

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
**JNANA SANGAMA, Belagavi - 590 018.**



**2020 - 2021**

**A**  
**mini project report on**  
**“TAXI MANAGEMENT SYSTEM”**

Submitted in partial fulfillment of the requirements for the award of the degree of  
**BACHELOR OF ENGINEERING**  
in  
**COMPUTER SCIENCE & ENGINEERING**

**Submitted by**

**UDDESHYA JHA (1AT18CS114 )**

**NITESH KUMAR (1AT18CS067)**

**Under the guidance of.**  
**Mr. RAJENDRA & Mrs. PADMINI**  
Assistant Professor  
Dept. of CSE, ATRIA I. T.



**ATRIA INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**  
Anandnagar, Bengaluru-560 024

**ATRIA INSTITUTE OF TECHNOLOGY**  
(Affiliated to Visvesvaraya Technological University)  
ASKB Campus, Anandnagar,  
Bengaluru – 560024

**Department of Computer Science and Engineering**



**CERTIFICATE**

Certified that the project work entitled **“Taxi Management System”** **UDDESHYA JHA (1AT18CS114), NITESH KUMAR (1AT18CS067)** are bonafide students of **ATRIA INSTITUTE OF TECHNOLOGY**, Bengaluru, in partial fulfillment for the award of Degree of **Bachelor of Engineering in Computer Science & Engineering** of **Visvesvaraya Technological University, Belagavi**, during the academic year **2020-21**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

**Guide**

**Mr. RAJENDRA  
&  
Mrs. PADMINI**  
(Asst. Prof., CSE)

**HOD**

**Dr. Aishwarya P**  
  
(Dept. of CSE)

**Principal**

**Mr. T N Sreenivasa**  
  
(Atria I T)

**External Viva**

**Name of Examiners**

**Signature with date**

1.

2.

# DECLARATION

We, **UDDESHYA JHA (1AT18CS114)**, **NITESH KUMAR (1AT18CS067)**, students of **5th semester Bachelor of Engineering, Department of Computer Science and Engineering, Atria Institute of Technology, Bengaluru**, would hereby declare that the project entitled “**Taxi Management System**” has been carried out by us at **Atria Institute of Technology, Bengaluru**, and submitted in partial fulfillment of the course requirement for the award of degree of **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi**, during the academic year **2020-21**.

We further declare that the work embodied in this report has not been submitted to any other university or institution for the award of any other degree.

Place: Bengaluru

Signature of the students

Date:

**UDDESHYA JHA**  
**(1AT18CS114)**

**NITESH KUMAR**  
**(1AT18CS067)**

## **ABSTRACT**

When it comes to cab rental services, Taxi Service is the most trusted and reliable name in the travel business. The most advanced travel agents offering cab rental and car hire in India, making full use of information technology to improve the level of our efficiency. However, this is only one aspect of services. And this project continually strive to offer the best of services - both in terms of man and machine, to our clients

Moreover, this project has a fleet of cars ranging from luxury to budget cabs. While, it offers online cab hire service for corporate houses. And this project claim to offer the best of rates, which are tailor-made depending upon the facilities, availed and offer both intercity and intra-city cab facilities. All cabs have proper permits and documentation so that the clients couldn't be hassled for the lack of documents. However, this project has strategic backup system for any eventuality. Cab drivers are educated, polite, and reliable and are trained to handle acute breakdowns. The cab service includes all categories of cars from luxury to budget.

Further, this project's utmost priority is quality. To achieve this, vehicles are well maintained and tested for delivering optimum and uninterrupted performance. Team of professionals in the travel business enables this system to design trips that suits to all budgets and preferences of the travelers. In addition, workforce including drivers and administrative staff are well trained to discharge their duties with a lot of efficiency.

# ACKNOWLEDGEMENT

We are grateful to our institution, **Atria Institute of Technology**, for having provided us with the facilities to successfully complete this mini project on TAXI MANAGEMENT SYSTEM.

We thank Mr. **T N Sreenivasa, Principal** and Mrs. **Dr. Aishwarya P, HOD, CSE** for providing us all the necessary facilities for the successful completion of our mini-project.

Deadlines play a very important role in the successful completion of the academic project on time, efficiently and effectively. We take this opportunity to express our deep sense of gratitude to our guide and coordinator **Mr. RAJENDRA & Mrs. PADMINI, Assistant Professor, Department of CSE** for their valuable guidance and help throughout the course of the academic mini-project. They have always been patient with us and helped immensely in completing the task on hand. We also thank them for their immense support, guidance, specifications & ideas without which seminar would have been completed without full merit.

Last but not least from the Department of Computer Science and Engineering, teaching and non-teaching staffs for their constant encouragement, support, patience, and endurance shown during the preparation of this report were remarkable. We also thank the management.

Finally, We thank our parents and friends for their motivation, morale and material support.

**UDDESHYA JHA**  
**(1AT18CS114)**

**NITESH KUMAR**  
**(1AT18CS067)**

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>DECLARATION</b>	<b>i</b>
	<b>ABSTRACT</b>	<b>ii</b>
	<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>1.</b>	<b>INTRODUCTION TO DATABASE</b>	<b>1</b>
1.1	Database Environment System	4
1.2	Advantages of Using DBMS Approach	5
1.3	Architecture of Database	6
<b>2.</b>	<b>INTRODUCTION TO PROJECT</b>	<b>8</b>
2.1	Brief Description	8
2.2	Table Description	9
<b>3.</b>	<b>DESIGN</b>	<b>14</b>
3.1	Entity Relationship Diagram	14
3.2	Normalization	15
3.3	Schema Diagram	16
<b>4.</b>	<b>SYSTEM ANALYSIS</b>	<b>17</b>
4.1	Various divisions in the Project	17
4.2	User Characteristics	18
4.3	Feasibility Study	18
<b>5.</b>	<b>SYSTEM REQUIREMENTS</b>	<b>20</b>
5.1	Requirements	20
5.1.1	Software Requirements	20
5.1.2	Software Features	21
5.1.3	Hardware Requirements	21
5.1.4	Security Requirements	22
<b>6.</b>	<b>TESTING</b>	<b>23</b>
6.1	Levels of Testing	23
6.1.1	Unit Testing	23
6.1.2	System Testing	23
6.2	Test Cases	23
	<b>CODE SNIPPETS</b>	<b>25</b>
A.1	Login Authentication Query	25

A.2 Insert Query	26
A.3 Update Query	26
A.3.1 Location and Destination Updating	26
<b>SCREENSHOTS</b>	<b>28</b>
B.1 Home Page	28
B.2 Login Page	29
B.3 Customer Details page	29
B.4 Driver Details Page	30
B.5 Destination Page	30
B.6 Car Details page	31
B.7 Customer Taxi booking Page	31
<b>CONCLUSION AND FUTURE SCOPE</b>	<b>37</b>
<b>REFERENCES</b>	<b>38</b>

## LIST OF FIGURES

<b>FIG NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
1.1	Simplified database environment system	4
1.2	Architecture of DBMS	7
3.1	ER Diagram	14
3.3	Schema Diagram in 3 <sup>rd</sup> normal form	16
A.1	Login Authentication Query	25
A.2	Insert Query	26
A.3	Update Query	26
A.3.1	Location and Delivery Updating	26
A.3.2	Profile Updation	27

B.1 Home Page	28
B.2 Login Page	29
B.3 Customer details Page	30
B.4 Driver details Page	30
B.5 Destination Page	31
B.6 Car details Page	31
B.7 Customer Taxi booking Page	32

## LIST OF TABLES

Table Number	Table Name	Chapter No	Page No
2.2.1	Login Page	2	9
2.2.2	Car_information	2	9
2.2.3	Customer_information	2	10
2.2.4	Destination_Price_Table	2	10
2.2.5	Driver_Information	2	11



## CHAPTER – 1

# INTRODUCTION TO DATABASE

Database and database technology has a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, electronic commerce, engineering, medicine, genetics, law, education, and library science. The word database is so commonly used that we must begin by defining what the database is.

Our initial definition is quite general. A database is a collection of related data. By data, we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book or you may have stored it on a hard drive, using personal computers and software such as Microsoft excel. This collection of related data with an implicit meaning is a database.

The preceding definition of a database is quite general, for example, we may consider the collection of words that make up this page of text to be related data and hence to constitute a database. However, the common use of the term database is usually more restricted. A database has the following properties:

- A database represents some aspect of the real world, sometimes called the mini world or the universe of discourse. The changes to the mini world are reflected in the database.
- A database is a logically coherent collection of data with some inherent meaning. A random assortment of data cannot correctly be referred to as a database.
- A database is designed, built and populated with data for a specific purpose. It has an intended group of users and some preconceived applications in which these users are interested.

In other words, a database has some source from which data is derived, some degree of interaction with events in the real world, and an audience that is actively interested in its contents. The end-users of the database may perform business transactions (for example a customer buys a camera) or events may happen that may cause the information in the database to change. In order for a database to be accurate and reliable at all times, it must be a true reflection of the mini world that it represents; therefore changes must be reflected in the database as soon as possible.

