



**Jordan University of Science and Technology**  
**College of Computer Sciences & Information Technology**

## **SMART AUTOMATED GATE**

*A project submitted  
in partial fulfillment of the requirements for the degree of  
Bachelor in Software Engineering*

**by**

<b>Name</b>	<b>#ID</b>
Omar Mohammed Yousef Hasan	145569
Mohammed Kamel Mostafa Alshawaqfih	148543
Heba Adel Mahmoud Obeidat	141433
Najat Mohammed Ali Alrawashdeh	156813

**Supervised by**

DR. Mohammed Malkhawi

**Committee Member Names**

**Dr. Asmahan Alhasan**

**Dr. Malek Qasaimeh**

**June/2024**

# UNDERTAKING

This is to declare that the project entitled “Smart Automated Gate” is an original work done by undersigned, in partial fulfillment of the requirements for the degree “Bachelor in Software Engineering” at Software Engineering Department, College of Computer and Information Technology, Jordan University of Science and Technology.

All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student 1: Omar Mohammed Yousef Hasan (145569)

Student 2: Mohammed Kamel Alshawaqfeh (148543)

Student 3: Heba Adel Mahmoud Obeidat (141433)

Student 4: Najat Mohammed Ali Alrawashdeh (156813)

## ABSTRACT

*The Smart Portal Access Control System is an innovative solution designed to automate the process of determining vehicle access authorization based on license plate recognition. The project addresses the growing need for efficient and secure vehicle access management in various environments, such as gated communities, parking lots, and restricted zones. Leveraging advanced technologies including license plate recognition (LPR), database management, and access control algorithms, the system aims to enhance security, streamline operations, and improve user experience.*

*The project begins with thorough research and analysis to understand the requirements and challenges associated with vehicle access control.*

## **ACKNOWLEDGMENT**

Thank you so much for all the people who helped us get good at working as a programming team. And we thank all the members of the project team for their constant work, fatigue and hardship until we reach this ingenious conclusion.

A special thanks to our awesome supervisor, Dr. Mohammed Malkawi, who kept helping us all the way. And thank you to our families, friends, and co-workers who supported us during this journey. The teachers who've been with us since the beginning deserve a shout out too!

# CONTENT

<b>UNDERTAKING .....</b>	<b>II</b>
<b>ABSTRACT.....</b>	<b>III</b>
<b>ACKNOWLEDGMENT .....</b>	<b>IV</b>
<b>LIST OF TABLES.....</b>	<b>VI</b>
<b>LIST OF FIGURES .....</b>	<b>VII</b>
<b>LIST OF ACRONYMS AND ABBREVIATIONS .....</b>	<b>VIII</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>9</b>
1.1 OVERVIEW.....	9
1.2 PROJECT MOTIVATION .....	9
1.3 PROBLEM STATEMENT .....	9
1.4 PROJECT AIM AND OBJECTIVES .....	11
<b>CHAPTER 2: PLANNING PHASE .....</b>	<b>15</b>
2.1 SCOPE OF THE PROJECT .....	15
2.2 PROJECT RISKS AND PRODUCT RISKS .....	16
2.3 FEASIBILITY STUDY .....	17
2.4 PROJECT SCHEDULE .....	18
2.5 PROJECT SOFTWARE AND HARDWARE REQUIREMENTS .....	19
<b>CHAPTER 3: RELATED EXISTING SYSTEM.....</b>	<b>20</b>
<b>CHAPTER 4: REQUIREMENT ENGINEERING AND ANALYSIS.....</b>	<b>22</b>
4.1 USED TECHNIQUES .....	23
4.3 FUNCTIONAL REQUIREMENT & MODELLING .....	27
4.3.1 Use Case Diagram.....	28
4.3.2 Use Cases: Description & Details .....	29
4.4 NONFUNCTIONAL REQUIREMENTS: QUALITY & CONSTRIANTS .....	35
<b>CHAPTER 5: ARCHITECTURE &amp; DESIGN.....</b>	<b>37</b>
5.1 SOFTWARE ARCHITECTURE .....	37
5.2 SOFTWARE DETAILED DESIGN .....	38
5.2.1 Use Cases Internal Interactions.....	38
5.2.2 Class Diagram.....	41
5.2.3 State transition Diagram .....	42
5.2.4 Data Storage Organization .....	44
<b>CHAPTER 6: IMPLEMENTATION PLAN &amp; PROTOTYPING.....</b>	<b>45</b>
6.1 INTRODUCTION.....	45
6.2 PROTOTYPING.....	46
<b>CHAPTER 7: TESTING PLAN.....</b>	<b>51</b>
6.1 TOOLS.....	53
6.2. SYSTEM TESTING PLAN (BLACK BOX) .....	55
<b>CONCLUSIONS .....</b>	<b>57</b>
<b>REFERENCES.....</b>	<b>58</b>
<b>APPENDIX-A: MANUAL .....</b>	<b>59</b>

## LIST OF TABLES

<b>TABLE 1:</b> REGISTER ACCOUNT USE CASE DESCRIPTION & DETAILS.....	28
<b>TABLE 2:</b> LOGIN USE CASE DESCRIPTION & DETAILS .....	28
<b>TABLE 3:</b> BOOK INVITATION CARD USE CASE DESCRIPTION & DETAILS .....	29
<b>TABLE 4:</b> MODIFY BOOKING USE CASE DESCRIPTION & DETAILS .....	29
<b>TABLE 5:</b> CANCEL BOOKING USE CASE DESCRIPTION & DETAILS .....	30
<b>TABLE 6:</b> VIEW BOOKING STATUS USE CASE DESCRIPTION & DETAILS .....	30
<b>TABLE 7:</b> PERMIT CARD USE CASE DESCRIPTION & DETAILS .....	31
<b>TABLE 8:</b> RECEIVE BOOKING CONFIRMATION USE CASE DESCRIPTION & DETAILS.....	31
<b>TABLE 9:</b> PAYMENT DESCRIPTION & DETAILS .....	32
<b>TABLE 10:</b> GATE SYSTEM DESCRIPTION & DETAILS .....	32
<b>TABLE 11:</b> ENABLE GATE DESCRIPTION & DETAILS .....	33
<b>TABLE 12:</b> NONFUNCTIONAL REQUIREMENTS (PERFORMANCE, SECURITY, RELIABILITY) .....	34
<b>TABLE 13:</b> NONFUNCTIONAL REQUIREMENTS (MAINTAINABILITY, SCALABILITY, USABILITY) .....	35
<b>TABLE 14:</b> EQUIVALENCE PARTITIONING (EMAIL). .....	54
<b>TABLE 15:</b> BOUNDARY VALUE ANALYSIS (EMAIL). .....	55
<b>TABLE 16:</b> EQUIVALENCE PARTITIONING (PASSWORD). .....	55
<b>TABLE 17:</b> BOUNDARY VALUE ANALYSIS (PASSWORD). .....	56
<b>TABLE 18:</b> EQUIVALENCE PARTITIONING (USERNAME). .....	57
<b>TABLE 19:</b> BOUNDARY VALUE ANALYSIS (USERNAME). .....	57

# LIST OF FIGURES

<b>Figure 1:</b> CONTEXT DIAGRAM .....	15
<b>Figure 2:</b> GANTT CHART .....	18
<b>Figure 3:</b> SMART-GATES WEBSITE .....	21
<b>Figure 4:</b> MIGHTY-MULE WEBSITE .....	21
<b>Figure 5:</b> AUTOMATED GATE SERVICES .....	22
<b>Figure 6:</b> QUESTION1 .....	23
<b>Figure 7:</b> QUESTION2 .....	23
<b>Figure 8:</b> QUESTION3 .....	24
<b>Figure 9:</b> QUESTION4 .....	24
<b>Figure 10:</b> QUESTION5 .....	25
<b>Figure 11:</b> QUESTION6 .....	25
<b>Figure 12:</b> QUESTION7 .....	26
<b>Figure 13:</b> QUESTION8 .....	26
<b>Figure 14:</b> USE CASE DIAGRAM .....	27
<b>Figure 15:</b> COMPONENT DIAGRAM .....	36
<b>Figure 16:</b> SEQUENCE DIAGRAM .....	37
<b>Figure 17:</b> CLASS DIAGRAM .....	41
<b>Figure 18:</b> ACTIVITY DIAGRAM .....	42
<b>Figure 19:</b> DATABASE TABLE .....	44
<b>Figure 20:</b> ER DIAGRAM .....	44
<b>Figure 21:</b> HOME PAGE1 .....	47
<b>Figure 22:</b> HOME PAGE2 .....	47
<b>Figure 23:</b> HOME PAGE3 .....	47
<b>Figure 24:</b> SIGN UP PAGE .....	48
<b>Figure 25:</b> LOGIN IN PAGE .....	49
<b>Figure 26:</b> Booking Invitation Page – User Information .....	50
<b>Figure 26:</b> Permit Card Page – Employee Information .....	50
<b>Figure 26:</b> Permit Card Page – Student Information .....	51

## LIST OF ACRONYMS AND ABBREVIATIONS

UI	User Interface.
ER	Entity Relationship.



# CHAPTER 1: INTRODUCTION

## 1.1 Overview

The Smart Car Portal Access Control System is an intelligent solution designed to regulate and manage vehicle access into specified areas using license plate recognition technology. This system incorporates advanced algorithms to identify vehicles and determine their authorization status based on their license plate numbers.

## 1.2 Project Motivation

The motivation behind developing a Smart Portal Access Control System stems from the need for efficient and secure management of vehicle access to designated areas. Several key motivations drive the development and implementation of such a system:

- **Enhanced Security:** Traditional access control methods, such as manual gatekeepers or keycard systems, may be susceptible to human error or unauthorized access. By implementing a Smart Portal Access Control

System, the process becomes automated and less prone to manipulation or oversight, thereby enhancing overall security.

- **Improved Efficiency:** Manual verification of vehicles entering a premises can be time-consuming and inefficient, leading to delays and congestion, especially in high-traffic areas. With a Smart Portal System, vehicles can be quickly identified and authorized for entry based on their license plate numbers, streamlining the access process and reducing wait times.

### 1.3 Problem Statement

The problem at hand is to develop a Smart Portal Access Control System capable of seamlessly and accurately determining whether a vehicle is permitted to enter a designated area using its license plate number. This system must tackle the following key challenges:

- **Accuracy:** The system must accurately capture and recognize license plate numbers, even in adverse weather conditions, varying lighting conditions, and different vehicle sizes and orientations.
- **Real-time Decision Making:** The system must make access control decisions in real-time, ensuring swift responses to vehicle entry requests while maintaining robust security measures.

- **Security:** Ensuring that only authorized vehicles gain access to the designated area is paramount. The system must be resistant to tampering and unauthorized access attempts, preventing security breaches.

## **1.4 Project Aim and Objectives**

The aim of the project is to design and implement a Smart Portal Access Control System capable of accurately determining the authorization status of vehicles based on their license plate numbers. The system will provide seamless and efficient vehicle access management while enhancing security and user experience.

### **Project Objectives:**

- **Develop License Plate Recognition (LPR) System:** Design and implement a robust LPR system capable of accurately capturing and recognizing license plate numbers from vehicle images in real-time, considering factors such as varying lighting conditions, vehicle orientation, and plate obstructions.
- **Implement Database Management System:** Develop a centralized database management system to store information about authorized vehicles, including their license plate numbers, associated owners, and permission levels. Ensure efficient data storage, retrieval, and update functionalities.

- **Design Access Control Logic:** Develop algorithms and decision-making logic to determine the authorization status of vehicles based on the license plate numbers retrieved from the LPR system and matched against the database records.
- **Create User Interface:** Design and implement a user-friendly interface for administrators to manage the database, configure access control rules, view access logs, and monitor system status. Additionally, develop a user interface for vehicle owners to request access or inquire about their vehicle's status.
- **Ensure System Security:** Implement security measures to safeguard the system against unauthorized access, tampering, and data breaches. Employ encryption techniques and access control mechanisms to protect sensitive data and ensure compliance with privacy regulations.
- **Test and Validation:** Conduct comprehensive testing and validation of the Smart Portal Access Control System to ensure accuracy, reliability, and performance under various operating conditions.
- **Deployment and Integration:** Deploy the system at designated entry points, integrating hardware components such as cameras, sensors, and access control mechanisms as necessary.

**Q1. What is the goal that this project wants to achieve?**

The goal of the SMART AUTOMATED GATE project is to create a sophisticated access control system that automates the process of determining whether a vehicle is authorized to enter a designated area. The primary aim is to enhance security, efficiency, and convenience in managing vehicle access while reducing reliance on manual intervention.

**Q2. How this project can achieve this goal?**

1. **Research and Analysis:** Conduct thorough research to understand the requirements, challenges, and existing solutions in automated access control systems. Analyze the specific needs and constraints of the target environment.
2. **System Design:** Design a comprehensive system architecture that integrates components such as license plate recognition (LPR) cameras, database management systems, access control algorithms, and user interfaces. Define data flow, communication protocols, and system interactions.
3. **Development and Implementation:** Develop and implement the SMART AUTOMATED GATE system according to the designed architecture. This involves programming algorithms for license plate recognition, database management, access control decision-making, and user interface

development. Integrate hardware components such as LPR cameras, sensors, and gate control mechanisms.

## CHAPTER 2: PLANNING PHASE

### 2.1 Scope of the project

The scope of the project encompasses the development and implementation of a Smart Portal Access Control System designed to regulate vehicle entry into designated areas based on license plate recognition. The system aims to enhance security, efficiency, and user experience by automating the process of determining vehicle access authorization.

