

CONTENTS

SL. NO.	TITLE	PAGE NO.
1.	INTRODUCTION	
1.1	General Overview	01
1.2	Introduction to DBMS	02
2.	LITERATURE REVIEW	
2.1	Existing System Model	03
2.2	Proposed System Model	03
2.3	Requirements Specification	04
2.4	Structured Query Language	04
2.5	Entities and Attributes	05
2.6	Relationships	06
2.7	ER Diagram	10
2.8	Relational Mapping	12
2.9	Normalization	13
2.10	Tools and Environment Used	15
3.	PROPOSED WORK	
3.1	ER Model	19
3.2	Relational Model	20
3.3	Normalization	21
4.	RESULT AND SNAPSHOTS	23
5.	CONCLUSION	26
6.	BIBLIOGRAPHY	27

CHAPTER 1

INTRODUCTION

1.1 GENERAL OVERVIEW

Online Bus Booking Management System contains of admin and user, the admin can manage the booking from the customer like monitor booking, add bus information, assign location, and the user is represent as a customer no need to create account just write information and copy the generate code after booking, and the admin can contact the code after booking customer. This simple bus booking system is develop in PHP MySQL database, using JavaScript, HTML, CSS, modal, the system is easy to use and simple to understand.

1.1.1 MODULE DESCRIPTION

Admin Module:

The admin module allows us to manage the number of users and bus list, location list, add or remove it we can also confirm that customer paid the amount its handy and so simple website to book bus

User Module:

The user module is bus booking members that they can book bus from searching location or selecting available bus list from the location they want to travel, they can book number of seats required max seats are 30 per bus, after booking references number will be given the user can pay the traveling amount when boarding by showing the reference number.

1.2 DATABASE MANAGEMENT SYSTEM (DBMS)

DBMS is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, manipulating and sharing databases among various users and applications.

A Relational database is a database that has a collection of tables of data items, all of which is formally described and organized according to the relational model. Data in a single table represents a relation, from which the name of the database type comes. In typical solutions, tables may have additionally defined relationships with each other. In the relational model, each table schema must identify a column or group of columns, called the primary key, to uniquely identify each row. A relationship can then be established between each row in the table and a row in another table by creating a foreign key, a column or group of columns in one table that points to the primary key of another table.

1.2.1 Characteristics of Database Management Systems

- Self-describing nature.
- Keeps a tight control on data redundancy.
- Enforces user defined rules to ensure that integrity of table data.
- Provides insulation between Programs and data, Data abstraction.
- Supports multiple views of the data.
- Helps sharing of data and Multi-user transaction processing.

1.2.2 Advantages of using DBMS approach

- Controlling the redundancy.
- Restricting unauthorized access.
- Providing persistent storage for program objects.
- Providing storage structures for efficient query processing.
- Providing backup and recovery.
- Providing multiple users interfaces

CHAPTER 2

LITERATURE REVIEW

2.1 EXISTING SYSTEM MODEL

The existing Bus Booking System is not completely computerized. The customer has to visit any booking branch if he wants to book a ticket. Bus scheduling, ticket booking, bill generation and many other operations are done manually. This may lead to incorrect entries and there is a lot of room for errors as the data is not completely synced. The availability of seats is not centrally maintained and the travel operator is not fully aware of the availability and occupancy of the seats in his buses. This is the major bottle neck.

2.2 PROPOSED SYSTEM MODEL

The Bus Booking System is a web-based application which is connected to a complete database. The database includes information about buses, no. of seats available, occupancy, availability, days and time of operation, no. of buses from point 'A' to point 'B', Price ranges, automated report and bill generation etc.

The database is updated time to time and a user can reserve his ticket from any part of the world and any time.

The application gives utmost importance to security and usability.

2.3 REQUIREMENTS SPECIFICATION

A software requirement definition is an abstract description of the services which the system should provide, and the constraints under which the system must operate. It should only specify the external behavior of the system. For the successful, efficient and problem free designing any project or program, the system should meet some requirements.

Hardware Requirements

- Processor – Intel core Duo 2.0 GHz or more
- RAM - 512 MB or more
- Monitor
- Hard disk - 60 GB or more
- Mouse
- CD drive LG
- Keyboard Standard

Software configuration

- Operating System - Windows XP Professional
- Software – Xampp (for apache server)
- Database - MYSQL
- Front End - HTML5, CSS, JS
- Back End – PHP

2.4 STRUCTURED QUERY LANGUAGE (SQL)

SQL is a comprehensive database language. SQL is a standard language for storing, manipulating and retrieving data in databases. The ANSI standard SQL provides basic functions for data manipulation, transaction control, and record retrieval from the database. However, most end users interact with Oracle through application that provides an interface that hides the underlying SQL and its complexity. SQL uses the terms table, row, and column for relation, tuple, and attribute, respectively.

Applications of SQL:-

SQL is one of the most widely used query language over the databases that:

- Allows users to access data in the relational database management systems.

