VIETNAM NATIONAL UNIVERSITY

UNIVERSITY OF ENGINEERING AND TECHNOLOGY

# PROJECT REPORT

# CAR RENTAL MANAGEMENT WEBSITE

COURSE: INT2208E 25

TEACHER: TS. Vũ Thị Hồng Nhạn

Lê Minh Quý: 22028190 (Leader)

Đào Thị Thu Hường: 22028292

Hoàng Linh Chi: 22028132

Nguyễn Thu Hà: 22028282

Hoàng Văn Dũng: 22028188

| Mã sinh viên | Họ và tên | Công việc |
|---|---|---|
| 22028190 | Lê Minh Quý (Leader) | Frontend, Thuyết trình, làm báo cáo phần System Design và Deployment. |
| 22028292 | Đào Thị Thu Hường | Làm báo cáo phần Introduction, Requirements Specification |
| 22028132 | Hoàng Linh Chi | Làm slide, báo cáo phần Requirements Specification. |
| 22028282 | Nguyễn Thu Hà | Làm báo cáo phần Introduction, Requirements Specification |
| 22028188 | Hoàng Văn Dũng | Backend, làm báo cáo phần Testing và Implementation |

## Table of Contents

# I. INTRODUCTION

## 1. Problem

The human need for mobility is ever-growing, placing significant pressure on the transportation systems in major cities. Owning a personal car is costly and adds to the parking woes. Therefore, car rental services are becoming increasingly popular, especially in the era of Industry 4.0.

Industry 4.0, with the assistance of the Internet of Things (IoT) and Big Data, offers comprehensive solutions for every industry, including car rental services. Thanks to technology, car rental businesses can manage and control operations more efficiently while enhancing customer experience.

However, managing vast amounts of information in the car rental industry poses a significant challenge for businesses, including large corporations. Recognizing this, our team has decided to develop a "Car Rental Management Website" to address the difficulties in managing and operating car rental services, while providing a fast and convenient rental experience for customers.

## 2. Project Goals

Our goal is to optimize the car rental experience by leveraging advanced technology. Through the development of the Car Rental Management Website, we aim to alleviate the burden of managing large volumes of information in this industry. Our platform provides efficient management tools, from scheduling to tracking vehicle maintenance services. This helps car rental businesses optimize their operations and provide better services to customers. We are committed to driving development and progress in the industry by integrating technology with the aim of providing the most comprehensive and convenient solution for all stakeholders.

## 3. Project Scope

This report focuses on analyzing the problem, completing a requirements analysis, as well as designing the interface, database, and programming basic functionalities to develop the management product. Currently, the problem is limited to payment using various forms to simplify the system for employee use, minimizing errors, and optimizing workflow.

### 4. System overview

The Car Rental Management Website provides an extensive system that is crafted to streamline every aspect of car rental operations. At its core, the system includes various modules that seamlessly work together to facilitate efficient management and control.

## II. REQUIREMENTS SPECIFICATION

### 1. Actors

- Customer: Reserve, pick up, and return vehicles through the website. They can also search and compare different types of vehicles, make payments, and manage their personal information
- Employee: Perform tasks related to customer support and vehicle management. Their responsibilities include assisting customers with booking, addressing inquiries, facilitating vehicle handovers, and inspecting vehicle conditions before and after rentals
- Admin: Manage customer and employee information, vehicle data, and transactions on the website. They ensure the system operates smoothly and securely.

### 2. Functional requirements

| No. | Group | Function | Description |
|---|---|---|---|
| 1 | Authentication | Register | Guests can create account |
| 2 | | Login | When register successfully, users can login to their |

|  |  |  |  |
|---|---|---|---|
|  |  |  | account |
| 3 |  | Forgot password | Users could reset password if forgot |
| 4 | Home | Rent car | Users can rent cars in the product page |
| 5 |  | Read blog | Users can read blogs about cars in the blog page |
| 6 |  | Contact | Users can contact the administrator via phone, email |
| 7 | Account | Edit information | Account owners can edit their information |
| 8 |  | View information | Account owners can view their information |
| 9 |  | Change password | Account owners can change their password |
| 10 | Customer | Book car | Customers can book cars |
| 11 |  | Return car | Customers can return cars |
| 12 |  | View car information | Customers can view information about car rental |
| 13 | Employee | Manage car | Employees can add, delete, and edit information about rental cars |
| 14 |  | View booking data information | Employees can view information about car rental orders |

|  |  |  |  |
|---|---|---|---|
| 15 |  | View customer information | Employees can view information about customer |
| 16 |  | Manage contract | Employees can manage contracts |
| 17 | Admin | Manage employee | Admin can manage employees |
| 18 |  | Manage customer | Admin can manage customers |
| 19 |  | Resolve problems | Admin can resolve problems |

### 3. Non – Functional requirements

| STT | Category | Non – Functional requirements |
|---|---|---|
| 1 | Usability | The system should be easy to use and accessible Practical: System UI is designed with ReactJS, which can create awesome and friendly UI |
| 2 | Reliability | The system should be reliable and available at all times Practical:<br>- Hosting with Vercel, one of the most reliable popular web development hosting in 2024<br>- Able to roll back to previous versions if necessary |
| 3 | Performance | The system should respond quickly and be able to handle a large number of users Practical:<br>- ReactJS provides performance and fast rendering<br>- NodeJS handle concurrent requests |

|  |  |  |
|---|---|---|
|  |  | - MySQL stores data efficiently<br>- ExpressJS manages routing for optimized system performance |
| 4 | Scalability | The system should be able to handle increasing amounts of data and users over time Practical: Use NodeJS Cluster, able to create child processes (workers) that run simultaneously to handle multiple requests |
| 5 | Security | The system should protect user data and resources against unauthorized access and other security threats Practical:<br>- Sanitizing: Filter data before saving to database, can prevent XSS attack<br>- Argon2: More secure in password hashing<br>- Cloudflare: Prevent DDOS attack |
| 6 | Maintainabilit y | The system should be easy to maintain and upgrade over time Practical:<br>- Version control: Git, a platform to manage system source code<br>- Modules structure: Source code is built into modules, which is easy to follow up and maintainable<br>- Standard rules and formatting: Uniform code styles between contributors |
| 7 | Compatibility | The system should be compatible with different platforms, browsers, and devices Practical: Responsive design with CSS media query |
| 8 | Legal | All activities must comply with the law |

## 4. Common constraints

### 4.1. Interface constraints

- Guests can view car information, see blog’s posts, search for experts

- Otherwise, only authenticated users can rent car

- Only verified employees can view and edit contracts.

### 4.2. Server constraints

- Resources: Inaccessible from other domain (CORS policy protection)

- Availability: Server must be available at all time (auto reverse to previous checkpoint if current version is not available)

## 5. Business analysis

### 5.1. Business Analysis Technique

### 5.1.1. Core Concept Model in BACCM$^{TM}$ Business Analysis

The core model in BACCM$^{TM}$ business analysis is a conceptual framework for business analysis. It includes what business analysis is and its significance for business analysts regardless of perspective, industry, methodology, or organizational level. It consists of 6 terms with common meanings for all business analysts and helps them discuss both Business Analysis and its relationships to common terms. Each of these terms is considered a core concept.

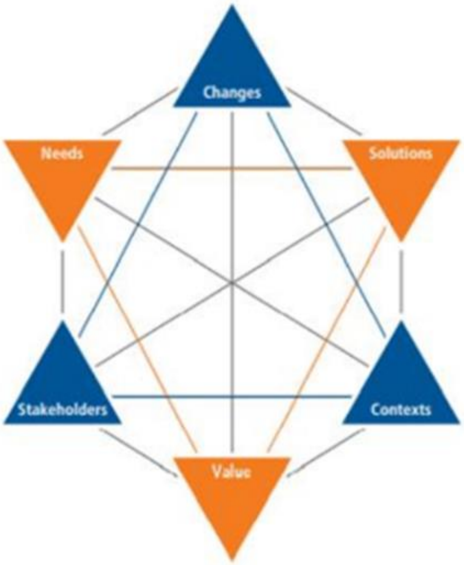


Figure 1: BACCM$^{TM}$

BACCM$^{TM}$ can be used to:

- Describe business and the scope of business analysis
- Communicate within business analysis using common terminology
- Evaluate the relationships of key concepts in business analysis
- Enhance business analysis by assessing the overall relationship of the 6 concepts
- Assess the impact of concepts and their relationships at all stages of the work process to establish a foundation and path.

| Core concept | Describe concepts |
|---|---|
| Change | The act of modifying or altering something in response a need. |
| Need | A requirement or desire that necessitates attention or action. |
| Solution | A specific approach or method used to address one or ore needs within a given context. |
| Stakeholder | A group or individual who has an interest or volvement in the change, need, or solution. |
| Value | The significance or benefit provided by a factor in lation to stakeholders within a particular context. It can compass potential or actual gains, accomplishments, provements, or even devaluation in the form of losses, sks, or costs. |

| | |
|---|---|
| Context | The circumstances, conditions, or environment that fluences or is influenced by the change. It provides nderstanding and perspective on the nature and impact of e change. |

### 5.1.2. Business Analysis Knowledge Area

The Business Analysis Knowledge Area represents specialized domains within the field of business analysis. It includes the following knowledge areas:

Business Analysis Planning and Monitoring: This describes the tasks performed by business analysts to organize and coordinate the efforts of business analysts and stakeholders. It involves planning and monitoring the activities related to business analysis.

Elicitation and Collaboration: This describes the tasks performed by business analysts to prepare and conduct elicitation activities to gather and confirm requirements. It also includes communication with stakeholders after the business analysis information has been collected and ongoing collaboration with them throughout the business analysis activities.

Requirements Life Cycle Management: This describes the tasks performed by business analysts to manage and maintain requirements and design information from inception to retirement. These tasks involve establishing meaningful relationships between related requirements and designs, as well as evaluating, analyzing, and achieving consensus on proposed changes to requirements and designs.

Strategy Analysis: This describes the work that business analysts must perform to collaborate with stakeholders in identifying strategic or tactical needs (business needs) that enable the enterprise to respond to those needs and adjust resulting strategies with higher-level and lower-level strategies.

Requirements Analysis and Design Definition: This describes the tasks performed by business analysts to structure and organize requirements discovered in elicitation activities, identify and model requirements and designs, validate and verify information, identify solution options to meet business needs, and estimate the potential value that can be realized for each solution option. This knowledge area includes iterative and incremental activities, ranging from initial concept and need exploration to transforming those needs into a proposed solution.

Solution Evaluation: This describes the tasks performed by business analysts to assess the performance and value delivered by a solution that is being used by the enterprise and propose the removal of barriers or constraints that prevent the full realization of value.

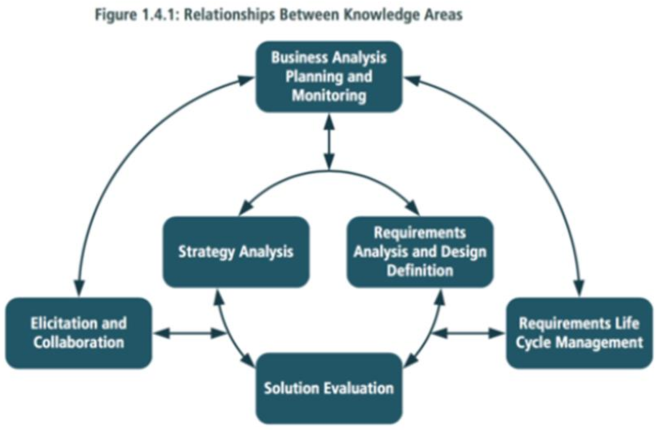


Figure 1.4.1: Relationships Between Knowledge Areas
Business Analysis Planning and Monitoring
Strategy Analysis
Requirements Analysis and Design Definition
Elicitation and Collaboration
Requirements Life Cycle Management
Solution Evaluation

Figure 2: Relationships Between Knowledge Areas

### 5.1.3. Requirements Elicitation Techniques

Requirements elicitation is the process of gathering information from stakeholders to understand their needs and identify potential solutions that can meet those needs. In most cases, multiple techniques are used in requirements elicitation activities, and selecting the right techniques and ensuring each technique is performed accurately is crucial for the success of the elicitation process.

Some commonly used techniques include:

- Interviews: Asking questions to stakeholders to explore their needs, identify issues or opportunities.

- Document Analysis: Used to review systems, contracts, business procedures, and existing policies, standards, and regulations.

- Surveys or Questionnaires: Used to collect business analysis information, including information about customers, products, workflows, and attitudes, from a group of people in a structured manner and within a relatively short time frame.

- Workshops: Used to gather business analysis information, including information about customers, products, workflows, and attitudes, from a group of people in a collaborative manner, creating a conducive environment for discussion and idea generation.

## 5.2. Identifying the Business Model in the Problem

### 5.2.1. Core Concept Model in BACCM$^{TM}$ Business Analysis

The core concept model in business analysis is determined as follows:

- Needs: Car rental is becoming increasingly popular, yet many companies in this business still manage manually without a management system.

- Solutions: Building a website, an application to allow customers to view information about cars, rental costs anytime, anywhere, conveniently, and quickly. Especially, developing a rental management system to support businesses in managing their operations.

- Changes: Companies need to change their structures and working processes to be more convenient.

- Stakeholders include: Businesses, car rental companies, employees, accountants, managers, customers, end-users.

- Value for businesses: Increase in revenue, brand recognition, revenue management support, error reduction. For customers: Increased service usage, customer retention, time-saving, convenience.

- Contexts: Users or customers increasingly prefer convenience, time-saving, and the application of information technology is becoming more common.

### 5.2.2.Identifying the Stakeholders

List of stakeholders in the system: Board of Directors, Accounting Department, Customer Service Representatives, Vehicle Maintenance Staff, Customers.