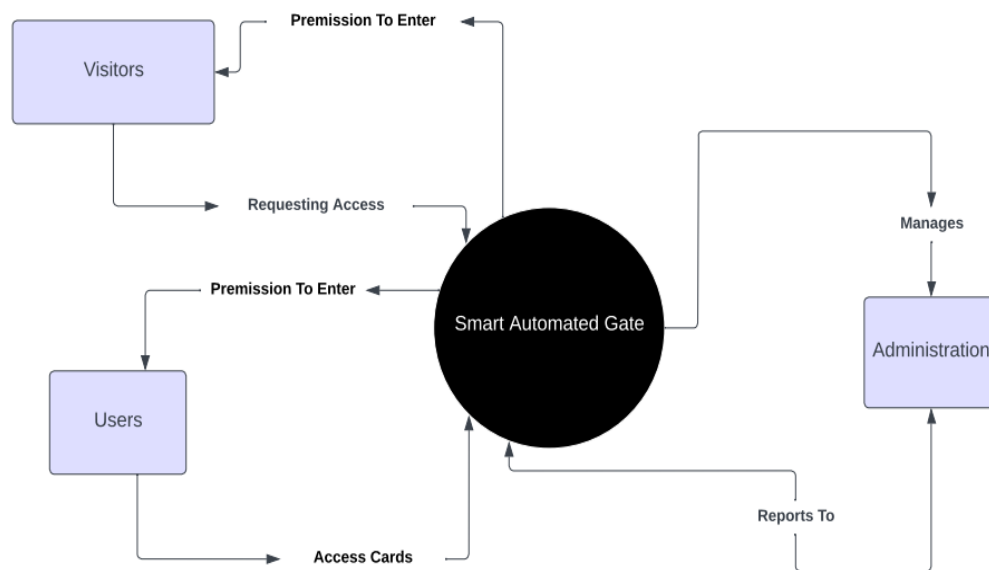


Figure 1: Context Diagram.

## **2.2 Project risks and Product risks**

**1. Cybersecurity Threats:** Smart automated gate systems are vulnerable to hacking and cyber-attacks. If a hacker gains access to the system, they may be able to open the gate remotely, bypass security measures, or even use it as a point of entry into the property.

**2. Malfunction or Technical Issues:** Automated gate systems rely on complex technology such as sensors, motors, and control systems. Malfunctions or technical issues can occur, leading to the gate getting stuck open or closed, posing security risks or inconvenience to the users.

**3. Power Outages:** Automated gate systems require a power source to operate. In the event of a power outage, the gate may become inoperable, potentially trapping vehicles inside or outside the property.

**4. Safety Hazards:** Automated gates can pose safety hazards if not installed or maintained properly. There's a risk of the gate closing on vehicles, pedestrians, or pets if the sensors fail or are improperly calibrated

**5. Dependency on Remote Access:** Many smart gate systems offer remote access capabilities via mobile apps or web interfaces. While convenient, reliance



on remote access can become a vulnerability if the system is not adequately secured or if the remote access credentials are compromised.

**6. Cost of Maintenance and Upkeep:** Smart gate systems may require regular maintenance and updates to ensure proper functioning and security. The cost of maintenance and upkeep should be factored into the overall cost of ownership.

## **2.3 Feasibility study**

### **Technical Feasibility:**

- **Technology Assessment:** Evaluate the availability and suitability of the required technologies for implementing the smart automated gate system, including license plate recognition (LPR) systems, sensors, cameras, gate control mechanisms, and networking infrastructure.

### **Scheduling Feasibility:**

- **Project Timeline:** Develop a comprehensive project schedule outlining key milestones, tasks, and deliverables from project initiation to completion, considering factors such as system design, testing, and training putting in mind one-week extra for unpredicted circumstances and everything goes as expected.

## 2.4 Project Schedule



Figure 2: Gantt Chart.

## **2.5 Project Software and Hardware Requirements**

### **2.5.1 Software requirement:**

#### **Employees, Visitors, Student:**

- They should have Internet access, and any kind of browsers.

#### **Developers:**

1. Visual Studio.
2. Web API, HTML, CSS, JavaScript, PHP, Oracle.
3. Cloud hosting servers, like Firebase or mysql.
4. API, for online payment.

### **2.5.2 Hardware requirement:**

- A server or a cloud-based infrastructure to host the system.
- A computer or laptop with sufficient processing power and memory to run the development environment.
- A stable internet connection to access the cloud-based infrastructure and other online resources.
- A development environment that supports the chosen programming language and frameworks.
- A device such as a mobile phone or a tablet to access the system.

## CHAPTER 3: RELATED EXISTING SYSTEM



Figure 3: smart-gate Model.



Figure 4: smart-gate Model.



Figure 5: smart-gate Model.

# CHAPTER 4: REQUIREMENT ENGINEERING AND ANALYSIS

## 4.1 Used Techniques

- **Surveys:** We used questionnaire methodology to collect data from users regarding their opinions and perception of the concept of this system.

How frequently do you use smart gates?

60 responses

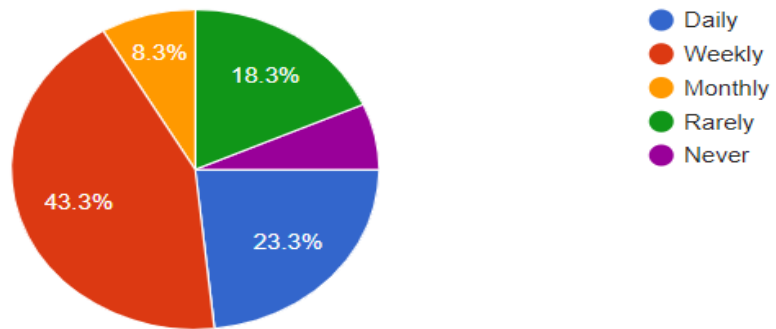


Figure 1-Question1.

What are the main challenges you face when using smart gates?

60 responses

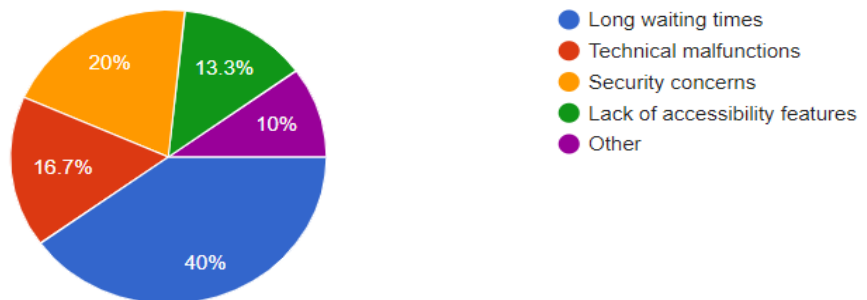


Figure 7-Question2.

What features do you think are essential for a smart gate?

60 responses

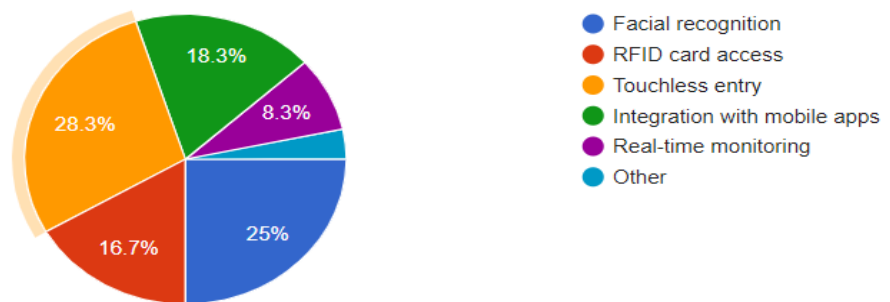


Figure 8-Question3.

For example (university): Can this smart portal save time?

60 responses

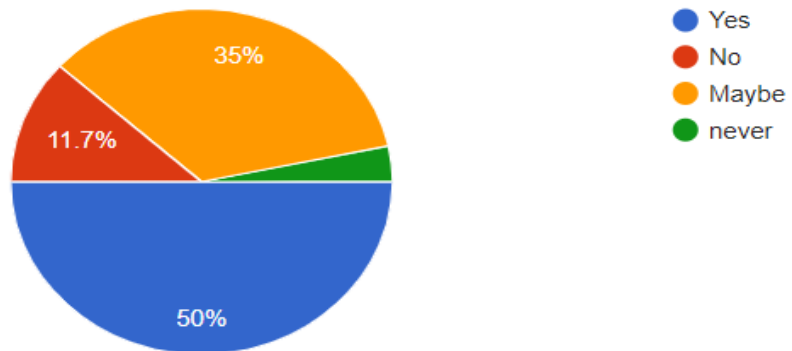


Figure 9-Question4.



How much time do you have to wait for a technician to service?

[Copy](#)

46 responses

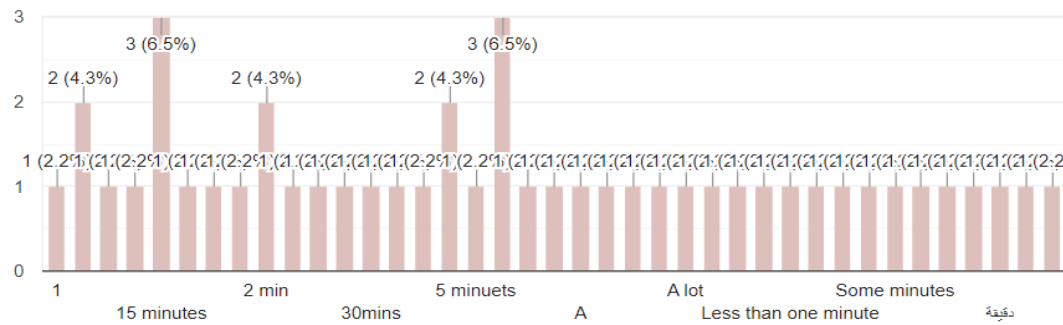


Figure 10-Question5.

Please select the factors that contribute to your decision to use smart gates. (Select all that apply)

[Copy](#)

60 responses

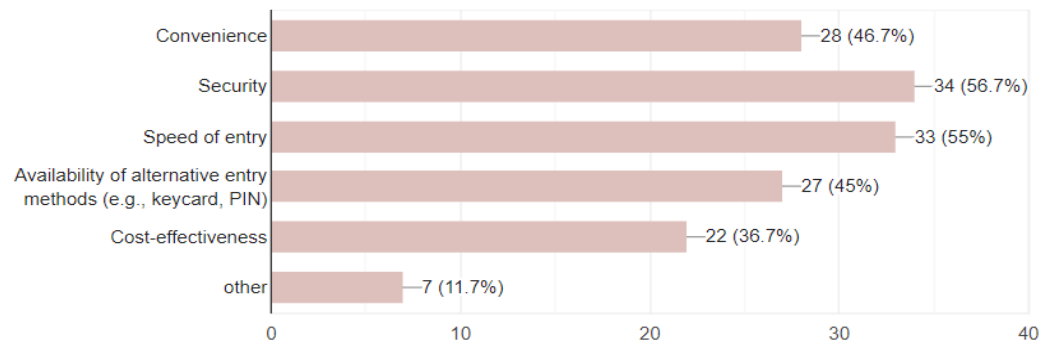


Figure 11-Question6.



Please select any concerns you have about using smart gates. (Select all that apply)

[Copy](#)

60 responses

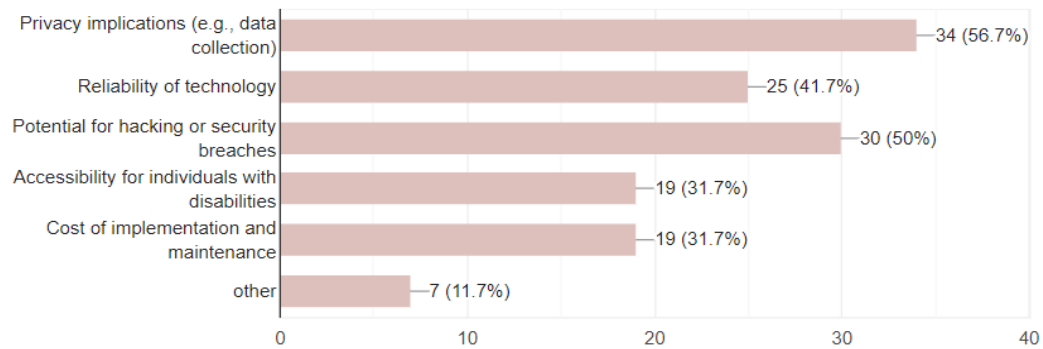


Figure 12-Question7.

Which of the following smart gate features do you believe would benefit your organization or community the most? (Select all that apply)

[Copy](#)

60 responses

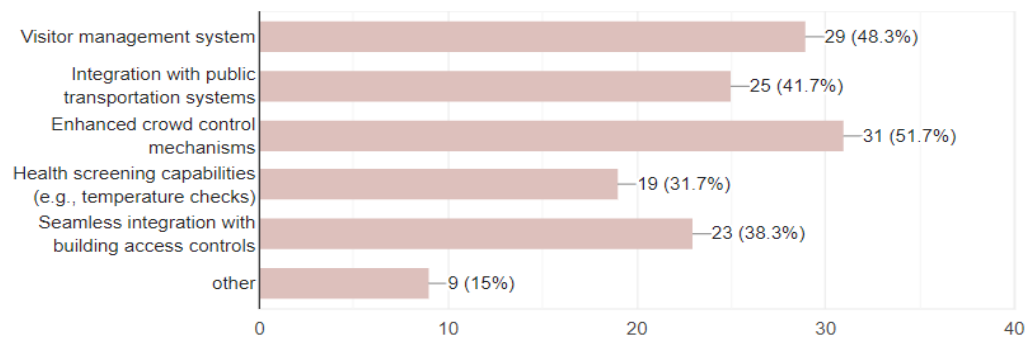


Figure 13-Question8.

### **4.3 Functional Requirement & Modelling**

- The system shall allow users to create an account using their email address and password.
- The system shall enable users to log in to their accounts to access personalized features.
- The system shall provide a mechanism for users to request invitation cards for specific dates and times.
- The system shall allow users to cancel or modify their existing bookings up to a certain time before the scheduled visit.
- The system shall send an email confirmation to users once their booking is successful or if there is a change in their booking status.
- The system shall integrate with a payment gateway to facilitate payments for bookings if any.
- The system shall provide an administrative interface for system administrators to view, manage, and report on bookings.
- The system shall automatically update the availability of bookings based on current reservations and cancellations.