- Allows users to describe the data.
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.

2.5 ENTITIES AND ATTRIBUTES

An entity may be an object with a physical existence (for e.g. A particular person, car or employee) or it may be an object with a conceptual existence (e.g. company, university).

Each entity has attributes i.e., the particular properties that describe it. The attribute values that describe each entity become a major part of the data store in the database. Whenever an attribute of one entity type refers to another entity type, a relationship exists. In the initial design of entity types, relationships are typically captured in the form of attributes. As the design is refined these attributes get converted into relationships between entity types. In the ER diagrams, the emphasis is on representing the schemas rather than the instances. This is more useful in the database design because a database schema changes rarely, whereas contents of the entity sets change frequently.

Types of Attributes

Simple attribute: An attribute is classified as a simple attribute if it cannot be partitioned into smaller components. For example, age and sex of a person. A simple attribute is represented by an oval.

Composite attribute: A composite attribute can be subdivided into smaller components which further form attributes. For example, 'name' attribute of an entity "person" can be broken down into first name and last name which further form attributes. Grouping of these related attributes forms a composite attribute.

Single valued attribute: If an attribute of a particular entity represents single value for each instance, then it is called a single-valued attribute. For example, Ramesh, Kamal and Suraj are the instances of entity 'student' and each of them is issued a separate roll number. A single oval is used to represent this attribute.

Multi valued attribute: An attribute which can hold more than one value, it is then termed as multi-valued attribute. For example, phone number of a person.

Derived attribute: A derived attribute calculates its value from another attribute. For example, 'age' is a derived attribute if it calculates its value from 'current date' and 'birth date' attributes. A derived attribute is represented by a dashed oval.

Stored attribute: In some cases, two or more attribute values are related for example, the Age and Birthdate attributes of a person. For a particular person entity, the value of Age can be determined from the current date and the value of the person's Birthdate. The Age attribute hence called derived attribute and is said to be derivable from the Birthdate attribute, which is called a stored attribute.

Complex attribute: It is a Composite and Multivalued attribute.

2.6 RELATIONSHIPS

A relationship is defined as bond or attachment between 2 or more entities. Normally, a verb in a sentence signifies a relationship.

For example: An employee assigned a project.

Teacher teaches a student.

Author writes a book.

A diamond is used to symbolically represent a relationship in the ER diagram.



Figure: Relationship connecting two Entities.

Various terms related to relationships:

Degree of relationship: It signifies the number of entities involved in a relationship.

Degree of a relationship can be classified into following types:

Unary relationship: If only single entity is involved in a relationship then it is a unary relationship. For example, an employee (manager) supervises another employee.

Binary relationships: When two entities are associated to form a relation, then it is known as a binary relationship. For example, a person works in a company. Most of the times we use only binary relationship in an ER diagram.

Other types of relationships are ternary and quaternary: As the name signifies a ternary relationship is associated with three entities and a quaternary relationship is associated with four entities.

Connectivity of a relationship: Connectivity of a relationship describes, how many instances of one entity type are linked to how many instances of another entity type.

Various categories of connectivity of a relationship are:

One to One (1:1) - "Restaurant has menu" signifies a one-to-one relationship because only one instance of an entity is related with exactly one instance of another entity type.

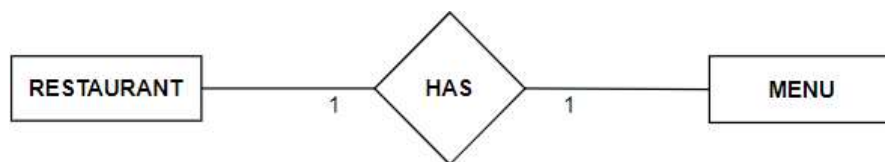


Figure: One to One Relationship

One to Many (1: M) - "A Customer places Order" is a one-to-many relationship because a customer can place more than one order, but a order is related to only one Customer.



Figure: One to Many Relationship

Many to One (M: 1) - "Menu has items" is a many-to-one relationship.

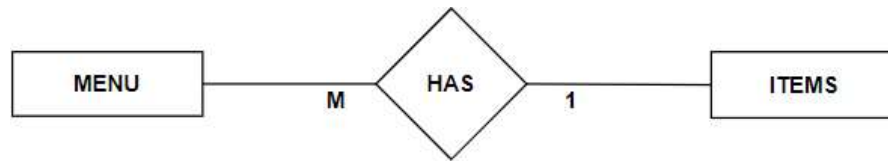


Figure: Many to One Relationship

Many to Many (M: N) - "Author writes books" is a many-to-many relationship because an author can write many books and a book can be written by many authors.



Figure: Many to Many Relationship

Total Participation and Partial Participation

Total Participation: It specifies that each entity in entity set must compulsorily participate in at least one relationship instance in that relationship set. It is also called as Mandatory Participation. The total participation is represented using a double line between the entity set and the relationship set.

