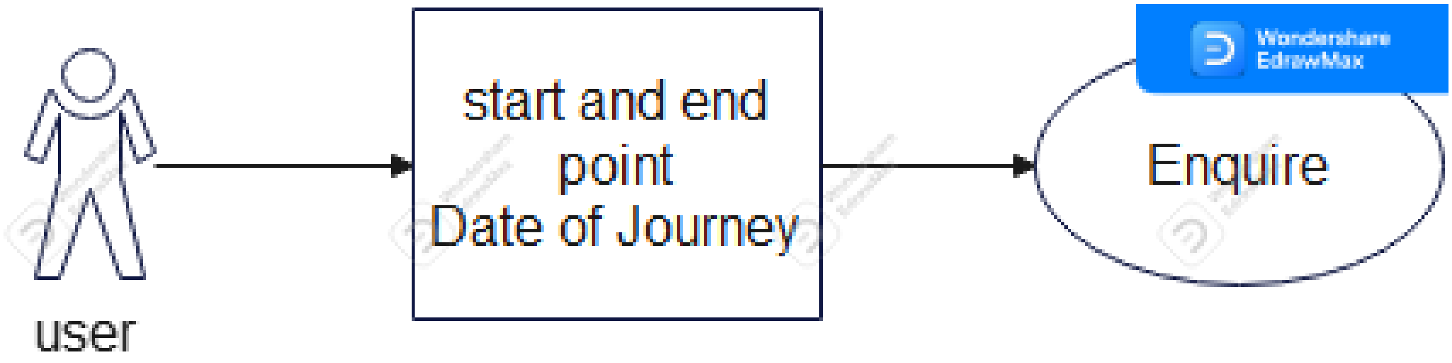
5.1.1 Register interface:



Interface	Register interface
Summary	User have to register in the database
Actor	User
Trigger	admin, user
Primary scenario	User have to fill in the details to register
Alternative scenario	If user exists, then user can login directly
Exceptional scenario	None

Pre conditions	None
Post conditions	The user will be redirected to login page
Assumptions	User is connected to the internet

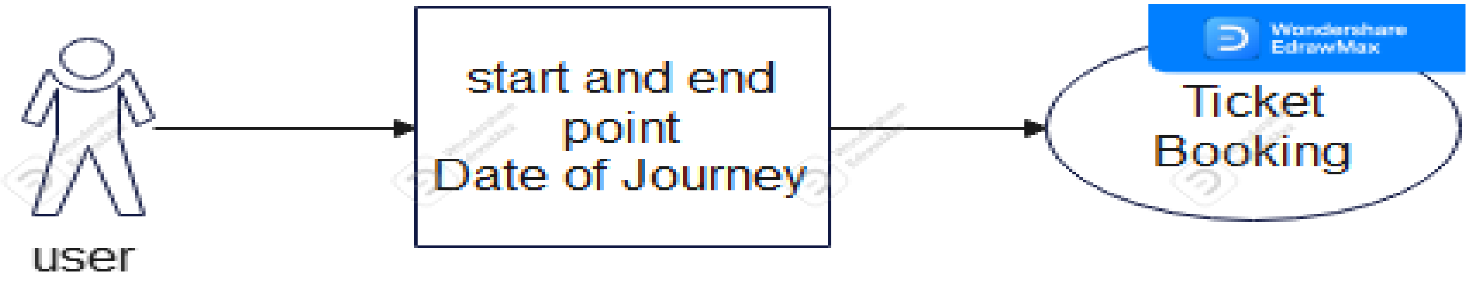
5.1.2 Train Enquiry



interface	Train Enquiry page
summary	User can view the trains scheduled
actor	Users
trigger	admin/user
Primary scenario	Member gives start and end point with date of journey
Alternate scenario	If details are wrong, error message is shown

