University of Belize

Report #1: Gated Community Application

Group #1

Kelsey A., Luigi A., Jordani A., Arthur B., Aiyesha C., Aiden P.

University of Belize

CMPS4131- Software Engineering

Mr. Medina Manuel

March 5, 2025

Table of Contents

Customer Statement of Requirements (CSR)

Problem Statement

Security Personnel's Perspective

Currently, security personnel face significant challenges with the manual visitor management system. Visitors are recorded by hand in logbooks, which leads to inefficiencies and potential errors. Each time a visitor arrives, their details must be manually written down, causing delays, especially during peak hours. This results in long queues and a slower check-in process, making it difficult for security personnel to manage the flow of visitors effectively.

A critical issue arises during identity verification. If a host is unavailable or unresponsive to phone calls, security personnel cannot easily verify if a visitor is authorized to enter. This lack of confirmation poses a security risk and creates delays. Even when the host is available, the reliance on manual approval processes wastes time and can lead to critical delays, particularly in urgent situations when quick decisions are needed.

Furthermore, the paper-based system is prone to human error. Inaccurate entries and unclear handwriting make it difficult to track visitors properly. Retrieving past records is time-consuming and inefficient, especially in situations where quick access to information is needed, such as in emergency response scenarios.

The security personnel also lack real-time access to a list of visitors on-site. This makes it challenging to determine whether a person is authorized to enter the building or if there is someone unauthorized on the premises. Without this instant access to visitor data, security becomes vulnerable to potential breaches. The situation worsens when security has to rely on hosts for entry approvals. If a host is unavailable or does not respond in time, the visitor is left waiting, and security is unable to make a timely decision, compromising both efficiency and security.

Security Staff Perspective

The manual system currently in place is outdated and inefficient, relying heavily on human effort for tasks that could be automated. Paper logbooks and the need for phone calls to hosts slow down the visitor entry process significantly. During peak hours, security personnel struggle to manage the visitor flow effectively, as the manual system cannot keep up with the volume of visitors.

Security staff are particularly challenged by the lack of real-time access to the list of visitors currently on-site. Without this immediate data, it is difficult for security personnel to verify if someone is authorized to enter or if there is a person on the premises who should not be there. The inability to access visitor data instantly creates a vulnerability that could potentially lead to security breaches. This issue is compounded when approval from hosts is required. The delays caused by waiting for host responses make security staff dependent on the hosts' availability, which can be unreliable.

Security personnel also find the process of retrieving past visitor records cumbersome. The paper-based logbooks make it time-consuming to look up previous records, which becomes especially problematic when quick access to past data is necessary in emergencies. This lack of efficient record retrieval hampers the security staff’s ability to act quickly and decisively in critical situations.

Visitor’s Perspective

Visitors, whether they are family members, delivery personnel, service providers, or other types of guests, often find the current system frustrating and time-consuming. First-time visitors are required to fill out lengthy registration forms, which adds unnecessary time to the check-in process. Even regular visitors are inconvenienced by the need to provide the same information every time they arrive, making the process feel redundant and inefficient.

One of the major frustrations is the waiting time when the host is unavailable to approve entry. If the host does not answer the phone, visitors are left uncertain about whether they will be allowed in, which adds stress and prolongs their wait. Visitors often find themselves in limbo, unsure of

the next steps, especially if they do not have prior approval. This uncertainty leads to confusion and inconvenience, creating a negative experience for the visitors.

Moreover, for visitors arriving without prior approval, the entire experience becomes even more chaotic. They are left unsure about whether they will be allowed in, leading to delays and confusion. The lack of a streamlined and predictable process makes it difficult for visitors to know what to expect, causing frustration and dissatisfaction.

To address the challenges outlined, we envision a system that will significantly enhance efficiency, reduce human error, and improve overall security. The proposed system will include pre-registration, allowing visitors to be registered before their arrival. This will eliminate the need for manual data entry each time a visitor arrives, speeding up the check-in process and particularly benefiting frequent visitors. The system will also provide security personnel with real-time access to data on visitors currently on-site, enabling immediate verification of whether a visitor is authorized to enter. Technologies such as QR code scanning or facial recognition can be implemented to facilitate quicker and more accurate identification of visitors, ensuring faster and safer entry.

With digital approval processes, hosts will be able to approve or deny visitor entry directly from their mobile devices, eliminating the need for phone calls and reducing approval delays. This will allow security personnel to handle approvals on the spot, speeding up the entry process and minimizing waiting times. The system will include real-time alerts for unauthorized entry attempts or potential security threats, allowing security staff to respond swiftly to any incidents. Additionally, digital visitor records will be stored securely, providing easy and efficient access to past visitor data, which will be crucial in emergency situations.

To ensure security, efficiency, and compliance, the visitor management system will implement key policies. Data protection will be a priority, safeguarding visitor information per regulations like GDPR. Authorization levels will restrict sensitive actions to security personnel and designated hosts, reducing unauthorized access. Visitors will be categorized (e.g., family, delivery personnel) for a smoother check-in process. These policies will enhance security, streamline operations, and improve the visitor experience.

Glossary of Terms

1. **Access Control –** The process of restricting entry to a facility based on predefined authorization levels.
2. **Authorization Levels –** System-defined access control measures that restrict specific actions (e.g., approving visitors, viewing confidential details) to authorized security personnel and designated hosts.
3. **Check-in Process** – The procedure visitors follow upon arrival, including identity verification, host approval, and registration.
4. **Compliance –** Adherence to regulations and policies (such as GDPR) to ensure the secure processing and storage of visitor data.
5. **Data Protection** – Measures implemented to safeguard visitor information from unauthorized access or breaches.
6. **Digital Approval Process –** A system that allows hosts to approve or deny visitor entry remotely via mobile devices, eliminating reliance on phone calls and reducing delays.
7. **Emergency Response –** The ability of security personnel to quickly retrieve visitor records and access real-time data to act promptly in urgent situations.
8. **Identity Verification –** The process of confirming a visitor's authorization to enter the premises, which may include QR code scanning, facial recognition, or host approval.
9. **Manual Visitor Management System –** The traditional paper-based method of recording visitor details in logbooks, which leads to inefficiencies, errors, and security risks.
10. **Pre-registration –** A feature that allows visitors to register before arrival, reducing manual data entry and expediting the check-in process.
11. **Queue Management** – The process of reducing visitor wait times by implementing efficient check-in procedures.
12. **Real-time Access –** The ability of security personnel to instantly view the list of visitors currently on-site for quick verification and decision-making.
13. **Security Breach –** An incident where an unauthorized individual gains access to a restricted area due to system vulnerabilities or procedural failures.
14. **Security Personnel –** Staff responsible for managing visitor entry, verifying identity, ensuring compliance, and responding to security threats.

15. **Security Threats –** Potential risks such as unauthorized entry attempts, unverified visitors, or delays in security processes that may compromise safety.
16. **Self-check-in** – A feature that allows visitors to complete the check-in process independently using a digital kiosk or mobile app.
17. **Streamlined Check-in Process –** A system that categorizes visitors (e.g., family, delivery personnel) for faster and more efficient entry, particularly benefiting frequent visitors.
18. **Unauthorized Entry Attempt** – When an individual without proper approval tries to access a restricted area.
19. **Visitor Categories –** Groups such as family members, delivery personnel, and service providers, each with tailored entry requirements to facilitate a smooth check-in process.
20. **Visitor Management System (VMS) –** A proposed digital system designed to improve visitor tracking, reduce errors, and enhance security by automating registration, identity verification, and approval processes.

System Requirements

Priority Weight	Description
1	Not Important
2	Low Importance
3	Normal
4	Important
5	Very Import

Table 1. System Requirements Priority Scale

Our input for the system requirements was based on research from best practices and recommendations for gated communities. We reviewed multiple sources to ensure that our features align with the security, convenience, and efficiency needs of gated communities (PalAmerican Security, 2021).

Enumerated Functional Requirements

REQ-X	Priority Weight	Description
REQ-1	1	The system shall allow all users (administrators, residents, and security guards) to log in using a username and password.
REQ-2	4	The system shall allow residents and security guards to communicate via notification system.
REQ-3	3	The system shall allow visitors to provide feedback after their visit.
REQ-4	4	The system shall allow residents and security to report visitors for blacklisting.
Admin		
REQ-5	5	The system shall allow administrators to approve blacklist requests.
REQ-6	4	The system shall allow administrators to view resident and security guard information.
REQ-7	4	The system shall allow administrators to add resident and security guard residents.
REQ-8	4	The system shall allow administrators to edit resident and security guard information.

REQ-9	5	The system shall allow administrators to delete resident and security guard records.
REQ-10	5	The system shall allow administrators to remove visitors from the blacklist if necessary.
Security Personnel		
REQ-11	4	The system shall allow security guards to view visitor history.
REQ-12	5	The system shall allow security guards to scan QR codes.
REQ-13	4	The system shall allow security guards to scan the next QR code as people enter.
REQ-14	4	The system shall allow security guards to verify recurring visitor QR codes against the numbers of attendees.
REQ-15	5	The system shall allow security guards to log any incidents or concerns about visitors.
Residence		
REQ-16	4	The system shall allow residents to view and edit visitor information.
REQ-17	4	The system shall allow residents to add visitor details.
REQ-18	5	The system shall allow residents to generate QR codes for visitors.
REQ-19	5	The system shall allow residents to send QR codes to visitors.
REQ-20	4	The system shall allow residents to register recurring visitors with predefined schedules.
REQ-21	4	The system shall allow residents to modify or revoke access for recurring visitors at any time.
REQ-22	4	The system shall allow residents to generate one QR code for multiple event guests.
REQ-23	3	The system shall allow residents to rate visitor experiences.

Table 2. Functional Requirements

Enumerated Nonfunctional Requirements

NONREQ-X	Priority Weight	Description	FURPS+
NONREQ-1	1	The system shall lock an account for 30 minutes after five (5) failed login attempts.	Security
NONREQ-2	4	The system shall ensure that only authorized residents can generate and send QR codes.	Security
NONREQ-3	5	The system shall log all user activities, including resident updates and security guard scans.	Security
NONREQ-4	1	The system should only allow one instance of a login for an account.	Security
NONREQ-5	5	The system shall ensure that all data is kept secure to protect residents' and visitors' information.	Security
NONREQ-6	2	The system shall work smoothly on both mobile and desktop devices.	Performance
NONREQ-7	5	The system shall allow access only if the entered credentials match those in the database.	Security
NONREQ-8	4	The system shall have an intuitive user interface (UI) for easy navigation across all user roles.	Usability
NONREQ-9	5	The system shall allow QR code scanning only if the visitor is registered and meets access conditions.	Security
NONREQ-10	3	The system shall be available at least 99% of the time to ensure users can always access it.	Reliability
NONREQ-11	3	The system shall provide a "Remember Me" option for user convenience to login (excluding security guards).	Usability
NONREQ-12	5	The system shall deliver messages or alerts between residents and security guards within 5 seconds to ensure timely communication.	Reliability
NONREQ-13	5	The system shall ensure QR codes for recurring visitors expire after their scheduled time.	Reliability
NONREQ-14	3	The system shall notify residents when their recurring visitor successfully checks in or is denied entry.	Reliability

NONREQ-15	3	The system shall provide clear error messages when guests attempt to use an expired or invalid QR code.	Usability
NONREQ-16	3	The system shall ensure visitor feedback is submitted anonymously unless the user chooses otherwise.	Security
NONREQ-17	3	The system shall allow authorized users (residents, security, admins) to review and respond to feedback.	Usability
NONREQ-18	4	The system shall provide real-time alerts when a blacklisted visitor attempts to enter.	Security
NONREQ-19	5	The system shall ensure only authorized personnel can modify the blacklist.	Security

Table 3. Nonfunctional Requirements

On-Screen Appearance Requirements

ONSREQ-X	Priority Weight	Description
ONSREQ-1	4	Confirmation messages (e.g., "QR Code Scanned Successfully") shall be displayed in green for clarity.
ONSREQ-2	2	The system shall show a loading spinner when processing login, QR code scans, or data retrieval.
ONSREQ-3	5	The system must provide clear navigation options for users to easily access their assigned features.
ONSREQ-4	5	The Admin's and Security’s screen must be actively updated to show the latest resident and visitor logs.
ONSREQ-5	5	The Resident's screen must display visitor status updates, including pending and approved QR codes.
ONSREQ-6	2	The system must provide clear error messages after each user action.