Partial Participation: It specifies that each entity in entity set may or may not participate in the relationship instance in that relationship set. It is also called as Optional Participation. The partial participation is represented using a single line between the entity set and the relationship set.

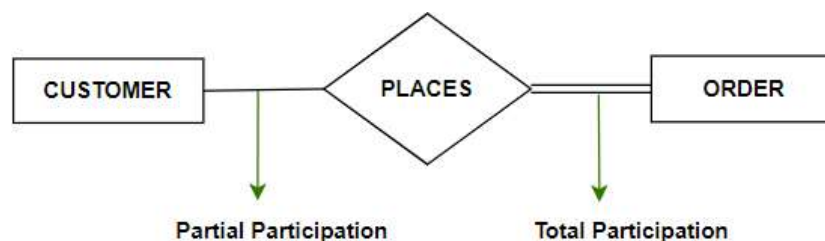


Figure: Total and Partial Participations

Types of Entities

Strong entity: A strong entity has a primary key attribute which uniquely identifies each entity.

Symbol of strong entity is same as an entity.

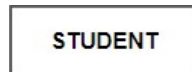


Figure: Strong Entity

Weak entity: A weak entity does not have a primary key attribute and depends on other entity via a foreign key attribute.




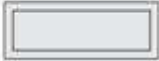





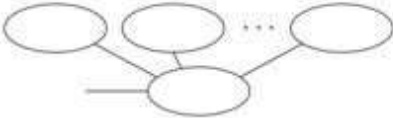



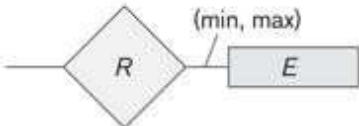
Figure: Weak Entity

2.7 ER DIAGRAM

An Entity-relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of ER model are:

- Entity set.
- Relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. An Entity-Relationship (ER) model is an abstract way to describe a database.

Symbol	Meaning	Figure Sum for E
	Entity	
	Weak Entity	
	Relationship	
	Identifying Relationship	
	Attribute	
	Key Attribute	
	Multivalued Attribute	
	Composite Attribute	
	Derived Attribute	
	Total Participation of E_2 in R	
	Cardinality Ratio 1: N for $E_1:E_2$ in R	
	Structural Constraint (min, max) on Participation of E in R	

Notations for ER Diagram [1]

2.8 RELATIONAL MAPPING

A relational schema for a database is an outline of how data is organized. It can be a graphic illustration or another kind of chart used by programmers to understand how each table is laid out, including the columns and the types of data they hold and how tables connect. A database schema usually specifies which columns are primary keys in tables and which other columns have special constraints such as being required to have unique values in each record. In this section we describe the steps of an algorithm for ER to relational mapping. The above ER diagram is used to derive the respective relational schema. The mapping will create tables with simple single valued attributes. The steps are:

- Mapping of Regular Entity Types.
- Mapping of Weak Entity Types.
- Mapping of Binary 1: 1 Relationship Types.
- Mapping of Binary 1: N Relationship Types.
- Mapping of Binary M: N Relationship Types.
- Mapping of Multivalued Attributes.
- Mapping of N-ary Relationship Types.

Mapping of 1: N Relationship type:

For each regular binary 1: n relationship type R, identify the relation S that represents the participating entity type at the N-side of the relationship type. Include as the foreign key in S the primary key of relation T that represents other entity type in R. We have to do this because each entity instance on N side is related to at most one entity instance on 1-side of relationship type.

Mapping of M: N Relationship type:

For each m:n relationship type R, create a new relation S to represent R. Include as foreign key attributes in S the primary keys of the relations that represent the participating entity type, their combination will form the primary key of S. Also include any simple attributes of m:n relationship type.

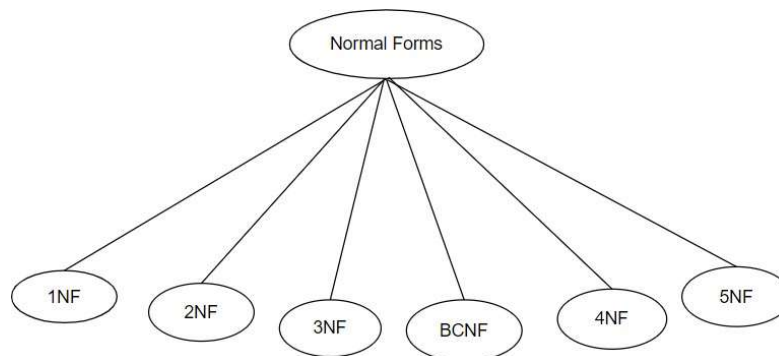
Mapping of 1:1 Relationship type:

For each 1:1 relationship type R can be migrated to any participating entity types. This relationship type ensures that each user in the database can lodge one complaint

2.9 NORMALIZATION

- Normalization is the process of organizing the data in the database.
- Normalization is used to minimize the redundancy from a relation or set of relations.
- It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
- Normalization divides the larger table into the smaller table and links them using relationship.
- The normal form is used to reduce redundancy from the database table.

