**CAR RENTAL SYSTEM**

A project report on the need to computerize the data processing activities of Car Rental Company

**PREPARED BY**:

**EMMANUEL KIPROTICH KIRUI**

**REG NO**:

**INP04/00014/22**

**SUPERVISOR**:

**MR. KOECH SAMUEL**

**YEAR**:

**2023**

**SCHOOL**:

**KISII UNIVERSITY**

**SUBMITED TO**:

**KISII UNIVERSITY**

**IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF DIPLOMA IN INFORMATION TECHNOLOGY(IT)**

**DATE**

**JUNE, 2023**

**Abstract**

This project report details the development and implementation of a computerized Car Rental System to replace the existing manual processes. The system aims to automate vehicle rental and reservation processes, thereby improving efficiency, reducing errors, and enhancing customer satisfaction. This document includes the objectives, methodologies, tools used, testing procedures, and the system's overall impact.

**Table of Contents**

Contents

[A project report on the need to computerize the data processing activities of Car Rental Company 1](#_Toc170147976)

[Abstract i](#_Toc170147977)

[Table of Contents ii](#_Toc170147978)

[List of Figures x](#_Toc170147979)

[Declaration x](#_Toc170147980)

[Dedication xi](#_Toc170147981)

[Acknowledgment xii](#_Toc170147982)

[Abbreviations xiii](#_Toc170147983)

[Executive Summary xiv](#_Toc170147984)

[Chapter 1: Introduction 16](#_Toc170147985)

[**1.1 Introduction** 16](#_Toc170147986)

[**1.2 Objective of the Project** 16](#_Toc170147987)

[**1.3 Methodology/Procedure** 16](#_Toc170147988)

[**1.4 Project Framework** 16](#_Toc170147989)

[**1.5 Data and Information** 17](#_Toc170147990)

[**1.6 Tools Used** 17](#_Toc170147991)

[**1.7 Testing** 17](#_Toc170147992)

[**1.8 Unit Testing** 17](#_Toc170147993)

[**1.8.1 Integration Testing** 18](#_Toc170147994)

[**1.8.2 System Testing** 18](#_Toc170147995)

[Chapter 2: Task and Activities Performed 19](#_Toc170147996)

[**2.1 System Analysis** 19](#_Toc170147997)

[**2.2 Problem Analysis** 19](#_Toc170147998)

[**2.3 Design and Development Problems** 19](#_Toc170147999)

[**2.4 Feasibility Analysis** 20](#_Toc170148000)

[**2.4.1 Economical Analysis** 20](#_Toc170148001)

[**2.4.2 Software Analysis** 20](#_Toc170148002)

[**2.4.3 Data Conversion** 20](#_Toc170148003)

[**2.4.4 Operational Feasibility** 20](#_Toc170148004)

[**2.5 Use Case Diagram** 21](#_Toc170148005)

[21](#_Toc170148006)

[**2.6 Sequence Diagram** 21](#_Toc170148007)

[22](#_Toc170148008)

[**2.7 Data Flow Diagram** 23](#_Toc170148009)

[23](#_Toc170148010)

[**2.8 ER Diagram** 23](#_Toc170148011)

[23](#_Toc170148012)

[**2.9 Flow Chart** 24](#_Toc170148013)

[24](#_Toc170148014)

[**2.10 Gantt Chart** 25](#_Toc170148015)

[25](#_Toc170148016)

[**2.11 Application’s Output** 26](#_Toc170148017)

[Chapter 3: User Requirements 29](#_Toc170148018)

[Chapter 4: Software Requirement Specification (SRS) 30](#_Toc170148019)

[Chapter 5: Hardware Requirement Specification 31](#_Toc170148020)

[Chapter 6: Database Design 32](#_Toc170148021)

[Chapter 7: Security Considerations 33](#_Toc170148022)

[Chapter 8: Maintenance and Support 34](#_Toc170148023)

[Chapter 9: Code Documentation 35](#_Toc170148024)

[Chapter 10: API Documentation 36](#_Toc170148025)

[**Endpoints:** 36](#_Toc170148026)

[**Parameters:** 36](#_Toc170148027)

[**Responses:** 36](#_Toc170148028)

[**Usage Examples:** 36](#_Toc170148029)

[Chapter 11: Error Handling 41](#_Toc170148030)

[Chapter 12: User Guide 42](#_Toc170148031)

[Chapter 13: Installation Guide 43](#_Toc170148032)

[Chapter 14: Test Plan 45](#_Toc170148033)

[Chapter 15: Test Cases 47](#_Toc170148034)

[Chapter 16: Test Results 50](#_Toc170148035)

[Chapter 17: Project Management 52](#_Toc170148036)

[Chapter 18: Risk Management 54](#_Toc170148037)

[Chapter 19: Conclusion 56](#_Toc170148038)

[Chapter 20: Recommendations 57](#_Toc170148039)

[Chapter 21: References 58](#_Toc170148040)

[Chapter 22: Appendices 59](#_Toc170148041)

[**Appendix A: Use Case Diagram** 59](#_Toc170148042)

[59](#_Toc170148043)

[**Appendix B: Sequence Diagram** 60](#_Toc170148044)

[60](#_Toc170148045)

[**Appendix C: Data Flow Diagram** 61](#_Toc170148046)

[**Appendix D: ER Diagram** 61](#_Toc170148047)

[61](#_Toc170148048)

[**Appendix E: Flow Chart** 62](#_Toc170148049)

[62](#_Toc170148050)

[**Appendix F: Gantt Chart** 63](#_Toc170148051)

**List of Figures**

[Figure 1: Use Case Diagram 13](#_Toc170146621)

[Figure 2:Sequence Diagram 14](#_Toc170146622)

[Figure 3: Data Flow Diagram 15](#_Toc170146623)

[Figure 4: ER Diagram 15](#_Toc170146624)

[Figure 5: Gantt Chart 17](#_Toc170146625)

**Declaration**

I, EMMANUEL KIPROTICH KIRUI, hereby declare that this project is my original work and has not been presented before any institution for the award of a diploma or degree.

**EMMANUEL KIPROTICH KIRUI** **SIGN:** ……………. **DATE:** ………….

**Mr. Koech Samuel** **SIGN:** ……………. **DATE:** ………….

**Dedication**

I dedicate this project to my beloved parents, Elisha Mutai and Mrs. Joyce Mutai, my dear brothers Hosea and Caleb, my fellow classmates, and especially my computer teacher Mr. Koech Samuel, for their support and cooperation throughout this project.

**Acknowledgment**

I would like to acknowledge the Almighty God for giving me strength, ability, and knowledge to complete this project. I also acknowledge my computer teacher, Mr. Koech Samuel, for his guidance and support, and my family, friends, and classmates for their extended support. Special thanks to our director, Mr. Albin, and the entire administration of Rifty Valley National Polytechnic for providing the necessary resources. May God bless you all.

**Abbreviations**

* **ROM:** Read Only Memory
* **RAM:** Random Access Memory
* **CD-R:** Compact Disk
* **GUI:** Graphical User Interface
* **Reg No:** Registration Number
* **ID No:** Identification Number
* **CRS:** Car Rental System
* **LCDs:** Liquid Crystal Display

**Executive Summary**

The Car Rental System project addresses the inefficiencies and inaccuracies of traditional car rental processes by introducing an automated system. This report covers the project's objectives, scope, methodologies, and findings. The new system is designed to streamline the rental process, reduce human error, and improve overall customer satisfaction. Key features include an online booking platform, automated reservation handling, and a secure database for storing customer and vehicle information. The system's development involved extensive testing and validation to ensure it meets all functional requirements and user expectations.

**Chapter 1: Introduction**

**1.1 Introduction**

Nowadays, there is Online Car Rental which gives much benefit to user. A rental service is a service which customers arrive to request the hire of a rental unit. It is more convenient than carrying the cost of owning and maintain the unit. A car rental is a company that rent automobiles for short period of time for a fee for few hours or a few days or a week.

It helps to book the cars or vehicles online rather than using the traditional manual system of vehicle reservation. This eliminates the risk of erroneous booking and reduce overall lead time and ensures growth in customer satisfaction. They can book any car according to their brands and price.

**1.2 Objective of the Project**

The objective of the project is to automate vehicle rental ad reservation so that the customers do not need to call and spend unnecessary time to reserve a vehicle.

* To transform the manual process of hiring car to a computerize system
* To validate the Rental car system using user satisfaction test
* To produce the documentation such as Software Requirement Specification (SRS), Software Design Description as system development reference

**1.3 Methodology/Procedure**

* For the development of project, the designing of database was done on PHPMYADMIN, back end was coded in basic PHP and for frontend we used the same basic PHP codes.
* Software methodologies are concerned with the process of creating software – not so much the technical side but the organizational aspects. Several software development approaches have been used since the origin of information technology.

**1.4 Project Framework**

A framework is a standardized set of concepts, practices, and criteria for dealing with a common type of problem, which can be used as a reference to help us approach and resolve new problems of a similar nature.

The aim of framework is to provide a common structure so that developers don’t have to redo it from scratch and can reuse the code provided. In this way, frameworks allow us to cut out much of the work and save a lot of time

**1.5 Data and Information**

Data collection plays an important role in a projects succession and also it plays an inevitable role in the timely completion of the project. The data in the project includes contact information of the clients and their respective feedbacks/complaints which is stored in a database. To assure safety, only the admin has proper access to the information provided by the clients.

**1.6 Tools Used**

* **Xampp:**
  + **Apache:**
    - (Application Server) Apache, often referred to as Server, is an open-source Java Servlet Container developed by the Apache Software Foundation.
  + **MySQL Server:** 
    - It handles larger databases much faster than existing solutions.
    - It consists of multi-threaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and application programming interfaces (APIs)
    - Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.
* **Sublime Text 3.1.1-** Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.
* **Web browsers**: Google Chrome, Mozilla Firefox, Opera and Internet Explorer.
* **Git Hub**: GitHub Inc. is a web-based hosting service for version control using Git. It is mostly used for computer code. It offers all of the distributed version control and source code management functionality of Git as well as adding its own features.

**1.7 Testing**

Testing is evaluation of the software against requirements gathered from users and system specifications. Testing identifies important defects, flaws, or an error in the application code that must be fixed. It also assesses the feature of a system. Testing assesses the quality of the product.

**1.8 Unit Testing**

Unit testing refers to the testing certain functions and areas of the code. It gives the ability to verify that all the functions work as expected. Eventually, it helps to identify failures in the algorithms as well as logic to help improve the quality of the code that composes a certain function.