Table 4. On-Screen Requirements

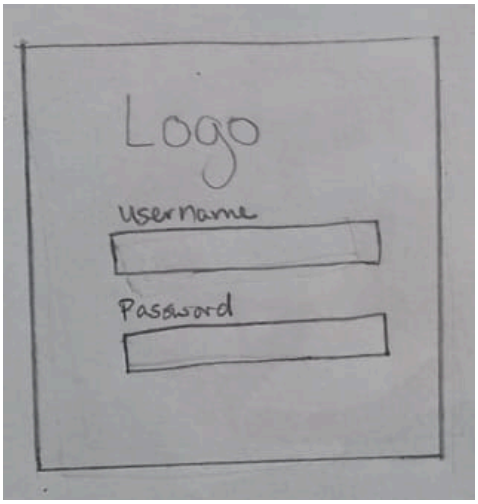

Fig 1. Login Screen

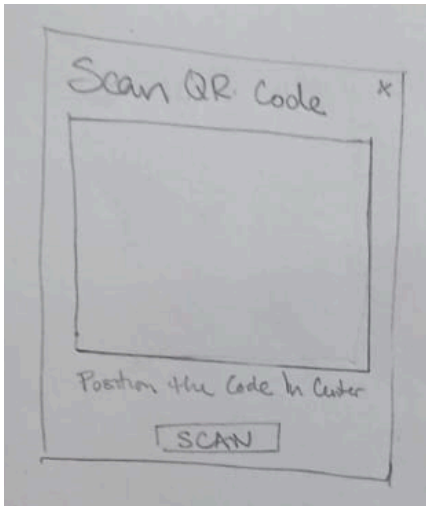

Fig 2. Scan QR Code

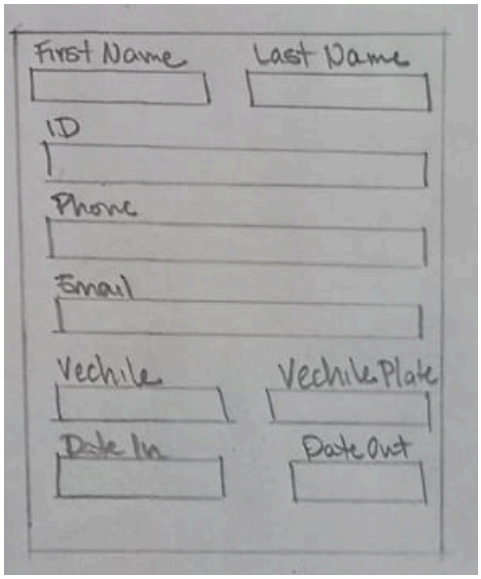

Fig 3. Adding A Visitor

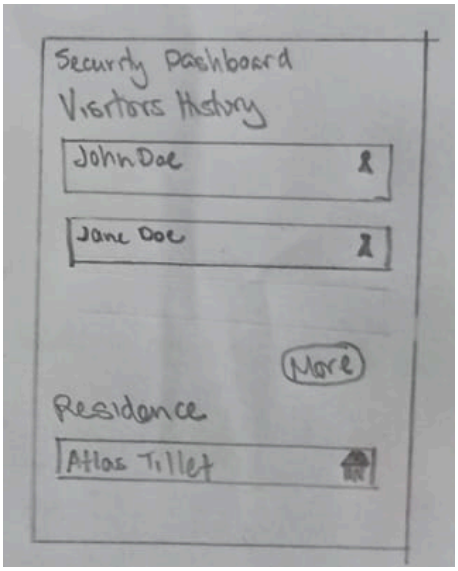

Fig 4. Security Dashboard

Our app features are inspired by Hope, Tzib and Shol's first prototype (2024). Some features will also be incorporated along with our team's new ideas.

Functional Requirements Specification

Stakeholders

Residents	Visitors	Security Guards
Administrators	Homeowners' Association	

Table 5. Stakeholders of the System

Actors

Actor	Roles	Type	Goals
Residents	● Manage visitor access by generating and sending QR codes, ● Registering recurring visitors, ● Reporting visitors for blacklisting, and ● Rating visitor experiences.	Initiating	● Ensure secure, controlled access for their visitors. ● Streamline visitor management and maintain oversight over guest activity. ● Provide feedback to improve system functionality and security.
Visitors	● The end-users who present QR codes to gain entry and provide feedback after their visit.	Initiating	● Gain authorized and hassle-free entry into the gated community. ● Communicate their visit experience for potential improvements.
Security Guards	● Verify visitor entries by scanning QR codes, ● Checking recurring visitor schedules, ● Logging incidents, and ● Monitoring the blacklist status.	Participating	● Maintain the safety and security of the community by ensuring only authorized visitors are admitted. ● Accurately log and report any incidents or anomalies.
Administrators	● Oversee system operations, ● Manage resident records, ● Approve or reject blacklist requests	Initiating	● Ensure the system operates securely, efficiently, and reliably. ● Maintain up-to-date and accurate records for residents and visitors.

	• Remove visitors from the blacklist when necessary.		• Enforce community policies and ensure compliance with security standards.

Table 6. Actor Types and Goals

Use Cases

Casual Description

UC Name	Actor	Actor's Goal	Req #
UC-1 Login	Resident/ Security Guard/ Administrator	To access the system using their login credentials.	REQ-1 NONREQ-1 NONREQ-4 NONREQ-7 NONREQ-11
UC-2 Create Visitor Schedule	Resident	To register a visit with a predefined schedule.	REQ-16 REQ-17 REQ-20 REQ-21 REQ-22 NONREQ-8 NONREQ-10 NONREQ-13
UC-3 Generate QR Code	Resident	To generate a QR code for an expected visitor.	REQ-18 REQ-22 NONREQ-2 NONREQ-8 NONREQ-10
UC-4 Send QRCode	Resident	To send a generated QR code to a visitor for entry.	REQ-19 NONREQ-2 NONREQ-8 NONREQ-10
UC-5 Send Notifications	System	To be informed when a visitor successfully checks in or is denied entry.	REQ-2 NONREQ-10 NONREQ-12 NONREQ-14
UC-6 Submit Feedback	Visitor	To provide feedback on their visit.	REQ-3 REQ-17

			REQ-23 NONREQ-15 NONREQ-16 NONREQ-17
UC-7 Scan QR	Security Guard	To scan a visitor's QR code and allow or deny entry based on access conditions.	REQ-12 REQ-13 REQ-14 NONREQ-6 NONREQ-8 NONREQ-9 NONREQ-13 NONREQ-14 NONREQ-15
UC-8 View History	Security Guard	To review past visitor entries.	REQ-11 NONREQ-3 NONREQ-8 NONREQ-14
UC-9 Manage Blacklist	Resident/ Security Guard	To report a visitor for an act that is not acceptable by the community or a specific resident.	REQ-4 REQ-15 NONREQ-8 NONREQ-18 NONREQ-19
UC-10 Add to Blacklist	Administrator/ Resident	To approve a blacklist request submitted by residents or security.	REQ-5 NONREQ-8 NONREQ-19
UC-11 Remove from Blacklist	Administrator	To remove a visitor from the blacklist when necessary.	REQ-10 NONREQ-8 NONREQ-19
UC-12 ManageUsers	Administrator	Manage user accounts, including creating new accounts, modifying account details, or deleting accounts.	REQ-6 REQ-7 REQ-8 REQ-9 NONREQ-5 NONREQ-8
UC-13 AddUser	Administrator	To register a new user in the system.	REQ-7 NONREQ-5 NONREQ-8
UC-14 EditUser	Administrator	To manage user details by editing records.	REQ-6 REQ-8

			NONREQ-5 NONREQ-8
UC-15 Delete User	Administrator	To manage user details by deleting records.	REQ-6 REQ-9 NONREQ-5 NONREQ-8
UC-16 View Feedback	Administrator/ Resident/ Security Guard	To view feedback submitted by visitors.	REQ-3 REQ-23 NONREQ-8 NONREQ-16 NONREQ-17

Table 7. Brief Description of Use Case

Use Case Diagram

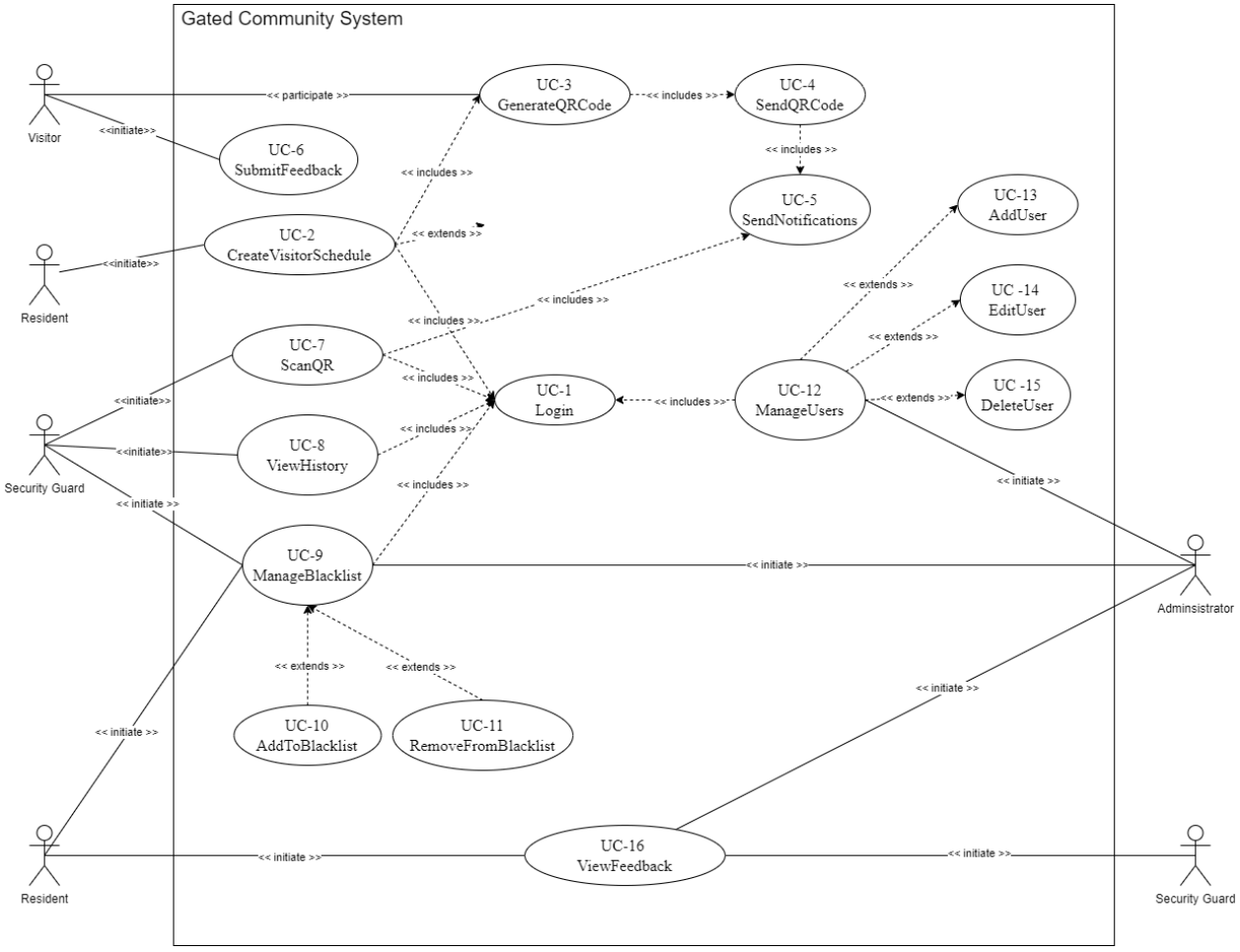

Fig 5. Use Case Diagram for the Gated Community System

Traceability Matrix

Use Cases		UC -1	UC -2	UC -3	UC -4	UC -5	UC -6	UC -7	UC -8	UC -9	UC -10	UC -11	UC -12	UC -13	UC -14	UC -15	UC -16
PW	Reqs																
1	REQ-1	X															
4	REQ-2					X											
3	REQ-3						X										X
4	REQ-4									X							
5	REQ-5										X						
4	REQ-6												X		X	X	
4	REQ-7												X	X			
4	REQ-8										X		X		X		
5	REQ-9												X			X	
5	REQ-10											X					
4	REQ-11								X								
5	REQ-12							X									
4	REQ-13							X									
4	REQ-14							X									
5	REQ-15									X							
4	REQ-16		X														
4	REQ-17		X				X										
5	REQ-18			X													
5	REQ-19				X						X						
4	REQ-20		X														
4	REQ-21		X														
4	REQ-22		X	X													
3	REQ-23						X										X
1	NONREQ-1	X															
4	NONREQ-2				X												
5	NONREQ-3								X								
1	NONREQ-4	X															
5	NONREQ-5												X	X	X	X	
2	NONREQ-6							X					X				
5	NONREQ-7	X															
4	NONREQ-8		X					X	X	X		X	X	X			X

