Database Design Document

* Bus Tracking System

## 

.

**INSTRUCTOR : Dr. PRANESH DAS**

**CS3002D-Database Management Systems**

**DATE : 9 NOVEMBER 2022**

**Table of Contents:**

**1 Purpose**

* 1. Document Objectives
  2. Intended Audience
  3. Acronyms and Abbreviations
  4. Key Personnel

1. **Assumptions, Constraints and Dependencies**
   1. Assumptions
   2. Constraints & Dependencies
2. **System Overview.**
   1. Database Software Utilities
   2. Support Software
3. **Architecture**
   1. Hardware Architecture
   2. Software Architecture
   3. Datastores
4. **Database-Wide Design Decisions**
   1. Interfaces
   2. Key Factors Influencing Design
   3. Behavior
   4. DBMS Platform
   5. Security and Availability
   6. Distribution
   7. Backup and Restore Operations
   8. Maintenance
   9. Performance and Availability Decisions
5. **Database Administrative Functions**
   1. Database Identification
   2. Schema Information
   3. Physical Structure
   4. Entity Mapping
   5. Mapping Rules
   6. Operational Implications
   7. Backup and Recovery.
   8. Application/Systems Using the Database
   9. Relationship to other databases

**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.
* 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 |

# **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 bus is to be recorded to have the correct values.

# **3 System Overview**

This database was developed for app applications used by bus users. This database stores details of buses/users , as well as administrators who track buses where abouts 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.

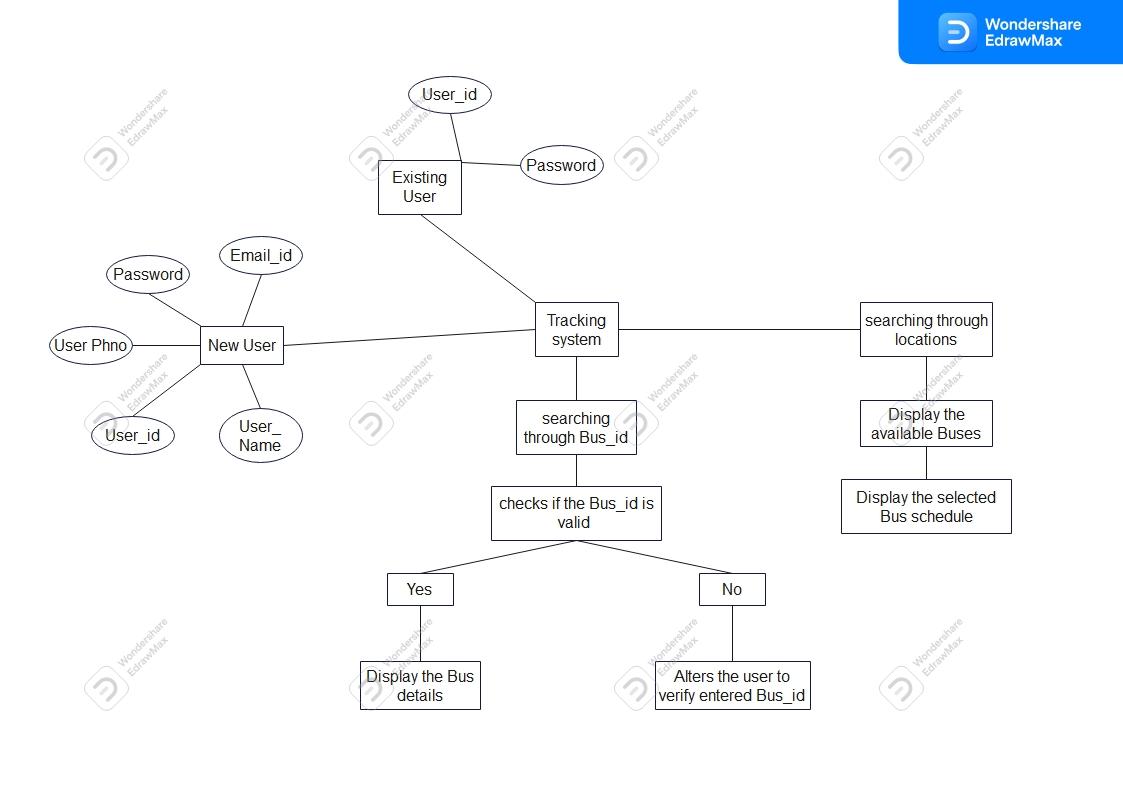
My SQL is a app server database and all the details regarding it and Apache server is the one through which we are going to handle all the files related to the database.

**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:*



**User *table:***

This table has user\_id,user\_name of the users of the application here user\_id is the primary key.

**Phone number:**

This table has id,phone\_number of users and drivers and id is primary key.

**Location :**

This table has location\_name ,map\_link.location\_name is primary key.

**Bus :**

This table bus\_id,bus\_name,speed,start\_time,end\_time from\_loc, to\_loc where bus\_id,from\_loc,to\_loc are the primary keys

**Driver**

This table has driver\_id,driver\_name as attributes where driver\_id is the primary key.

