

|  |
| --- |
| **Project Document** |
| **Car Rental Service Website** |

|  |  |
| --- | --- |
| **Group 1** | |
| **Group Members** | **Hoang Anh - C2010G0141**  **Vu Duc Anh - C2108L0272**  **Nguyen Dinh Duy  - C2105L0183**  **Pham Dang Duc - C2108L0255**  **Nguyen Trung Duc - C2108L0265** |
| **Supervisor** | **Dao Quang Linh** |
| **Project code** | **CRSW** |

**- Hanoi, 12/2023 -**

**Table of Contents**

[Acknowledgement 4](#_heading=h.30j0zll)

[Definition and Acronyms 4](#_heading=h.3znysh7)

[**1**](#_heading=h.3znysh7) **Project Introduction 5**

[1.1](#_heading=h.2et92p0) Project Information 5

[1.2](#_heading=h.tyjcwt) Project Team 5

[1.3](#_heading=h.3dy6vkm) Product Background 5

[1.4](#_heading=h.1t3h5sf) Existing Systems 5

[1.5](#_heading=h.4d34og8) The proposed system 5

[1.6](#_heading=h.2s8eyo1) Project Scope & Limitations 5

[**2**](#_heading=h.17dp8vu) **Project Management Plan 6**

[2.1](#_heading=h.3rdcrjn) The purpose and target of a project 6

[2.2](#_heading=h.35nkun2) Management Approach 6

[2.3](#_heading=h.2jxsxqh) Responsibility Assignments 7

[2.4](#_heading=h.z337ya) Project Communications 7

[2.5](#_heading=h.3j2qqm3) Project Work Plan 7

[2.6](#_heading=h.4i7ojhp) Configuration Management 7

[**3**](#_heading=h.2bn6wsx) **Software Requirement Specification 8**

[3.1](#_heading=h.qsh70q) User Requirements 8

[3.2](#_heading=h.3as4poj) Functional Requirements 9

[3.3](#_heading=h.49x2ik5) Non-Functional Requirements 10

[3.4](#_heading=h.2p2csry) Other Requirements… 10

[**4**](#_heading=h.147n2zr) **Software Design Description 11**

[4.1](#_heading=h.3o7alnk) 1. System Design 11

[4.2](#_heading=h.32hioqz) 2. Database Design 11

[4.3](#_heading=h.1hmsyys) 3. Detailed Design 11

[**5**](#_heading=h.2grqrue) **Software Testing Documentation 12**

[5.1](#_heading=h.vx1227) 1. Scope of Testing 12

[5.2](#_heading=h.3fwokq0) 2. Test Strategy 12

[5.3](#_heading=h.19c6y18) 3. Test Plan 12

[5.4](#_heading=h.28h4qwu) 4. Test Cases 12

[5.5](#_heading=h.37m2jsg) 5. Test Reports 12

[**6**](#_heading=h.1mrcu09) **Release Package & User Guides 13**

[6.1](#_heading=h.46r0co2) 1. Deliverable Package 13

[6.2](#_heading=h.111kx3o)  2. Installation Guides 13

# **Acknowledgement**

During the time of studying, we have well provided and imparted all the necessary and highly valuable professional knowledge by our teacher in Aptech. We have our faithful, careful style and creative mind in studying in the future.

Project is a good opportunity to apply appropriate knowledge and subjects. While performing projects, we have actual and valuable experience. After one term, we were getting all one’s thoughts together and working on a project. During executing this project, we have received a lot of helping from our guider, teachers.

Especially, we wish to thank our teacher - Mr: Dao Quang Linh - for his enthusiastic, thoughtful, and careful guidance.

# **Project Introduction**

## **Project Information**

* **Project Information**
* Project Name (English): Car Rental Service Website
* Project Name (Vietnamese): Ứng dụng website dịch vụ cho thuê xe ô tô
* Project Code: CRSW
* Project Type: Web API Project
* Timeline: From 27th September to 8th November

## **Project Team**

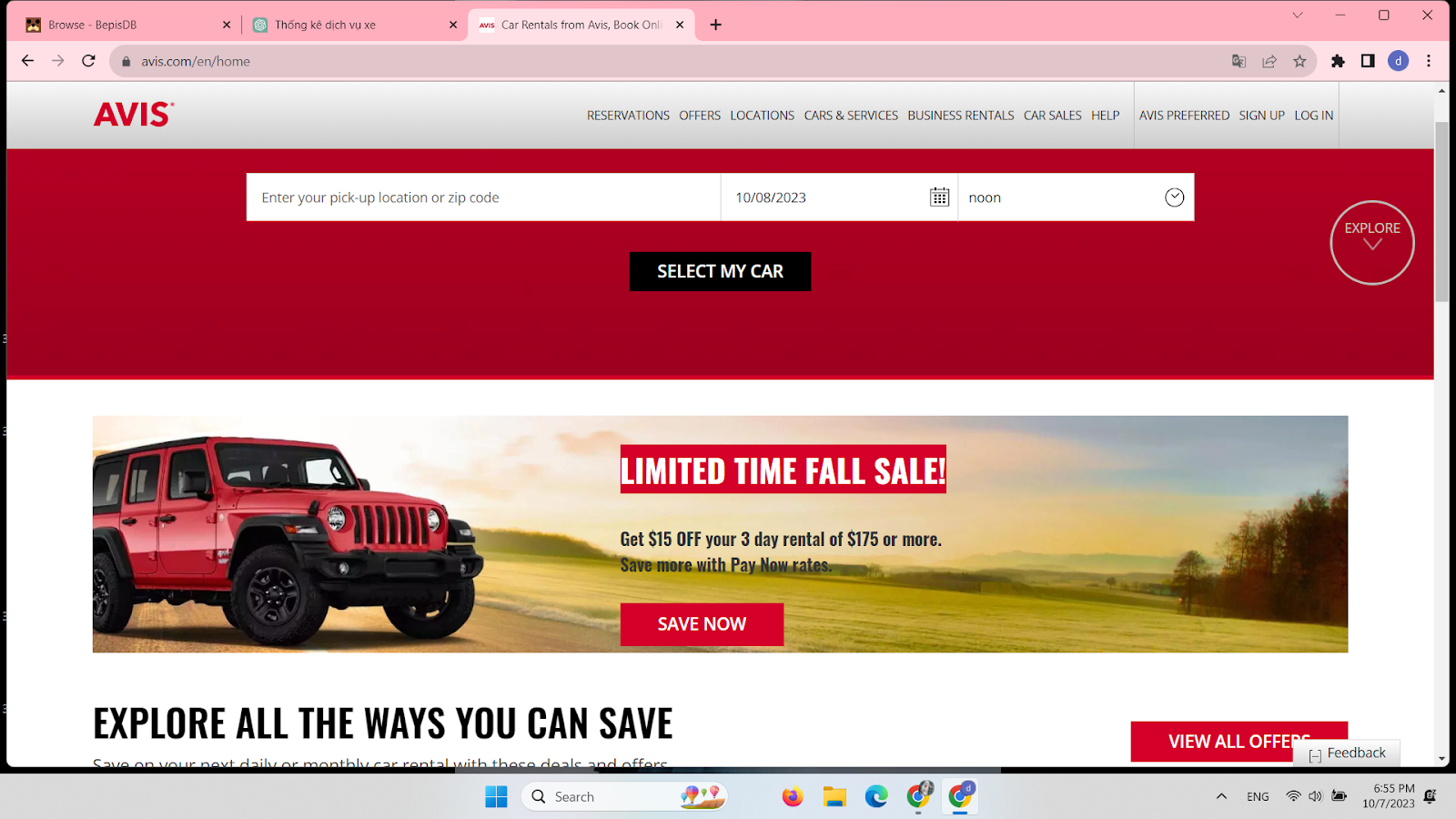
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Full name** | **Roll number** | **Position** | **Phone Number** | **Email** |
| Hoàng Anh | C2010G0141 | Member | 0988869520 | anh.h.953@aptechlearning.edu.vn |
| Vũ Đức Anh | C2108L0272 | Member | 0975502334 |  |
| Nguyễn Đình Duy | C2105L0183 | Leader | 0368415649 | duy.nd.1849@aptechlearning.edu.vn |
| Phạm Đăng Đức | C2108L0255 | Member | 0704125993 | duc.pd.1930@aptechlearning.edu.vn |
| Nguyễn Trung Đức | C2108L0265 | Member | 0836710765 | duc.nt.1940@aptechlearning.edu.vn |