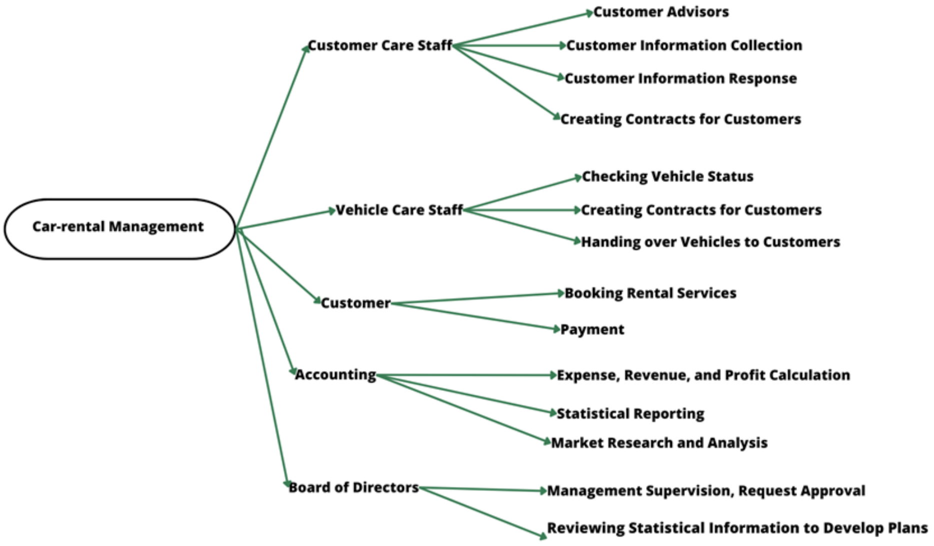
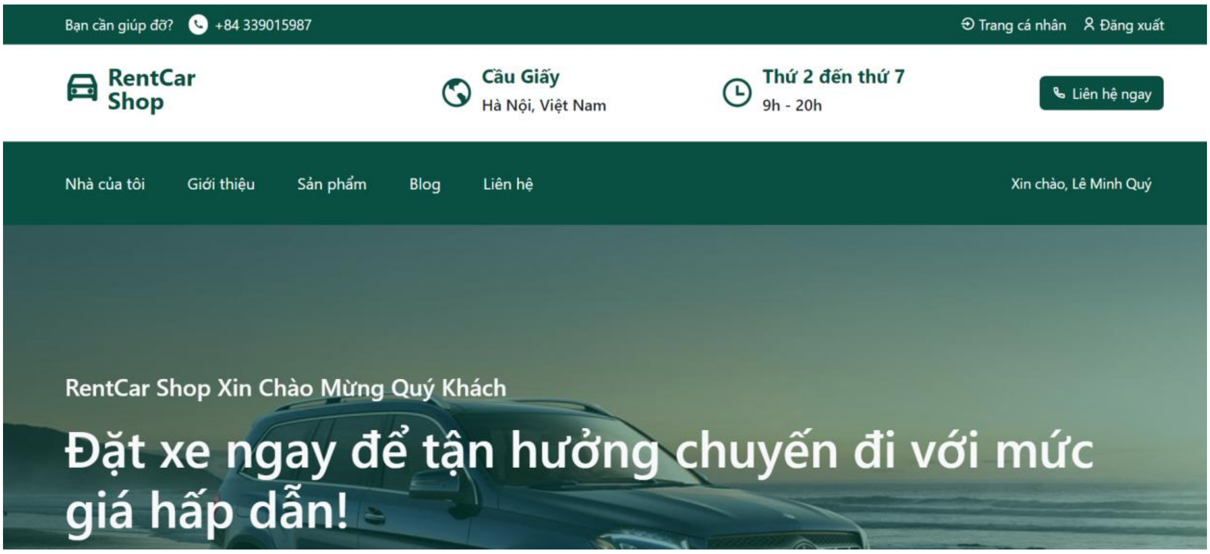

Car-rental Management
Customer Care Staff
Customer Advisors
Customer Information Collection
Customer Information Response
Creating Contracts for Customers
Vehicle Care Staff
Checking Vehicle Status
Creating Contracts for Customers
Handing over Vehicles to Customers
Customer
Booking Rental Services
Payment
Accounting
Expense, Revenue, and Profit Calculation
Statistical Reporting
Market Research and Analysis
Board of Directors
Management Supervision, Request Approval
Reviewing Statistical Information to Develop Plans

Figure 3: Mindmap stakeholder

## 6. System Prototype

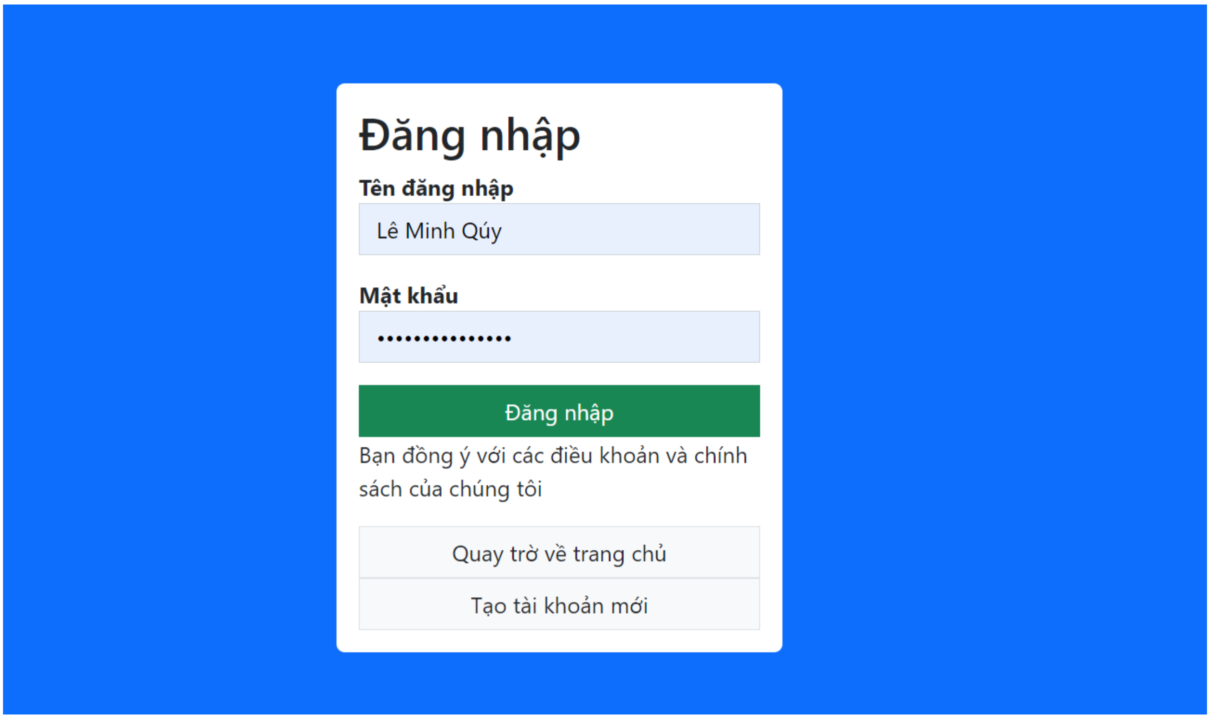
### 6.1. Home



RentCar Shop Xin Chào Mừng Quý Khách

Đặt xe ngay để tận hưởng chuyến đi với mức giá hấp dẫn!

### 6.2. Login form



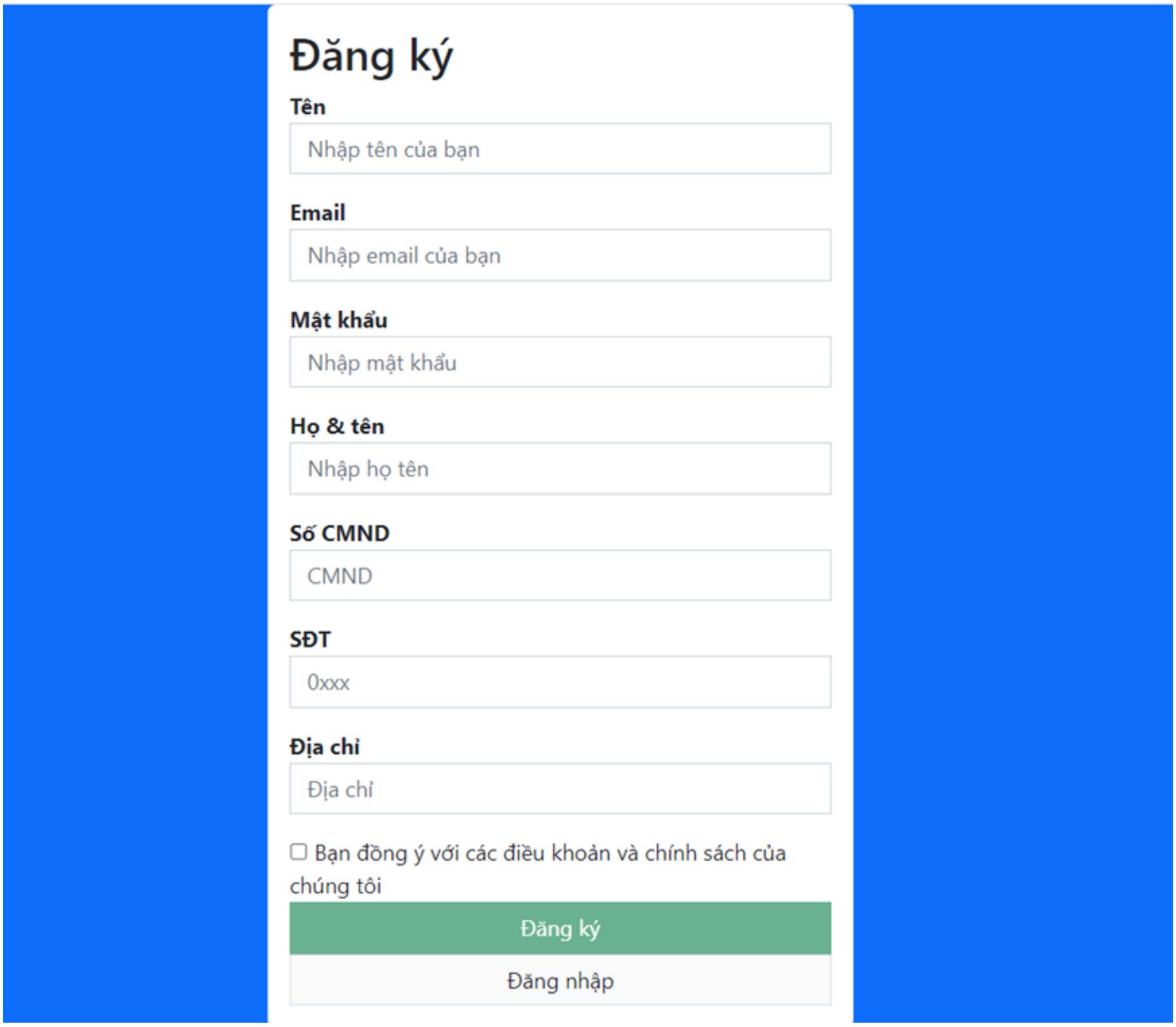
Đăng nhập

Tên đăng nhập: Lê Minh Qúy

Mật khẩu: •••••••••••••••

Bạn đồng ý với các điều khoản và chính sách của chúng tôi

## 6.3. Register form



Đăng ký

Tên

Nhập tên của bạn

Email

Nhập email của bạn

Mật khẩu

Nhập mật khẩu

Họ & tên

Nhập họ tên

Số CMND

CMND

SĐT

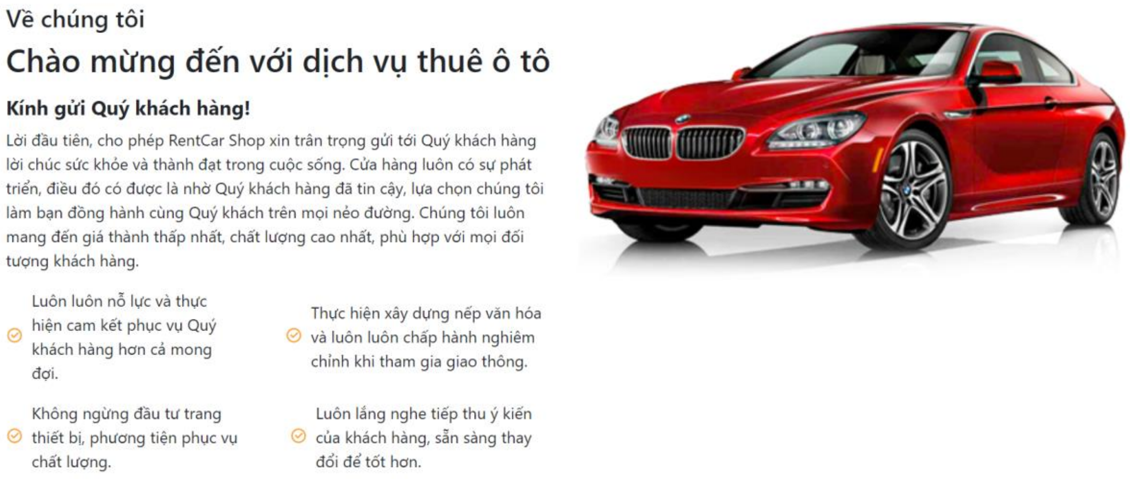
0xxx

Địa chỉ

Địa chỉ

□ Bạn đồng ý với các điều khoản và chính sách của chúng tôi

## 6.4. Introduce



Về chúng tôi

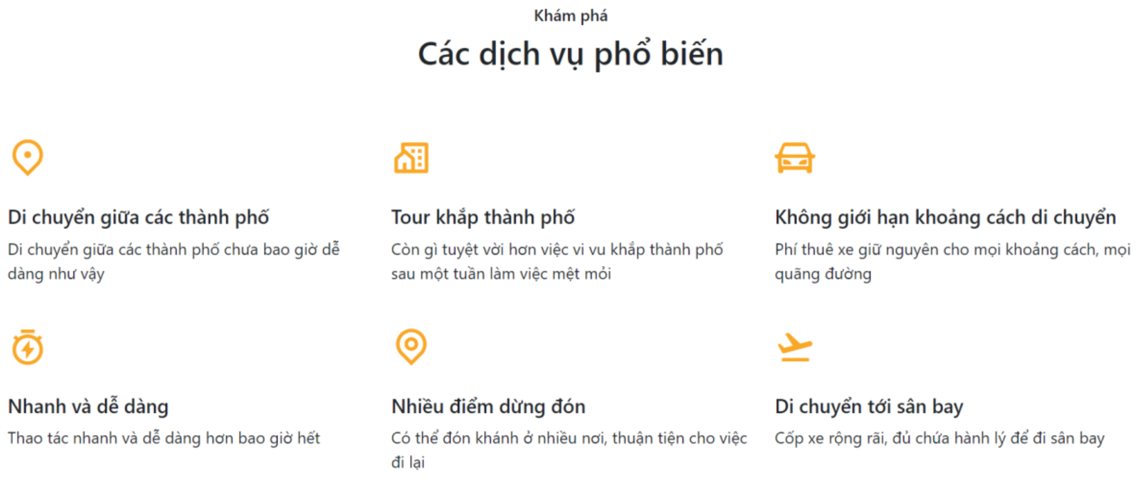
Chào mừng đến với dịch vụ thuê ô tô

**Kính gửi Quý khách hàng!**

Lời đầu tiên, cho phép RentCar Shop xin trân trọng gửi tới Quý khách hàng lời chúc sức khỏe và thành đạt trong cuộc sống. Cửa hàng luôn có sự phát triển, điều đó có được là nhờ Quý khách hàng đã tin cậy, lựa chọn chúng tôi làm bạn đồng hành cùng Quý khách trên mọi nẻo đường. Chúng tôi luôn mang đến giá thành thấp nhất, chất lượng cao nhất, phù hợp với mọi đối tượng khách hàng.