Types of Normal forms



First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

- It should only have single (atomic) valued attributes/columns.
- Values stored in a column should be of the same domain.
- All the columns in a table should have unique names.
- And the order in which data is stored, does not matter.

Second Normal Form (2NF)

For a table to be in the Second Normal Form,

- It should be in the First Normal form.
- It should not have Partial Dependency.

Third Normal Form (3NF)

A table is said to be in the Third Normal Form when,

- It is in the Second Normal form.
- It doesn't have Transitive Dependency.

Boyce and Codd Normal Form (BCNF)

Boyce and Codd Normal Form is a higher version of the Third Normal form. This form deals with certain type of anomaly that is not handled by 3NF. A 3NF table which does not have multiple overlapping candidate keys is said to be in BCNF. For a table to be in BCNF, following conditions must be satisfied:

1. R must be in 3rd Normal Form
2. For each functional dependency ($X \twoheadrightarrow Y$), X should be a super Key.

2.10 TOOLS AND ENVIRONMENTS USED

Our whole system is designed via PHP and MySQL database. This is web-based application so if we host it in some live web server then only system requirement will be internet and a device. Though the listed items are used to build this project:

Xampp

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows and x64 package of macOS and Linux.

XAMPP is an abbreviation where *X stands for Cross-Platform, A stands for Apache, M stands for MYSQL and the P's stand for PHP and Perl*, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executable along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies, Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL. The detailed description of these components is given below.

Many other components are also part of this collection of software and are explained below.

Cross-Platform: Different local systems have different configurations of operating systems installed in it. The component of cross-platform has been included to increase the utility and audience for this package of Apache distributions. It supports various platforms such as packages of Windows, Linus, and MAC OS.

Apache: It is an HTTP a cross-platform web server. It is used worldwide for delivering web content. The server application has made free for installation and used for the community of developers under the aegis of Apache Software Foundation. The remote server of Apache delivers the requested files, images, and other documents to the user.

Maria DB: Originally, MySQL DBMS was a part of XAMPP, but now it has been replaced by Maria DB. It is one of the most widely used relational DBMS, developed by MySQL. It offers online services of data storage, manipulation, retrieval, arrangement, and deletion.

PHP: It is the backend scripting language primarily used for web development. PHP allows users to create dynamic websites and applications. It can be installed on every platform and supports a variety of database management systems. It was implemented using C language. PHP stands for **Hypertext Processor**. It is said to be derived from Personal Home Page tools, which explains its simplicity and functionality.

Perl: It is a combination of two high-level dynamic languages, namely Perl 5 and Perl 6. Perl can be applied for finding solutions for problems based on system administration, web development, and networking. Perl allows its users to program dynamic web applications. It is very flexible and robust.

PhpMyAdmin: It is a tool used for dealing with Maria DB. Administration of DBMS is its main role.

OpenSSL: It is the open-source implementation of the Secure Socket Layer Protocol and Transport Layer Protocol.

XAMPP Control Panel: It is a panel that helps to operate and regulate upon other components of the XAMPP. A detailed description of the control panel will be done in the next section of the tutorial.

Webalizer: It is a Web Analytics software solution used for User logs and provide details about the usage.

Mercury: It is a mail transport system. It is a mail server, which helps to manage the mails across the web.

Tomcat: It is a servlet based on JAVA to provide JAVA functionalities.

Filezilla: It is a File Transfer Protocol Server, which supports and eases the transfer operations performed on files.

MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons-

- MySQL is released under an open-source license. So you have nothing to pay to use it.
- MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and works well even with large data sets.
- MySQL is very friendly to PHP, the most appreciated language for web development.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
- MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

PHP (Hypertext Preprocessor)

The PHP Hypertext Preprocessor (PHP) is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP. PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