### 4.3.1 Use Case Diagram

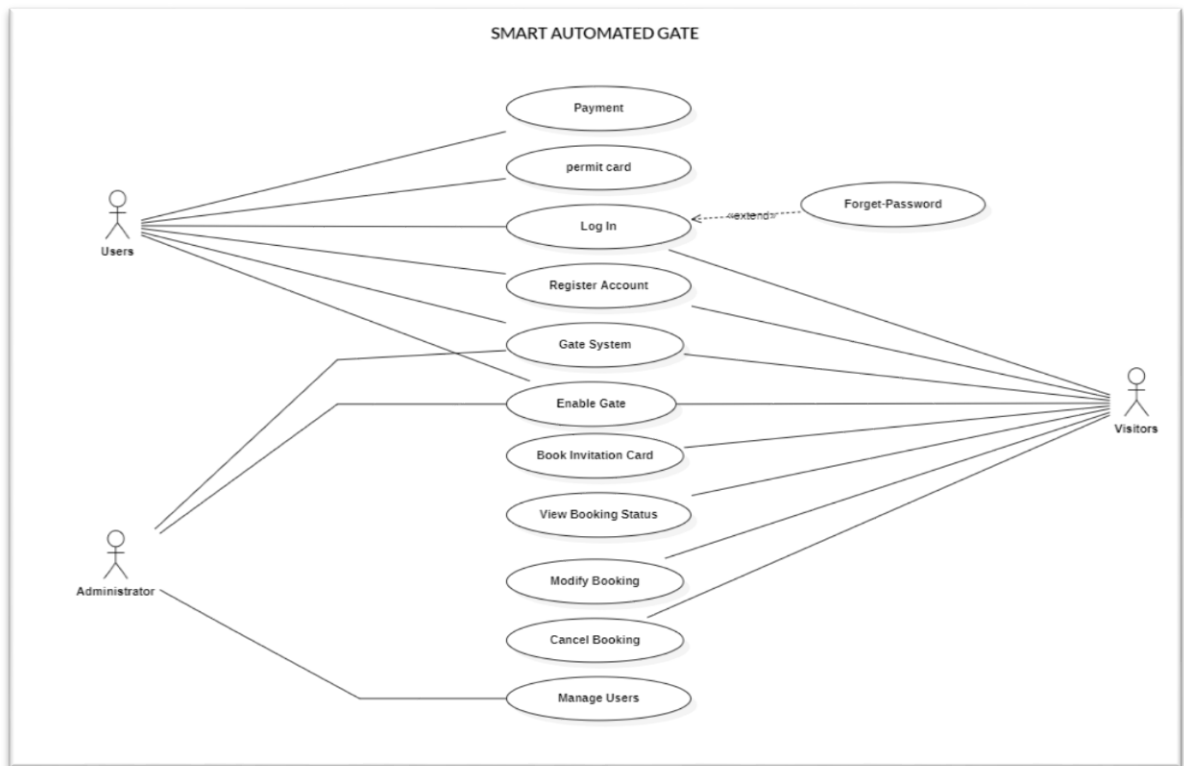


Figure 14-Use Case Diagram.

### 4.3.2 Use Cases: Description & Details

#### ➤ Smart Automated Gate Use Cases:

Use case name	1. Register Account
Brief Description	The Users & Visitors needed register to can login.
Primary Actor	Users
Secondary Actor	Visitors
Preconditions	The Users should have phone number & email.
Main Flow	<ol style="list-style-type: none"><li>1 Open website.</li><li>2 Choose "sign up" button.</li><li>3 Enter Users &amp; visitor information as a (Name, Email, Password, etc.)</li><li>4 Click on "Register".</li></ol>
Post conditions	The Users have account.

Table 1 - Register Account Use Case Description & Details.

Use case name	2. Login
Brief Description	The Users & Visitors needed to login on his account to can reserve an invitation card.
Primary Actor	Users
Secondary Actor	Visitors
Preconditions	The Users & Visitors should be having account.
Main Flow	<ol style="list-style-type: none"><li>1. Open website.</li><li>2. Enter username and password.</li><li>3. Click on "Login" button.</li></ol>
Post conditions	The Users & Visitors login is successful and enter Users to website.

Table 2 - Login Use Case Description & Details.

<b>Use case name</b>	<b>3. Book Invitation Card</b>
<b>Brief Description</b>	The Visitors needs an invitation card book to be allowed entry.
<b>Primary Actor</b>	Visitors.
<b>Secondary Actor</b>	None.
<b>Preconditions</b>	The system must have availability for the requested date & time.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Open website and click on "Book Invitation Card" button.</li> <li>2. Select "Date and Time" button.</li> <li>3. Confirm booking.</li> <li>4. System confirms booking.</li> </ol>
<b>Post conditions</b>	The booking details are stored in the system.

Table 3 – Book Invitation Card Use Case Description & Details.

<b>Use case name</b>	<b>4. Modify Booking</b>
<b>Brief Description</b>	allows a Visitors to modify an existing booking. Modifications can include changing the date & time.
<b>Primary Actor</b>	Visitors.
<b>Secondary Actor</b>	None.
<b>Preconditions</b>	The visitor must have an existing booking to modify.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Open website and select "Modify Booking" button.</li> <li>2. Choose "modification options".</li> <li>3. Input new details.</li> <li>4. System updates booking.</li> </ol>
<b>Post conditions</b>	The updated booking details are stored in the system.

Table 4 – Modify Booking Use Case Description & Details.

Use case name	5. Cancel Booking
Brief Description	enables a Visitors to cancel a previously made booking.
Primary Actor	Visitors.
Secondary Actor	None.
Preconditions	The Users must have an existing booking they wish to cancel.
Main Flow	<ol style="list-style-type: none"> <li>1. Open website and Select “existing booking” button.</li> <li>2. Initiate cancellation.</li> <li>3. Confirm cancellation.</li> <li>4. Process cancellation.</li> <li>5. Send confirmation of cancellation.</li> </ol>
Post conditions	The Users receives an email confirming the cancellation of the booking.

Table 5 - Cancel Booking Use Case Description & Details.

Use case name	6. View Booking Status
Brief Description	allows a Visitors to view the current status of their bookings
Primary Actor	Visitors.
Secondary Actor	None.
Preconditions	The Users has one or more bookings made through the system.
Main Flow	<ol style="list-style-type: none"> <li>1. Open website and Select “view details” button.</li> <li>2. Access booking history.</li> <li>3. Select “Display booking details” button.</li> </ol>
Post conditions	The system records the Users access to his or her reservation history for security

Table 6 - View Booking Status Use Case Description & Details.

Use case name	7. Permit Card
Brief Description	This card acts as an entry pass to the designated place to the Employee or Student by paying the card fee.
Primary Actor	Users.
Secondary Actor	None.
Preconditions	The system must have all the information necessary to issue the permit card.
Main Flow	<ol style="list-style-type: none"> <li>1. Open website and Generate permit card.</li> <li>2. Send permit card to Users.</li> <li>3. Downloads or prints permit card.</li> </ol>
Post conditions	The Users receives a confirmation of the payment along with a digital receipt via email.

Table 7 - Permit Card Use Case Description & Details.

Use case name	8. Manage Users
Brief Description	enables a system administrator to manage user accounts, including creating, updating, and deleting user information.
Primary Actor	Administrator.
Secondary Actor	None.
Preconditions	The administrator must be logged into the system with administrative privileges.
Main Flow	<ol style="list-style-type: none"> <li>1. Administrator accesses user management interface.</li> <li>2. Create a new user.</li> <li>3. Update existing user.</li> <li>4. Delete a user.</li> </ol>
Post conditions	The administrator receives confirmation successfully executed.

Table 8 - Manage Users Use Case Description & Details.

Use case name	9. Payment
Brief Description	Describes the process by which a user completes payment to reserve entry cards through the gate.
Primary Actor	Users.
Secondary Actor	None.
Preconditions	The user's payment information must be verified and up-to-date if stored on the system.
Main Flow	<ol style="list-style-type: none"> <li>1. The user enters or confirms payment information and selects the 'Pay Now' option.</li> <li>2. The system processes the payment through an integrated payment gateway.</li> </ol>
Post conditions	

Table 9 – Payment Description & Details.

Use case name	10. Gate System
Brief Description	describes how the gate system verifies invitation cards or Permit card to control entry to an event or location.
Primary Actor	Gate System.
Secondary Actor	Users.
Third Actor	Visitors.
Preconditions	The user has either a permit card accessible on a mobile device.
Main Flow	<ol style="list-style-type: none"> <li>1. The user presents their invitation card or permit card to the scanner.</li> <li>2. The system logs the entry attempt, noting the time, user ID (if available), and result of the attempt (granted or denied).</li> </ol>
Post conditions	The status of the entry (allowed or denied) in the system for security.

Table 10 – Gate System Description & Details.



Use case name	11. Enable Gate
Brief Description	describes the administrative actions required to activate and configure a gate within the system.
Primary Actor	Administrator.
Secondary Actor	Users.
Third Actor	Visitors.
Preconditions	The gate hardware (scanner/RFID reader) and associated software must be installed and in a ready state.
Main Flow	
Post conditions	The administrator receives confirmation that the gate is ready for use, ensuring that access control is maintained as planned.

Table 11 – Enable Gate Description & Details.

Use case name	12. Forget Password
Brief Description	The Users can change the password again.
Primary Actor	Users.
Secondary Actor	None.
Preconditions	The Users needs to enter the website and go to log in.
Main Flow	<ol style="list-style-type: none"> <li>1. Open website.</li> <li>2. Go to home page.</li> <li>3. Click on “Profile” button.</li> <li>4. Click on reset password.</li> <li>5. Click on save change.</li> </ol>
Post conditions	The password is changed.

Table 12 – Forget Password Description & Details.

#### 4.4 Nonfunctional Requirements: Quality & Constraints

Security	Reliability	Performance
<ul style="list-style-type: none"><li>Unauthorized Access: The system should prevent unauthorized vehicles from entering. This includes measures against license plate spoofing or tampering.</li><li>Data Security: The authorized car list and any captured license plate data should be securely stored and encrypted to prevent unauthorized access.</li><li>System Integrity: The system should be resistant to hacking or tampering attempts.</li></ul>	<ul style="list-style-type: none"><li>The system must reliably handle the registration and approval process for users and visitors.</li><li>The system must safely handle unexpected inputs or failures during booking, ensuring a seamless user experience.</li></ul>	<ul style="list-style-type: none"><li>The system must efficiently handle user interactions, such as booking submission and approval, even with a large user base.</li><li>The system must maintain optimal performance under extreme usage scenarios when multiple users access the platform simultaneously.</li></ul>

<b>Maintainability</b>	<b>Scalability</b>	<b>Usability</b>
<ul style="list-style-type: none"> <li>• The system should be designed with modularity and clear documentation, facilitating easy maintenance and updates without disrupting the entire platform.</li> </ul>	<ul style="list-style-type: none"> <li>• The system shall be scalable to accommodate an increasing number of users and bookings as demand grows.</li> </ul>	<ul style="list-style-type: none"> <li>• The system must provide an easy-to-use interface for users and visitors on the site, ensuring easy of use and clarity in the reservation process for the invitation card for visitors and the entry card for users.</li> </ul>

# CHAPTER 5: ARCHITECTURE & DESIGN

## 5.1 Software Architecture

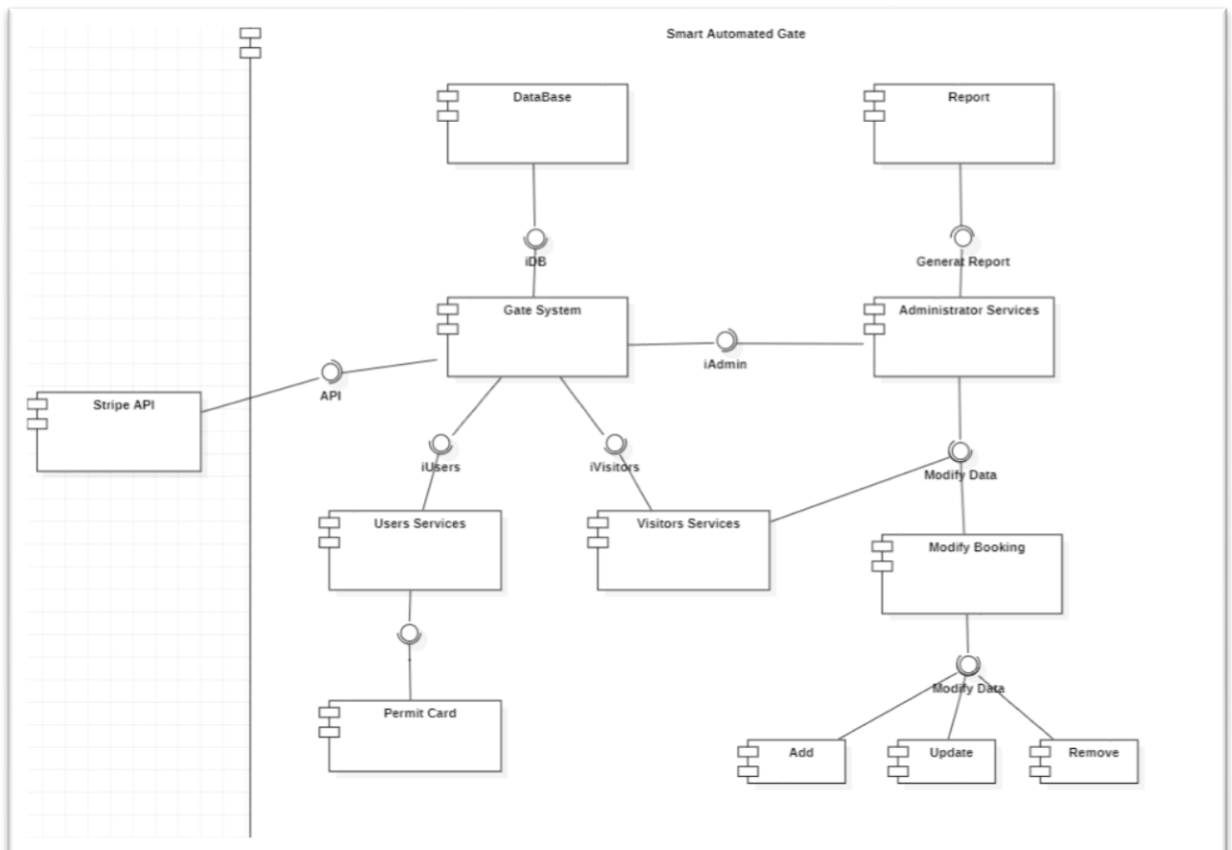


Figure 15 – Component Diagram.