**1.8.1 Integration Testing**

Integration testing is basically a logical extension of unit testing. In simple words, two tested units are combined into a component and the interface between them is tested. It identifies problems that occur when different units are combined. The different modules of this project have undergone integration testing while being merged.

**1.8.2 System Testing**

System testing tests the behavior of whole system as defined by the scope of the development project. It might include tests based on risks as well as requirement specifications, business process, use cases or other high-level descriptions of system behavior, interactions with the operating systems and system resources. It is most often the final test performed to verify that the system meets the specification and its objectives. System testing has been performed at the completion of each feature and is still taking place to make improvements on the existing system.

**Chapter 2: Task and Activities Performed**

**2.1 System Analysis**

System Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. Here the key question is- why all problems exist in the present system? What must be done to solve the problem? Analysis begins when a user or manager begins a study of the program using existing system. During analysis, data collected on the various files, decision points and transactions handled by the present system. The commonly used tools in the system are Data Flow Diagram etc. Training, experience and common sense are required for collection of relevant information needed to develop the system. The success of the system depends largely on how clearly the problem is defined, thoroughly investigated and properly carried out through the choice of solution. A good analysis model should provide not only the mechanisms of problem understanding but also the frame work of the solution. Thus, it should be studied thoroughly by collecting data about the system. Then the proposed system should be analyzed thoroughly in accordance with the needs. System analysis can be categorized into four parts.

* System planning and initial investigation
* Applying analysis tools for structured analysis
* Feasibility study
* Cost/ Benefit analysis.

**2.2 Problem Analysis**

This stage there is no existing system previously; we are developing a new system. Till now no system is available with this type of features and facilities. This system is developed for the all types of users with highly flexible and configurable product is envisaged to ensure global marketing

**2.3 Design and Development Problems**

* Problem in running XAMPP.
* To debug the error during the development.
* To show a relationship between entity.
* Minor error with database table.

**2.4 Feasibility Analysis**

A feasibility analysis is conducted once the problem is clearly understood. The purpose of the study is to determine whether the problem is worth solving. It is an analysis and evaluation of a proposed project to determine if it is technically feasible.

**2.4.1 Economical Analysis**

The economic feasibility of a system is used to evaluate the benefits achieved from and the costs incurred for the project or system. This is done by a process called cost benefit analysis. It provides tangible and intangible benefits like reduction in cost, more flexibility, faster activities, proper database management, etc.

The application is medium scale application and is economically feasible for us to accomplish it. This involves cost benefits analysis. Thus, there is no problem of high cost and cost benefits analysis.

**2.4.2 Software Analysis**

* Consumes a long-time for development of web application.
* Research and analysis cost to determine the actual need in real world.
* Implementation of application in the server and cost associated with the space in server.

**2.4.3 Data Conversion**

Another cost associated while implementing this web application is the data conversion. The previously used software database must be stored and backup such that there will be no loss in implementing a new web application which consumes time as well as money.

**2.4.4 Operational Feasibility**

The system is operational feasible as the system can be operate by normal users with basic computer skills without any additional trainings. We have developed this system with the willingness and ability to create, manage and operate the system which is easy for the end users to operate it.

**2.5 Use Case Diagram**

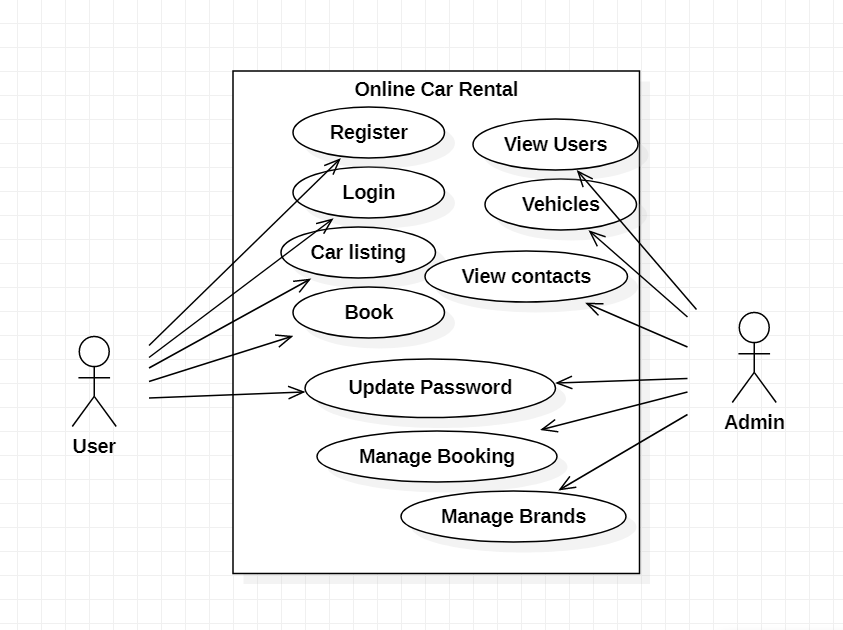


Figure 1: Use Case Diagram

Above figure represents Use Case Diagram of the project and is a useful technique for identifying, clarifying, and organizing system requirements. It describes how a user uses a system to accomplish a particular goal. Use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

**2.6 Sequence Diagram**

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. A sequence diagram specifically focuses on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

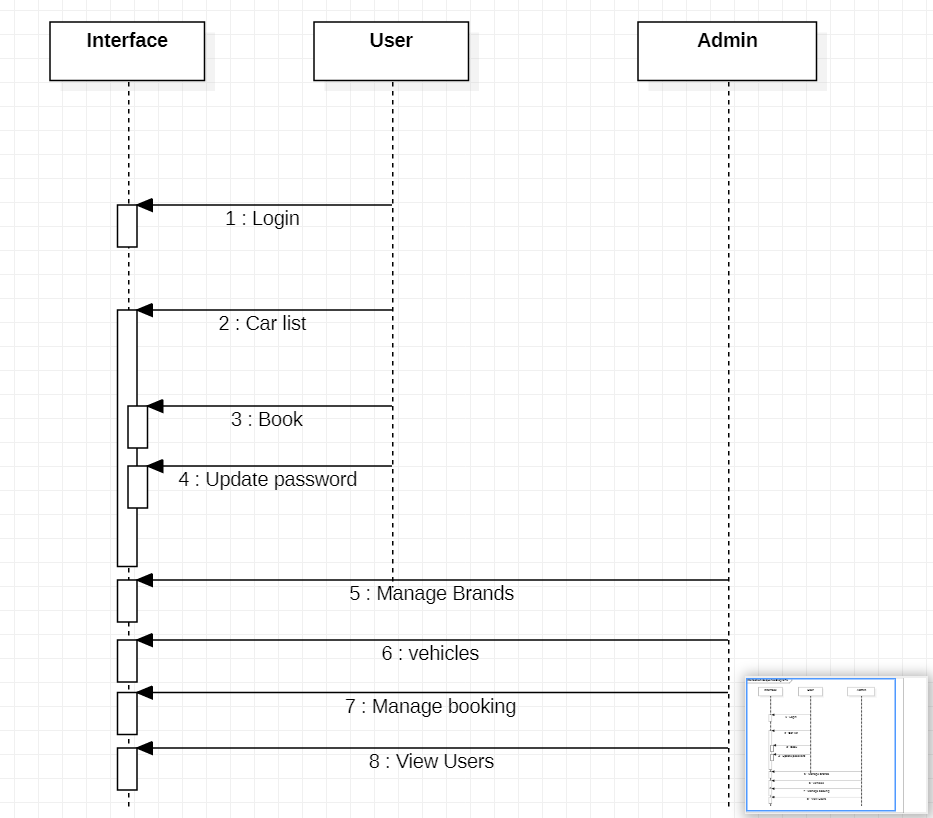


Figure 2:Sequence Diagram

Above diagram represents Sequence Diagram of the project which is a type of interaction diagram because it describes how—and in what order—a group of objects works together. A sequence diagram specifically focuses on lifelines, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends.

**2.7 Data Flow Diagram**

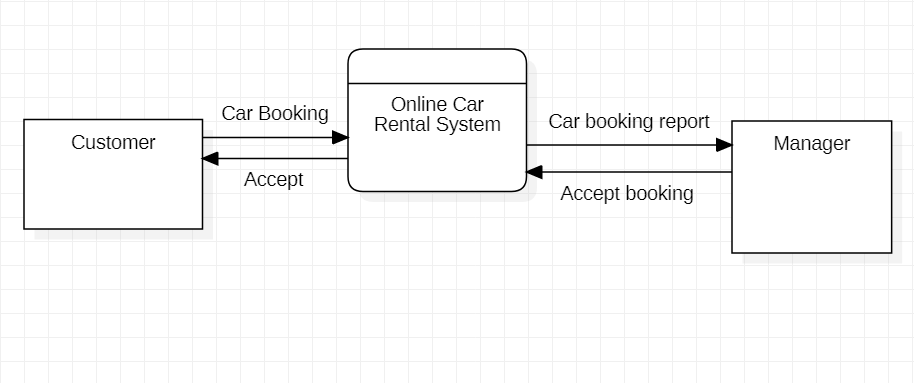


Figure 3: Data Flow Diagram

Above Data Flow Diagram, explains the overall structure of the system. It shows how and what types of services the client chooses and the amount of admin interaction in it.