A database can be of any size and complexity. A database may be generated and maintained manually or computerized. For example, a library card catalog is a database that may be created and maintained manually. A computerized database may be created and maintained either by a group of application programs written specifically for that task or by a database management system.

A database is a collection of data, typically describing the activities of one or more related organizations. For example, a university database might contain information about the following:

- Entities such as students, faculty, courses, and classrooms.
- Relationships between entities, such as student's enrolment in courses, faculty teaching courses, and the use of rooms for courses.

A database management system, or DBMS, is software designed to assist in maintaining and utilizing a large collection of data. The need for such systems as well as their use is growing rapidly. The alternative to using a DBMS is to store the data in files and write application-specific code to manage it.

### **File system versus DBMS**

To understand the need for a DBMS, let us consider a motivating scenario: a company has a large collection (say 500 GB) of data on employees, departments, products, sales, and so on. This data is accessed concurrently by several employees. Questions about the data must be answered quickly, changes made to the data by different users must be applied consistently and access to certain parts of the data must be restricted. We can try to manage the data by storing it in operating system files. This approach has many drawbacks, including the following

- We probably do not have 500GB of main memory to hold all the data. We must, therefore, store data in a storage device such as a disk or tape and bring relevant parts into the main memory for processing as needed.
- Even if we have 500 GB of main memory, on computer systems with 32 bit addressing, we cannot refer directly to more than about 4 GB of data. We have to program some method of identifying all data items.

- We have to write special programs to answer each question a user may want to ask about the data. These programs are likely to be complex because of the large volume of data to be searched.
- We must protect the data from the inconsistent changes made by different users accessing the data concurrently. If applications must address the details of such concurrent access, this adds greatly to their complexity.
- We must ensure that the data is restored to a consistent state if the system crashes while changes are being made.
- Operating systems provide only a password mechanism for security. This is not sufficiently flexible to enforce security policies in which different users have permission to access different subsets of the data.

A DBMS is a piece of software designed to make the preceding tasks easier. By storing data in DBMS rather than as a collection of operating system files, we can use the DBMS's features to manage the data in a robust and efficient manner. As the volume of data and the number of users grow hundreds of gigabytes of data and thousands of users are common in current corporate database DBMS support becomes indispensable.

## 1.1 DATABASE ENVIRONMENT SYSTEM

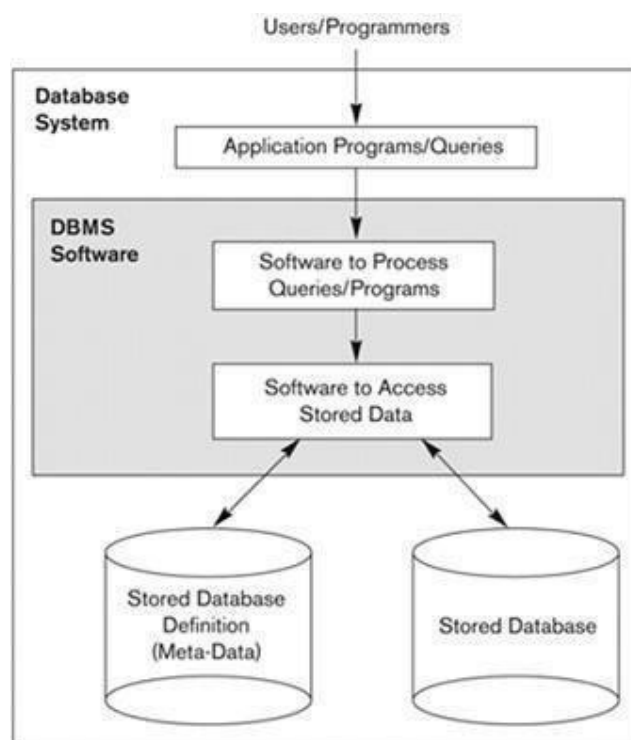


Fig 1.1: Simplified database environment system

A database management system (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. Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called Metadata. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to DBMS. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by DBMS include protecting the database and maintaining it over a long period of time, protection includes system protection against hardware or software malfunction and security protection against unauthorized or malicious access. A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use general-purpose DBMS software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own special purpose DBMS software. In either case, whether we use a general-purpose DBMS or not we usually have deployed a considerable amount of complex software. In fact, most DBMSs are very complex software systems. Fig 1.1 shows a simplified database environment system.

## 1.2 ADVANTAGES OF USING DBMS APPROACH

Using a DBMS to manage data has many advantages:

- **Data Independence:** application program should not, ideally, be expected to details of data representation and storage, the DBMS provides an abstract view of the data that hides such details.

- 
- **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
  - **Data Integrity and Security:** if data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.
  - **Data Administration:** when several users share data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.
  - **Concurrent Access and Crash Recovery:** A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.
  - **Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to data, facilitates quick application development. DBMS applications are also likely to be more robust than a similar stand-alone application because many important tasks are handled by the DBMS.

## 1.3 ARCHITECTURE OF DATABASE

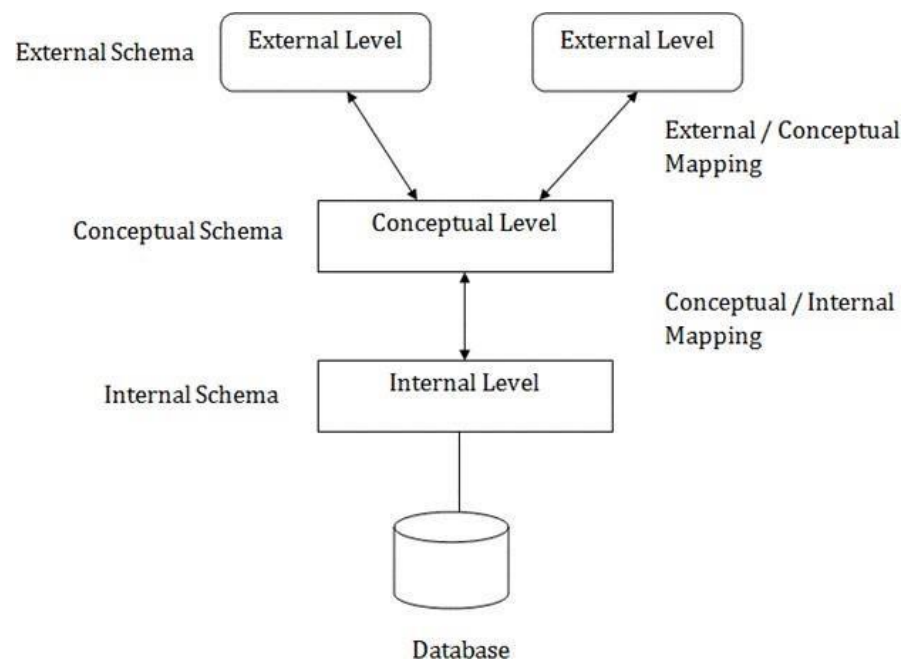
### The Three-Schema Architecture

The goal of the three-schema architecture illustrated in the figure is to separate the user application from the physical database. In this architecture, schemas can be defined at the following three levels:

- The internal level has an internal schema, which describes the physical storage structure of the database. The internal schema uses a physical data model and describes the complete details of data storage and access paths for the database.

- The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. Usually, a representational data model is used to describe the conceptual schema when a database system is implemented. This implementation conceptual schema is often based on a conceptual schema design in a high-level data model.
- The external or view level includes a number of external schemas or user views. Each external schema describes the part of a database that a particular user group is interested in and hides the rest of the database from that user group. As in the previous level, each external schema is typically implemented using a representational data model, possibly based on external schema design in a high-level data model.

Fig 1.2 shows the architecture of DBMS.



**Fig 1.2: Architecture of DBMS**

## **CHAPTER 2**

# **INTRODUCTION TO PROJECT**

## **TAXI MANAGEMENT SYSTEM**

### **2.1 BRIEF DESCRIPTION**

This project is aimed at developing a “Taxi Management System”. It is a system meant to maintain all the details regarding employees, transaction details which include Taxi details, customer details, goods details, retrieval of all data’s from database just by giving the delivery\_id.

#### **Objectives**

The main objective of the project on Taxi Management System is for the cargo management people that enables them to receive the goods from a source and send them to a required destination and for customers to track their status from time to time.

The main benefit of this project is that it facilitate user to communicate in a faster manner in comparison of manual system. Through this system the current status of the Taxi can be known easily whereas in manual system it is a difficult task.

This software manages the daily transactions and also keeps the historical data in the database for future references. The system is used for daily activities. It is very difficult to do this process manually. This project is reliable only when the database is maintained properly.

This software application which avoids more manual hours that needs to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all the users simultaneously. No specific training is required for the employees to use this application. They can easily use this system to decrease manual hours spending for normal things and hence increase the performance.

The customer’s need is fulfilled by Taxi Management System software which is online software for the management people that enables them to receive the goods from a source and send them to a required destination and track their status from time to time.

This system can be used for daily activities such as placing a Taxi and track the Taxi etc.

## 2.2 TABLE DESCRIPTION

### Tables

The existing system is a manual one in which users are maintaining ledgers to store the information, customer details as well as driver details. Maintaining critical information in the files and manuals is full of risk and tedious process.

