

BUS MANAGEMENT SYSTEM

By
LINET M SHAJI
ROLL NO: P191314

MSCT23 – SOFTWARE ENGINEERING PROJECT REPORT

MASTER OF SCIENCE in COMPUTER SCIENCE



CENTRAL UNIVERSITY OF TAMIL NADU
THIRUVARUR – 610 005
JUNE 2020

CENTRAL UNIVERSITY OF TAMILNADU
DEPARTMENT OF COMPUTER SCIENCE

BONAFIDE CERTIFICATE

This is to certify that this Software Engineering Lab record entitled “**BUS MANAGEMENT SYSTEM**” is a bonafide record of work done by **LINET M SHAJI, Reg.No: P191314**, of I Year/ II Semester during the academic year 2019-2020 in partial fulfillment for the award of degree of Master of Computer Science of the Central University of Tamil Nadu.

This work has not been submitted elsewhere for the award of any other degree to the best of my knowledge.

Submitted for the University Practical Examination held on _____

INTERNAL EXAMINERS

DECLARATION

I hereby declare that this project titled “**BUS MANAGEMENT SYSTEM**” submitted for the course **MSCT23: Software Engineering**, Department of Computer Science, School of Mathematics and Computer Science, Central University of Tamil Nadu, Thiruvavur – 610 005, is a record of bonafide project work carried out by us under the guidance and supervision of **Dr.A.Martin**, Department of Computer Science, Central University of Tamil Nadu, Thiruvavur – 610 005. This work is original and has not been submitted, in part or full to this or any other University / Institution.

LINET M SHAJI

Place: Thiruvavur

Date:

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

ABSTRACT

1. INTRODUCTION

1.1 ABOUT THE PROJECT

2. PROBLEM DEFINITION & FEASIBILITY ANALYSIS

2.1 PROBLEM DEFINITION

2.1.1 EXISTING SYSTEM

2.1.2 DRAWBACKS OF EXISTING SYSTEM

2.1.3 PROPOSED SYSTEM

2.1.4 FEATURES OF PROPOSED SYSTEM

2.2 FEASIBILITY ANALYSIS

2.2.1 OPERATIONAL FEASIBILITY

2.2.2 TECHNICAL FEASIBILITY

2.2.3 ECONOMICAL FEASIBILITY

3. SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

3.2 SCOPE

3.3 GENERAL DESCRIPTIONS

3.3.1 PRODUCT FUNCTION

3.3.2 USER CHARACTERISTICS

3.3.3 GENERAL CONSTRAINTS

3.3.4 FUNCTIONAL REQUIREMENTS

3.3.4.1 TECHNICAL ISSUES

3.3.5 INTERFACE REQUIREMENTS

3.3.5.1 HARDWARE REQUIREMENTS

3.3.5.2 SOFTWARE REQUIREMENTS

3.3.6 OTHER FUNCTIONAL ATTRIBUTES

3.3.6.1 SECURITY

3.3.6.2 RELIABILITY

4. SYSTEM ANALYSIS

4.1 BEHAVIORAL DIAGRAMS

4.1.1 USE CASES AND USE CASE DIAGRAM

4.1.2 SEQUENCE DIAGRAM

4.1.3 ACTIVITY DIAGRAM

4.1.4 STATE DIAGRAM

4.2 STRUCTURE DIAGRAMS

4.2.1 CLASS DIAGRAM

4.2.2 PACKAGE DIAGRAM

4.2.3 DEPLOYMENT DIAGRAM

5. SYSTEM DESIGN

5.1 ARCHITECTURAL DESIGN

5.2 DATA DESIGN

5.3 USER INTERFACE DESIGN

6. IMPLEMENTATION

7. TESTING

7.1 TEST PLAN

7.2 TYPES OF TESTING

7.2.1 UNIT TESTING

7.2.3 INTEGRATION TESTING

7.2.4 VALIDATION TESTING

8. CONCLUSION

9. BIBLIOGRAPHY

10. APPENDIX

ACKNOWLEDGEMENT

“Sometimes our light goes out, but is blown into flame by another human being. Each of us owes our deepest thanks to those who have rekindled this light”.

I hereby dedicated this project to the god almighty who's abundant and grace and mercy enabled its successful completion. I would like to extend my gratitude to all the people who have inspired and motivated me to make this project success.

I express my gratitude to **Dr. Martin**, faculty of Central university of Tamilnadu, for his guidance and supervision as well as providing necessary information regarding this project and inspiration and encouragement shown to me throughout the semester.

I would like to extend my sincere thanks to my friends **Shobi** and **Midhun**, without whose support and encouragement this project would not have attained its perfection in its total. I convey my thanks to all who helped me directly and indirect to carry out the project successfully.

Linnet M Shaji

ABSTRACT

The purpose of Bus Management System is to automate the existing manual system. The aim is to automate its existing manual system by the help of computerized equipments and full-fledged computer software, fulfilling their requirements, so that their valuable data/information can be stored for a longer period with easy accessing and manipulation of the same. Basically the project describes how to manage for good performance and better services.

Objectives of Project

The main objectives of the Project are to manage the details of bus, ticket booking, employee, passenger, bus, route, daily cash collection. It manages all the information about bus, bus schedule, bus route, bus. The purpose of the project is to build an application program to reduce the manual work for managing the bus, daily cash collection, ticket booking. It tracks all the details about the booking, employee, passenger, bus, cash collection.

Functionalities provided by Bus Management System are as follows:

- Provides the searching facilities based on various factors. Such as bus, booking, employee, passenger, cash collection.
- Manage the information of ticket
- To increase efficiency of managing the bus, ticket
- Manage cash collection
- Manage employee and passenger
- Manage seats
- Manage booking
- Provides report viewing facilities of daily cash collection, feedbacks given by passenger and employee.

Modules of Bus Management System

- **Registration Module:** Manage registration details of passenger and employee
- **Login Module:** Manage login details of passenger and employee
- **Bus Management Module:** Used for adding, updating and deleting details of bus
- **Bus Route Module:** Used for managing details of route
- **Ticket Management Module:** used for managing the information and details of the ticket

- **Booking Module:** Used for managing the booking details
- **Admin Module:** Used for managing admin module
- **Passenger Module:** Used for managing passenger details
- **Employee Module:** Used for managing employee details
- **Cash collection Module:** used for managing cash collection module

CHAPTER 1

INTRODUCTION

1.1 ABOUT THE PROJECT

The “**Bus Management System**” is web based application developed using **PHP**. The “**Bus Management System**” has been developed to override the problems prevailing in the practicing manual system. This software is supported to eliminate and in some cases reduce the hardships faced by the existing system. The application is reduced as much as possible to avoid errors while entering the data. It also provides error message while entering invalid data. No formal knowledge is needed for the user to use this system. Thus by this all it provides it is user-friendly.

The “**Bus Management System**” has three modules **Admin, Employee, Passenger**. The main objectives of the project are to manage the details of bus, ticket booking, employee, passenger, bus, route, daily cash collection. It manages all the information about bus, bus schedule, bus route, bus. The purpose of the project is to build an application program to reduce the manual work for managing the bus, daily cash collection, ticket booking. It tracks all the details about the booking, employee, passenger, bus, cash collection.

CHAPTER 2

PROBLEM DEFINITION & FEASIBILITY ANALYSIS

2.1 PROBLEM DEFINITION

2.1.1 EXISTING SYSTEM

Existing system refers to the system that is being followed till now. The existing system requires more computational time, more manual calculations, and the complexity involved in selection of features is high. The other disadvantages are lack of security of data, deficiency of data accuracy, time consuming etc. To avoid all these limitations and make the working more accurately the system needs to be computerized.

2.1.2 DRAWBACKS OF EXISTING SYSTEM

The existing system has following drawbacks

- Lack of security of data.
- More man power.
- Time consuming.
- Consumes large volume of pare work.
- Needs manual calculations.
- No direct role for the higher officials.

To avoid all these limitations and make the working more accurately the system needs to be computerized.

2.1.3 PROPOSED SYSTEM

The aim of proposed system is to develop a system of improved facilities. The proposed system can overcome all the limitations of the existing system. The system provides proper security and reduces the manual work. The existing system has several disadvantages and many more difficulties to work well. The proposed system tries to eliminate or reduce these difficulties up to some extent. The proposed system will help the user to reduce the workload and mental conflict. The proposed system helps the user to work user friendly and he can easily do his jobs without time lagging.

2.1.4 ADVANTAGES OF PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has following features

- Ensure data accuracy.
- Minimize manual data entry.
- Minimum time needed for the various processing

- Greater efficiency
- Better Service
- Minimum time required
- It would also help in providing adequate data to the corporation, particularly with regard to boarding of passengers from fare stages and important.
- Easily passengers can search for bus and their ticket by providing source and destination
- Easily admin can view daily cash collection report by login.
- Easily passengers can update daily collection report by entering required details.

2.2 FEASIBILITY STUDY

A feasibility study is conducted to select the best system that meets performance requirement. This entails an identification description, and evaluation of candidate system and the selection of best system for the job. The system required performance is defined by a statement of constraints, the identification of specific system objective and a description of outputs.