5	NONREQ-9							X									
3	NONREQ-10		X	X	X	X											
3	NONREQ-11	X															
5	NONREQ-12					X											
5	NONREQ-13		X					X									
3	NONREQ-14					X		X	X								
3	NONREQ-15						X	X									
3	NONREQ-16						X										X
3	NONREQ-17						X										X
4	NONREQ-18									X							
5	NONREQ-19									X		X					
	Max PW	1	5	3	3	1	5	5	5	5	3	3	5	5	2	3	5
	Total PW	11	32	12	12	15	19	35	16	22	14	14	28	13	13	14	16

Table 8. Requirements Mapped to Use Cases

Fully-Dressed Description

Use Case UC-2	**CreateVisitorSchedule**
Related Requirements:	Functional Requirements: REQ-16, REQ-17REQ-20, REQ-21, REQ-22 Non-Functional Requirements: NONREQ-8,NON-REQ-10 NONREQ-13
Initiating Actor(s):	Resident
Actor's Goal:	Successfully register a visit session in the system with a predefined schedule.
Participating Actor:	Visitor
Preconditions:	● The resident must be logged into the system.
Post conditions:	● The visitor is successfully registered in the system ● The QR code was generated and sent to the visitor
Flow of Events for Main Success Scenario:	
→	1. Resident enters credentials: Include UC-1 Login
	2. Resident navigates to the visitor registration page.
	3. Resident enters valid visitor details, including visit frequency and duration.
←	4. System generates a persistent QR code for the visitor.
	6. System notifies the security team about the new recurring visitor: Include UC-5 SendNotifications
	7. Visitor details are stored in the database for future visits.
Flow of Events for Extensions (Alternate Scenarios):	
	3. The user enters invalid information for the user. a. The system notifies the user with an error message.

Table 9. Fully Dressed Description of CreateVisit

Use Case UC-6	**SubmitFeedback**
Related Requirements:	Functional Requirements: REQ-3, REQ-17, REQ-23 Non-Functional Requirements: NONREQ-15, NONREQ-16, NONREQ-17
Initiating Actor(s):	Visitor
Actor's Goal:	To provide feedback on their visit.
Participating Actor:	Database
Preconditions:	● Systems must have stored visitor check-in data
Post conditions:	● The feedback is successfully submitted and stored in the system.
Flow of Events for Main Success Scenario:	
→	1. Visitor scans the QR code.
←	2. The system retrieves the visitor's details and sends an email containing a feedback link.
→	3. The visitor opens the email and clicks the feedback link.
	4. The visitor is (a)redirected to the link, (b) fills out the feedback form and (c) submits it.
←	5. The system processes and stores the feedback.
	6. The system displays a confirmation message to the visitor.

Table 10. Fully Dressed Description of SubmitFeedback

Use Case UC-7	**ScanQR**
Related Requirements:	Functional Requirements: REQ-12, REQ-13, REQ-14 Non-Functional Requirements: NONREQ-6, NONREQ-8, NONREQ-9, NONREQ-13, NONREQ-14, NONREQ-15
Initiating Actor(s):	Security Guard
Actor's Goal:	Successfully scan a visitor's QR code and allow or deny entry based on access conditions.
Participating Actor:	Visitor
Preconditions:	● Visitors must be registered in the system with a valid QR code. ● Security guards must have access to the check-in system.
Post conditions:	The visitor ● Checks in ● Entry is granted, and ● The system logs the event.
Flow of Events for Main Success Scenario:	
→	1. Security enters credentials: Include UC-1 Login
	2. Security navigates to the visitor scanning page.
	3. (a)Visitor arrives at the security checkpoint with QRcode and the (b) security guard scans the visitor's QR code using the check-in system
←	4. System validates the QR code against registered visitors.
	5. System notifies the resident of visitor arrival.
→	6. Visitors are granted entry.
Flow of Events for Extensions (Alternate Scenarios):	
→	3. Visitor arrives at the security checkpoint with QRcode and the security guard scans the visitor's QR code using the check-in system. The QRcode was invalid. a. The visitor is not granted access.
←	4. The resident is notified of the potential visitor.

Table 11. Fully Dressed Description of ScanQR

Use Case UC-8	**ViewHistory**
Related Requirements:	Functional Requirements: REQ-11 Non-Functional Requirements: NONREQ-3, NONREQ-8, NONREQ-14
Initiating Actor(s):	Security Guard
Actor's Goal:	Review past visitor entries
Participating Actor:	Database
Preconditions:	● Security Guard must be logged into the system ● Systems must have stored visitor check-in data
Post conditions:	The visitor history is displayed.
Flow of Events for Main Success Scenario:	
→	1. Security enters credentials: Include UC-1 Login
	2. Security guard navigates to the "Visitor History" section in the system
←	3. System displays a search interface.
→	4. Security guards enter filter information.
←	5. System (a) retrieves search information and (b) displays the relevant visitor history.
	6. Security guard reviews the records.
Flow of Events for Extensions (Alternate Scenarios):	
←	5. System returns (a) no data because information does not exist or (b) system fails to retrieve data.
	6. Security Guard receives a note that states if (a) there is no existing record or (b) the database is having specific issues.

Table 12. Fully Dressed Description of ViewHistory

Use Case UC-9	**ManageBlacklist**
Related Requirements:	Functional Requirements: REQ-4, REQ-15 Non-Functional Requirements: NONREQ-8, NONREQ-18, NONREQ-19
Initiating Actor(s):	Resident or Security Guard
Actor's Goal:	To report a visitor for an act that is not acceptable by the community or a specific resident.
Participating Actor:	Database
Preconditions:	● The resident or security guard must be logged into the system. ● Visitors must have a record in the system.
Post conditions:	The visitor ● Is added to the blacklist, preventing future check-ins or ● Is removed from the blacklist, allowing access
Flow of Events for Main Success Scenario:	
→	1. Resident or Security enters credentials: Include UC-1 Login
	2. Resident or Security guard navigate to the "Manage Blacklist" section
←	3. System displays the list of visitors and search options
→	4. User enters a name or ID to search.
←	5. The system retrieves and displays the visitor's profile.
→	6. Resident or security guard selects "Add to Blacklist" or "Remove from Blacklist"
←	7. The system updates the visitor's access.
Flow of Events for Extensions (Alternate Scenarios):	
6a. Selected activity entails manage blacklist: Include UC-11 AddToBlacklist	
6b. Selected activity entails manage blacklist: Include UC-12 RemoveFromBlacklist	
←	5. System returns (a) no data or (b) system fails to retrieve data.
	6. Security Guard or Resident receives a note that states if (a) there is no existing record or (b) the database is having specific issues.

Table 13. Fully Dressed Description of ManageBlacklist

Use Case UC-12	**ManageUsers**
Related Requirements:	Functional Requirements: REQ-4, REQ-15 Non-Functional Requirements: NONREQ-8, NONREQ-18, NONREQ-19
Initiating Actor(s):	Administrator
Actor's Goal:	Manage user accounts, including creating new accounts, modifying account details, or deleting accounts.
Participating Actor:	Database
Preconditions:	● The administrator must be logged into the system. ● Usersmust have a record in the system.
Post conditions:	A user is added, edited or deleted from the system.
Flow of Events for Main Success Scenario:	
→	1. Administrator enters credentials: Include UC-1 Login
	2. Administrator navigate to the "Manage Users" section
	3. Administrator selects the desired action "Add User," "Edit User", or "Delete User"
←	4. The system displays the relevant interface for (a) adding a user where a form is provided to enter user details or a (b) a search field to select a user for (b1) editing a user or (b2) deleting a user.
→	5. The admin provides the necessary input and submits the request
←	6. The system validates the input and processes the request.
	7. A confirmation message is displayed, indicating that the action was successfully completed.
Flow of Events for Extensions (Alternate Scenarios):	
4a. Selected activity entails manage users: Include UC-13 AddUser	
4b. Selected activity entails manage users: Include UC-14 EditUser	
4c. Selected activity entails manage users: Include UC-15 DeleteUser	
←	5. Required fields are missing when adding or editing a user: a. The system displays an error message stating "Please fill out all required fields before submitting"

Table 14. Fully Dressed Description of ManageUsers

Use Case UC-16	**View Feedback**
Related Requirements:	Functional Requirements: REQ-3, REQ-23 Non-Functional Requirements: NONREQ-8, NONREQ-16, NONREQ-17
Initiating Actor(s):	Administrator, Resident, or Security Guard
Actor's Goal:	To view visitor feedback submitted through the system.
Participating Actor:	Database
Preconditions:	● The Administrator, Resident, or Security Guard must be logged into the system. ● At least one feedback entry must exist in the system.
Post conditions:	The Administrator, Resident, or Security Guard successfully view visitor feedback.
Flow of Events for Main Success Scenario:	
→	1. Administrator, Resident, or Security Guard enters credentials: Include UC-1 Login
	2. Administrator, Resident, or Security Guard navigates to the "Visitor Feedback" section.
←	3. System retrieves and displays visitor feedback.
→	4. Administrator, Resident, or Security Guard view feedback.
→	5. The Administrator, Resident, or Security Guard exits the feedback.
Flow of Events for Extensions (Alternate Scenarios):	
←	3. System notifies the Administrator, Resident, or Security Guard with a message: "No feedback available at the moment."
→	4. Administrator, Resident, or Security Guard chooses to either (a) exit the section or (b) retry fetching feedback later.