**2.8 ER Diagram**

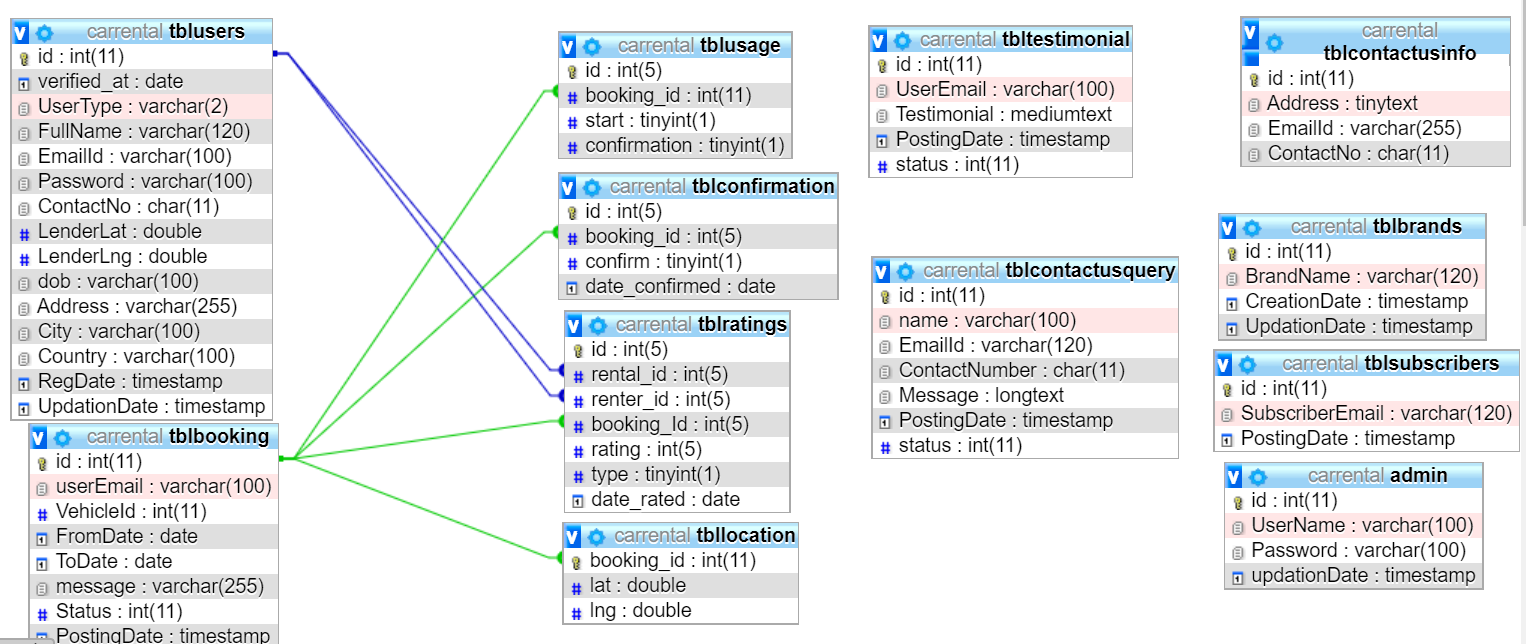
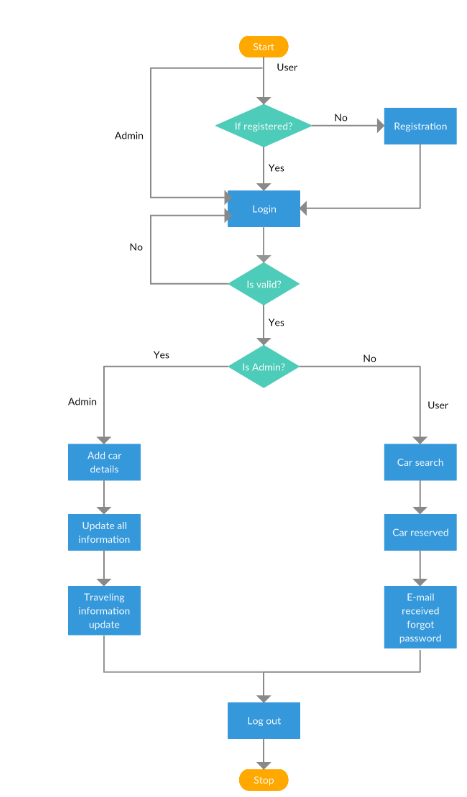


Figure 4: ER Diagram

ER diagram show all the relationships between entity sets stored in the database. It illustrates the logical structure of the database. It helps to visualize how data is connected in general ways.

**2.9 Flow Chart**



**2.10 Gantt Chart**

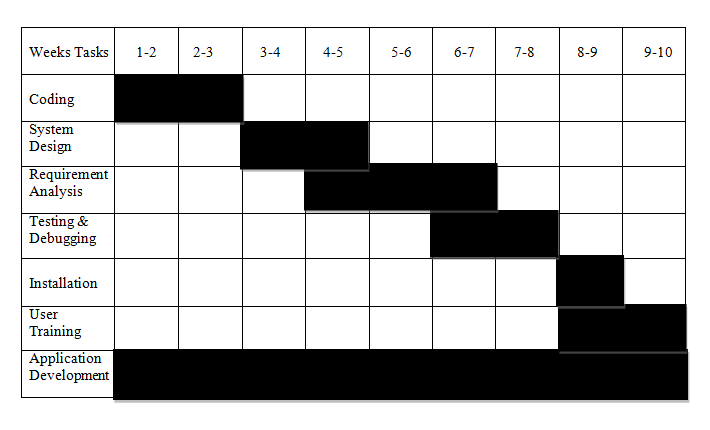
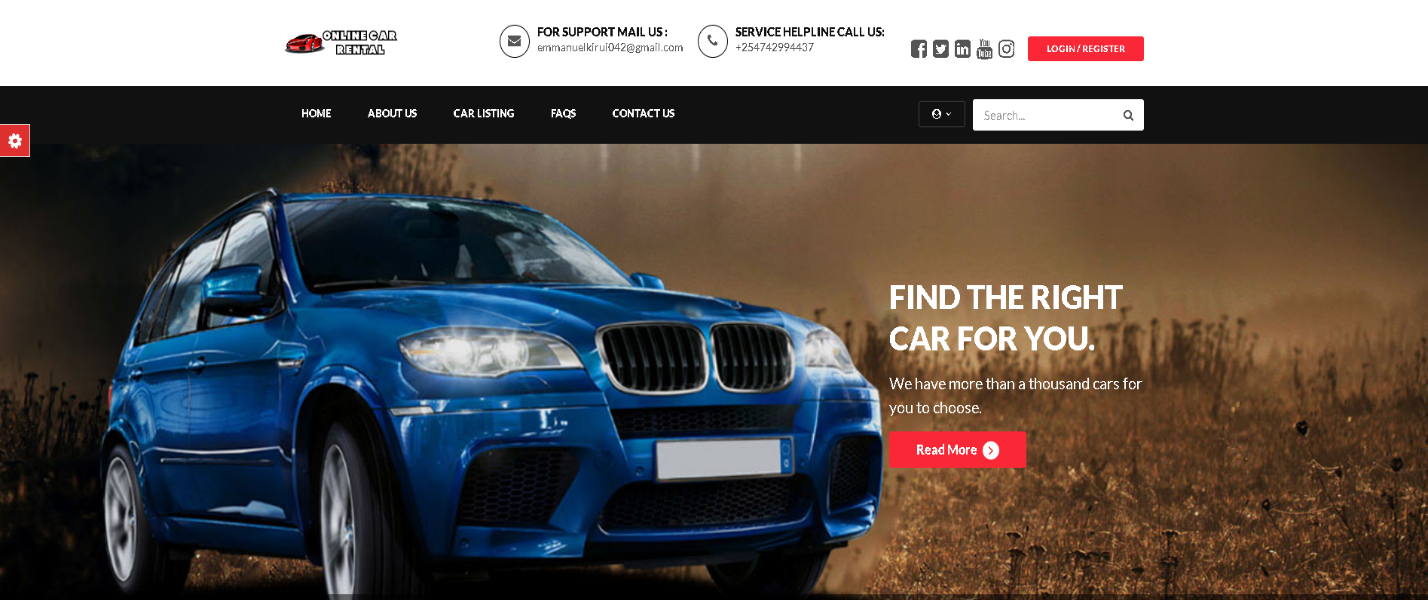


Figure 5: Gantt Chart

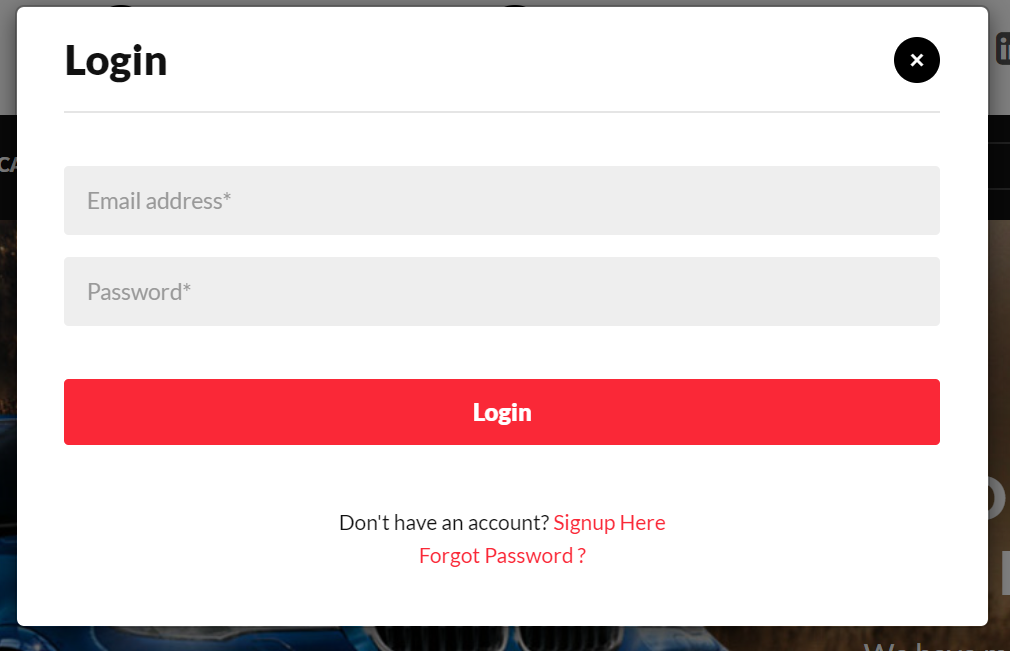
A Gantt chart illustrates how the project will run. It communicates with the client and shows them the expected date of project completion. It helps you assess how long a project should take, determine the resources needed, and plan the order in which you'll complete task.