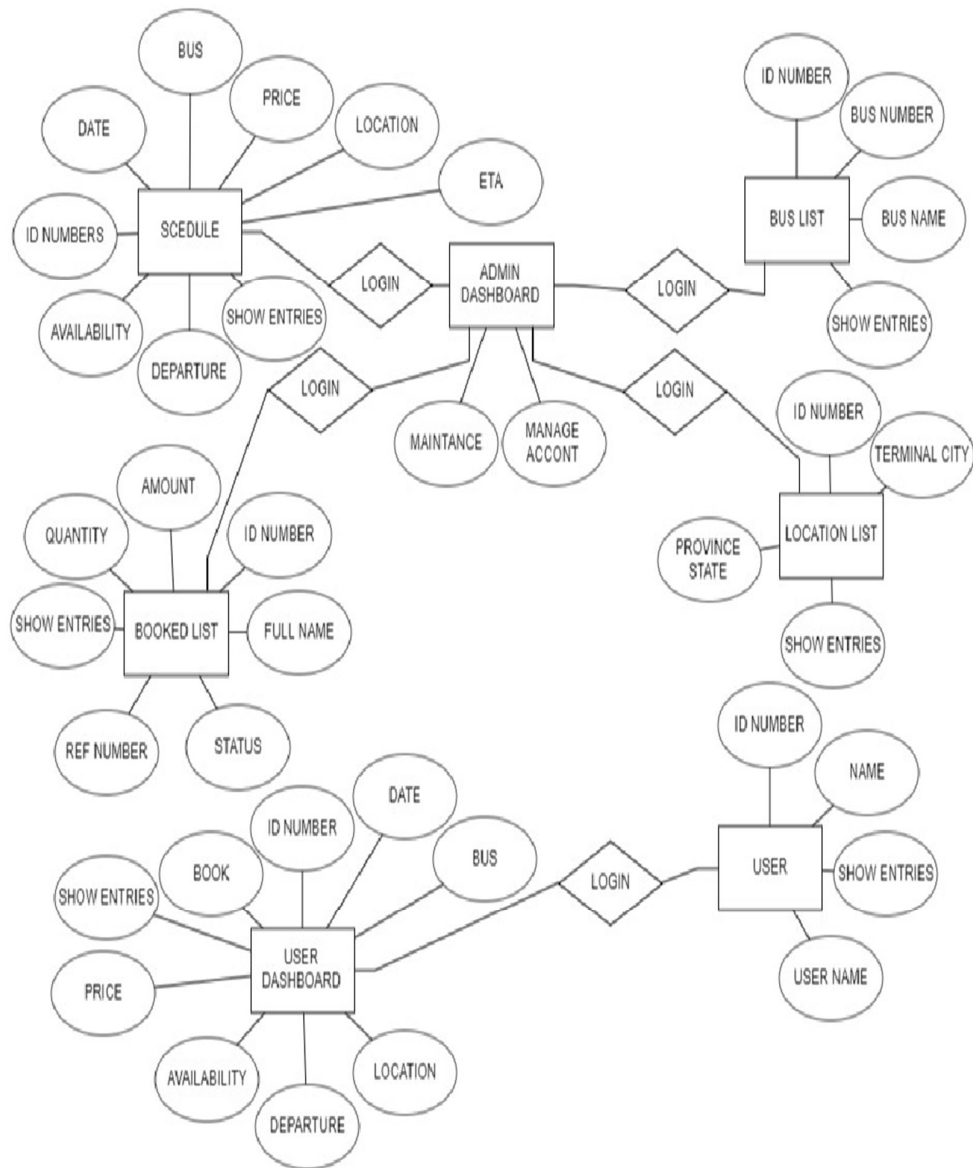
PHP is a must for students and working professionals to become a great Software Engineer especially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

- PHP is a recursive acronym for "PHP: Hypertext Pre-processor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the UNIX side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