## **Product Background**

* For each person and each family, a car is not only a means of transportation but also a major family asset and not every family can afford it. Not only that, some families, although well-off and able to buy a car, still choose to use self-driving car rental services because it brings many conveniences.
* Nowadays, car prices are increasing very rapidly, compared to the average income of each person, owning a car is very difficult. Based on the news website (https://baotintuc.vn), car prices in 2023 have increased 2.5 times compared to the market price of previous years, leading to workers gradually shifting to renting self-driving cars. According to Mordor Intelligence, the self-driving car rental market in Vietnam is valued at 463.19 million USD in 2021, and is expected to reach 884.71 million USD in 2027 with a compound annual growth rate of 13.82% in the period 2022 - 2027. Meanwhile, according to Statista, the revenue of the self-driving car rental market in Vietnam in 2023 is expected to reach 0.92 billion USD. In addition, the revenue of this market is expected to reach 1.19 billion USD by 2027 with a compound annual growth rate of 6.57% in the period 2023 - 2027.

## **Existing Systems**

* We have some overviews of similar existing solutions, overview of existing methods, limitations of some existing applications.
* Avis.com



|  |  |
| --- | --- |
| Pros:   * Easy to use. * Support for search owners’s names; informations of owners, car….; times-setting * Personalized. * Information is accurate and reliable. | Cons:   * Users and car owners must download apps from the web to fill in their necessary information. * The interface is rough and looks unfriendly * No support Vietnamese |

* Xego.vn



|  |  |
| --- | --- |
| Pros:   * Personalized. * Information is accurate and reliable. * Support Vietnamese | Cons:   * Users and car owners must download app of web to fill in their necessary informations. * Less informations. |

## **The proposed system**

* Online car rental service is a step forward from traditional car rental service. Although building an online car rental service has overcome many limitations of traditional services, the number of car rental websites with a user-friendly interface on the web is not many. The avis.com and xego.vn websites, as listed as examples, are among the largest service websites in this field, but there are still limitations related to personal information registration. In the immediate future, we want to create a website that has the function to fill in important information of individual car renters and owners. Besides, we want the website to have a convenient, user-friendly interface.
* The application will have functions to help users in their work most effectively. Functions such as:   
   1. **Display images and information of rented vehicles** -> help customers and users access a variety of models and types of vehicles that customers want instead of the limited number of models compared to traditional car rentals.  
   2. **Convenient and proactive** -> help customers and users proactively plan and book calendars at any time and in the most comfortable position.  
   3. **Reputation and quality** -> the app is an intermediary that accurately verifies both the customer and the owner of the rental car. Become a tool to ensure the quality and transparency of car leases and vehicle documents.  
   4. **Safety and privacy** -> legal support application, including customer support teams and vehicle owners when issues related to the agreement between the two parties occur.  
   5. **Reviews** -> show reviews of users to provide information to customers and consumers so that they can make wise rental/purchase decisions and have a comprehensive review of the product

## Project Scope & Limitations

**About our project scope:**

* **Consumer demand**
  + Help users and customers who want to rent self-driving cars access the service easily, flexibly and conveniently.
  + Create a platform that seamlessly connects the person who needs to rent a car with the owner of the rental car.
  + Become a tool to help car rental owners easily manage, do business, access more customer files.
* **Conditions and feasibility of the project:**
  + The project has a low investment, minimizes costs, helps users not to pay any additional fees from the website
  + The demand of customers in the market is high, the application has many opportunities to reach users
* **Limitations**
  + The website requires a stable internet connection to launch smoothly.
  + The website does not have an accompanying mobile application.
  + Verifying documents takes time and requires precise operations as instructed.

# **Project Management Plan**

## **The purpose and target of a project**

### **Project Objectives**

* This report is a guide to regulation and supervision of organization and plan of the project. It describes an overview of the project including the proposed system, boundaries and development environment of the capstone project. The ultimate purpose of Project management plan is to clearly deﬁne the roles, responsibilities and the processes of team members to assign tasks and deadlines.

### **Project Risks ( Risk Breakdown Structure (RBS)** )

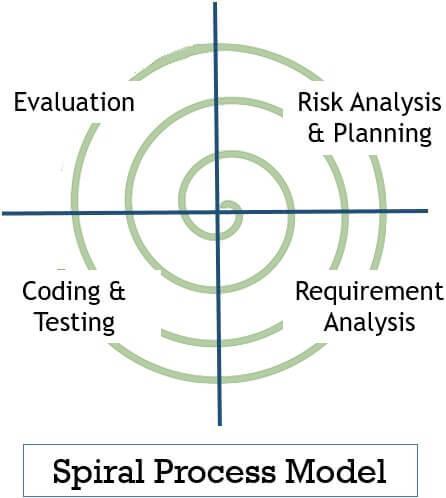
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Probability | Impact | | | |
|  | Low | Medium | High |
| Low |  |  |  |
| Medium |  |  |  |
| High |  | Risk 1 | Risk2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | First Indicator | Prevention  Approach | Mitigation  Approach |
| 1 | Team members are spaced apart | * Teamwork’s efficiency decrease. * Hard to coordinate team members. * It is difficult for group members to meet each other |  |  |
| 2 | Lack of equipment and machinery for the project | * Slow down the completion of tasks * Difficulty sharing information and plans of the project * Unable to do work handed over |  |  |
| 3 |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## **Management Approach**

* ***Software Process Model***
* **We choose this Spiral Model, because it has many advantages:**
* Spiral model enables the better cost estimation
* Spiral model provides continuous and repeated development which helps in risk management.
* Spiral model provides the fast development and the features are added in a systematic manner.
* In a spiral model, client’s get the opportunity to see the software/product after every cycle.
* The spiral model in SDLC is considered as the most preferable model for large and complex projects/software.
* **There are some drawbacks which spiral model consists of:**
* The spiral model is expensive due to the high level of expertise required for risk analysis. Also, the projects take time to develop that causes the overall expenses.
* Due to its high cost, the spiral model is not suited for small projects.

### **Project Process**



* 1. **Planning**

This phase includes the planning process, tasks, resource defining, team planning, timelines and gathering of other project related information. Planning phase includes the estimating costs, schedule for iteration. Once the planning finalizes the team proceed to the next step i.e., Risk Analysis.

* 1. **Risk Analysis**

In Risk Analysis phase, the project prototype is put up for the solution. All the planning which is being planned in the planning phase proceeds with the risk analysis phase to overcome the problems and risks at the beginning stage of project development. Technical and management risk are the factors that are included in the risk analysis phase.

* 1. **Engineering & Execution**

In this phase, the execution part is being done by engineers and developers. The planning and risk analysis once finalized is being processed in the execution phase where all the coding, testing and deploying of software takes place.

* 1. **Evaluation**

In the evaluation phase, the product is being assessed by the client and provided with the revert if any changes required from client side. Evaluation phase includes all the above phases whether it is about planning, risk analysis, engineering & execution the client goes through each phase for evaluating the product or software.