The System uses the following tables to maintain the information:

- Login
- Cars\_information
- Customer\_information
- Destination\_price\_table
- Driver\_information

The table details are as follows: -

**Table 2.2.1: Login**

The screenshot displays the MySQL Workbench interface. On the left, the 'SCHEMAS' pane shows the 'taximangement' database selected, with a tree view of its tables including 'login\_page'. The main query editor shows a query: `desc taximangement.login_page;`. Below the query editor, the 'Result Grid' shows the table structure:

Field	Type	Null	Key	Default	Extra
Name	varchar(45)	YES		NULL	
user_id	varchar(45)	NO	PRI	NULL	
Password	varchar(45)	YES		NULL	

Below the table structure, the 'Output' pane shows the execution results of the query, including the time taken and the number of rows returned.

#	Time	Action	Message	Duration / Fetch
9	11:33:20	desc taximangement.login_page	3 row(s) returned	0.000 sec / 0.000 sec
10	11:33:33	SELECT * FROM taximangement.cars_table LIMIT 0, 1000	8 row(s) returned	0.156 sec / 0.000 sec
11	11:33:43	desc taximangement.cars_table	5 row(s) returned	0.000 sec / 0.000 sec



**Table 2.2.2: Car information**

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'Schemas' tree with the 'taximangement' database selected. The 'Tables' list under 'taximangement' includes 'booking\_table', 'cars\_table', 'customer\_info', 'destination', 'driver\_info', and 'login\_page'. The 'cars\_table' is selected, and its structure is shown in the 'Result Grid' pane.

**Table: login\_page**

**Columns:**

Column Name	Data Type	Constraints
Name	varchar(45)	
user_id	varchar(45)	PK
Password	varchar(45)	

**cars\_table Structure:**

Field	Type	Null	Key	Default	Extra
D_ID	int	NO	PRI	NULL	
D_Name	varchar(45)	YES		NULL	
Car_Class	varchar(45)	YES		NULL	
Car_No	varchar(45)	YES		NULL	
Availability	varchar(45)	YES		NULL	

**Action Output:**

#	Time	Action	Message	Duration / Fetch
9	11:33:20	desc taximangement.login_page	3 row(s) returned	0.000 sec / 0.000 sec
10	11:33:33	SELECT * FROM taximangement.cars_table LIMIT 0, 1000	8 row(s) returned	0.156 sec / 0.000 sec
11	11:33:43	desc taximangement.cars_table	5 row(s) returned	0.000 sec / 0.000 sec

Query Completed

**Table 2.2.3: Customer information**

MySQL Workbench interface showing the 'taximangement' database schema. The 'customer\_info' table is selected in the 'Tables' list. The table structure is displayed in the 'Result Grid'.

Field	Type	Null	Key	Default	Extra
C_ID	int	NO	PRI	NULL	
C_Name	varchar(45)	YES		NULL	
SLocation	varchar(45)	YES		NULL	
ELocation	varchar(45)	YES		NULL	
C_Pho	int	YES		NULL	
C_AccNo	int	YES		NULL	
Car_Class	varchar(45)	YES		NULL	
Date_Of_Journey	varchar(45)	YES		NULL	

Table: login\_page

Columns:

- Name: varchar(45)
- user\_id: varchar(45) PK
- Password: varchar(45)

Query 1: desc taximangement.customer\_info;

Result 2: SELECT \* FROM taximangement.cars\_table LIMIT 0, 1000

Action Output:

#	Time	Action	Message	Duration / Fetch
9	11:33:20	desc taximangement.login_page	3 row(s) returned	0.000 sec / 0.000 sec
10	11:33:33	SELECT * FROM taximangement.cars_table LIMIT 0, 1000	8 row(s) returned	0.156 sec / 0.000 sec
11	11:33:43	desc taximangement.cars_table	5 row(s) returned	0.000 sec / 0.000 sec

**Table 2.2.4: Destination price**

MySQL Workbench interface showing the 'taximangement' database schema. The 'destination' table is selected in the 'Tables' list. The table structure is displayed in the 'Result Grid'.

Field	Type	Null	Key	Default	Extra
Dest_ID	int	NO	PRI	NULL	
Destination1	varchar(45)	YES		NULL	
Destination2	varchar(45)	YES		NULL	
Distance(in km)	varchar(45)	YES		NULL	
Micro_Price	int	YES		NULL	
Sedan_Price	int	YES		NULL	
SUV_Price	int	YES		NULL	

Table: login\_page

Columns:

- Name: varchar(45)
- user\_id: varchar(45) PK
- Password: varchar(45)

Query 1: desc taximangement.destination;

Result 2: SELECT \* FROM taximangement.cars\_table LIMIT 0, 1000

Action Output:

#	Time	Action	Message	Duration / Fetch
9	11:33:20	desc taximangement.login_page	3 row(s) returned	0.000 sec / 0.000 sec
10	11:33:33	SELECT * FROM taximangement.cars_table LIMIT 0, 1000	8 row(s) returned	0.156 sec / 0.000 sec
11	11:33:43	desc taximangement.cars_table	5 row(s) returned	0.000 sec / 0.000 sec

**Table 2.2.5: Driver information**

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 customer\_info destination driver\_info cars\_table login\_page

SCHEMAS

Filter objects

- dlab
- drinks
- library
- sakila
- sys
- taximangement**
  - Tables
    - booking\_table
    - cars\_table
    - customer\_info
    - destination
    - driver\_info
    - login\_page
  - Views
  - Stored Procedures
  - Functions
- world

Administration Schemas

Information

Table: login\_page

Columns:

- Name varchar(45)
- user id varchar(45) PK
- Password varchar(45)

Object Info Session

Query 1

```
1 desc taximangement.driver_info;
```

Result Grid

Field	Type	Null	Key	Default	Extra
D_ID	int	NO	PRI	NULL	
D_Name	varchar(45)	YES		NULL	
D_Phno	int	YES		NULL	
D_CarClass	varchar(45)	YES		NULL	
D_CarNo	varchar(45)	YES		NULL	
D_LicNo	varchar(45)	YES		NULL	
D_Location	varchar(45)	YES		NULL	
Availability	varchar(45)	YES		NULL	

Result 2 x Read Only

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	11:33:20	desc taximangement.login_page	3 row(s) returned	0.000 sec / 0.000 sec
10	11:33:33	SELECT * FROM taximangement.cars_table LIMIT 0, 1000	8 row(s) returned	0.156 sec / 0.000 sec
11	11:33:43	desc taximangement.cars_table	5 row(s) returned	0.000 sec / 0.000 sec