Table 15. Fully Dressed Description of ViewFeedback

Sequence Diagrams

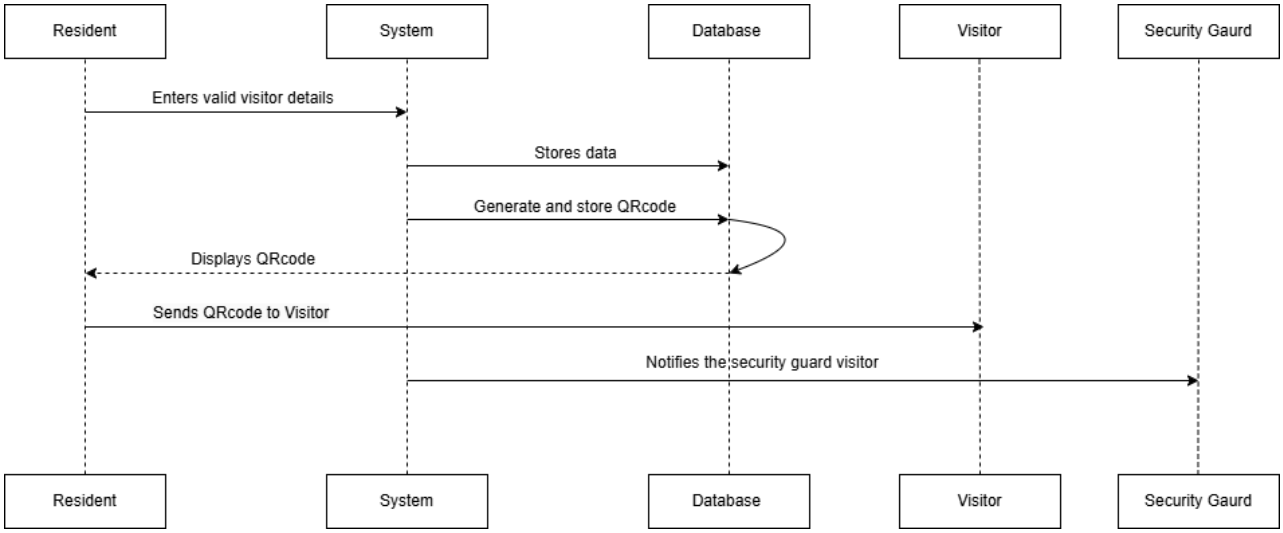

Fig 6. UC-2 CreateVisitorSchedule

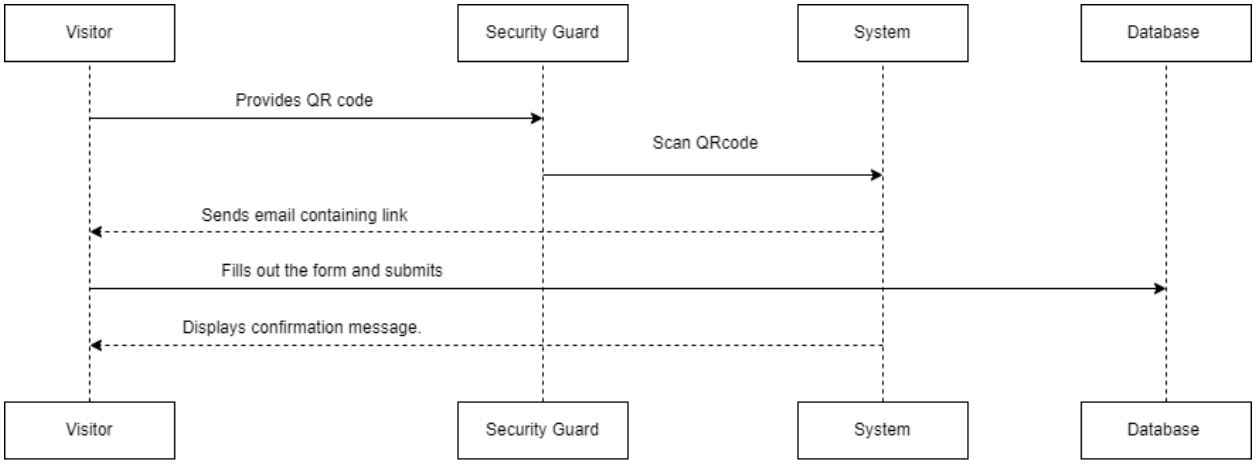

Fig 7. UC-6 SubmitFeedback

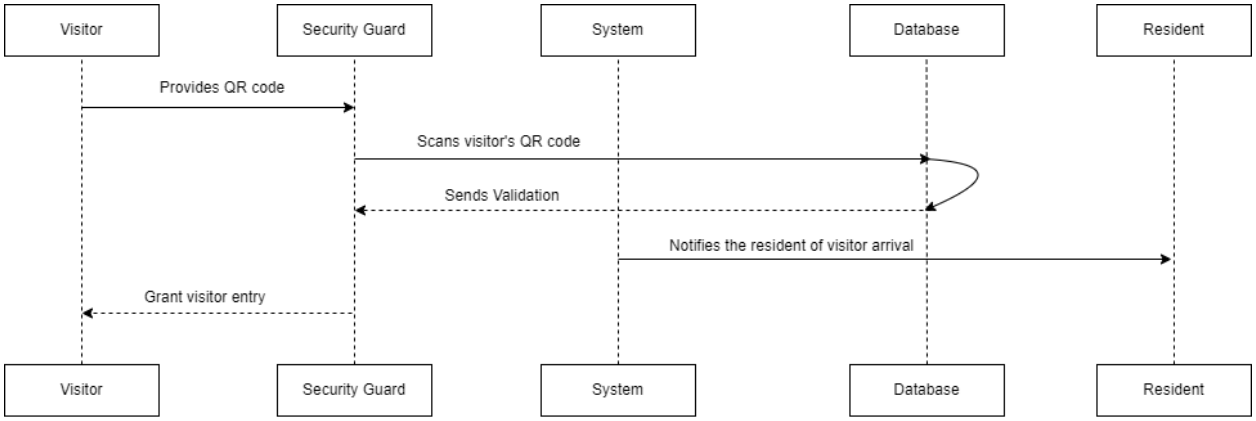

Fig 8. UC-7 ScanQR

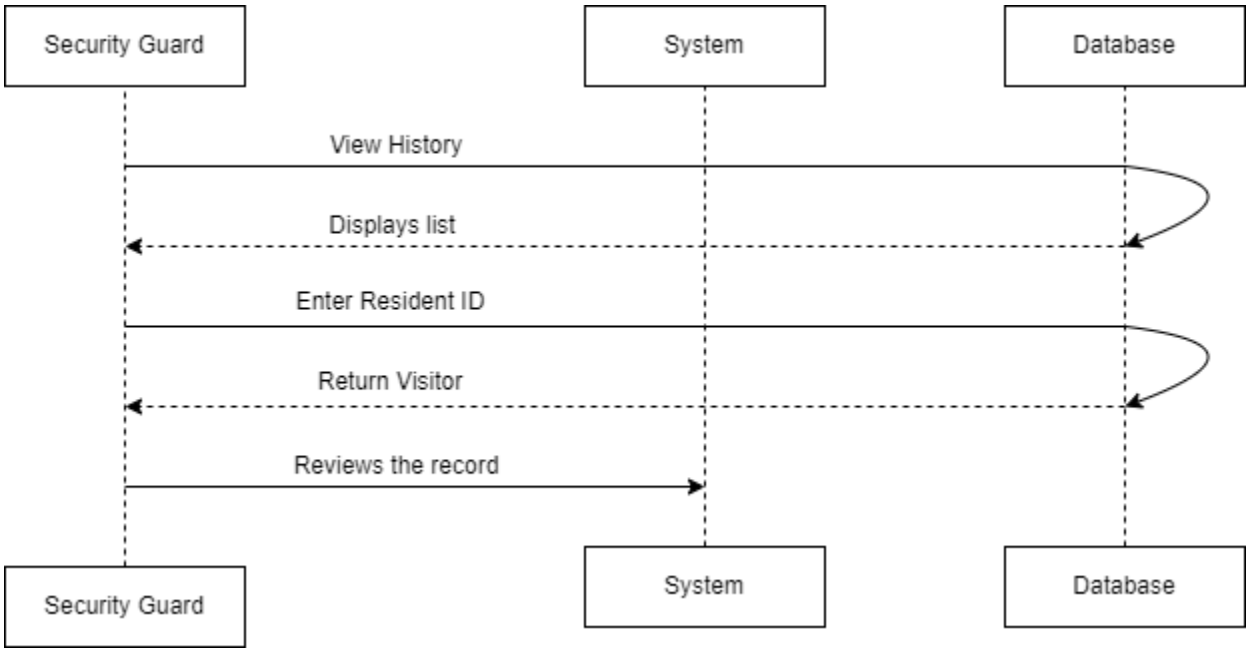

Fig 9. UC-8 ViewHistory

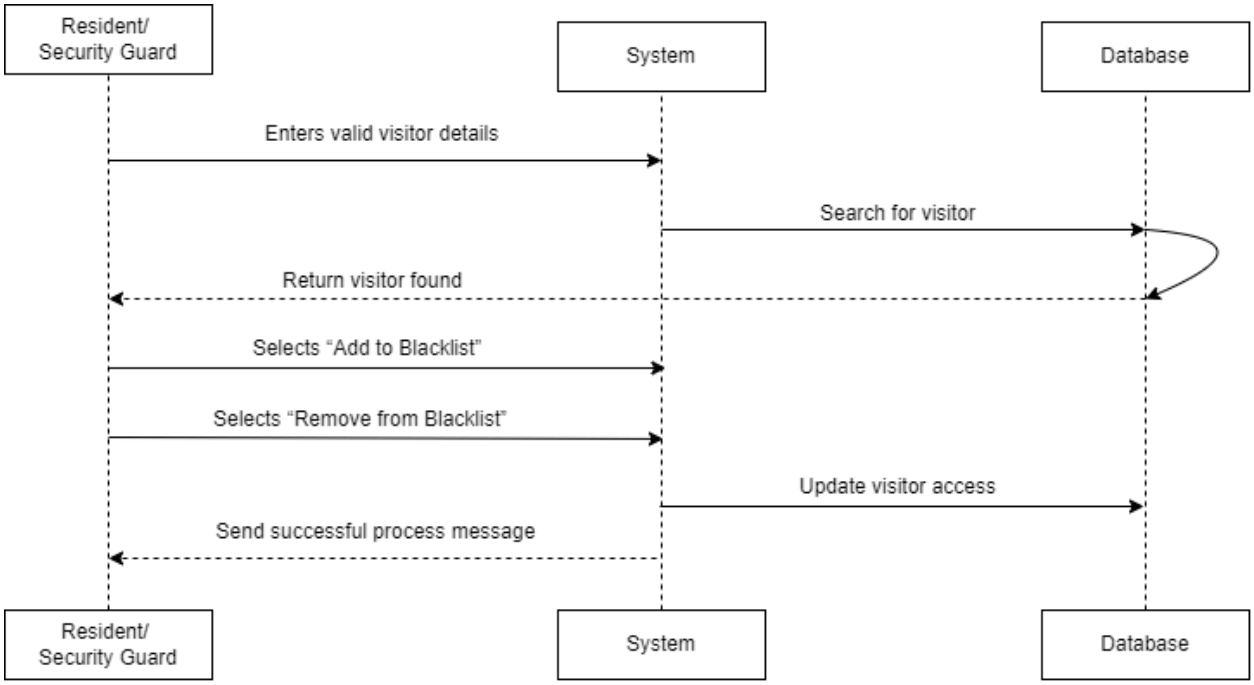

Fig 10. UC-9 ManageBlacklist

To effectively manage users, each process has a distinct flow, requiring a separate sequence diagram to accurately represent each one.

