INTERNATIONAL UNIVERSITY OF BUSINESS AGRICULTURE AND TECHNOLOGY

REPORT ON BUS TICKET MANAGEMENT SYSTEM

Course Code: CSC-384

Course Tittle: Programming Java Lab

Submitted To

Rubayea Ferdows

Assistant Professor

Department of Computer Science & Engineering

Submitted By

Team Name: STANGER 4.0

Section: A

Name	ID	Signature
MD Shahriar Hoque	21103092	
Touhid Ahmed	21103019	
Sadia Hossain	21103009	
Nusrat Jahan	21103010	

Date of Submission: 15 April 2023

PROJECT REPORT

ON

CSC 384

PROGRAMMING JAVA LAB

CONTENT PAGE

LETTER OF TRANSMITTAL

April 2023

To

Rubayea Ferdows

Department of Computer Science & Engineering, IUBAT

Uttara, Sector 10

Dhaka -1212

Dear Madam:

Thank you for assigning us such an abrasive topic. We have tried out best to make the project successful on although there were some limitations. After completing all the jobs, we have written this report, which help you know about our project. It is expected that the report will tell that to focus on user data**.**

We are expressing our heartiest gratitude to you to go through this report and make your valuable marks. It would be very kind of you, if u evaluate our performance regarding this report.

Yours Sincerely,

		Signature
Md Shahriar Hoque	21103092	
Touhid Ahmed	21103019	
Sadia Hossain	21103009	
Nusrat jahan	21103010	

ABSTRACT

This project is mainly developed for the project requirement of Programing Java (CSC 384) course in IUBAT to ensure the development skills of knowledge, by this application system all activities of any Bus Ticket Management of an organization will be performed. Using online form people can be member here. As the purpose to keep record of bus tickets which are being booked by "Stanger bus Services Limited" may use this system to keep the actual information about Passengers who are booking and paying for bus ticket. This system have the ability to manage the schedule for this organization and also some other features like buy ticket, add buses, delete buses, and many more.

DECLARATION

I'm, Shahriar Hoque a student of BCSE in computer science & Engineering department of IUBAT with my 3 team members declaring that, this on the topic of "Bus Management System", has been prepared for the fulfilment of the project course CSC 384.

This report and the project on "Buy Ticket System" is originally prepare by Md Shahriar Hoque, Touhid Ahmed, and Sadia Hossain. All module and procedure of this project is being made after proper inspection and internet information.

It has not been prepared for any other purpose, rewards or presentations.

CHAPTER 01

INTRODUCTION

1.1 | “Objective”

All of these are symptoms of not keeping a proper "Bus Management System." These are prioritized lists of all the tasks that you need to carry out. They list everything that you have to do, with the most important tasks at the top of the list, and the least important tasks at the bottom.

By keeping such a list, you make sure that your tasks are written down all in one place so you don't forget anything important. And by prioritizing tasks, you plan the order in which you'll do them, so that you can tell what needs your immediate attention, and what you can leave until later.

Bus Management System are essential if you're going to beat work overload. When you don’t use them effectively, you'll appear unfocused and unreliable to the people around you.

1.2 | “System Benefit”

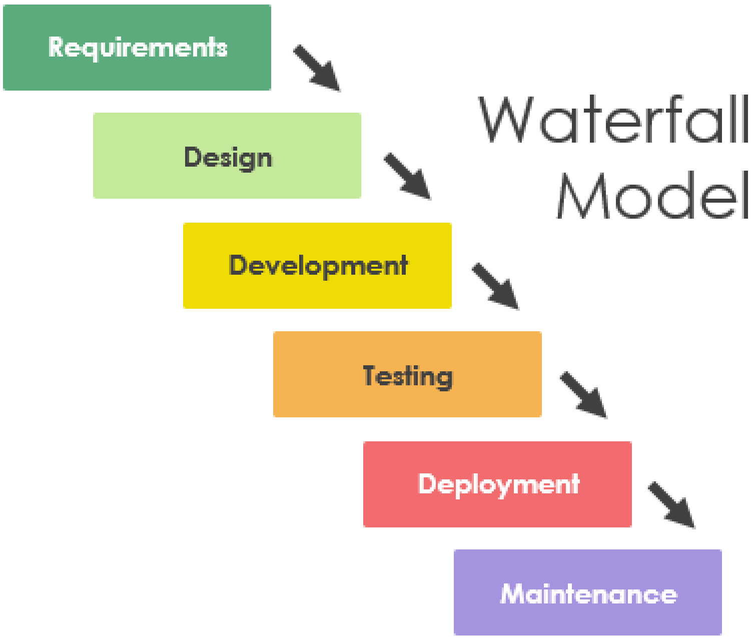
When you do use them effectively, you'll be much better organized, and you'll be much more reliable. You'll experience less stress, safe in the knowledge that you haven't forgotten anything important More than this, if you prioritize intelligently, you'll focus your time and energy on high-value activities, which will mean that you're more productive, and more valuable to your team.

Keeping a properly structured and thought-out list sounds simple enough. But it can be surprising how many people fail to use them at all, never mind use them effectively.

In fact, it's often when people start to use them effectively and sensibly that they make their first personal productivity breakthroughs, and start making a success of their careers. The article, below, gives some tips on how you can start to use to-do lists more effectively.

1.3 | Application process model

- Requirements
- Design
- Implementation
- Testing
- Deployment
- Maintenance



1.4 | “Why Chosen”

- This application is simple and easy to understand and use.
- Its process activities are clearly separated and organized.
- Waterfall model works well of small project where requirement are very well understand.
- All the requirements for this project are already know.
- Technology is understand.
- It will save time.
- It will help more reliable.
- Easy to Buy Ticket & Safe you’re Data.

1.5 | “AIM OF THE PROJECT”

In the existing system the number of user tickets, payments, and bus schedule and ticket confirmation are done only manually paper-pen based system but in proposed system we have to computerize all and all the processing information and transaction using the Bus ticket Management System. All sort of information manage by an administrator and that is the reason of our main module is Administrative module.

CHAPTER 02

REQUIREMENT ENGINEERING

2.1 | Requirement Overview

Requirement engineering encompasses the tasks that tasks that lead to an understanding of what the business impact of the software will be, what the customer wants, and how end users will interact with the software. Requirements engineering is defined in defined in terms of its major activities-

- ❖ Understanding problems
- ❖ Solution determination
- ❖ Specification of a solution that is testable, understandable, maintainable and that satisfies project quality guidelines

2.3 | "MODULES"

This module is the main module which performs all the main operation in the system.

The major operation in the system are:

- ➢ **Login Panel**
  - ✓ User Name
  - ✓ Password
  - ✓ Login
  - ✓ Close
- ➢ **Home Page**
  - ✓ **Dashboard**
    - o Available Buses
    - o Today's Income
    - o Total Income
    - o Income Chart
  - ✓ **Available Buses**
    - o Add Buses
    - o Update Buses
    - o Delete Buses
    - o Reset Selected Buses
    - o Show Table
    - o Search Buses
  - ✓ **Booking Ticket**
    - o Customer Information
    - o Select
    - o Reset
    - o Selected Customer Information Section

    - Payment Section
    - Receipt Section
- ✓ **Customers**
    - Search
    - Details Table
- ✓ **Log Out**
    - Login Panel

2.3 | “Functional requirement”

User:

1. Name
2. Password
3. Log-in
4. Dashboard
5. Available Buses
6. Booking Ticket
7. Customers details
8. Log-out

Security:

- Protected by Username & Password
- Use best security system to secure the data

Chapter 03

SYSTEM DESIGN

3.1 | “System Design overview”

System design for a bus management system project involves creating a high-level plan for how the various components of the system will interact with each other to achieve the desired functionality. Here are some key considerations when designing a bus management system:

1. Define the system requirements: Before designing the system, it is important to clearly define the functional and non-functional requirements of the system. This may include features such as route management, real-time tracking, fare collection, reporting, and analytics.

2. Identify the system components: Once the requirements have been defined, the next step is to identify the key system components that will be required to implement the functionality. This may include hardware components such as GPS trackers, onboard fare collection devices, and control systems, as well as software components such as a user interface, a database, and an analytics engine.

3. Design the system architecture: With the components identified, the next step is to design the system architecture. This involves defining how the different components will interact with each other to achieve the desired functionality. This may involve creating diagrams that show the flow of data between components, as well as specifying the interfaces between them.

4. Develop the software: Once the system architecture has been designed, the next step is to develop the software that will run on the various components. This may involve developing custom software for each component, as well as integrating off-the-shelf software components.

5. Test the system: After the software has been developed, it is important to test the system to ensure that it meets the specified requirements. This may involve creating test plans and scenarios, as well as conducting user acceptance testing.