### **Quality Management**

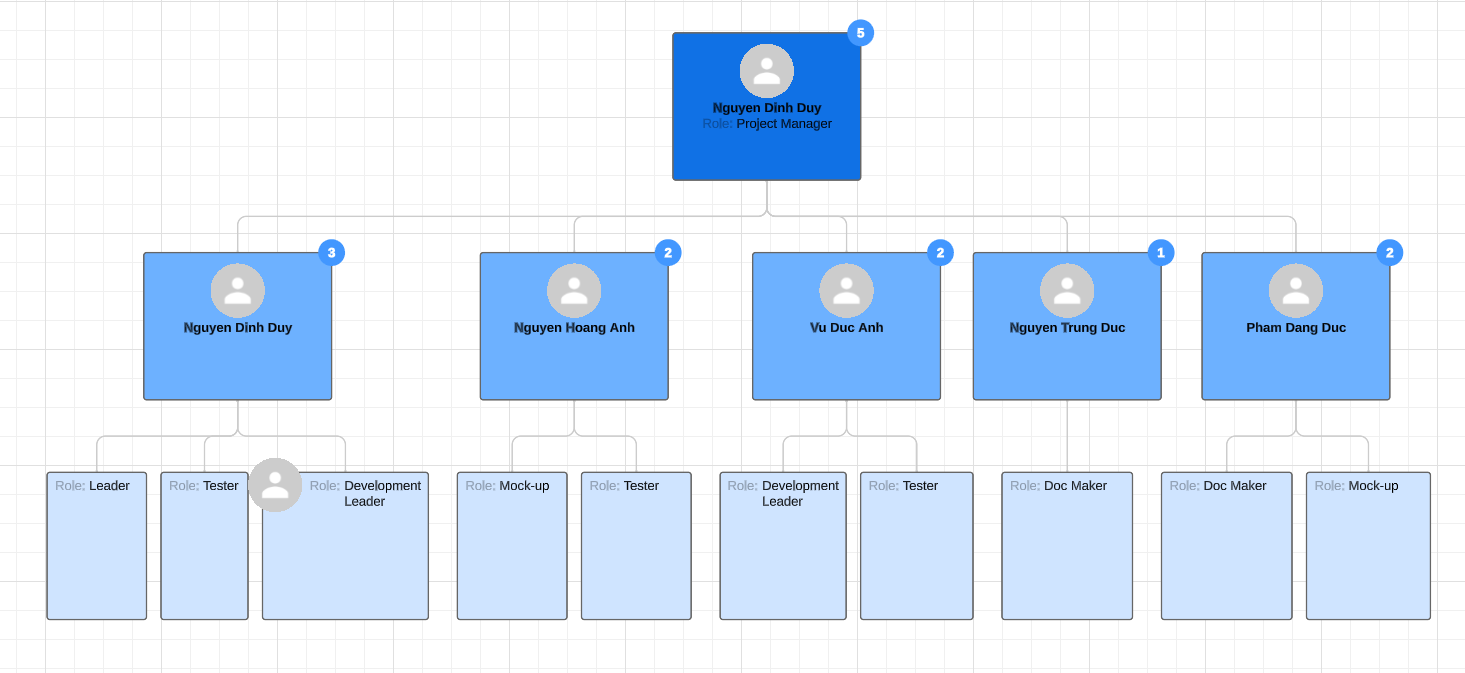
*[Provide the approach you would apply to improve the project quality, reach the project quality objectives*

### **Training Plan**

*[You need to plan the training activities in case any of your team member lack of knowledge/skills to handle the project works]*

*=>*

## **Responsibility Assignments**



|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Roles** | **Responsibilities** |
| **1** | **Nguyen Dinh Duy** | * Project Manager. * Technical Leader. * Development Team Leader. | * Managed and lead the team project. * sAnalyzed and built materials for the project implementation process. * Developed project plan and implementation schedule. * Assigned responsibilities and tasks to members. * Maintained an organization’s work flow and kept its lines of communication open. * Monitored the progress of project implementation * Identified, assessed and mitigated project risks for successful project completion. * Communicated with other members to guarantee the project was on schedule and within scope. * Reviewed and supported requirement analysis. * Led requirement analysis team. * Understood clearly about project requirements and wrote SRS. * Maintained all documents by preparing standard templates, managing the process of creating, reviewing and updating documents in a team. * Researched algorithms and learned new techniques and architecture to use for projects. * Guided other members in the team to understand and to apply new techniques to the project. * Supported team members. |
| **2** | **Vu Duc Anh** | * Development Team Member * Design Team Leader | * Reviewed and supported requirement analysis. * Understood clearly about the project requirements and wrote SRS. * Identified certified solutions to meet the requirements. * Had a basic understanding of technology used for the project. * Guided other members to understand and to apply the technology to the project. * Summarized and made the project weekly report * Supported other team members. |
| **3** | **Hoang Anh** | * Development Team Member. * Design Team Member | * Had a basic understanding of technology used for the project. * Confirmed developing plan and assigned coding functions to other members. * Supported team in developing some functions. * Supported other team members. |
| **4** | **Nguyen Trung Duc** | * Development Team Member. * Test Team Member. * Document Team Member | * Had a basic understanding of technology used for the project. * Read & understand SRS and deploy document. * Optimized coding. * Executed testing. * Supported other team members. |
| **5** | **Pham Dang Duc** | * Test Team Leader. * Document Team leader. | * Supported requirement analysis. * Executed test cases, recorded test case results, documented and tracked defects, and performed test coverage analysis. * Supported team to develop some functions. |

## **Project Communications**

* Team meetings are held twice a week, on Wenesday and Saturday. Face-to-face meetings are the most effective way to understand requirements and communicate on the project. The outcomes of the meeting are the report of the completed and not completed work to improve and corresponding solutions for the project.
* All members will perform the task assigned by the leader on Thursday. On Monday, members will report what they have done in a week. After that, the team will work together to solve the problem and have plan for the coming week.
* **Communication Channels:**
* Email.
* Facebook.
* Phone.
* Face-to-face meetings
* Zalo
* **File Management:**
* Google Drive.
* Github

## **Project Work Plan**



## **Configuration Management**

### **Document Management**

* We manage project document by using Microsoft Word 2016
* Microsoft Word is an editing program that allows users to work on text documents with many effects such as fonts, text colors, graphic effects, multimedia effects, audio, video.
* Through Microsoft Word formatting tools, you can create professional quality documents effectively. Besides, Word also includes editing and revision tools that help you collaborate with people easily.
* Microsoft Word has the following functions: Drafting contracts and minutes; Drafting reports; Create CV; Create mail; Create envelopes; Watermark; Autocorrect; Document protection; Count from; Set display mode; …

### **Source Code Management**

* We manage project source code by using Git/GitHub
* **Manage source code easily**
* When you create a repo, the entire source code of that repo is saved on GitHub. Here, you can review your progress through comments after each commit. And the beauty here is that many people can make the same repo.
* The first benefit is that you know who committed and what. Next, your source can develop into many branches. Principles for working with branches like this: You can branch many branches to develop the project. But in the end, you have to merge back into the MASTER branch to get the complete project.
* **Tracking changes across versions**
* When there are many members working on a project, it is quite complicated to keep track of revisions - who changed what, when and where those files are stored. Don't worry because GitHub has taken this into account for you, by always saving the changes you push to the repository. Similar to Microsoft Word or Google Drive, you have a version history in case previous versions are lost or not saved.
* **Github is a great resource**
* With the Explore function, you can track and search for open source projects according to the technology pattern you like. Github supports code search regardless of whether it is a separate project or a website. In addition, this platform also has quite good SEO so users can search for any code string shared publicly.