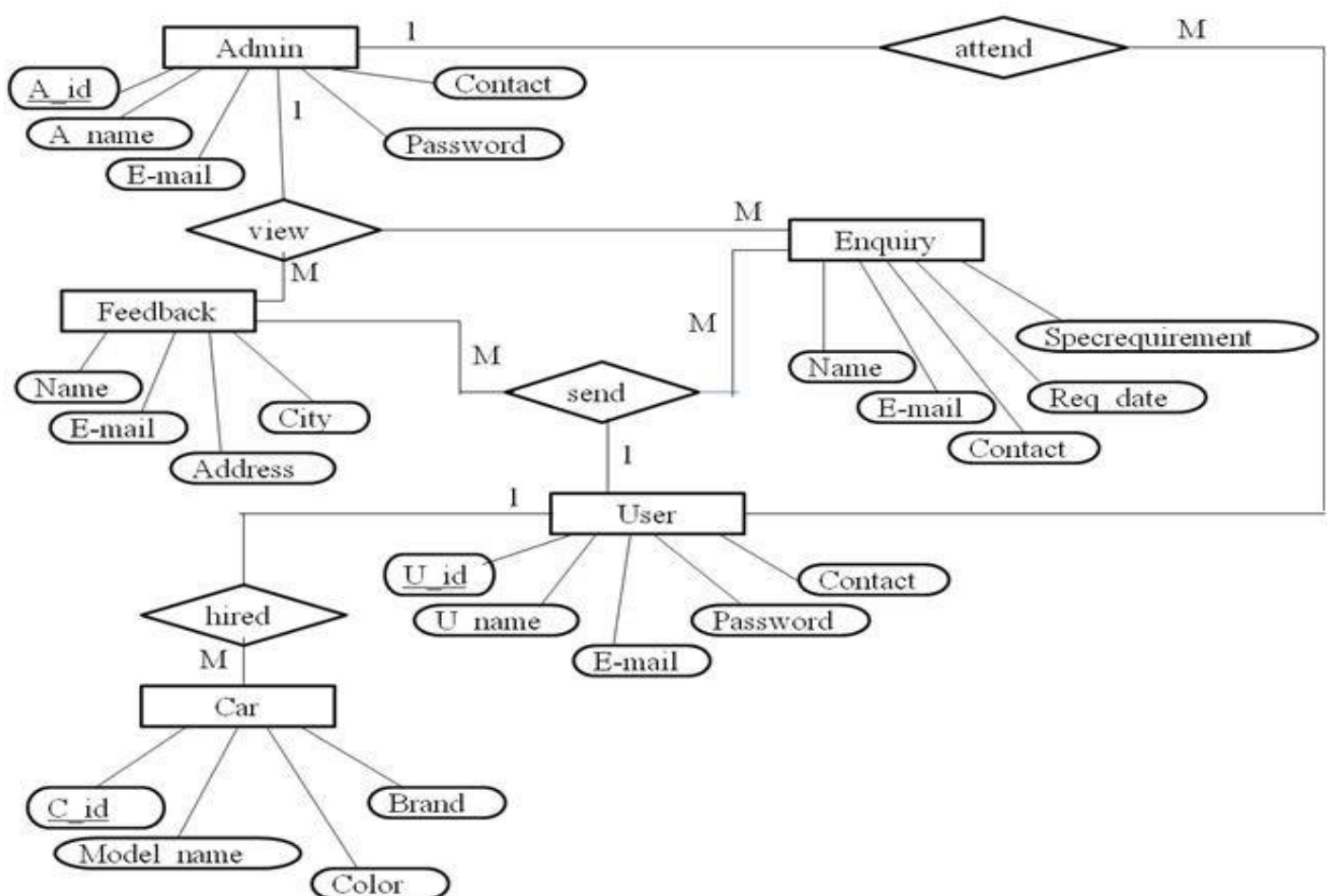
Query Completed

## CHAPTER 3

### DESIGN

#### 3.1 ENTITY RELATIONSHIP DIAGRAM:

ER Relationship model allows us to describe the data involved in a real-world enterprises in terms of objects and their relationship widely used to develop an initial database design. It is primarily important in its role in database design.



**Fig 3.1: ER Diagram for Taxi Management System**

In this ER Diagram shown in Fig 3.1 Entities **admin**, **car**, **user**, **Enquiry**, **feedback**, are represented by rectangles, attributes of the tables are represented by ovals and relationships are represented using diamonds.

## 3.2 NORMALIZATION

### 3.2.1 First Normal Form (1NF)

A Relational Schema R is in 1st Normal Form, if it satisfies the following conditions:

- Contains only atomic values (simple, indivisible).
- There are no repeating groups.

### 3.2.2 Second Normal Form (2NF)

A Relational Schema is in 2nd Normal form, if every non-prime attribute A in R is fully functionally dependent on R.

- It is in 1st Normal Form.
- All non-key attributes is fully functional dependent on primary key.

### 3.2.3 Third Normal Form (3NF)

A Relational Schema R is in 3rd Normal Form, if it satisfies the following conditions:

- If it is in 2nd Normal Form.
- If there is no Transitive Functional Dependency.

The tables in this project satisfies all the three normal forms.

## 3.3 SCHEMA DIAGRAM

A database schema can be represented in a visual diagram, which shows the database object and their relationship which represents the logical view of the database and how the relationships among them are represented.

This Schema Diagram in Fig 3.3 represents different tables used and underlined attributes are primary keys and arrows are used to represent foreign keys.

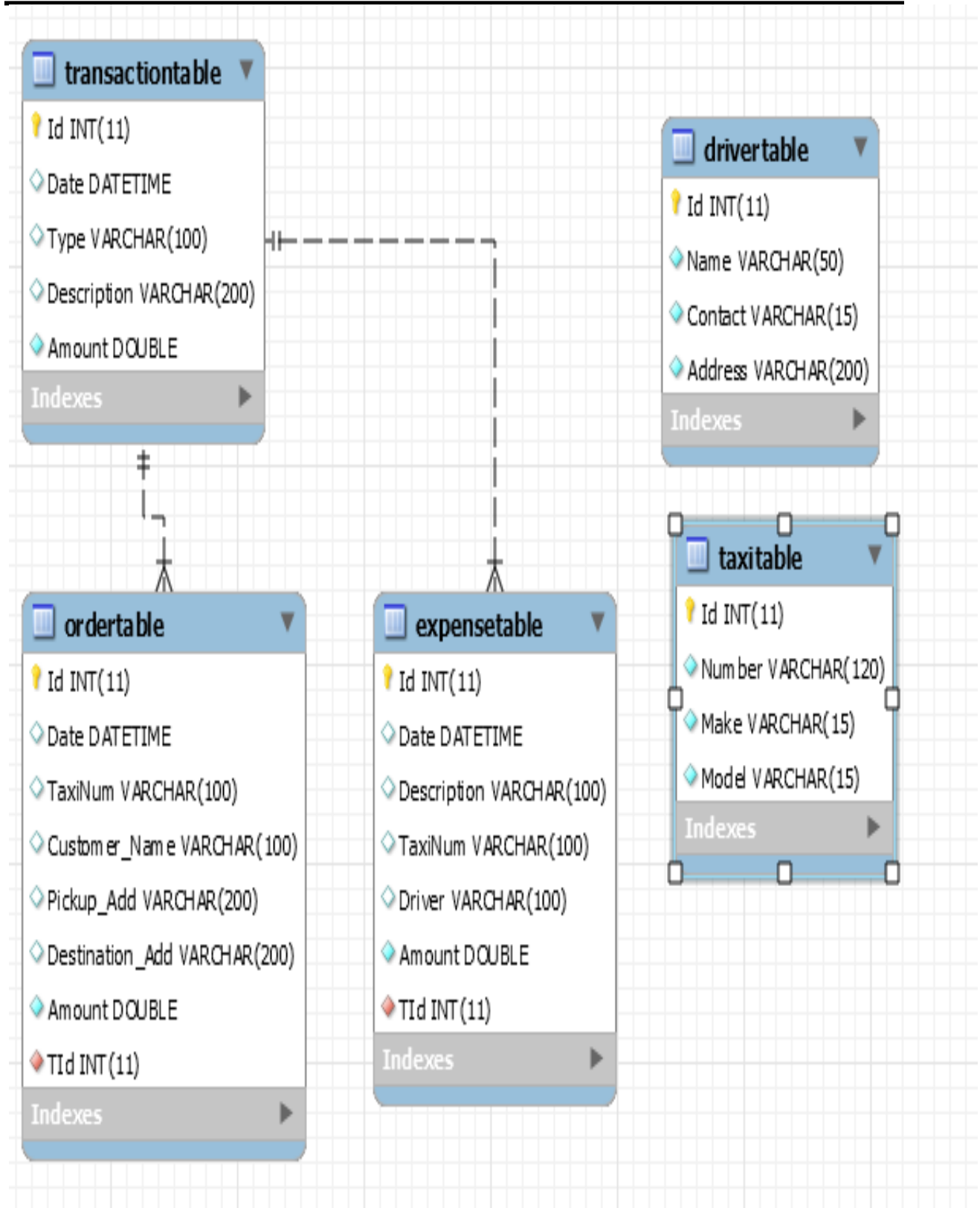


Fig 3.3: Schema Diagram

## **CHAPTER-4**

### **SYSTEM ANALYSIS**

Analysis is the process of breaking a complex topics or substance into smaller parts to gain a better understanding of it. Analysts in the field of engineering look at requirements, structures, mechanisms, and systems dimensions. Analysis is an exploratory activity.

Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down.

Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements.

#### **4.1 Various divisions in the project**

##### **4.1.1 Home and Login**

###### **As Customer**

1. If you login as customer you can book the taxi.
2. It allows you track your taxi.

###### **As Driver**

1. If you login as driver you can pickup the customer.
2. It allows you to update the location and also deliver the customer to the destination.

##### **4.1.2 Log in**

This section allows the customer/Driver to fill in the page for further login.

##### **4.1.3 Contact**

This tells us about the Driver/customer and all his/her personal details and contact details.

## 4.2 User Characteristics

### 4.2.1 End Users

- No specific knowledge or skills are required from the end user.
- End user should have basic idea about computer operations and database.

### 4.2.2 Administrator

- Administrator must be having good knowledge of database management system.
- Administrator must be capable to manage user rights.
- Manual interfaces cannot be fully avoided. Documented proofs like data entry of employees etc. will have to be verified by the concerned management staff before entering it into the computerized system.

## 4.3 Feasibility Study

Every project is feasible for given unlimited resources and infinitive time. Feasibility study is an evaluation of the proposed system regarding its workability, impact on the organization, ability to meet the user needs and effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development. Feasibility and risk analysis and related in many ways. If a project risk is great and feasibility of producing software is reduced. During the feasibility analysis in this project has been discussed below in the abovementioned topics.

### ❖ Technical Feasibility:

Technical feasibility is frequently the most difficult area to ensure this stage. It is essential that the process of analysis and definition to be conducted parallel to an assessment of the technical feasibility.

The consideration that is normally associated with technical feasibility includes the resources availability of the Organization where the project is to be developed and implemented. By taking these facts into consideration before developing the resource availability at Retail Outlet of Hindustan Petroleum was observed. As very limited resources are required for this project hence this project is considered feasible for development.



**❖ Economic Feasibility:**

An evaluation of development cost is weighted against the ultimate income or benefits derived from the developed system. There was no need of extra hardware and software for development of this project. Hence this project has economically justified for development in this organization.

**❖ Operational Feasibility:**

Feasibility of the working of the system after the installation in the organization as mentioned in the feasibility analysis.

**❖ Motivational Feasibility:**

An evaluation of the probability that the organization is sufficient motivation to support the development and implementation of the application with necessary user participation, resources, training etc. The interest and support shown by the organization during the system study do not seem that the new system developed to have efficient support from the organization.