- ✓ Luôn luôn nỗ lực và thực hiện cam kết phục vụ Quý khách hàng hơn cả mong đợi.
- ✓ Thực hiện xây dựng nếp văn hóa và luôn luôn chấp hành nghiêm chỉnh khi tham gia giao thông.
- ✓ Không ngừng đầu tư trang thiết bị, phương tiện phục vụ chất lượng.
- ✓ Luôn lắng nghe tiếp thu ý kiến của khách hàng, sẵn sàng thay đổi để tốt hơn.

## 6.5. Service provided



Khám phá

Các dịch vụ phổ biến

**Di chuyển giữa các thành phố**
Di chuyển giữa các thành phố chưa bao giờ dễ dàng như vậy

**Tour khắp thành phố**
Còn gì tuyệt vời hơn việc vi vu khắp thành phố sau một tuần làm việc mệt mỏi

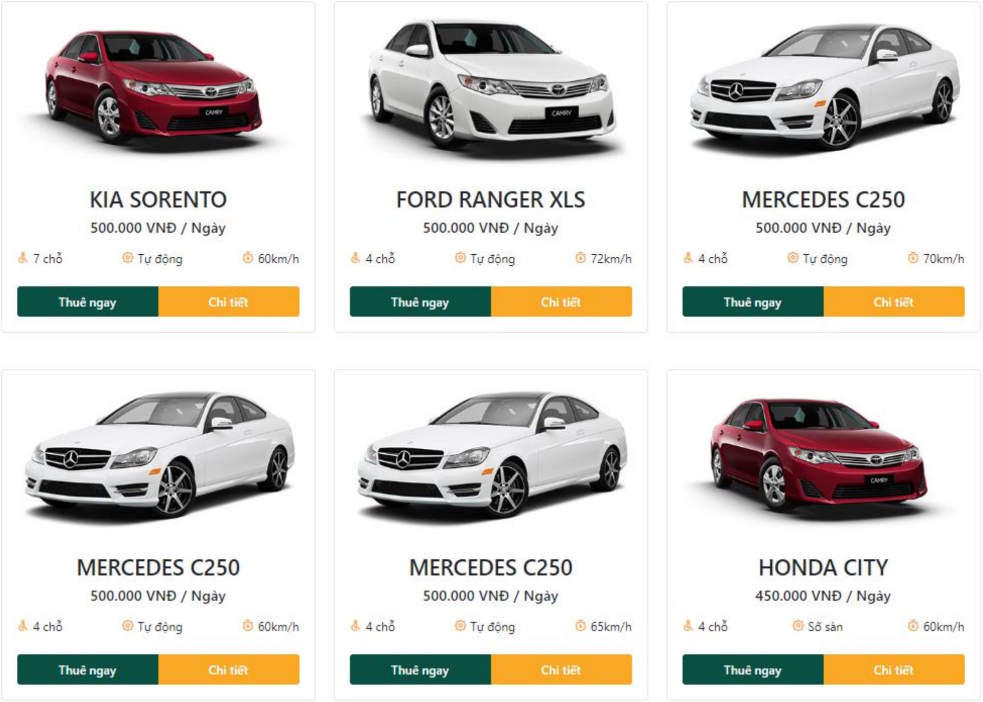
**Không giới hạn khoảng cách di chuyển**
Phí thuê xe giữ nguyên cho mọi khoảng cách, mọi quãng đường

**Nhanh và dễ dàng**
Thao tác nhanh và dễ dàng hơn bao giờ hết

**Nhiều điểm dừng đón**
Có thể đón khánh ở nhiều nơi, thuận tiện cho việc đi lại

**Di chuyển tới sân bay**
Cốp xe rộng rãi, đủ chứa hành lý để đi sân bay

## 6.6. Products



KIA SORENTO
500.000 VNĐ / Ngày
7 chỗ · Tự động · 60km/h
Thuê ngay · Chi tiết

FORD RANGER XLS
500.000 VNĐ / Ngày
4 chỗ · Tự động · 72km/h
Thuê ngay · Chi tiết

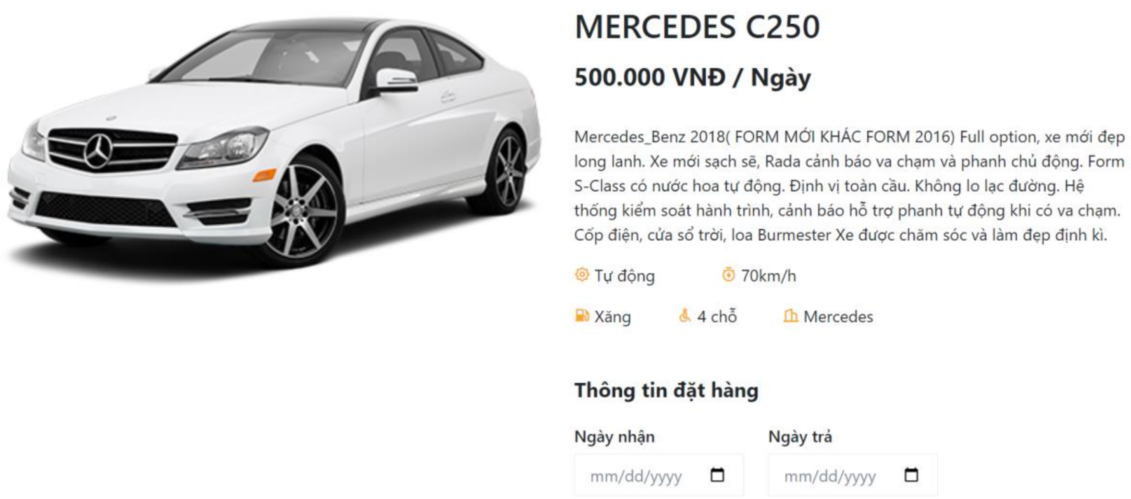
MERCEDES C250
500.000 VNĐ / Ngày
4 chỗ · Tự động · 70km/h
Thuê ngay · Chi tiết

MERCEDES C250
500.000 VNĐ / Ngày
4 chỗ · Tự động · 60km/h
Thuê ngay · Chi tiết

MERCEDES C250
500.000 VNĐ / Ngày
4 chỗ · Tự động · 65km/h
Thuê ngay · Chi tiết

HONDA CITY
450.000 VNĐ / Ngày
4 chỗ · Số sàn · 60km/h
Thuê ngay · Chi tiết

## 6.7. Products information



MERCEDES C250

500.000 VNĐ / Ngày

Mercedes_Benz 2018( FORM MỚI KHÁC FORM 2016) Full option, xe mới đẹp long lanh. Xe mới sạch sẽ, Rada cảnh báo va chạm và phanh chủ động. Form S-Class có nước hoa tự động. Định vị toàn cầu. Không lo lạc đường. Hệ thống kiểm soát hành trình, cảnh báo hỗ trợ phanh tự động khi có va chạm. Cốp điện, cửa sổ trời, loa Burmester Xe được chăm sóc và làm đẹp định kì.

Tự động 70km/h

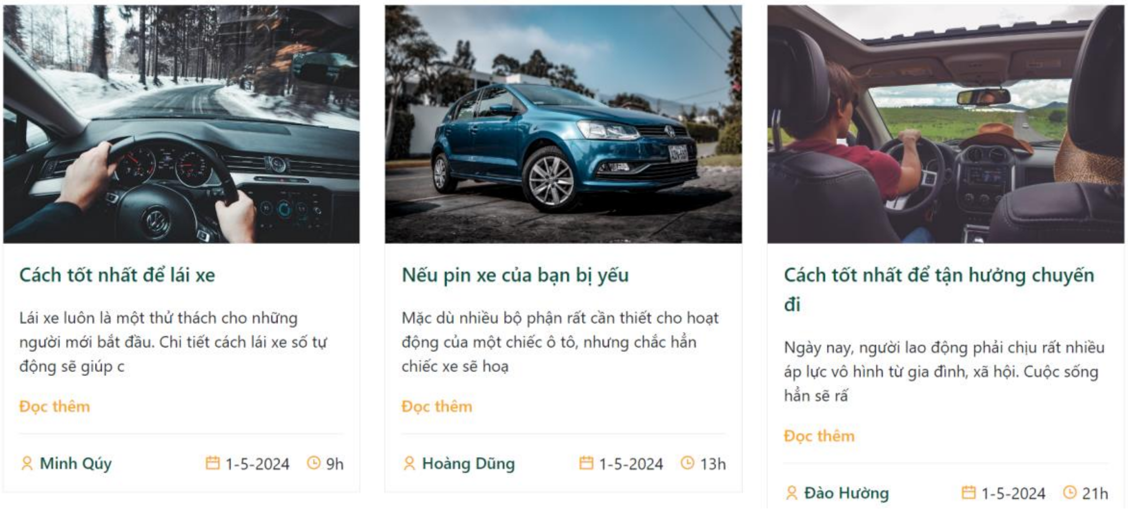
Xăng 4 chỗ Mercedes

Thông tin đặt hàng

Ngày nhận: mm/dd/yyyy

Ngày trả: mm/dd/yyyy

## 6.8. Blog



Cách tốt nhất để lái xe

Lái xe luôn là một thử thách cho những người mới bắt đầu. Chi tiết cách lái xe số tự động sẽ giúp c

Đọc thêm

Minh Qúy 1-5-2024 9h

Nếu pin xe của bạn bị yếu

Mặc dù nhiều bộ phận rất cần thiết cho hoạt động của một chiếc ô tô, nhưng chắc hẳn chiếc xe sẽ hoạ

Đọc thêm

Hoàng Dũng 1-5-2024 13h

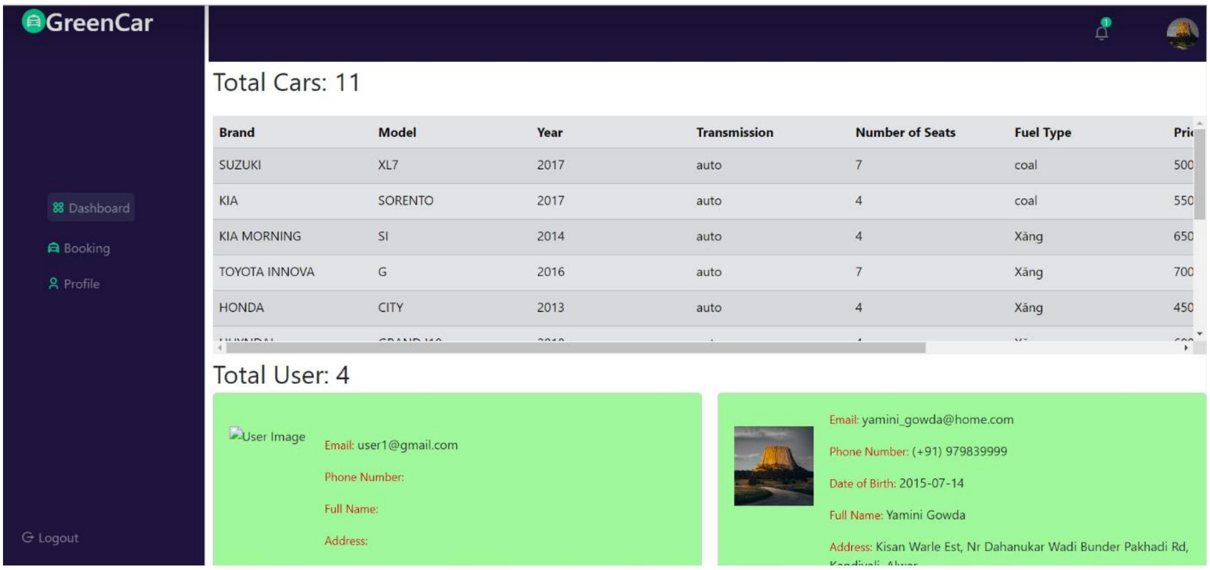
Cách tốt nhất để tận hưởng chuyến đi

Ngày nay, người lao động phải chịu rất nhiều áp lực vô hình từ gia đình, xã hội. Cuộc sống hẳn sẽ rấ

