Chapter 1

Introduction

Overview

This project report is regarding database management based on reviewing automobiles (cars and bikes) and details of vehicles that are available in the market. This system consists of reviewing and displaying reviews and rating of other customers for all the vehicles available in the database.

It has been developed using front-end and back-end platforms and a platform to connect both the ends and locally hosted. Most the development and functioning has been designed using netbeans framework.

Objective

- A system consisting of vehicles sorted based on company and their respective models.
- Particular selection of a vehicle and display all the necessary details and specifications.
- A system to display reviews and rating regarding a particular vehicle.
- A system to review and give ratings for a particular vehicle by a customer.

Methodology

To implement the above goals, the following methodology needs to be followed.

- Specifying the Application and various components of the Architecture.
- Collecting the data of individual data elements.
- Implementing the system to incorporate the data with the architecture.
- Analysis of the functioning of the system satisfying all the requirements.

Chapter 2

System Analysis

According to certain sources overall statistics and analytic data has been obtained regarding the sales of automobiles. There has been a exponential growth in the sales of automobiles and customers use automobile review websites for getting feedback and to make choice on the basis of reviews given by other customers and automobile experts. A 2013 study conducted by Dimensional research showed the 90% of customers were influenced into buying a vehicle based on reading positive reviews.

2.1 Existing System Explanation

The existing system of automobile reviewing system consists of the following:

- Customer can only find reviews of a particular type of automobile.
- There are separate websites for different type of automobiles like a particular website exclusively for cars, bikes or heavy vehicles.
- Reviews for all vehicles are not available so customers are supposed to checkout different websites.

The proposed system overcomes the above problems that are in existing system with more improvements and addition of more features.

2.2 Proposed System Explanation

The proposed system was developed to overcome all the problems that were present in the existing system. It provides an integrated system which reduces the overhead of visiting different websites.

- This system brings all kinds of automobiles under one roof.
- Customers can get reviews about any kind of vehicles which includes bikes, cars, trucks, buses etc.
- All are categorized separately to make searching for a product more convenient.
- Separate review system for each category of automobile from customers and automobile experts.

Chapter 2

2.3 System Requirements

The system requires following software and hardware requirements :

Software Requirements

- Operating System - Windows 10 (32 bit)
- Server - XAMPP
- Front End - Net beans framework
- Back End - 1] MySQL 2] phpMyAdmin

Hardware Requirements

- HDD/SSD - 20GB or more
- Processor - Intel core i3, 1 GHz or more
- RAM - 2GB DDR3

Chapter 3

System Design

Design engineering encompasses the set of principles, concepts and practices to lead to the development of high quality product. The goal of design engineering is to produce a model that exhibits firmness, commodity and delight. The following analysis models are used translated into a design model.

3.1 Use Case Diagram

A Use-case captures the interaction that occurs between the producer and consumers and the system itself. As an example of the use case we have chosen the administrator adding new entities (cars, bikes) and reviews into the existing database and users can view these records and write their own reviews.

Use Case: Add a new car or bike

Primary Actor: Administrator

Goal in Context: Enter the review of the entity into the database

Preconditions: System has been programmed for password for various users

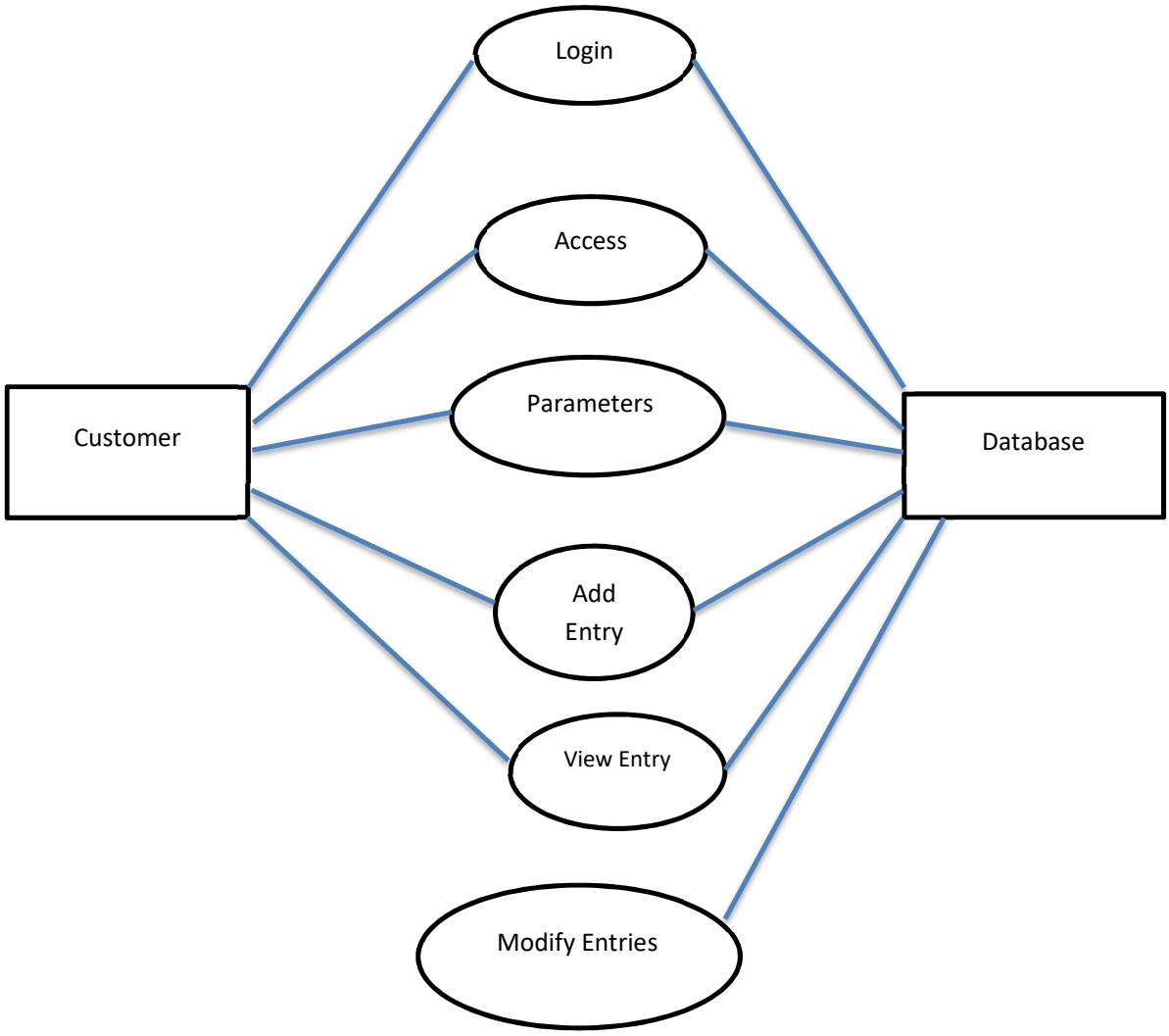
Scenario:

- Admin enters the username and password.
- Admin selects add new entity.
- Admin fills all the fields.
- Admin records the entry in database.
- User signs up using name, email and password.
- User signs in using username and password.
- User can add vehicle name and company only.
- User can search a vehicle using drop down menu and get details including image.
- User can view reviews given by other customers.
- User can add own reviews.

Exceptions:

- Password is incorrect. Prompts for try again.
- All the details are not filled. Prompts for filling all the details.

Fig 3.1 Use Case Diagram



3.2 Schema Diagram

The description of the database is called the Database Schema, which is specified during the design of database and is not expected to change frequently. Most data models have certain conventions for displaying schemas as diagrams. A displayed schema is called a Schema diagram.

The following figure shows the schema diagram for the "Automobile Review" database. The diagram displays the structure of each record type but not the actual instances of record. Each object in the schema such as USER or BIKES is called a schema construct.

Fig 3.2 Schema Diagram

USERS

user_id	Name	Email	Password

REVIEW

Id	Name	Review	Rate

BIKES

Bike_id	Company	Model	Weight	Engine
Mileage	Max speed	Gears	Cost	

CARS

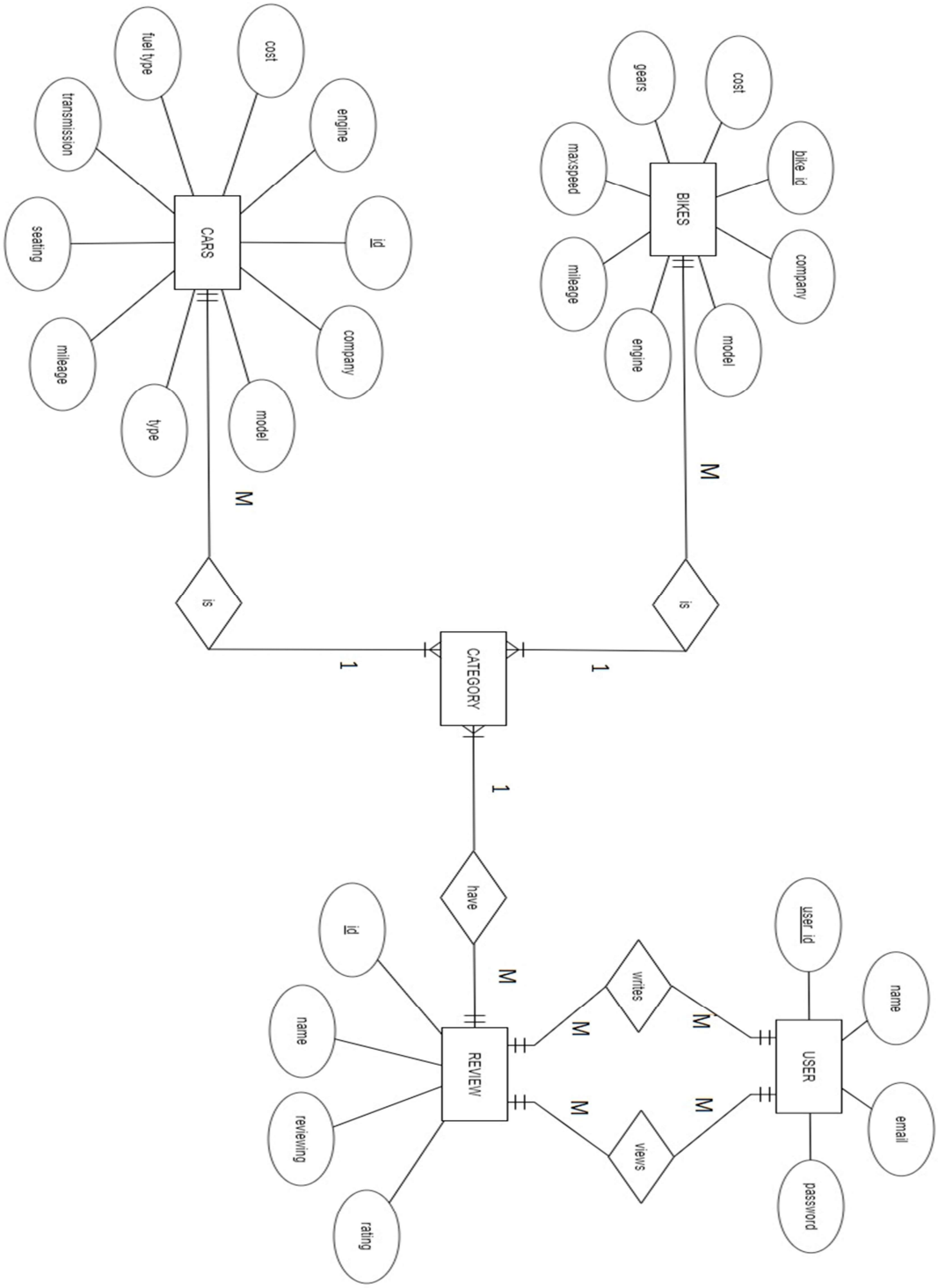
id	Company	Model	Type	Mileage
Seating	Transmission	Fuel Type	Cost	Engine