**Schedule**

This table has day,Sbus\_id,start\_loc,end\_loc as attributes.

**4.2 Software Architecture:**

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

**1.User login interface**

**2.Bus details extracting interface**

**1)user login interface**

In this interface the user can register by entering his user\_id and password and once registered user can login.

**2) Bus details extracting interface**

On entering the from and to locations it displays the all buses between them.

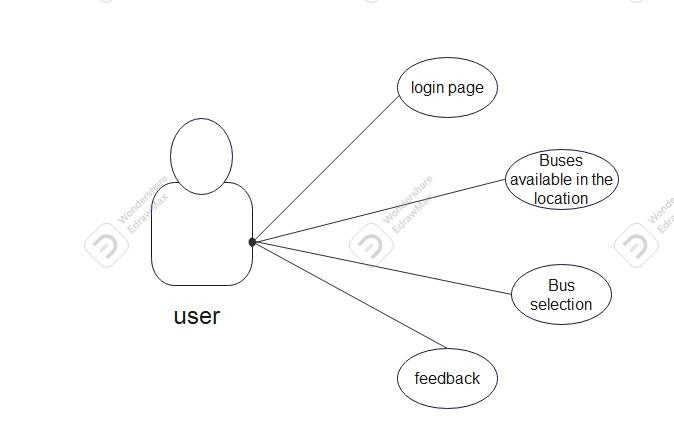
**4.3 Datastores :**

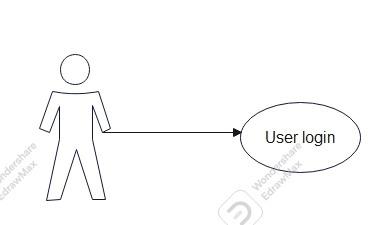
Data is stored in the library management database and in the tables mentioned above. For file management, we are storing the content in htdocs in the xampp folder. Node, react native, sqlite are used for backend and html ,css and javascript files are used for front end.

# **Database-Wide Design Decisions**

## 5.1 Interfaces

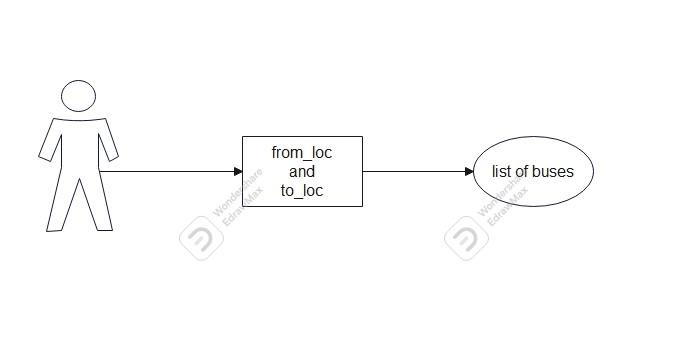
### **5.1.1** Register interface:

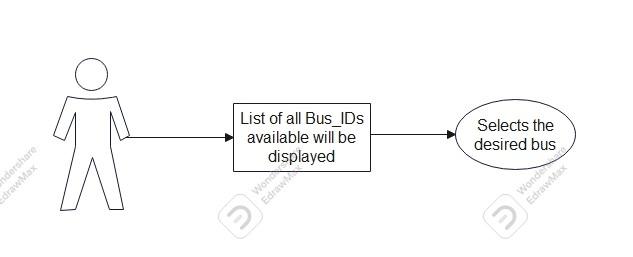




|  |  |
| --- | --- |
| Interface | Register interface |
| Summary | User have to register in the database |
| Actor | User |
| Trigger | User |
| Primary scenario | User have to fill in the details to register |
| Alternative scenario | If user exists, then user can login directly |
| Exceptional scenario | None |
| Pre conditions | None |
| Post conditions | The user will be redirected to login page |
| Assumptions | User is connected to the internet |

### **5.1.2 Bus Enquiry**





|  |  |
| --- | --- |
| interface | Bus Enquiry page |
| summary | User can view the trains scheduled |
| actor | Users |
| trigger | Users |
| Primary scenario | Member gives start and end point with date of journey or Bus number |
| 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 |

## 5.2 Key Factors Influencing Design

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

**5.2.1 Functional requirements**

**User login**

* The system must only allow user with email id, mobile number and password to enter the system
* 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

### **5.2.2 Nonfunctional requirements**

* **Flexibility:** The operation may be flexible and reports can be presented in many ways.
* **Maintainability:** After the deployment of the project if any error occurs then it can be easily maintained by the software developer.
* **Portability:** The software can be deployed at any machine.
* **Reusability:** The data and record that are saved in the database can be reused if needed.
* **Robustness:** If there is any error in any window or module then it does not affect the remaining part of the software.
* **Timelines:** The time limit is very important. It will save much time and provide fast accessing.
* **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.

## 5.3 Behavior

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

## 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.

## If the same user again tries to register it will echo the outputs users already registered and there will be no redirection.

## Email and the number are unique so when matching details are entered it will show data exists

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

## If login credentials entered are wrong it will show the data entered is wrong.

## 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).

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 their login credentials.
* 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

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.