The key consideration in feasibility study is:

- Economic Feasibility
- Technical Feasibility
- Operational Feasibility

2.2.1 ECONOMIC FEASIBILITY

It looks at the financial aspects of the project. It determines whether the management has enough resources and budget to invest in the proposed system and estimate time for the recovery of cost incurred. It also determines whether it is worthwhile to invest the money in the proposed project. Economic feasibility determines by the means of cost benefit analysis. The proposed system is economically feasible because we don't have to give the salary to the admin panel. As per the nature of this application we don't need to invest much more money in this system. Because it is not specified for the outside marketing world. It is only for the former students and teachers. So investing money is not needed. And one more thing, we are not applying any kind of fees like procedure to the users. So it is completely free for the usable. That is it is economically feasible for users, they don't have to spend money for being a user. This is also applicable in the case of the developing side. The less time involved also helped in its economic feasibility.

Because of initially we are planning to release online course portal in a particular subject/course. So no need for a high performing server. The backend required for storing other details is also the same database that is MYSQL. The computers in the organization are highly sophisticated and don't need extra components to load the software. Hence the organization can implement the new system without any additional expenditure. Hence, it is economically feasible.

2.2.2 TECHNICAL FEASIBILITY

It is a measure of the practicality of a specific technical solution and the availability of technical resources and expertise. The proposed system uses php as Front-end and MYSQL as Back-end tool. MYSQL is a popular tool used to design and develop database objects such as table view, indexes. The above tools are readily available, easy to work with and widely used for developing commercial application.

Hardware used in developing this project are Intel CORE i3 CPU @ 2.00GHz, 4.00 GB RAM, 64-bit operating system, x64-based processor, 1 TB Hard disk. This hardware's were already available on the existing computer system. The software Windows 10 is installed on the existing computer system. So no additional hardware and software were required to purchase and it is feasible. As the users increase we have to buy a new powerful server. IN initial stage is not required.

2.2.3 OPERATIONAL FEASIBILITY

The system will be used if it is developed well then be resistance for users that undetermined. No major training and new skills are required as it is based on DBMS model. It will help in the time saving and fast processing and dispersal of user request and applications. New product will provide all the benefits of present system with better performance. Improved information, better management and collection of reports.

User support.

User involvement in the building of present system is sought to keep in mind the user specific requirement and need. User will have control over their information. Faster and systematic processing of user request approval.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

3.1 INTRODUCTION

An SRS is basically an organization's understanding of a customer or potential client's system requirements and dependencies at a particular point in time past to any actual design or development work. Software Requirement Specification has been developed for future reference in case of any ambiguity and misunderstanding. SRS provides a detailed description of the requirement, behaviors, constraints and performance of the system.

3.2 SCOPE

The scope of this Software Requirement Specification states that the goals and objectives of the software describing it in the context of the computer based systems. This document is used to throughout the project life cycle right from the inception to the final stage of implementation. This statement defines the requirement of the systems, function and non-functional, by the users and developers. It is referred as in case any ambiguities or confusion arise, the developer is responsible for asking clarification whenever necessary and won't make any alterations without the permission of the client.

3.3 GENERAL DESCRIPTIONS

Modules of Bus Management System as follows

- **Registration Module:** Manage registration details of passenger and employee
- **Login Module:** Manage login details of passenger and employee
- **Bus Management Module:** Used for adding, updating and deleting details of bus
- **Bus Route Module:** Used for managing details of route
- **Ticket Management Module:** used for managing the information and details of the ticket
- **Booking Module:** Used for managing the booking details
- **Admin Module:** Used for managing admin module
- **Passenger Module:** Used for managing passenger details
- **Employee Module:** Used for managing employee details
- **Cash collection Module:** used for managing cash collection module
- **Expense Module:** used for managing expense

- **Invoice Module:** used for managing invoice
- **History Module:** used for managing history

3.3.1 PRODUCT FUNCTION

- Provides the searching facilities based on various factors. Such as bus, booking, employee, passenger, cash collection.
- Manage the information of ticket
- To increase efficiency of managing the bus, ticket
- Manage cash collection
- Manage employee and passenger
- Manage expense and invoice
- Manage booking
- Provides report viewing facilities of daily cash collection, feedbacks given by passenger and employee.

3.3.2 USER CHARACTERISTICS

Three types of users are there such as **Admin, Employee and Public user**

1.Admin

- Manage cash collection
- Manage employee and passenger
- Manage seats
- Manage booking
- View daily cash collection
- View expense
- View invoice
- Manage history of invoice, cash collection, expense

2. Employee

- Register
- Login
- Update cash collection as per as day
- Update expense as per as day
- Update invoice as per as day

3. Public user

- Search bus

3.3.4 FUNCTIONAL REQUIREMENTS

Functional requirements define the fundamental actions that system must perform. The functional requirements for the system are divided into three cash collection, expense, invoice.

3.3.4.1 DISTRIBUTED DATABASE:

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network.

3.3.4.2 CLIENT-SERVER SYTEM

The term client/server refers primarily to an architecture or logical division of responsibilities, the client is the application (also known as the front-end), and the server is the DBMS (also known as the back-end).

A client/server system is a distributed system in which,

- Some sites are client sites and others are server sites.
- All the data resides at the server sites.
- All applications execute at the client sites.

In this system the front end is designed using of **HTML, PHP, CSS Java script** and the back end is designed using **MySQL** which is used to design the databases.

3.3.5 INTERFACE REQUIREMENTS

The Interface Requirement specifies requirements on a given external interface required of a System of Interest (SoI). These interface requirements have the same status as any other requirements, i.e., they represent characteristics required of any SoI solution for that solution to be acceptable

3.3.5.1 HARDWARE REQUIREMENTS

- Processor : Intel® Core™ 2 Duo
- RAM : 2 GB
- Hard disk : 320 GB
- Monitor : 14 inch
- Keyboard : Standard Keyboard
- Mouse : Optical Mouse
- A browser which supports CGI, HTML & Javascript.

3.3.5.2 SOFTWARE SPECIFICATION

- Visual Studio code
- Windows 10
- PHP-XAMPP SERVER

The front end is designed using of **HTML, PHP, CSS Java script** and the back end is designed using **MySQL** which is used to design the databases.

3.3.5 OTHER FUNCTIONAL ATTRIBUTE

3.3.6.1 SECURITY

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

Employee have to login to the system with username and password to update details and the public user cannot have access to the system, public user can only search the bus, to book the bus user must have to register with required details.

3.3.6.2 RELIABILITY

This specifies the factors required to establish the required reliability of the software system at time of delivery.

CHAPTER 4

SYSTEM ANALYSIS

4.1 BEHAVIORAL DIAGRAMS

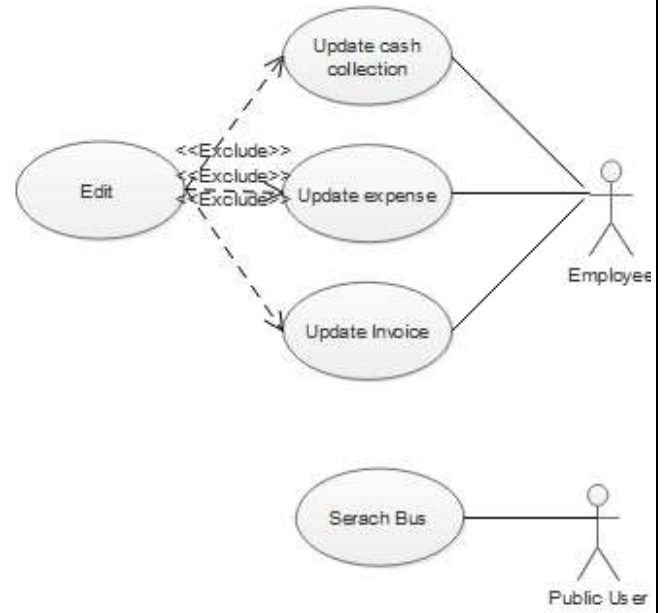
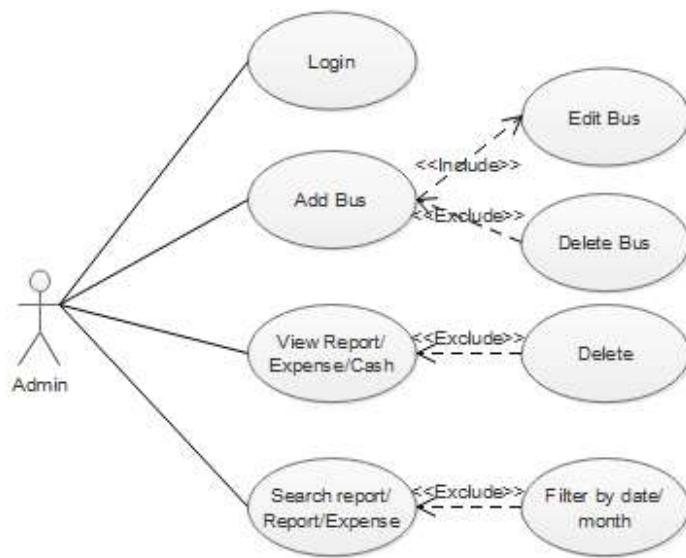
UML (Unified Modeling diagram) behavioral diagrams visualize,specify,construct and document the dynamic aspects of a system. The behavioral diagrams are categorized as follows: Use case Diagram, Sequence Diagram, Activity Diagram and State Machine Diagram. There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry. One of the major tasks here is to understand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that might have been developed by another programmer This project makes a novel attempt to address the issue of program analysis and generation of diagrams, which can depict the structure of a program in a better way. Today, UML is being considered as an industrial standard for software engineering design process.