Đọc thêm

Đào Hường 1-5-2024 21h

## 6.9. Employee



Total Cars: 11

| Brand | Model | Year | Transmission | Number of Seats | Fuel Type | Pri |
|---|---|---|---|---|---|---|
| SUZUKI | XL7 | 2017 | auto | 7 | coal | 500 |
| KIA | SORENTO | 2017 | auto | 4 | coal | 550 |
| KIA MORNING | SI | 2014 | auto | 4 | Xăng | 650 |
| TOYOTA INNOVA | G | 2016 | auto | 7 | Xăng | 700 |
| HONDA | CITY | 2013 | auto | 4 | Xăng | 450 |

Total User: 4

User Image

Email: user1@gmail.com

Phone Number:

Full Name:

Address:

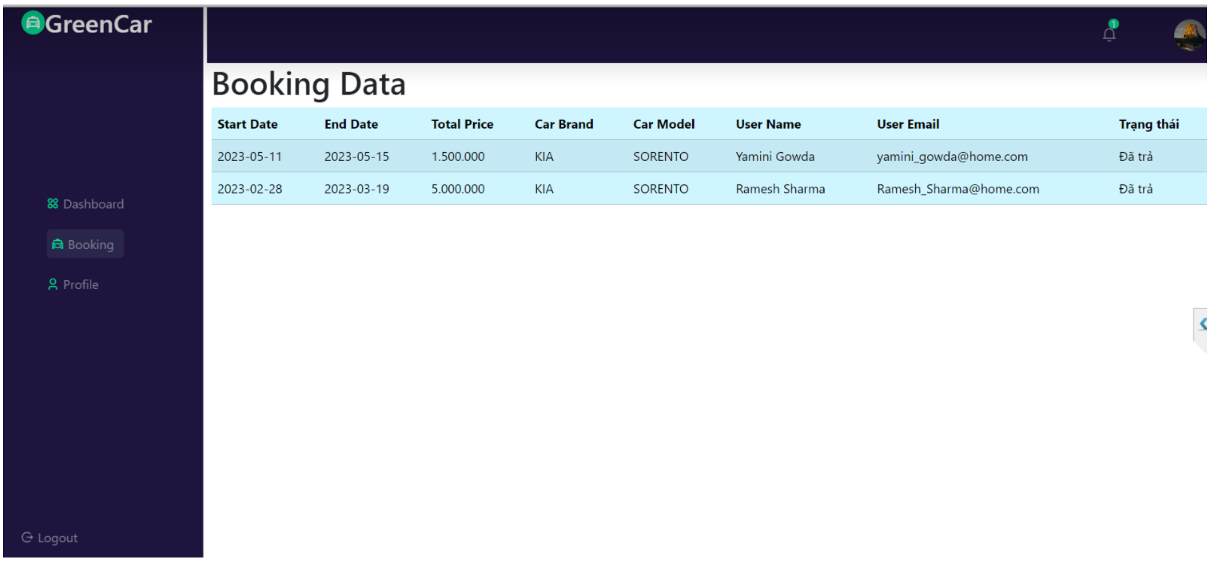
Email: yamini_gowda@home.com

Phone Number: (+91) 979839999

Date of Birth: 2015-07-14

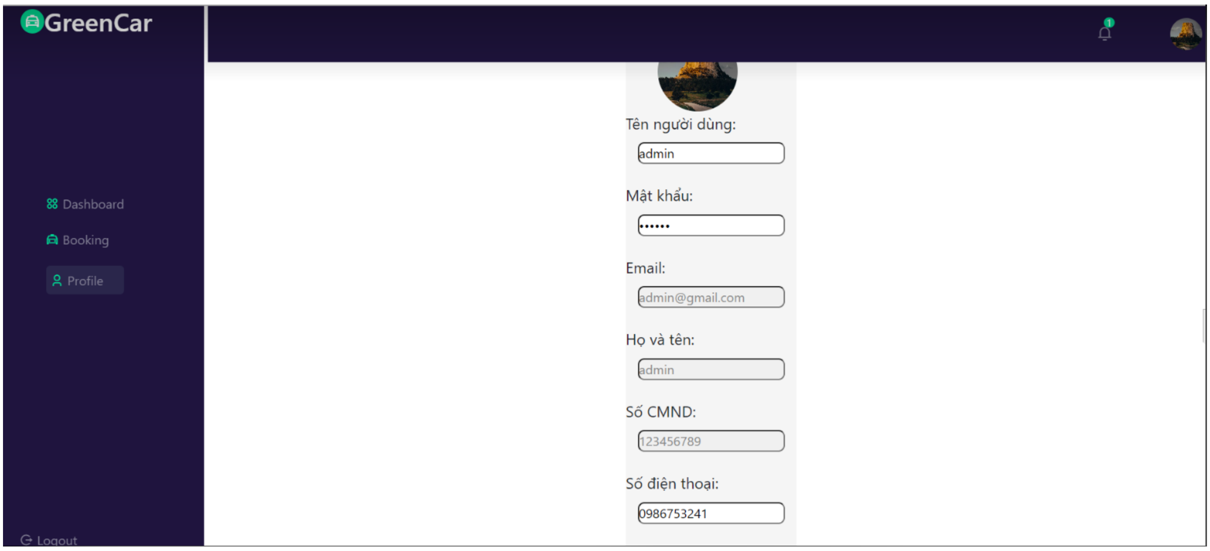
Full Name: Yamini Gowda

Address: Kisan Warle Est, Nr Dahanukar Wadi Bunder Pakhadi Rd,



Booking Data

| Start Date | End Date | Total Price | Car Brand | Car Model | User Name | User Email | Trạng thái |
|---|---|---|---|---|---|---|---|
| 2023-05-11 | 2023-05-15 | 1.500.000 | KIA | SORENTO | Yamini Gowda | yamini_gowda@home.com | Đã trả |
| 2023-02-28 | 2023-03-19 | 5.000.000 | KIA | SORENTO | Ramesh Sharma | Ramesh_Sharma@home.com | Đã trả |



Tên người dùng: admin

Mật khẩu: ••••••

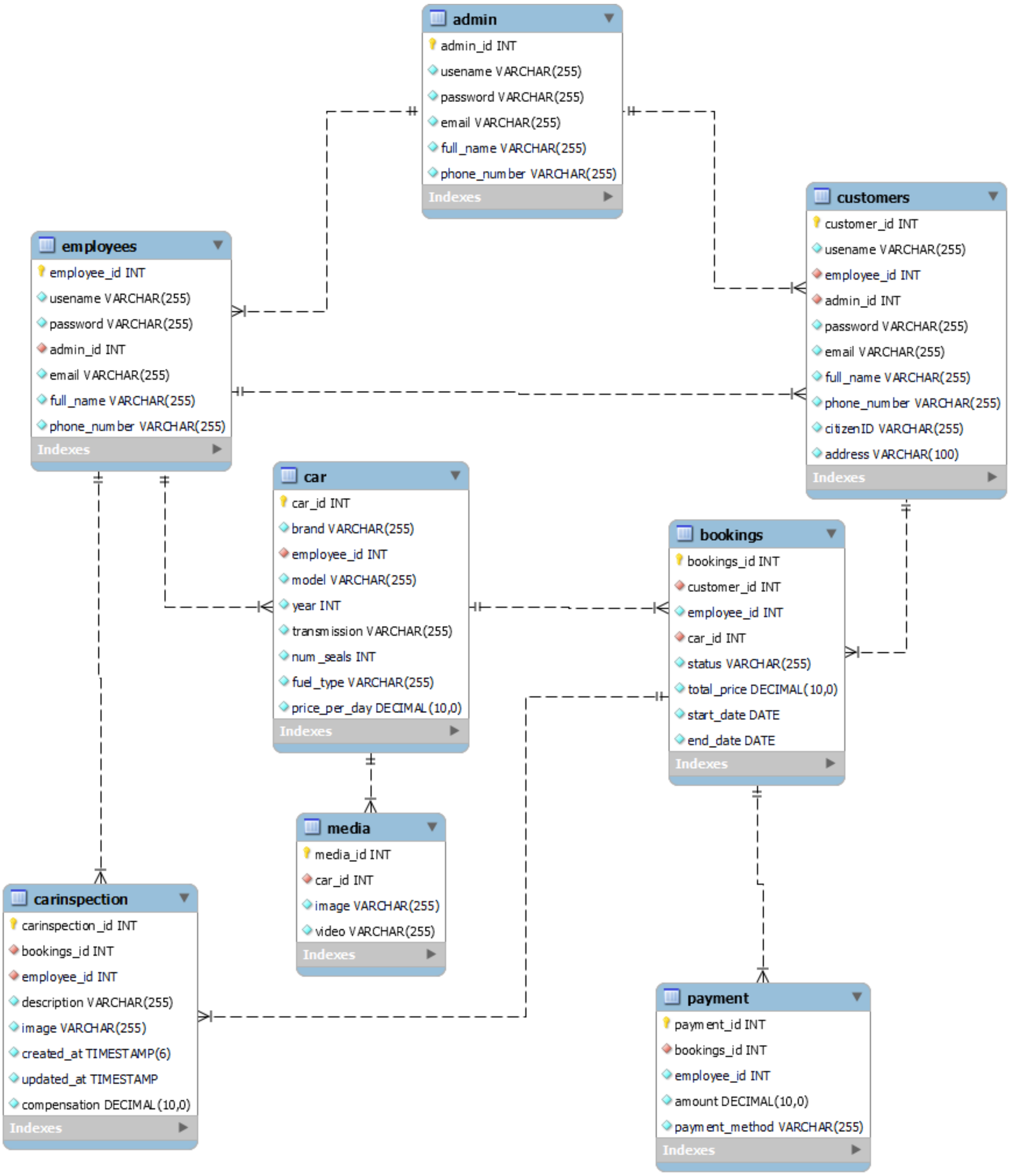
Email: admin@gmail.com

Họ và tên: admin

Số CMND: 123456789

Số điện thoại: 0986753241

## 7. Database Schemas



## 8. UML Use Case Diagram

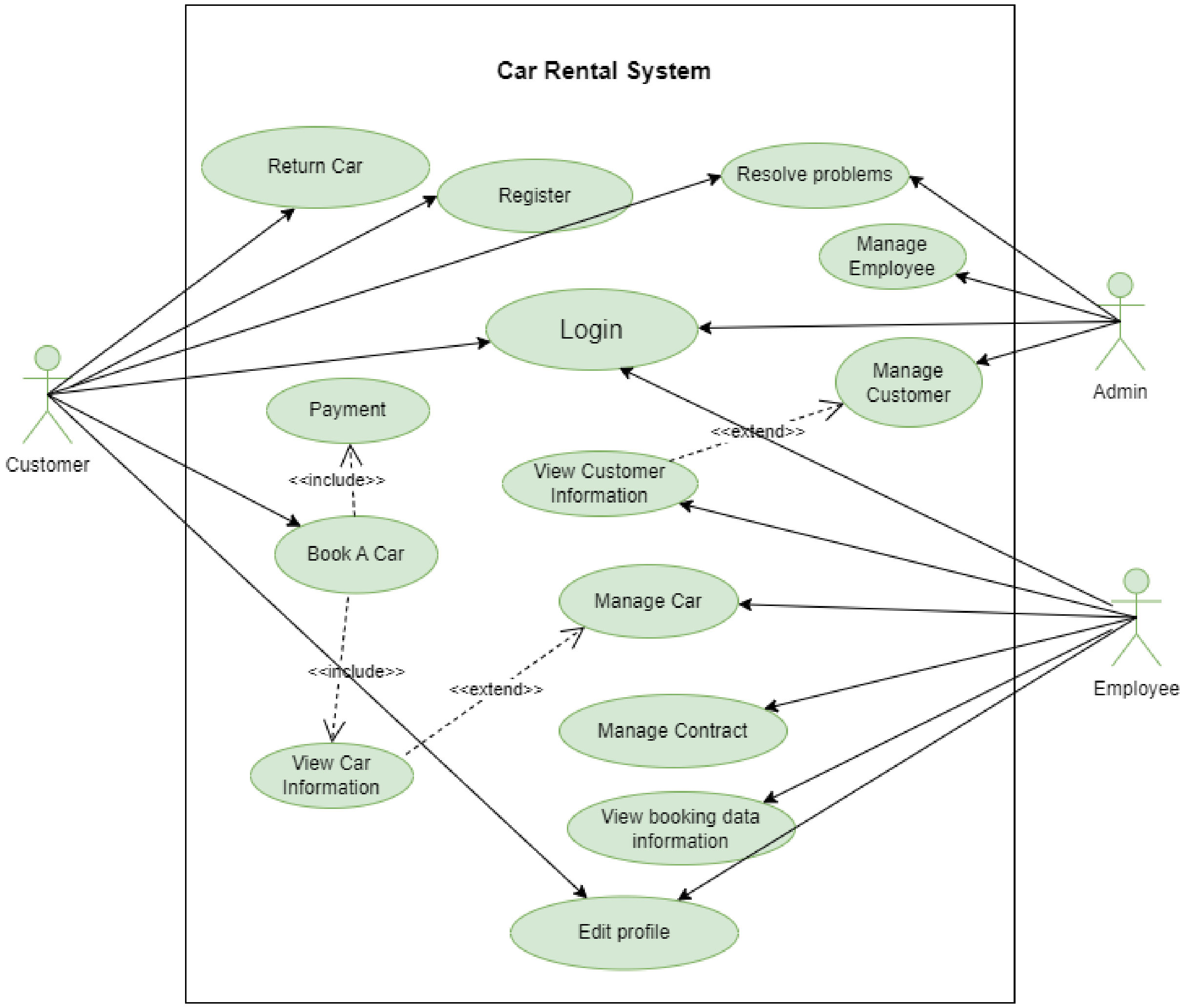


Figure 5: Overall Use Case Diagram

## 9. UML Activity Diagram

### 9.1. Authentication Functionality

| | |
|---|---|
| Use Case name | Register |
| Actor | User |
| Pre – Condition | System connection |

| | |
|---|---|
| Post - Condition | 1. Registered users must be recorded<br>2. The registration was audited in the System log file<br>3. Generate access token and set to client browser cookie |
| Trigger | 1. Register with missing some fields<br>2. Duplicate some unique fields |
| Main Event | Register successfully |
| Extra Event | 1. Register with missing some fields<br>2. Duplicate some unique fields |