## 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.

## 

## DATBASE ADMINISTRATIVE FUNCTIONS:

## Database Identification:

|  |  |  |
| --- | --- | --- |
| **Element** | **Element name** | **Description** |
| db\_name | Bus | *Bus* is the name of the database when it is originally created. |
| db\_path | C:\xampp\htdocs\bus | The full path to where the database is stored on the system. |
| db\_location | C:\xampp\htdocs\bus | Location where the database is stored |

## 6.2 Schema Information:

## Entities:

## 1. User

## 2. Bus

## 3.Driver

## 4.user\_phoneNo 5.driver\_phoneNo

6.Bus\_info

7.From\_location

8.To\_Location

## Relations:

## 1.Is\_at

## 2.has (bus-driver)

## 

## 

## 

* 1. **Physical structure:**

**Normalization of database**

* It is a technique used to remove redundancy in the database and to maintain consistency and integrity.
* It is used to remove certain anomalies in the database in order to maintain consistency.

**TABLES IN OUR DATABASE :-**

**USER:**

|  |  |  |  |
| --- | --- | --- | --- |
| **User\_id** | **User\_name** | **Password** | **email** |
|  |  |  |  |

**User\_phoneNo:**

|  |  |
| --- | --- |
| **User\_id** | **Phone\_no** |
|  |  |

**FromLocation:**

|  |  |
| --- | --- |
| **Location\_name** | **Map\_link** |
|  |  |

**To\_Location:**

|  |  |
| --- | --- |
| **Location\_name** | **Map\_link** |
|  |  |

**Bus:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Bus\_ID** | **Speed** | **Start\_time** | **End\_time** | **From\_loc** | **To\_loc** |
|  |  |  |  |  |  |

**Bus\_info:**

|  |  |
| --- | --- |
| **Bus\_ID** | **Bus\_Name** |
|  |  |

**Driver:**

|  |  |
| --- | --- |
| **Driver\_id** | **Driver\_name** |
|  |  |

**Driver\_phoneNo:**

|  |  |
| --- | --- |
| **Driver\_id** | **PhoneNo** |
|  |  |

**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 2nd normal form:**

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.**

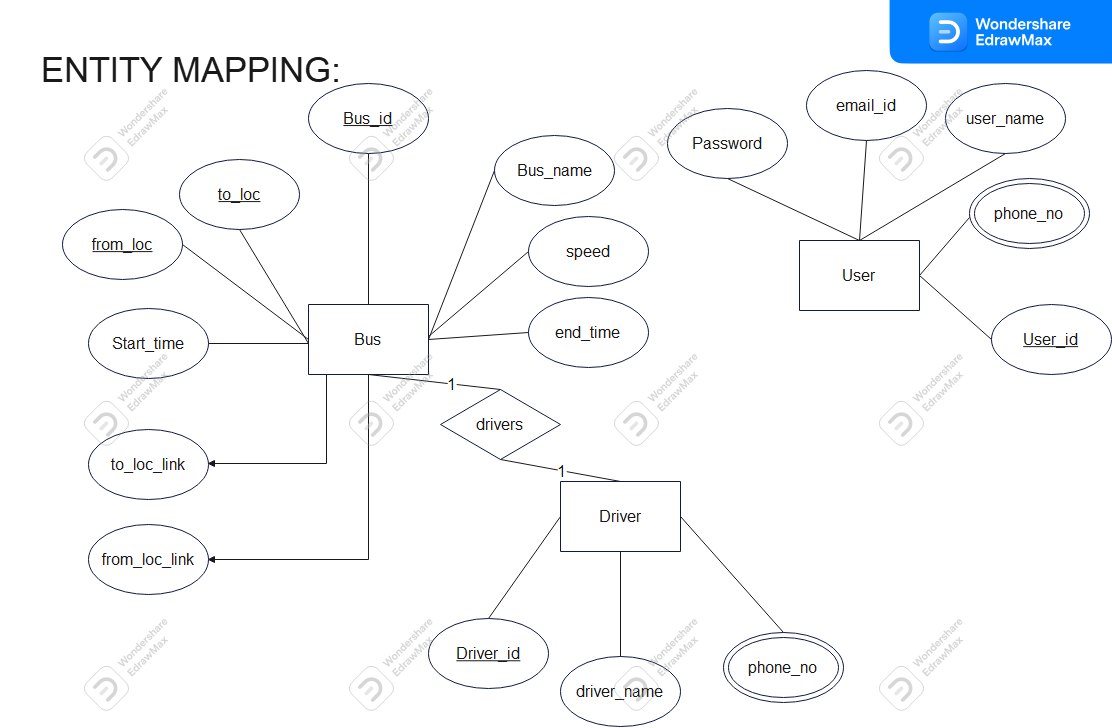
**Checking 3rd normal form:**

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.**

* 1. **Entity Mapping:**



* 1. **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 five and suppose entry contains a name “xyzruv” only xyzru will be mapped into table and “v” is ignored.

* 1. **Operational Implications:**

**Refresh:**

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

**Update:**

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

## 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

## 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.

## Relationship to Other databases:

## As there is only one database there can’t be any relationship with other databases.