6. Deploy the system: Once the system has been tested and validated, the final step is to deploy the system. This may involve installing hardware components on buses, configuring software components, and training users on how to use the system.

Overall, designing a bus management system requires a thorough understanding of the functional and non-functional requirements, as well as expertise in software and hardware engineering.

3.2 | “Logical Design”

Logical design is the process of translating the conceptual design of a system into a more detailed representation that can be used as the basis for physical implementation. In other words, it involves creating a logical model of the system that defines the various components and how they interact with each other to achieve the desired functionality.

During logical design, the focus is on defining the system's data structures, data flow, and data processing logic. This involves creating a detailed specification of the data that will be stored and processed by the system, as well as the algorithms and processing rules that will be used to manipulate that data.

Logical design is important because it provides a detailed blueprint of the system that can be used by developers to create the physical implementation of the system. It also serves as a reference for testing and validation, and can help ensure that the system meets the specified requirements.

Some key activities involved in logical design include:

1. Creating a data model: This involves defining the various data entities that will be stored and processed by the system, as well as the relationships between those entities.

2. Defining data flow: This involves specifying how data will flow through the system, from its source to its destination, and how it will be processed along the way.

3. Creating processing logic: This involves defining the algorithms and rules that will be used to manipulate the data within the system.

4. Creating system interface designs: This involves designing the user interfaces that will be used to interact with the system, including data entry screens, report formats, and data display formats.

Overall, logical design is a critical step in the system development process, as it lays the foundation for the physical implementation of the system.

3.3 | "Implementation of Model"

3.3.1 | "Database Design"

Database design is the process of creating a structured representation of the data that will be stored and managed by a database management system (DBMS). A well-designed database ensures efficient data storage, retrieval, and maintenance, while also supporting data security and integrity.

The database design process involves several steps:

1. **Requirements gathering:** This involves identifying the data that needs to be stored and managed, as well as the relationships between different data entities.

2. **Conceptual design:** This involves creating a high-level data model that captures the key entities and relationships of the system. This model is usually represented using an Entity-Relationship (ER) diagram.

3. **Logical design:** This involves creating a detailed specification of the data structures, data flow, and data processing logic of the system. This includes defining the data

types, relationships, and constraints that will be used by the DBMS to manage the data.

4. **Physical design:** This involves specifying the physical layout of the database, including the file structures, indexing schemes, and data access methods. It also involves determining the hardware and software requirements for the DBMS.

5. **Implementation:** This involves creating the database schema and populating the database with data.

6. **Testing and validation:** This involves testing the database to ensure that it meets the specified requirements, including performance, security, and data integrity.

7. **Maintenance:** This involves ongoing monitoring, tuning, and updating of the database to ensure that it continues to meet the changing needs of the system and its users.

Overall, the database design process is critical to the success of any database application. A well-designed database can improve data management, enhance system performance, and support effective decision-making.

3.3.2 | “Entity Relationship Model”

The Entity-Relationship (ER) model is a graphical representation used in database design to describe the data and relationships between entities within a system. The ER model is composed of three main components:

1. **Entities:** An entity is a person, object, concept, or event that can be identified and stored in a database. In the ER model, entities are represented by rectangles with their names at the top.

2. **Attributes:** Attributes are the characteristics or properties of an entity. In the ER model, attributes are represented by ovals connected to the entity rectangle.

3. **Relationships:** Relationships describe the associations between entities. In the ER model, relationships are represented by diamonds connected to the entity rectangles with lines.

Entities, attributes, and relationships are linked together to create an ER diagram that shows how the data is structured and how it relates to other data within the system. ER diagrams are used as a blueprint for designing and implementing a database system.

ER diagrams can be used to model various aspects of a system, including:

1. **One-to-one relationships**: This relationship type indicates that each instance of one entity is related to exactly one instance of another entity.

2. **One-to-many relationships**: This relationship type indicates that each instance of one entity can be related to many instances of another entity.

3. **Many-to-many relationships:** This relationship type indicates that many instances of one entity can be related to many instances of another entity.

4. **Cardinality and optionality:** Cardinality refers to the number of instances of one entity that can be related to another entity, while optionality refers to whether a relationship is mandatory or optional.

Overall, the ER model provides a simple, yet powerful way to visualize the structure and relationships of data within a system, making it a useful tool for database design and implementation

3.4 | “Identifying Entities”

3.4.1 | “According to our design identifying the entities”

- **Login Page**
- **Home Page**
  - i. Dashboard
  - ii. Available Buses
  - iii. Booking Ticket
  - iv. Customers
  - v. Log Out
  - vi. Close
  - vii. Minimize

3.4.2 | “Attribute of our System under every Entity”

- **Login Page**
  - Database
  - Dashboard.fxml
  - Login Page.fxml
  - Login Page.css
- **Home Page**
  - **a) Dashboard**
    - Available buses
    - Button
    - Chart
    - Form
    - Today Income
    - Total income
  - **b) Available Buses**
    - AvailableBusAdd
    - availableBusUpdate()
    - availableBusDelete()
    - availableBusReset()
    - comboBoxStatus()
    - availableBusBusData()
    - availableBShowBusData()

    - avaialbleBSelectBusData()
    - availableSearch()
    - busIdList()
    - LocationList()
    - typeList()
    - ticketNumList()

c) Booking Ticket

- bookingTicketSelect()
- bookingTicketReset()
- genderList()
- bookingTicketPay()

d) Customer

- Search
- Table Of Content
- ---
- ----

e) Log Out

- Logout()
- Alert
- LoginPage.fxml

f) Close

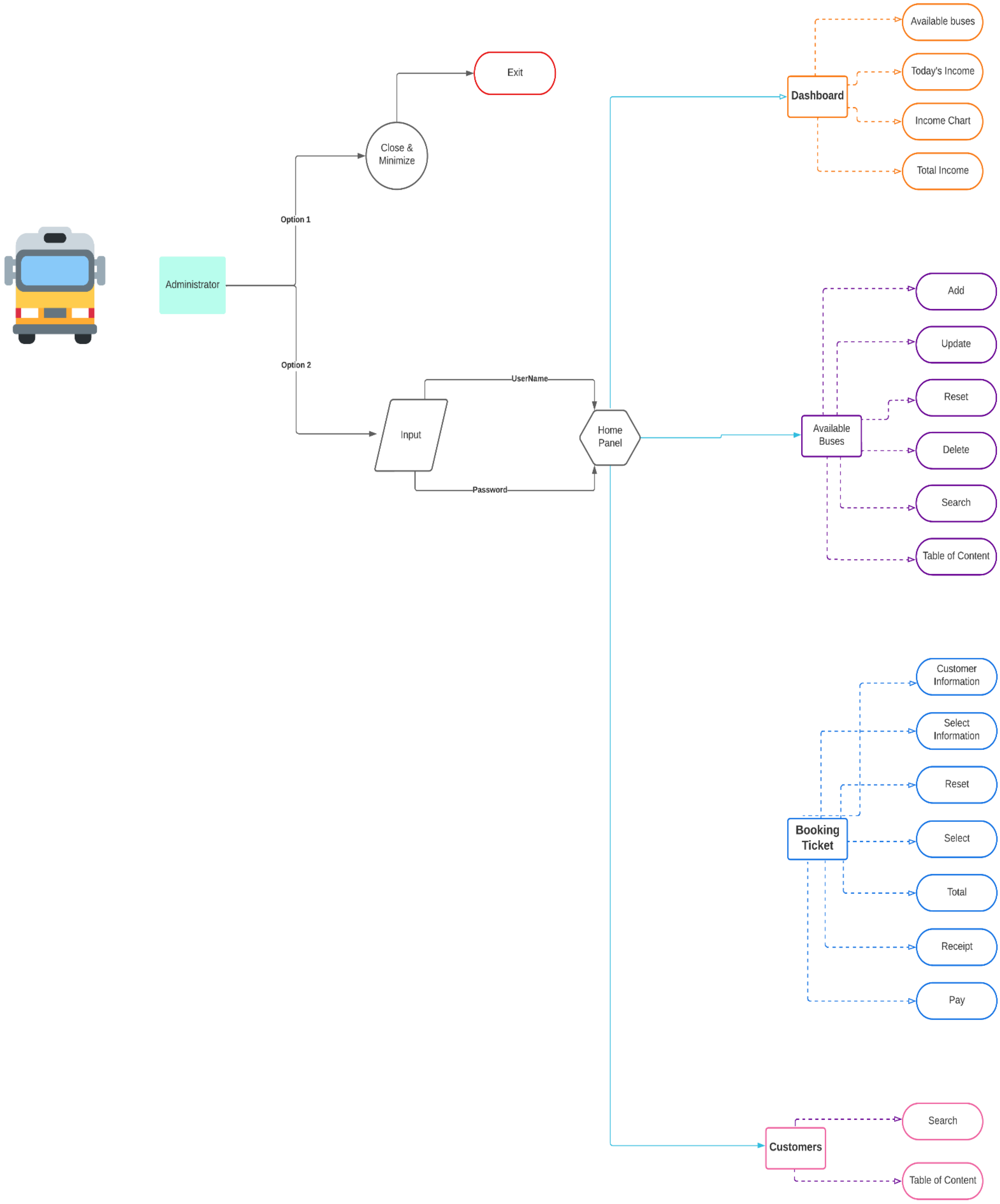
- Exit(0)
- Button

g) Minimize

- Stage
- Button
- Scene

ENTITY RELATIONSHIP DIAGRAM

3.5 | "Map Diagram"