It essential provides several diagramming tools that can express different aspects/ characteristics of program such as

- Use cases: Elicit requirement from users in meaningful chunks. Construction planning is built around delivering some use cases n each interaction basis for system testing.
- Class diagrams: shows static structure of concepts, types and class. Concepts how users think about the world; type shows interfaces of software components; classes shows implementation of software components.
- State diagram: show how single object behaves across many use cases.
- Activity Diagram: shows behavior with control structure. Can show many objects over many uses, many object in single use case, or implementation methods encourage parallel behavior etc. The end-product of this project is a comprehensive tool that can parse any vb.net program and extract most of the object oriented features inherent in the program such as polymorphism, inheritance, encapsulation and abstraction.

4.1.1 USE CASE DIAGRAM

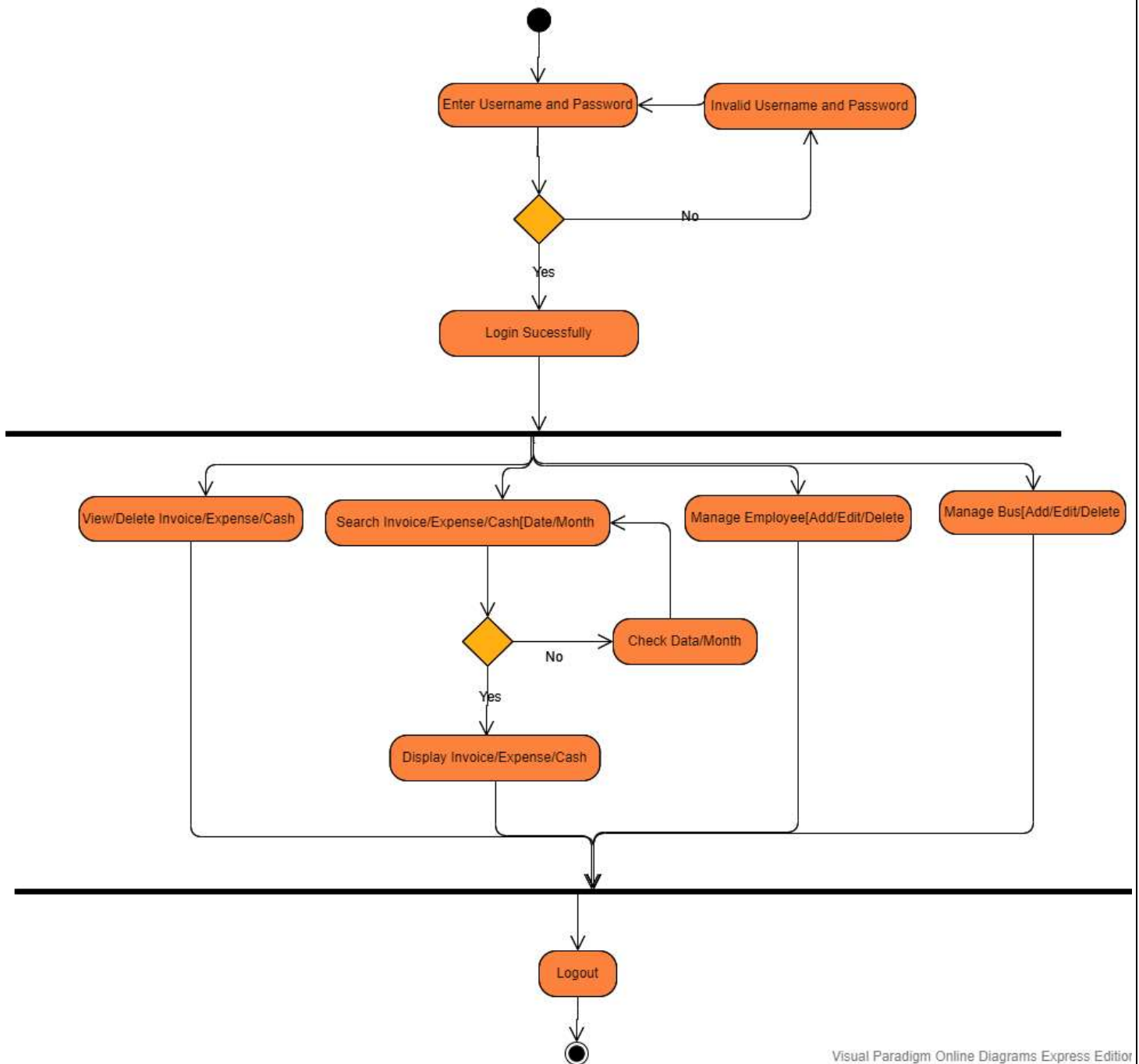
Elicit the requirements from users in meaningful chunks. Construction planning is built around delivering some use cases in each interaction basis for system testing.



4.1.2 ACTIVITY DIAGRAM.

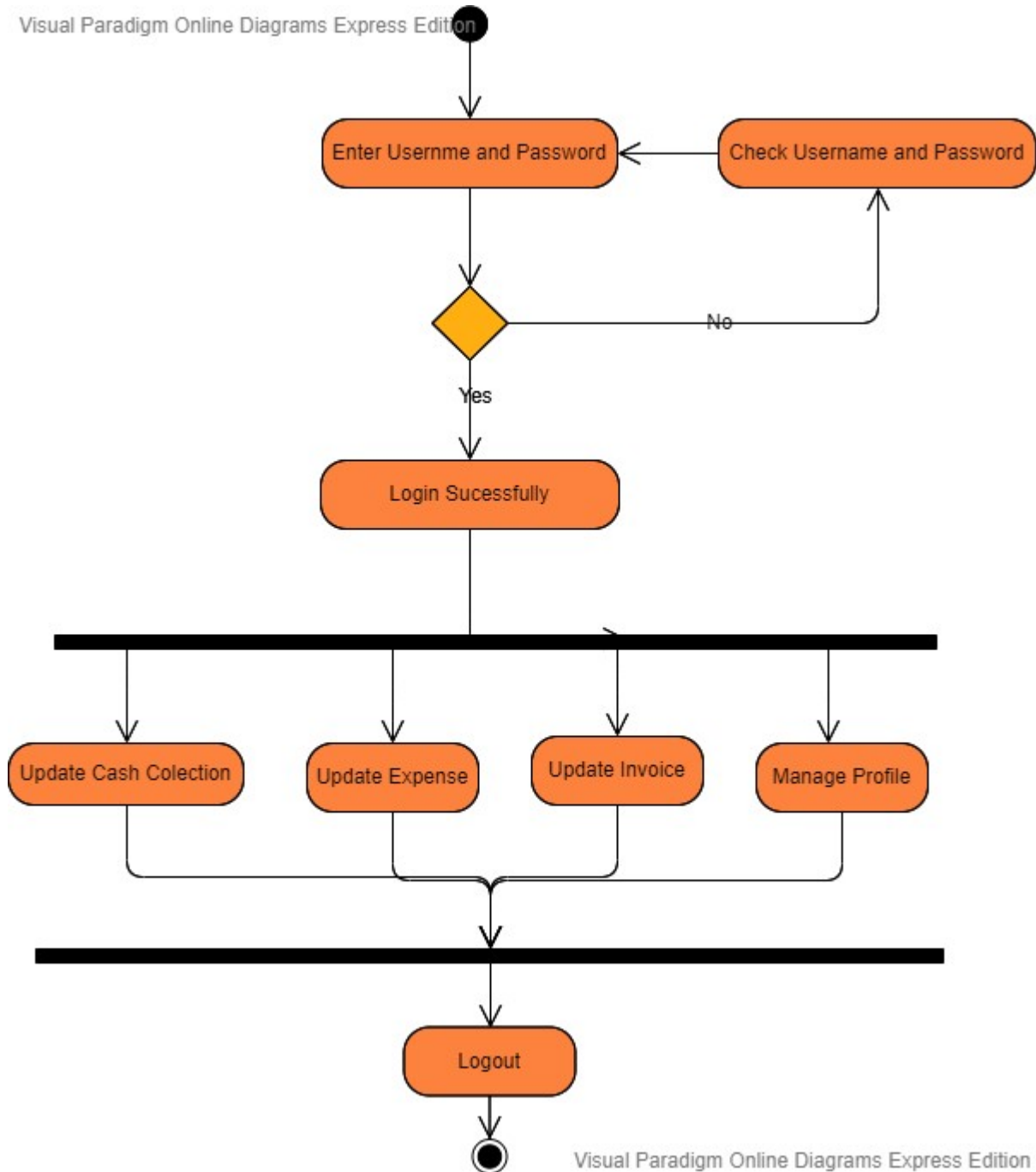
Shows behavior of control structure. Can show many objects over many users, many objects in single use case or implementations methods encourage parallel behavior, etc.