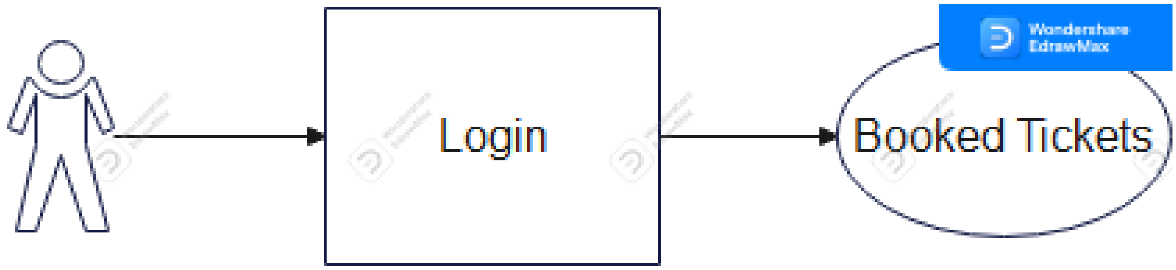
Exceptional scenario	None
Pre conditions	users should know their credentials
Post conditions	If details are correct, will be redirected to respected member page
assumptions	User is connected to the internet

5.1.3 Train Booking



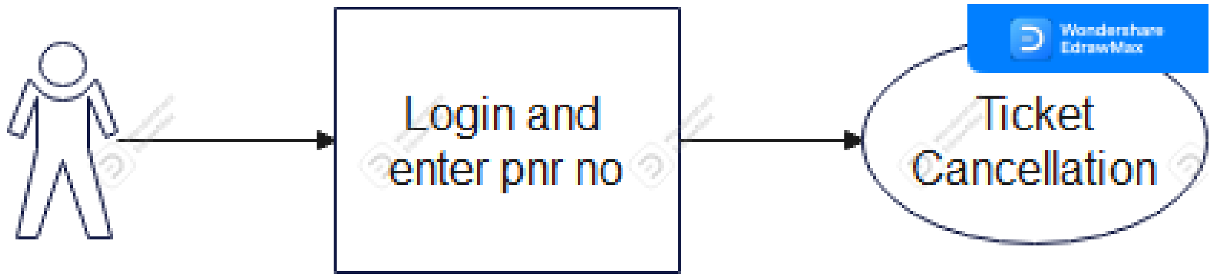
interface	Book a train
summary	User can search for required trains
actor	User
trigger	Book button
Primary scenario	User can book for required train through book button
Alternative scenarios	None
Exceptional scenarios	None
Pre conditions	User should successfully log in
Post conditions	User can request for the issue of the train tickets he searched for
assumptions	User is connected to the internet

5.1.4 Re-viewing Booked tickets:



interface	User history
summary	User can see the tickets booked
actor	User
trigger	Submit button
Primary scenario	User can re-view the tickets he/she booked
Alternative scenarios	None
Exceptional scenarios	None
Pre conditions	User should successfully login
Post conditions	None
assumptions	User is connected to the internet

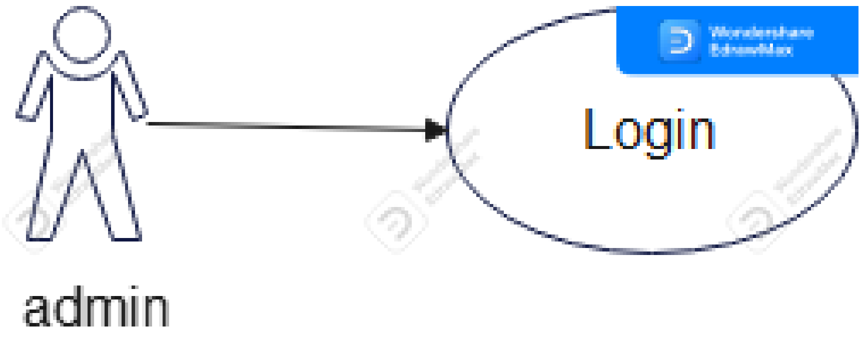
5.1.5 Tickets Cancellation



interface	Tickets cancellation interface
summary	User can cancel the tickets booked by him.
Actor	User
Trigger	Cancel button
Primary scenario	Student should enter The PNR number.
Alternative scenario	None
Exceptional scenario	None

Pre conditions	User should login successfully
Post conditions	User successfully cancels the ticket
assumptions	User is connected to the internet