3.6 | “System Requirements”

- **Hardware requirement**
  - Pc or Laptop

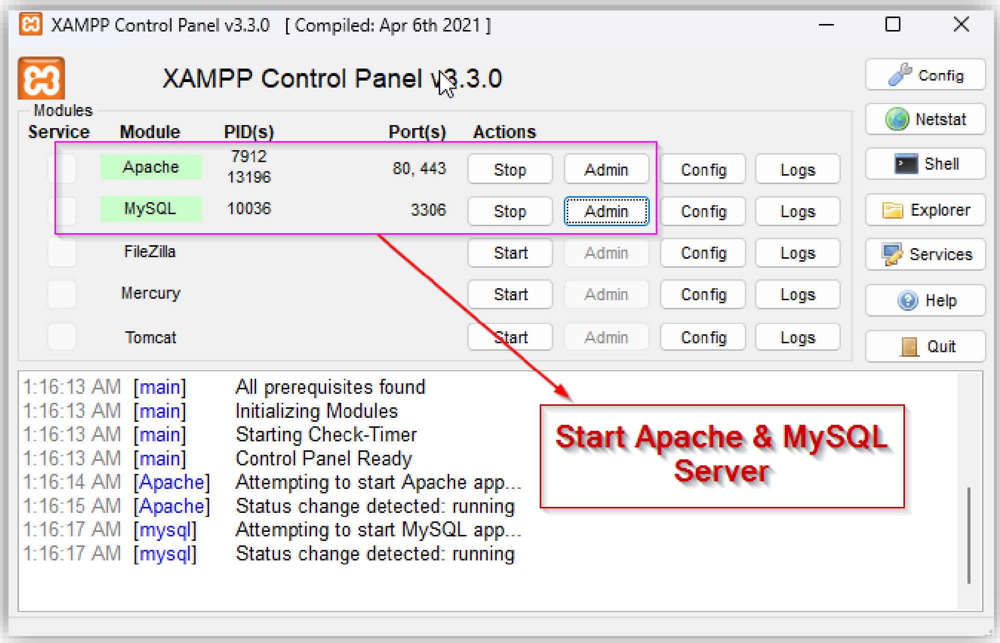
- **Software requirement**
  - Operating System Linux or Windows
  - Net Beans IDE
  - Java Development Kit
  - XAMMP
  - Scene Builder

3.7 | “Testing”

Testing is a set of activity that can be plumed in advanced and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing as it viral success of the system testing objectives, there are several rules that can serve as testing objects. Testing is a process of executing a program intend of finding errors. A good test case is one that has high possibility of finding an undiscovered error.

SCREEN SHOOTS

3.7.1 | “Test Case - Run the project”



XAMPP Control Panel v3.3.0 [Compiled: Apr 6th 2021]

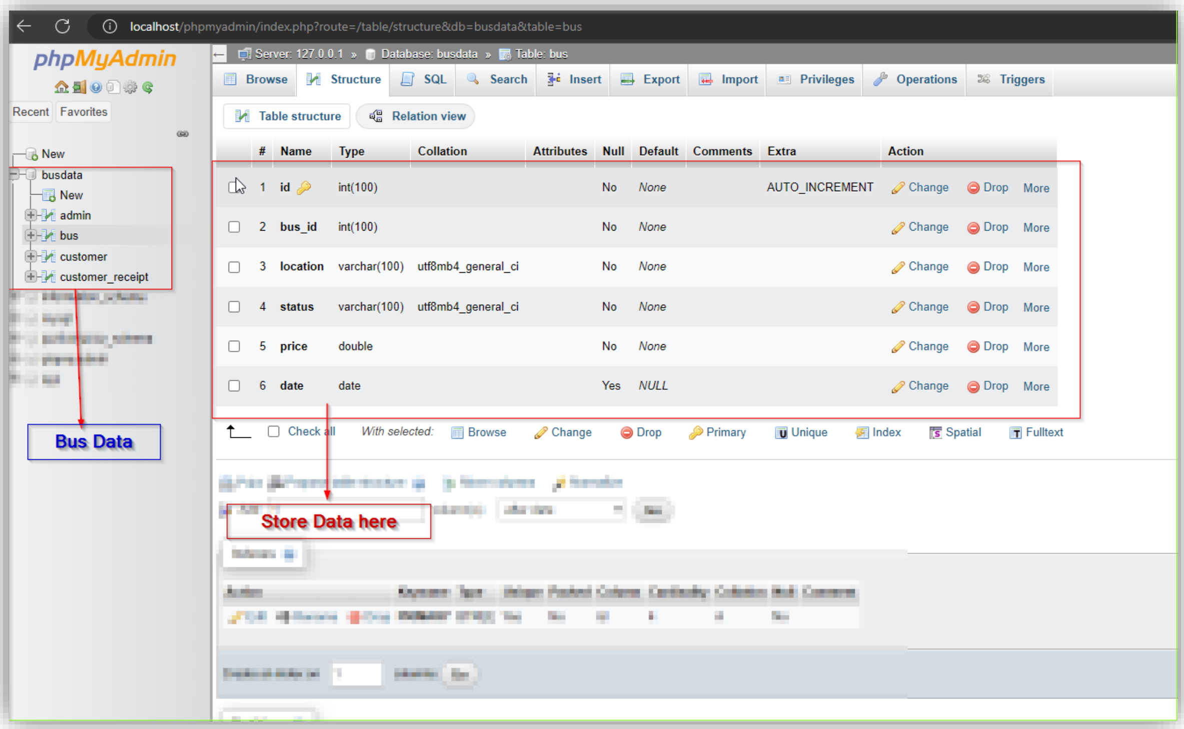
XAMPP Control Panel v3.3.0

Modules

Service	Module	PID(s)	Port(s)	Actions			
	Apache	7912 13196	80, 443	Stop	Admin	Config	Logs
	MySQL	10036	3306	Stop	Admin	Config	Logs
	FileZilla			Start	Admin	Config	Logs
	Mercury			Start	Admin	Config	Logs
	Tomcat			Start	Admin	Config	Logs

```
1:16:13 AM  [main]     All prerequisites found
1:16:13 AM  [main]     Initializing Modules
1:16:13 AM  [main]     Starting Check-Timer
1:16:13 AM  [main]     Control Panel Ready
1:16:14 AM  [Apache]   Attempting to start Apache app...
1:16:15 AM  [Apache]   Status change detected: running
1:16:17 AM  [mysql]    Attempting to start MySQL app...
1:16:17 AM  [mysql]    Status change detected: running
```