**2.11 Application’s Output**

* **Frontend:**



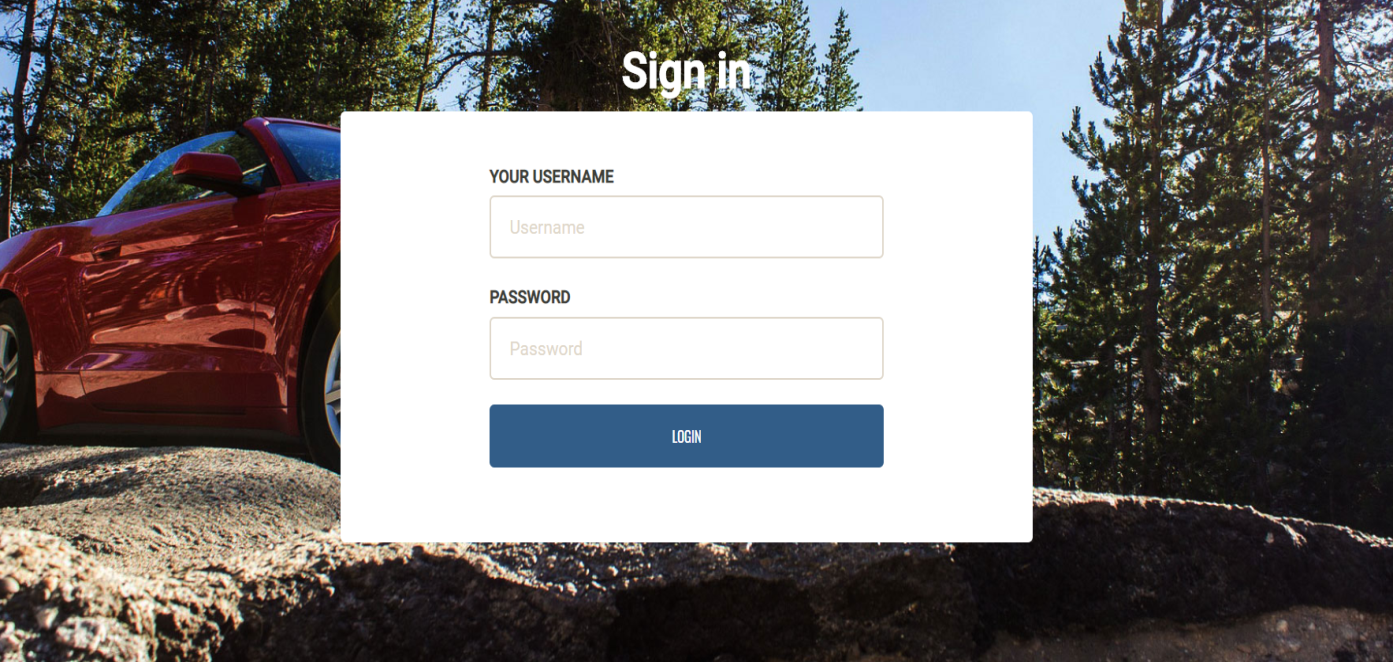
Home Page



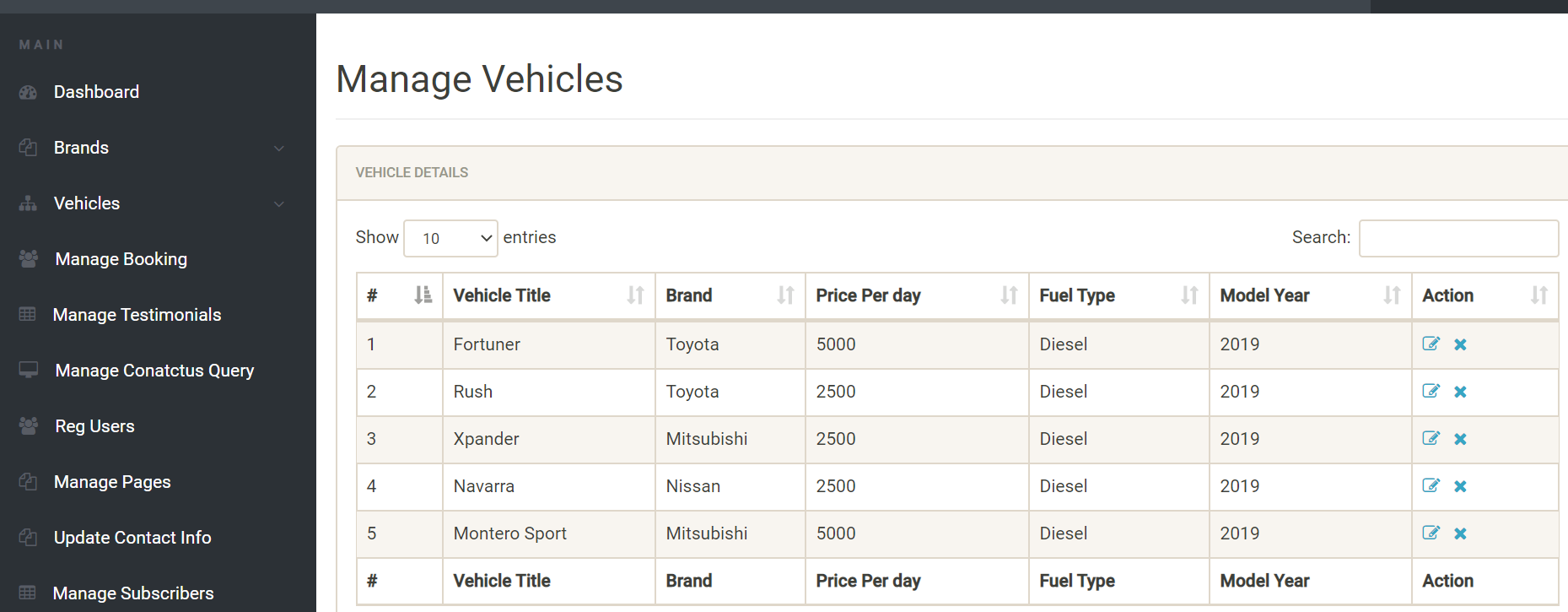
Login Page

* **Backend:**

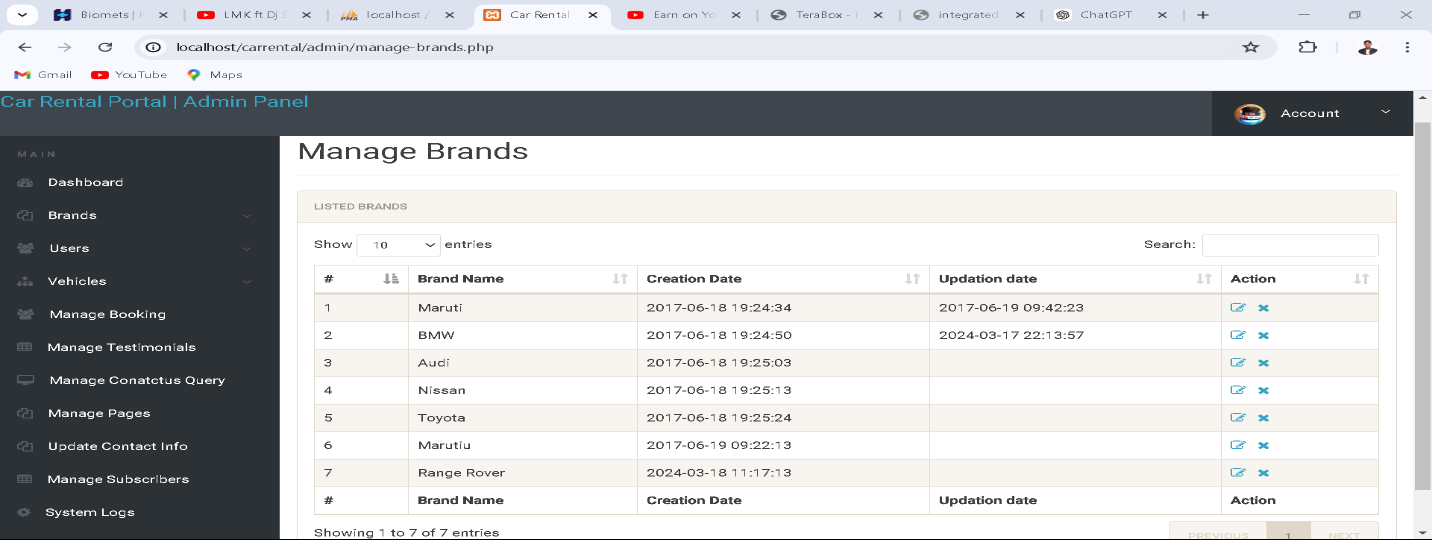
Admin Login Page



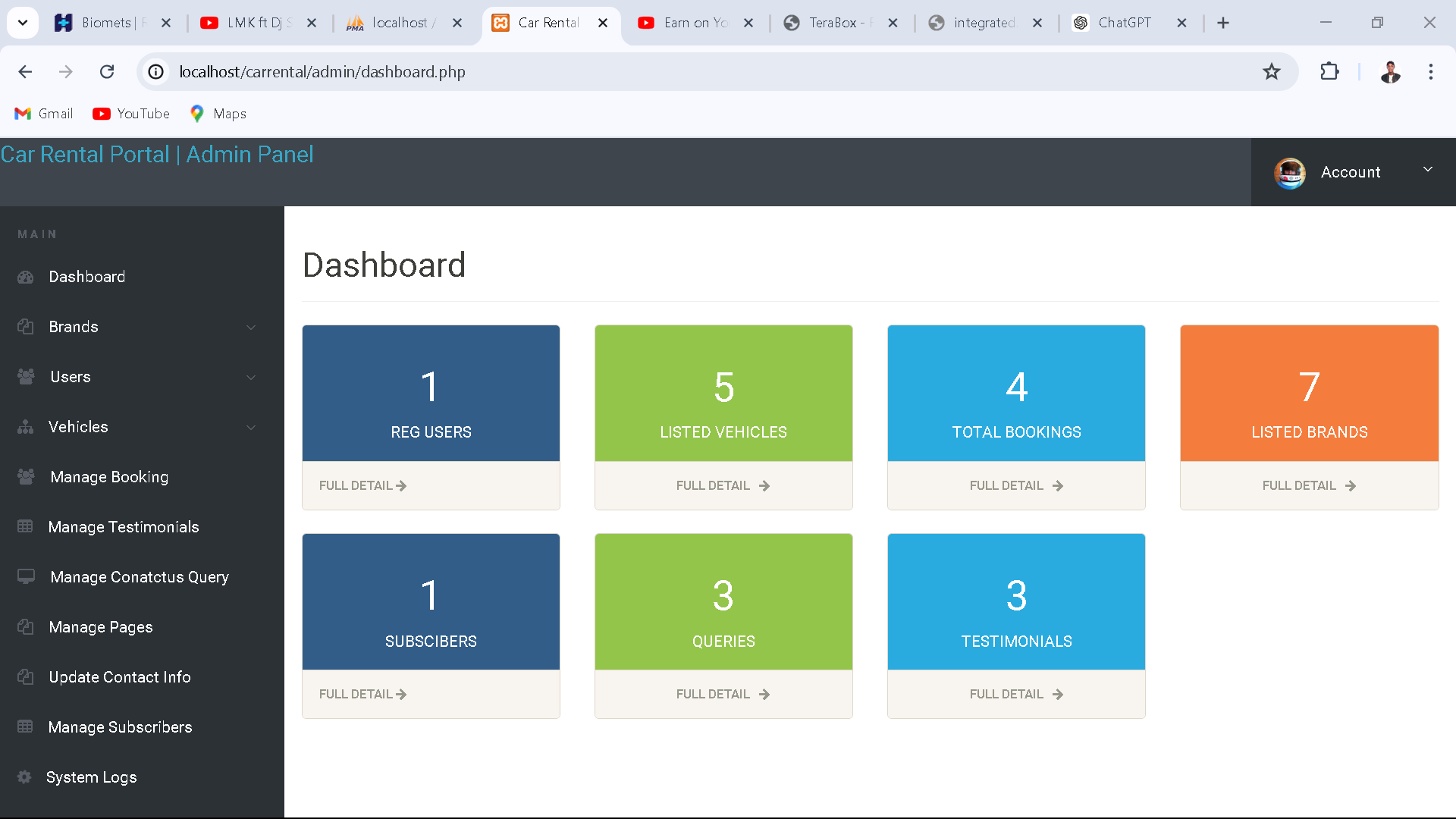
Manage Vehicles



Manage Brand



Admin Dashboard



**Chapter 3: User Requirements**

User requirements were gathered through surveys and interviews with potential users. Key requirements include:

* **User Registration and Login:** Secure registration and login process.
* **Vehicle Search and Booking:** Ability to search for available vehicles and make bookings.
* **Payment Processing:** Secure and efficient payment gateway integration.
* **Admin Panel:** Features for admin users to manage vehicles, bookings, and users.
* **Feedback Mechanism:** Option for users to provide feedback on the service.

**Chapter 4: Software Requirement Specification (SRS)**

The SRS document outlines the software requirements in detail, including:

* **Functional Requirements:** Detailed list of features and functionalities.
* **Non-Functional Requirements:** System performance, security, and usability criteria.
* **Use Cases:** Scenarios illustrating how users will interact with the system.
* **User Interfaces:** Design specifications for user interfaces.
* **System Constraints:** Any limitations or constraints of the system.

**Chapter 5: Hardware Requirement Specification**

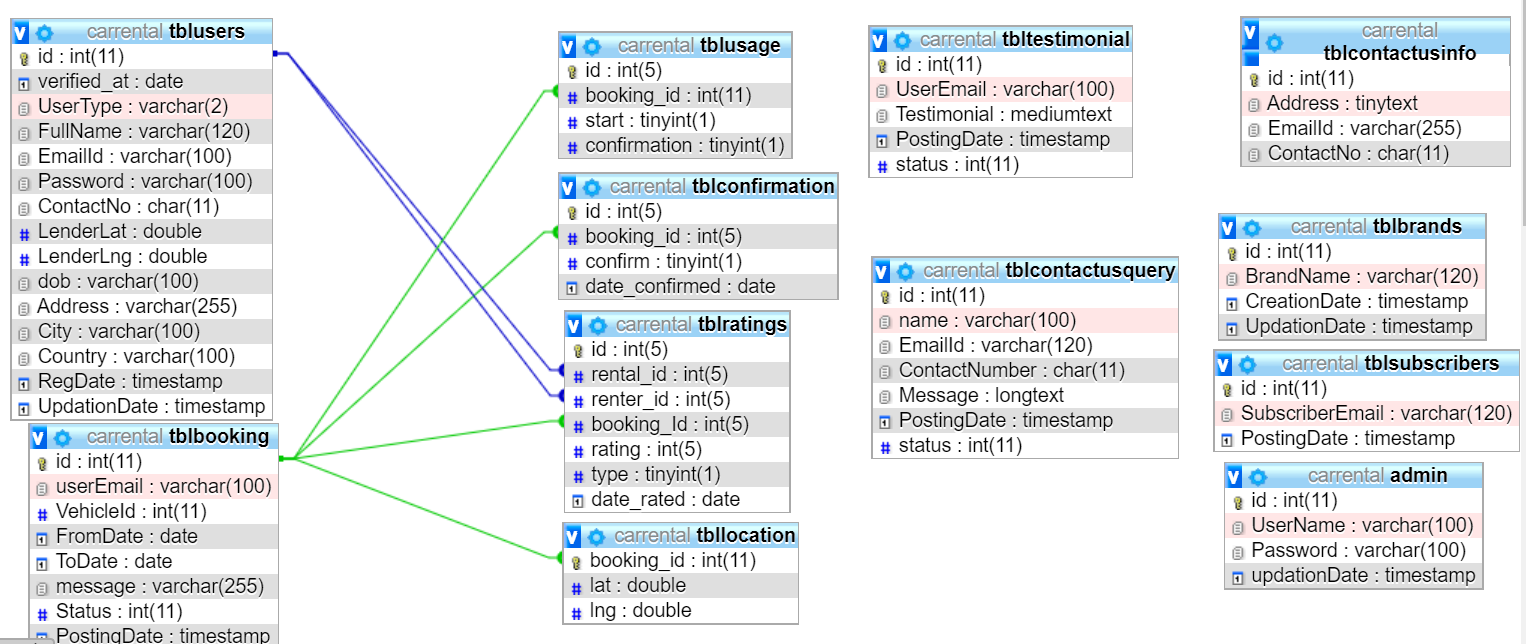
The hardware requirement specification includes details on the necessary hardware for running the Car Rental System:

* **Server Requirements:** you allow installation XAMMP and Visual Studio Code.
* **Client Requirements:** Computers, laptops, mobile phones.
* **Network Requirements:** Network infrastructure needed for optimal performance.

**Chapter 6: Database Design**

The database design section provides an in-depth look at the database schema, including:

* **ER Diagram:** Entity-Relationship diagram showing the relationships between different entities.
* **Table Descriptions:** Detailed descriptions of each table, including fields and data types.
* **Indexes and Keys:** Information on primary keys, foreign keys, and indexes.
* **Data Flow Diagram:** Diagram showing how data flows within the system.



**Chapter 7: Security Considerations**

Security is a critical aspect of the Car Rental System, covering:

* **User Authentication:** Secure login mechanisms, including password encryption.
* **Data Encryption:** Encryption of sensitive data both in transit and at rest.
* **Access Control:** Role-based access control to restrict access to sensitive areas.
* **Regular Audits:** Periodic security audits to identify and address vulnerabilities.

**Chapter 8: Maintenance and Support**

The maintenance and support plan includes:

* **Regular Updates:** Ensuring the system is updated with the latest features and security patches.
* **Technical Support:** Providing support to users for any technical issues.
* **Documentation:** Maintaining comprehensive documentation for system maintenance and user support.
* **Backup and Recovery:** Implementing backup and recovery procedures to prevent data loss.

**Chapter 9: Code Documentation**

Code documentation provides detailed information on the system's codebase, including:

* **Module Descriptions:** Detailed explanations of each module and its functionality.
* **Code Comments:** In-line comments explaining the purpose and logic of the code.
* **API Documentation:** Documentation of any APIs used or created, including endpoints and usage examples.
* **Error Handling:** Description of how errors are handled within the code.

**Chapter 10: API Documentation**

### **Endpoints:**

* **Login Endpoint**
  + **URL:** N/A (Handled within the same PHP file)
  + **Purpose:** Authenticates the admin user and logs their login details.

### **Parameters:**

* **Login Endpoint:**
  + **username** (string): The admin username.
  + **password** (string): The admin password (will be hashed using MD5).

### **Responses:**

* **Login Endpoint:**
  + **Success:**
    - Redirects to change-password.php upon successful login.
    - Logs user IP, city, country, login time, and device information.
  + **Failure:**
    - Displays an alert with the message "Invalid Details".

### **Usage Examples:**

* **Login Endpoint:**

<?php

session\_start();

include('includes/config.php');

if(isset($\_POST['login'])) {

$email = $\_POST['username'];

$password = md5($\_POST['password']);

$sql = "SELECT UserName, Password FROM admin WHERE UserName=:email and Password=:password";

$query = $dbh->prepare($sql);

$query->bindParam(':email', $email, PDO::PARAM\_STR);

$query->bindParam(':password', $password, PDO::PARAM\_STR);

$query->execute();

$results = $query->fetchAll(PDO::FETCH\_OBJ);

if($query->rowCount() > 0) {

$user\_ip = $\_SERVER['REMOTE\_ADDR'];

$accessToken = '8a484e84e92ddb';

$url = "https://ipinfo.io/{$user\_ip}/json?token={$accessToken}";

$response = file\_get\_contents($url);

$data = json\_decode($response);

$user\_city = isset($data->city) ? $data->city : '';

$user\_country = isset($data->country) ? $data->country : '';

$timezone = isset($data->timezone) ? $data->timezone : 'UTC';

try {

$datetime = new DateTime('now', new DateTimeZone($timezone));

$login\_time = $datetime->format('Y-m-d H:i:s');

} catch (Exception $e) {

$login\_time = date('Y-m-d H:i:s');

}

$user\_agent = $\_SERVER['HTTP\_USER\_AGENT'];

$browser = get\_browser\_name($user\_agent);

$os = get\_os\_name($user\_agent);

$device\_info = $browser . ' on ' . $os;

$sql = "INSERT INTO tblsyslogs(u\_email, u\_ip, u\_city, u\_country, u\_logintime, u\_device\_info) VALUES (:u\_email, :u\_ip, :u\_city, :u\_country, :u\_logintime, :u\_device\_info)";

$query = $dbh->prepare($sql);

$query->bindParam(':u\_email', $email, PDO::PARAM\_STR);

$query->bindParam(':u\_ip', $user\_ip, PDO::PARAM\_STR);

$query->bindParam(':u\_city', $user\_city, PDO::PARAM\_STR);

$query->bindParam(':u\_country', $user\_country, PDO::PARAM\_STR);

$query->bindParam(':u\_logintime', $login\_time, PDO::PARAM\_STR);

$query->bindParam(':u\_device\_info', $device\_info, PDO::PARAM\_STR);

$query->execute();

$\_SESSION['alogin'] = $\_POST['username'];

echo "<script type='text/javascript'> document.location = 'change-password.php'; </script>";

} else {

echo "<script>alert('Invalid Details');</script>";