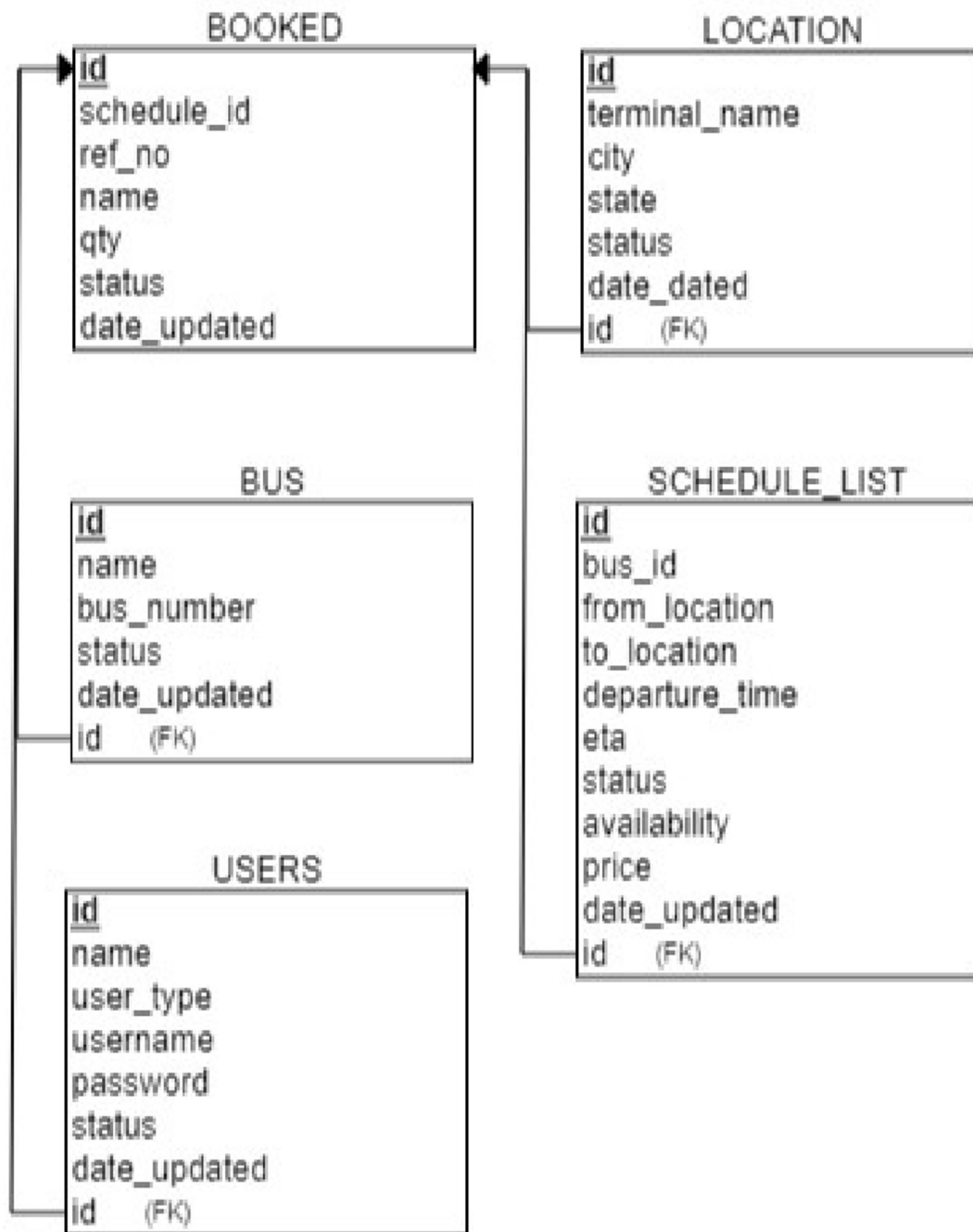
CHAPTER 3

PROPOSED WORK

3.1 ER MODEL



3.2 RELATIONAL MODEL



3.3 NORMALIZATION

The normalization process was proposed by Codd, it takes a relation schema through a series of tests to certify whether it satisfies a certain normal form. The process proceeds in a top-down fashion by evaluating each relation against the criteria for normal forms and decomposing relations as necessary can thus be considered a relational design by analysis. Normalization of data can be looked upon as a process of analyzing the given relation schemas based on their Functional Dependencies and primary keys to achieve the desirable properties of:

- Minimizing redundancy.
- Minimizing the insertion, deletion, and update anomalies.

BOOKED

id	Schedule_id	Ref_no	name	qty	status	Date_updated
↑	↑	↑	↑	↑		↑

FD1: {id} {name,, schedule id, ref_no, qty, Status,date_updated }

- The BOOKED relation is in 1NF because there are no multivalued attributes in the relationalschema.
- It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
- It is in 3NF because there are no transitive dependencies.

BUS

id	name	Bus_number	status	Date_updated
↑	↑			

FD1: {id} {name, bus_number, status, date_updated}

Bus Booking Management

- The BUS relation is in 1NF because there are no multivalued attributes in the relational schema.
- It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
- It is in 3NF because there are no transitive dependencies.

USERS

id	name	User_type	username	password

FD1: {id} {name, user_type, username, password}

- The USERES relation is in 1NF because there are no multivalued attributes in the relational schema.
- It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
- It is in 3NF because there are no transitive dependencies.

LOCATION

id	Terminal_name	city	state	Status	Date_updated

FD1: {id} {terminal_name, city, state, status, date_updated}

- The LOCATION relation is in 1NF because there are no multivalued attributes in the

Bus Booking Management

relationalschema.

- It is in 2NF because all the attributes in the relational schema are fully functional dependent on the primary key.
- It is in 3NF because there are no transitive dependencies.