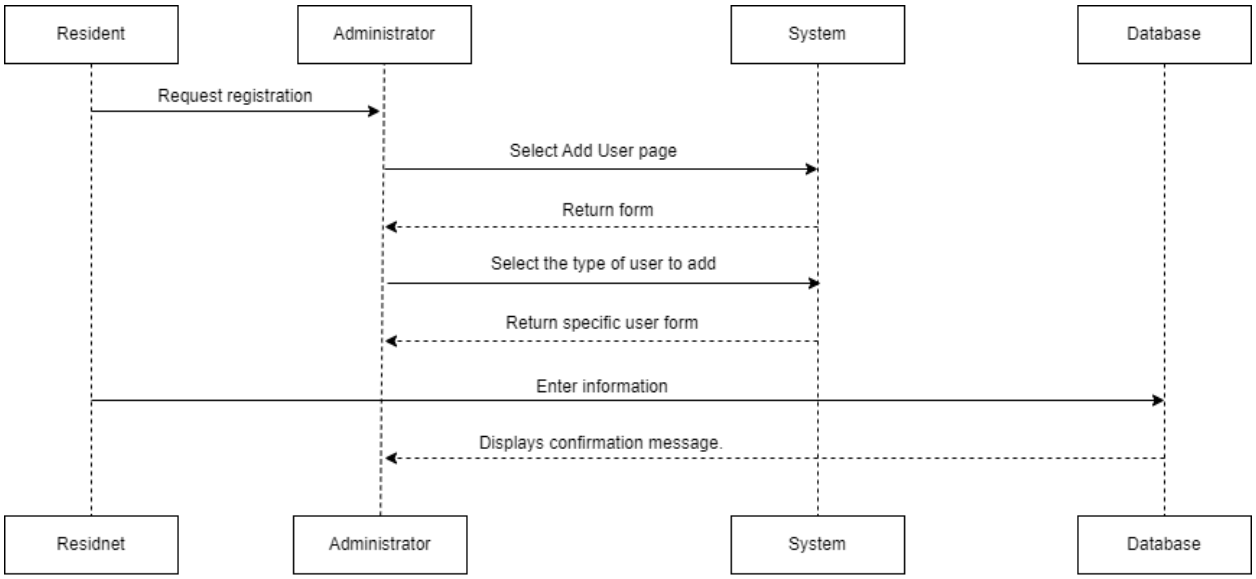

Fig 11. UC-13 AddUsers

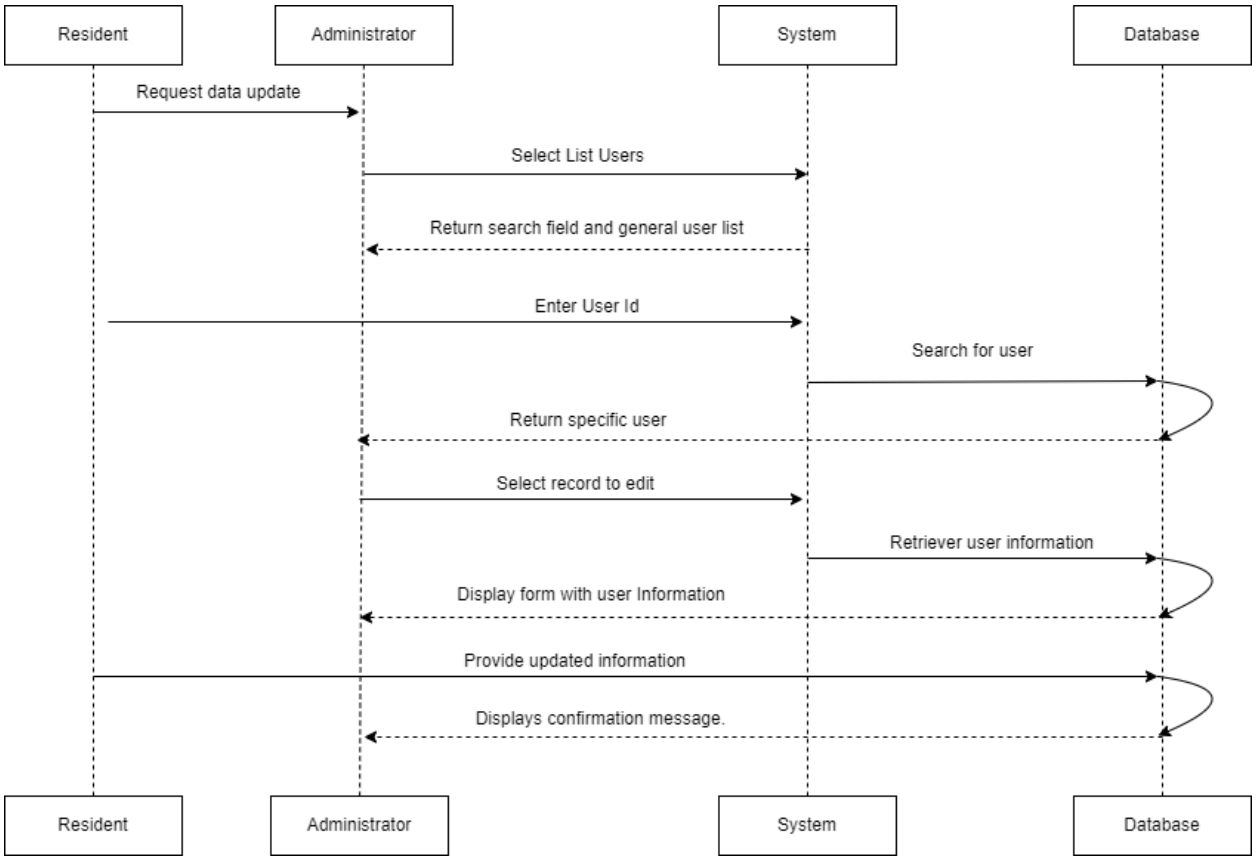

Fig 12.UC-14 EditUser

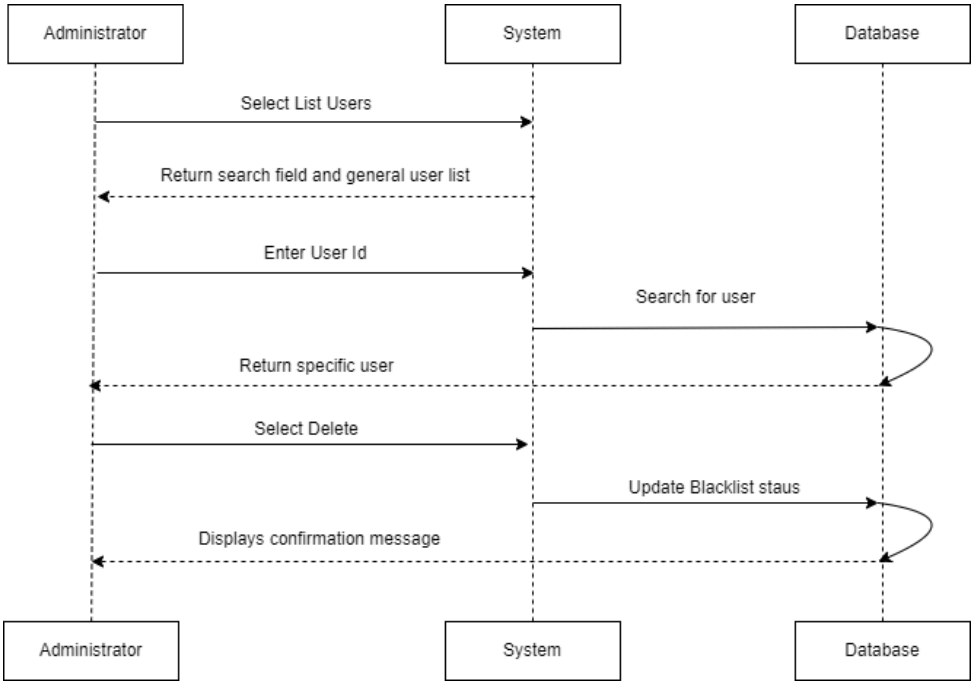

Fig 13.UC-15 DeleteUser

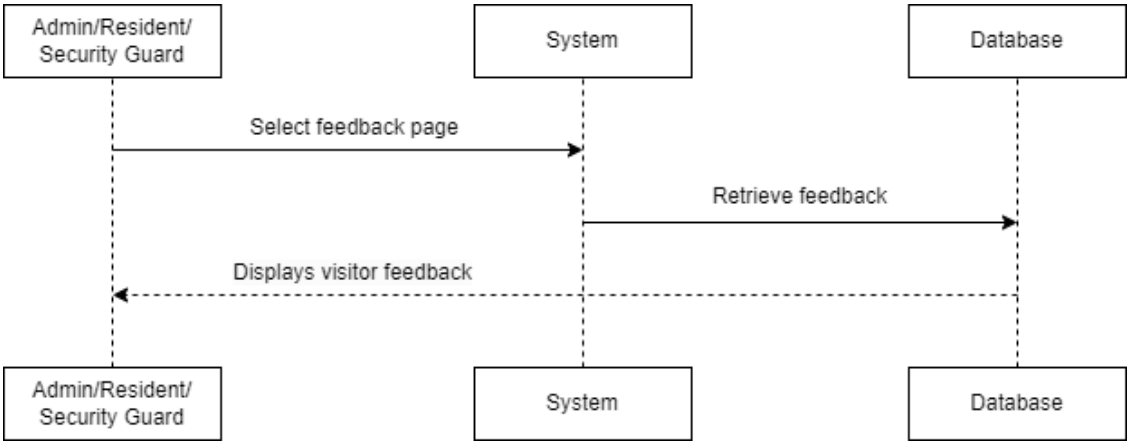

Fig 14. UC-16 ViewFeedback

User Interface Specifications

Preliminary Design

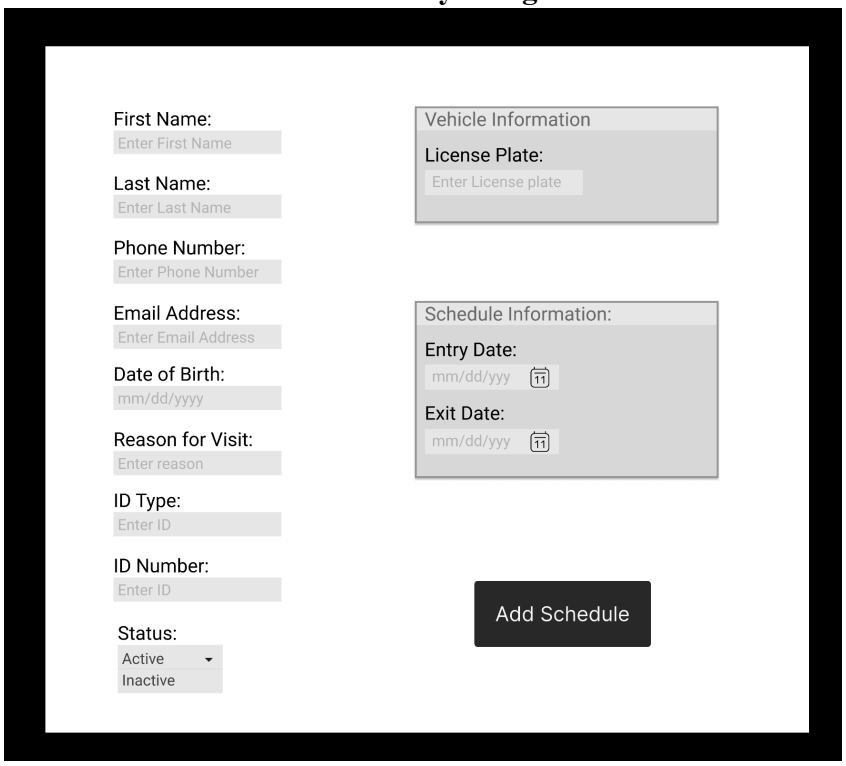

Fig 15. UC-2 CreateVisitor Schedule

To create a visitor schedule, the user must first fill in visitor information, vehicle details, and schedule information before clicking the 'Add Schedule' button. The visitor can be either active or inactive.

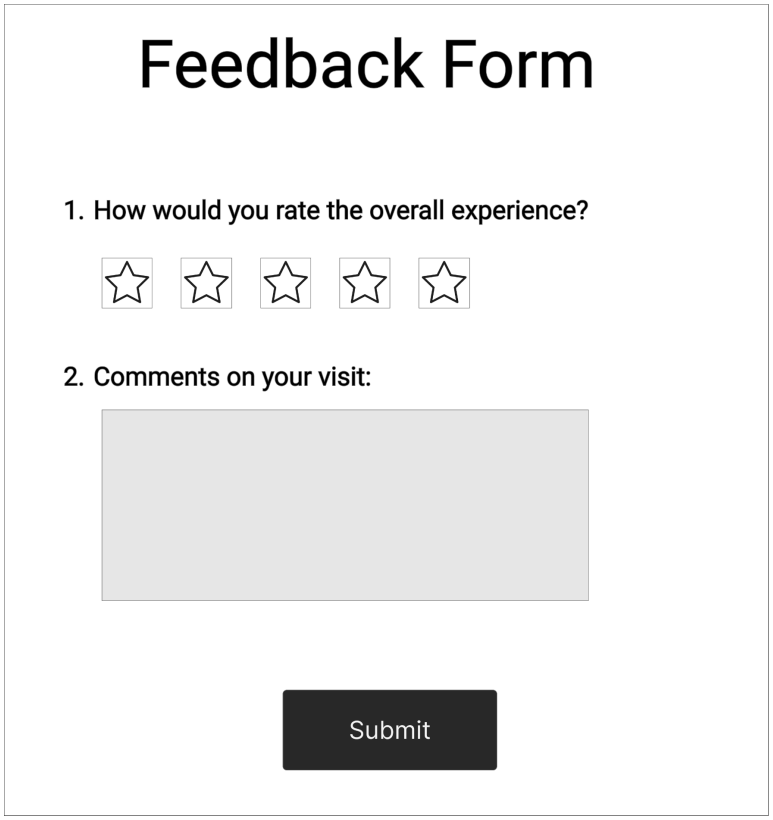

Fig 16. UC-6 SubmitFeedback

To submit a feedback form, the visitor will provide feedback using the star rating and can enter comments if desired.