## 5.2 Software Detailed Design

### 5.2.1 Use Cases Internal Interactions

- Sign in sequence:

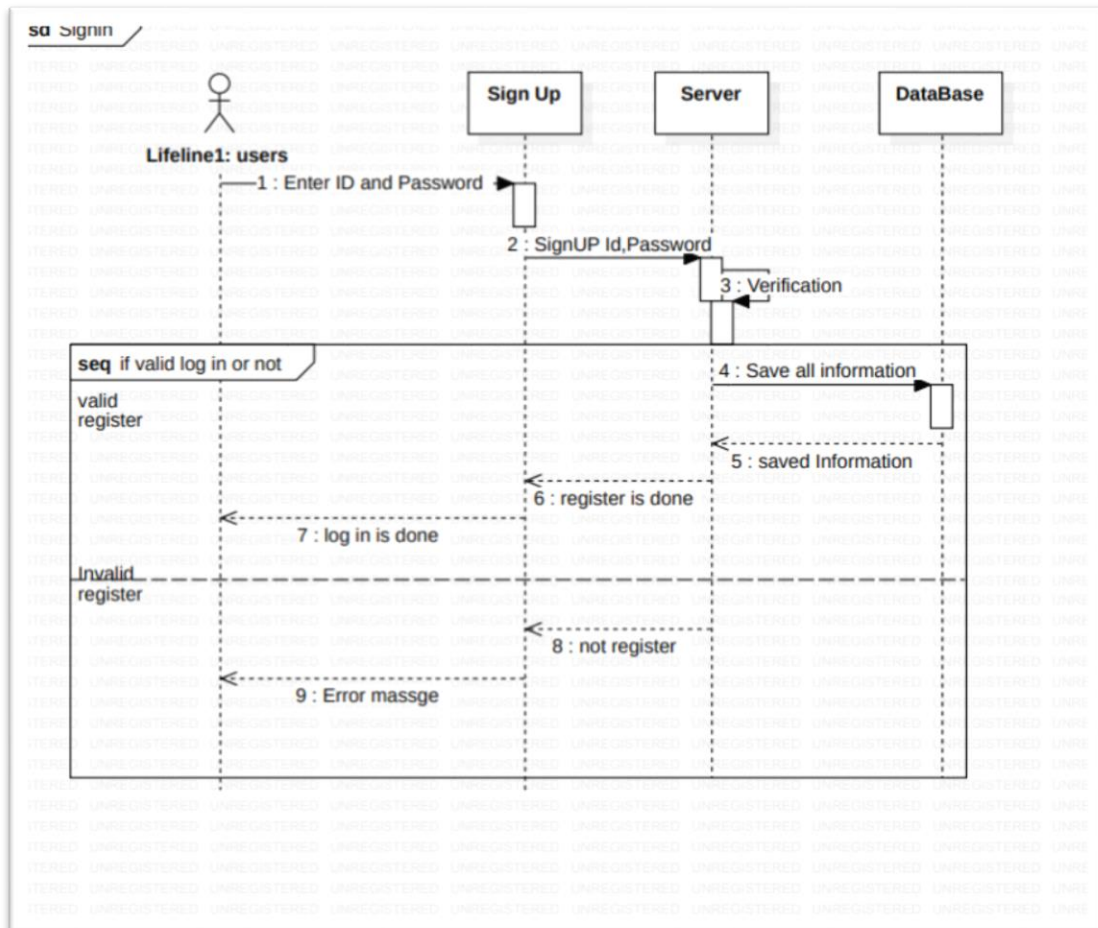


Figure 16: Sequence diagram (Sign in).

➤ Log in sequence:

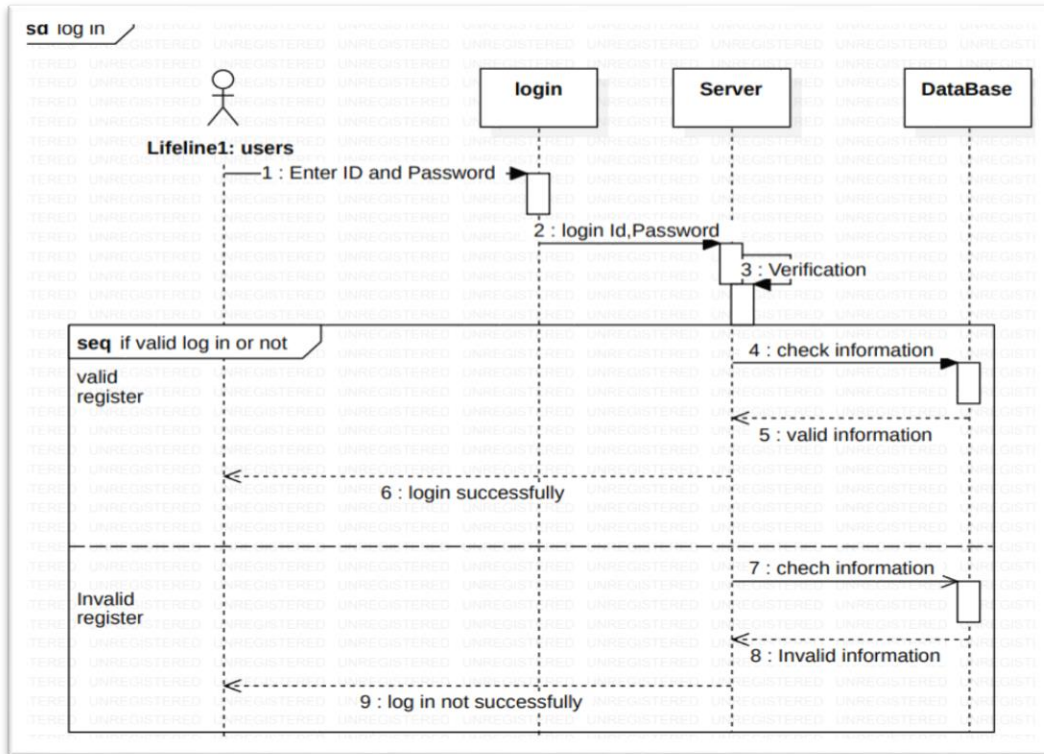


Figure 17: Sequence diagram (Log in).

➤ Send Description:

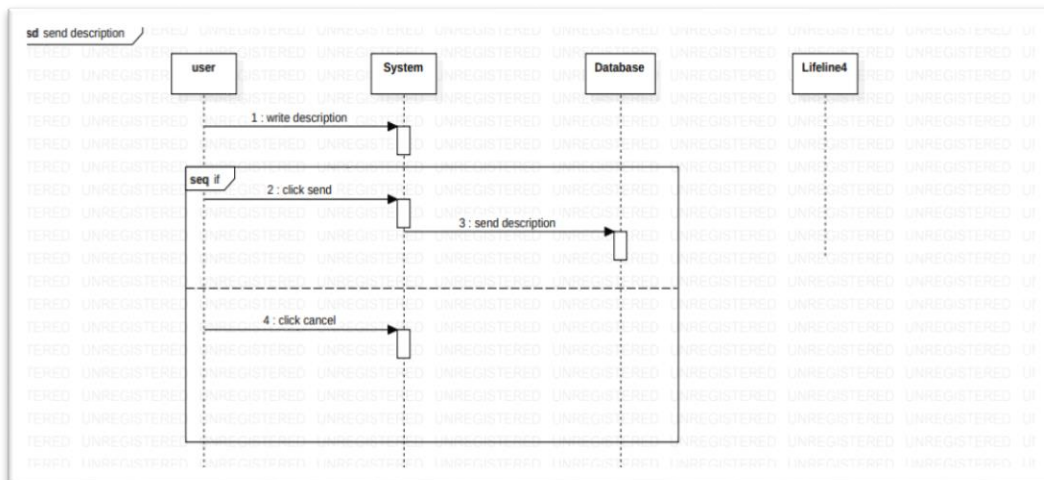


Figure 18: Sequence diagram (Send Description).

➤ Booking invitation card:

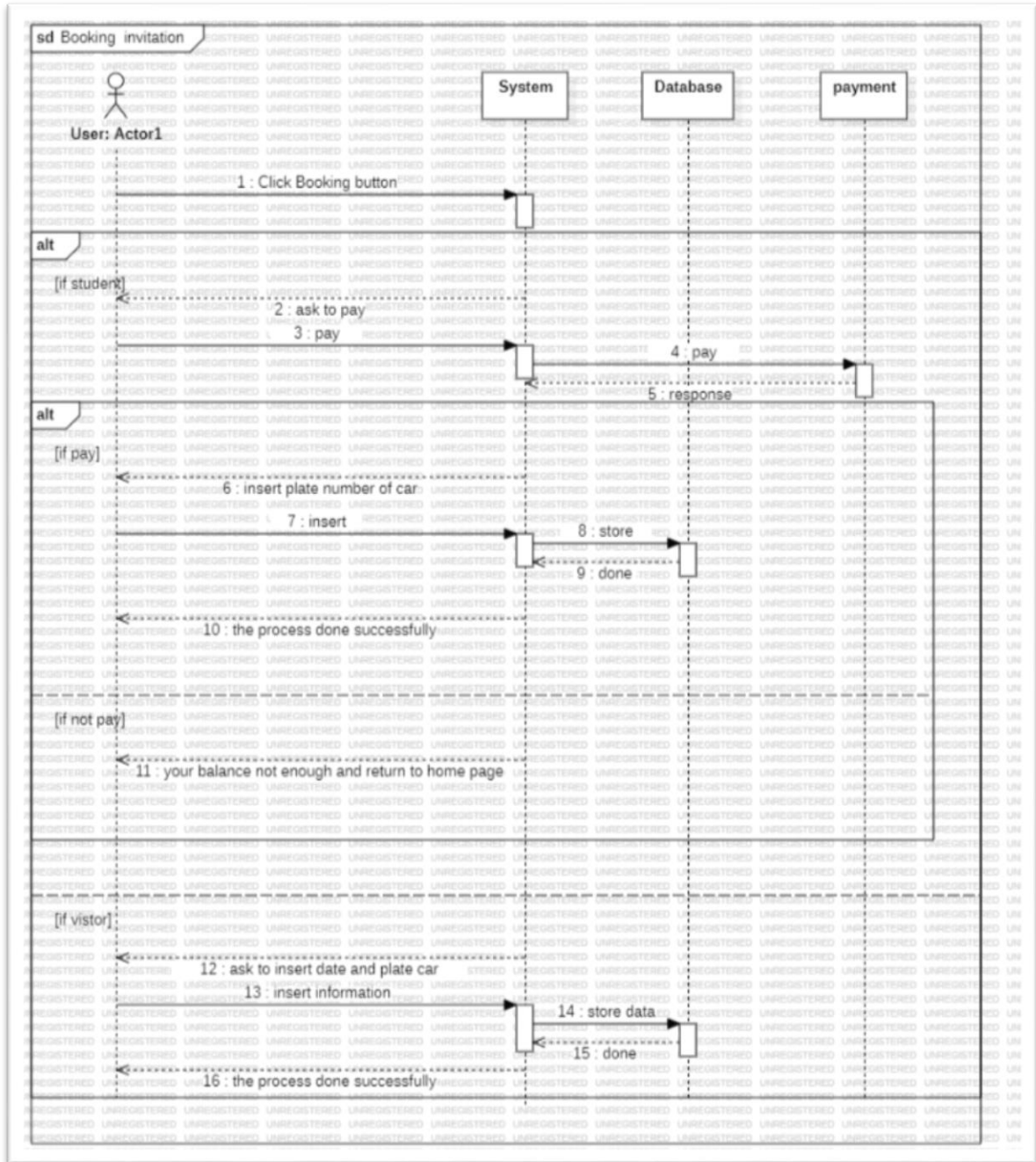


Figure 19: Sequence diagram (Booking Invitation Card).

➤ Enable Gate:

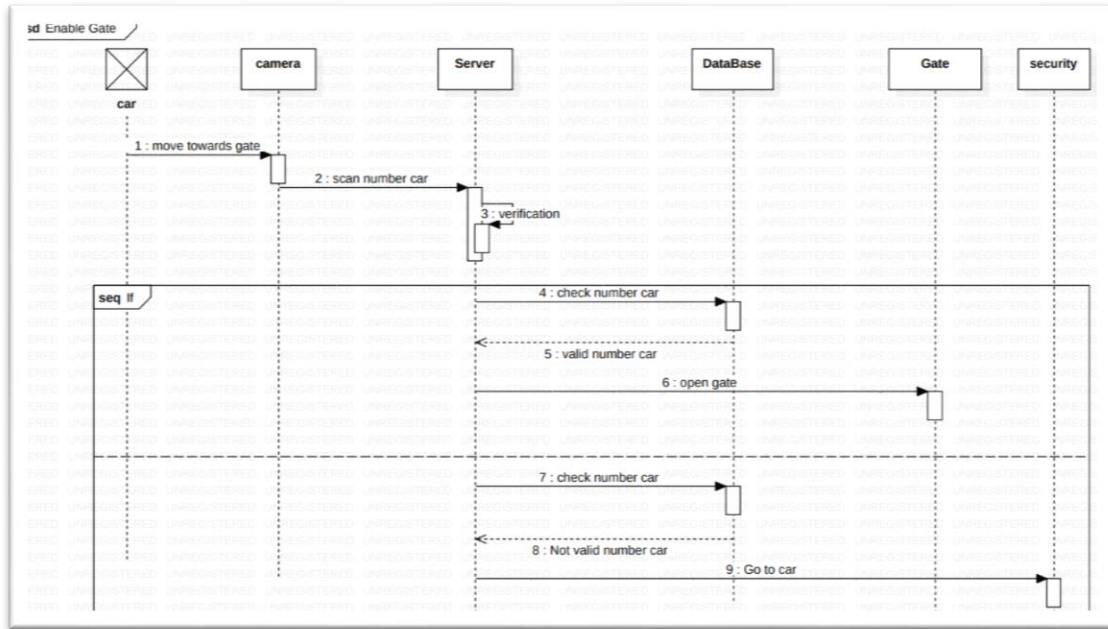


Figure 20: Sequence diagram (Enable Gate).