Activity Diagram for Admin



Activity Diagram for Employee

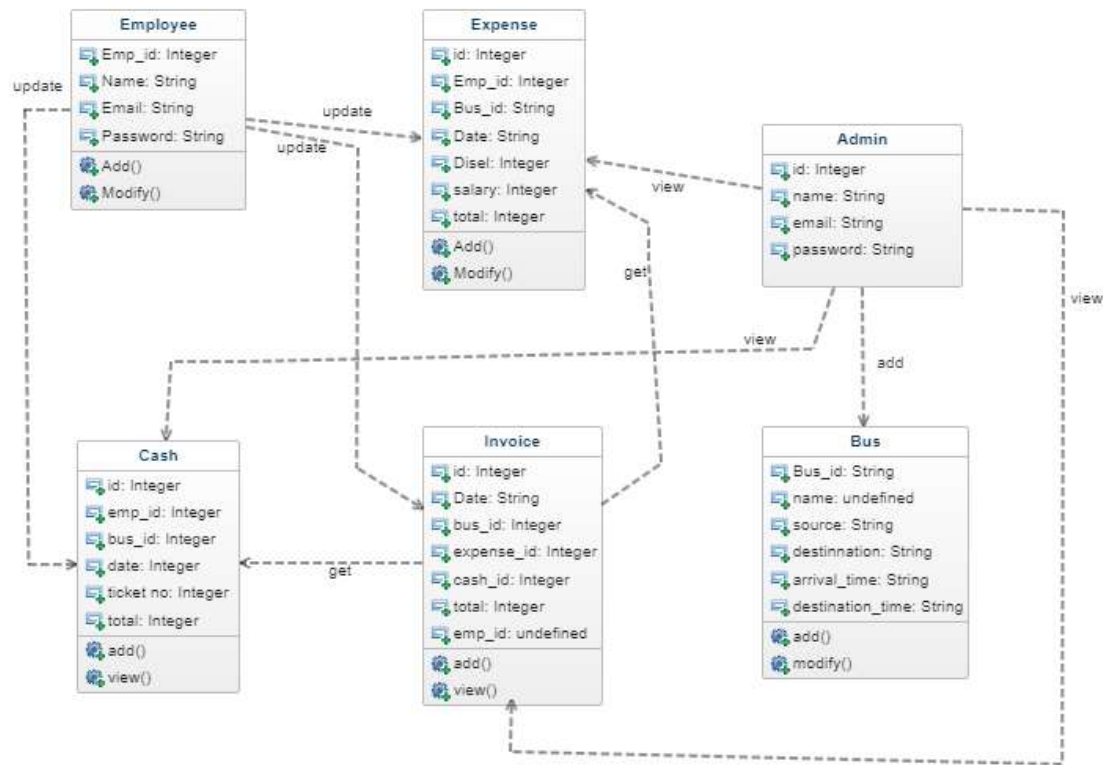
Visual Paradigm Online Diagrams Express Edition



Visual Paradigm Online Diagrams Express Edition

4.1.3 CLASS DIAGRAM

Shows static structure of concepts, types and class. Concepts contains how users think about the world, type shows interfaces of software components, classes shows implementation of software components.



CHAPTER 5

SOFTWARE DESIGN

5.1 ARCHITECTURAL DESIGN

Once interactions between the system and its environment have been understood, you use this information for designing the system architecture.

- You identify the major components that make up the system and their interactions, and then may organize the components using an architectural pattern such as a layered or client-server model.
- The weather station is composed of independent subsystems that communicate by broadcasting messages on a common infrastructure

5.2 DATA DESIGN

Data design is the first design activity, which results in less complex, modular and efficient program structure. The information domain model developed during analysis phase is transformed into data structures needed for implementing the software. The data objects, attributes, and relationships depicted in entity relationship diagrams and the information stored in data dictionary provide a base for data design activity. During the data design process, data types are specified along with the integrity rules required for the data. For specifying and designing efficient data structures, some principles should be followed. These principles are listed below.

- The data structures needed for implementing the software as well-as the operations that can be applied on them should be identified.
- A data dictionary should be developed to depict how different data objects interact with each other and what constraints are to be imposed on the elements of data structure.
- Stepwise refinement should be used in data design process and detailed design decisions should be made later in the process.
- Only those modules that need to access data stored in a data structure directly should be aware of the representation of the data structure.
- A library containing the set of useful data structures along with the operations that can be performed on them should be maintained.
- Language used for developing the system should support abstract data types.

5.3 USER INTERFACE DESIGN

User interface is the front-end application view to which user interacts in order to use the software. User can manipulate and control the software as well as hardware by means of user interface. Today, user interface is found at almost every place where digital technology exists, right from computers, mobile phones, cars, music players, airplanes, ships etc.

User interface is part of software and is designed such a way that it is expected to provide the user insight of the software. UI provides fundamental platform for human-computer interaction.

UI can be graphical, text-based, audio-video based, depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

- Attractive
- Simple to use
- Responsive in short time
- Clear to understand
- Consistent on all interfacing screens

UI is broadly divided into two categories:

- Command Line Interface
- Graphical User Interface

Command Line Interface (CLI)

CLI has been a great tool of interaction with computers until the video display monitors came into existence. CLI is first choice of many technical users and programmers. CLI is minimum interface a software can provide to its users.

CLI provides a command prompt, the place where the user types the command and feeds to the system. The user needs to remember the syntax of command and its use. Earlier CLI were not programmed to handle the user errors effectively.

A command is a text-based reference to set of instructions, which are expected to be executed by the system. There are methods like macros, scripts that make it easy for the user to operate.

CLI uses less amount of computer resource as compared to GUI.

Graphical User Interface

Graphical User Interface provides the user graphical means to interact with the system. GUI can be combination of both hardware and software. Using GUI, user interprets the software.

Typically, GUI is more resource consuming than that of CLI. With advancing technology, the programmers and designers create complex GUI designs that work with more efficiency, accuracy and speed.

GUI Elements

GUI provides a set of components to interact with software or hardware. Every graphical component provides a way to work with the system. A GUI system has following elements such as:

- **Window** - An area where contents of application are displayed. Contents in a window can be displayed in the form of icons or lists, if the window represents file structure. It is easier for a user to navigate in the file system in an exploring window. Windows can be minimized, resized or maximized to the size of screen. They can be moved anywhere on the screen. A window may contain another window of the same application, called child window.
- **Tabs** - If an application allows executing multiple instances of itself, they appear on the screen as separate windows. **Tabbed Document Interface** has come up to open multiple documents in the same window. This interface also helps in viewing preference panel in application. All modern web-browsers use this feature.
- **Menu** - Menu is an array of standard commands, grouped together and placed at a visible place (usually top) inside the application window. The menu can be programmed to appear or hide on mouse clicks.
- **Icon** - An icon is small picture representing an associated application. When these icons are clicked or double clicked, the application window is opened. Icon displays application and programs installed on a system in the form of small pictures.
- **Cursor** - Interacting devices such as mouse, touch pad, digital pen are represented in GUI as cursors. On screen cursor follows the instructions from hardware in almost real-time. Cursors are also named pointers in GUI systems. They are used to select menus, windows and other application features.

CHAPTER 6

IMPLEMENTATION

Implementation is the process of realizing the design as a program. After you have carefully planned your project, you will be ready to start the project implementation phase, the third phase of the project management life cycle. The implementation phase involves putting the project plan into action. It's here that the project manager will coordinate and direct project resources to meet the objectives of the project plan. As the project unfolds, it's the project manager's job to direct and manage each activity, every step of the way. That's what happens in the implementation phase of the project life cycle: you follow the plan you've put together and handle any problems that come up.

The implementation phase is where you and your project team actually do the project work to produce the deliverables. The word "deliverable" means anything your project delivers. The deliverables for your project include all of the products or services that you and your team are performing for the client, customer, or sponsor, including all the project management documents that you put together.

Implementation Issues

Focus here is not on programming, although this is obviously important, but on other implementation issues that are often not covered in programming texts:

- **Reuse** - Most modern software is constructed by reusing existing components or systems. When you are developing software, you should make as much use as possible of existing code.
- **Configuration management** - During the development process, you have to keep track of the many different versions of each software component in a configuration management system.
- **Host-target development** - Production software does not usually execute on the same computer as the software development environment. Rather, you develop it on one computer (the host system) and execute it on a separate computer (the target system).

Tools used for implementation

- **Wamp server** - WampServer is a Windows web development environment. It allows you to create web applications with Apache2, PHP and a MySQL database. Alongside, PhpMyAdmin allows you to manage easily your database.
- **Visual Studio** - Visual Studio Code is a lightweight but powerful source code editor which runs on your desktop and is available for Windows, macOS and Linux. It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).