Start Apache & MySQL Server



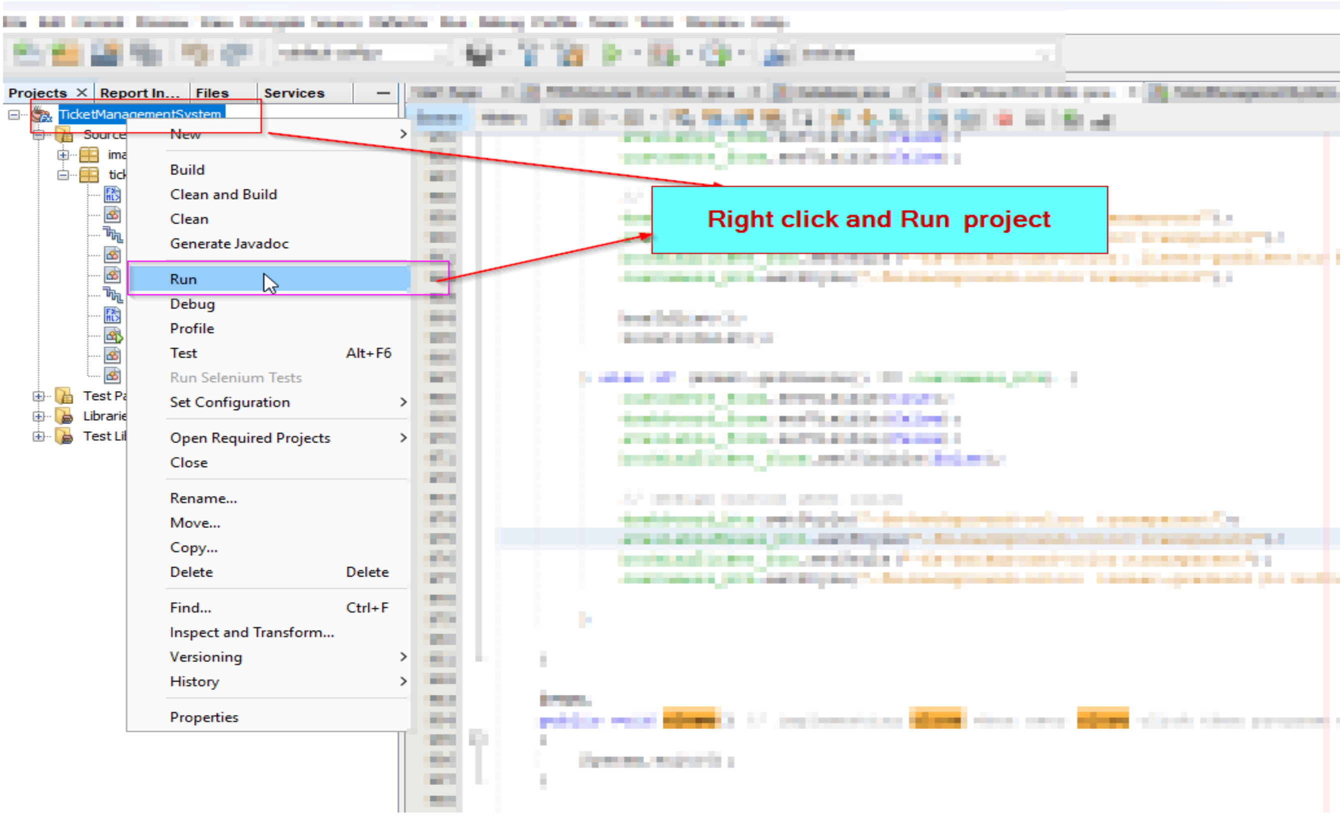
localhost/phpmyadmin/index.php?route=/table/structure&db=busdata&table=bus

Server: 127.0.0.1 » Database: busdata » Table: bus

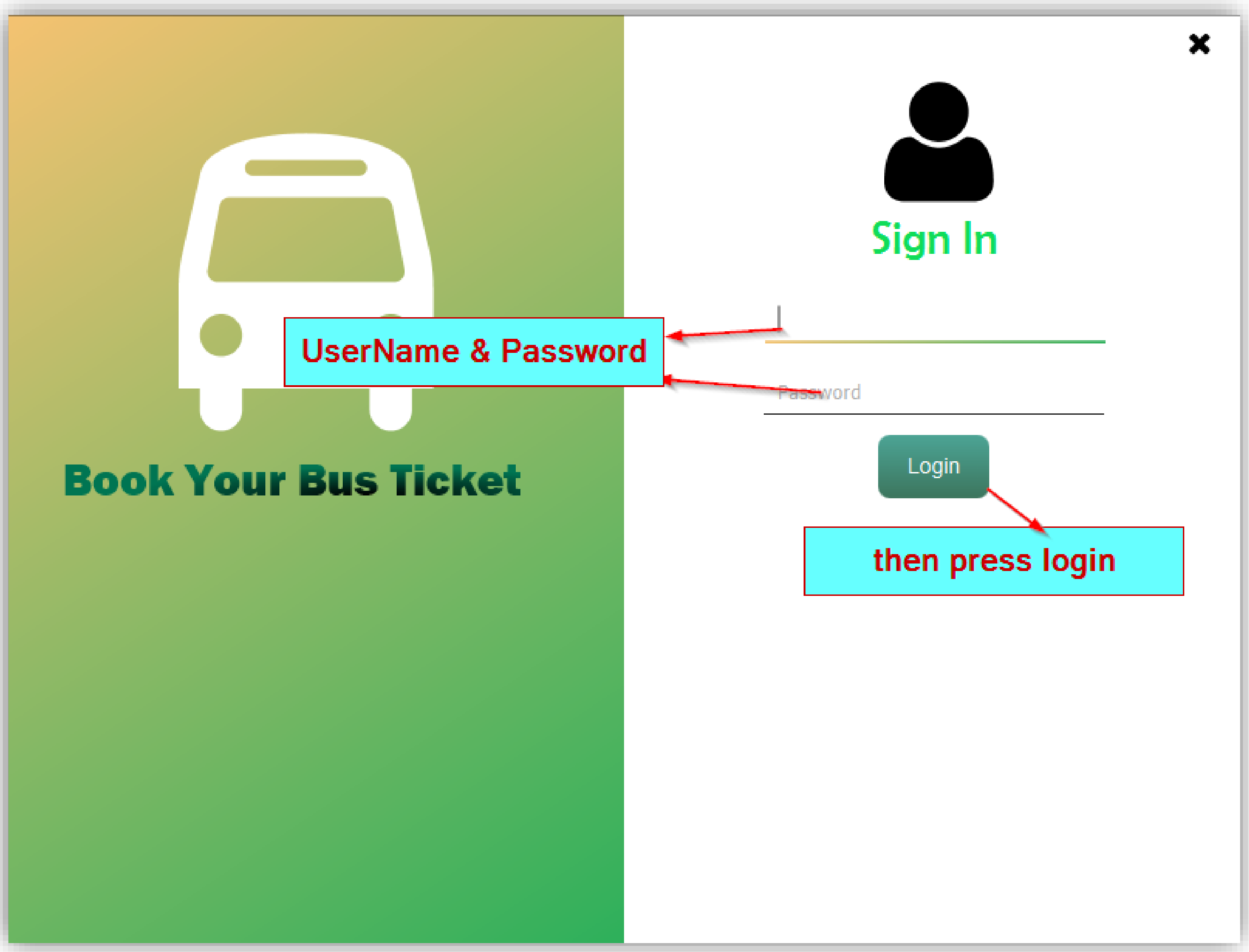
	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action		
□	1	id	int(100)			No	*None*		AUTO_INCREMENT	Change	Drop	More
□	2	bus_id	int(100)			No	*None*			Change	Drop	More
□	3	location	varchar(100)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	4	status	varchar(100)	utf8mb4_general_ci		No	*None*			Change	Drop	More
□	5	price	double			No	*None*			Change	Drop	More
□	6	date	date			Yes	*NULL*			Change	Drop	More