SCHEDULE_LIST

id	Bus_id	From_location	to_location	Departure_time	eta	status
↑	↑	↑	↑	↑	↑	↑

availability	price	Date_updated
↑	↑	↑

CHAPTER 4

RESULT

- Admin page

Bus Booking Management System

[Home](#)[Schedule](#)[Booked List](#)[Maintenance](#) ▾[Administrator](#) ▾

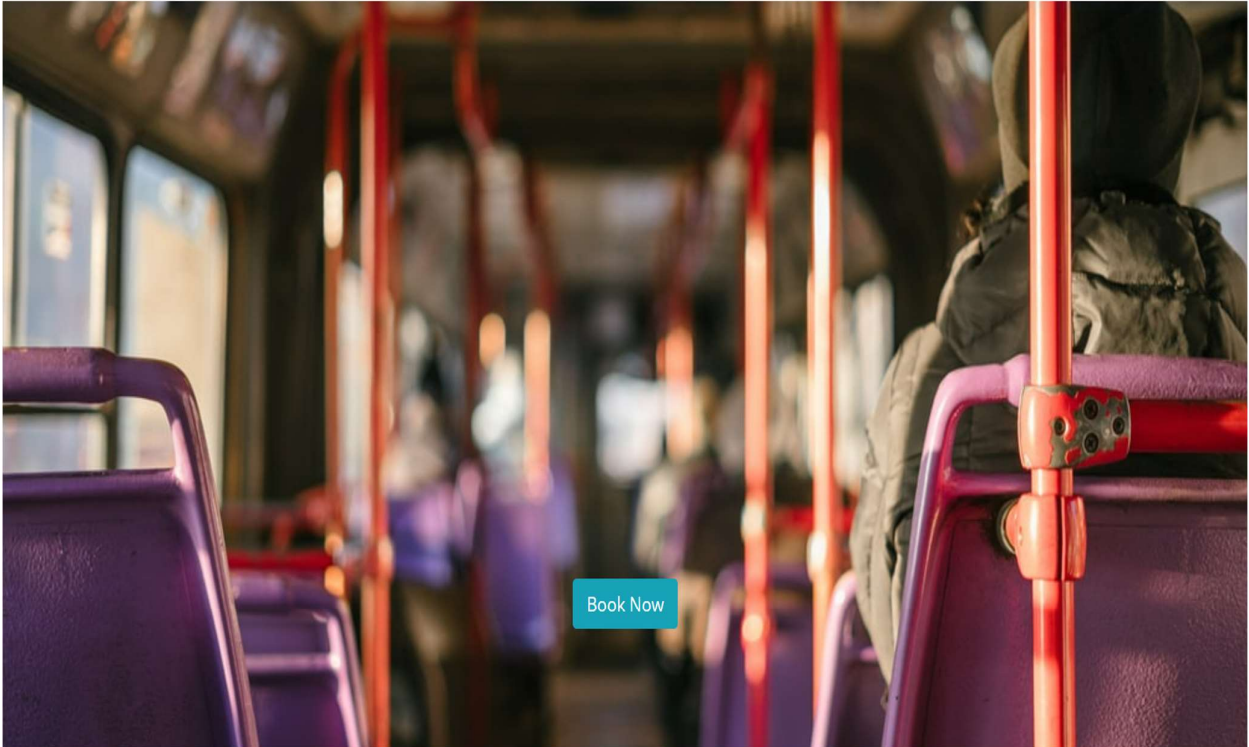
Welcome



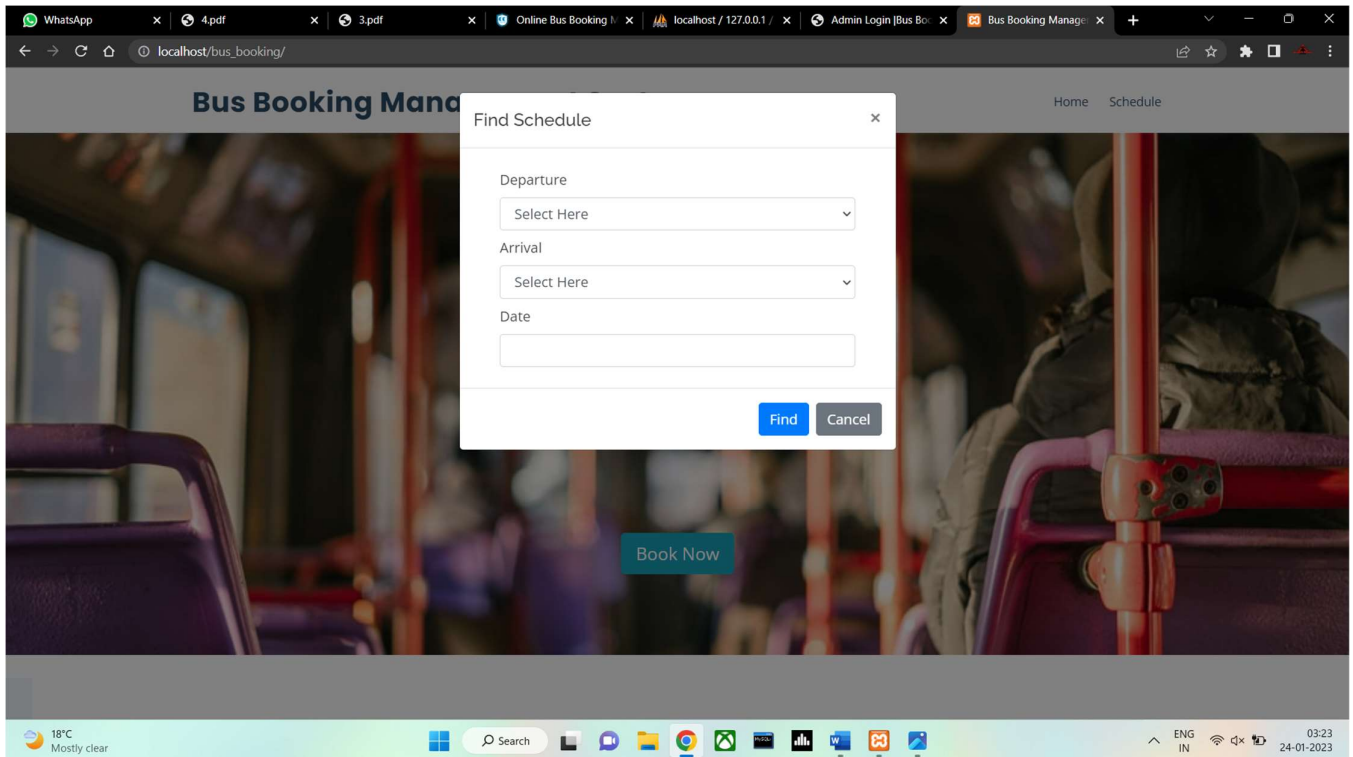
-
- **User page**

Bus Booking Management System

[Home](#) [Schedule](#)



- **Bus Booking Menu**



- **Schedule list**

Bus Booking Management System

[Home](#) [Schedule](#) [Booked List](#) [Maintenance](#) [Administrator](#)

[Add New](#)

Show entries

Search:

#	Date	Bus	Location	Departure	ETA	Availability	Price	Action
1	Jan 11, 2023	1 economy	bengaluru terminal, bengaluru, Karnataka - mysore terminal, mysore, Karnataka	11:45 PM	Jan 12,2023 11:45 PM	30	250	Edit Delete
2	Jan 11, 2023	2 sleeper	bengaluru terminal, bengaluru, Karnataka - mysore terminal, mysore, Karnataka	11:45 PM	Jan 12,2023 11:45 PM	30	500	Edit Delete
3	Jan 11, 2023	3 ac-sleeper	bengaluru terminal, bengaluru, Karnataka - mysore terminal, mysore, Karnataka	11:46 PM	Jan 12,2023 11:46 PM	30	750	Edit Delete
4	Jan 11, 2023	1 economy	mysore terminal, mysore, Karnataka - bengaluru terminal, bengaluru, Karnataka	11:46 PM	Jan 12,2023 11:46 PM	30	250	Edit Delete
5	Jan 11, 2023	2 sleeper	mysore terminal, mysore, Karnataka - bengaluru terminal, bengaluru, Karnataka	11:46 PM	Jan 12,2023 11:47 PM	30	500	Edit Delete
6	Jan 11, 2023	3 ac-sleeper	mysore terminal, mysore, Karnataka - bengaluru terminal, bengaluru, Karnataka	11:47 PM	Jan 12,2023 11:47 PM	30	750	Edit Delete

Showing 1 to 6 of 6 entries

Previous Next

CHAPTER 5

CONCLUSION

Online ticket booking system is an application where the customer can book a ticket online and 24*7 hours a day from anyplace in the world. Customers can also interact with the ticket booking website to know any other details they want. Online ticket booking system has been developed successfully. System performance is also found to be