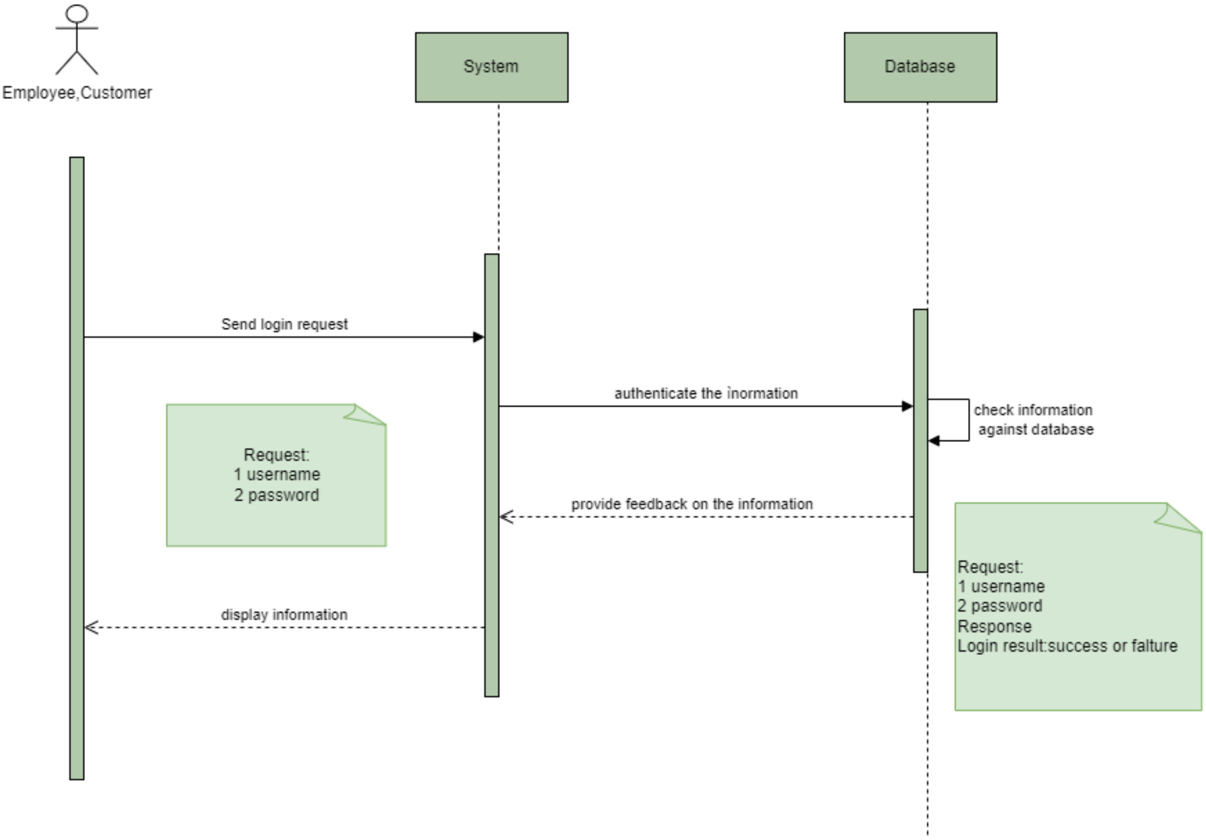


Figure Sequence Diagram for Login Functionality

| | |
|---|---|
| Usecase name | Login |
| Actor | Employee, Customer |
| Pre-Condition | N/A |
| Minimum Guarantee | The employee, customer must have an account to log in to the system |
| Success Guarantee | Successful login |
| Trigger | Login Button |
| Main Event | 1. The system displays a login window, consisting of username and password fields.<br>2. Enter login information, then click the login button.<br>3. The system checks the login account information.<br>4. Respond with the login result, accessing the system. |
| Extra Event | ● If the account information is incorrect or does not exist: The system displays "Login unsuccessful, incorrect username or password" and prompts for re-entry. |

## 9.2. Employees

### 9.2.1. View customer’s information

| | |
|---|---|
| Use Case name | View Customer Information |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | The employee can view customer information on the system. |
| Trigger | Select the function to view customer information. |
| Main Event | 1. The system displays a window allowing the employee to view a list of customers.<br>2. The employee selects the customer whose information they want to view in detail.<br>3. The system displays the selected customer's information.<br>4. Exit the customer information viewing function. |

| | |
|---|---|
| Extra Event | N/A |

## 9.2.2. Manage Car

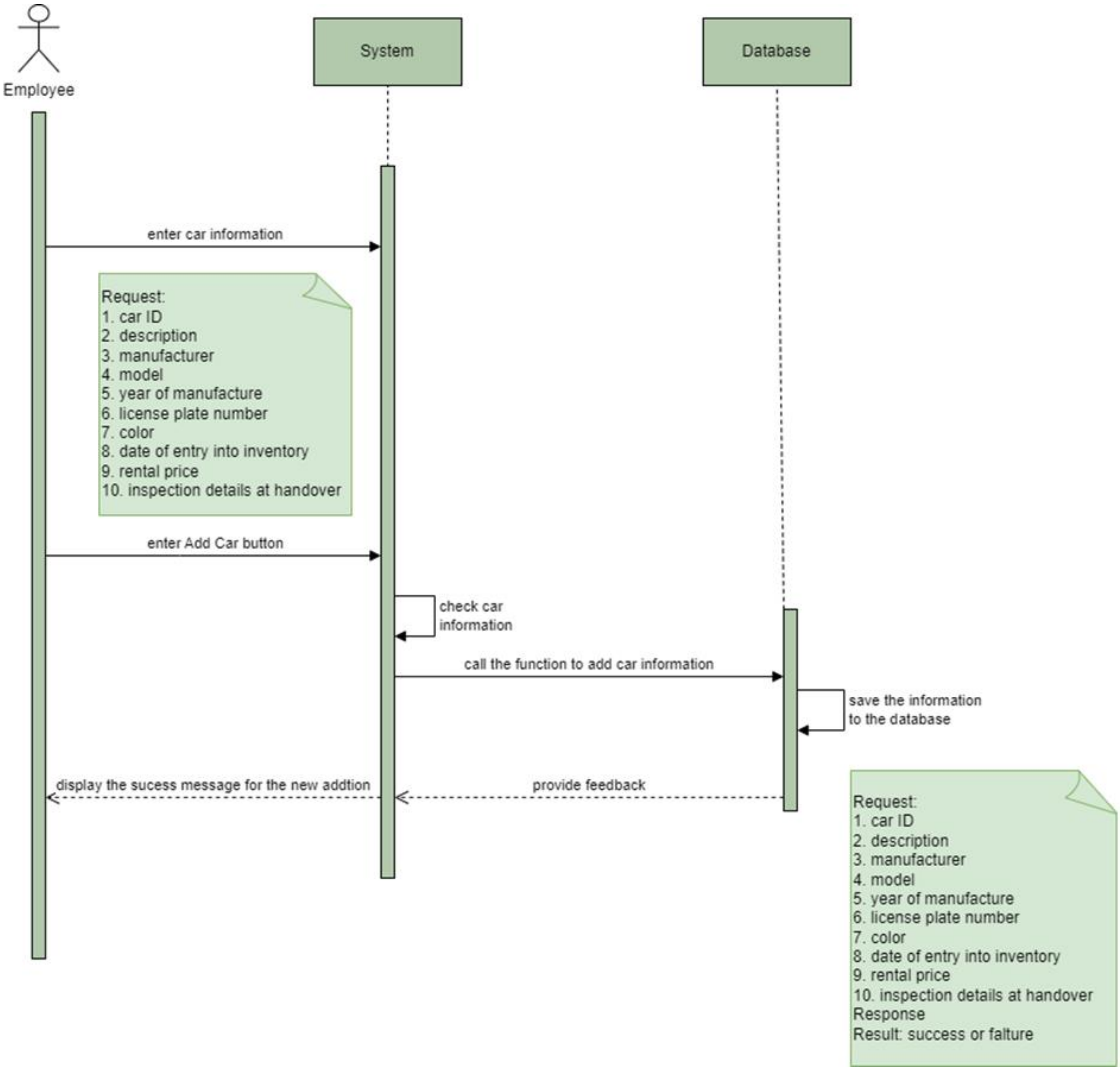


Figure : Sequence Diagram for Adding Vehicle Information

- Description of viewing vehicle information functionality

| | |
|---|---|
| Usecase name | View Vehicle Information |
| Actor | Employee |
| Description | This use case allows the user to view information about the types of vehicles currently in the system. |
| Pre - Condition | Logged in and navigated to the "Vehicle Management -> View Vehicle Information" function. |
| Post-Condition | Display a list of vehicles in the system. |
| Trigger | Execute the "Vehicle Management -> View Vehicle Information" function. |
| Main Event | 1. The system displays a list of available vehicle types.<br>2. Select a specific type of vehicle.<br>3. The system shows detailed information about that type of vehicle. |
| Extra Event | N/A |

- Description of updating vehicle information functionality

| | |
|---|---|
| Usecase name | Update Vehicle Information |
| Actor | Employee |
| Pre - Conditions | The employee must have successfully logged into the system. |
| Post - Conditions | The edited information is successfully saved into the database. |
| Trigger | Select the function "Vehicle Management -> Update Information". |
| Main Event | 1. The system displays a window allowing input of the vehicle information to be edited.<br>2. Enter the updated information of the vehicle.<br>3. The newly edited vehicle information is added to the database.<br>4. Exit the function. |
| Extra Event | N/A |

- Description of adding a vehicle functionality

| | |
|---|---|
| Usecase name | Add Vehicle |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | The new vehicle information is successfully added to the database. |
| Trigger | Select the function "Vehicle Management -> Add Vehicle". |
| Main Event | 1. The system displays a window allowing the employee to input information for a new vehicle.<br>2. Enter the information for the new vehicle.<br>3. The system validates the entered information for the new vehicle.<br>4. The new vehicle information is added to the database.<br>5. Exit the add vehicle function. |
| Extra Event | If the entered vehicle information already exists in the system: The system displays a message indicating that the information already exists and asks if the employee wants to continue adding a new entry. |

- Description of deleting a vehicle functionality

| | |
|---|---|
| Usecase name | Delete Vehicle |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | The vehicle information is successfully removed from the system. |
| Trigger | Select the function "Vehicle Management -> Delete Vehicle". |
| Main Event | 1. The system displays a list of vehicles registered in the system.<br>2. Select the vehicle to be deleted.<br>3. The system displays a confirmation message asking if the employee is sure they want to delete the selected vehicle.<br>4. The system deletes the vehicle and confirms that the deletion was successful.<br>5. Exit the delete function. |
| Extra Event | Cancel Option: If the employee chooses to cancel the deletion: The system cancels the deletion request and returns to the previous step. |

- Description of checking vehicle status functionality

| | |
|---|---|
| Usecase name | Check Vehicle Status |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Display the accurate status of the vehicle. |
| Activation | Select the function "Vehicle Management -> Check Vehicle Status". |
| Main Event | 1. Enter the vehicle code/name into the textbox for checking.<br>2. The system displays information about the vehicle, including its status and associated orders (if any).<br>3. Exit the vehicle status check function. |
| Extra Event | N/A |

- Description of updating vehicle status functionality

| | |
|---|---|
| Usecase name | Update Vehicle Status |

| | |
|---|---|
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Display the accurate status of the vehicle. |
| Trigger | Select the function "Vehicle Management -> Update Vehicle Status". |
| Main Event | The system updates the vehicle status based on its activities. |
| Extra Event | N/A |

- Description of viewing lists of available vehicles functionality

| | |
|---|---|
| Usecase name | View List of Available Vehicles |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Display the list of available empty vehicles. |