### **Tools & Infrastructures**

* 1. **Tools**
* **Version Control Systems:** Git 2.13.0 or higher.
* **Source Code Management Systems:** GitHub.
* **Project Management Tools:** Microsoft Office 2016 (Word, Excel, PowerPoint).
* **Diagram Tools:** https://app.diagrams.net ( website ).
* **Design Tools:** Photoshop CC 2019.
* ETC….
  1. **Techniques**
* **Programming Language**
* ***English***
* ***Vietnamese***

# Software Requirement Specification

## User Requirements

*[Provide specification of the user requirement: actor, use case diagram(s), use case description, etc.]*

**

***FIRGURE 3.2.1.2*** *Use Case Diagram*

**UC-01 : Register:**

|  |  |
| --- | --- |
| **Use Case ID** | UC - 01 |
| **Use Case Name** | Register |
| **Actors** | Guest, User |
| **Description** | Register an account in the system. |
| **Pre-Condition** | * User is in login screen. * Internet permission |
| **Post-Condition** | * An account is registered. * Go to home screen. |
| **Trigger** | * Tap on “Sign Up” button. * Tap on email field. * Tap on name field. * Tap on password field. * Tap on “Register” button. |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1.User click on “Sign Up” button |  | |  | 2. A resignter form will show up | | 3.User input form: Phonenum/Email, Name, Password, re-Password. Click and check box |  | |  | 4. The system updates user information -> adds the account to the database | | 5. Back to application interface |  | |
| **Alternative flow** | * When the user does not enter the required information in the registration form -> There are required fields that validated the user to fill in. * The “Confirm register” button will not available to click -> need to refresh the website |
| **Exception flow** | User enter not validated name, email or password. |
| **Business rule** | In order to log in to the system, user name and password are required. |

**UC-02: Login:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 02 |
| **Use Case Name** | Login |
| **Actors** | Guest, Registered User |
| **Description** | - This use case allows users who has username and password to login to the system to use system’s functions. Function that users can use must be allowed by admin.  - Allow user to login. |
| **Pre-Condition** | * User is in Home Screen * User have not logged in yet. * Internet permission. |
| **Post-Condition** | * Logged account. * Go to Home Screen. |
| **Trigger** | * Tap on “Login”. * User enters email/phonenum and passwords. * Tap on “Login” button. |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. Click on header - > “Sign in” button |  | |  | 2. Show “Login” screen with: Email, Password. | | 3. Enter email, password. Then, tap on “Sign In” button. |  | |  | 4. Show screen “Home Screen” with data of this account. | |
| **Alternative flow** | N/A |
| **Exception flow** | Not fill email and password to field or fill wrong email and password. |
| **Business rule** | In order to log in to the system, user name and password are required existed. |

**UC-03:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 03 |
| **Use Case Name** | Logout |
| **Actors** | Registered User |
| **Description** | * This Usecase allows users who have logged in to the site to log out and delete data at the logged out device |
| **Pre-Condition** | * The user has logged in to the site successfully * Internet permission. |
| **Post-Condition** | * Account is logged out. * Do not save the account after the user logs out * Reload the page without the logged out user's data |
| **Trigger** | * Tap on user profile. * Tap on “Logout” button.   2.  - Click at arrow icon   * Tap on “Logout” button. |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. Click on header - > “profile” button |  | |  | 2. Show Manage profile screen | | 3. Click “logout” button |  | |  | 4. Delete the account on this device  5. Back to screen “Home screen” without data of user | |
| **Alternative flow** | N/A |
| **Exception flow** | N/A |
| **Business rule** | N/A |

**UC-04:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 04 |
| **Use Case Name** | Forgot password |
| **Actors** | Car Owner, Car Rental |
| **Description** | * This Usecase allows the user to enter the registered information to reset the new password |
| **Pre-Condition** | * Getting login user failed. * Internet permission. * User at login screen |
| **Post-Condition** | * The user's account password is reset * User returns to login screen |
| **Trigger** | * Tap on “login” button * Click “Forgot password” * User enter “Username/Phonenum” to get SMS code * User enter SMS code * User enter new password and confirm new password |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. Click on header - > “Login” button |  | |  | 2. Show Login screen | | 3. Click “forgot password” button |  | |  | 4. Show Forgot pass word screen | | 5. Enter “Username/Phonenum” |  | |  | 6. Check Username/Phonenum validated with database.  7. Send SMS code to user’s phone. | | 8. User enter SMS code |  | |  | 9. Check SMS code validate.  10. Show reset password screen. | | 11. User enter new password.  12. User re-check pass word |  | |  | 13. Update new password to database.  14. Return to login screen | |
| **Alternative flow** | N/A |
| **Exception flow** | * The user filled in the wrong account name/phone number * The user entered the wrong SMS code |
| **Business rule** | To be able to reset the password. User's account must already exist |

**UC-05:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 05 |
| **Use Case Name** | Display all the rental cars in website |
| **Actors** | All User, Guest |
| **Description** | * This Usecase allows the user to enter home screen website then show all the items in website |
| **Pre-Condition** | * Internet permission. |
| **Post-Condition** | * The user screen displays the website homepage and displays a list of rental cars. |
| **Trigger** | * User logs in to the site following the path |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. Tap the link, or website logo |  | |  | 2. Get product list from data  3. Show vehicle list on home page | |
| **Alternative flow** | N/A |
| **Exception flow** | N/A |
| **Business rule** | N/A |

**UC-06:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 06 |
| **Use Case Name** | Searching Feature |
| **Actors** | All User |
| **Description** | * This Usecase allows users to look up and search for the desired product by filters or keywords |
| **Pre-Condition** | * Stable internet connection |
| **Post-Condition** | * Display a list of products that match user search results |
| **Trigger** | * The user on the home screen of the website.   The site has 2 data types for the search feature:   1. The user fills in the search bar with the keyword to be searched for. 2. Users choose tags that match their wants and needs |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. The user accesses the car rental service's website  2. Click on the search bar or select the card that suits the user's needs |  | |  | 3. The system displays the filters and tags of the product for users to select | | 4. User starts typing search criteria  5. User clicks "Search" icon to perform a search |  | |  | 4. The system conducts a search based on the criteria provided by the user.  5. The system displays search results on a separate page.  6. The results include a list of available rental cars and satisfy the search criteria. | |
| **Alternative flow** | 1. *No Results Found*  * After the user performs a search, the system does not find any results that match the search criteria. * The system displays a message to the user indicating that no results were found based on their search criteria. * The user may be advised to adjust their search criteria or try again with different values.  1. *Filtering Search Results*  * After viewing the list of search results, the user can choose to apply additional filters to narrow down the results. * They can filter by price, car type, brand, special features, and other criteria to customize the search results according to their preferences.  1. *Canceling Search*  * During the search process , the user may decide to cancel or go back to a previous step. The system provides options to cancel the current search and return to the main search page. |
| **Exception flow** | * Lost Internet Connection * Invalid Search Criteria * Database Error * Unspecified Search Error |
| **Business rule** | Valid Search Rules |

**UC-07:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 07 |
| **Use Case Name** | Display detail infomation car |
| **Actors** | All User |
| **Description** | * Allow the user to view detailed information about a specific car on the car rental service's website |
| **Pre-Condition** | * Stable internet connection * The user has accessed the car rental service's website * The selected car for rental must exist in the system and have detailed information. |
| **Post-Condition** | * The user has viewed detailed information about the car for rental and can decide to continue with the booking process or return to the search page. |
| **Trigger** | * Users can access the vehicle list page * View a list of available rental vehicles and select a vehicle to view details * Click the "View Details" option or car image |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. The user selects a specific car from the search results or the car listing page |  | |  | 2. The system displays a car detail page for the selected car | | 3. The user can scroll down to view all the detailed car information or look at relevant images |  | |
| **Alternative flow** | N/A |
| **Exception flow** | * Lost Internet Connection * In the event that the car is no longer available or detailed information cannot be found, the system should display an error message and guide the user back to the search page. |
| **Business rule** | N/A |