**❖ Schedule Feasibility:**

An evaluation of the time needed for the development of this project. The time schedule required for the development of this project is very important, since more development time effects machine time, costs and delays in the development of the other systems. So the project should be complete within affixed schedule time as far as the organization is concerned.

## **CHAPTER-5**

# **SYSTEM REQUIREMENTS**

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.

Characteristics of good SRS:

- ✓ Complete.
- ✓ Consistent.
- ✓ Feasible.
- ✓ Modifiable.
- ✓ Unambiguous.
- ✓ Testable.

## **5.1 Requirements**

### **5.1.1 Software Requirements:**

Operating System	: Windows
Programming Language	: JAVA
Database	: MYSQL

### 5.1.2 Software Features:

➤ JAVA

Java is a high-level programming language developed by Sun Microsystems. It was originally designed for developing programs for set-top boxes and handheld devices, but later became a popular choice for creating web applications.

➤ SQL

SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS). It is a standard language for Relational Database System. ... All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.

### 5.1.3 Hardware Requirements:

- Processor : INTEL core i3
- MEMORY : 8 GB
- RAM : 8 MB
- HDD : free space of 50 G

### 5.1.4 SECURITY REQUIREMENTS

Some of the factors that are identified to protect the software from accidental or malicious access, use, modification, destruction, or disclosure are described below.

- Ascertain functions to different modules
- Restrict communication between areas of the program
- Check data integrity for critical variables
- A later version of the software will incorporate encryption techniques in the user/license authentication process
- Communication needs to be restricted when the application is validating the user or license

## **CHAPTER-6**

# **TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub- assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

### **6.1 Levels of Testing**

#### **6.1.1 Unit testing:**

Unit testing focuses verification effort on the smallest unit of software design- the software component or module. The unit test is white-box oriented. The unit testing implemented in every module of student attendance management System. by giving correct manual input to the system ,the data's are stored in database and retrieved. If you want required module to access input or get the output from the End user. Any error will accrued the time will provide handler to show what type of error will accrued.

#### **6.1.2 System testing:**

System testing is actually a series of different tests whose primary purpose is to fully exercCSE the computer-based system. Below we have described the two types of testing which have been taken for this project. it is to check all modules worked on input basis .if you want change any values or inputs will change all information so specified input is must.

## 6.2 Test cases

Test case is an object for execution for other modules in the architecture does not represent any interaction by itself. A test case is a set of sequential steps to execute a test operating on a set of predefined inputs to produce certain expected outputs. There are two types of test cases:-manual and automated. A manual test case is executed manually while an automated test case is executed using automation.

In system testing, test data should cover the possible values of each parameter based on the requirements. Since testing every value is impractical, a few values should be chosen from each equivalence class. An equivalence class is a set of values that should all be treated the same.

Ideally, test cases that check error conditions are written separately from the functional test cases and should have steps to verify the error messages and logs. Realistically, if functional test cases are not yet written, it is ok for testers to check for error conditions when performing normal functional test cases. It should be clear which test data, if any is expected to trigger errors.

### Test Cases:

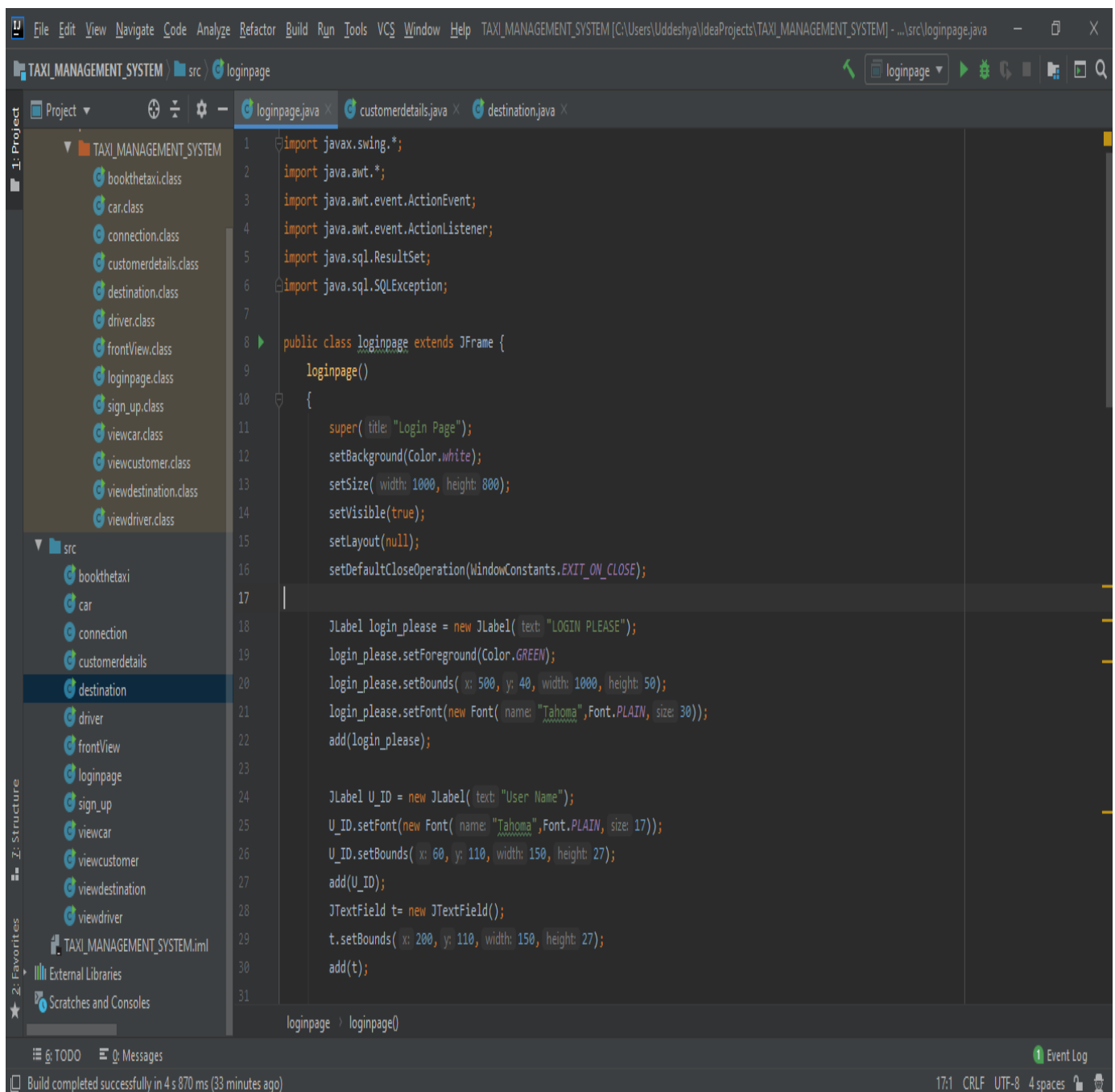
SI No	Test case Id	Test case name	Test case Description	Step	Expected result	Actual Result	Test case Status pass/fail
1	Login Driver	Validate login	To verify that login name on login page.	Enter the Login name and password and click Admin button	Login successful or error message "Invalid Username or Password".	Login successful	Pass
2	Login Customer	Validate login	To verify that login name on login page.	Enter the Login name and password and click Employee button.	Login successful or error message "Invalid Username or Password".	Login successful	Pass