CHAPTER 7

SYSTEM TESTING

7.1 TEST PLAN

A software test plan is a document describing the testing scope and activities. It is the basis formally testing any software/product in a project.

A test is a set of data that the system will process as normal input. However, data are created with the express intent of determining whether the system process them correctly. There are two general strategies for testing software. Software Testing have a test cases that result in executing every instruction in the module; but is every path through the program is tested.

7.2 LEVELS OF TESTING

Systems are not designed as entire systems nor are they tested as single terms. So, performing of both Unit and System testing is essential.

7.2.1 UNIT TESTING

Unit testing, we have to test the programs making up the system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses on the module independent of none another. In order locate errors, this enables, to detect errors in coding and logic that are contained within that module alone.

The primary goal of unit testing is to take the smallest piece of testable software in the application. Isolate it from the reminder of the code, and determine whether it behaves exactly as you between modules. Unit testing has proven its values in that a large percentage of defects are identified during its use.

The most common approach to unit testing requires drives and stubs to be written. The driver stimulates a calling unit and the stub simulates a called unit. The investment of developer time in this activity sometimes results in demoting unit testing to a lower level of priority and that is almost always a mistake. Even though the drivers and stubs cost time and money. Unit testing provides some undeniable advantage. It allows for automation of the testing process reduces difficulties of discovering error contained in more complex piece of the application, and test coverage is often enhanced because attention is given to each unit.

7.2.2 INTEGRATION TESTING

Integration testing is a systematic testing for conducting tests to uncover errors associated within the interface. The objective is to take unit tested modules and build a program structure. Here, correction is difficult because the vast expense of the entire program complicate the isolation of causes.

Integration testing is logical extension of unit testing. In its simplest form, to units that have already been tested are combined into a component and the interface between them is tested. A component, in this sense, refers to an integrated aggregate of more than one unit. In a realistic scenario, many units are combined into components which are in turn aggregated into even larger parts of the program. The idea is to test combination of pieces and eventually expand the process are tested together. Beyond that, if the program is composed of more than one process, they should be tested in pairs rather than all at once.

Integration testing identifies problems that occur when units are combined. By using a test plan that requires you to test each unit and ensure the viability of each before combining units, you know method reduces the number of possibilities to a far simpler level of analysis.

7.2.3 SYSTEM TESTING

In this phase, the entire software system was tested. After integration testing, the entire software was tested against various clients. The software has been tested for its functionality as well as limitation. The various interfaces developed were thoroughly debugged and were found to be working correctly.

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of black box testing, and as such, should require no knowledge of the inner design of the code or logic. As a rule, system testing takes, as its input, all of the "integrated" software components that have successfully passed integration testing and also the software system itself integrated with any applicable hardware systems. The purpose of integrated test is to detect any inconsistency between the software units that are integrated together called assemblages or between any of the assemblages and hardware. System testing is a more limiting type of testing; It seeks to detect defects both within the "inter assemblages" and also within the system as whole. System testing is performed on the entire system requirement specification and test not only the design, but also the software requirement specification (SRS). System testing is an investigatory testing phase, where the focus is to have almost a destructive attitude behavior and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirement specification

7.2.4 USER ACCEPTANCE TESTING

User acceptance testing is the software testing process where system tested for acceptability & validates the end to end business flow. Such type of testing is executed by client in separate environment & confirms whether system meets the requirements as per requirements specification or not. User Acceptance testing also known as Customer Acceptance Testing(CAT), if the system is being built or developed by an external supplier.

CHAPTER 8

CONCLUSION

Bus Management System is web based software developed using php for managing bus and cash collection. It eases the tasks of the administrator and employees of bus.

The administrator creates different bus details and provide to passengers to view the bus details. He is responsible for handling most of the tasks such as managing bus, managing employee, managing route, managing history, managing feedback, managing collection, etc. Here the public user can view the bus details and in case of booking he need a account, by login only booking is allowed in this system. The employee maintains the documentation of daily cash collection, expense, invoice and also reports to the admin. And also this project gives better understanding in software engineering concepts and php. Thank you martin sir for giving this opportunity to learn more.

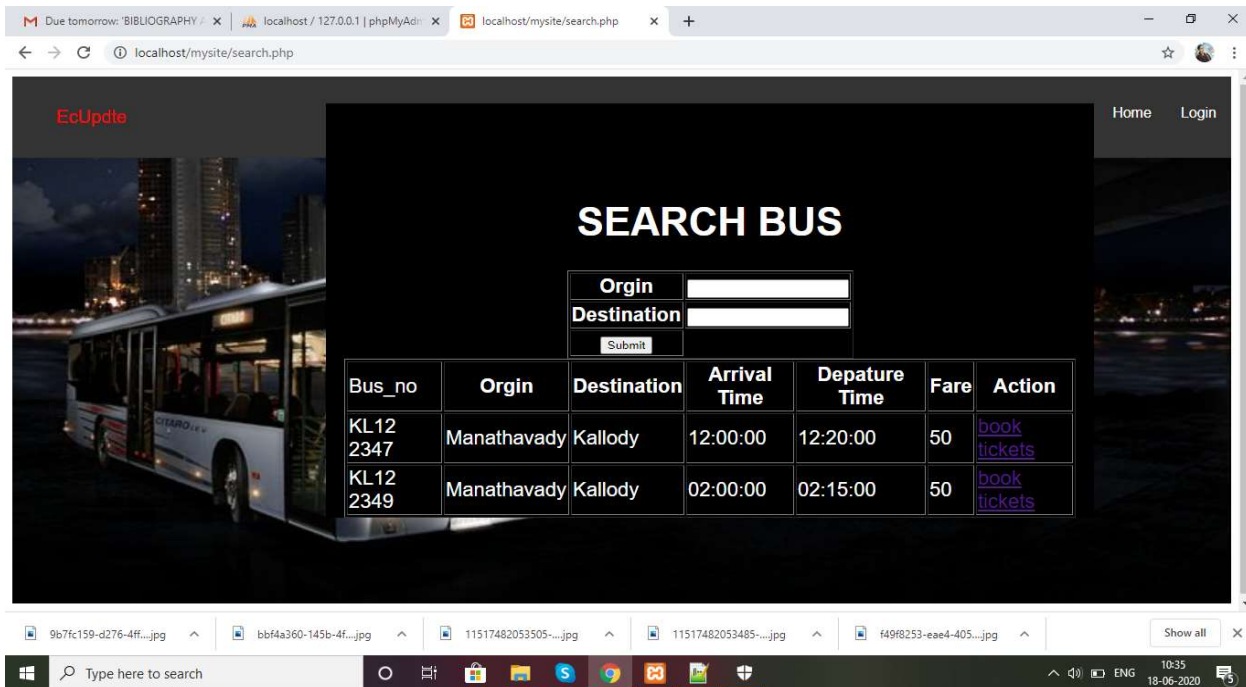
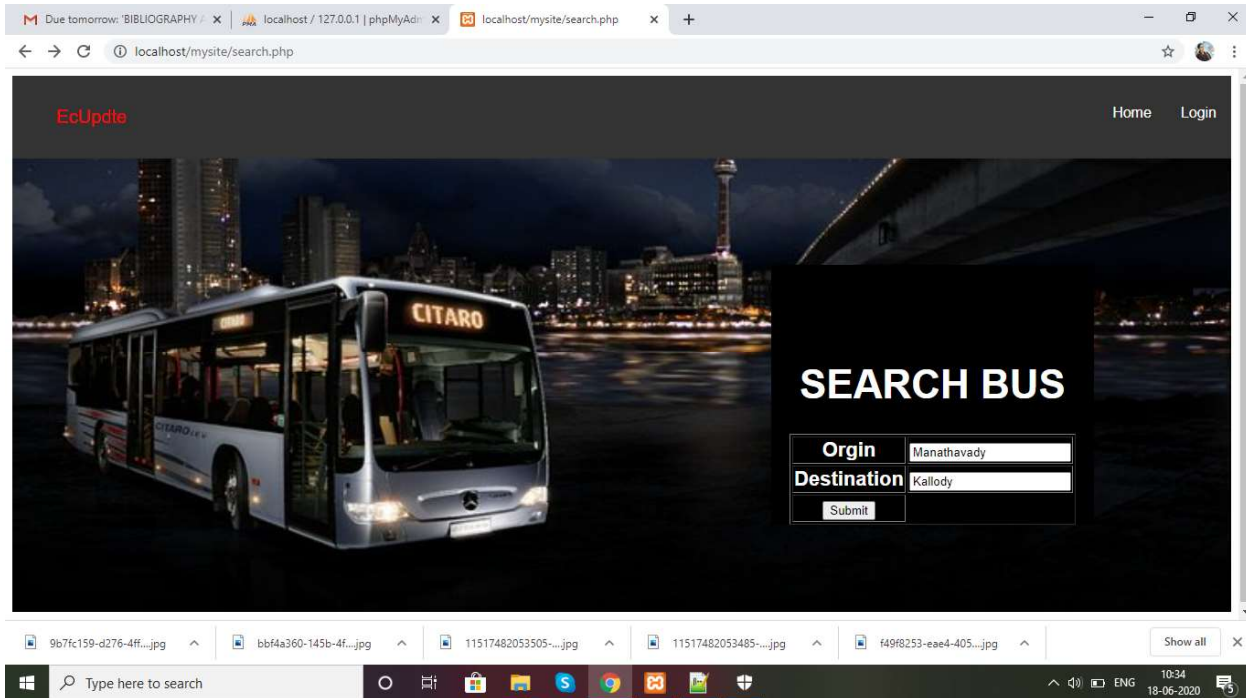
BIBLIOGRAPHY

This section describes the documents and other sources from which information was gathered.