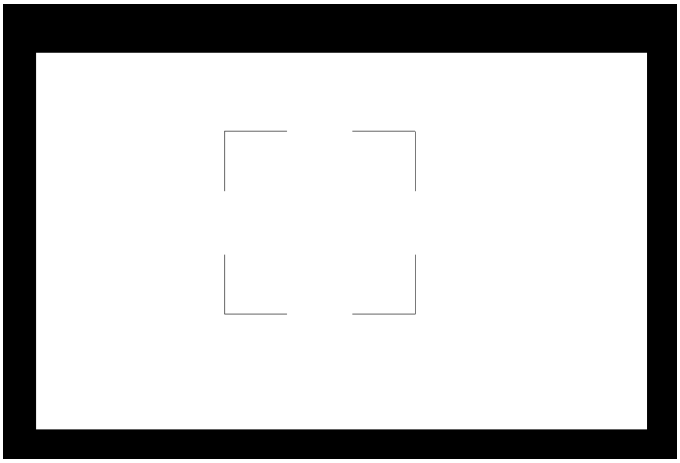

Fig 17. UC-7 ScanQR

To scan the QR code, the security guard will open the scanning page and scan the visitor's QR code to successfully grant entry.

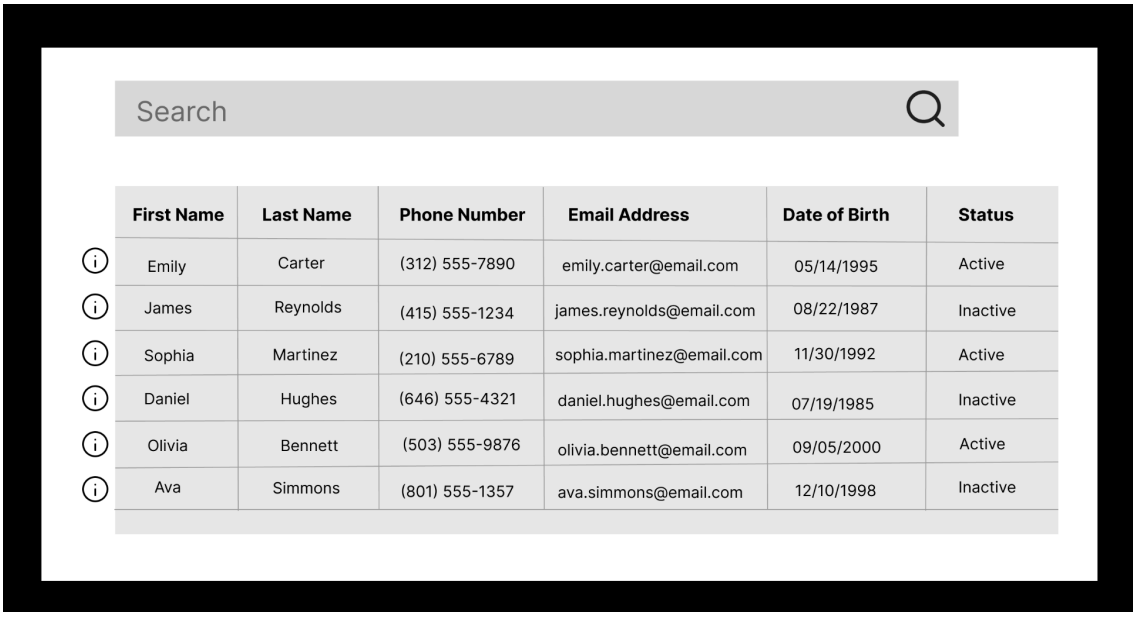

Fig 18. UC-8 ViewHistory

To view history, the user will navigate to the 'View History' page, where a list of visitors will be displayed. The user can also use the search bar to find specific visitors.Additionally, there is an information icon, and when clicked, it will display more details about the visitor.

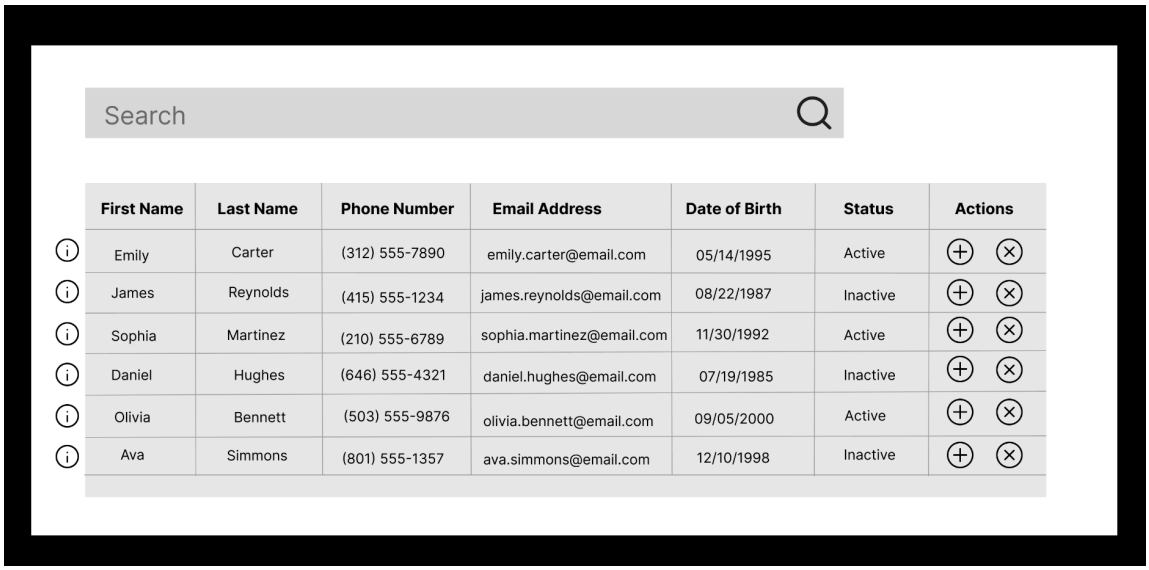

Fig 19. UC-9 ManageBlacklist

To manage the blacklist, the user will be displayed a table of all visitors with an actions column. They can add or remove a visitor from the blacklist by clicking the respective icons, which will trigger a confirmation popup.

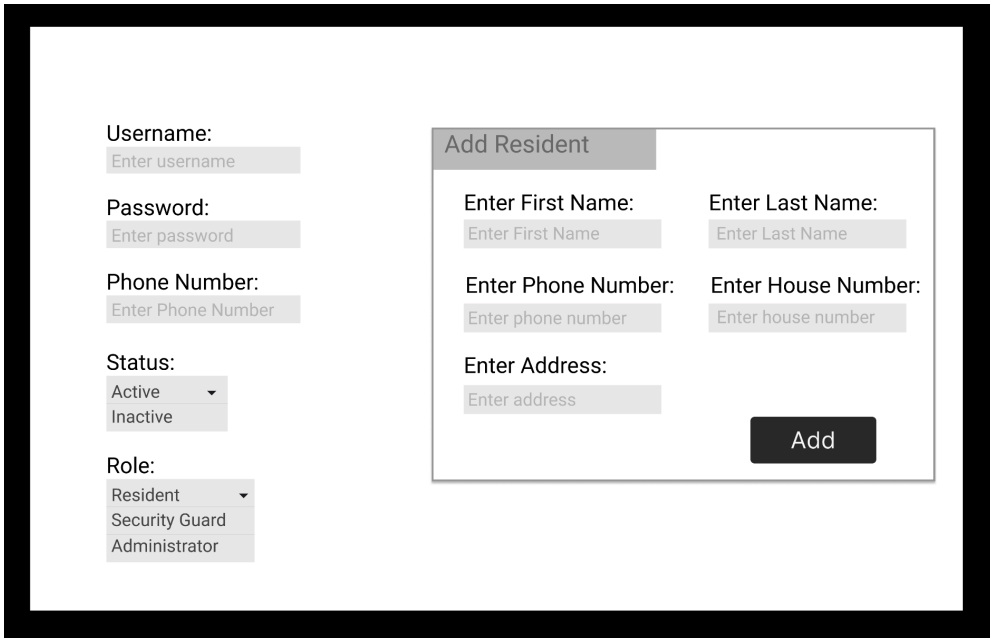

Fig 20. UC-13 AddUsers - Resident

To add a user, the admin must fill in all required fields. If the selected role is 'Resident,' a form will pop up on the side requesting resident information. The admin will enter the details and click the 'Add' button to successfully add the resident.

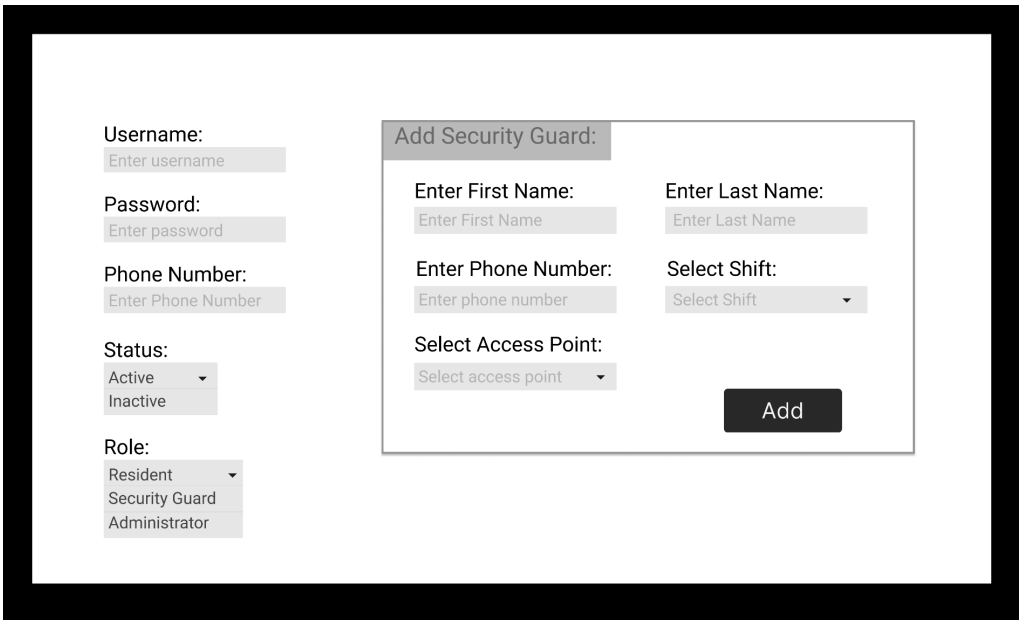

Fig 21. UC-13 AddUsers - Security Guard

The same process applies when adding a security guard. The admin must fill in the required information, and if 'Security Guard' is selected as the role, a form will pop up requesting security guard details. The admin will then enter the information and click the 'Add' button to successfully add the security guard.

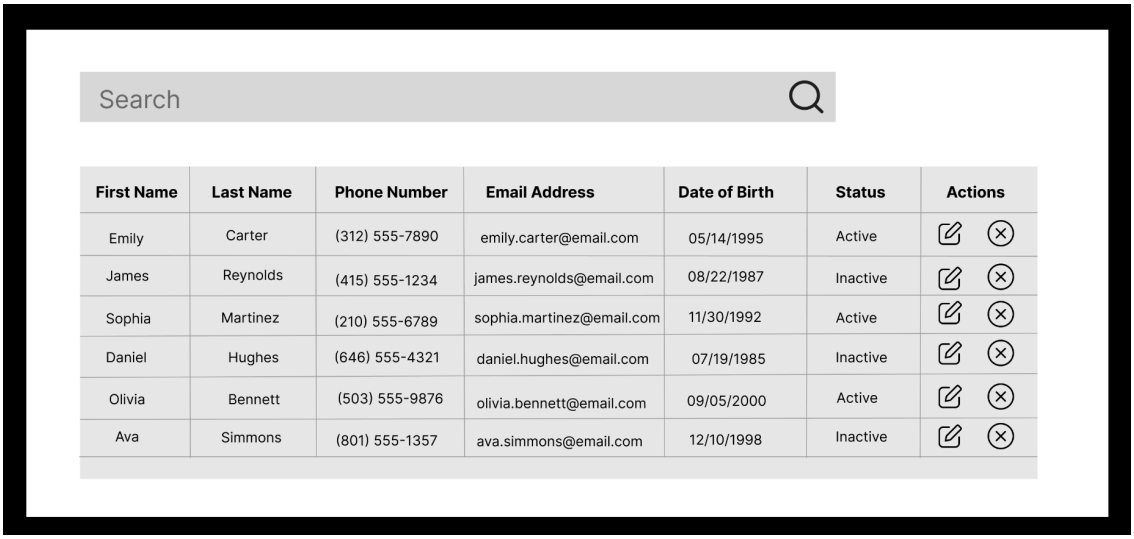

Fig 22. UC-14 EditUser and UC-5 Delete User

To edit or delete a user, the admin will open the 'List of Users' page, which will display a table with the list of users. The table includes an actions column with icons to edit or delete a user. The admin can choose to either edit or delete a user by clicking on the corresponding icon. If editing, they will be redirected to a page with the user's details to make changes. If deleting, a confirmation box will appear.