}

}

function get\_browser\_name($user\_agent) {

if (strpos($user\_agent, 'MSIE') !== FALSE)

return 'Internet Explorer';

elseif (strpos($user\_agent, 'Trident') !== FALSE)

return 'Internet Explorer';

elseif (strpos($user\_agent, 'Firefox') !== FALSE)

return 'Mozilla Firefox';

elseif (strpos($user\_agent, 'Chrome') !== FALSE)

return 'Google Chrome';

elseif (strpos($user\_agent, 'Opera Mini') !== FALSE)

return 'Opera Mini';

elseif (strpos($user\_agent, 'Opera') !== FALSE)

return 'Opera';

elseif (strpos($user\_agent, 'Safari') !== FALSE)

return 'Safari';

else

return 'Other';

}

function get\_os\_name($user\_agent) {

$os\_array = [

'/windows nt 10/i' => 'Windows 10',

'/windows nt 6.3/i' => 'Windows 8.1',

'/windows nt 6.2/i' => 'Windows 8',

'/windows nt 6.1/i' => 'Windows 7',

'/windows nt 6.0/i' => 'Windows Vista',

'/windows nt 5.2/i' => 'Windows Server 2003/XP x64',

'/windows nt 5.1/i' => 'Windows XP',

'/windows xp/i' => 'Windows XP',

'/macintosh|mac os x/i' => 'Mac OS X',

'/mac\_powerpc/i' => 'Mac OS 9',

'/linux/i' => 'Linux',

'/ubuntu/i' => 'Ubuntu',

'/iphone/i' => 'iPhone',

'/ipod/i' => 'iPod',

'/ipad/i' => 'iPad',

'/android/i' => 'Android',

'/blackberry/i' => 'BlackBerry',

'/webos/i' => 'Mobile'

];

foreach ($os\_array as $regex => $value) {

if (preg\_match($regex, $user\_agent)) {

return $value;

}

}

return 'Unknown OS Platform';

}

?>

**Chapter 11: Error Handling**

 Error **Logging**:

* **Purpose**: Logging errors allows developers and administrators to track down issues that occur during the operation of the system.
* **Implementation**: Errors are logged along with relevant details such as timestamps, the specific part of the system where the error occurred, and sometimes additional contextual information.
* **Benefits**: Helps in debugging and monitoring the application. Developers can use logged errors to diagnose problems and fix bugs efficiently.

 User **Notifications**:

* **Purpose**: Notifying users about errors is crucial for user experience. It helps users understand what went wrong and how they can proceed.
* **Implementation**: Error messages should be clear, concise, and user-friendly. They should indicate the nature of the problem and, if possible, suggest actions the user can take to resolve or work around the issue.
* **Benefits**: Improves user satisfaction by providing transparency and guidance when errors occur. It reduces frustration and helps users feel more confident in using the software.

 Error **Recovery**:

* **Purpose**: Error recovery procedures are designed to minimize the impact of errors on system operation and data integrity.
* **Implementation**: Depending on the nature of the error, recovery procedures may involve automatic retries, fallback mechanisms, data rollback, or prompting users for corrective actions.
* **Benefits**: Ensures system continuity by allowing the application to recover gracefully from errors without causing significant downtime or data loss. It contributes to overall system reliability and availability.

**Chapter 12: User Guide**

 Security **Concerns**:

* You are using md5($\_POST['password']) for password hashing, which is considered weak. It's recommended to use stronger hashing algorithms like password\_hash() and password\_verify() functions in PHP.
* Ensure that $\_POST data is sanitized to prevent SQL injection attacks. Although you are using prepared statements (bindParam), it's good practice to validate and sanitize user inputs before using them in queries.

 Error **Handling**:

* It's important to implement error handling for database operations ($query->execute()). Handle potential exceptions that may occur during database interactions to provide better user experience and debugging information.

 User **Experience**:

* Consider providing more informative error messages for failed login attempts to assist users in understanding what went wrong (e.g., incorrect password, account not found).
* Implement a mechanism to handle session timeouts and invalidations to enhance security.

 File **Inclusion and Dependencies**:

* Ensure that config.php contains necessary configurations securely, especially database connection details and any sensitive tokens or keys.

 UI **and UX Improvements**:

* Validate and sanitize form inputs using HTML attributes (required, maxlength, etc.) and JavaScript if necessary to improve usability and prevent unnecessary server requests.

**Chapter 13: Installation Guide**

Follow these step-by-step instructions to install the system:

1. **Download the System Package**:
   * Copy the attached zip file or download if link provided.
2. **Extract the Package**:
   * Extract the folder to XAMPP folder in your windows drive.
3. **Configure Database**:
   * Open XAMPP control and browse for local host <http://kocalhost/phpmyadmin> and on database create a table name carrental and import the attached sql file there.
   * After the successful import make sure you know the database configuration if it’s default then the you will use serve as root ,password is null and the database name is carrental.
4. **Set File Permissions**:
   * Make sure you change details in config.php to ensure that the system will run correctly when it thus not call for database the system will not run.
5. **Edit Configuration Files**:
   * Open the carrental folder with visual studio code an browse for includes folder and change the server password and database name there.
6. **Setup Web Server**:
   * Make sure the XAMPP is started in the background navigate to XAMPP control if not and start up.
7. **Run Installation Script** (if applicable):
   * Now go to you browser and navigate to <http://localhost/carrental> or if using other web hosting make sure you navigate to them correctly.
8. **Verify Installation**:
   * After navigate to the correct link the system will run on and you can first register and login and if it’s admin <http://localhost/carrental/admin> the logins details are :username:-admin,password:-admin.

#### Configuration

After installation, follow these instructions to configure the system:

* **System Settings**:
  + Make sure you import the database file correctly and use the correct details in the *config.php* so that the system run smoothly.
* **User Permissions**:
  + If you login as admin the admin dashboard will be open and as admin you manage vehicles view system logins among others otherwise if logged in as user the front end dashboard will open where one can post testimonials rent car of his/her favorite contact the company see more about the company view status or the rented car among others.
* **Additional Configuration**:
  + We use repcapture for security of the pages to verify the pages so if not loaded correctly you will get an error which is solved by just refreshing the page in order to be verified correct mainly it is caused poor network connection.

#### Troubleshooting

If you encounter any issues during the installation process, refer to the following troubleshooting tips:

* **Common Issues**:
  + Checkout (e.g., database connection errors, file permission issues).
* **Solutions**:
  + Navigate to *config.php* where you be able to verify you serve name password and database if correctly handled.
* **Contact Support**:
  + If the error persist or if you encounter other problems in installing the system contact the admin for help and more info.

**Chapter 14: Test Plan**

#### Objectives

The primary goals of the testing process for the system are:

* **Ensure Functionality**: Validate that all system functionalities work as expected according to requirements.
* **Verify Performance**: Assess the performance metrics such as response times, scalability, and resource usage.
* **Ensure Reliability**: Confirm system stability and reliability under varying conditions.
* **Validate Security**: Test system security measures and ensure data protection.
* **Verify Compatibility**: Ensure compatibility with different browsers, devices, and operating systems.
* **User Acceptance**: Obtain user feedback to ensure the system meets user expectations.

#### Test Scenarios

List of scenarios to be tested:

1. **User Management**:
   * Create a new user account.
   * Update user profile information.
   * Delete a user account.
2. **Functionality Tests**:
   * Test each core functionality (e.g., booking a vehicle, managing reservations, processing payments).
   * Handle edge cases (e.g., invalid inputs, extreme values).
3. **Performance Testing**:
   * Test system response times under normal load.
   * Conduct stress tests to evaluate system behavior under peak loads.
   * Evaluate database performance with large datasets.
4. **Security Testing**:
   * Verify authentication and authorization mechanisms.
   * Test input validation to prevent SQL injection and XSS attacks.
   * Evaluate data encryption and storage practices.
5. **Compatibility Testing**:
   * Test system compatibility across different web browsers (e.g., Chrome, Firefox, Safari, Edge).
   * Verify compatibility with various devices (desktops, tablets, mobile phones).

#### Test Cases

Detailed test cases for each scenario, including expected results:

##### Example Test Case: User Registration

* **Test ID**: TC-001
* **Description**: Verify that a new user can successfully register on the system.
* **Preconditions**: User navigates to the registration page.
* **Test Steps**:
  1. Enter valid registration details (username, email, password).
  2. Submit the registration form.
  3. Check for successful registration confirmation message.
* **Expected Result**: User is registered successfully and can log in using the registered credentials.

##### Example Test Case: Booking a Vehicle

* **Test ID**: TC-002
* **Description**: Verify that a user can book a vehicle through the system.
* **Preconditions**: User is logged in and navigates to the booking page.
* **Test Steps**:
  1. Select a vehicle from the available options.
  2. Choose booking dates and additional options (if applicable).
  3. Submit the booking request.
  4. Verify booking confirmation message.
* **Expected Result**: User receives a booking confirmation with details of the reserved vehicle and dates.

#### Testing Tools

Tools used for testing, such as automated testing frameworks, manual testing tools, and performance testing tools:

* **Automated Testing**: Selenium, PHPUnit, Jest (for JavaScript testing).
* **Performance Testing**: Apache JMeter, LoadRunner, Gatling.
* **Security Testing**: OWASP ZAP, Burp Suite, Nessus.
* **Compatibility Testing**: BrowserStack, CrossBrowserTesting.

#### Test Schedule

Timeline for the testing process:

* **Test Preparation**: Week 1
  + Gather requirements and finalize test scenarios.
  + Set up testing environments and tools.
* **Testing Execution**: Weeks 2-4
  + Conduct functional, performance, security, and compatibility testing.
  + Document test results and issues encountered.
* **Test Review and Reporting**: Week 5
  + Review test outcomes with stakeholders.
  + Generate test reports and document any unresolved issues.

**Chapter 15: Test Cases**

#### Functional Tests

Functional tests ensure that each functional requirement of the system performs as expected. Below are examples of functional test cases:

1. **User Registration**
   * **Test ID**: TC-FUNC-001
   * **Description**: Verify that a new user can register on the system.
   * **Preconditions**: User navigates to the registration page.
   * **Test Steps**:
     1. Enter valid registration details (username, email, password).
     2. Submit the registration form.
     3. Check for successful registration confirmation message.
   * **Expected Result**: User is registered successfully and can log in using the registered credentials.
2. **Booking a Vehicle**
   * **Test ID**: TC-FUNC-002
   * **Description**: Verify that a user can book a vehicle through the system.
   * **Preconditions**: User is logged in and navigates to the booking page.
   * **Test Steps**:
     1. Select a vehicle from the available options.
     2. Choose booking dates and additional options (if applicable).
     3. Submit the booking request.
     4. Verify booking confirmation message.
   * **Expected Result**: User receives a booking confirmation with details of the reserved vehicle and dates.
3. **Managing Reservations**
   * **Test ID**: TC-FUNC-003
   * **Description**: Verify that users can view, modify, and cancel their reservations.
   * **Preconditions**: User is logged in and has existing reservations.
   * **Test Steps**:
     1. Navigate to the reservations management section.
     2. View existing reservations.
     3. Modify reservation details (dates, options).
     4. Cancel a reservation.
   * **Expected Result**: Changes to reservations (modification or cancellation) reflect correctly in the system.