➤ View Booking Gate:

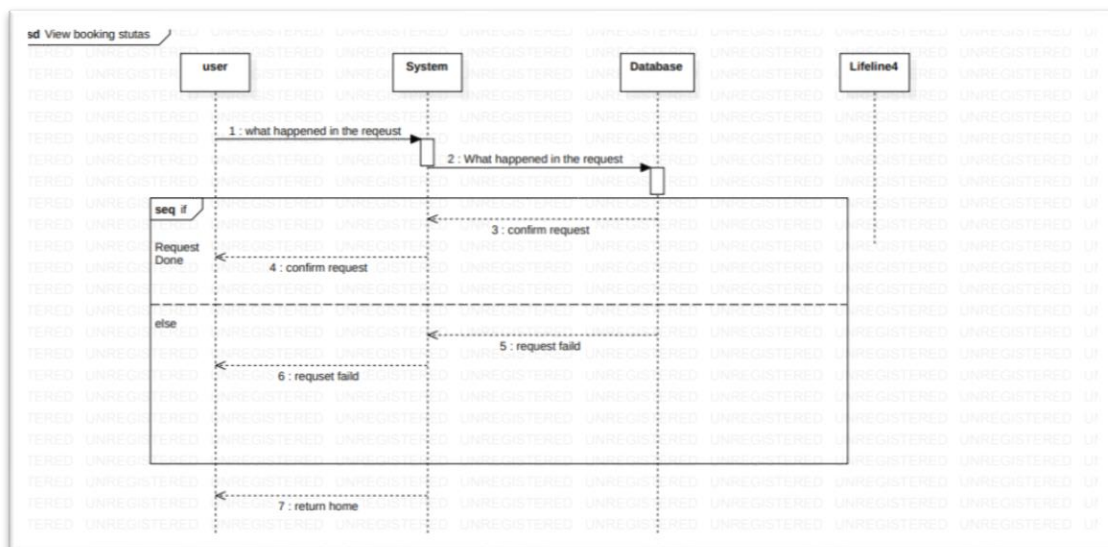


Figure 21: Sequence diagram (View Booking Gate).



## 5.2.2 Class Diagram

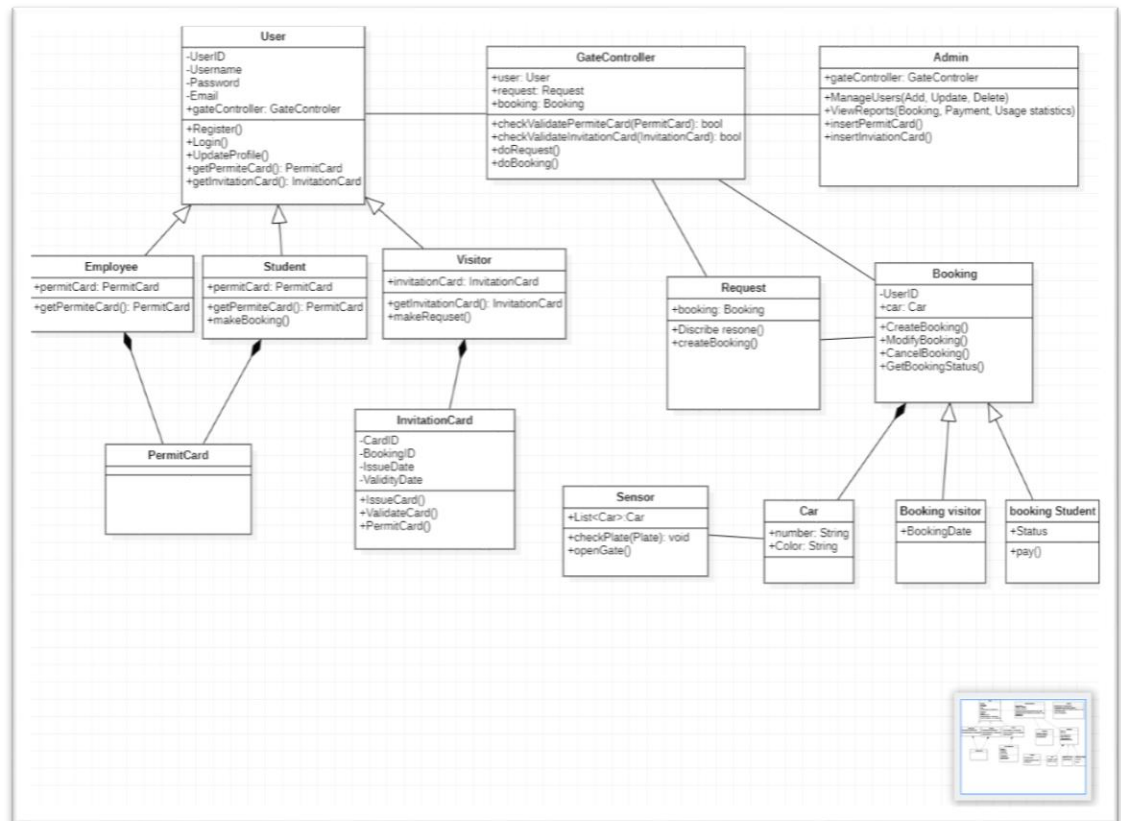


Figure 22: Class diagram.

### 5.2.3 Activity Diagram

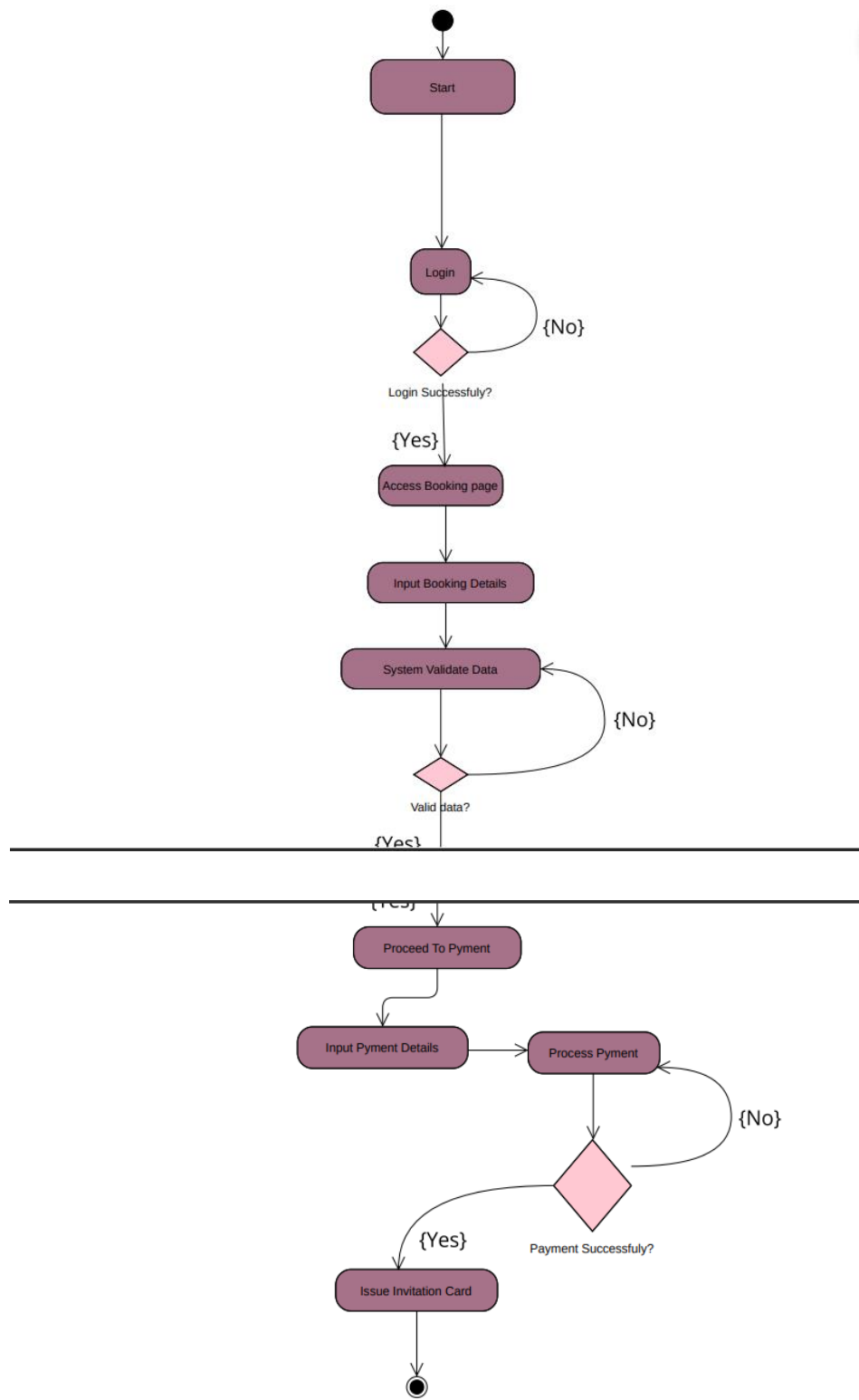


Figure 23: Activity diagram.

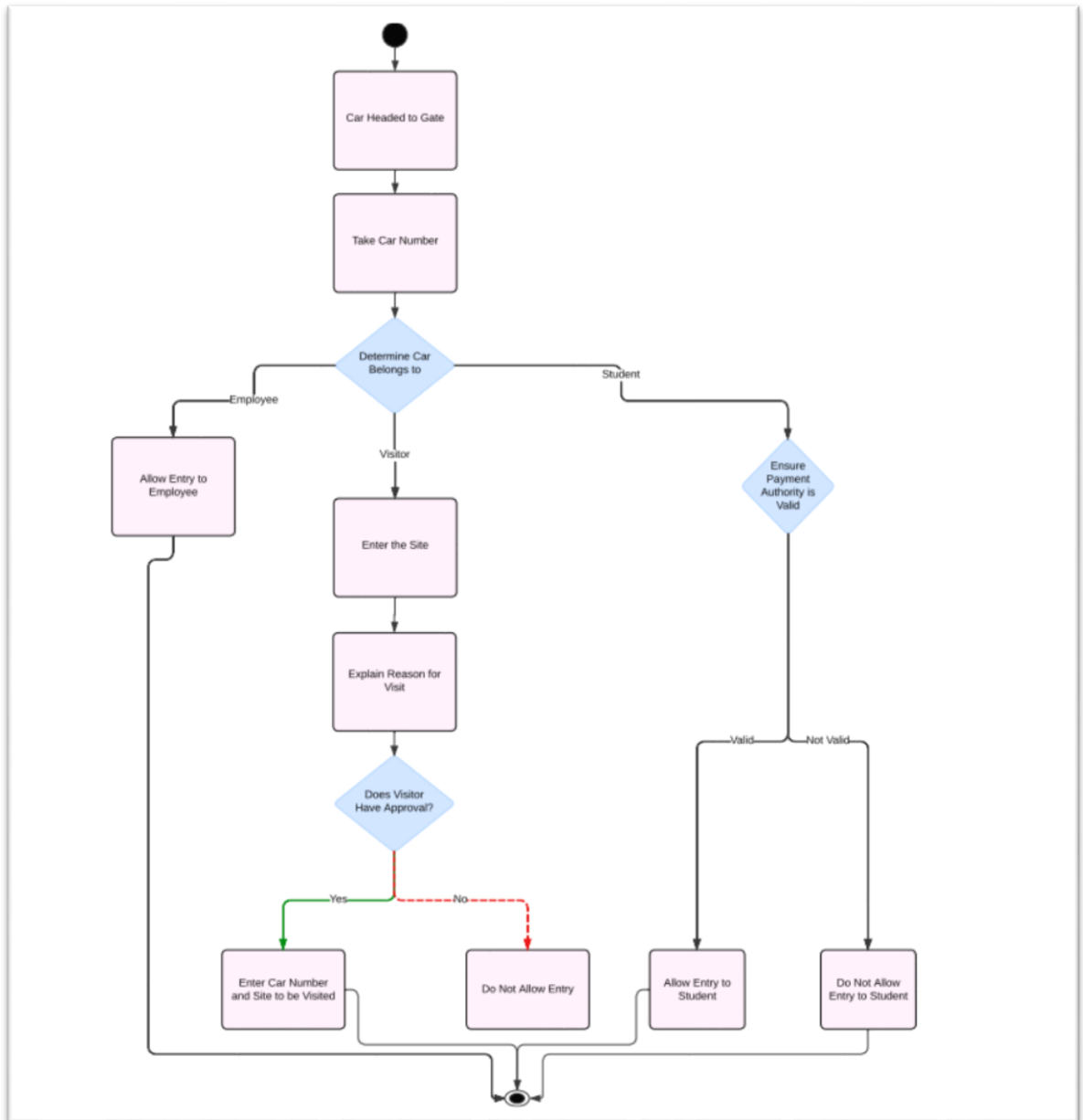


Figure 24: Activity diagram.