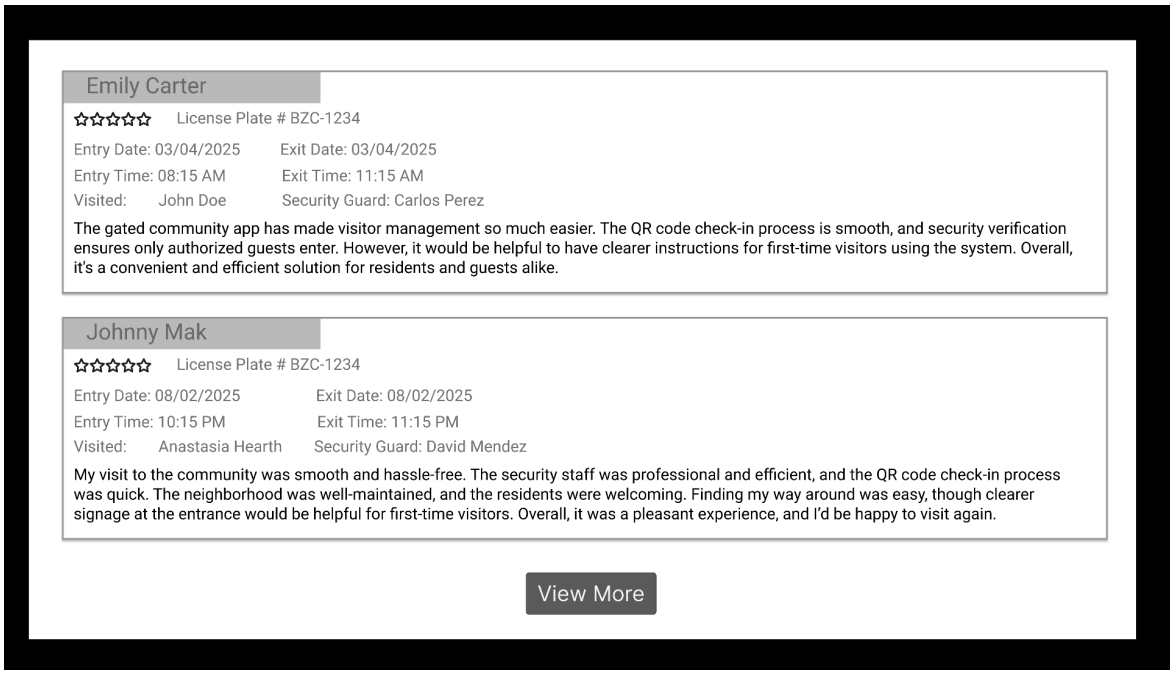

Fig 23. UC-16 View Feedback

To view visitor feedback, the user will go to the 'View Feedback' page, where a list of feedback will be displayed. They can click the 'View More' button to see additional visitor feedback.

User Effort Estimation

Create Visitor Schedule (UC-2) - total 14 mouse clicks

1. Click "Schedule Visit" on the Home page.
2. Click on First Name Field
3. Click on Last Name Field
4. Click on Phone Number field
5. Click on Email Field
6. Click on DOB field
7. Click on Reason for Visit
8. Click on ID type Field
9. Click on ID Number Field
10. Select Status
11. Click on the licence plate field.
12. Click on the entry date field.
13. Click on the exit date field.
14. Click "Add Schedule."

Data Entry – 13 mouse clicks & 100 + keystrokes

1. Click in the First Name field and type First Name.
2. Click in the Last Name field and type Last Name.
3. Click in the Phone Number field and type Phone Number .
4. Click in the Email field and type Email .
5. Click in the DOB field and type Date of Birth.
6. Click in the Reason for Visit field and type Reason.
7. Click in the ID Type field and select ID Type (Dropdown).
8. Click in the ID Number field and type ID Number.
9. Click in the Status field and select Status (Dropdown).
10. Click in the License Plate field and type License Plate .
11. Click in the Entry Date field and type Entry Date.
12. Click in the Exit Date field and type Exit Date.
13. Click "Add Schedule."

Submit Feedback (UC-6) – total 5 mouse clicks

1. Click on the “Feedback Form” link sent through email.
2. Click on Star Rating to select a rating.
3. Click on the Comment Box to enter feedback.
4. Type Comments .
5. Click "Submit."

Data Entry – 2 mouse clicks & 50 keystrokes

1. Click on Star Rating to select a rating.
2. Click in the Comment Box to enter feedback.
3. Type the Comments.
4. Click "Submit."

Scan QR (UC-7) – total 1 mouse click

1. Click "Scan QR" on the Home page.

Data Entry – 0 keystrokes

1. Hold QR code up to the scanner.

View History (UC-8) – total 2 mouse clicks

1. Click "Visitor History" on the Home page.
2. Click in the Search field to enter Visitor ID or Name.

Data Entry – 1 mouse click & 12 keystrokes

1. Click in the Search field to enter Visitor ID or Name.
2. Type Visitor ID or Name.

Manage Blacklist (UC-9)

Add To Blacklist (UC-10) – total 4 mouse clicks

1. Click "Manage Blacklist" on the Home page.
2. Click in the Search field to enter Visitor Name.
3. Click the "+" (add) icon to add a record.
4. Click "Confirm."

Remove From Blacklist (UC-11) – total 4 mouse clicks

5. Click "Manage Blacklist" on the Home page.
6. Click in the Search field to enter Visitor Name.
7. Click the "x" (delete) icon to remove a record.
8. Click "Confirm."

Manage User (UC-12)

Add User (UC-13) – total 1 mouse clicks

1. Click “Create User” on the home screen
2. Enter username, password and phone number
3. Select user status
4. Select the role of the user to be added.

Data Entry – 12 mouse clicks & 40 keystrokes

If the selected role is a resident:

1. Click in the First Name field and enter information.
2. Click in the Last Name field and enter information.
3. Click in the Phone Number field and enter information.
4. Click in the House Number field and enter information.
5. Click in the Address field and enter information.
6. Click “Add” to add the user.

If the selected role is a security guard:

1. Click in the First Name field and enter information.
2. Click in the Last Name field and enter information.
3. Click in the Phone Number field and enter information.
4. Click in the Shift field and select shift.
5. Click in the Access Point field and select the access point.
6. Click “Add” to add the user.

Edit User (UC-14) – total 3 mouse clicks

1. Click "List of Users" on the Home page.
2. Click on Search field
3. Click on the Edit icon located on the actions column

Data Entry – 22 mouse clicks & 40 keystrokes

To edit a resident:

1. Click on Username and enter the updated username.
2. Click on Password and enter the updated password.
3. Click on Phone Number and enter the updated phone number.
4. Click on Status and select the status.
5. Click on Role and select the role.
6. Click in the First Name field and enter the updated first name.
7. Click in the Last Name field and enter the updated last name.
8. Click in the Phone Number field and enter the updated phone number.

9. Click in the House Number field and enter the updated house number.
10. Click in the Address field and enter the updated address.
11. Click “Update” to save changes.

To edit a security guard:

1. Click on Username and enter the updated username.
2. Click on Password and enter the updated password.
3. Click on Phone Number and enter the updated phone number.
4. Click on Status and select the updated status.
5. Click on Role and select the updated role.
6. Click in the First Name field and enter the updated first name.
7. Click in the Last Name field and enter the updated last name.
8. Click in the Phone Number field and enter the updated phone number.
9. Click on Shift and select the updated shift.
10. Click on Access Point and select the updated access point.
11. Click “Update” to save changes.

Delete User (UC-15) – total 3 mouse clicks & 15 keystrokes

1. Click "List of Users" on the Home page.
2. Click on the Search field and enter the User ID or name.
3. Click on the Delete icon in the actions column.
4. Click “Confirmation” to confirm deletion.

View Feedback (UC-16)

Navigation – total 2 mouse clicks

1. Click "Visitor Feedback" on the Home page.
2. Click the “View more” button to view more feedback.

Data Entry – 0 keystrokes

1. No keystrokes required for this action

System Architecture

Identifying Subsystems

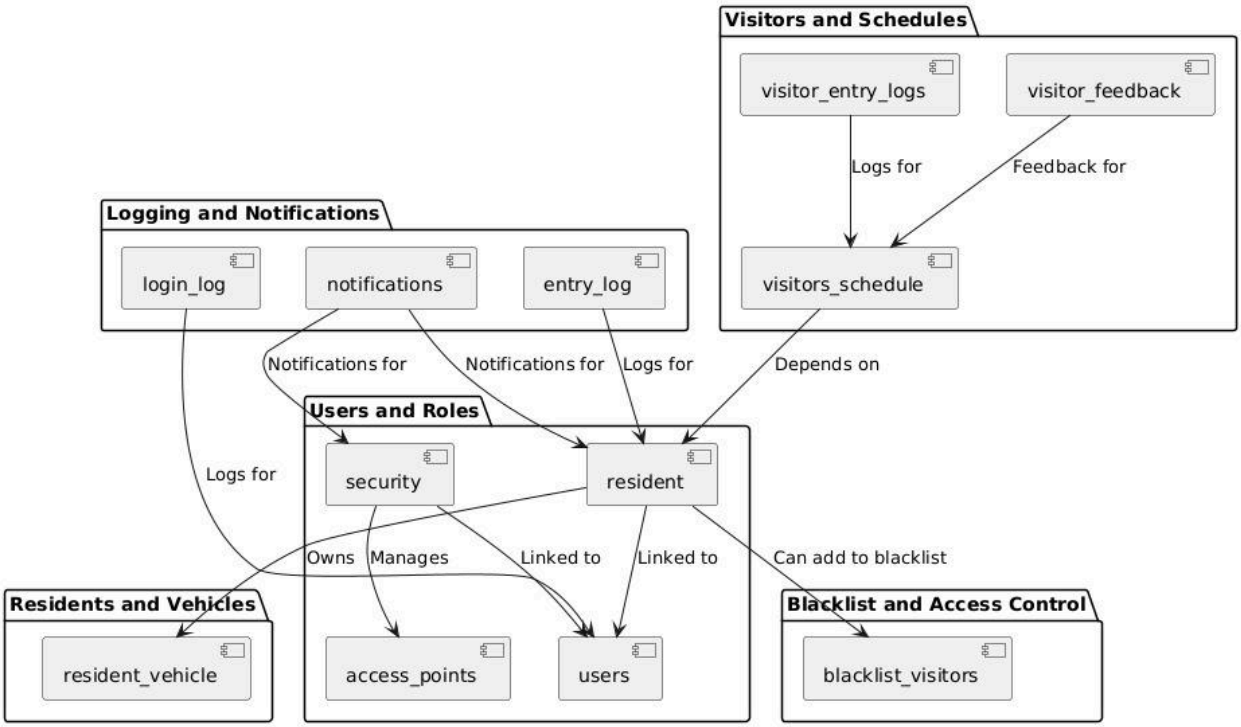

Fig 24. UML Package of Subsystems

Figure 24 illustrates the key subsystems of the gated community application and their relationships. The Users and Roles package manages residents and security personnel, linking them to access control and visitor management. The Visitors and Schedules package handles visitor entry logs, feedback, and scheduled visits. Logging and Notifications ensures system transparency by tracking logins, entries, and sending alerts. The Blacklist and Access Control package restricts unauthorized visitors, allowing security and residents to manage blacklists. Lastly, Residents and Vehicles connect residents to their registered vehicles for seamless access. These subsystems work together to ensure efficient security, visitor tracking, and communication within the gated community.