**UC-08:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 08 |
| **Use Case Name** | Manage the account |
| **Actors** | Registered User |
| **Description** | * To allow users or account managers to perform account management activities, such as updating personal information, changing passwords, and managing account settings on the website |
| **Pre-Condition** | * Stable internet connection * The user has accessed the service's website and has logged into their account. * The account has been verified through a username and password or other authentication methods. |
| **Post-Condition** | * The user or account manager has completed account management activities and can return to using their account or perform other account management tasks. * Changes are stored and updated in the system. |
| **Trigger** | * Users successfully logs into their account * Tap on “Profile” icon * User Chooses Account Management Option * Users confirms saving their installation settings |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. User login successful  2. Accesses the account management section from the main interface. |  | |  | 3. Show “Profile” screen.  4. The system displays account management options for the user. | | 5. The user or account manager selects a specific option to proceed. |  | |  | 6. The system allows to update information.  7. save the changes and exit the account management page. | |
| **Alternative flow** | * Change Password: The user selects the "Change Password" option. They must provide the current password and then enter a new password. The system checks the validity of the new password and allows them to save the changes. * due to an expired login session, the system should redirect them back to the “profile” screen |
| **Exception flow** | * Lost Internet Connection * Invalid Password * Invalid Personal Information * Duplicate personal information * Authentication Failure |
| **Business rule** | N/A |

**UC-09:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 09 |
| **Use Case Name** | Authenticate Identification |
| **Actors** | Registered User |
| **Description** | * This usecase to verify the identity of a user to ensure security and access rights. |
| **Pre-Condition** | * Stable internet connection * The user is attempting to access an account that requires identity authentication to be able to book a car rental |
| **Post-Condition** | * Successful authentication |
| **Trigger** | * Users successfully logs into their account * Tap on “Profile” icon or “Register to be owner” * Tap on “Identity authentication” * User enter ID, Name, DOB, images of ID * User confirm update. |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. User login successful  2. requires identity authentication. |  | |  | 3. The system requests authentication through the established authentication method | | 4. User enter ID, Name, DOB, images of ID |  | |  | 5. The system checks the provided authentication information valid  6. Checks information against stored data  7. To “Rental Register” screen. | |
| **Alternative flow** | N/A |
| **Exception flow** | * Lost Internet Connection * Authentication Failure * information is incorrect or does not match the stored data |
| **Business rule** | * The authentication process should adhere to legal and regulatory requirements, such as data protection laws and industry-specific standards |

**UC-10:**

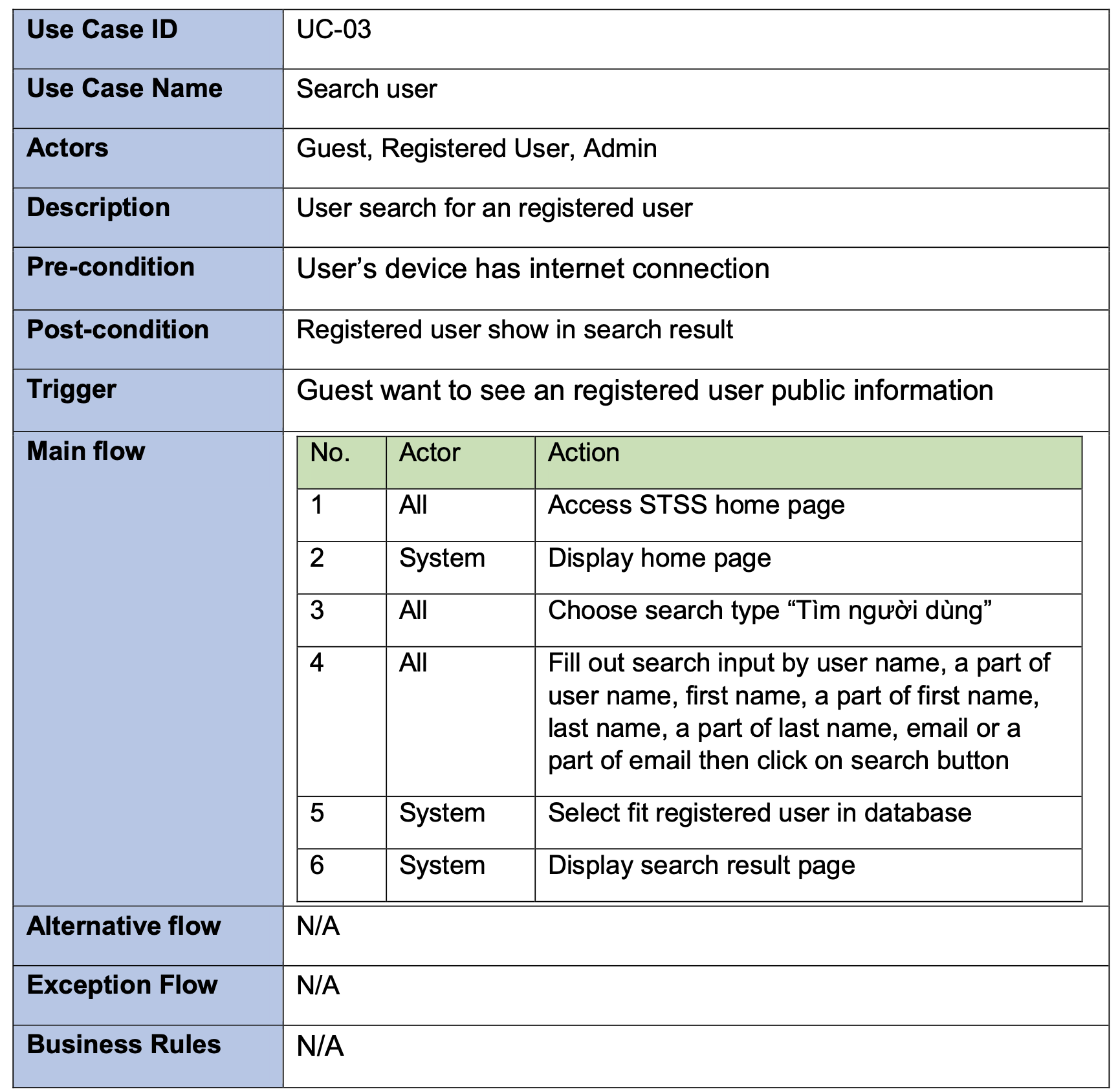
|  |  |
| --- | --- |
| **Use Case ID** | UC – 10 |
| **Use Case Name** | Rental registry |
| **Actors** | Registered User |
| **Description** | * This Usecase for users to register an account as a car owner, rent a car |
| **Pre-Condition** | * Account logged in successfully * Account successfully verified for identification |
| **Post-Condition** | * The account has the right to publish rental cars, manage rental cars * Account granted vehicle owner rights |
| **Trigger** | * Users successfully logs into their account * Tap “Register to be owner” button * Tap on “Identity authentication” * User enter ID, Name, DOB, images of ID * User confirm update. * User enters car license plate number * Enter the parameters of the car and description for the vehicle * User taps next * Set your listing price * User uploads vehicle image as required by data * Confirm |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. User tap “Rental registry” button |  | |  | 2. Show “information vehical” form | | 3. The user enters all the required information. |  | |  | 4. Checks the provided information valid | | 5. Tap to confirm information of cars |  | |  | 6. Show “Set price” screen | | 7. Enter listing price |  | |  | 8. Check valid of price  9. Show “Set images” form | | 9. User set images of car  10. User enterVehicle registration/ vehicle inspection required |  | |  | 11. Verify that the vehicle details match the required data, which is preset  12. Database Duplicate Check  13. If the requirements are met, add the product to the product list  14. Display success notification panel | |
| **Alternative flow** | N/A |
| **Exception flow** | * Lost Internet Connection * Authentication Failure * No Vehicles Available * System Error |
| **Business rule** | * Vehicle Condition Report: A condition report is usually completed before and after each rental to document any pre-existing damage or issues with the vehicle. Customers are responsible for any new damage during the rental. * Insurance Coverage: required to have valid auto insurance coverage * Reasonable price with the quality of the car * The primary vehicle, owned by the registered owner. * The vehicle must be licensed and legally operable in accordance with the laws of the host country. |