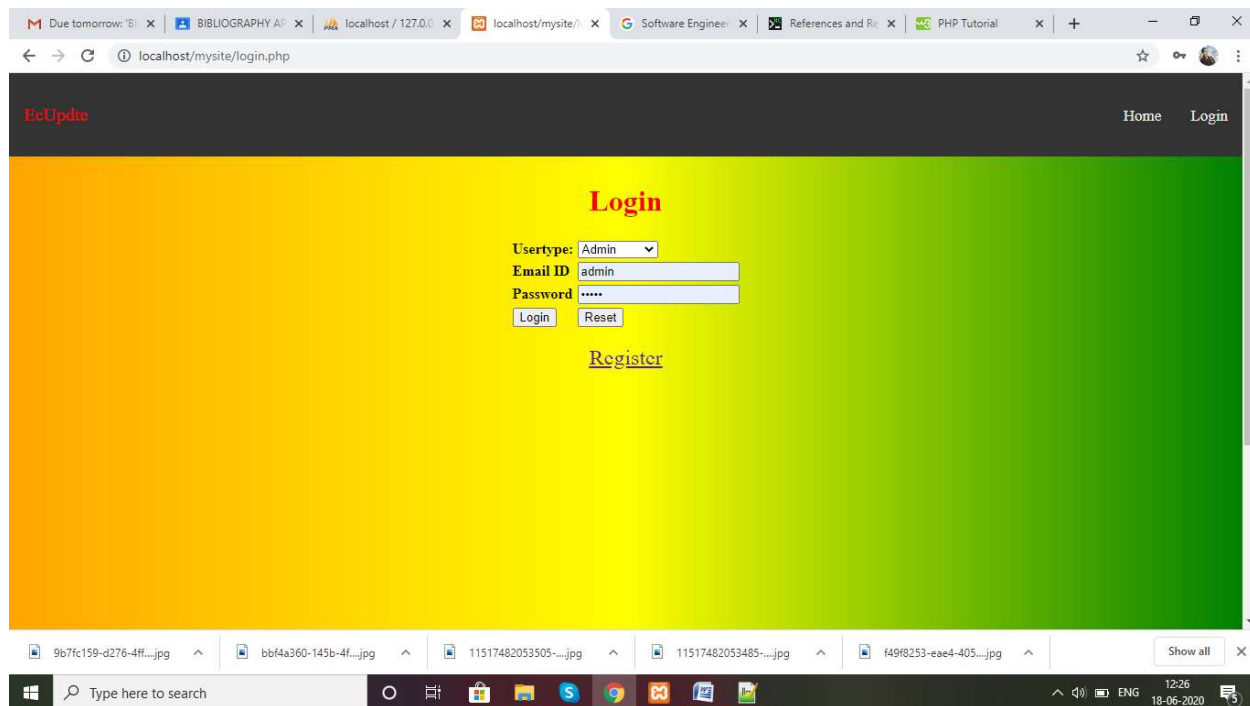
1. <https://www.w3schools.com/js/default.asp>
2. <https://www.w3schools.com/css/default.asp>
3. <https://www.w3schools.com/php/default.asp>
4. <https://www.w3schools.com/html/default.asp>
5. Roger S Pressman, “Software Engineering A Practitioner’s Approach”.

APPENDIX A – SCREEN SHOTS

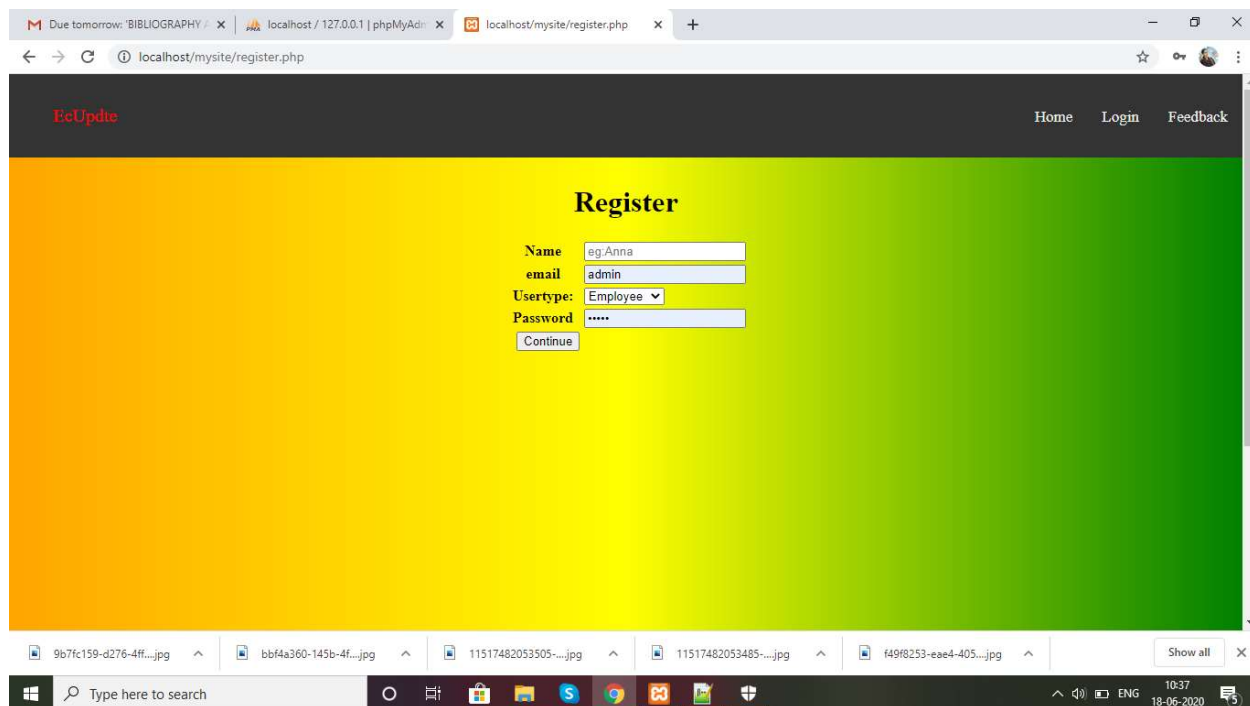
1. Search Bus



2. Login



3. Register



4. Add bus

The screenshot shows a web browser window with the URL `localhost/mysite/add_bus.php`. The page has a dark header with the logo 'EcUpdte' and navigation links: Home, View, Add, Feedback, Logout. The main content area has a yellow-to-green gradient background. The title 'ADD BUS' is centered. Below it is a form with the following fields: Bus NO, Orgin, Destination, Arrival Time, Depature Time, Fare, and a Submit button. The Windows taskbar at the bottom shows the time as 12:28 on 18-06-2020.

Due tomorrow: 'Bil' x BIBLIOGRAPHY A x localhost / 127.0.0.1 x localhost/mysite/ x Software Engineer x References and Re x PHP Tutorial x

localhost/mysite/add_bus.php

EcUpdte Home View Add Feedback Logout

ADD BUS

Bus NO

Orgin

Destination

Arrival Time

Depature Time

Fare

9b7fc159-d276-4ff...jpg bbf4a360-145b-4ff...jpg 11517482053505-...jpg 11517482053485-...jpg f49f8253-eae4-405-...jpg Show all x

Type here to search 12:28 18-06-2020 ENG

5.

Update

Cash

Collection

The screenshot shows a web browser window with the URL `localhost/mysite/Cash_collection.php`. The page has a dark header with the logo 'EcUpdte' and navigation links: Home, Update, Feedback, Logout. The main content area has a yellow-to-green gradient background. The title 'Update Cash collection' is centered. Below it is a form with the following fields: Enter Bus NO, Enter Ticket Amount, No. of sold, Date, starting ticket No, Ending Ticket No, Total Amount, and a Submit button. The Windows taskbar at the bottom shows the time as 12:24 on 18-06-2020.