satisfactory. This is a user-friendly application. Through this application, the cost can be reduced and efficiency is increased. There are several procedures that can be selected by customers. With the help of this application customers can book tickets, can know the status of a flight, bus or trains, a Source station and destination can be chosen according to their choice, can select seats, can choose the

time, and pay through the portal after reaching the station or airport. Thus online ticket booking system target internal and external audiences. Online ticket booking system is very big to maintain but it always provides excellent facilities to accomplish the goal and help to reduce a complex paperwork process through a mobile application. This can be a benefit using online ticket booking system application rather searching on several websites. With the help of online ticket booking system records are maintained and the database is updated with time to time. Through Online ticket booking system, technologies and features have been introduced.

FUTURE ENHANCEMENTS:

- To improve the interface of the application.
- It can be enhanced in such a way that, portal can have payment method
- To implement the mobile version

CHAPTER 6

BIBLIOGRAPHY

Books:

- [1] Ramez Elmasri and Shamkant B Navathe, 'Fundamental of Database Systems', Addison Wesley, 2011, Sixth Edition.
- [2] Abraham Silberschatz, Henry F. Korth, S.Sudarshan, 'Database System Concepts', Tata McGraw-Hill, 2010, Sixth Edition.
- [3] Robin Nixon, 'Learning PHP, MySQL, JavaScript, CSS & HTML5 – A Step By Step Guide To Creating Dynamic Websites', Third Edition.

Websites:

<https://www.javatpoint.com/dbms-tutorial>

<https://www.geeksforgeeks.org/dbms/?ref=lbp>

<https://www.techtarget.com/searchdatamanagement/definition/database-management-system>

<https://www.w3schools.com/html/>

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://www.w3.org/Style/CSS/Overview.en.html>

<https://www.javascript.com/>

<https://en.wikipedia.org/wiki/PHP>