3.3 Entity-Relationship Diagram

Entity-Relationship Diagram or E-R Diagram is a data modelling technique that can help define business process and can be used as the foundation for a relational database. An ERD is a graphical representation of entities and their relationships to each other. An entity is a real world object about which data is stored. Relationship defines how the entities relate to each other. The following diagram depends on E-R Diagram for the "**Automobile Review System**" database.

The below diagram depicts the number of occurrences of objects/entities in a given relationship. Cardinality is the specification of one entity that can be related to number of occurrences of another entity. For example, one entity can relate to only one other entity (1:1), one entity can relate to many entities (1:m),some occurrences of one entity can relate to some other occurrences of another entity(m:m) , cardinality also defines the maximum number of entities that can participate in a relationship.

Fig 3.3 E-R Diagram



Chapter 4

System Implementation

The software and hardware requirements are stated above that was used to implement the software. The process model used is Extreme Engineering. XP is an agile process model. Agile process models are adaptable and provide scope for requirements changes if any.

The system has an administrator and end users. The administrator has full access to the database. User can view as well as post reviews related to a particular product.

The database includes many features such as:

- Searching vehicle based on type, company and model.
- Adding reviews.
- Storing user reviews.
- Anonymous reviews.
- Viewing details of vehicles.
- Secured access.

4.1 Establish Connection to Database

The following code snippet establishes the connection to the database from the Graphical User Interface.

```
import java.sql.DriverManager;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Statement;

Class.forName("java.sql.DriverManager");
Connection con = (Connection)DriverManager.getConnection
("jdbc:mysql://localhost:3306/project?zeroDateTimeBehavior=convertToNull","root",
"");
Statement stmt = (Statement) con.createStatement();
```

4.2 Searching for vehicle

The vehicle can be searched using drop down menus. One is for company and the other is for model.

Code for company drop down

```
String query ="Select distinct name from car";

 Statement stmt = (Statement) con.createStatement();

  ResultSet rs=stmt.executeQuery(query);
  while(rs.next())
  {
     String name=rs.getString("name");
     jComboBox1.addItem(name);
  }
      }
      catch(Exception e)
         { JOptionPane.showMessageDialog(this, e.getMessage()); }
  }
```

Code for model drop down

```
        String query="SELECT model FROM Car WHERE Name='"+name1+"';";

   Statement stmt = (Statement) con.createStatement();

  ResultSet rs=stmt.executeQuery(query);
  while(rs.next())
  {
     String name=rs.getString("model");
     jComboBox2.addItem(name);
  }
      }
      catch(Exception e)
         { JOptionPane.showMessageDialog(this, e.getMessage()); }
  }
```

Chapter 5

System Testing

Testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. In general, testing is finding out how well something works. In computer hardware and software development, testing is used at key checkpoints in the overall process to determine whether objectives are being met. For example, in software development, product objectives are sometimes tested by product user representatives. When the design is complete, coding follows and the finished code is then tested at the unit or module level by each programmer; at the component level by the group of programmers involved; and at the system level when all the components are combined together. At early or late stages, a product or service may also be tested for usability.

5.1 Unit Testing

Unit Testing is a level of software testing where individual units/components of software are tested. The purpose is to validate that each unit of the software performs are designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/superclass, abstract class or derived/child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module).

Unit testing frameworks, drivers, stubs, and mock/fake objects are used to assist in unit testing.

The modules of project were tested to verify the working of the modules using unit testing. The following results are obtained.

Username	Password	Results
Correct	Correct	Access Granted
Correct	Wrong	Incorrect Password
Wrong	Correct	Incorrect Username
Wrong	Wrong	Incorrect Username

Fig. 5.1 Unit Testing

5.2 Integration testing

Upon completion of unit testing, the units or modules are to be integrated which gives raise to integration testing. The purpose of integration testing is to verify the functional, performance, and reliability between the modules that are integrated.

The three main integration testing strategies are as follows:

- Big Bang: Involves integrating the modules to build a complete software system. This is considered a high-risk approach because it requires proper documentation to prevent failure.
- Bottom-Up: Involves low-level component testing followed by high-level components. Testing continues until all hierarchical components are tested. Bottom-up testing facilitates efficient error detection.
- Top-Down: Involves testing the top integrated modules first. Subsystems are tested individually. Top-down testing facilitates detection of lost module branch links.