Bus Data

Store Data here

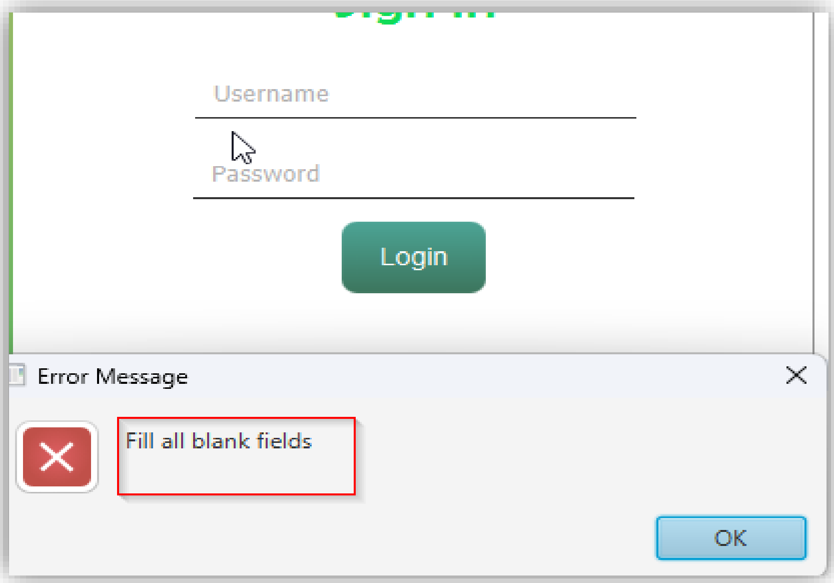


Right click and Run project



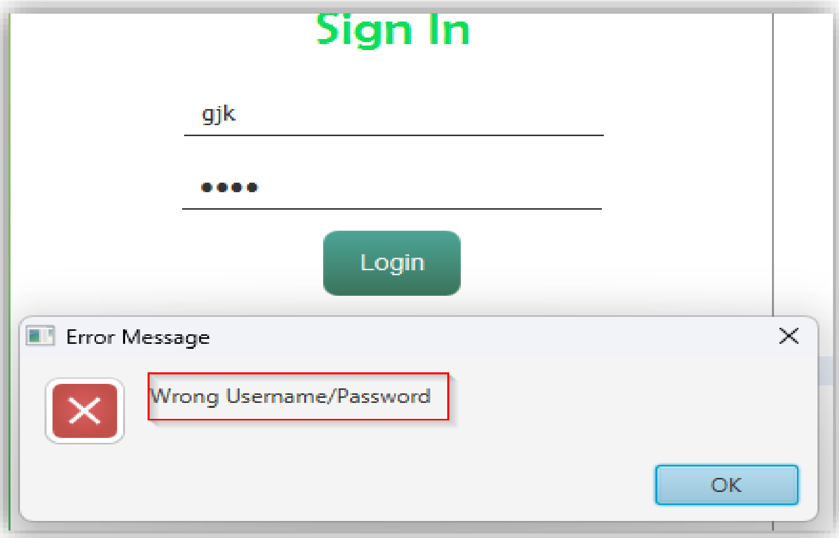
UserName & Password

then press login



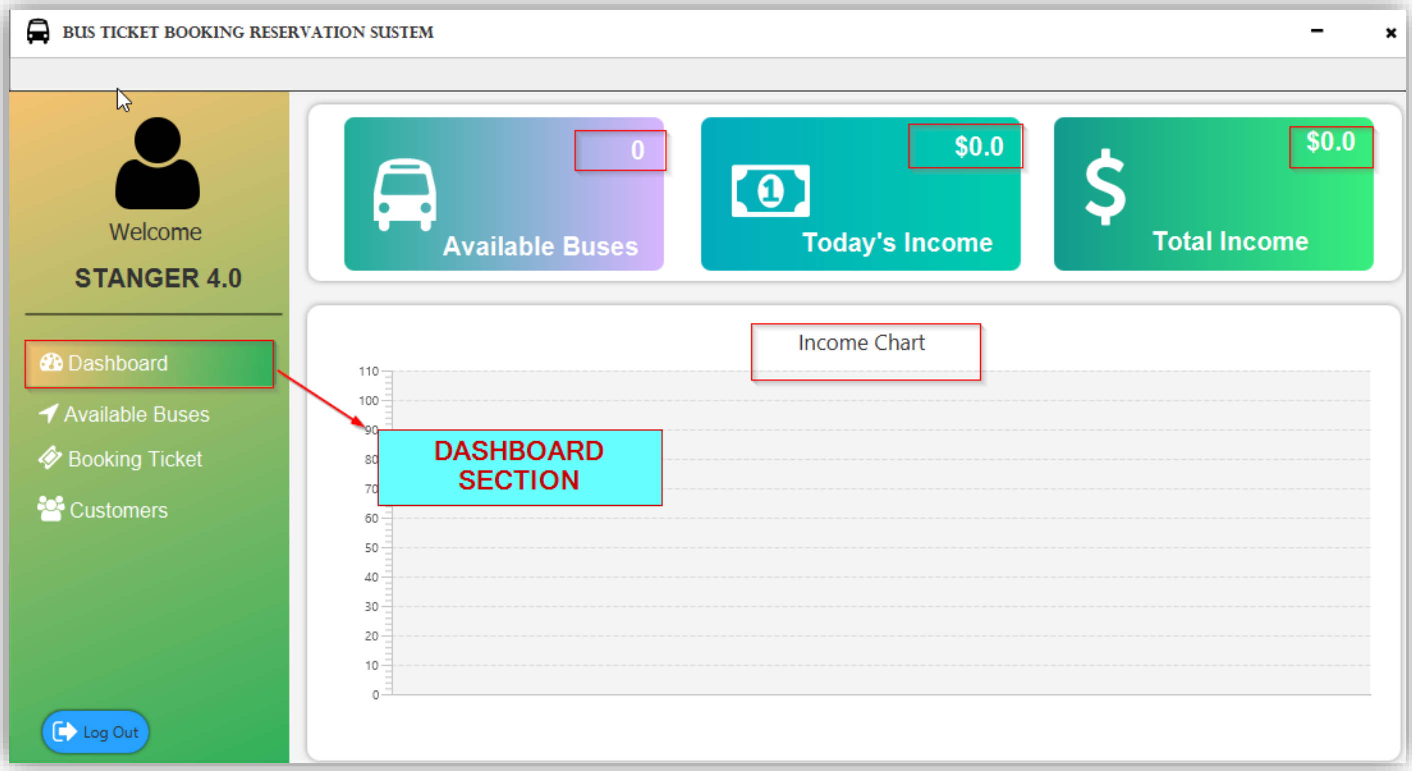
Error Message

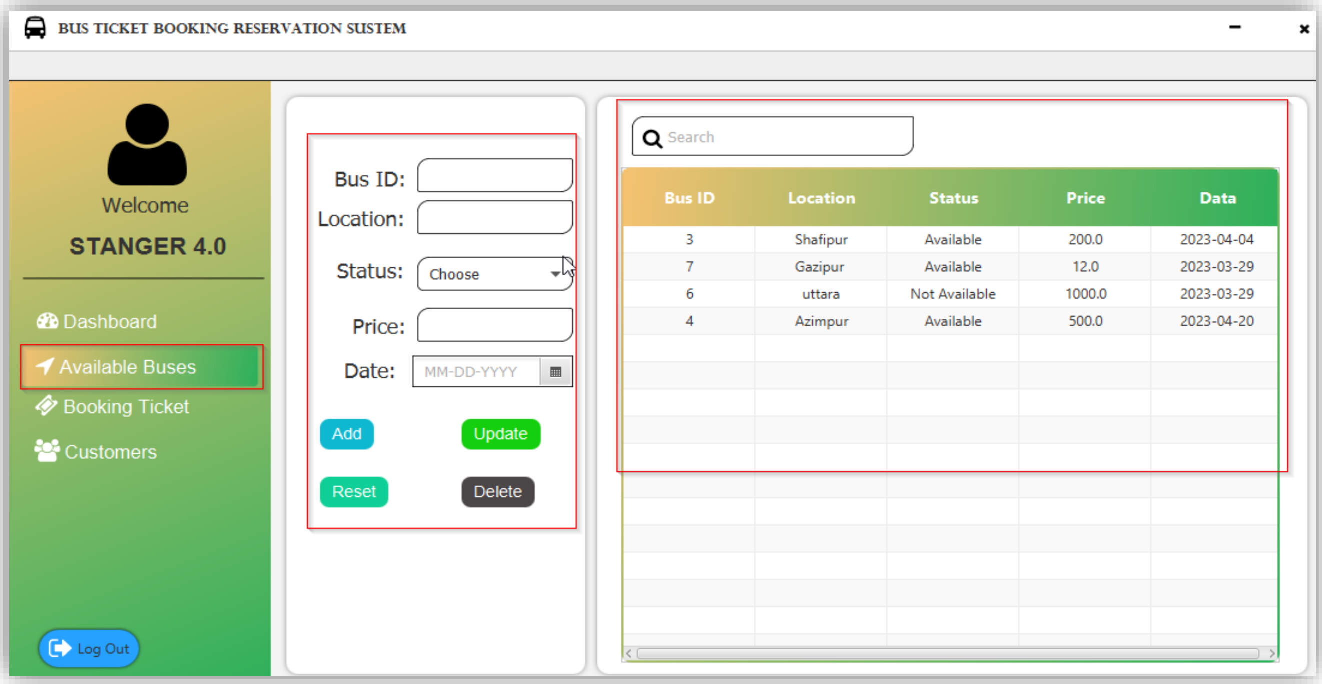
Fill all blank fields



Error Message

Wrong Username/Password





Bus ID:

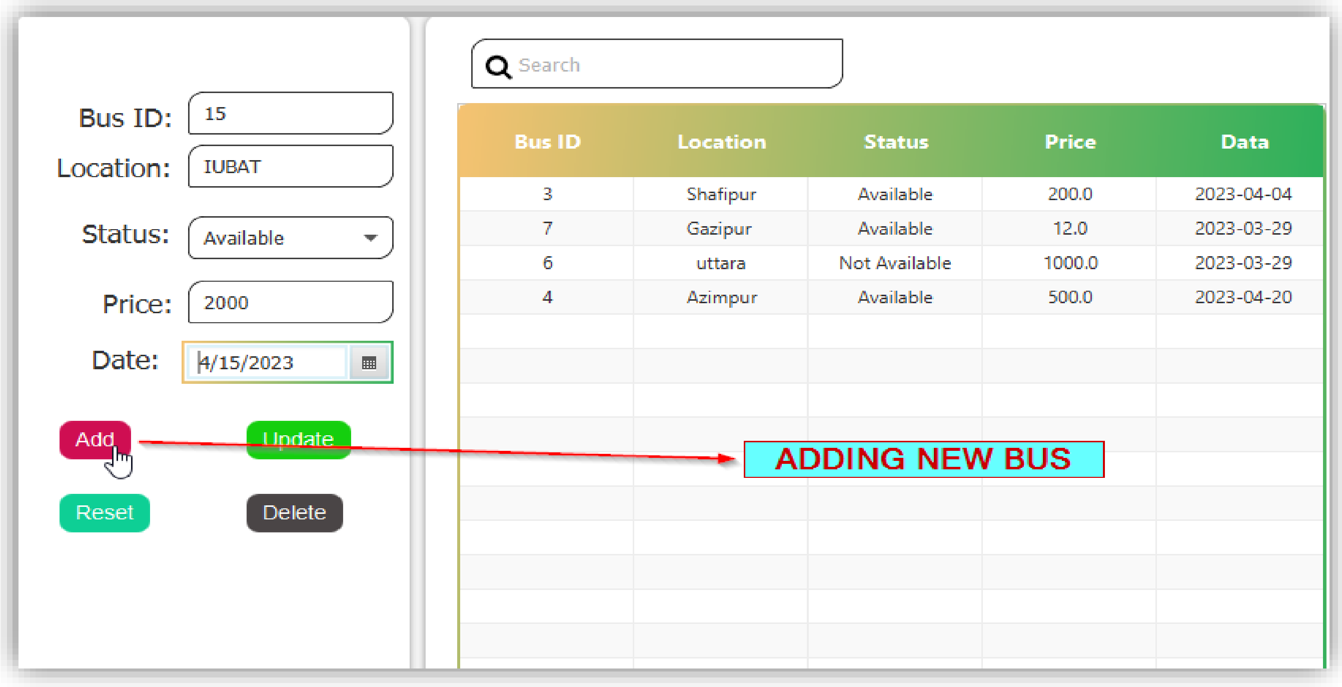
Location:

Status: Choose

Price:

Date: MM-DD-YYYY

Bus ID	Location	Status	Price	Data
3	Shafipur	Available	200.0	2023-04-04
7	Gazipur	Available	12.0	2023-03-29
6	uttara	Not Available	1000.0	2023-03-29
4	Azimpur	Available	500.0	2023-04-20



Bus ID: 15

Location: IUBAT

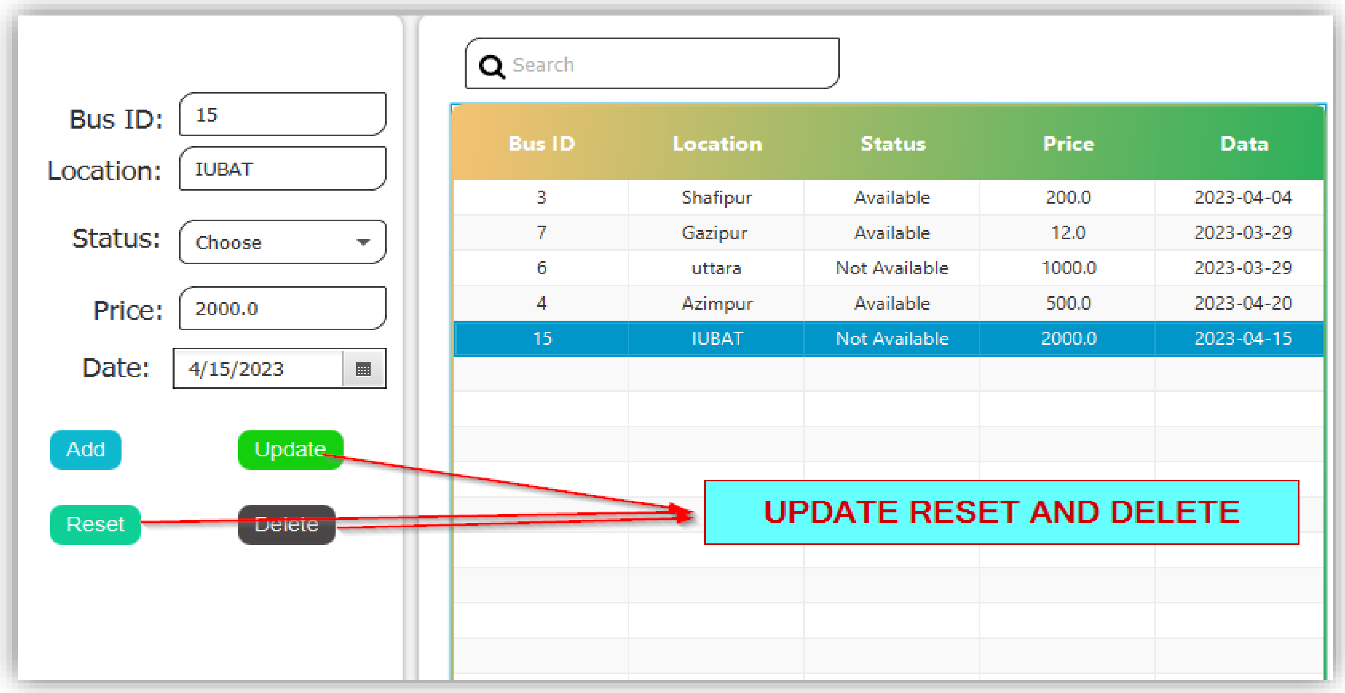
Status: Available

Price: 2000

Date: 4/15/2023

Bus ID	Location	Status	Price	Data
3	Shafipur	Available	200.0	2023-04-04
7	Gazipur	Available	12.0	2023-03-29
6	uttara	Not Available	1000.0	2023-03-29
4	Azimpur	Available	500.0	2023-04-20

ADDING NEW BUS



Bus ID: 15

Location: IUBAT

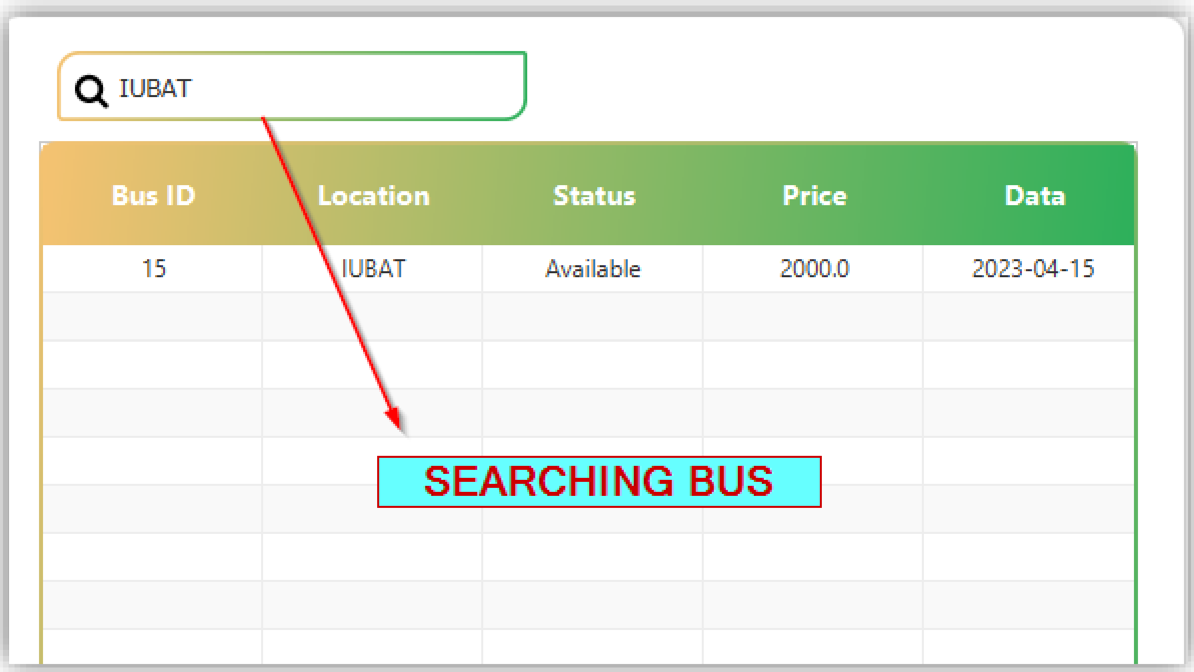
Status: Choose

Price: 2000.0

Date: 4/15/2023

Bus ID	Location	Status	Price	Data
3	Shafipur	Available	200.0	2023-04-04
7	Gazipur	Available	12.0	2023-03-29
6	uttara	Not Available	1000.0	2023-03-29
4	Azimpur	Available	500.0	2023-04-20
15	IUBAT	Not Available	2000.0	2023-04-15