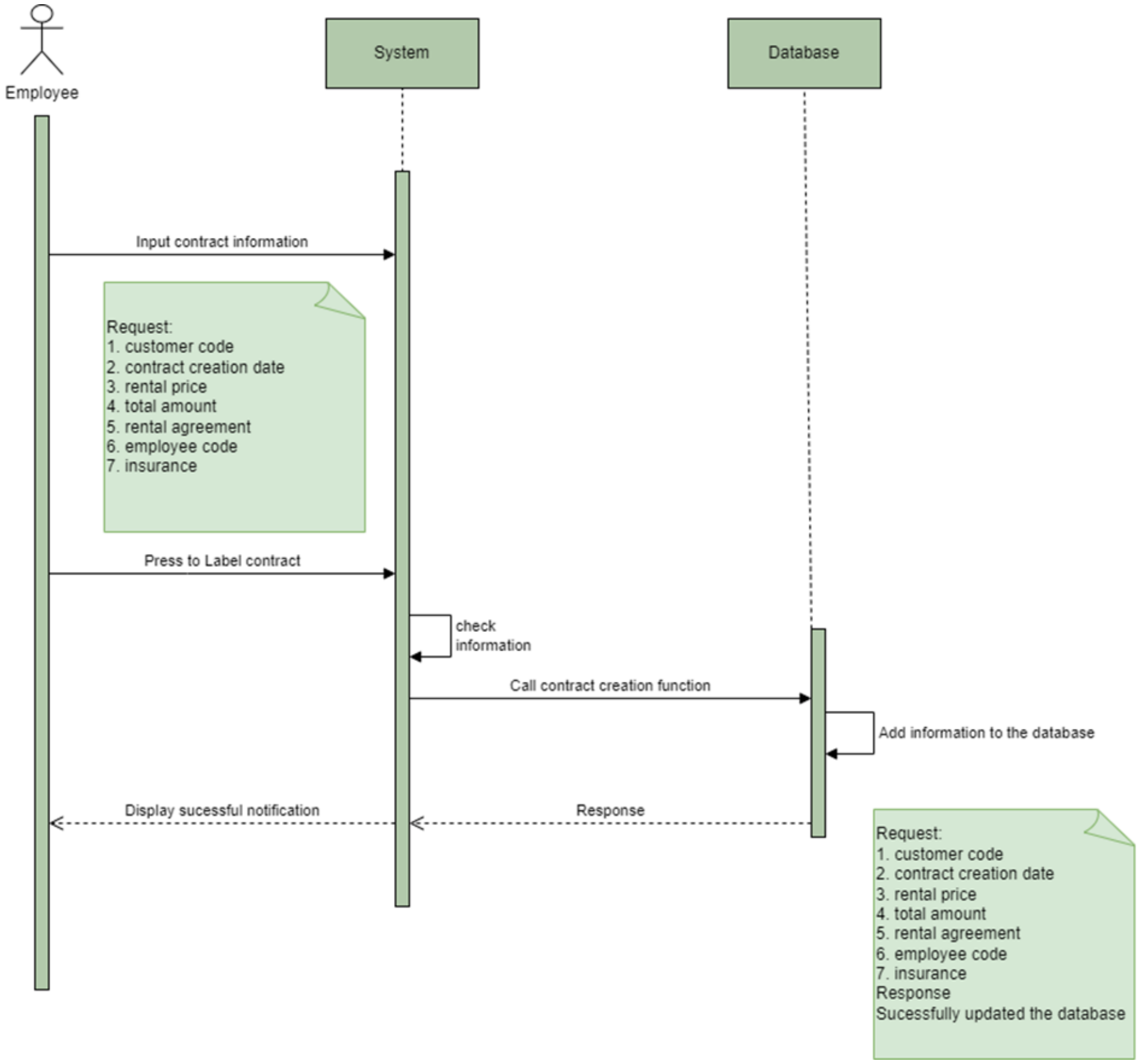
| | |
|---|---|
| Trigger | Select the function "Vehicle Management -> Empty Vehicle List". |
| Main Event | The system displays a list of empty vehicles available. |
| Extra Event | N/A |

- Description of viewing lists of rented vehicles functionality

| Usecase name | View List of Rented Vehicles |
|---|---|
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Display the correct list of rented vehicles. |
| Activation | Select the function "Vehicle Management -> Rented Vehicle List". |
| Main Event | The system displays a list of vehicles that are currently rented. |
| Extra Event | N/A |

### 9.2.3. Manage Contract

Figure : UseCase Diagram for Contract Management



- Description of creating a contract functionality

| | |
|---|---|
| Usecase name | Create Contract |

| | |
|---|---|
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Successfully create a contract. |
| Trigger | Select the function "Print Contract". |
| Main Event | 1. Display contract information on the contract screen.<br>2. Enter information into the fields on the screen.<br>3. Select "Print Contract" and save. |
| Extra Event | N/A |

- Description of canceling a contract functionality

| | |
|---|---|
| Usecase name | Cancel Contract |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |

| | |
|---|---|
| Post - Conditions | Successfully cancel a contract. |
| Trigger | Select the "Cancel Contract" button. |
| Main Event | 1. Navigate to contract management -> Display list of contracts.<br>2. Select the contract to be canceled and click on "Cancel Contract".<br>3. Display confirmation message "Are you sure you want to cancel this contract?".<br>4. Select "Yes" -> The system cancels the contract and confirms successful cancellation. |
| Extra Event | N/A |

- Description of viewing a list of contracts functionality

| | |
|---|---|
| Usecase name | View List of Contracts |
| Actor | Employee |
| Pre - Conditions | The employee has successfully logged into the system. |
| Post - Conditions | Successfully view a list of contracts. |

| | |
|---|---|
| Trigger | Select the "Contract Management" function. |
| Main Event | 1. Choose "Contract Management" from the menu.<br>2. Click on "List of Contracts".<br>3. Display the list of contracts. |
| Extra Event | N/A |

### 9.3. Booking

#### 9.3.1. Contract

| | |
|---|---|
| Usecase name | Contract of booking |
| Actor | Customer |
| Pre - Condition | Customer has selected a car and provided booking details |
| Post - Condition | 1. The customer agrees to the terms and conditions outlined in the rental contract.<br>2. The booking is confirmed, and the contract is generated. |

| | |
|---|---|
| | |
| Minimum Guarantee | The system ensures that all essential contract terms are presented clearly to the customer. |
| Success Guarantee | The customer successfully agrees to the contract terms, and the booking is confirmed with accurate details. |
| Trigger | The completion of the booking details input by the customer and the request to confirm the reservation. |
| Main Event Flow | 1. The car rental system presents the rental contract terms and conditions to the customer.<br>2. The customer reads through the contract terms, including rental duration, rates, insurance coverage, and other relevant policies.<br>3. The customer indicates acceptance of the contract terms by confirming agreement<br>4. Upon agreement, the system generates a formal contract document summarizing the rental details and agreed terms<br>5. The system confirms the contract acceptance and displays a confirmation message to the customer. |

| | |
|---|---|
| Alternate Event Flow | 1. If the customer disagrees with specific terms or conditions, they may choose not to proceed with the booking.<br>2. The system may provide options to modify the booking details or seek clarification on certain terms. |

### 9.3.2 Payment

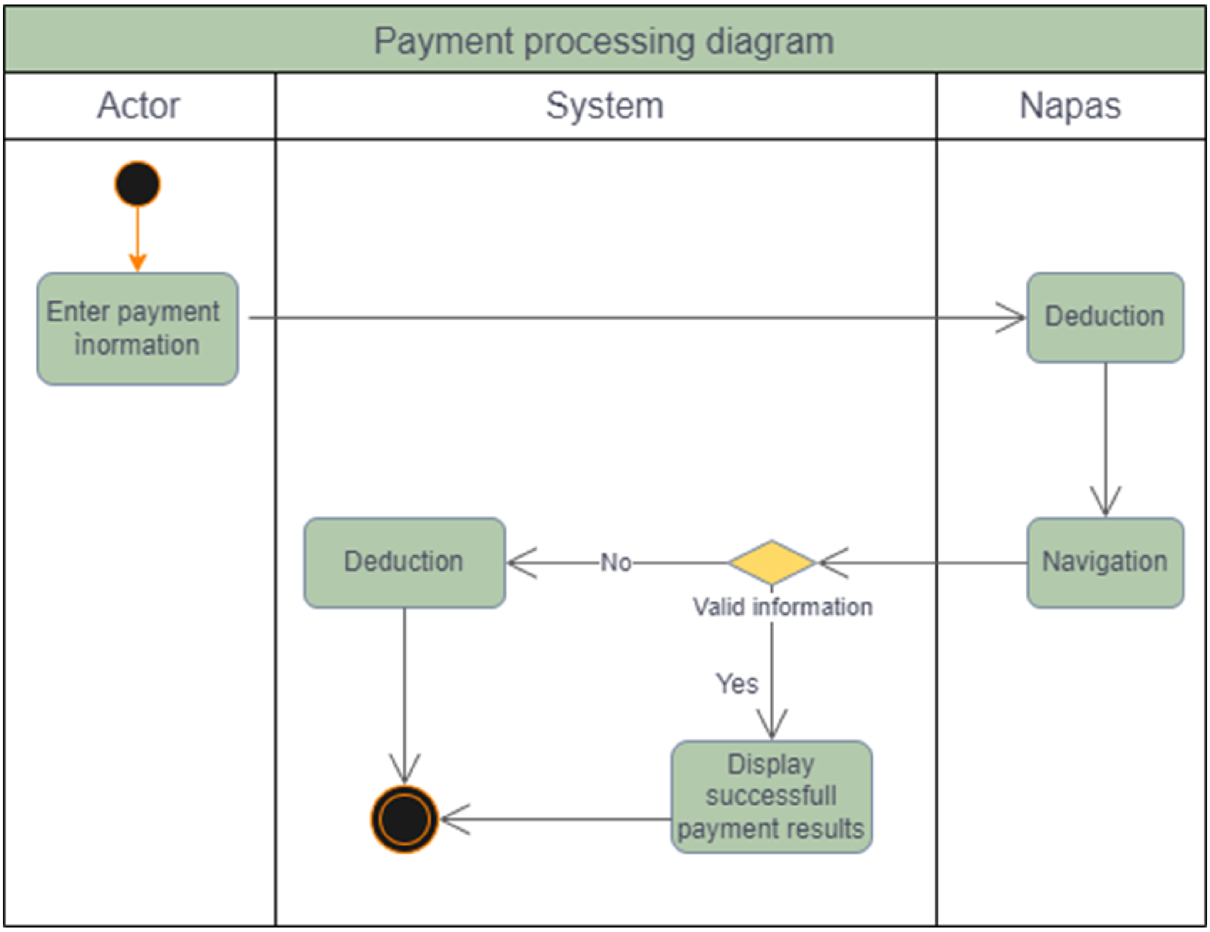


Figure 15: Payment Processing Diagram

- Description of the diagram:

| Step | Description | Notes |
|---|---|---|
| 1 | Customer enters payment information using bank account details or scans a QR code. | |
| 2 | The bank deducts the payment amount from the customer's account to settle the fee. | |
| 3 | Redirect back to the system. | |
| 4.1 | Verify that the following conditions are met:<br>● Contract ID (order ID) exists and matches the previously confirmed ID.<br>● The amount deducted from the customer's account matches the previously confirmed amount.<br>● Customer's account phone number exists and is active.<br>● Payment status is "pending confirmation." | |
| 4.2 | Display corresponding error message if the verification fails due to invalid information. | |

| | | |
|---|---|---|
| 5 | The system displays the result of the successful payment. | |

- Receiving Result Return Flow

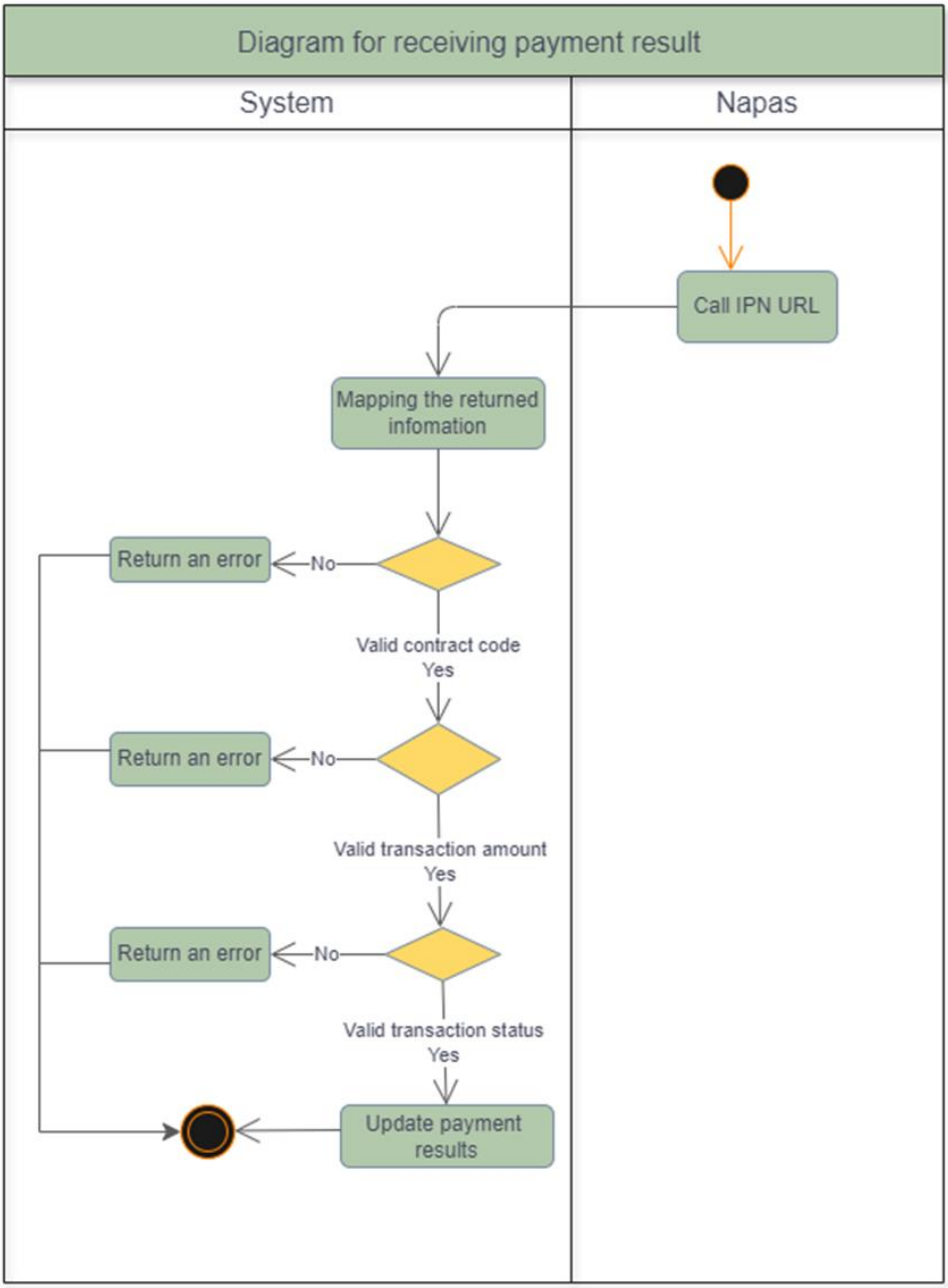


Figure : Receiving Result Return Diagram

- Description of the diagram:

| Step | Description | Notes |
|---|---|---|
| 1 | Napas calls the API and returns the payment result. | |
| 2 | The system maps the returned information to check the following conditions:<br>● Contract ID (order ID) exists and matches the previously confirmed ID.<br>● Transaction amount is valid and matches the previously confirmed amount.<br>● Payment status is "pending confirmation." | Reference (Bảng mã lỗi · Cổng thanh toán VNPAY (vnpayment.vn)) |
| 3.1 | Update the payment result in the system. | |
| 3.2 | Display an error message if the update is unsuccessful. | |

## 9.4. Admin

### 9.4.1. Manage Customer

- Admin can view, add, update, or delete customer information.