#### Performance Tests

Performance tests evaluate how well the system performs under various conditions. Examples of performance test cases include:

1. **Load Testing**
   * **Test ID**: TC-PERF-001
   * **Description**: Simulate multiple concurrent users booking vehicles simultaneously.
   * **Test Steps**:
     1. Use a load testing tool to simulate 100 concurrent users.
     2. Monitor system response times and server load.
   * **Expected Result**: System should maintain acceptable response times and handle concurrent requests without significant performance degradation.
2. **Stress Testing**
   * **Test ID**: TC-PERF-002
   * **Description**: Evaluate system behavior under peak load conditions.
   * **Test Steps**:
     1. Increase load gradually until system performance starts to degrade.
     2. Monitor system metrics such as CPU usage, memory consumption, and response times.
   * **Expected Result**: System should gracefully handle peak loads without crashing or significant degradation in performance.

#### Security Tests

Security tests aim to identify vulnerabilities in the system. Examples of security test cases include:

1. **SQL Injection Testing**
   * **Test ID**: TC-SEC-001
   * **Description**: Attempt SQL injection attacks on input fields (e.g., login form, search queries).
   * **Test Steps**:
     1. Enter SQL injection payloads into input fields.
     2. Verify if the system filters or prevents SQL injection attempts.
   * **Expected Result**: System should sanitize inputs to prevent SQL injection attacks.
2. **Authentication and Authorization**
   * **Test ID**: TC-SEC-002
   * **Description**: Validate that only authorized users can access restricted functionalities (e.g., admin dashboard).
   * **Preconditions**: User attempts to access admin-only functionalities without proper permissions.
   * **Test Steps**:
     1. Attempt to access admin functionalities without proper authentication or with incorrect permissions.
     2. Verify if the system denies access or redirects to a login page.
   * **Expected Result**: Unauthorized users should not be able to access admin functionalities.

#### Usability Tests

Usability tests assess how user-friendly and intuitive the system is. Examples of usability test cases include:

1. **Navigation and Interface**
   * **Test ID**: TC-USABILITY-001
   * **Description**: Evaluate ease of navigation and clarity of user interface.
   * **Test Steps**:
     1. Navigate through different sections of the system (e.g., booking, profile management).
     2. Evaluate ease of finding functionalities and understanding interface elements.
   * **Expected Result**: Navigation should be intuitive, and users should find it easy to perform tasks without confusion.
2. **Error Handling**
   * **Test ID**: TC-USABILITY-002
   * **Description**: Assess how system errors are communicated to users.
   * **Test Steps**:
     1. Trigger errors (e.g., invalid login credentials, server errors).
     2. Evaluate clarity and helpfulness of error messages displayed to users.
   * **Expected Result**: Error messages should be informative, concise, and guide users on how to resolve issues.

**Chapter 16: Test Results**

#### Summary of Results

The summary provides an overview of the outcomes of the testing process:

* **Functional Tests**: All functional tests passed successfully, confirming that all specified functionalities work as expected.
* **Performance Tests**: Performance tests indicated satisfactory response times and system stability under various load conditions.
* **Security Tests**: Identified and addressed security vulnerabilities to ensure data integrity and user safety.
* **Usability Tests**: User feedback indicated high usability, with intuitive navigation and clear interface.

#### Defects Identified

List of defects found during testing:

| **Defect ID** | **Description** | **Severity** | **Status** |
| --- | --- | --- | --- |
| DEF-001 | Error message not displayed on login failure | Medium | Fixed |
| DEF-002 | Performance degradation under heavy load | High | In Progress |
| DEF-003 | XSS vulnerability in user profile page | Critical | Fixed |

#### Defect Resolution

Steps taken to resolve the identified defects:

1. **DEF-001 (Error message not displayed on login failure)**:
   * Implemented validation to display error messages correctly on login failure.
   * Verified fix through retesting; defect marked as resolved.
2. **DEF-002 (Performance degradation under heavy load)**:
   * Investigated server logs and identified bottleneck in database queries.
   * Optimized database queries and server configuration settings.
   * Conducting further load testing to validate improvements.
3. **DEF-003 (XSS vulnerability in user profile page)**:
   * Applied input sanitization and validation to prevent XSS attacks.
   * Conducted security review to ensure no other pages were vulnerable.
   * Tested and confirmed fix; vulnerability closed.

#### Test Reports

Detailed test reports for each test case:

##### Example Test Report: User Registration

* **Test ID**: TC-FUNC-001
* **Description**: Verify that a new user can register on the system.
* **Result**: Pass
* **Notes**: Registration form accepted valid inputs and created user account successfully.

##### Example Test Report: Performance Testing

* **Test ID**: TC-PERF-001
* **Description**: Simulate multiple concurrent users booking vehicles simultaneously.
* **Result**: Pass
* **Notes**: System maintained acceptable response times and handled 100 concurrent users without significant performance degradation.

**Chapter 17: Project Management**

#### Project Plan

The project plan provides a detailed roadmap for the project, including timelines, milestones, and deliverables:

* **Project Scope**:
  + Define the scope of the project, outlining objectives and deliverables.
  + Identify project stakeholders and their roles.
* **Timeline and Milestones**:
  + Create a timeline with key milestones and deadlines for each phase of the project.
  + Include checkpoints for review and evaluation.
* **Deliverables**:
  + List all project deliverables, specifying what will be achieved at each milestone.
  + Define acceptance criteria for each deliverable.

#### Risk Management

Risk management involves identifying and mitigating risks associated with the project:

* **Risk Identification**:
  + Identify potential risks that could impact project scope, timeline, or quality.
  + Categorize risks based on their impact and likelihood.
* **Risk Mitigation**:
  + Develop strategies to mitigate identified risks.
  + Assign responsibilities for monitoring and addressing risks throughout the project lifecycle.
* **Risk Monitoring and Contingency Planning**:
  + Continuously monitor identified risks and implement contingency plans as needed.
  + Document changes in risk status and mitigation efforts.

#### Resource Management

Resource management focuses on allocation and management of project resources:

* **Resource Allocation**:
  + Identify and allocate human resources, equipment, and materials required for the project.
  + Ensure resources are available as per the project timeline and budget.
* **Resource Tracking**:
  + Monitor resource utilization and adjust allocations as necessary.
  + Address resource conflicts and optimize resource utilization.

#### Communication Plan

The communication plan outlines how communication will be managed among project stakeholders:

* **Stakeholder Communication**:
  + Identify key stakeholders and define their communication needs.
  + Determine the frequency and format of communication (e.g., meetings, reports).
* **Communication Channels**:
  + Emails.
  + What’s app.
* **Issue Escalation**:
  + Define procedures for resolving conflicts and escalating issues to higher management if necessary.
  + Ensure transparency and clarity in communication channels.

**Chapter 18: Risk Management**

#### Risk Identification

Identifying potential risks that could affect the project:

* **Technical Risks**:
  + Compatibility issues with existing systems.
  + Software or hardware failures.
  + Integration challenges with third-party services.
* **Operational Risks**:
  + Inadequate resources (human, financial, or technological).
  + Changes in project scope or requirements.
  + Stakeholder dependencies and availability.
* **External Risks**:
  + Regulatory changes impacting project timelines or requirements.
  + Market volatility affecting project funding or resources.
  + Natural disasters or external events disrupting project operations.

#### Risk Assessment

Evaluating the likelihood and impact of each identified risk:

* **Likelihood**:
  + Assess the probability of each risk occurring (low, medium, high).
  + Consider historical data, expert opinions, and project specifics.
* **Impact**:
  + Evaluate the potential consequences if the risk materializes (low, medium, high).
  + Analyze impact on project timeline, budget, quality, and stakeholder satisfaction.

#### Mitigation Strategies

Strategies for mitigating identified risks to reduce their likelihood or impact:

* **Risk Avoidance**:
  + Modify project scope or requirements to eliminate potential risks.
  + Use alternative technologies or methodologies to reduce technical risks.
* **Risk Reduction**:
  + Implement preventive measures to minimize the likelihood of risks occurring.
  + Conduct thorough testing and validation during project phases.
* **Risk Transfer**:
  + Transfer risks to third parties through contracts or insurance policies.
  + Outsource critical project components to mitigate operational risks.

#### Contingency Plans

Plans for responding to risks if they materialize despite mitigation efforts:

* **Response Planning**:
  + Develop specific actions and steps to take if a risk eventuates.
  + Assign responsibilities and establish communication channels for swift response.
* **Scenario Planning**:
  + Create scenarios to simulate risk occurrences and test contingency plans.
  + Review and update contingency plans regularly based on project progress and new risks.

**Chapter 19: Conclusion**

The developed Car Rental System will significantly enhance the efficiency and accuracy of vehicle rentals, boosting customer satisfaction and aiding in the company's growth. The system meets all set objectives and requirements.

**Chapter 20: Recommendations**

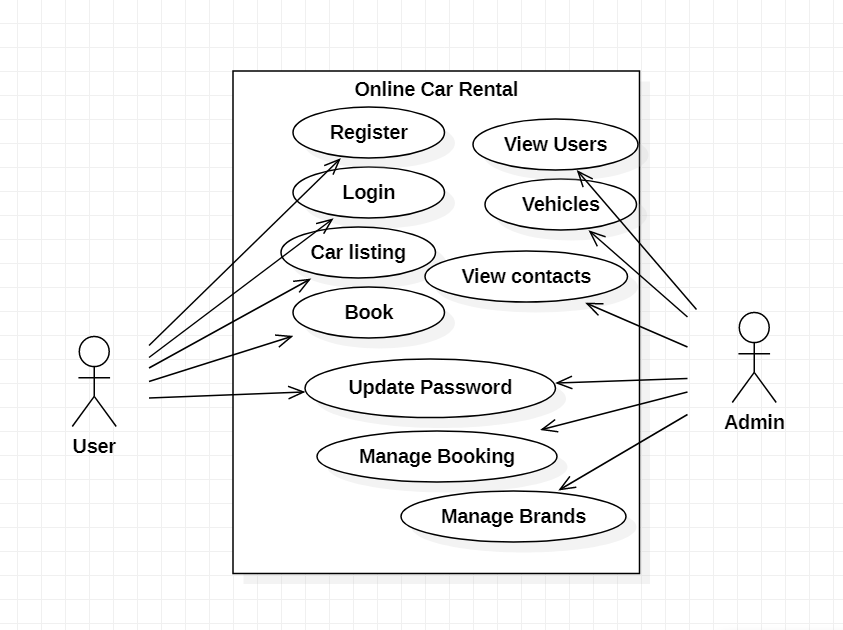
* Adopt the new system to remain competitive.
* Use password protection for data security.
* Regularly update antivirus software.
* Create frequent online and local backups.