**Table 6.2 Testcases**

## CODE SNIPPETS

### A.1 LOGIN AUTHENTICATION QUERY

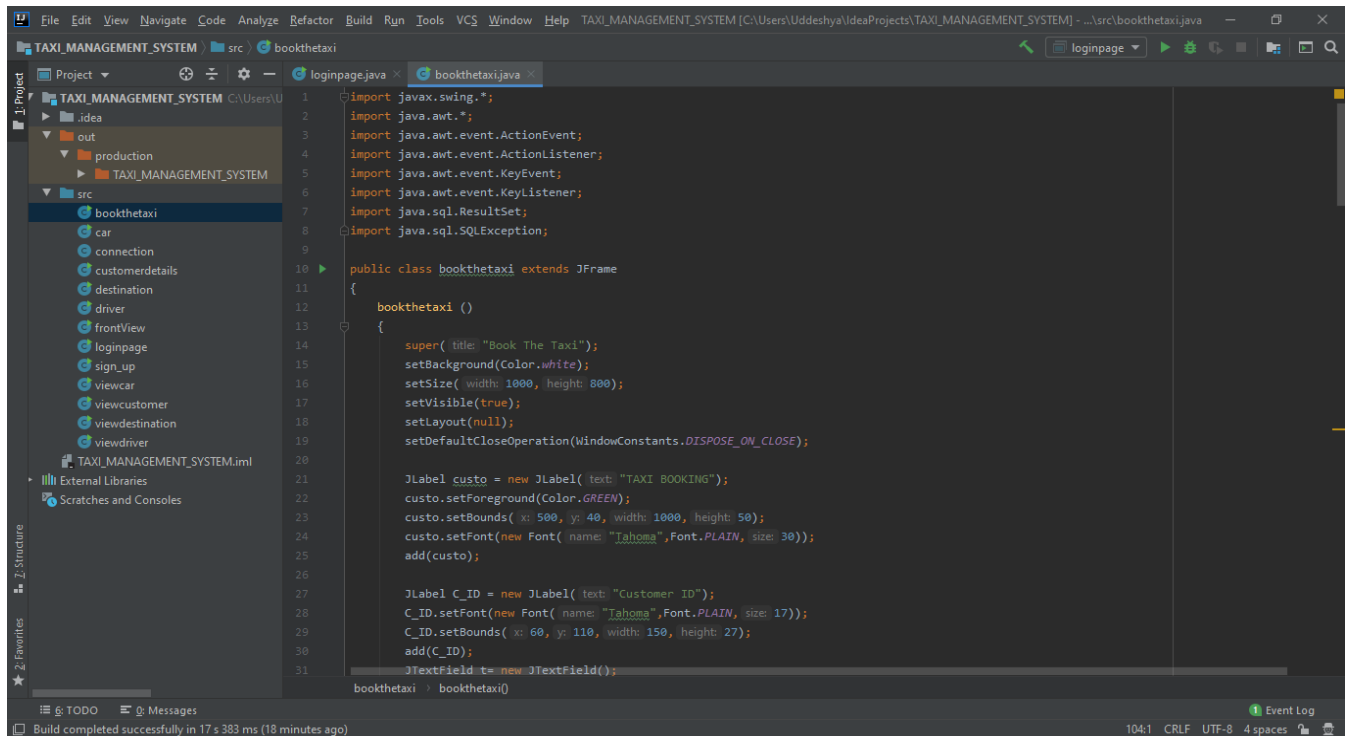
This query recognizes the user's identity. It associates an incoming request with the set of identifying credentials.



```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7
8 public class loginpage extends JFrame {
9     loginpage()
10    {
11        super( "Login Page");
12        setBackground(Color.white);
13        setSize( width: 1000, height: 800);
14        setVisible(true);
15        setLayout(null);
16        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
17
18        JLabel login_please = new JLabel( text: "LOGIN PLEASE");
19        login_please.setForeground(Color.GREEN);
20        login_please.setBounds( x: 500, y: 40, width: 1000, height: 50);
21        login_please.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 30));
22        add(login_please);
23
24        JLabel U_ID = new JLabel( text: "User Name");
25        U_ID.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 17));
26        U_ID.setBounds( x: 60, y: 110, width: 150, height: 27);
27        add(U_ID);
28        JTextField t= new JTextField();
29        t.setBounds( x: 200, y: 110, width: 150, height: 27);
30        add(t);
31    }
32 }
```

## A.2 BOOK\_THE\_TAXI

This query is to book the taxi.



```

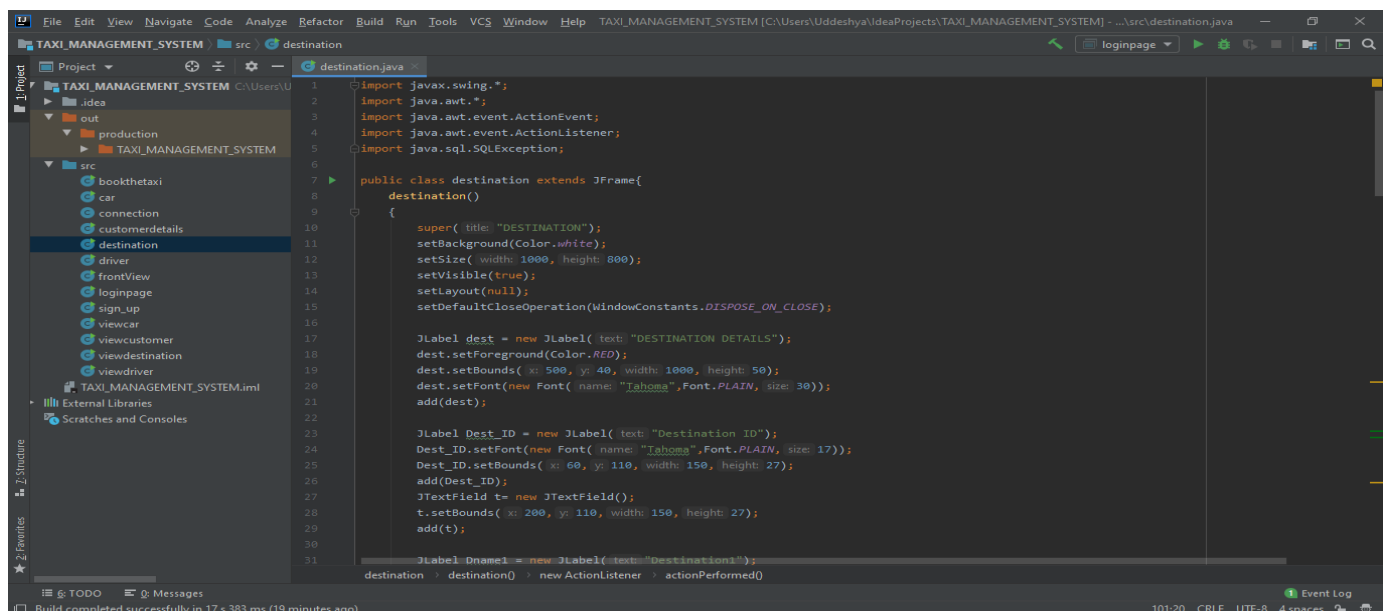
1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.awt.event.KeyEvent;
6  import java.awt.event.KeyListener;
7  import java.sql.ResultSet;
8  import java.sql.SQLException;
9
10 public class bookthetaxi extends JFrame
11 {
12     bookthetaxi ()
13     {
14         super( title: "Book The Taxi");
15         setBackground(Color.white);
16         setSize( width: 1000, height: 800);
17         setVisible(true);
18         setLayout(null);
19         setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
20
21         JLabel custo = new JLabel( text: "TAXI BOOKING");
22         custo.setForeground(Color.GREEN);
23         custo.setBounds( x: 500, y: 40, width: 1000, height: 50);
24         custo.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 30));
25         add(custo);
26
27         JLabel C_ID = new JLabel( text: "Customer ID");
28         C_ID.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 17));
29         C_ID.setBounds( x: 60, y: 110, width: 150, height: 27);
30         add(C_ID);
31         JTextField t= new JTextField();
32         bookthetaxi > bookthetaxi()

```

## A.3 UPDATE QUERY

### A.3.1 Location and Destination Updating

This query is for Location updating and to check whether the respective taxi is either mini ,micro or sedan and to check the status of the destination whether it has been successfully booked or not.



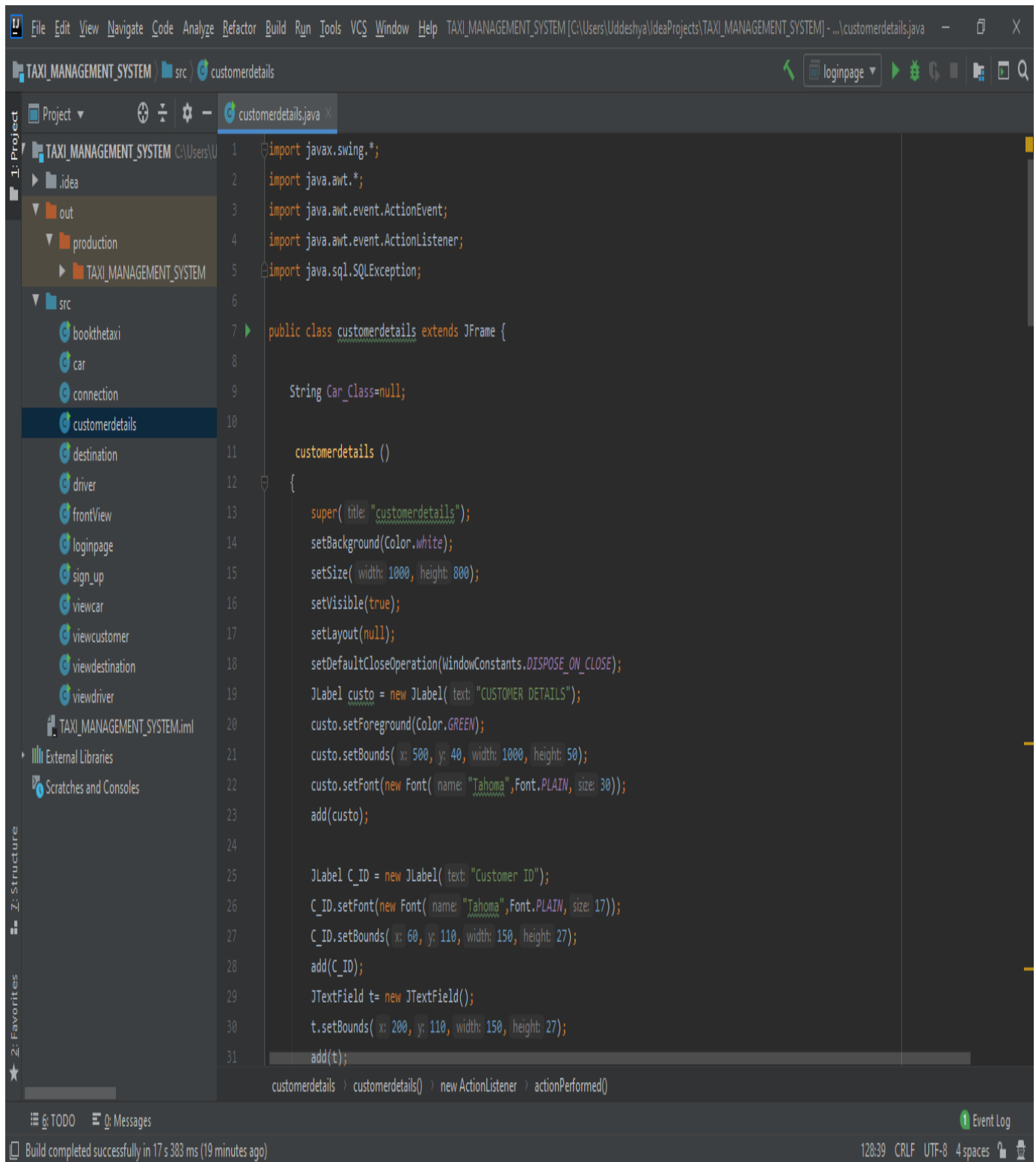
```

1  import javax.swing.*;
2  import java.awt.*;
3  import java.awt.event.ActionEvent;
4  import java.awt.event.ActionListener;
5  import java.sql.SQLException;
6
7  public class destination extends JFrame{
8      destination()
9      {
10         super( title: "DESTINATION");
11         setBackground(Color.white);
12         setSize( width: 1000, height: 800);
13         setVisible(true);
14         setLayout(null);
15         setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
16
17         JLabel dest = new JLabel( text: "DESTINATION DETAILS");
18         dest.setForeground(Color.RED);
19         dest.setBounds( x: 500, y: 40, width: 1000, height: 50);
20         dest.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 30));
21         add(dest);
22
23         JLabel Dest_ID = new JLabel( text: "Destination ID");
24         Dest_ID.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 17));
25         Dest_ID.setBounds( x: 60, y: 110, width: 150, height: 27);
26         add(Dest_ID);
27         JTextField t= new JTextField();
28         t.setBounds( x: 200, y: 110, width: 150, height: 27);
29         add(t);
30
31         JLabel Dname1 = new JLabel( text: "Destinations");
32         destination > destination() > new ActionListener > actionPerformed()