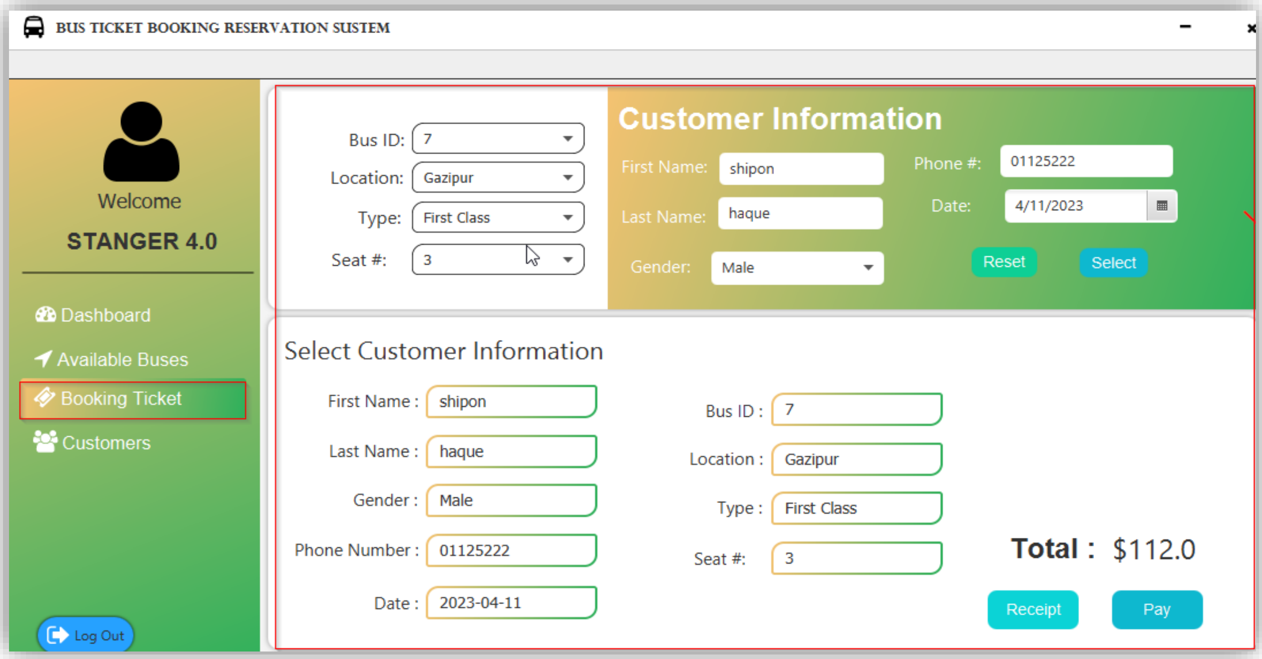
UPDATE RESET AND DELETE



IUBAT

Bus ID	Location	Status	Price	Data
15	IUBAT	Available	2000.0	2023-04-15

SEARCHING BUS



BUS TICKET BOOKING RESERVATION SUSTEM

Bus ID: 7

Location: Gazipur

Type: First Class

Seat #: 3

Customer Information

First Name: shipon

Last Name: haque

Gender: Male

Phone #: 01125222

Date: 4/11/2023

Select Customer Information

First Name : shipon

Last Name : haque

Gender : Male

Phone Number : 01125222

Date : 2023-04-11

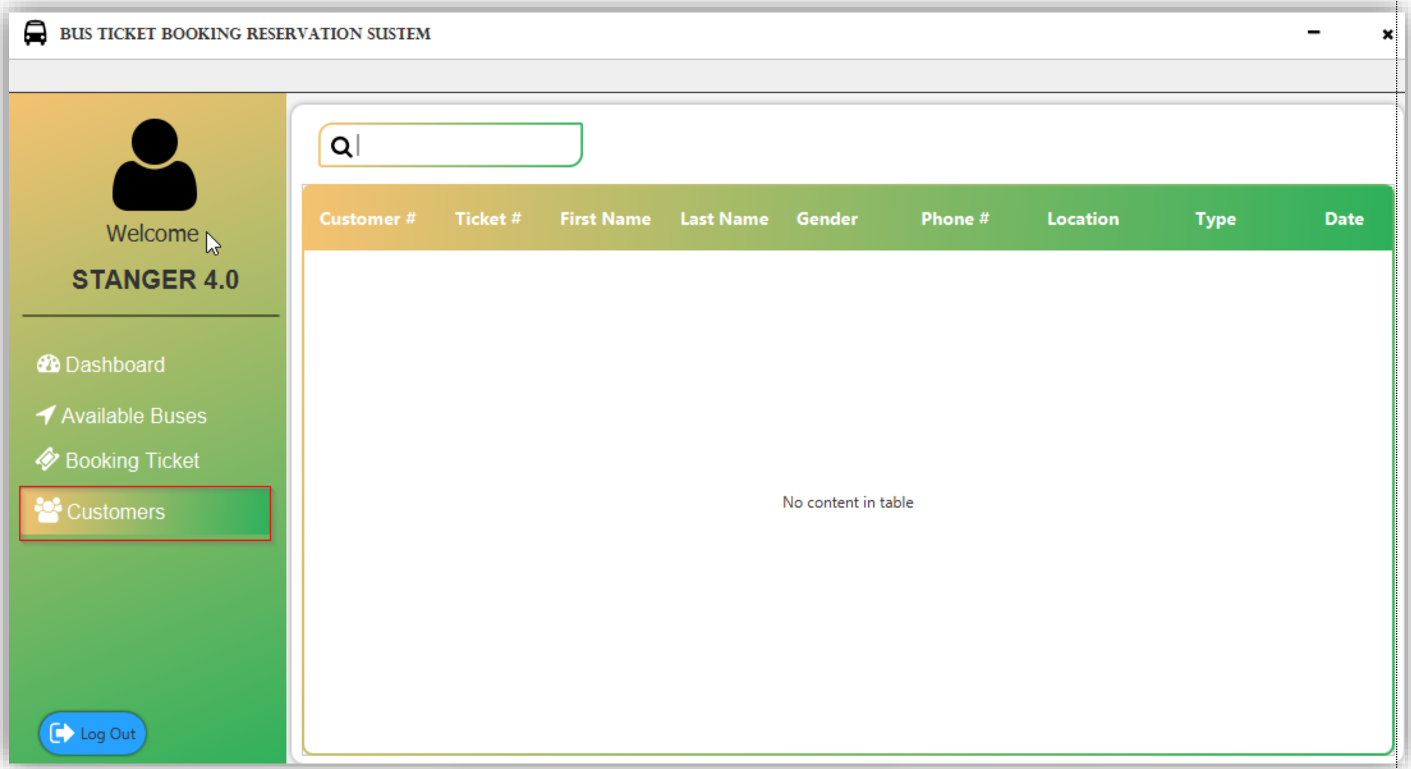
Bus ID : 7

Location : Gazipur

Type : First Class

Seat #: 3

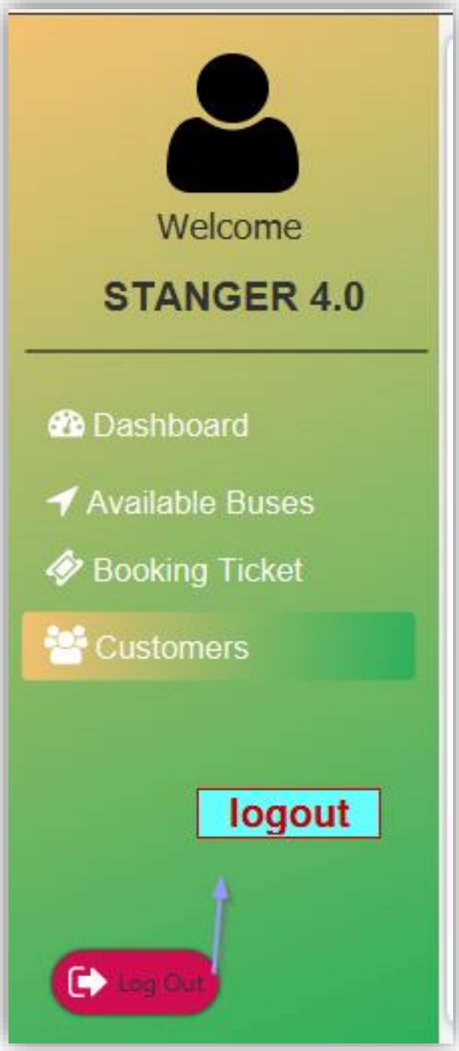
Total : $112.0



BUS TICKET BOOKING RESERVATION SUSTEM

Customer #	Ticket #	First Name	Last Name	Gender	Phone #	Location	Type	Date

No content in table



Chapter 04

ANALYSIS

4.1 “Analysis Modelling”

Analysis modelling is the process of creating models that help to understand, predict, and improve a system or process. It is an essential step in any system development life cycle, as it enables developers to identify and analyze requirements and design a system that meets those requirements.

The purpose of analysis modeling is to break down complex systems into smaller, more manageable components that can be analyzed, understood, and optimized. This can be done using a variety of modeling techniques, including data flow diagrams, use case diagrams, sequence diagrams, and activity diagrams.