## 5.2.4 Data Storage Organization

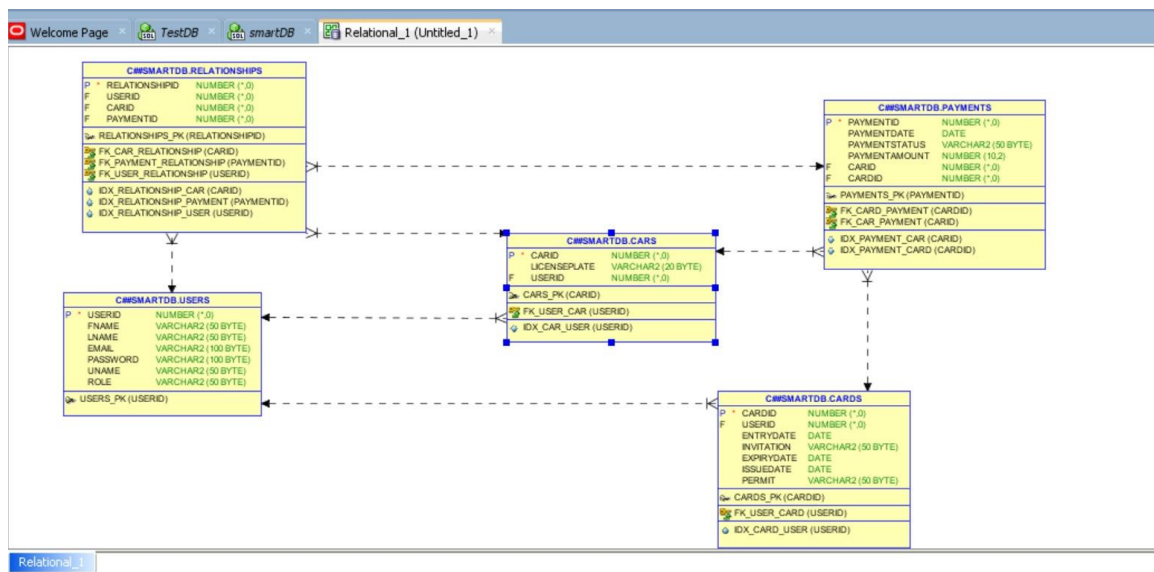


Figure 25: Database Table.

## ER Diagram:

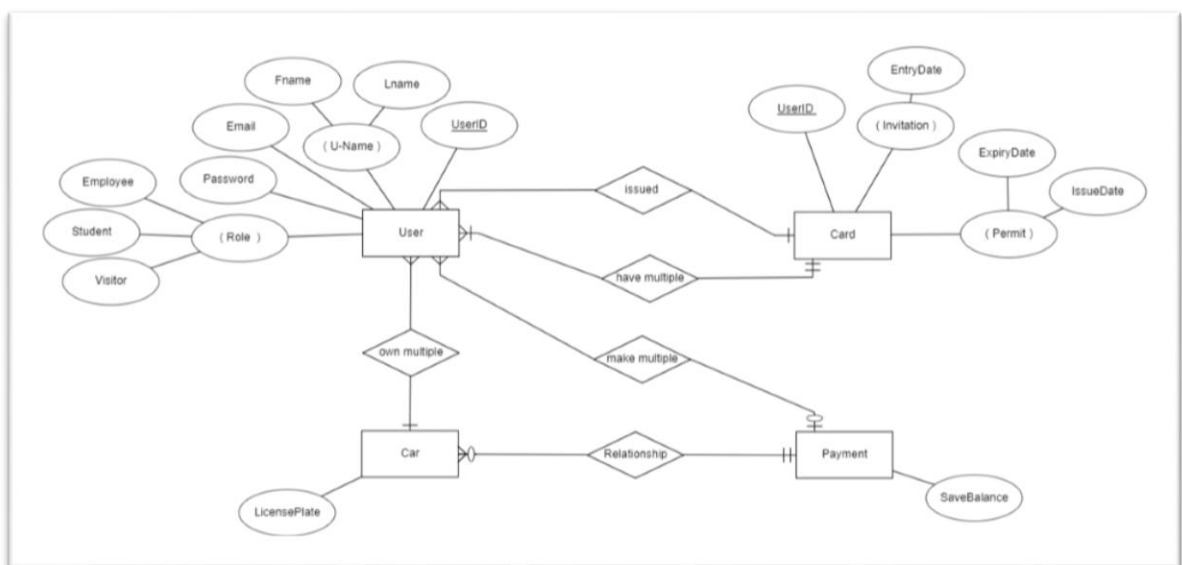


Figure 26: ER diagram.

## Database Code:

```
CREATE TABLE Users (  
    UserID INT PRIMARY KEY,  
    FName VARCHAR2(50),  
    LName VARCHAR2(50),  
    Email VARCHAR2(100),  
    Password VARCHAR2(100),  
    UName VARCHAR2(50),  
    Role VARCHAR2(50),  
    CONSTRAINT role_check CHECK (Role IN ('Employee', 'Student', 'Visitor'))  
);  
  
CREATE TABLE Cars (  
    CarID INT PRIMARY KEY,  
    LicensePlate VARCHAR2(20),  
    UserID INT,  
    CONSTRAINT fk_user_car FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);  
  
CREATE TABLE Cards (  
    CardID INT PRIMARY KEY,  
    UserID INT,  
    EntryDate DATE,  
    Invitation VARCHAR2(50),  
    ExpiryDate DATE,  
    IssueDate DATE,  
    Permit VARCHAR2(50),  
    CONSTRAINT fk_user_card FOREIGN KEY (UserID) REFERENCES Users(UserID)  
);  
  
CREATE TABLE Payments (  
    PaymentID INT PRIMARY KEY,  
    PaymentDate DATE,  
    PaymentStatus VARCHAR2(50),  
    PaymentAmount DECIMAL(10, 2),  
    CarID INT,  
    CardID INT,  
    CONSTRAINT fk_car_payment FOREIGN KEY (CarID) REFERENCES Cars(CarID),  
    CONSTRAINT fk_card_payment FOREIGN KEY (CardID) REFERENCES Cards(CardID)  
);  
  
CREATE TABLE Relationships (  
    RelationshipID INT PRIMARY KEY,  
    UserID INT,  
    CarID INT,  
    PaymentID INT,  
    CONSTRAINT fk_user_relationship FOREIGN KEY (UserID) REFERENCES Users(UserID),  
    CONSTRAINT fk_car_relationship FOREIGN KEY (CarID) REFERENCES Cars(CarID),  
    CONSTRAINT fk_payment_relationship FOREIGN KEY (PaymentID) REFERENCES Payments(PaymentID)  
);  
  
CREATE INDEX idx_car_user ON Cars(UserID);  
CREATE INDEX idx_card_user ON Cards(UserID);  
CREATE INDEX idx_payment_car ON Payments(CarID);  
CREATE INDEX idx_payment_card ON Payments(CardID);  
CREATE INDEX idx_relationship_user ON Relationships(UserID);  
CREATE INDEX idx_relationship_car ON Relationships(CarID);  
CREATE INDEX idx_relationship_payment ON Relationships(PaymentID);
```

# **CHAPTER 6: IMPLEMENTATION PLAN & PROTOTYPING**

## **6.1 Introduction**

**Tools, third-party applications, and languages required to create a water ball game.**

**1 - language: HTML, CSS, JS, PHP & Oracle.**

**2 - Database: mysql .**

**3 - Design: Figma.**

**4 - Website: Visual Studio.**

**Software Requirements:**

**Development Tools:**

**Website: Visual Studio.**

**Programming Languages: HTML, CSS, JS & PHP.**

**Graphics and Design Software: Figma**

## 6.2 Prototyping

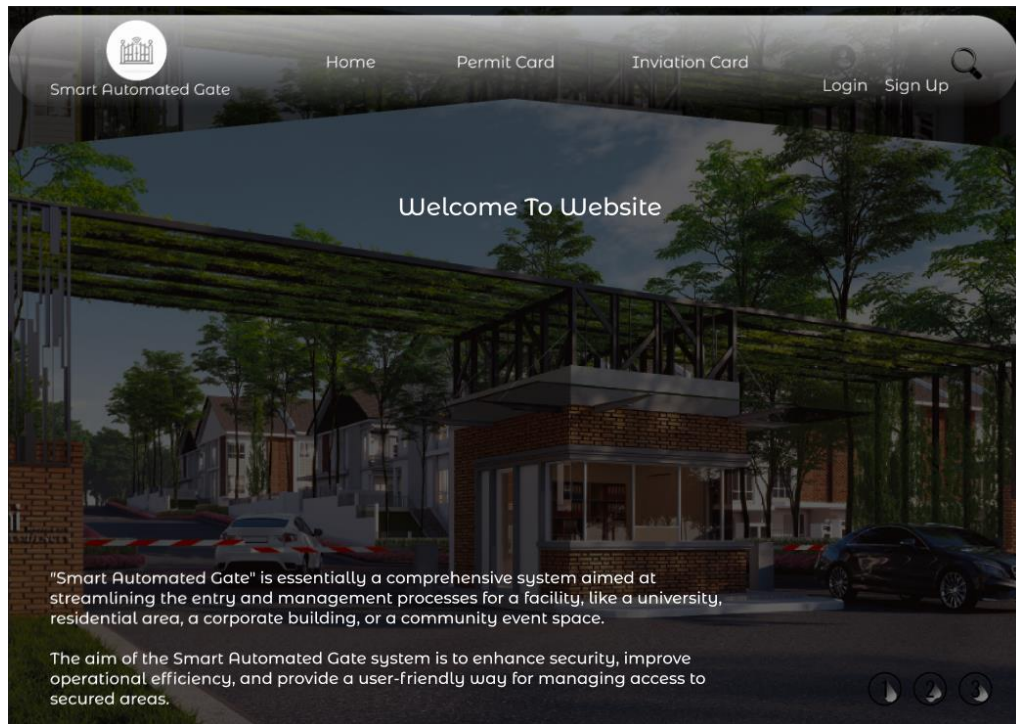


Figure 27: Home Page1.

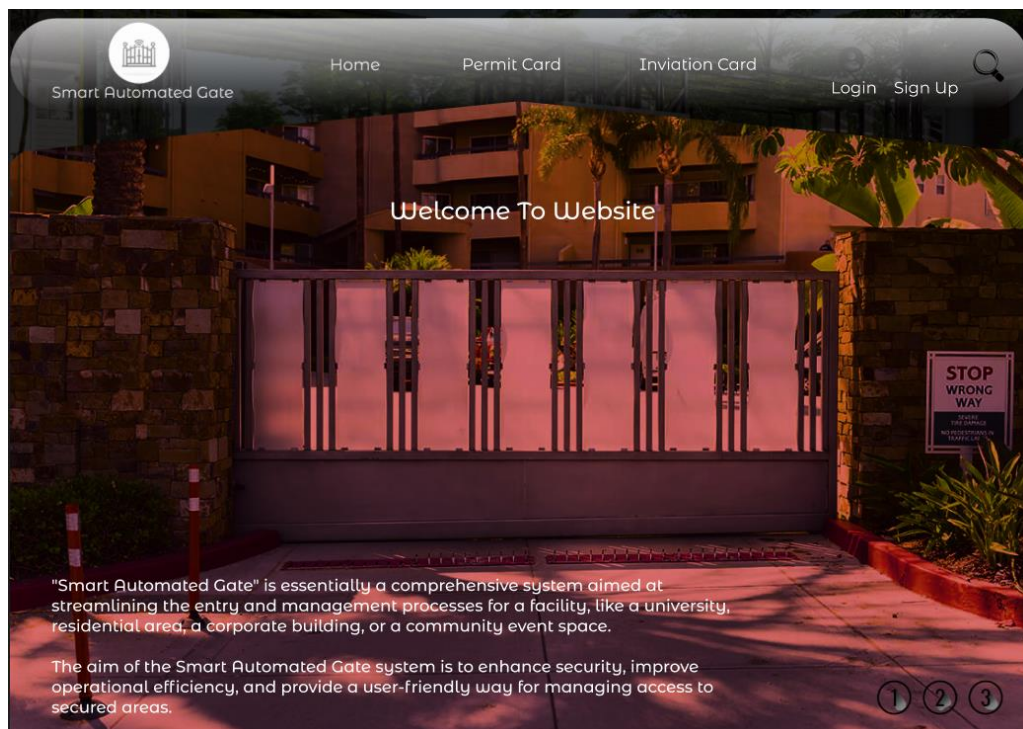


Figure 28: Home Page2.



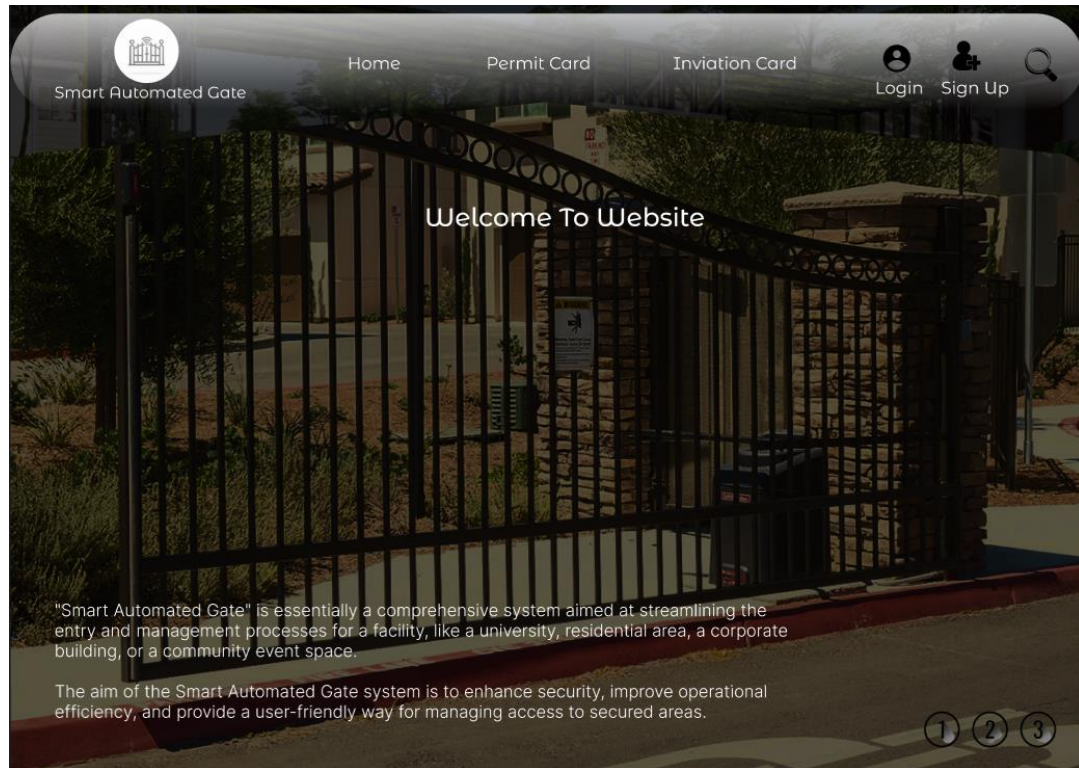


Figure 29: Home Page3.