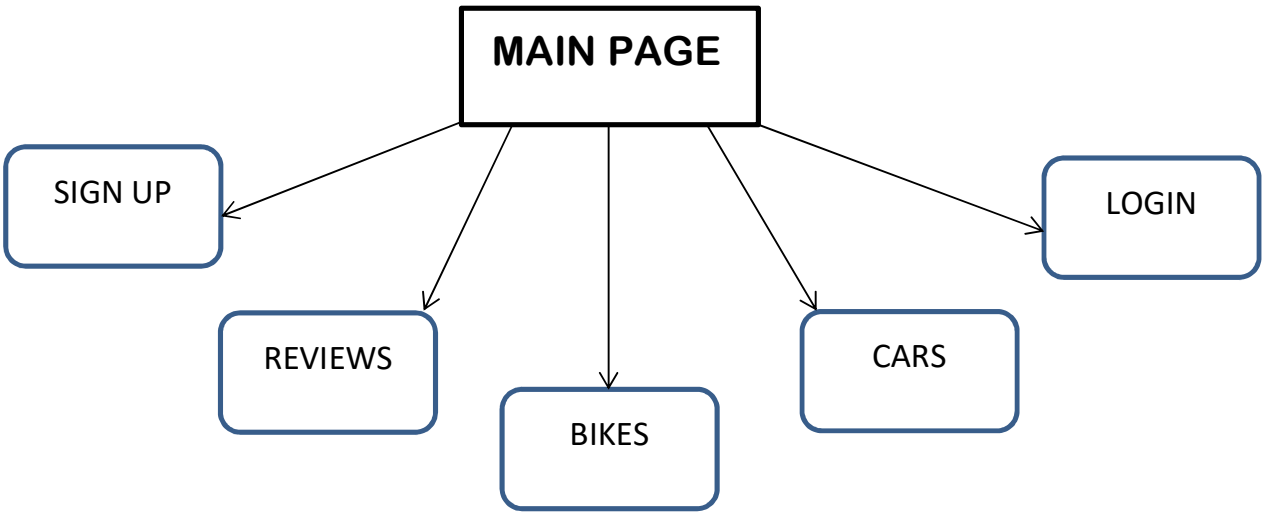


Fig. 5.2 Integration Testing

5.3 Black Box Testing

Black box testing, also known as Behavioural Testing, is a software testing method in which the internal structure/design/implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional.

The software developed was subjected to black-box testing to test the functionality of the Graphical User Interface without checking the internal logic of the program. For Example, the use of back button resulting in the appearance of right frames etc.

5.4 Test Cases

Test Cases	Description	Expected Result	Actual Result	Result
TC01	Execute the application	Application should run without interrupts	Application executing properly	Pass
TC02	Verification of the login with valid username and password	Username and password should be verified with database	Username and password is checked in database	Pass
TC03	Select category on home page	It should allow user to select a category and view all details.	User is able to select particular category and view details	Pass
TC04	Write review and give ratings	User should be able to write review for a particular item belonging to that category and submit it.	Reviews are submitted and can be viewed in the item's page	Pass
TC05	Search and display details	It should allow user to search item in different categories	Search by name displayed properly	Pass
TC06	View review of other users	A user should be able to view all the reviews posted by a particular individual	User are able to view reviews of different user	Pass
TC07	Display specific views	User will be able to see views	User will be able to see views	Pass
TC08	Add vehicle name and model	Admin will be able to view and verify the added details.	Admin will add the following details.	Pass