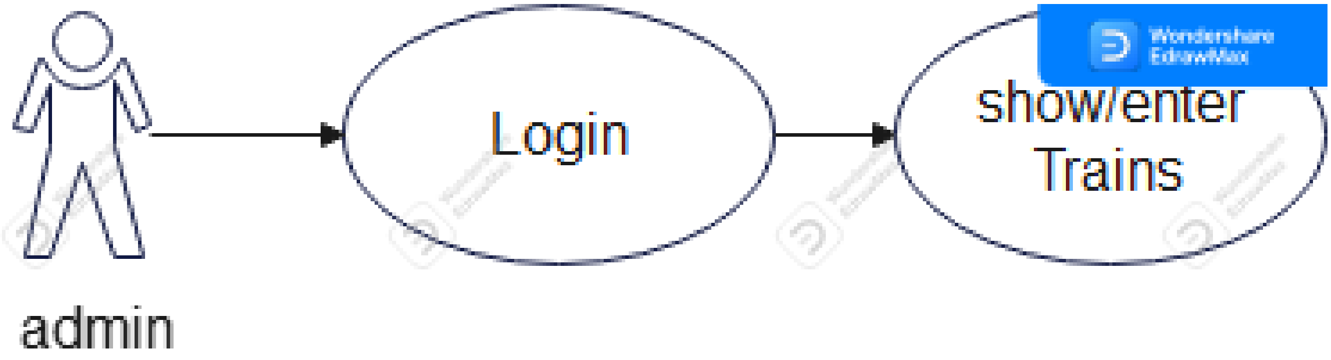
5.1.6 Admin Login



interface	Admin interface
summary	Admin logs into the website
Actor	Admin
Trigger	Add login button
Primary scenario	Admin gets to enter his/her credentials.
Alternative scenario	None
Exceptional scenario	None
Pre conditions	None

Post conditions	Admin successfully gets login into the admin portal
assumptions	Admin is connected to the internet

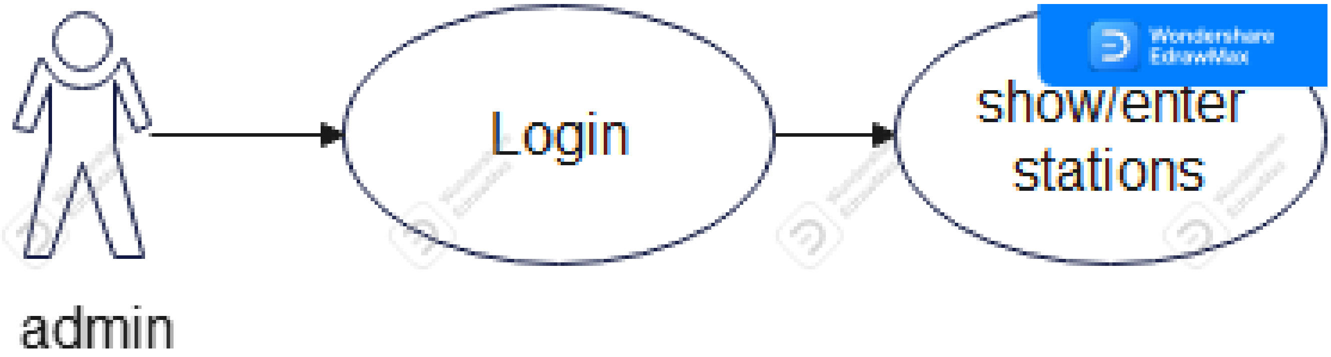
5.1.7 Admin modifies train details:



interface	View Admin interface
summary	Admin can view and enter trains by viewing Show /Enter Trains
Actor	Admin
Trigger	View Show/Enter button
Primary scenario	Admin can view trains that are added by him previously.
Alternative scenario	None
Exceptional scenario	None
Pre conditions	Admin should login successfully

Post conditions	Admin successfully adds and views the required trains
assumptions	Admin is connected to the internet

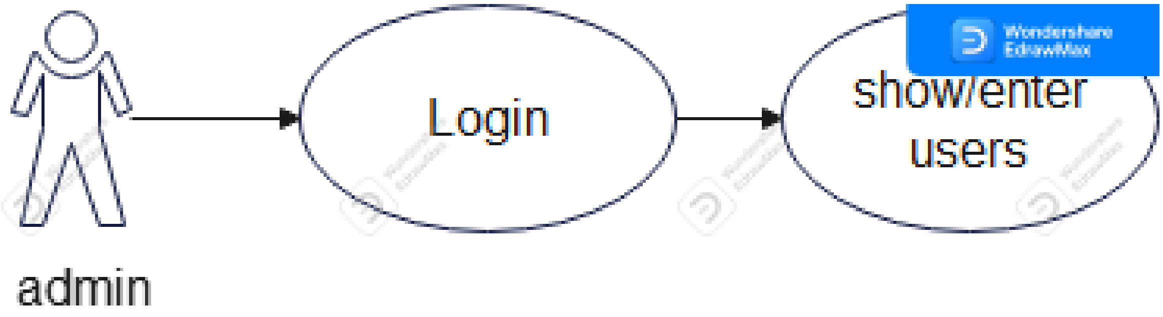
5.1.8 Admin modifies Stations details:



interface	Admin interface
summary	Admin can view and add the extra stations
Actor	Admin
Trigger	View Show/Enter button
Primary scenario	Admin can view the prevoisly added stations by him.
Alternative scenario	None
Exceptional scenario	None
Pre conditions	Admin should login successfully
Post conditions	Admin can view and add stations.

assumptions	Admin is connected to the internet

5.1.9 Currently issued books:



Interface	Currently Admin interface
Summary	Admin can view the users registerd
Actor	Admin
Trigger	Currently issued view and enter users button
Primary scenario	Admin can keep track of users that are logged into the website.
Alternative scenario	None
Exceptional scenario	None
Pre conditions	Admin should login successfully
Post conditions	Admin can keep track of users.
Assumptions	Admin is connected to the internet

5.2 Key Factors Influencing Design

Key factors influencing the design are the functional and nonfunctional requirements.

5.2.1 Functional requirements

User login

- The system must only allow user with email id, mobile number and password to enter thesystem
- The system performs an authorization process which decides what user level can access to.
- The user must be able to logout after they finish using the system.

Register.

- System must be able to verify information
- System must be able to delete information if information is wrong.

Add trains

- System must be able to verify information
- System must be able to enter the number of trains scheduled into a table.
- System must be able to not allow two trains having same train id

Tickets Cancellation:

- System must be able to enter issue information in the database.
- System must be able to cancel the tickets on user demand.

Recommend trains

- System must be able to enter recommended train information into the database.

5.2.2 Nonfunctional requirements

1. Flexibility: The operation may be flexible and reports can be presented in many ways.

2. Maintainability: After the deployment of the project if any error occurs then it can be easily maintained by the software developer.

3. Portability: The software can be deployed at any machine.

4. Reusability: The data and record that are saved in the database can be reused if needed.

5. Robustness: If there is any error in any window or module then it does not affect the remaining part of the software.

6. Timelines: The time limit is very important. It will save much time and provide fast accessing.

7. Security: Data is backed up regularly, the database is protected and users cannot edit the database. For security reasons, only administrators are allowed to edit the database, and administrators can only access the database.

8. Performance: Performance should be accurate and response time should be kept to a minimum so that there are no issues with processing the request. Data should be backed up regularly as a log file to prevent data loss if the data is completely lost due to server crash or file corruption. To be more effective, the DBMS needs to be able to process large amounts of data and take action in less time.

5.3 Behavior

Decisions on behavior in response to each input or query in each page/interface:

1) In the register page users enter all required details and clicks submit then the user is redirected to the login page and details are stored in a database.

2) if the same user again tries to register it will echo the outputs user already registered and there will be no redirection.

3) email and number are unique so when matching details are entered it will show data exists.

4) On the login page if details are entered correctly the user is redirected to the user page.

5) if login credentials entered are wrong it will show the data entered is wrong.

6) special credentials are provided for admin when he enters those credentials, he will be redirected to the admin page.

7) In the admin page he gets access to check and verify all the users as well as the railway details whether they are accurate or not.

8) The admin can add trains in the add trains section with certain details and the page gets reloaded and displays the container with no details.

9) admin can similarly add stations, schedule and many more.

10) Users can request,review for the issue, renew and these respective pages will be redirected after the admin’s response.

11) Users can recommend admin in the request section by entering details after submitting the page will reload.

12) Both users and admin can logout from their respective pages by hitting the logout button and will be redirected to the home page.

5.4 DBMS Platform

This DBMS software can be implemented with ease on Linux based operating systems, windows and mac OS. An open-source cross platform web-server should be installed to create a database (ex: xampp)
Which can have baseline requirements to meet to install.