```

### A.3.2 Profile Updating query

This query is used for editing and updating the profile of the user.



```
1 import javax.swing.*;
2 import java.awt.*;
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5 import java.sql.SQLException;
6
7 public class customerdetails extends JFrame {
8
9     String Car_Class=null;
10
11     customerdetails ()
12     {
13         super( title: "customerdetails");
14         setBackground(Color.white);
15         setSize( width: 1000, height: 800);
16         setVisible(true);
17         setLayout(null);
18         setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
19         JLabel custo = new JLabel( text: "CUSTOMER DETAILS");
20         custo.setForeground(Color.GREEN);
21         custo.setBounds( x: 500, y: 40, width: 1000, height: 50);
22         custo.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 30));
23         add(custo);
24
25         JLabel C_ID = new JLabel( text: "Customer ID");
26         C_ID.setFont(new Font( name: "Tahoma",Font.PLAIN, size: 17));
27         C_ID.setBounds( x: 60, y: 110, width: 150, height: 27);
28         add(C_ID);
29         JTextField t= new JTextField();
30         t.setBounds( x: 200, y: 110, width: 150, height: 27);
31         add(t);
32     }
33
34     customerdetails() {
35         new ActionListener() {
36             actionPerformed()
37         }
38     }
39 }
```

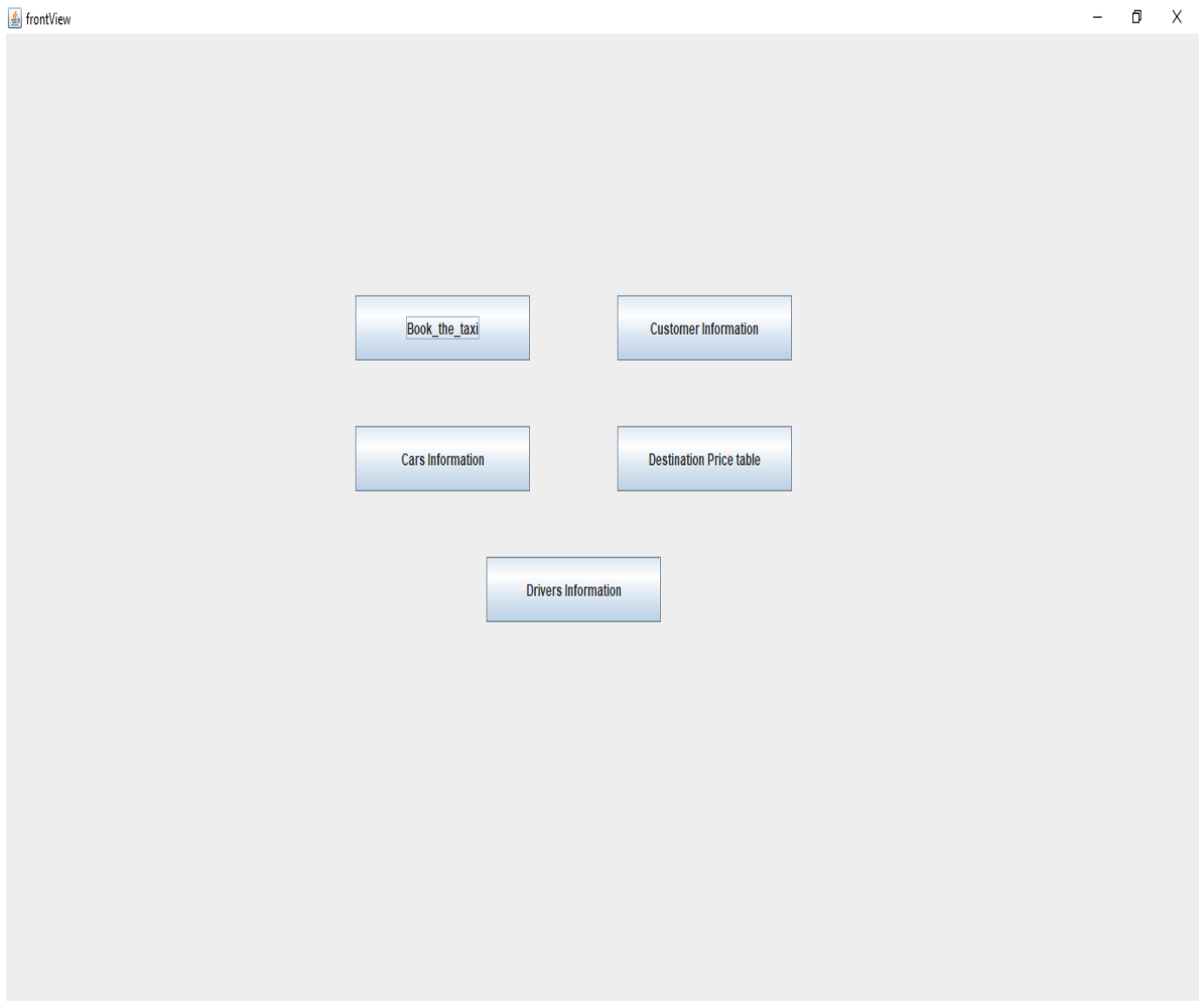


---

# SCREENSHOTS

## B.1. HOME PAGE

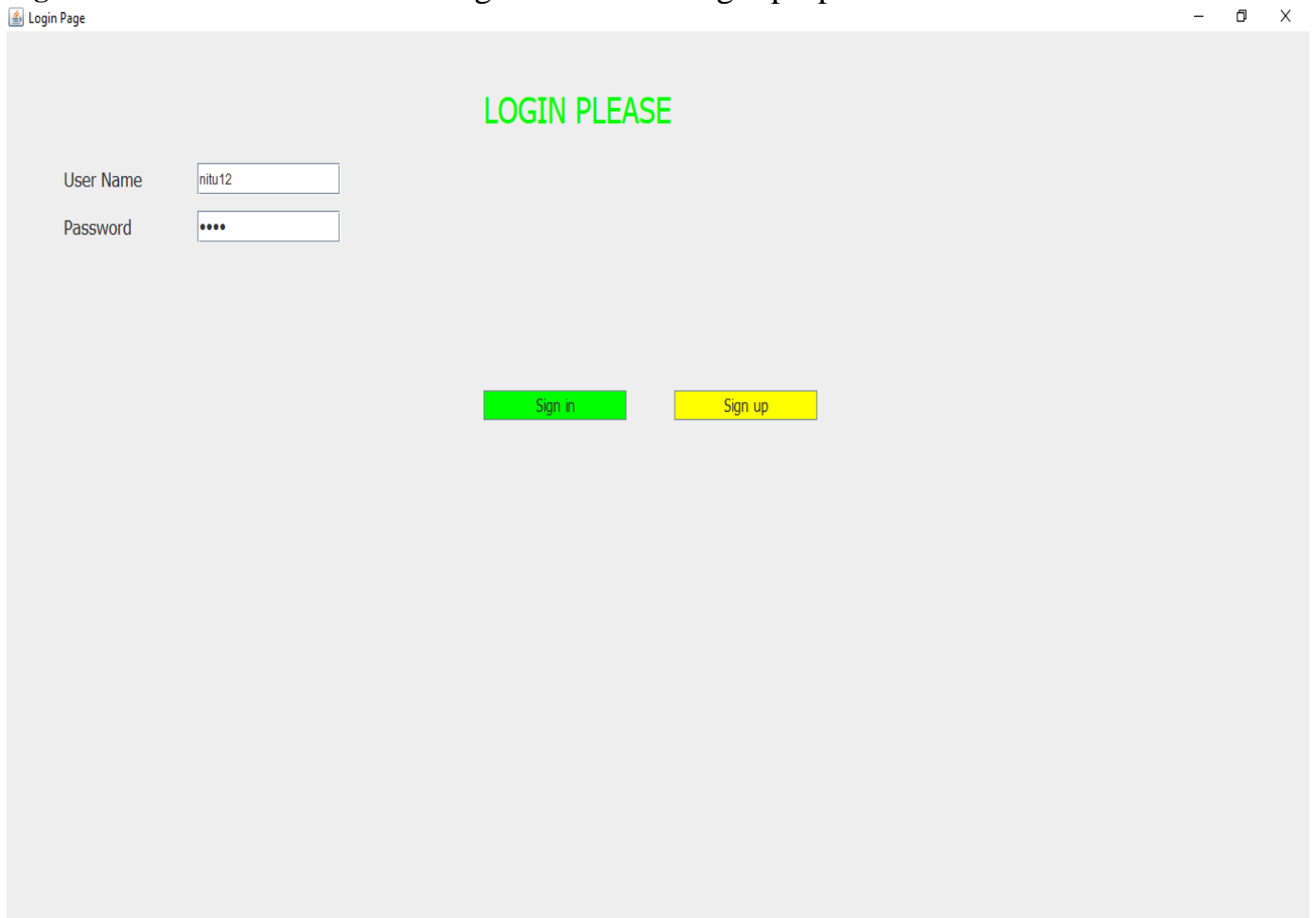
**Figure B.1:** shows the Home page. It shows the buttons/navigation bars. By clicking them it will go to the appropriate pages.



**Figure B.1: Home page**

## B.2 LOGIN PAGE

**Figure B.2** shows the buttons/navigation bars for login purpose of the customer and driver.



The screenshot shows a web browser window titled "Login Page". The page has a light gray background. At the top center, the text "LOGIN PLEASE" is displayed in green. Below this, there are two input fields: "User Name" with the text "nitu12" and "Password" with four dots. At the bottom center, there are two buttons: a green "Sign in" button and a yellow "Sign up" button.

**Figure B.2 Login page**

---

## B.3 CUSTOMER DETAILS PAGE

**Figure B.3** shows the registration process. Here, the customer needs to enter the details so that it will be added.

**CUSTOMER DETAILS**

Customer ID:

NAME:

Starting Location:

Ending Location:

Contact Number:

Account Number:

Date of Journey:

CAR CLASS: ☐ MICRO ☒ MINI ☐ SEDAN

**Message**

Customer Added

OK