Fig 5.3 Test Cases

Chapter 6

Results

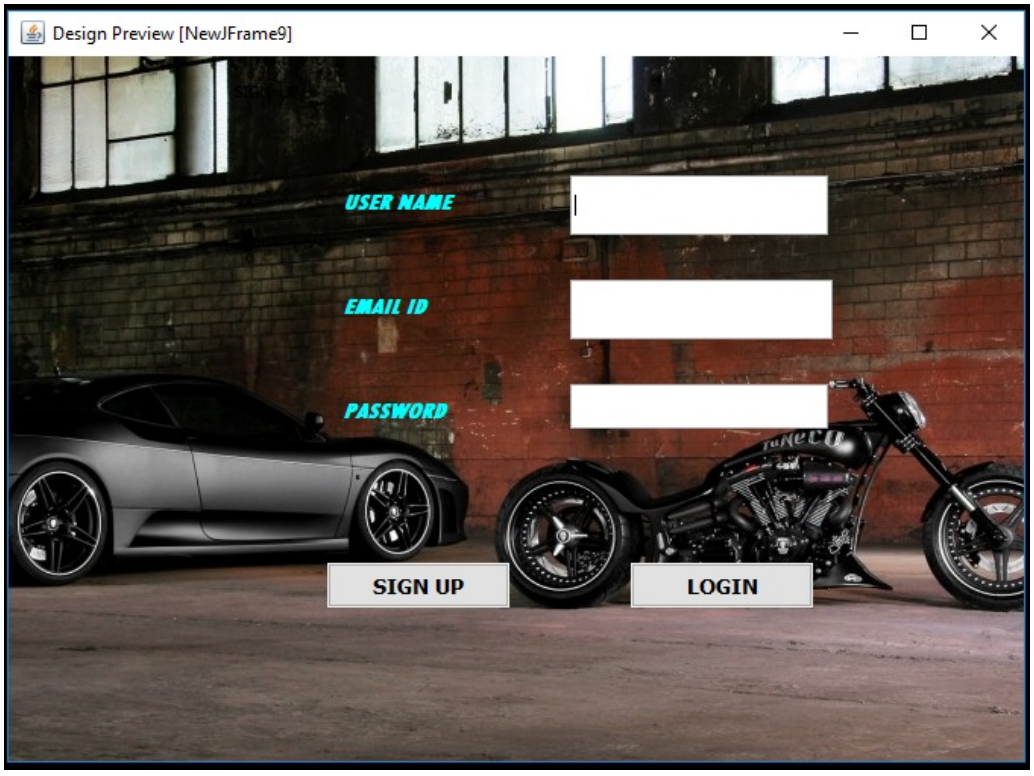


Fig. 6.1 Sign Up Page

Fig. 6.2 Sign In Page

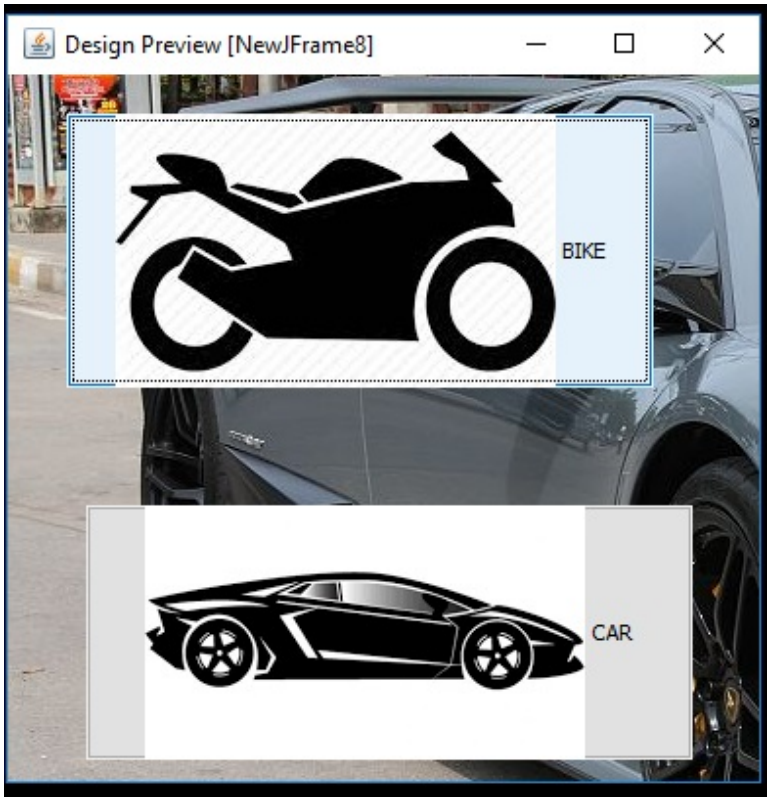


Fig. 6.3 Select Vehicle type

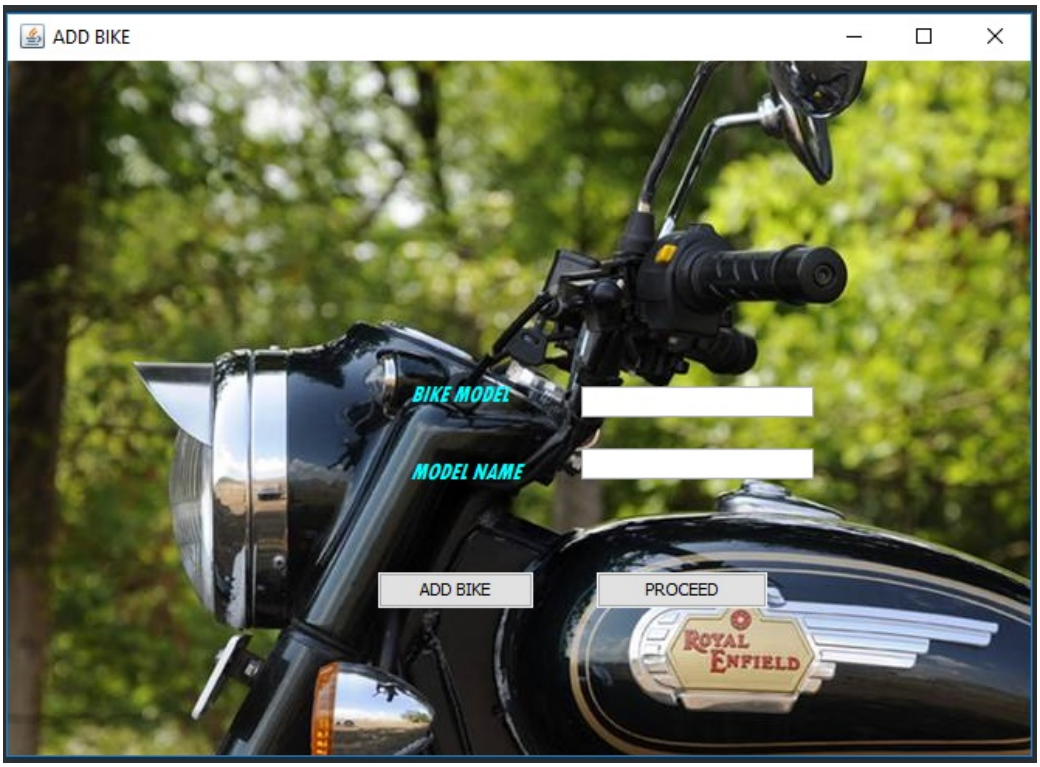


Fig. 6.4 Add Bike

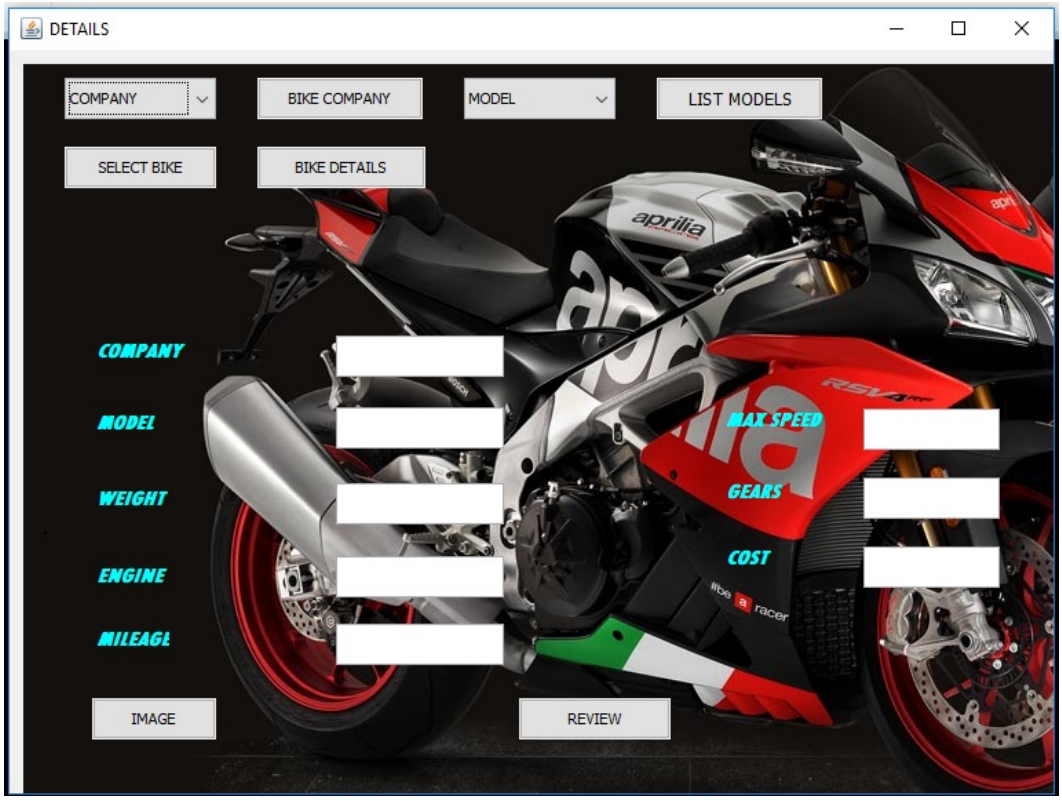


Fig. 6.5 Details of Bike

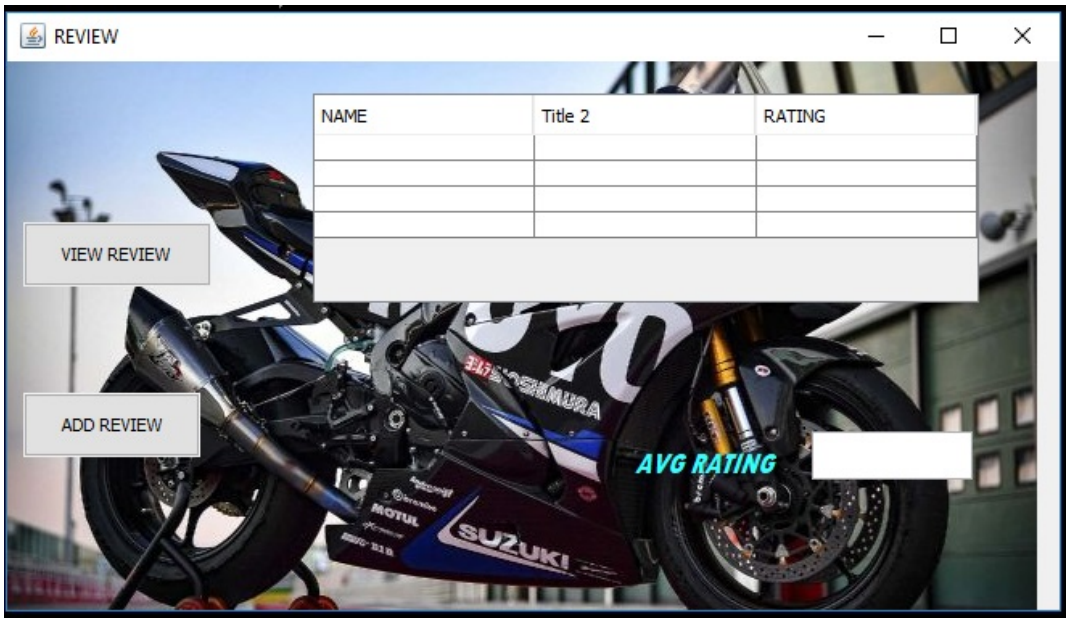


Fig. 6.6 View Reviews

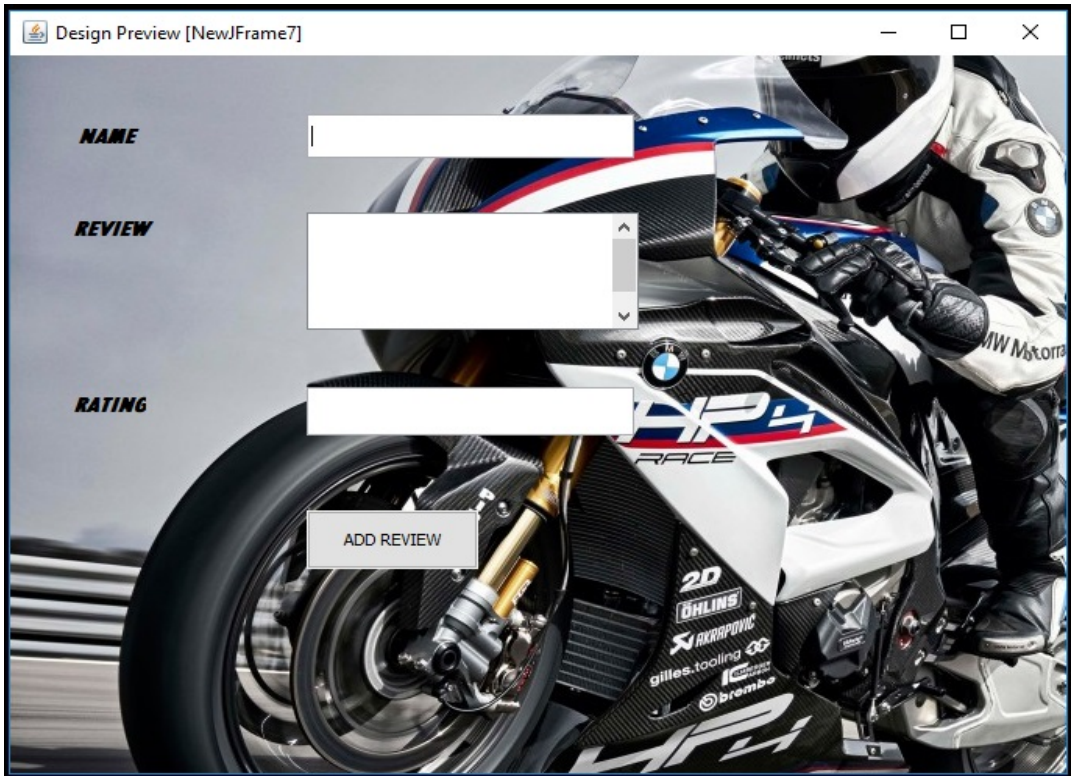


Fig. 6.7 Add Reviews

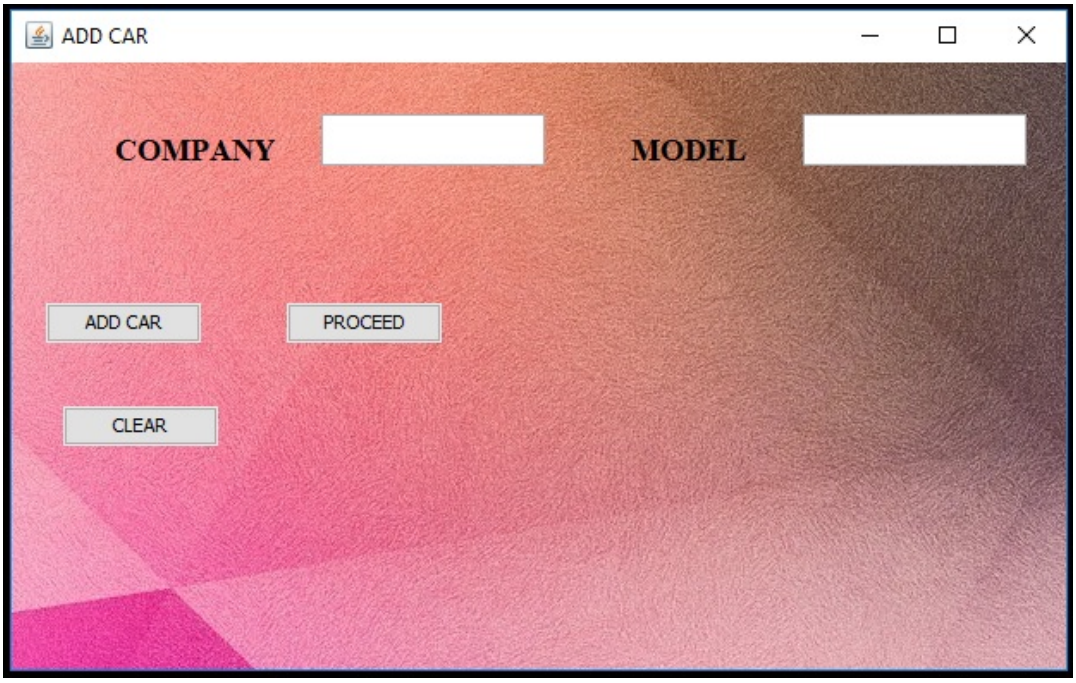


Fig. 6.8 Add Car

Fig. 6.9 Car Details