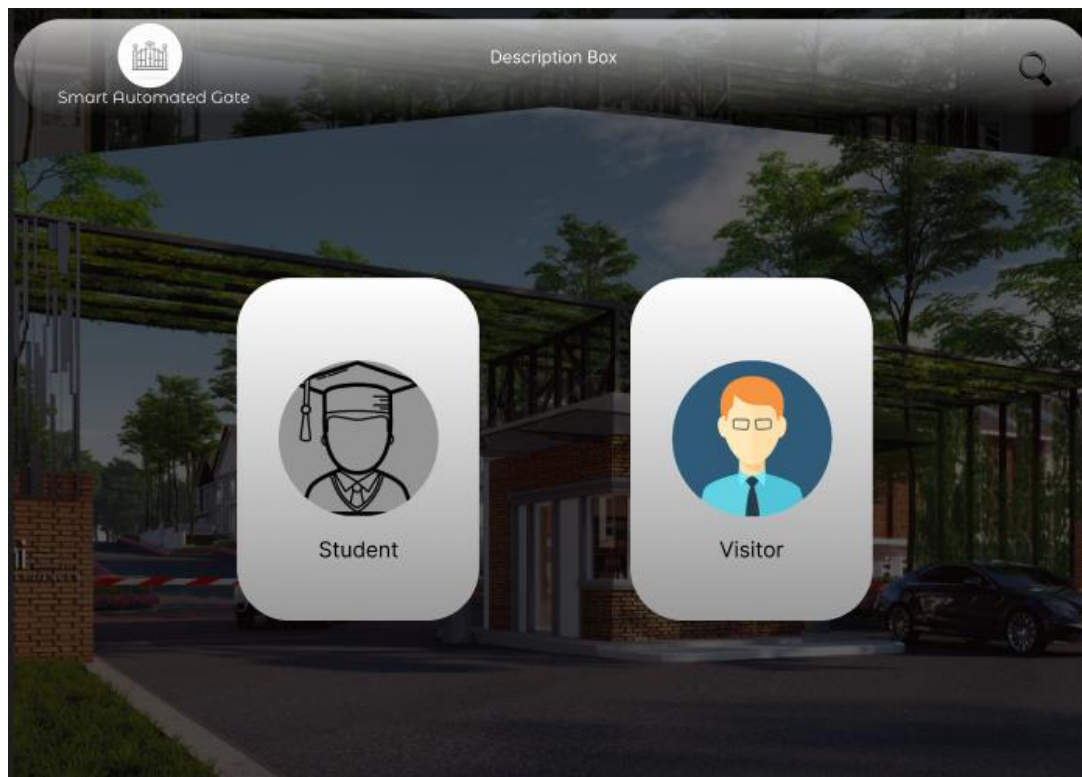


Figure 30: Choose Student or Visitor.



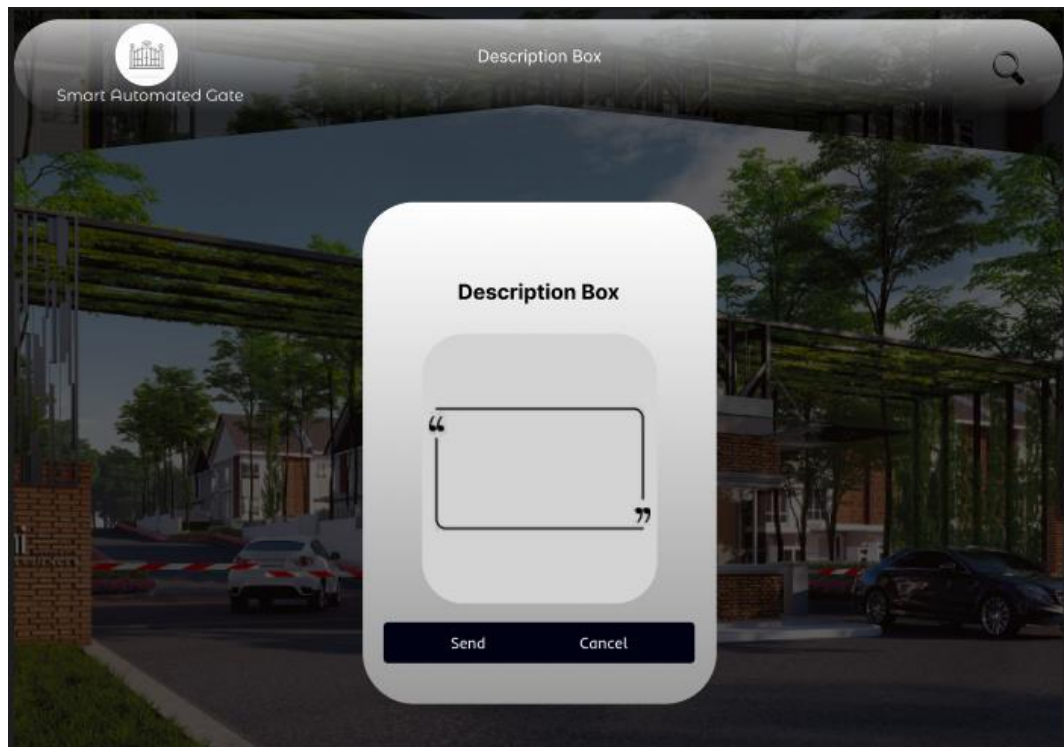


Figure 31: Description Box.

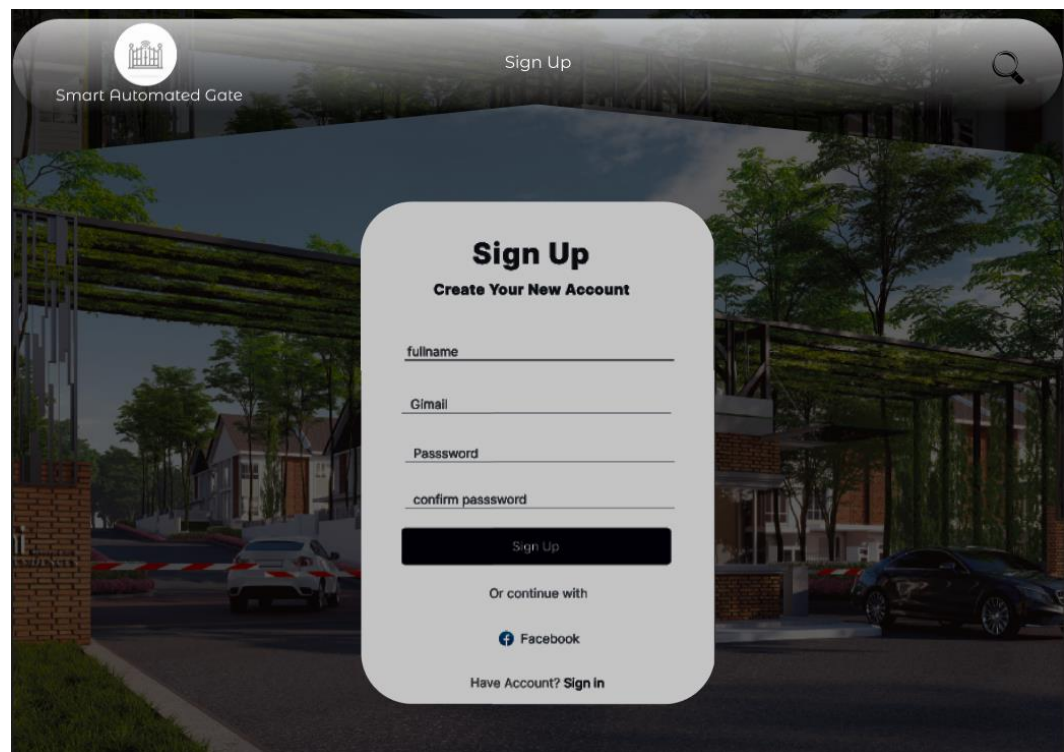


Figure 32: Sign Up Page.

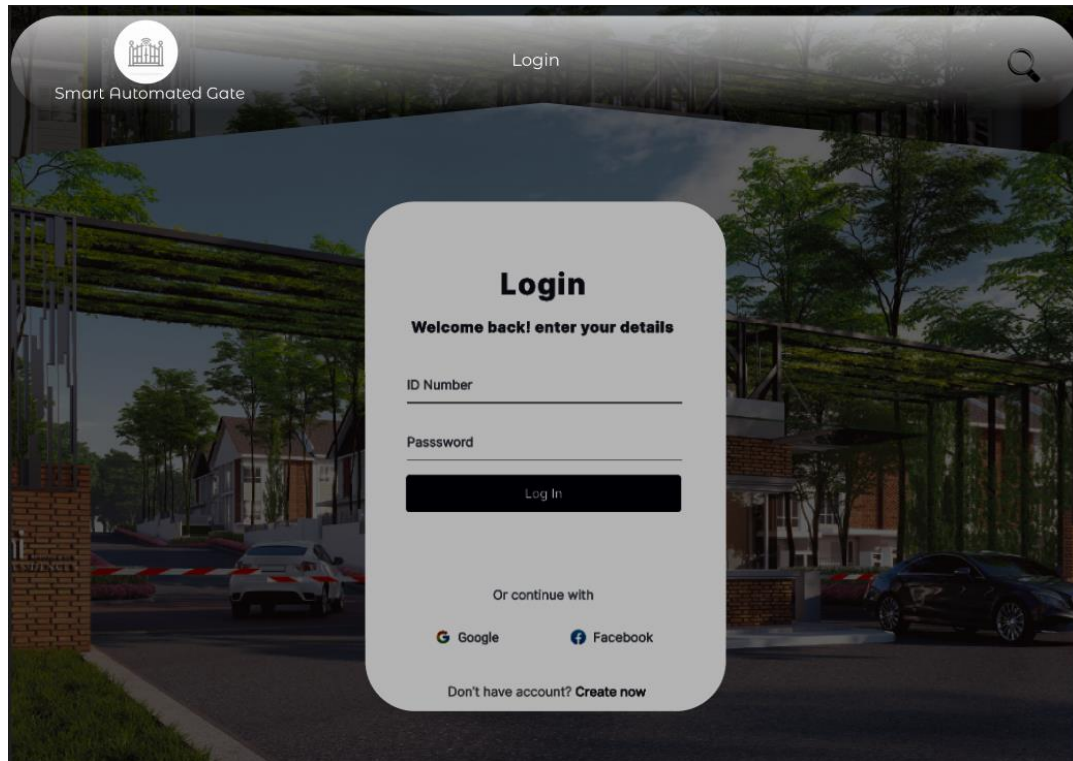


Figure 33: Log in Page.

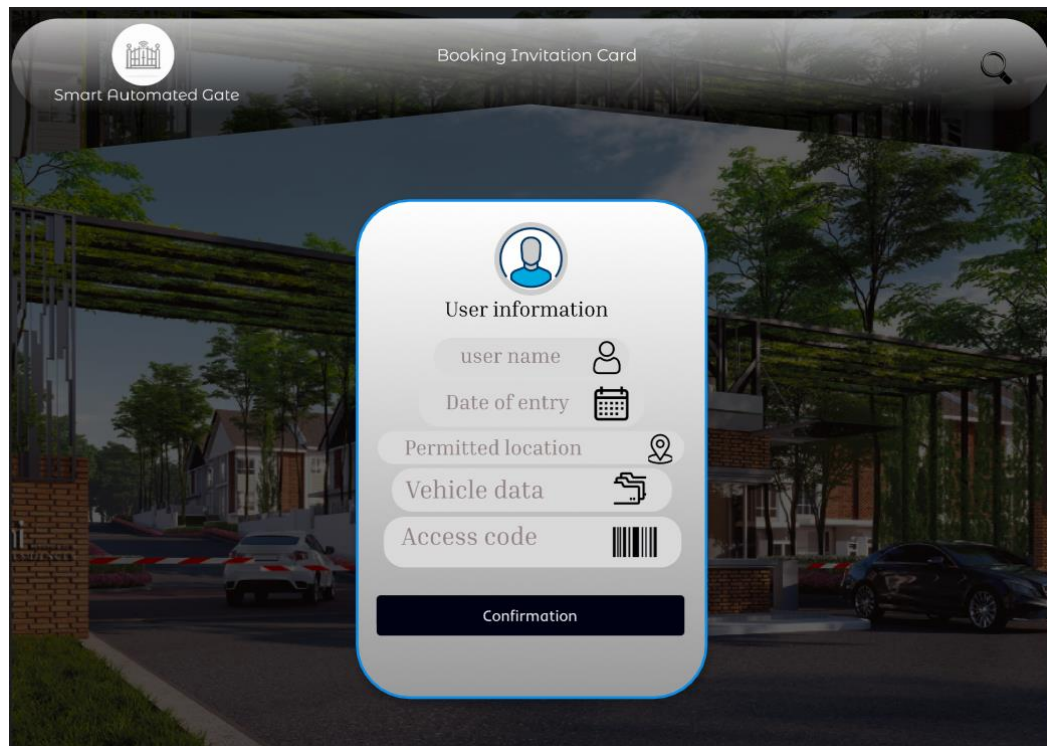


Figure 34: Booking Invitation Page – User Information.