Baseline requirements for the system:

- Windows 2008 server or later
- Mac os X 10.6
- CentOS,Ubuntu,Fedora,Gentoo,Arch,SUSE
- Hard disk space : 40GB, RAM :256MB
- Processor: Pentium(R)Dual-core CPU

5.5 Security and Availability

This software will,

- Authenticate each user based on the type of user which are admin and student
- When a user performs an action which is not authorized by the user, the system will display an error message if it's found to be unauthorized.

5.6 Distribution

The master database is the SQL Server primary configuration database. It contains information about all databases on the server, including physical database files and their location. The master database file also contains SQL Server configuration settings and login account information.

Components in the master database:

- Registrations and Remote Logins
- Local Databases and Database Files
- Login Accounts

- Server Configuration Settings
- Processes and Locks

A current backup will be kept updating it from time to time as it is critical to any server recovery.

Integrity standards are high and the privacy is kept for the user, no unauthorized information will not be disclosed. No business rules are included.

5.7 Backup and Restore Operations

The user information will be kept private for safety and security issues and will not be disclosed to any other third-party organizations so that user privacy is intact and information is safe.

The data is backed up on a regular basis so the data will not be lost if database crashes or any other harm which leads to loss of data. Also, as a safety measure, the data is stored on a private storage so it can’t be accessed from outside. For detailed information on actions in backup please refer to the 6.16 module below.

5.8 Maintenance

Maintenance includes modifications in the software product after it is delivered. Automatic logging and error reporting techniques, automatic error message generation.

5.9 Performance and Availability Decisions

The developed DBMS software needs to be able to output information effectively when needed and save the data without waiting time to avoid problems. Several factors that affect performance are the need for adequate system resources and the basic requirements for the software to run successfully.

Software availability is sufficient for users and some features are limited to users due to the risk of data loss and data protection issues. The database can only be edited by an administrator to change the data as needed.

6. DATBASE ADMINISTRATIVE FUNCTIONS:

6.1 Database Identification:

Element	Element name	Description
db_name	Railway	*Railway* is the name of the database when it is originally created.
db_path	C:\xampp\htdocs\railway	The full path to where the database is stored on the system.
db_location	C:\xampp\htdocs\railway	Location where the database is stored

6.2 Schema Information:

In Entity Relation Schema,

Entities:

1. Reserved Tickets

2. Train

3. Class

4. Passen

ger_detail

s

5. User

6. Station

Relations:

1. Cancel_Ticket

2. Seat_capacity

3. Train_schedule

Cardinality:

Cancellation_ticket is 1:1

Pnr enquiry is 1:n

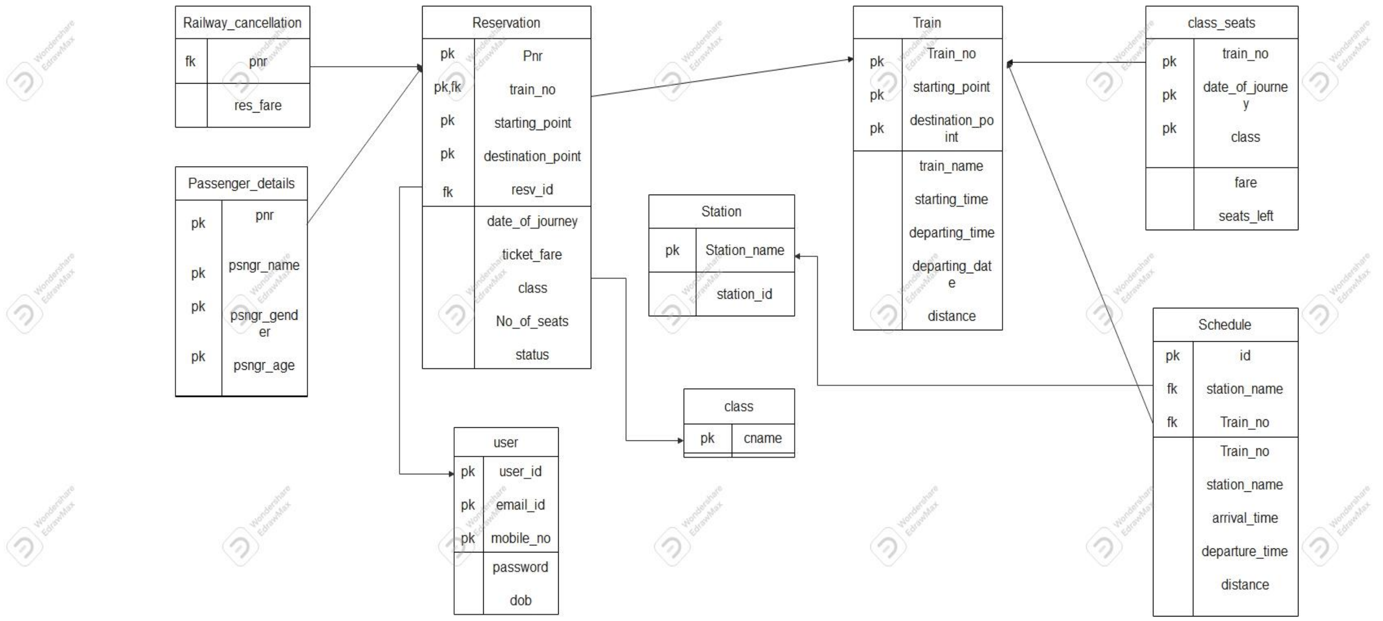
Ticket_booked_in is n:1

User_booking is 1:n

Seat_capacity is n:m

Scheduling of trains is n:m

Relational Mapping



6.3 Physical structure:

Normalization of database

1) It is a technique used to remove redundancy in the database and to maintain consistency and integrity.

2) It is used to remove certain anomalies in the database in order to maintain consistency.

The anomalies are:

1. Insertion Anomalies 2. Deletion Anomalies 3. Update Anomalies

All the tables in our database:

Reservation:

pnr	id	trainno	sp	dp	doj	Ticketfare	status	No of seats	Class

Train :

Train_no	tname	sp	st	dp	dt	dd	Distance

class:

Class_name

Railway station:

Station_id	Station_name

Railway_class_seats:

trainno	sp	dp	doj	class	fare	seatsleft

Train_schedule :

Station_id	Train_no	Station_name	Arrival_time	Departure_time	distance

Railway_user:

id	Email_id	password	Mobile_no	dob

Railway_passenger_details:

pnr	Passenger_name	Passenger_age	Passenger_gender

Railway_cancellation:

pnr	rfare

All the tables above are in normal form already and the tables are normalized up to the Boyce Codd normal form

Checking 1st normal form:

As we can see all our tables are already in normal form as in each table there are no redundancies so the table is in **1NF**

Checking for 2nd normal form:

And in register and book table there is no composite key so no transitive

Dependencies and in other tables date completely depend on the two key attributes so no partial dependencies.so table is in **2NF.**