**UC-11:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 11 |
| **Use Case Name** | Create/Delete/Edit/Update for rental cars |
| **Actors** | Car Owner |
| **Description** | * This use case focuses on the actions related to managing the inventory of rental cars within a car rental service. It involves creating new car listings, deleting listings, editing existing listings, and updating information about rental cars. |
| **Pre-Condition** | * Must have the necessary access privileges and authentication to manage the rental car inventory. |
| **Post-Condition** | * New car listings are added to the inventory when created. * Deleted car listings are removed from the inventory. * Edited car listings are updated with the new information. * Updated car listings reflect any changes made to their information. * To ensure that the rental car inventory is up-to-date and accurate, providing customers with a selection of available cars to rent. |
| **Trigger** | * Users successfully logs into their account * Tap on “Profile” icon * Tap on “My cars” button/icon * Select the vehicle that the user wants to manage and modify |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. User tap “Profile” icon  2. Tap “My cars” Button |  | |  | 3. Show “Manage car” screen | | * Create Option:   4. Tap “Register new car” |  | |  | 5. The system prompts the user to input car details such as make, model, year, rental price, available dates, and other relevant information.  6. Once entered, the new car listing is added to the inventory. | | * Delete Option: | 7. The system displays a list of available car listings. | | 8. The car owner selects a car listing to delete. |  | |  | 9. A panel will show up to confirm delete  10. The selected car listing is removed from the inventory. | | * Edit Option:   11. The user selects a car listing to edit. |  | |  | 12. The system allows the administrator to make changes to the car's details  13. After the edits are made, the updated information is saved in the system. | |
| **Alternative flow** | * If there are errors in creating, deleting, editing, or updating a car listing, the system should provide appropriate error messages and allow the administrator to correct the issue. |
| **Exception flow** | * Lost Internet Connection * Unauthorized Access: If the account has not been registered as a vehicle owner, the 'Manage Cars' section will not appear |
| **Business rule** | * Only car owner can create new car listings. All required information, such as the make of the car, model, rental price, and availability dates, must be provided when creating a car listing. * Only car owner can delete car listings. The deletion process must be confirmed to prevent accidental deletion. * Only car owner can edit existing car listings. * The system should check for errors when administrators perform create, delete, edit, or update actions on car listings. Error messages and guidance for resolving errors should be provided when errors occur. |

**UC-12:**

|  |  |
| --- | --- |
| **Use Case ID** | UC – 12 |
| **Use Case Name** | Rent cars |
| **Actors** | Registered User |
| **Description** | * This use case focuses on the actions related to managing the inventory of rental cars within a car rental service. It involves creating new car listings, deleting listings, editing existing listings, and updating information about rental cars. |
| **Pre-Condition** | * Must have the necessary access privileges and authentication to manage the rental car inventory. |
| **Post-Condition** | * New car listings are added to the inventory when created. * Deleted car listings are removed from the inventory. * Edited car listings are updated with the new information. * Updated car listings reflect any changes made to their information. * To ensure that the rental car inventory is up-to-date and accurate, providing customers with a selection of available cars to rent. |
| **Trigger** | * Users successfully logs into their account * Tap on “Profile” icon * Tap on “My cars” button/icon * Select the vehicle that the user wants to manage and modify |
| **Main flow** | |  |  | | --- | --- | | Actors | System | | 1. User tap “Profile” icon  2. Tap “My cars” Button |  | |  | 3. Show “Manage car” screen | | * Create Option:   4. Tap “Register new car” |  | |  | 5. The system prompts the user to input car details such as make, model, year, rental price, available dates, and other relevant information.  6. Once entered, the new car listing is added to the inventory. | | * Delete Option: | 7. The system displays a list of available car listings. | | 8. The car owner selects a car listing to delete. |  | |  | 9. A panel will show up to confirm delete  10. The selected car listing is removed from the inventory. | | * Edit Option:   11. The user selects a car listing to edit. |  | |  | 12. The system allows the administrator to make changes to the car's details  13. After the edits are made, the updated information is saved in the system. | |
| **Alternative flow** | * If there are errors in creating, deleting, editing, or updating a car listing, the system should provide appropriate error messages and allow the administrator to correct the issue. |
| **Exception flow** | * Lost Internet Connection * Unauthorized Access: If the account has not been registered as a vehicle owner, the 'Manage Cars' section will not appear |
| **Business rule** | * Only car owner can create new car listings. All required information, such as the make of the car, model, rental price, and availability dates, must be provided when creating a car listing. * Only car owner can delete car listings. The deletion process must be confirmed to prevent accidental deletion. * Only car owner can edit existing car listings. * The system should check for errors when administrators perform create, delete, edit, or update actions on car listings. Error messages and guidance for resolving errors should be provided when errors occur. |



## Functional Requirements

### System Functional Overview

*[Provide functionality overview of software system: screen flow, screen descriptions, system user roles, screen authorization, non-screen functions, ERD]*

# 3.2.1.1 Mockup

* Screen: Application Interface

***FIRGURE* *3.2.1.1***

**Description:**

## Non-Functional Requirements

### Reliability

-This is a communication website, so reliability is very important. The information from this website must be real and exactly (phone number, store address)

### Availability

Systems created using the platform should be generally available to users.

### 3.2.3.3 Security

The security is necessary for almost website. Protect user information is important

### Maintainability

The platform should contain interfaces that are extensible to allow new communication protocols to be added with ease.

### Portability

We can connect website by every device have internet

## Other Requirements…

# Software Design Description

*[Provide final software design information follow the template as part II in the Report #4]*

## 1. System Design

### 1.1 System Architecture

*[The content of this section includes the overall diagram which includes the sub-systems, the external systems, and the relationship/connection among them. You need also provide the explanation for each of the diagram components (modules, sub-systems, external systems, etc.)].*

### 1.2 Package Diagram

*[Provide the package diagram for each sub-system. The content of this section includes overall package diagram(s) and the explanation for each package (or namespace)]*

## 2. Database Design

4.2.2.1 Class Diagram

A screenshot of a computer

Description automatically generated

* **Table detail**
* **Table car**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| car\_id | Integer | -primary key  -unsigned | Car id |
| price | Float |  | Car price |
| year\_manufacture | Integer |  | Year of car’s manufacture |
| image | Varchar(255) |  | Car image |
| is\_delete | Integer |  | car status info |
| number\_plate | Varchar(255) |  | Car number plate |
| description | Text |  | Car description |
| Address | Varchar(225) |  | Car address |
| custumer\_id | Integer |  | Custumer id |
| brand\_id | Integer |  | Car brand id |
| Model\_id | Integer |  | Car model id |
| car\_type\_id | Integer |  | Car type id |
| car\_status\_id | Integer |  | Car status id |

* **Table customer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| custumer\_id | Integer | -primary key  -unsigned | Custumer id |
| phone | Varchar(50) |  | Custumer phone number |
| email | Varchar(255) |  | Custumer email address |
| name\_display | Varchar(50) |  | Custumer name display on system |
| full\_name | Varchar(50) |  | Custumer full name |
| birthday | Varchar(255) |  | Custumer birthday |
| password | Varchar(50) |  | Custumer account password |
| role\_name | String |  | Custumer role |
| id\_number | Varchar(255) |  | Custumer national id |
| id\_fontside | Varchar(255) |  | Custumer national id fontside |
| id\_backside | Varchar(255) |  | Custumer national id backside |
| Verify\_flg | Integer |  | Verify status |

* **Table booking**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| booking\_id | Varchar(20) | -primary key  -unsigned | Booking id |
| start\_date | Varchar(20) |  | Booking start date |
| start\_time | Varchar(20) |  | Booking start time |
| end\_date | Varchar(20) |  | Booking end date |
| end\_time | Varchar(20) |  | Booking end time |
| Total | Integer |  | Booking total |
| Complete\_flg | Integer |  | Booking status |
| status\_id | Intger |  | Booking status id |
| create\_by | Integer |  | Booking create by |
| car\_id | Integer |  | Car id |