Architecture Styles

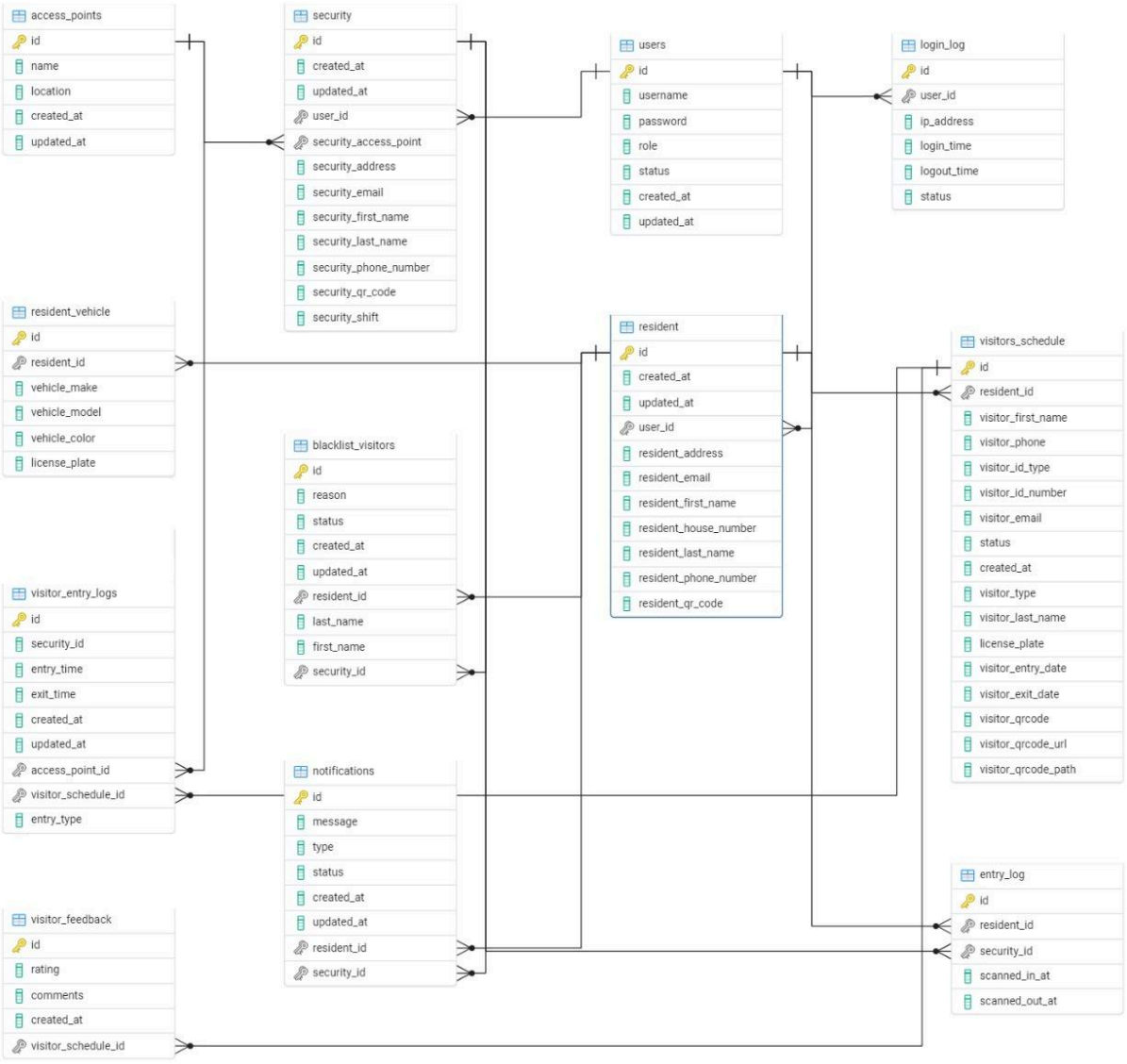

Fig 25. Database Schema

Based on Figure 25, the architectural design of the gated community system primarily follows a layered architecture with elements of a client-server model and event-driven architecture.

1. Data Layer
   The schema defines structured entities for managing users, visitors, vehicles, security logs, notifications, and access control. Tables such as users, security, and residents define user roles, while visitor_schedule, visitor_entry_logs, and entry_log track visitor movements. The use of relational database design ensures data integrity, with foreign key relationships linking residents, security personnel, and visitors.
2. Business Logic Layer
   The system's logic enforces access control through relationships between security

personnel, residents, and access points. Tables such as blacklist_visitors and notifications suggest an event-driven approach, where actions such as unauthorized visitor entries trigger notifications and log updates. The security team can manage access dynamically, ensuring real-time response to events.

3. Presentation Layer
   The web-based interface allows users to interact with the system, submit visitor schedules, and review logs. The system ensures role-based access, where security personnel monitor entry_log, residents manage their visitor_schedule, and administrators oversee the entire operation.

By combining a relational database model with event-driven mechanisms, the system ensures secure, scalable, and real-time management of visitor access, resident activities, and security enforcement.

Mapping Systems to Hardware

As a web-based, mobile-responsive application, it will allow residents to access features from their smartphones, tablets, or computers once an internet connection is available. Security personnel will use the system at multiple stations if the community has multiple entries and exit points, accessing it via devices like tablets or desktops to verify visitors and log incidents. Administrators can manage the system from their offices, overseeing user accounts, blacklists, and system operations. A centralized web server will handle authentication, data storage, and notifications, ensuring real-time synchronization and secure access across all user roles.

Connectors and Network Protocols

The application will use Prisma alongside Next.js, utilizing built-in API routes for seamless communication between the frontend and backend. Prisma, an ORM (Object-Relational Mapping) tool, will handle database interactions efficiently, connecting to a PostgreSQL database. This choice is ideal for our system because Prisma simplifies complex database queries, improves security by reducing the risk of SQL injection, and enhances maintainability with its type-safe query system. Next.js API routes allow for a structured, scalable approach to handling requests, making it easier to manage authentication, data retrieval, and other backend operations essential for our gated community application.

Global Control Flow

Our gated community application is event-driven, meaning it does not follow a strict linear sequence of steps for every user. Instead, it responds to user actions, such as residents generating QR codes, and security personnel scanning visitor passes. Each user can interact with the system in a different order depending on their needs.

Regarding time dependency, the system includes some timed operations. For example, it ensures that QR codes for recurring visitors expire after their scheduled time, and it locks user accounts for 30 minutes after multiple failed login attempts. Additionally, real-time alerts and notifications are sent instantly to residents and security personnel when a visitor checks in or is denied entry.

The system operates in an event-response model rather than a strict real-time system. However, certain actions require real-time performance, such as QR code scanning, blacklist alerts, and security logs, which must be processed immediately to ensure smooth entry management.

Hardware Requirements

Our system relies on several critical resources to ensure smooth operation and optimal performance. The server must have at least 32GB of memory and a RAID 5 configuration with three 500GB hard drives to support data redundancy, security, and efficient storage management. A stable and high-speed network connection is essential for seamless communication between users and the database. Security personnel will utilize 10-inch tablets for scanning visitor QR codes, which require responsive touchscreens and reliable internet connectivity for real-time verification. Additionally, the system depends on modern web browsers for residents and administrators to access the platform, with a recommended minimum screen resolution of 1280 × 720 pixels for an optimal user experience.

Project Management

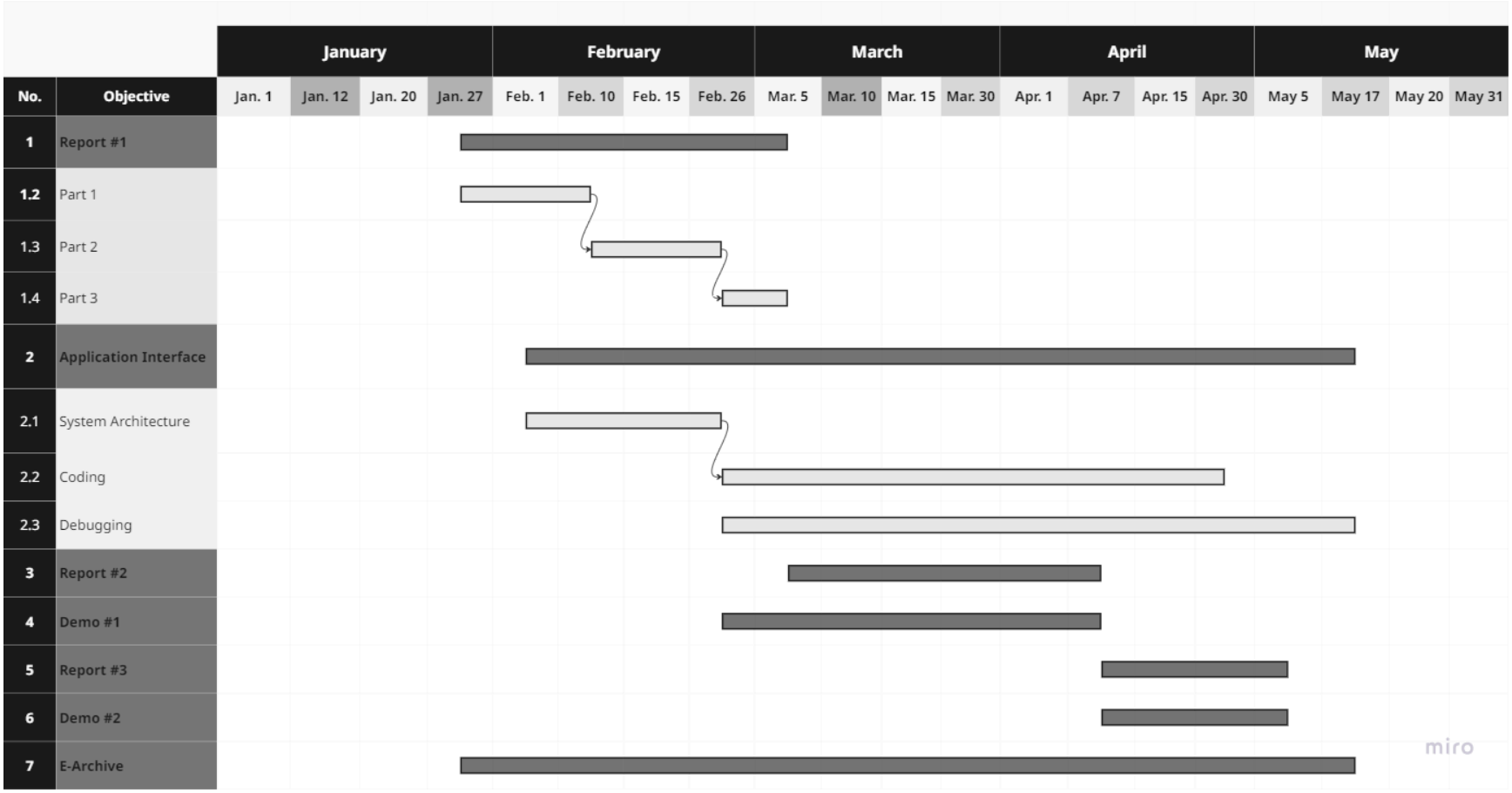

Fig 26. Gantt Chart

Responsibilities	Team Members					
	Aiyesha Coleman	Kelsey Aban	Jordani Alpuche	Aiden Pinelo	Luigi Acuna	Arthur Butler
Part 1						
Customer Statement of Reqs	-	✅	✅	-	-	-
System Requirements	✅	-	-	-	-	-
Project Management	✅	-	-	-	-	-
Part 2						
Stakeholders	✅	-	-	-	-	-
Actors	✅	-	-	-	-	-
Casual Descriptions	✅	-	-	-	-	-
Use Case Diagrams	✅	✅	-	-	-	-
Fully Dressed Descriptions	✅	✅	✅	-	-	-
Sequence Diagrams	-	-	✅	-	-	-
Preliminary Design	-	✅	✅	-	-	-
User Effort Estimation	-	-	-	✅	-	-
Project Management	✅	-	-	-	-	-
Part 3						
System Architecture	✅	✅	✅	-	-	-
Project Management	✅	-	-	✅	-	-

Table 16. Breakdown of Responsibilities

References

PalAmerican Security.(2021, August 10). *6 Security Procedures Your Gated Community Needs.* PalAmerican Security. https://www.palamerican.com/community/6-security-procedures-your-gated-community-needs/#_kx1a848liam6

Hope, H., Tzib, E., & Shol, J. (2024, December 8). *SSP5: Gated community management system.* Retrieved from https://sites.google.com/ub.edu.bz/gated-community-management-app/home/ssp5