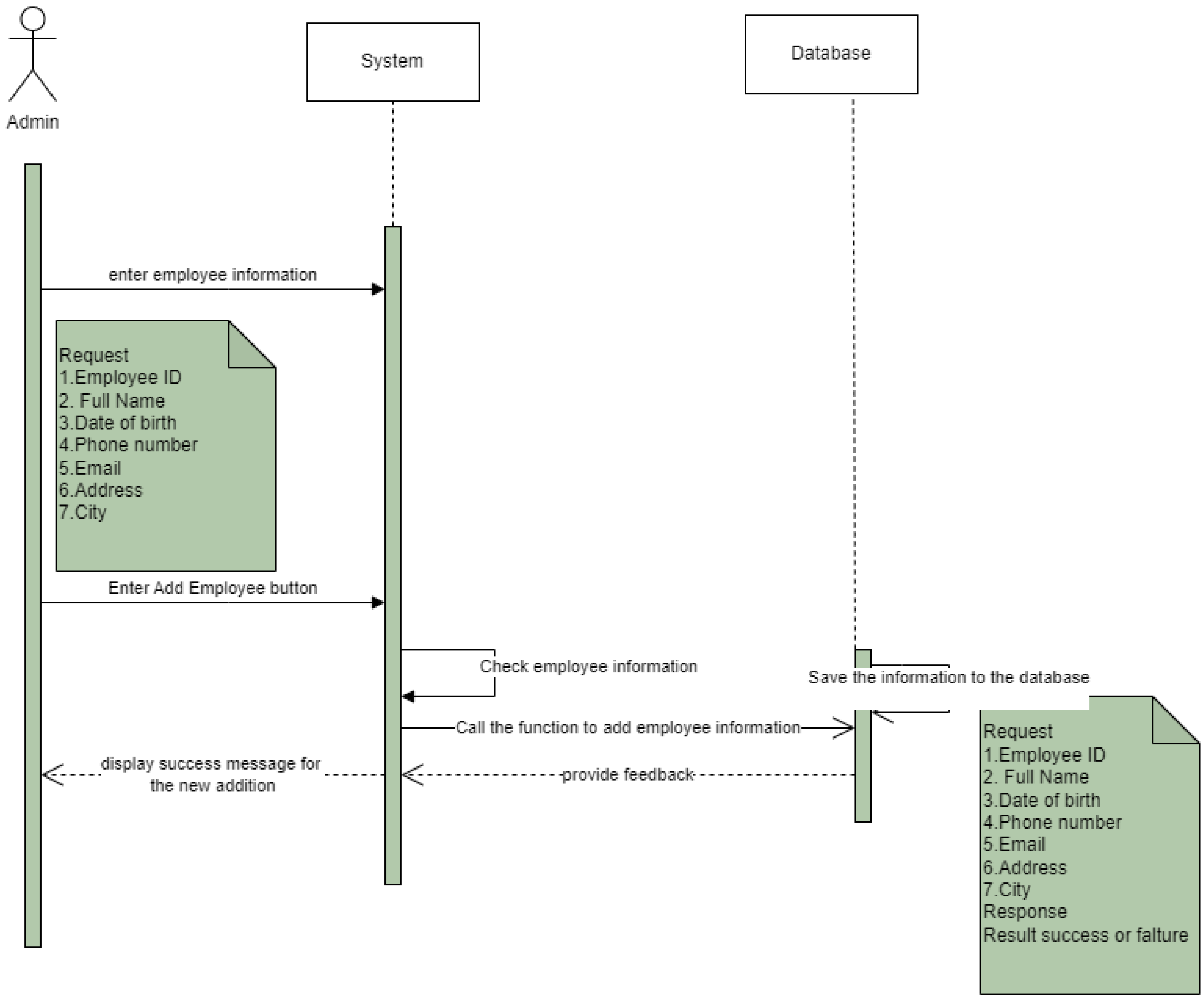

Admin
System
Database
enter employee information
Request
1.Employee ID
2. Full Name
3.Date of birth
4.Phone number
5.Email
6.Address
7.City
Enter Add Employee button
Check employee information
Save the information to the database
Call the function to add employee information
display success message for the new addition
provide feedback
Request
1.Employee ID
2. Full Name
3.Date of birth
4.Phone number
5.Email
6.Address
7.City
Response
Result success or failure

Figure: Sequence Diagram for Adding Customer Information

- Description of viewing customer information functionality

| | |
|---|---|
| Use Case name | View Customer Information |

| | |
|---|---|
| Overview | Admin views customer information |
| Actor | Admin |
| Trigger | Accessing the customer management section.<br>Selecting the option to view customer information. |
| Pre-Condition | · Admin is authenticated.<br>· Admin's role is admin. |
| Post-Condition | Customer information is successfully viewed. |
| Main events | Admin navigates to the customer management section.<br>Admin selects the option to view customer information.<br>The system displays a list of all customers with relevant details. |
| Extra events | Customer information viewed successfully.<br>Unauthorized access (Invalid permission). |

- Description of adding a new customer functionality

| | |
|---|---|
| Use Case name | Add Customer Information |

| | |
|---|---|
| Overview | Admin adds new customer information to the system. |
| Actor | Admin |
| Trigger | Accessing the customer management section.<br>Selecting the option to view customer information. |
| Pre-Condition | · Admin is authenticated.<br>· Admin's role is admin. |
| Post-Condition | New customer information is successfully added to the system. |
| Main events | Admin navigates to the customer management section.<br>Admin selects the option to add a new customer.<br>Admin fills in the required fields for the new customer.<br>The system validates the entered information.<br>If validation is successful, the system adds the new customer to the database. |
| Extra events | Customer information added successfully.<br>Unauthorized access (Invalid permission). |

- Description of updating customer information functionality

| | |
|---|---|
| Use Case name | Update Customer Information |
| Overview | Admin updates existing customer information in the system. |
| Actor | Admin |
| Trigger | Accessing the customer management section.<br>Admin updates existing customer information in the system. |
| Pre-Condition | · Admin is authenticated.<br>· Admin's role is admin. |
| Post-Condition | Existing customer information is successfully updated in the system |
| Main events | Admin navigates to the customer management section.<br>Admin selects the customer whose information needs to be updated.<br>Admin modifies the customer's information as required.<br>The system validates the updated information. |

| | |
|---|---|
| | If validation is successful, the system updates the customer's details in the database.. |
| Extra events | Customer information updated successfully.<br>Unauthorized access (Invalid permission). |

- Description of deleting a customer functionality

| | |
|---|---|
| Use Case name | Delete Customer Information |
| Overview | Admin deletes customer information from the system. |
| Actor | Admin |
| Trigger | Accessing the customer management section.<br>Selecting the customer whose information needs to be deleted. |
| Pre-Condition | · Admin is authenticated.<br>· Admin's role is admin. |

| | |
|---|---|
| Post-Condition | Selected customer information is successfully deleted from the system. |
| Main events | Admin navigates to the customer management section.<br>Admin selects the customer whose information needs to be deleted.<br>Admin confirms the deletion action.<br>The system removes the selected customer's information from the database. |
| Extra events | Customer information deleted successfully.<br>Unauthorized access (Invalid permission). |

**9.4.2. Manage Employees**

- Admin can view, add, update, or delete employee information.

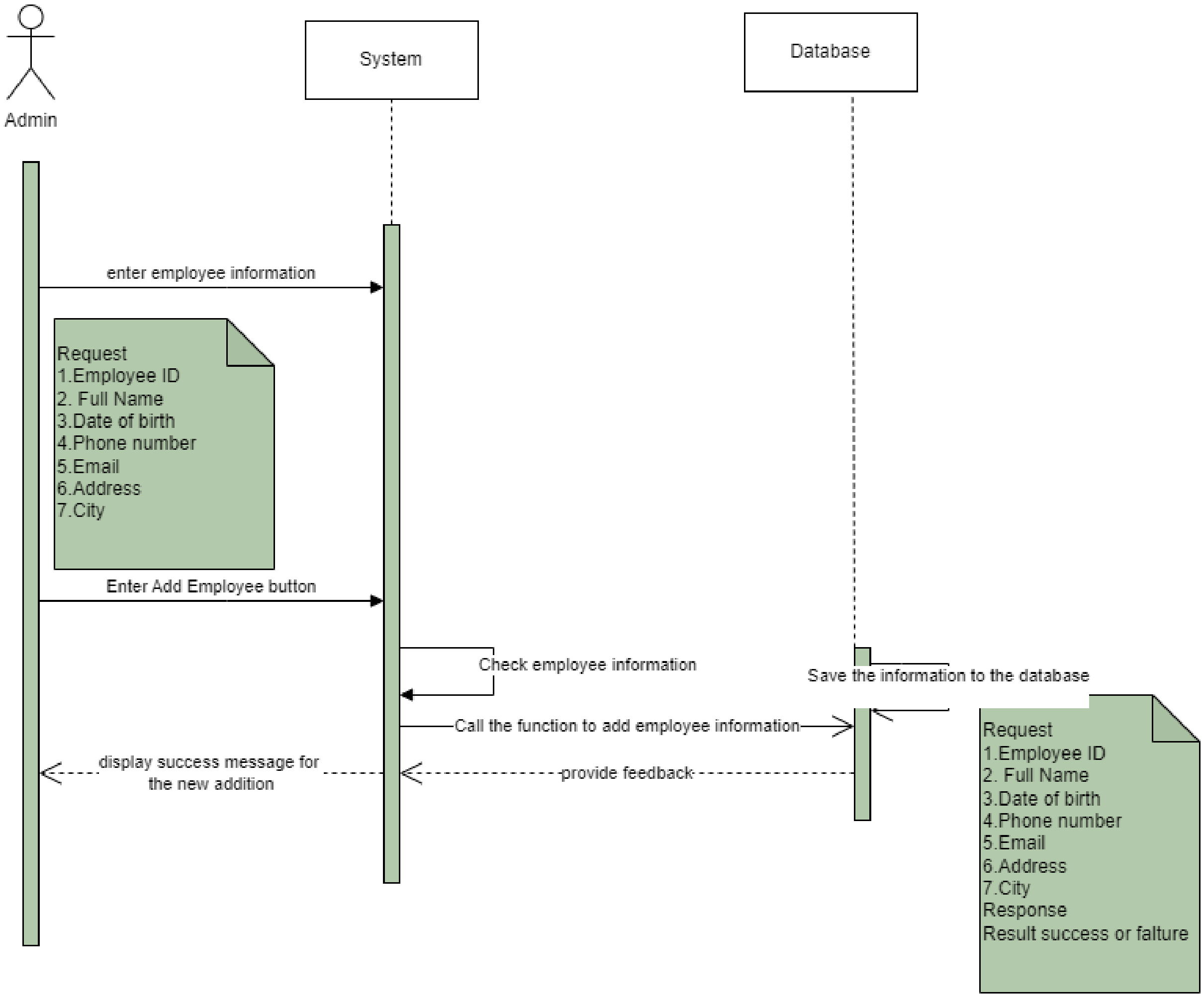

Admin
System
Database
enter employee information
Request
1.Employee ID
2. Full Name
3.Date of birth
4.Phone number
5.Email
6.Address
7.City
Enter Add Employee button
Check employee information
Save the information to the database
Call the function to add employee information
display success message for the new addition
provide feedback
Request
1.Employee ID
2. Full Name
3.Date of birth
4.Phone number
5.Email
6.Address
7.City
Response
Result success or falture

Figure : Sequence Diagram for Adding Employee Information

- Description of viewing employee information functionality

| | |
|---|---|
| Use Case name | View Customer Information |
| Overview | Admin views employee information. |
| Actor | Admin |
| Trigger | Accessing the employee management section. |

| | |
|---|---|
| | Selecting the option to view employee information. |
| Pre-Condition | - Admin is authenticated.<br>- Admin's role is admin. |
| Post-Condition | Customer information is successfully viewed. |
| Main events | Admin navigates to the employee management section.<br>Admin selects the option to view employee information.<br>The system displays a list of all employees with relevant details. |
| Extra events | Employee information viewed successfully.<br>Unauthorized access (Invalid permission). |

- Description of adding a new employee functionality

| | |
|---|---|
| Use Case name | Add Employee Information |
| Overview | Admin adds new employee information to the system. |

| | |
|---|---|
| Actor | Admin |
| Trigger | Accessing the employee management section.<br>Selecting the option to add a new employee. |
| Pre-Condition | - Admin is authenticated.<br>- Admin's role is admin. |
| Post-Condition | Customer information is successfully viewed. |
| Main events | Admin navigates to the customer management section.<br>Admin selects the option to view customer information.<br>The system displays a list of all customers with relevant details. |
| Extra events | Customer information viewed successfully.<br>Unauthorized access (Invalid permission). |

- Description of updating employee information functionality

| | |
|---|---|
| Use Case name | Update Employee Information |

| | |
|---|---|
| Overview | Admin updates existing employee information in the system. |
| Actor | Admin |
| Trigger | Accessing the employee management section.<br>Selecting the employee whose information needs to be updated. |
| Pre-Condition | - Admin is authenticated.<br>- Admin's role is admin. |
| Post-Condition | Existing employee information is successfully updated in the system. |
| Main events | Admin navigates to the employee management section.<br>Admin selects the employee whose information needs to be updated.<br>Admin modifies the employee's information as required.<br>The system validates the updated information.<br>If validation is successful, the system updates the employee's details in the database. |