* **Table feedback**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| feedback\_id | Integer | -primary key  -unsigned | Feedback id |
| rate | Integer |  | Feedback star rate |
| comment | Text |  | Feedback content |
| create\_at | Varchar(20) |  | Feedback create at |
| create\_by | Integer |  | Feedback create by |
| car\_id | integer |  | Car id |

* **Table favourite\_car**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| favourite\_car\_id | integer | -primary key  -unsigned | Favourite car id |
| custumer\_id | Integer |  | Custumer id |
| car\_id | Integer |  | Car id |

* **Table address**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| address\_id | integer | -primary key  -unsigned | Address id |
| address\_name | Varchar(255) |  | Address name |
| customer\_id | integer |  | Custumer id |

* **Table booking\_status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| booking\_status\_id | Integer | -primary key  -unsigned | Booking status id |
| Booking\_status\_name | Varchar (50) |  | Booking status name |

* **Table car\_status**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| car\_status\_id | Integer | -primary key  -unsigned | Car status id |
| car\_status\_name | Varchar (50) |  | Car status name |

* **Table brand**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| Brand\_id | Integer | -primary key  -unsigned | Brand id |
| Brand\_name | Varchar (255) |  | Brand name |

* **Table model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| model\_id | Integer | -primary key  -unsigned | Model id |
| model\_name | Varchar(255) |  | Model name |
| brand\_id | Integer |  | Brand id |

* **Table vehicle\_registration**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| vehicle\_registration\_id | Integer | -primary key  -unsigned | Vehicle registration id |
| vehicle\_registration\_image | Varchar(255) |  | Vehicle registration image |
| vehicle\_inspection\_image | Varchar(255) |  | Vehicle inspection image |
| car\_id | Integer |  | Car id |

* **Table car\_type**

|  |  |  |  |
| --- | --- | --- | --- |
| **Column** | **Data type** | **Constraint** | **Description** |
| car\_type\_id | Integer | -primary key  -unsigned | Car type id |
| car\_type\_name | Varchar(255) |  | Car type name |
| number\_seats | Integer |  | Car number seats |

*[Provide the files description, database table relationship & table descriptions]*

## 3. Detailed Design

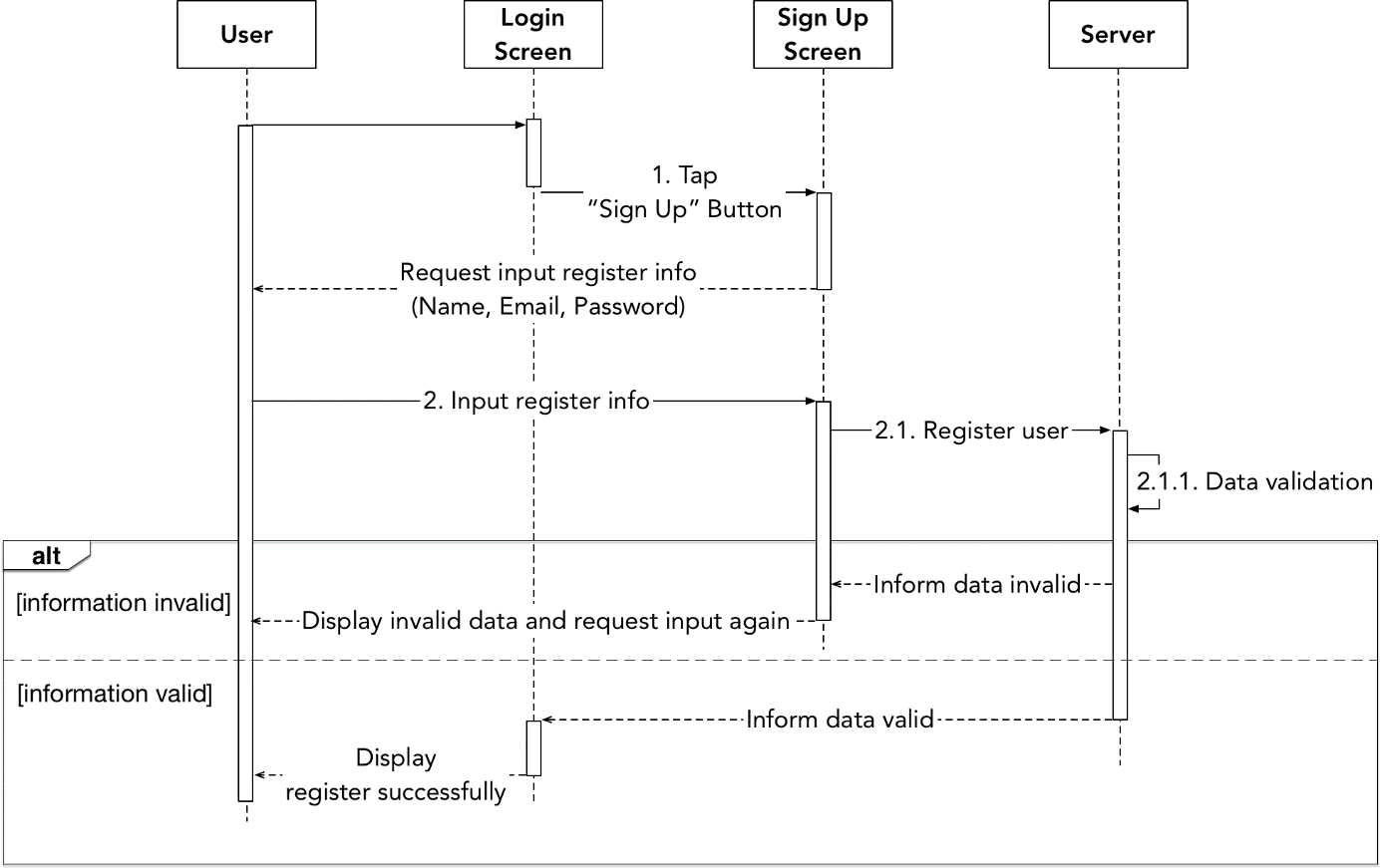
### 3.1 <Sequence Diagram>

*[Provide the detailed design for the feature <Feature Name1>. It includes Class Diagram, Class Specifications, and Sequence Diagram(s);* ***For the features/functions with the same structure of class & sequence diagrams, you need to provide the diagrams once for one feature/function and refer to those diagrams from other features/functions****]*