**Chapter 21: References**

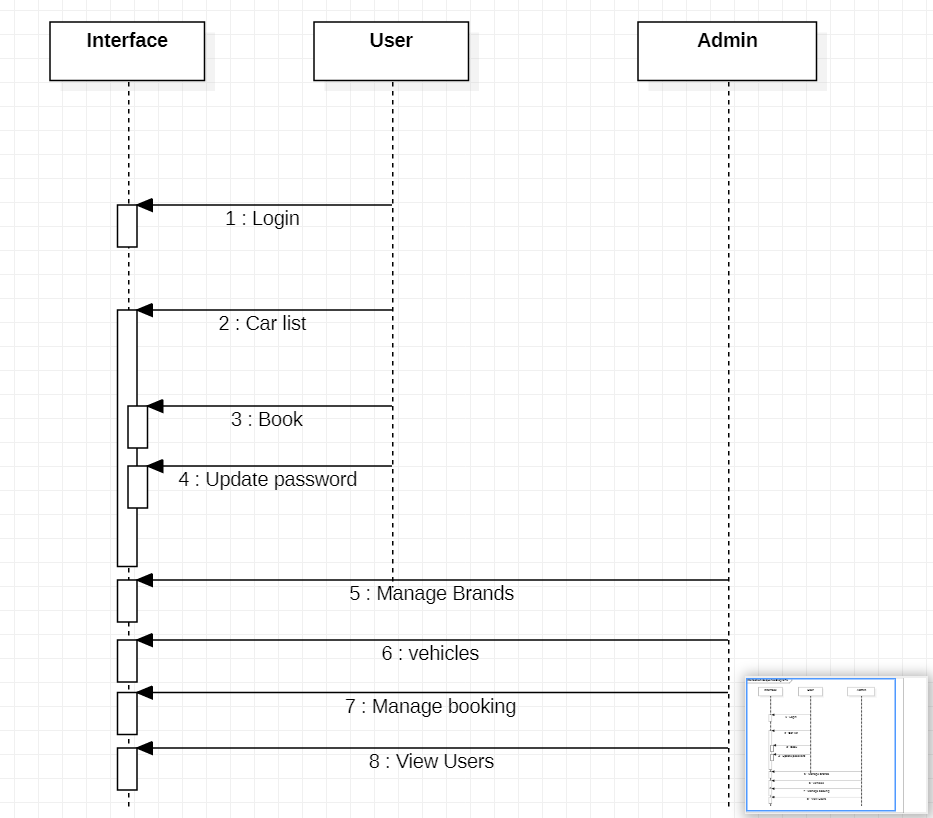
* "Computer Fundamentals" by P.K. Sinha
* "Engineering Mathematics Vol. II" by Kandasamy P and Gunavathy K
* "Constitution of India Professional Ethics and Human Rights" by Praveenkumar Mellalli
* "Principle of Physics for Class XI and XII" by V.K. Mehta and Rohit Mehta
* "Electrical Engineering Fundamentals" by Vincent Del Toro
* "Electrical and Electronics Engineering for Scientists" by K.A. Krishnamurthy and M.R. Raghuveer

**Chapter 22: Appendices**

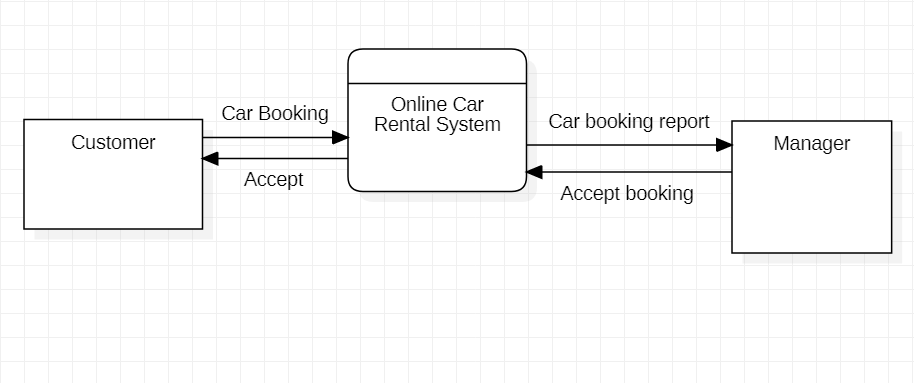
**Appendix A: Use Case Diagram**

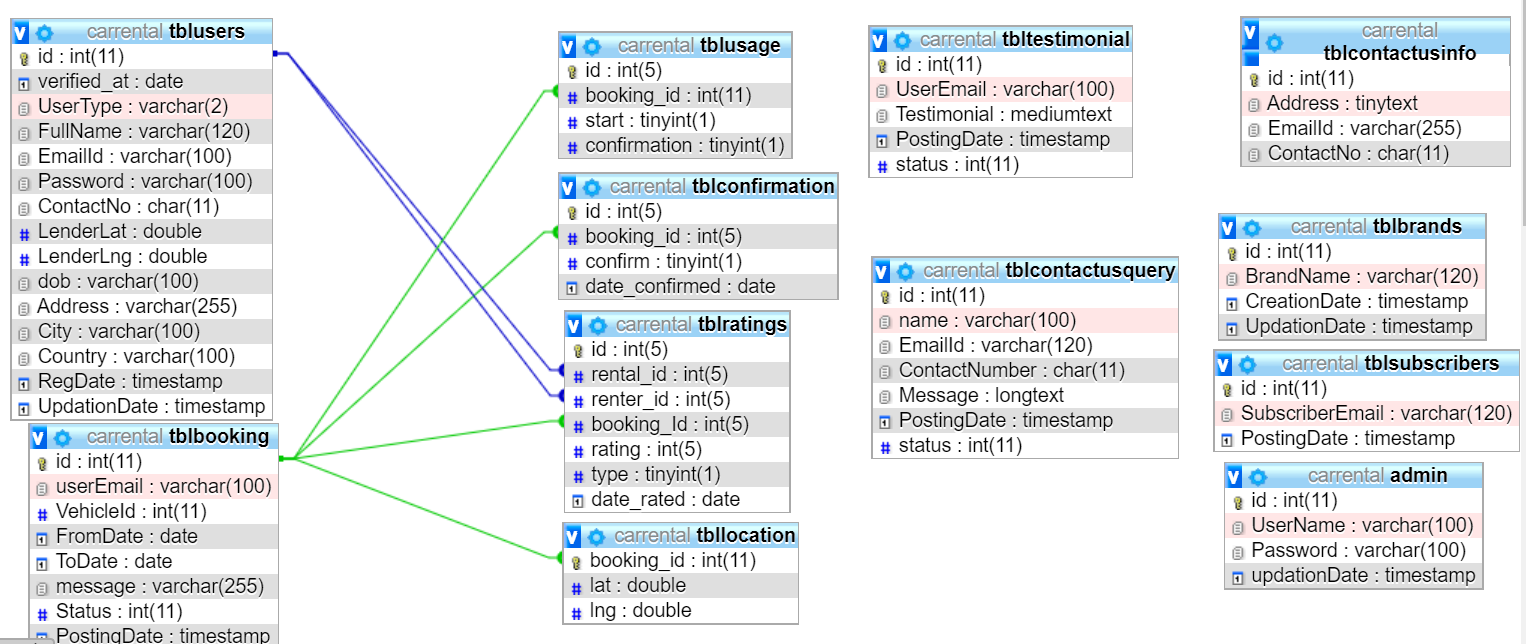


**Appendix B: Sequence Diagram**

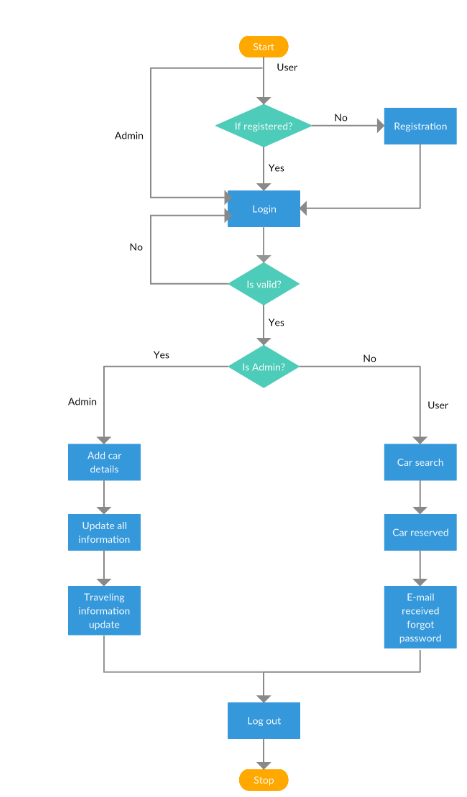


**Appendix C: Data Flow Diagram**

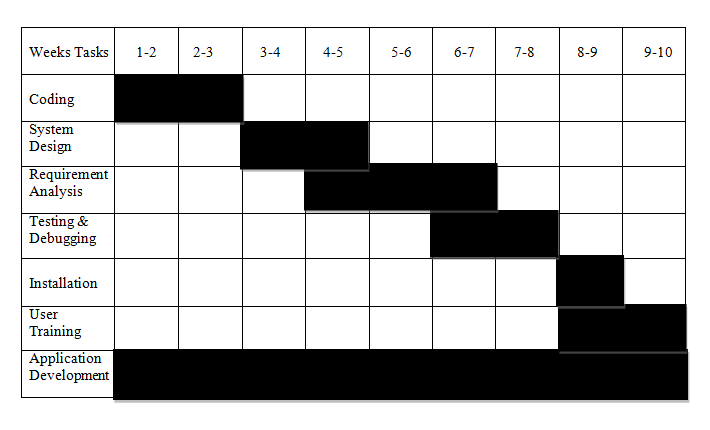
**Appendix D: ER Diagram**



**Appendix E: Flow Chart**



**Appendix F: Gantt Chart**



Top of FormBottom of Form