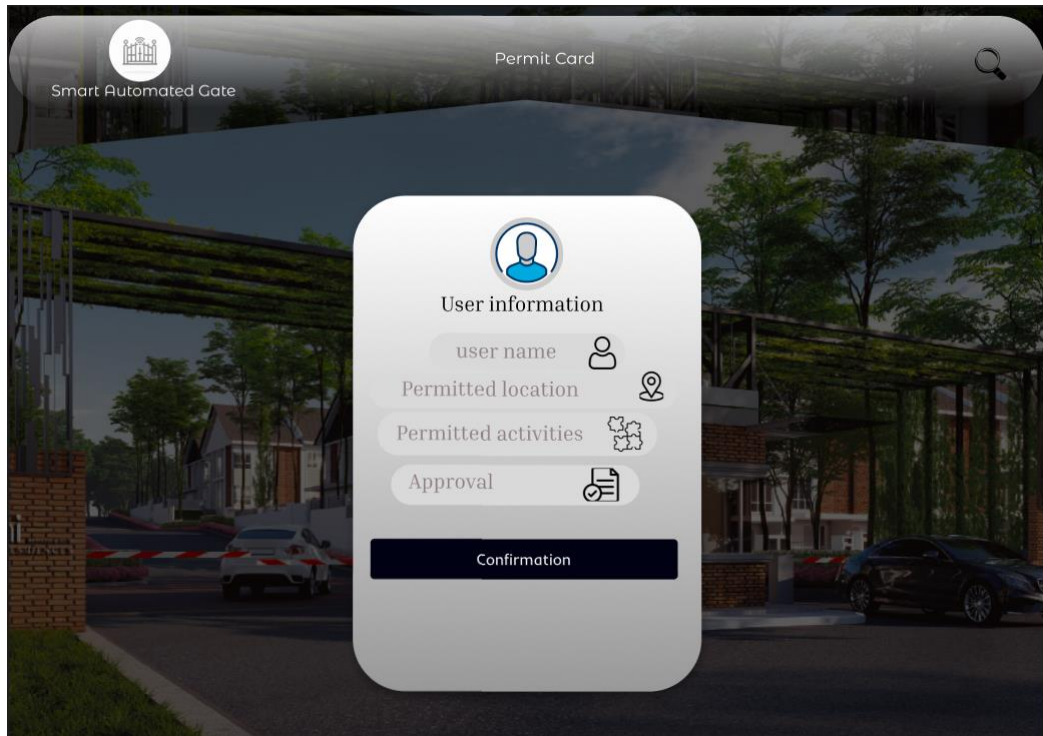


Figure 35: Permit Card Page – Employee Information.

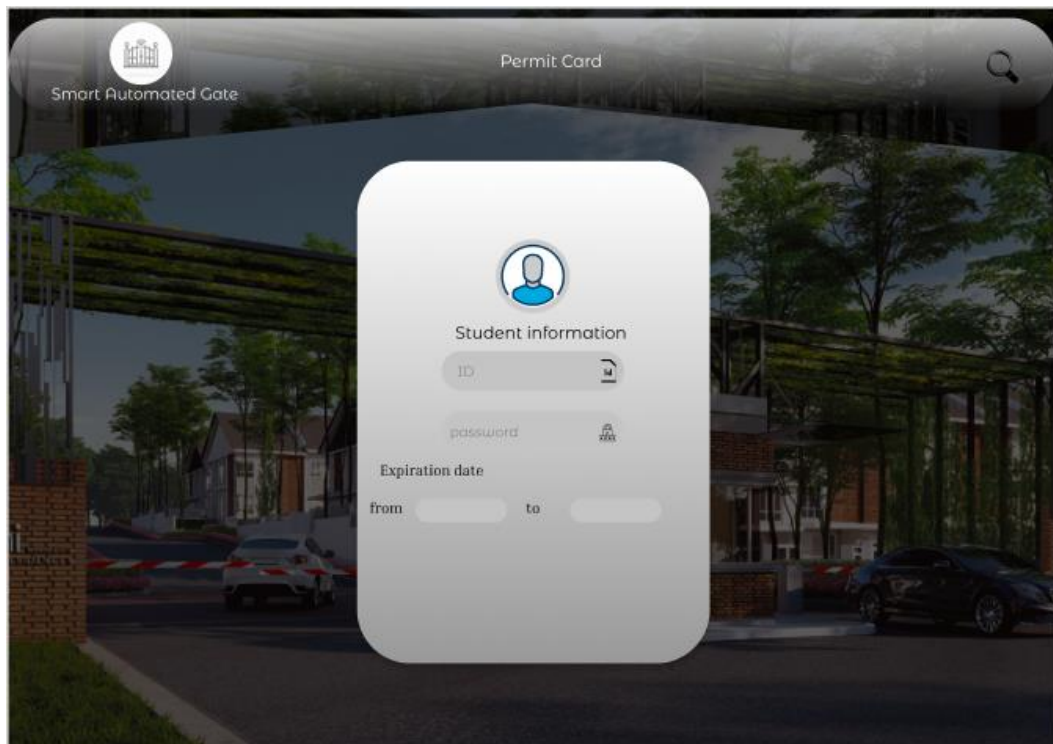


Figure 36: Permit Card Page – Student Information.



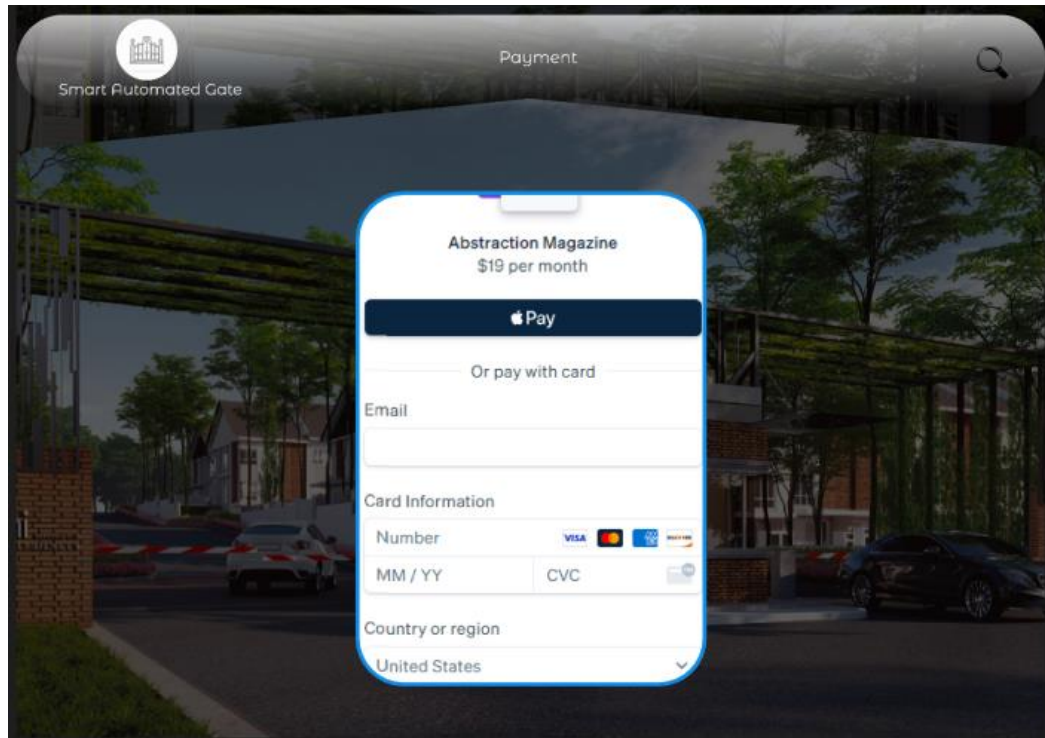


Figure 37: Payment Page – Student.

# CHAPTER 7: TESTING PLAN

## 7.1 Tools

### 1. Unit Testing Tools

pytest (for Python): These are popular frameworks for writing and running unit tests. They help verify the functionality of individual components in isolation.

### 2. Integration Testing Tools

TestNG or Selenium: Useful for integration testing where components are tested as a group to ensure they work together as expected.

### 3. Functional Testing Tools

Selenium: Ideal for automating web application testing across various browsers. Selenium can simulate user interactions with your web pages to ensure all functionalities are performing as intended.

Cypress: An alternative to Selenium, offering more straightforward setup and faster execution for modern web applications.

### 4. Load Testing Tools

JMeter or LoadRunner: These tools help simulate multiple users accessing your website simultaneously to test the server's capacity and response time under heavy load.

### 5. Security Testing Tools

OWASP ZAP or Burp Suite: Security tools to help identify vulnerabilities in your web application like SQL injection, XSS, etc.

### 6. API Testing Tools

Postman or SoapUI: Great for testing REST and SOAP APIs, ensuring that your backend services work correctly and efficiently.

## 7. User Interface Testing Tools

Appium (for mobile applications) or Selenium (for web applications): These are essential for ensuring that the user interface behaves as expected across different devices and platforms.

## 8. Performance Testing Tools

Gatling or Apache Bench: These tools are used to measure the performance of your site's components under various conditions.

## 7.2. System Testing Plan (Black box)

### 1. Log in for User & visitor:

Condition	Tag	Valid partition	Valid test case	Tag	Invalid partition	Tag	Invalid test Case
<b>Email</b>	<b>VE1</b>	12 to 30	omar_2002@gmail.com	<b>INE1</b>	<12 or >30	<b>INT1</b>	<a href="mailto:He@gmail.com">He@gmail.com</a>
	<b>VE2</b>	a-z or 0-9 or (_) only once		<b>INE2</b>	Doesn't have any Character	<b>INT2</b>	<a href="mailto:1023@5.77">1023@5.77</a>
	<b>VE3</b>	Has @ only once		<b>INE3</b>	Doesn't have (@)	<b>INT3</b>	Omar.gmail.com
	<b>VE4</b>	Has (.) only once		<b>INE4</b>	Doesn't have (.)	<b>INT4</b>	Mohammed@gmailcom
	<b>VE5</b>	Has a-z or 0-9 before @		<b>INE5</b>	Doesn't have any letters before (@)	<b>INT5</b>	<a href="mailto:2323@gmail.com">2323@gmail.com</a>
	<b>VE6</b>	Has a-z After (@) and (.)		<b>INE6</b>	Doesn't have any letters After (@) and (.)	<b>INT6</b>	najat@166.88

Table14: Equivalence partitioning (Email).

Condition	Tag	Valid boundaries	Tag	Valid test case	Tag	Invalid boundaries	Invalid test case
Email	VE1	12	VT1	<a href="mailto:H@gmail.com">H@gmail.com</a>	INE1	11	<a href="mailto:@gmail.com">@gmail.com</a>
	VE2	30	VT2	memoo2001Hamza1998Ezz2000@gmail.com	INE2	31	memoo2001Hamza1998Ezzzz2000@gmail.com

Table15: Boundary Value analysis (Email).

Condition	Tag	Valid partition	Valid test case	Tag	Invalid partition	Tag	Invalid test Case
Password	VP1	a-z / A-Z char 6 to 20	Omar-H@1998	INP1	Doesn't have any character	INT1	32545212313
	VP2	Has at least 1 capital latter		INP2	<6 or >20	INT2	Oma99
	VP3	Has at least 1 small latter		INP3	No capital letters	INT3	omarh@11
	VP4	Has at least 1 symbol		INP4	No small letters	INT4	OMARH@99
	VP5			INP5	No symbol	INT5	Moham2004

Table16: Equivalence partitioning (Password).



Condition	Tag	Valid boundaries	Tag	Valid test case	Tag	Invalid boundaries	Tag	Invalid test case
Password	VB 1	6	VT1	Om@20	INB1	5	INT1	Oa-2
	VB 2	20	VT2	Oma-Hasan-201@Lubani	INB2	21	INT2	OmarrHasanLubani20021

Table17: Boundary Value analysis (Password).

Condition	Tag	Valid partition	Valid test case	Tag	Invalid partition	Tag	Invalid test Case
<b>User Name</b>	<b>VP1</b>	a-z & A-Z char & digit	Omarr20	<b>INP1</b>	Non Alphabetical Characters & digit	<b>INT1</b>	--@
	<b>VP2</b>	4 to 25		<b>INP2</b>	<4 or >25	<b>INT2</b>	O8
				<b>INP3</b>	Contains symbol	<b>INT3</b>	omarr@8

Table18: Equivalence partitioning (Username).

Condition	Tag	Valid boundaries	Valid test case	Tag	Invalid boundaries	Tag	Invalid test case
<b>User Name</b>	<b>VB1</b>	4	Hoa9	<b>INB1</b>	3	<b>INT1</b>	O1a
	<b>VB2</b>	25	OmarMohammed Alubani200220	<b>INB2</b>	26	<b>INT2</b>	OomarMohammedAlubani200220

Table19: Boundary Value analysis (username).

## CONCLUSIONS

"Smart Automated Gate" is essentially a comprehensive system aimed at streamlining the entry and management processes for a facility, like a university, residential area, a corporate building, or a community event space.

The aim of the Smart Automated Gate system is to enhance security, improve operational efficiency, and provide a user-friendly way for managing access to secured areas.

## REFERENCES

- [1] Babineau W., Barry P., Furness Z., "Automated Testing within the Joint Training confederation (JTC)", Proceedings of the Fall 1998 Simulation Interoperability Workshop, Orlando, FL, USA. September 1998.
- [2] Banks C. "Introduction to Modeling and Simulation". Chapter 1 in book "Modeling and Simulation Fundamentals: Theoretical Underpinnings and Practical Domains". Catherine Banks, John Sokolowski Editors. Wiley. New Jersey, 2010.
- [3] Booth D., Haas H., McCabe F., Newcomer E., Champion M., Ferris C., Orchard D. "Web Services Architecture". 2004. <<http://www.w3.org/TR/ws-arch/>>. Accessed November 2010.
- [4] The Tool that we used to draw Diagrams(<https://www.drawio.com/>).

## **APPENDIX-A: MANUAL**