Due tomorrow: 'Bil' x BIBLIOGRAPHY A x localhost / 127.0.0.1 x localhost/mysite/ x Software Engineer x References and Re x PHP Tutorial x

localhost/mysite/Cash_collection.php

EcUpdte Home Update Feedback Logout

Update Cash collection

Enter Bus NO:

Enter Ticket Amount

No. of sold

Date

starting ticket No

Ending Ticket No

Total Amount

9b7fc159-d276-4ff...jpg bbf4a360-145b-4ff...jpg 11517482053505-...jpg 11517482053485-...jpg f49f8253-eae4-405-...jpg Show all x

Type here to search 12:24 18-06-2020 ENG

6. Update Expense

The screenshot shows a web browser window with the URL `localhost/mysite/expense.php`. The page has a dark header with the logo 'EcUpdte' and navigation links: Home, Update, Feedback, and Logout. The main content area has a yellow-to-green gradient background. The title 'Update Expense' is centered. Below it, there is a form with the following labels and input fields:

- Bus NO:
- Date:
- Maintaince:
- Diesel:
- Driver Conductor Salary:
- Other Expenses:
- Total Expenses:

A 'Submit' button is located below the 'Total Expenses' field. The browser's taskbar at the bottom shows several open files and the system clock indicating 12:24 on 18-06-2020.

7. Update Invoice

The screenshot shows a web browser window with the URL `localhost/mysite/report.php`. The page has a dark header with the logo 'EcUpdte' and navigation links: Home, Update, Feedback, and Logout. The main content area has a yellow-to-green gradient background. The title 'Update Invoice' is centered. Below it, there is a form with the following labels and input fields:

- Enter Bus NO:
- Date:
- Total Amount(8rs):
- Total Amount(10rs):
- Total Amount(12rs):
- Total Amount(8rs + 10 rs + 12rs):
- Total Amount(Expense):
- Balance Amount:

A 'Submit' button is located below the 'Balance Amount' field. The browser's taskbar at the bottom shows several open files and the system clock indicating 12:25 on 18-06-2020.

8. View Collection

Due tomorrow: 'BIBLIOGRAPHY' x localhost / 127.0.0.1 | phpMyAdmin x localhost/mysite/view_collection x +

localhost/mysite/view_collection.php

EcUpdte Home View Add Feedback Logout

View Collection

Bus_no	Date	Amount Ticket(8rs)	Amount Ticket(10rs)	Amount Ticket(12rs)	Total Amount	Expense	Balance
bs123	29/05/2020	200	300	400	900	300	1234
KL 12 2347	29/05/2020	80	100	60	240	50	190
KL 12 2349	2020-06-04	200	300	400	900	300	600
KL 12 2347	2020-06-06	200	300	400	900	200	700
KL 12 2349	2020-06-06	400	200	400	1000	200	800

9b7fc159-d276-4ff...jpg bbf4a360-145b-4ff...jpg 11517482053505-...jpg 11517482053485-...jpg f49f8253-eae4-405-...jpg Show all x

Type here to search 11:26 18-06-2020 ENG

9. View Expense

Due tomorrow: 'BIBLIOGRAPHY' x localhost / 127.0.0.1 | phpMyAdmin x localhost/mysite/view_expense.php x +

localhost/mysite/view_expense.php

EcUpdte Home View Add Feedback Logout

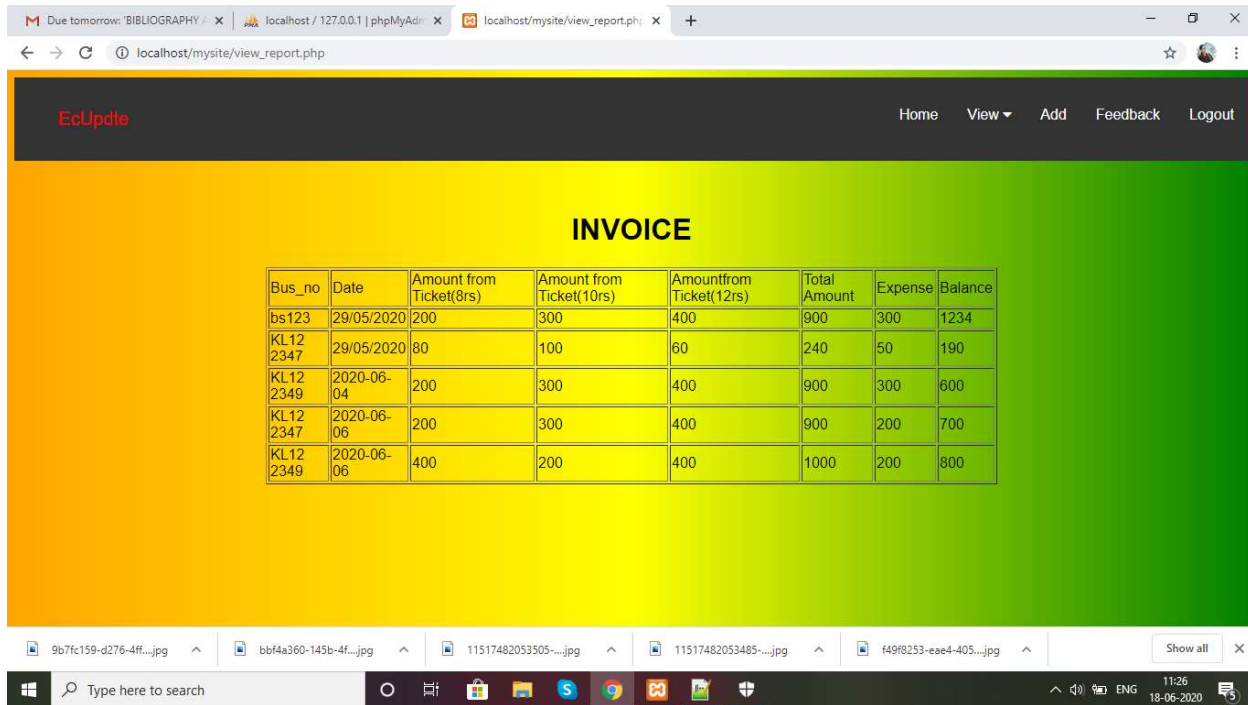
View Expense

Bus_no	Date	Maintaince	Diesel	Driver Conductor Salary	Other Expenses	Total Expenses
bs123	0000-00-00	100	100	2000	300	900
KL 12 2347	0000-00-00	100	100	2000	300	2300
KL 12 2347	2020-06-04	100	100	2000	300	2300

9b7fc159-d276-4ff...jpg bbf4a360-145b-4ff...jpg 11517482053505-...jpg 11517482053485-...jpg f49f8253-eae4-405-...jpg Show all x

Type here to search 11:26 18-06-2020 ENG

10. View Invoice



The screenshot shows a web browser window with multiple tabs. The active tab is titled 'localhost/mysite/view_report.php'. The browser's address bar shows 'localhost/mysite/view_report.php'. The web application has a dark header with the text 'EcUpdte' on the left and navigation links 'Home', 'View', 'Add', 'Feedback', and 'Logout' on the right. The main content area has a yellow-to-green gradient background and features the word 'INVOICE' in bold black text. Below the title is a table with 8 columns: Bus_no, Date, Amount from Ticket(8rs), Amount from Ticket(10rs), Amountfrom Ticket(12rs), Total Amount, Expense, and Balance. The table contains 6 rows of data. Below the browser window, a Windows taskbar is visible with a search bar and several application icons. The system clock shows 11:26 on 18-06-2020.

Bus_no	Date	Amount from Ticket(8rs)	Amount from Ticket(10rs)	Amountfrom Ticket(12rs)	Total Amount	Expense	Balance
bs123	29/05/2020	200	300	400	900	300	1234
KL 12 2347	29/05/2020	80	100	60	240	50	190
KL 12 2349	2020-06-04	200	300	400	900	300	600
KL 12 2347	2020-06-06	200	300	400	900	200	700
KL 12 2349	2020-06-06	400	200	400	1000	200	800

APPENDIX B – SAMPLE CODING

1. Login

```
<?php
$con=mysqli_connect("localhost","root","","bus");
$message = "";

if (isset($_POST['submit'])) {
    $username = $_POST['Username'];
    $password = $_POST['Password'];
    $userType = $_POST['Usertype'];

    // check passenger credentials
    if($userType == "Passenger") {
        $qry = "select * from passenger where username='{ $username}' and password='{ $password}'";
        $done = mysqli_query( $con, $qry );

        if(mysqli_num_rows($done) == 1) {
            header("Location: passenger_home.php");
        } else {
            $message = "passenger does not exist";

            header("Location: login.php");
        }
    }

    // check employee credentials
    if($userType == "Employee") {
        $qry = "select * from employee where username='{ $username}' and password='{ $password}'";
        $done = mysqli_query( $con, $qry );
        //die($qry);
        if(mysqli_num_rows($done) == 1) {
            header("Location: employee_home.php");
        } else {
            $message = "Employee does not exist";
            header("Location: login.php");
        }
    }

    //check admin credentials
    if($userType == "Admin") {
        $qry = "select * from admin where username='{ $username}' and password='{ $password}'";
        $done = mysqli_query( $con, $qry );
        //die($qry);
        if(mysqli_num_rows($done) == 1) {
            header("Location: admin_home.php");
        } else {
            $message = "Admin does not exist";
            header("Location: login.php");
        }
    }
}
```

```

    }

}

?>
<form class="form-4" method="POST">
    <center>
        <?php
            if($message) {
                echo $message;
            }
        ?>

        <table>
            <tr>
                <td style="font-weight: bold">Usertype:</td>
                <td>
                    <select name="Usertype">
                        <option value="Admin">Admin</option>
                        <option value="Employee">Employee</option>
                        <option value="Passenger">Passenger</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td style="font-weight: bold">Email ID</td>
                <td><input type="email" class="form-control" name="Username"
placeholder="eg:7510764323" required></td>
            </tr>
            <tr>
                <td style="font-weight: bold">Password</td>
                <td><input type="password" class="form-control" name="Password"
placeholder="eg:Johannes123" required></td>
            </tr>
            <tr>
                <td>
                    <input type="submit" class="btn btn-primary" name="submit" value="Login">
                </td>
                <td>
                    <input type="reset" class="btn btn-primary" name="submit" value="Reset">
                </td>
            </tr>
        </table>
    </center>
</form>
<br>
<center><a href="register.php"><u>
    <font size="5px">Register</font>