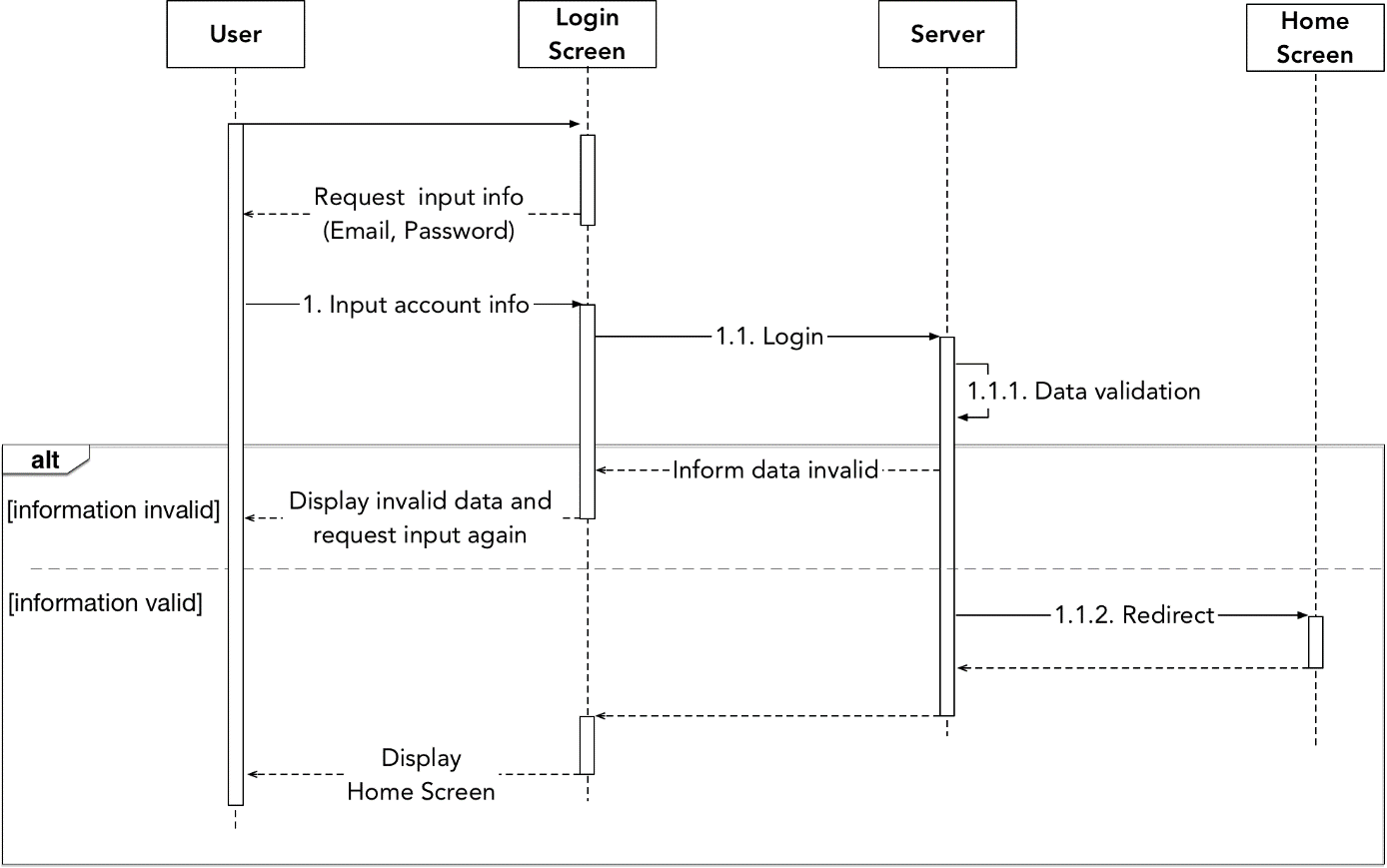
#### 3.1.1 Class Diagram

*[This part presents the class diagram for the relevant feature]*

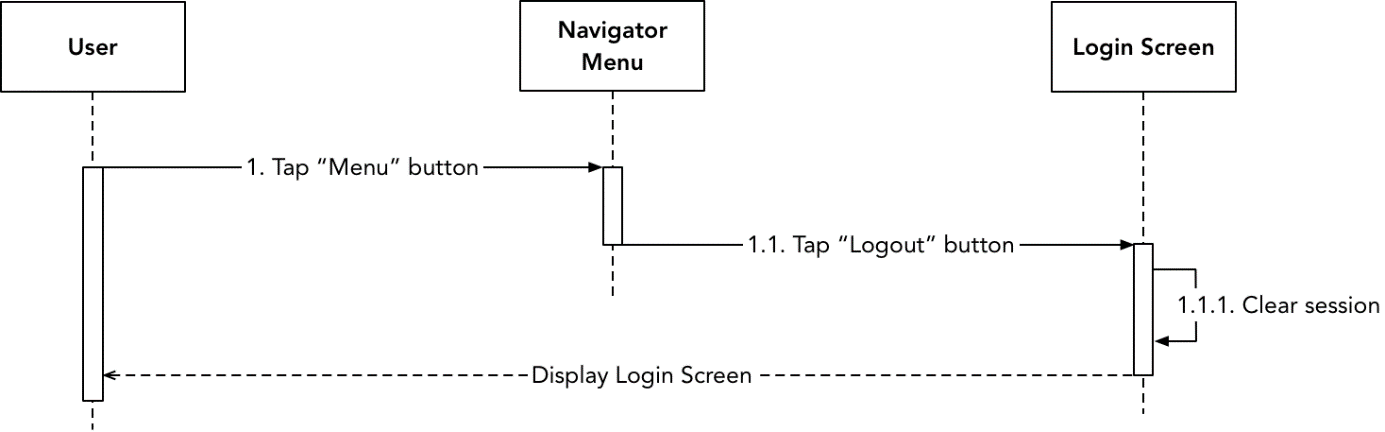
***4.3.1.1 <Sequence diagram for Use case “Register”>***

**

***4.3.1.2 <Use case “Login”>***



***4.3.1.3 < Sequence diagram for Use case “Logout”>***

******

***4.3.1.3 < Sequence diagram for Use case “Forgot password”>***

***4.3.1.3 < Sequence diagram for Use case “Display”>***

*A diagram of a car

Description automatically generated*

***4.3.1.3 < Sequence diagram for Use case “Searching”>***

***A diagram of a model

Description automatically generated***

### 3.2 <Feature/Function Name2>

# Software Testing Documentation

*[Provide final software testing information follow the template as part II in the Report #5]*

## 1. Scope of Testing

*[Describe the scopes of the test. Those include the target-of-test’s features, functions, and non-functional requirements that will or will not be tested.*

*Describe the stages/levels of testing that would be applied to your project - Unit, Integration, or System test. Each includes the in-charge, inputs/time, focuses, acceptance criteria.*

*List any constraints or assumptions made during the development of this document that may impact the design, development or implementation of testing]*

## 2. Test Strategy

*[List out and describe all testing types (you can refer the test types listed below or any other test types to selected the suitable ones for the project; for each selected test types you need to provide the following information: test objective, technique, completion criteria, etc.), test levels that those test types would be performed, & the details of test supporting tools would be used in the project]*

### 2.1 Testing Types

*[List out and describe here the testing types which you would apply in your project. You need to mention following information for each type of testing: objective, technique, completion criteria]*

### 2.2 Test Levels

*<List out and describe here the testing levels which you would execute in your project. Besides, clearly state the test types which are performed in each test level that you plan for this project>*

### 2.3 Supporting Tools

*<List of the test supporting tools which will be employed for this project>*

## 3. Test Plan

### 3.1 Human Resources

*[List and provide the details on roles and responsibilities of the project members who would involve in testing works]*

### 3.2 Test Environment

*[List and provide the details about the tools (software, hardware, infrastructure) which the project would use for testing]*

### 3.3 Test Milestones

*[Separate test milestones, which should be identified to communicate project status accomplishments]*

## 4. Test Cases

*[Prepare the details on the test cases following the provided template*

* *Unit Test Cases: Report5\_Unit Test.xls*
* *Other Test Cases (IT, ST, AT): Report5\_Test Report.xls]*

## 5. Test Reports

*[Provide the test result, statistics and the relevant test analysis for your testing in the project]*

# Release Package & User Guides

*[Provide final software testing information follow the template as part II in the Report #6]*

## 1. Deliverable Package

*[The section will list all source programs, scripts, documents with version number in this release. You can see the example following table for reference, can customize or delete if not using belong to each project characteristics]*

| **No.** | **Deliverable Item** | **Description** |
| --- | --- | --- |
| 1 | Schedule/Task Tracking |  |
| 2 | Project Backlog |  |
| 3 | Source Codes |  |
| 4 | Database Script(s) |  |
| 5 | Final Report Document |  |
| 6 | Test Cases Document |  |
| 7 | Defects List |  |
| 8 | Issues List |  |
| 9 | Slide |  |

## 2. Installation Guides

### 2.1 System Requirements

*[Define any system requirements necessary to support the application, including the software and relevant configurations]*

### 2.2 Installation Instruction

*[Includes installation instructions and configuration guidelines]*

## 3. User Manual

### 3.1 Overview

*[Descript the overview of the application and if could, insert the features workflow to help user has the overview of all the features in this application]*

### 3.2 Workflow 1

*[Describe the purpose of this workflow, draw workflow diagram and other relevant diagrams]*

*[Describe the detailed guides for the workflow by providing the brief description, step by step guides (attached with user interface) of how to use that function]*

### 3.3 Workflow 2