Data flow diagrams (DFDs) are used to show the flow of data through a system. They illustrate the inputs, outputs, processes, and data stores of a system and can be used to identify areas where data is being duplicated or processed inefficiently.

Use case diagrams are used to model the interactions between users and a system. They show the different use cases or scenarios that a user might encounter when interacting with a system and can help to identify requirements for the system.

Sequence diagrams are used to model the interactions between objects in a system. They show the messages exchanged between objects and the order in which they occur, which can be useful for identifying potential bottlenecks or performance issues.

Activity diagrams are used to model the workflow of a process. They show the activities, decisions, and branching paths of a process and can be used to identify areas where the process can be streamlined or optimized.

In summary, analysis modeling is an essential step in system development that involves creating models to better understand and optimize complex systems. Various modeling techniques are used to break down and analyze different components of a system, including data flow diagrams, use case diagrams, sequence diagrams, and activity diagrams.

Chapter 05

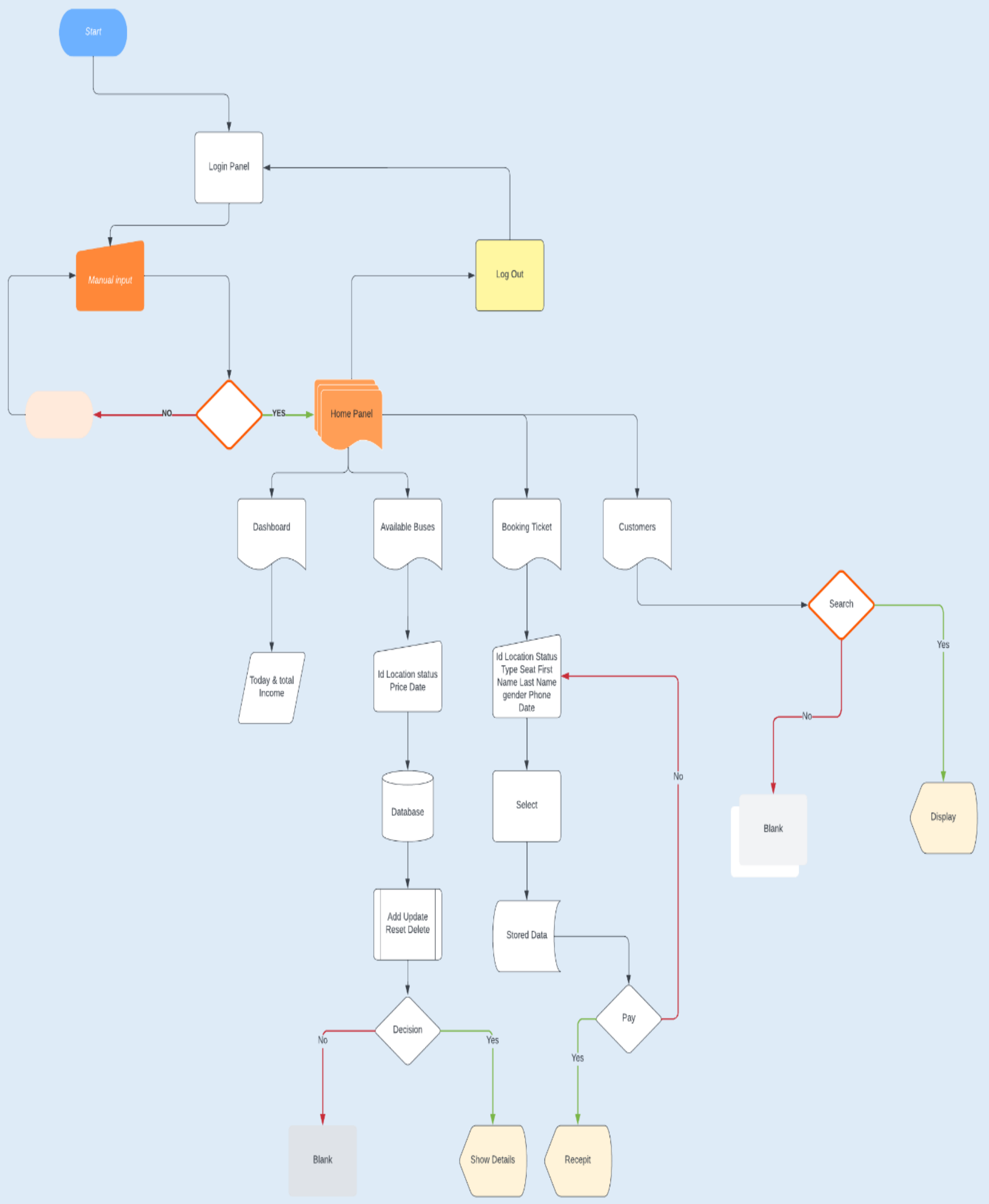
DESIGNING

FLOW CHART

5.1 “Flow Chart”

Project flowchart

shipon | April 14, 2023



Chapter 06

FUTURE PLANE

6.1 | “Add rest of Features”

In Bus Management System we can’t completed hole project because of short of time and limitation of knowledge. In Future we complete our project

6.2 | “Future Topic”

A future plane and bus management system would likely involve advanced technology and data-driven approaches to optimize transportation operations and improve the passenger experience. Here are some key features that such a system could include:

- Real-time monitoring: The system could use sensors and other monitoring devices to track the status of planes and buses in real-time. This data could be used to make real-time decisions about scheduling, routing, and maintenance.

- Predictive analytics: Advanced data analytics techniques could be used to predict flight and bus delays, identify potential maintenance issues before they occur, and optimize schedules to improve efficiency.

- Mobile apps: Passengers could use mobile apps to book and manage their flights or bus tickets, check in, receive updates on their travel status, and access a range of other services.

- Self-service kiosks: To reduce wait times and improve the passenger experience, self-service kiosks could be installed at airports and bus terminals, allowing passengers to check in, print boarding passes, and handle other tasks on their own.

- Automated baggage handling: Advanced robotics and conveyor systems could be used to automate the handling of baggage, reducing the risk of lost or delayed luggage and improving overall efficiency.

- Biometric authentication: To improve security and speed up the boarding process, biometric authentication technologies could be used to verify passengers' identities and streamline the boarding process.

- In-flight/buses entertainment: High-speed internet, personal entertainment systems, and other amenities could be provided on planes and buses to improve the passenger experience.

6.3 | “Consolation future plane”

Overall, a future plane and bus management system would likely involve a range of advanced technologies and data-driven approaches to improve efficiency, reduce delays, and enhance the passenger experience.

Chapter 07

CONCLUSION

7.1 | “Conclusion”

In conclusion, creating a to-do list application using Java and JavaFX for a university project can be a valuable learning experience for software development students. Although the project does not involve the use of a database, students can still gain important skills in object-oriented programming, user interface design, and file I/O operations.

To create a functional Bus management System application, students can use data structures such as arrays or Array Lists to store task information within their Java code. They can also use file I/O operations to read and write task information to a file on the user's computer.

When designing the user interface for the application, students should focus on creating a clean and intuitive layout that is easy to navigate. JavaFX provides a variety of tools for creating visually appealing interfaces with interactive elements such as buttons and checkboxes.

Overall, creating a to-do list application using Java and JavaFX can be a challenging but rewarding project for university students. It allows them to develop practical software development skills and create a useful tool that can improve their productivity and time management.

7.2 | “Limitation of the Project”

One limitation of a Bus Management System application created using only Java and JavaFX is that it may not be as robust or scalable as an application that uses a database.

Additionally, the application may not be able to handle multiple users or synchronize task information across different devices, since it is not connected to a central database. This can limit the usefulness of the application for teams or groups who need to collaborate on tasks.

Another limitation is that the application may not have the same level of security as an application that uses a database. Since task information is stored locally on the user's computer, it may be vulnerable to data loss or theft if the computer is compromised.

Lastly, the user interface of the application may be limited by the capabilities of JavaFX. While JavaFX provides a variety of tools for creating visually appealing interfaces, it may not have the same level of customization or flexibility as other user interface frameworks or libraries.

Chapter 08

REFERENCE

8.1 | “Online Resource”

We used varies online resource that’s help to make our project and also make easy to write our report.

To make this project we farce many difficulties’ link designing sector and functionality and so on. We solve this problem throwing internet and books resource we also used AI that’s helps a lot.

- YouTube: MarcoMan (https://www.youtube.com/@marcomanchannel)
- Git Hub
- Open AI :(https://chat.openai.com/)
- Lucid : (https://lucid.app)
- Java Point : (https://www.javatpoint.com/java-tutorial)
- W3School :
- JavaFx: online resource
- Scene Builder Resource

8.2 | “Offline Resource”

- Group member’s help a lot
- Take some idea from books
- Our Instructor **Rubayea Ferdows** mam also help to build our project
- Some Friend give some idea about project

THE END