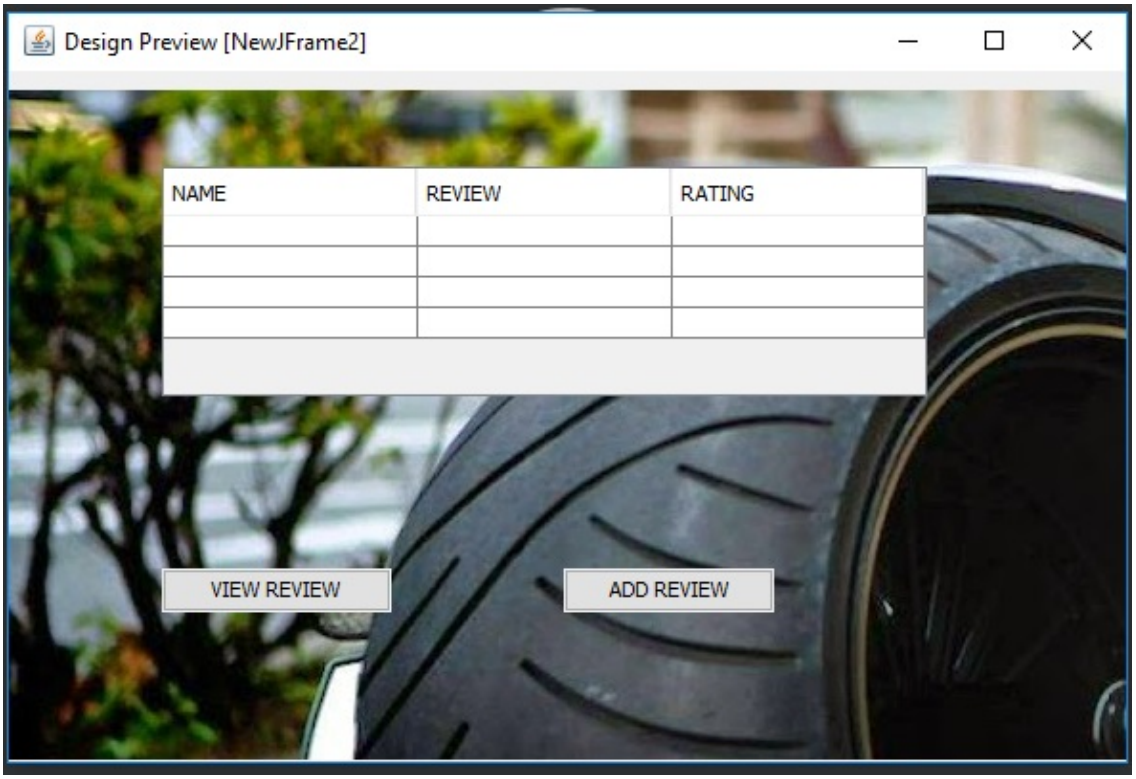


Fig. 7.0 View Reviews

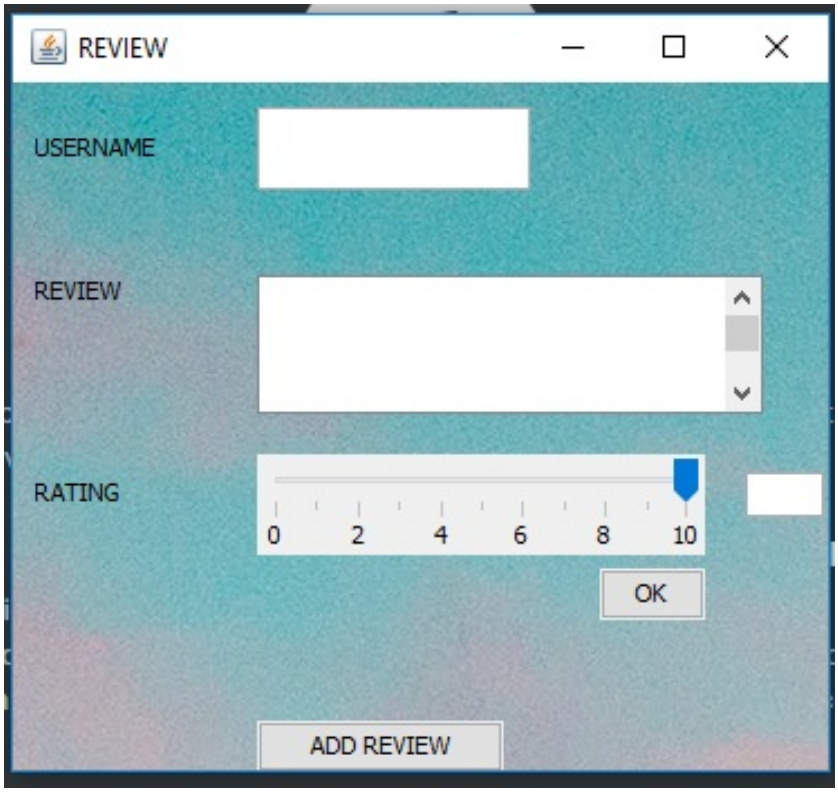


Fig. 7.1 Add Reviews

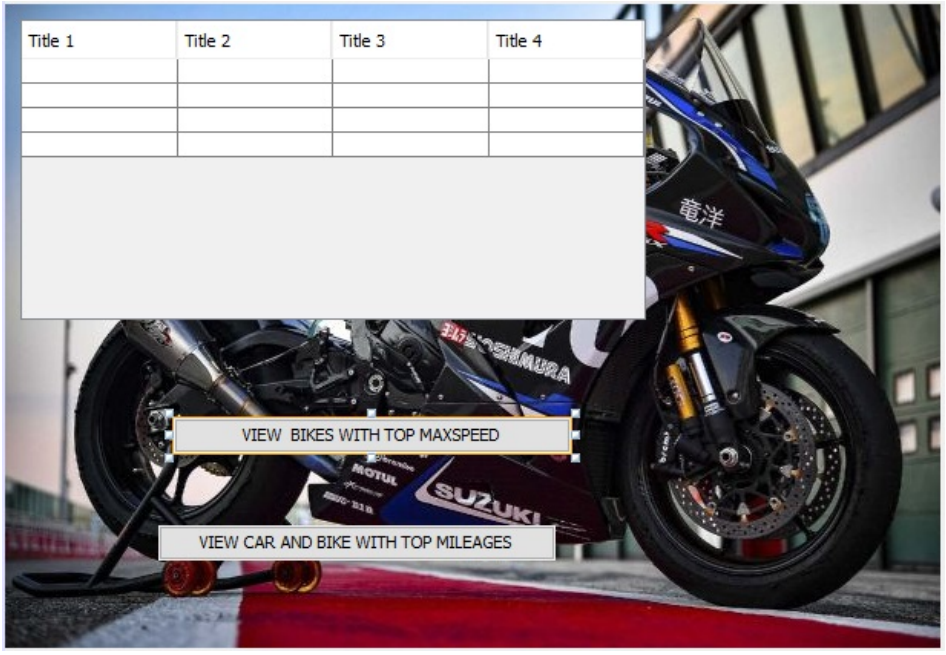


Fig. 7.2 Implementation of views

Conclusion

This project consists of reviews about automobiles which only include cars and bikes. Each individual automobile has its list of specifications which define the functionality and quality of the automobile. Customers can review the vehicles and give rating for the same. This ratings and reviews can be viewed by other customers while buying vehicle of their choice. It helps customers to make right choice as well as help others to take the right decision by posting their own review.

This consists of pictures of each individual automobile and also cost of every vehicle. Users can compare by checking different automobiles. Admin has predefined data in database consisting of 2 to 3 reviews for each individual automobile.

Future Enhancements

This system of reviewing can be expanded for more vehicle types for making it easier for customers to look for automobile of their choice. The system can be made more flexible and accessible for customers by using modern technologies like machine learning algorithms. It can be made accessible on various mobile and pc platforms like iOS, Android by converting it to a mobile application.

References

- https://www.w3schools.com/
- https://www.youtube.com/
- https://erdplus.com/#/
- https://www.tutorialspoint.com/index.htm
- http://www.mysqltutorial.org/
- https://www.autocarindia.com/
- http://overdrive.in/
- https://www.pexels.com/
- https://www.zigwheels.com/