```

2. Register

```

<form class="form-4" method="POST">
    <center>
        <table>

```


VALUES

```
$qry="INSERT INTO `employee`(`name`,`username`,`password`)
('$Name','$Username','$password');"

}

$exe=mysqli_query($con,$qry) or die(mysqli_error($con));
if($exe)
{
    ?>
    <script>
        alert("Sucess");
        window.location.assign("login.php");
    </script>
<?php
    }
    else
    {
        ?>

        <script>
            die($exe);
            alert("Fail");
            window.location.assign("register.php");
        </script>

        <?php
        }
        }
    ?>
```

3. Add Bus Code

```
<div class="main">
```

```
<br><br><br><h1><center>ADD BUS</center></h1>
```

```
<form class="form-4" method ="POST" >
```

```
<center><table<p>
```

```
<tr> <th>Bus NO</th>
```

```
<td><input type="text" name="bus_no" size="20" required></td></tr>
```

```
<tr> <th>Orgin</th>
```

```
<td><input type="text" name="spoint" size="20" required><br></td></tr>
```

```
<tr><th>Destination</th>
```

```
<td><input type="text" name="epoint" size="20" required><br></td>
```

```

        </tr>

        <tr> <th>Arrival Time</th>

                <td><input type="text" name="stime" size="20" required><br></td></tr>

        <tr> <th>Depature Time</th>

                <td><input type="text" name="dtime" size="20" required><br></td></tr>

        <tr>

                <th>Fare</th>

                <td><input type="text" name="fare" size="20" required><br></td>

        </tr>

        <tr>

                <th><input type="submit" class="button send" name="submit" value="Submit"
align="center"></th></tr>

</table></center>

</form>

<?php
        $con=mysqli_connect("localhost","root","","bus");

        if(isset($_POST['submit'])){

                $Bus_no=$_POST['bus_no'];

                $Starting_point=$_POST['spoint'];

                $Destination=$_POST['epoint'];

                $Starting_time=$_POST['stime'];

                $Depature=$_POST['dtime'];

                $Fare=$_POST['fare'];

        $qry="INSERT INTO `bus`(`bus_no`, `origin`, `destination`,`arrival_time`, `departure_time`, `fare`)
VALUES('$Bus_no','$Starting_point','$Destination','$Starting_time','$Depature','$Fare')";

        $exe=mysqli_query($con,$qry) or die(mysqli_error($con));

        if($exe {

?>

<script>

```

```

        alert("Sucess");

        window.location.assign("add_bus.php");

    </script>

<?php
    }

    else{

?>

    <script>

        die($exe);

        alert("Fail");

        window.location.assign("add_bus.php");

    </script>

    <?php
        }

    }

?>

</div>

```

4. Search Bus

```

<h1><center>SEARCH BUS</center></h1>

<form class="form-4" method ="POST" >

    <center><table border="1px"><p>

        <tr>

            <th>Orgin</th>

            <td>

                <input type="text" name="spoint" size="20" required><br></td></tr>

        <tr>

            <th>Destination</th>

```

```

        <td><input type="text" name="epoint" size="20" required><br></td></tr>

    <tr>

<th><input type="submit" class="button send" name="submit" value="Submit" align="center"></th>

    </tr>

</table></center>

</form>

<?php
    $con=mysqli_connect("localhost","root","","bus");

    if(isset($_POST['submit']))
    {

        ?>

        <table width="800" border="1px" bgcolor="black">

            <tr>

                <td>Bus_no</td>

                <th>Orgin</th>

                <th>Destination</th>

                <th>Arrival Time</th>

                <th>Depature Time</th>

                <th>Fare</th>

                <th>Action</th>

            </tr>

            <?php

                $Starting_point=$_POST['spoint'];

                $Destination=$_POST['epoint'];

                $qry="SELECT * from `bus` where origin = '{ $Starting_point}' and destination = '{ $Destination}' ";

                $done = mysqli_query( $con, $qry );

                while ( $row = mysqli_fetch_array( $done ) ) {

                    ?>

```

```

        <tr>
            <td><?php echo $row['bus_no'];?></td>
            <td><?php echo $row['origin'];?></td>
            <td><?php echo $row['destination'];?></td>
            <td><?php echo $row['arrival_time'];?></td>
            <td><?php echo $row['departure_time'];?></td>
            <td><?php echo $row['fare'];?></td>
            <td><a href="login.php">book tickets</a></td>
        </tr>
    </center>
    <?php
    }
?>
</tr>
</table>
<?php
}
?>

```

5. Update Report

```
<form class="form-4" method ="POST" >
```

```
<center><table>
```

```

        <p>
            <tr>
                <th>Enter Bus NO:&nbsp;</th>
                <td><input type="text" name="bus_no" size="20" required></td>
            </tr>
            <tr>
                <th>Date</th>
                <td><input type="text" name="date" size="20"
required><br></td>
            </tr>
            <tr>
                <th>Total Amount(8rs)</th>
                <td><input type="text" name="amount8" size="20"
required><br></td></tr>

```

```

        <tr>
        <th>Total Amount(10rs)</th>
        <td><input type="text" name="amount10" size="20"
required><br></td>

        </tr>
        <tr>
        <th>Total Amount(12rs)</th>
        <td><input type="text" name="amount12" size="20"
required><br></td>

        </tr>
        <tr>
        <th>Total Amount(8rs + 10 rs + 12rs)</th>
        <td><input type="text" name="total_amount" size="20"
required><br></td>

        </tr>
        <tr>
        <th>Total Amount(Expense)</th>
        <td><input type="text" name="expense" size="20"
required><br></td>

        </tr>
        <tr>
        <th>Balance Amount</th>
        <td><input type="text" name="balance" size="20"
required><br></td>

        </tr>
        <tr>
        <th><input type="submit" class="button send" name="submit"
value="Submit" align="center"></th>
        </tr>
</table></center>
</form>
<?php
$con=mysqli_connect("localhost","root","","bus");
if(isset($_POST['submit']))
{
    $Bus_no=$_POST['bus_no'];
    $Date=$_POST['date'];
    $Amount8=$_POST['amount8'];
    $Amount10=$_POST['amount10'];
    $Amount12=$_POST['amount12'];
    $Total_Amount=$_POST['total_amount'];
    $Expense=$_POST['expense'];
    $Balance=$_POST['balance'];
    $qry="INSERT INTO `report`(`bus_no`, `date`,
`amount(8rs)`,`amount(10rs)`,`amount(12rs)`,`total_amount`,`expense`,`balance`) VALUES
('$Bus_no','$Date','$Amount8','$Amount10','$Amount12','$Total_Amount','$Expense','$Balance')";

    $exe=mysqli_query($con,$qry) or die(mysqli_error($con));
    if($exe)
    {
        ?>
</script>

```

```

                                alert("Sucess");
                                window.location.assign("Cash_collection.php");
        </script>
        <?php
            }
            else
            {
                ?>

                <script>
                die($exe);
                alert("Fail");
                window.location.assign("Cash_collection.php");
                </script>

                <?php
                }
                }
            ?>

```

6. View Report

```

<form method="post">
    <table width="800" border="1px" >
        <tr>
            <td>Bus_no</td>
            <td>Date</td>
            <td>Amount from Ticket(8rs)</td>
            <td>Amount from Ticket(10rs)</td>
            <td>Amountfrom Ticket(12rs)</td>
            <td>Total Amount</td>
            <td>Expense</td>
            <td>Balance</td>
        </tr>
        <?php
        $con=mysqli_connect("localhost","root","","bus");
        $qry = "SELECT * FROM `report`";
        $done = mysqli_query( $con, $qry );
        while ( $row = mysqli_fetch_array( $done ) ) {
            ?>
            <tr>
                <td><?php echo $row['bus_no'];?></td>
                <td><?php echo $row['date'];?></td>
                <td><?php echo $row['amount(8rs)'];?></td>
                <td><?php echo $row['amount(10rs)'];?></td>
                <td><?php echo $row['amount(12rs)'];?></td>
                <td><?php echo $row['total_amount'];?></td>
                <td><?php echo $row['expense'];?></td>
                <td><?php echo $row['balance'];?></td>
            </tr></center>
        <?php
        }
    ?>

```


Submitted By Linet M Shaji