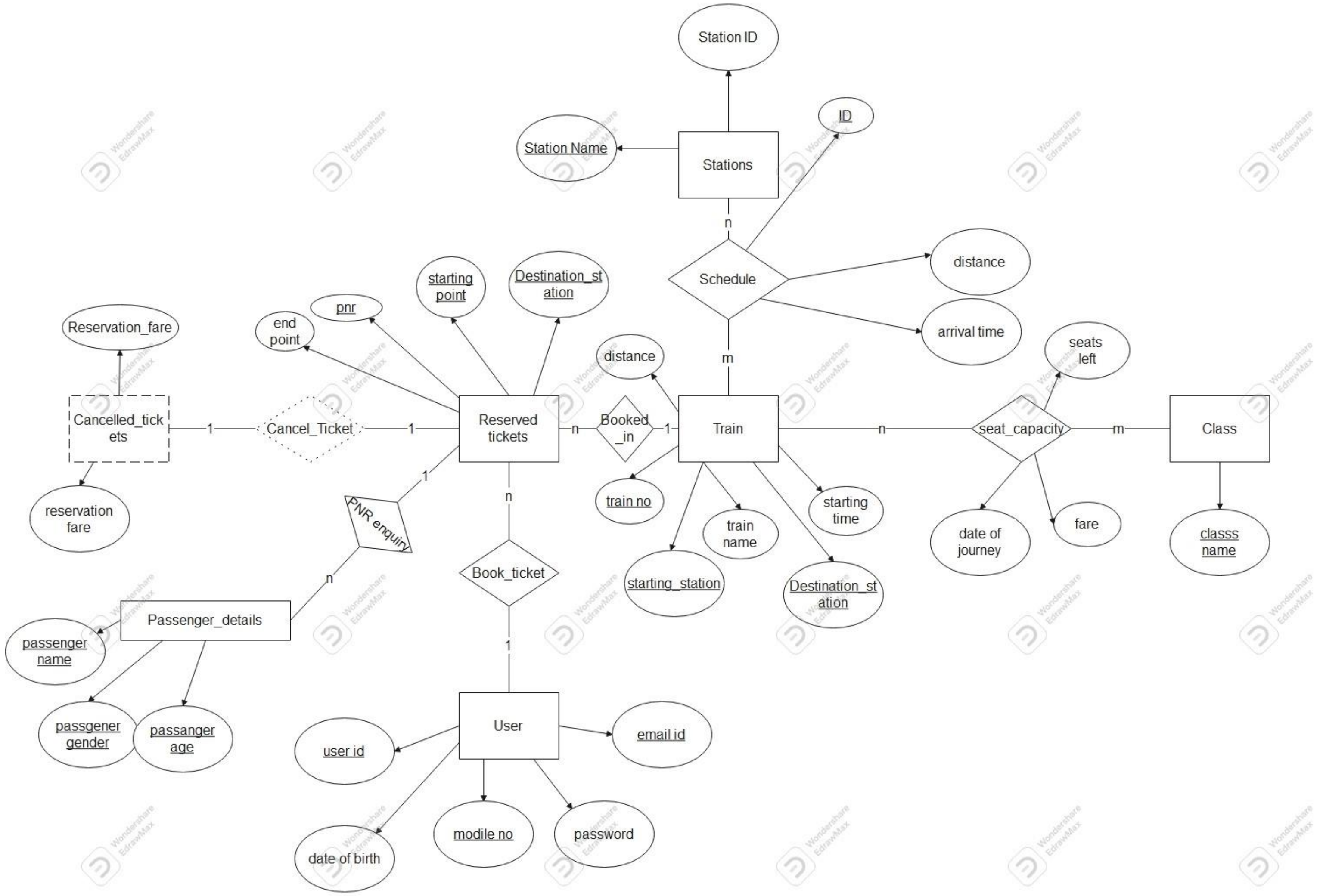
Checking for 3rd normal form:

And in tables register and Books and issue all non-prime attributes depend and prime attributes and in other three tables there is only one non-prime attribute so tables are in **3NF**

Checking for Boyce Codd normal form:

Since all the super keys in each of the tables determine tuples uniquely all tables are in **BCNF.**

6.4 Entity Mapping:



6.5 Mapping Rules:

Rules for mapping entries into tables:
Data type of the entry should be the same as the data type of the column.

For example: a column named id(int) will accept only integer data types

1. The length of any accepted value will be the same as the length defined for it; no additional characters or numbers will be inserted into the table.

Example: in table we defined name (varchar) of length four and suppose entry contains a name “xyzru” only xyzr will be mapped into table and “u” is ignored.

2.on the backend in php or any other language the values to which the names on the front end are inserted should be the names of tables and should not be different.

Example: Name is inserted into username in php then all the values entered for Name will be inserted into user name

3.All values should be entered correctly in their provided space.

6.6 Operational Implications

Refresh:

When the page is refreshed, it will redirect to the same web page.

Update:

1. As there is no separate link provided for updating user’s data there is no option for updating from the student side.

2. An admin can update the details of trains add stations the changes willbe reflected in the database

3. An admin can schedule the trains.

4. After updating by admin page returns to main menu or admin page.

6.7 Backup and Recovery

For backup in case server fails or pc is damaged a backup shall be created by following steps:

1. Go to the root directory where the xampp folder is located.
2. Go to the mysql folder in the xampp folder.
3. Go to bin folder
4. Go to sql.
5. Copy the path.

6. Open the terminal and go to the directory where xampp is located.
7. Paste the path.
8. Type the command “mysqldump -u root -p portal>backup.sql”
9. Execute the command
10. A backup will be created.
11. Save the backup file.
12. For recovery start xampp server open mysql\php my admin
13. Go to export select the backup .sql file and click export
14. Data will be recovered

6.8 Applications/Systems Using the Database

The database library created and accessed through phpMyAdmin is used only by web application “library management system” there is no specific version for it.

6.9 Relationship to Other databases:

Our application has only required only one database so there can’t be any relationship with other databases.