**SAVE** **VIEW**

**Figure B.3 Customer details**

## B.4 DRIVER DETAILS PAGE

**Figure B.5** Shows driver Page, where he/she can give his/her details.

**DRIVER DETAILS**

Driver ID:

Driver Name:

Driver Phone Number:

Driver Car Class:

Driver Car Number:

Driver License Number:

Driver Location:

Driver Availability:

**SAVE** **VIEW**

**Figure B.4 Driver details page**

## B.5 DESTINATION PAGE

**Figure B.5** Shows Customer destination Page, where he/she can edit their destination.



DESTINATION

### DESTINATION DETAILS

Destination ID

Destination1

Destination2

Distance

Micro Price

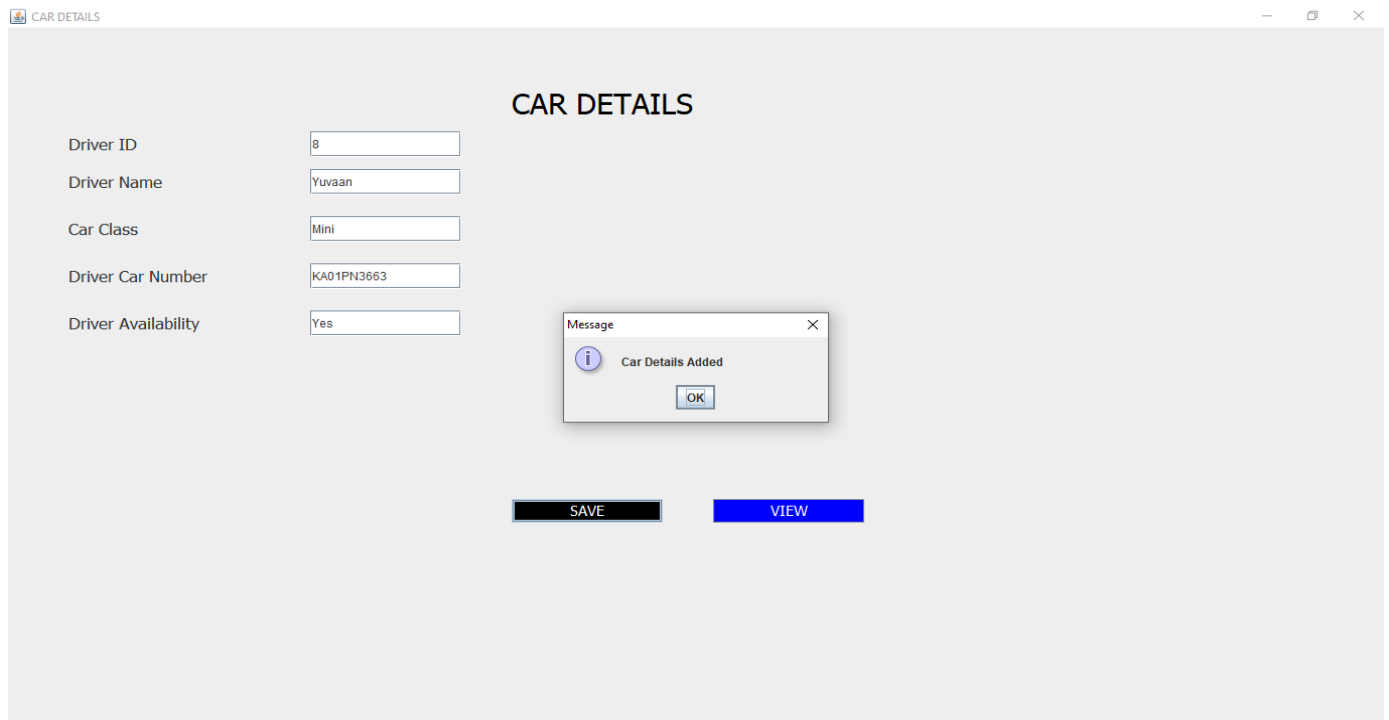
Mini Price

Sedan Price

**Figure B.5 Destination Page**

## B.6 CAR DETAILS PAGE

**Figure B.6** Shows Car details Page where they can track their booking by entering their ID.



CAR DETAILS

Driver ID


Driver Name

Car Class

Driver Car Number

Driver Availability

Message

 Car Details Added

**Figure B.6 Car details Page**

## B.7 CUSTOMER TAXI BOOKING PAGE

**Figure B.8** Shows Customer taxi booking Page, where they can book the taxi according to their wish.

The screenshot displays a web application window titled "Book The Taxi". The main heading "TAXI BOOKING" is centered at the top in green. On the left, there is a form with the following fields and values:

Field	Value
Customer ID	15
NAME	Vikas
Starting Location	Bangalore Cant St.
Ending Location	Sanjay Nagar
Contact Number	26780742
Account Number	123455
Date of Journey	Mini
CAR CLASS	09-JAN-20

Below the form is a green "BOOK" button. A modal message box is displayed in the center, titled "Message", with an information icon and the text "Congrats, YOUR TAXI IS BOOKED". An "OK" button is at the bottom of the message box.

**Figure B.7** Customer taxi booking Page

## CONCLUSION AND FUTURE SCOPE

This project is only a humble venture to satisfy the needs of the customer and driver. The project “Taxi Management System” is designed in order to avoid more manual hours that need to spend in record keeping and generating reports. This application keeps the data in a centralized way which is available to all users and employees simultaneously. They can book the Taxi and track the details online with just a few clicks.

The following sections describe the future scope of the system:

- Host the platform on online servers to make it accessible worldwide
- Add different payment options.
- Gives information to the customer about the person who is driving the Taxi.
- Gives notification to the customer when the destination is reached.

## References

- [1] <http://www.tutorialspoint.com/Php/>
- [2] <http://www.tutorialspoint.com/mysql/>
- [3] <http://www.w3schools.com>