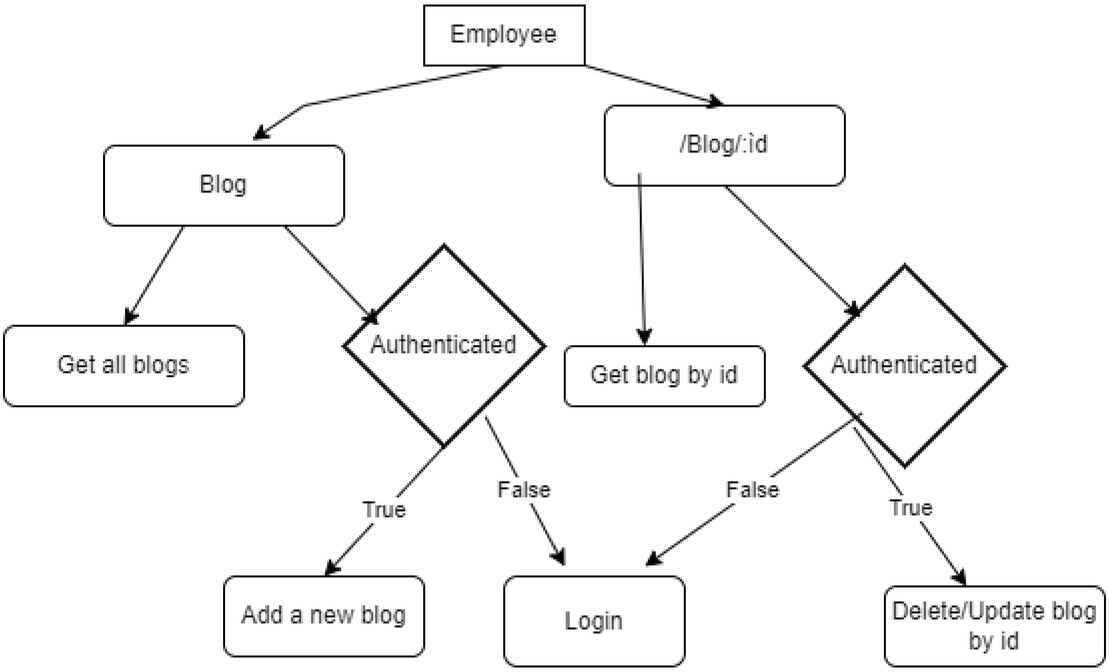
| | |
|---|---|
| Extra events | Employee information updated successfully.<br>Unauthorized access (Invalid permission). |

- Description of deleting a employee functionality

| | |
|---|---|
| Use Case name | Delete Employee Information |
| Overview | Admin deletes employee information from the system. |
| Actor | Admin |
| Trigger | Accessing the employee management section.<br>Selecting the employee whose information needs to be deleted. |
| Pre-Condition | - Admin is authenticated.<br>- Admin's role is admin. |
| Post-Condition | Selected employee information is successfully deleted from the system. |

|  |  |
|---|---|
| Main events | Admin navigates to the employee management section.<br>Admin selects the employee whose information needs to be deleted.<br>Admin confirms the deletion action.<br>The system removes the selected employee's information from the database |
| Extra events | Employee information deleted successfully.<br>Unauthorized access (Invalid permission). |

### 9.5. Blog



| | |
|---|---|
| Usecase name | Creates Blog |
| Actor | Employee |
| Pre - Condition | Employee is authenticated.<br>Employee has permission to create blog posts |
| Post-Condition | New blog post is successfully created and added to the blog database. |
| Trigger | Accessing the blog creation page.<br>Selecting the option to create a new blog post. |
| Main Event Flow | Employee navigates to the blog creation page.<br>Employee selects the option to create a new blog post.<br>Employee fills in the required fields for the new blog post, including title, content, and any relevant metadata.<br>Employee submits the blog post for creation.<br>The system validates the entered information.<br>If validation is successful, the system adds the new blog post to the blog database |

| | |
|---|---|
| Extra Event | Blog post created successfully.<br>Unauthorized access (Invalid permission).<br>Error messages displayed for missing or invalid information |

| | |
|---|---|
| Usecase name | Update blog |
| Actor | Employee |
| Pre - Condition | Employee is authenticated.<br>Employee has permission to edit blog posts. |
| Post-Condition | - Existing blog post is successfully updated with new content in the blog database. |
| Trigger | Accessing the blog editing page.<br>Selecting the option to edit a specific blog post. |
| Main Event | Employee navigates to the blog editing page.<br>Employee selects the blog post that needs to be updated. |

| | |
|---|---|
| | Employee modifies the content, title, or metadata of the blog post as required.<br>Employee submits the changes for update.<br>The system validates the updated information.<br>If validation is successful, the system updates the blog post in the blog database. |
| Extra Event | Blog post updated successfully.<br>Unauthorized access (Invalid permission).<br>Error messages displayed for missing or invalid information. |

| | |
|---|---|
| Usecase name | Delete blog |
| Actor | Employee |
| Pre - Condition | Employee is authenticated.<br>Employee has permission to delete blog posts. |
| Post-Condition | Selected blog post is successfully removed from the blog database. |

|  |  |
|---|---|
| Trigger | Accessing the blog management page.<br>Selecting the option to delete a specific blog post. |
| Main Events | Employee navigates to the blog management page.<br>Employee selects the blog post that needs to be deleted.<br>Employee confirms the deletion action.<br>The system removes the selected blog post from the blog database. |
| Extra Event | Blog post deleted successfully.<br>Unauthorized access (Invalid permission) |

## III. SYSTEM DESIGN

### 1.Technologies Uses

### 1.1. Front-end

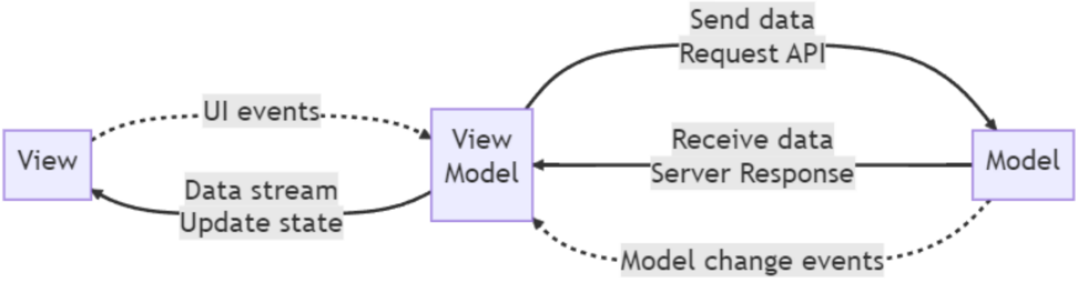
- HTML, CSS, JavaScript
- ReactJS (JavaScript framework for building user interfaces)
- Redux (state management library for JavaScript applications)

### 1.2. Back-end

- Node.js (JavaScript runtime)
- Express.js (web application framework for Node.js)
- MySQL

### 2. Software architecture

For separate use of Frontend (ReactJS) and Backend (Nodejs - Fastify), the software architecture does not follow the traditional MVC model but will be in the form of MVVM (Model – View – ViewModel) variant.



- View: Graphical User Interface (ReactJS)

- ViewModel: Combination of states, props and state management tools like React Context or React Redux - Model: Server – Side APIs

## IV. IMPLEMENTATION

## 1. Planning

- Brainstorm idea and create workflow for team with ClickUp
- Create system prototype design with Figma

# 2.Development

## 2.1.Specify workflow

- Separate frontend and backend: The frontend and backend will be developed independently and stored in separate repositories. The backend will include the frontend as a submodule, allowing both teams to work independently while still ensuring consistency and compatibility.
- Create API documentation with Postman: The backend team will use Postman to create documentation for the APIs. This documentation will include detailed information about the API endpoints, HTTP methods, parameters, request structures, and responses.
- Develop frontend based on API documentation. The frontend team will follow API documentation created by Postman to make requests to the server. They will develop user interfaces components and integrate them with APIs using HTTP requests.
- Testing and deployment: After completion, both teams will test the functionality of the website, including both frontend and backend. If all functions work as expected, the website will be deployed.

## 2.2. Setup Git for version control

- Create a Git repository for frontend (client).
- Create a Git repository for backend (server), include client repo as submodule.
- Specify standards and formatting must be followed by teammates:
    - Branch protection rules: Prevent force push, reviews are needed for pull request
    - Pre – commit hook: Run command before committing

- Es lint: Specify rules, prevent commit if not follow
- Prettier: Code formatting

## 2.3. Building environment

### 2.3.1. Frontend

- Create new ReactJS frontend project with Vite – Javas cript + SWC
- Install Redux and other dependencies using npm

### 2.3.2. Backend

- Create a new backend project with Node.js installed.
- Install Express.js and other dependencies using npm.
- Connect to MySQL.
- Run backend and frontend simultaneously using Concurrently.

## 2.4. Connect server – side and client – side components

- Implement API endpoints on the server – side for handling requests from client – side
- Use fetch API on client – side to connect to server – side

# V. TESTING

| Test | Requirement | Expected result | Data | Evaluation |
|---|---|---|---|---|
| i. For the client | | | | |

| | | | | |
|---|---|---|---|---|
| Registration | Access the registration page | Displays the registration screen | No | Passed |
| | Successfully create a new username if not already taken | Registered successfully | {<br>Address:” “<br>Citizen_id:“ “<br>Date_of_birth:” ”<br>Full_name:” “<br>Pass_word:” “<br>Phone_number: ” “<br>User_email:” “<br>User_name:” “<br>} | Passed |
| | After successful registration, it will be redirected to the login page | Displays the login screen | No | Passed |
| Login | Access the login page | Displays the login screen | No | Passed |
| | Login account verification | If user.access equals ‘user’, then it will redirect to the customer screen. | {<br>Access:” “ | Passed |

| | | | | |
|---|---|---|---|---|
| | | If user.access equals ‘employee’, then it will redirect to the employee screen.<br>If user.access equals ‘admin’, it will redirect to the admin screen | Address:” “<br>Citizen_id:“ “<br>Date_of_birth:” ”<br>Full_name:” “<br>Pass_word:” “<br>Phone_number: ” “<br>User_email:” “<br>User_image:” “<br>User_name:” “<br>} | |
| | Successful login with correct registered username and password | Login successfully | {<br>User_name:” “<br>Pass_word:” “<br>} | Passed |
| | Display a notification when there is no account in the system | Display the notification: “No record existed” | {<br>User_name:” “<br>Pass_word:” “<br>} | Passed |
| | Display customer information | Display the table of customer information | All users with access = ‘user’ | Passed |

| | | | | |
|---|---|---|---|---|
| Display events & items | Display car information | Display the table of car information | All cars | Passed |
| | Display the orders | Display the table of car order information | All orders | Passed |
| | Display the user information and employee information | Display user information | {<br>Address:” “<br>Citizen_id:“ “<br>Date_of_birth:” ”<br>Full_name:” “<br>Pass_word:” “<br>Phone_number:” “<br>User_email:” “<br>User_image:” “<br>User_name:” “<br>} | Passed |
| | Navigate to links accurately | Navigate to links successfully | No | Passed |
| View contact<br>information details | Redirect to external websites, Facebook, ... | Redirected successfully | No | Passed |
| ii. For the server | | | | |

| Test | Requirement | Expected result | | Evaluation |
|---|---|---|---|---|
| Car management | Add car | Add a car to the database when clicking on 'addcar' and entering information | New car:<br>{<br>brand:" "<br>moder:" "<br>year:" "<br>transmission:" "<br>number of seats:" "<br>fuel type:" "<br>price per day:" "<br>image:" "<br>} | Passed |
| | Delete car | Remove the car from the database when clicking the delete button | No | Passed |
| User information | Fixed information | The information that cannot be edited will be displayed but cannot be modified | {<br>Citizen_id:" "<br>Date_of_birth:" "<br>Full_name:" "<br>User_email:" "<br>User_image:" "<br>} | Passed |

| | | | | |
|---|---|---|---|---|
| | Updated information | Successfully updated on the database | {<br>Address:” “<br>Pass_word:” “<br>Phone_number:” “<br>User_name:” “<br>} | Passed |
| Account management | Add account | Save successfully registered accounts to the database | {<br>Address:” “<br>Citizen_id:“ “<br>Date_of_birth:” ”<br>Full_name:” “<br>Pass_word:” “<br>Phone_number:” “<br>User_email:” “<br>User_image:” “<br>User_name:” “<br>} | Passed |

## VI. DEPLOYMENT

- Bundle frontend project

- Deploy bundle frontend to Vercel or Netlify and create REST API by using Lambda Functions in backend project (serverless)

## VII. CONCLUSION

During the process of learning, researching, and developing, our team has concluded and drawn the benefits of applying technological achievements, information technology in management, and information storage to serve life, specifically in the field of car rental. The web application for car rental management has provided information about types of cars according to each segment, number of seats, and unique features. Currently, the website includes customers being able to look up, update car information, and rent cars for personal purposes.

Achieved results: After the process of doing a large assignment, we have grasped more knowledge about business analysis, database design, and language. We have understood more about using UML diagrams to analyze and databases to manage information about products, customers, and contracts. However, due to the limited semester time, we still do not have enough knowledge and experience in operating and developing the Web system, so the system is not yet complete and fully functional as required. Besides, the skill of writing reports and documents still has many shortcomings and is not neat and accurate.

Future research direction: The data in the system is still not complete, accurate, or raw, so in the future, we will integrate and update more to enrich the database. Along with that is the need to develop and perfect the functions of the system, resulting in a better user experience for customers.