Software Requirements Specification

for

A centralized QR-based campus system utilizing a single common QR code for entry/exit management, attendance, and access to institutional services

Version 2.0 approved by:
Alpha Team

Prepared by:
Alpha Team

Group 22

28 January 2026

Table of Contents

Revision History

Version	Date	Name	Reasons for changes
1.0	**21/01/2026**	**Initial SRS Submission**	N/A
2.0	**28/01/2026**	**SRS Version 2**	**Modified UML diagrams+inserted UML** **Added Clarification about certain if there is any need for changing the QR** **Added the functionality about deleting data for deregistered students.** **When you say version 1.0 approved, which alpha-team recommended the same needs to be highlighted** **Provided clarification regarding the location indicator in User Interface as it was causing some confusion.** **Used "justify" to align the entire document on both sides.** **Traceability matrices and templates for FR and NFR documented.** **Provided clarity regarding whether manual entry is mandatory or not (also given in the use case diagram and traceability matrix)** **Design and Coding standards added (template chosen)** **Added techstack** **A realistic sprint plan for the modules added**

1. Introduction

1.1 Purpose

This document presents a detailed description of functional and non-functional requirements of a centralized QR code-based campus system. The objective of this Software Requirements Specification (SRS) is to define system behavior, constraints, and interfaces to facilitate design, implementation, testing, and evaluation activities.A centralized qr code based application for campus in/out, attendance, and other activities for quick access of service.

1.2 Document Conventions

This document follows standard IEEE Software Requirements Specification (SRS) conventions.

The document is organized using numbered sections and subsections (e.g., 1, 1.1, 1.2, 2.1, etc.) for easy reference.

Priority:

- All mandatory requirements are specified using the keyword "shall".

- The keyword “should” is used to indicate recommended but non-mandatory features.
- The keyword “may” is used to indicate optional features. Each requirement is written as a separate, clear, and testable statement.

Requirements are presented in numbered or bulleted lists for better readability and all have equal priority.

1.3 Intended Audience and Reading Suggestions

The intended audience includes mentor,course Instructor and TAs, who will review the srs. Team Members who will use this document as a reference for system design and implementation. End users(students, faculty, guards and admin) who may refer to this document to understand some system functions.

Reading Suggestions:

Team Members should read the entire document carefully.

End users may focus mainly on the system features section.

1.4 Product Scope

-The QR-based entry/exit and attendance system shall provide a secure and efficient method for recording student entry, exit,and attendance across the campus using QR codes.

-The system shall allow authorized guards to scan QR codes at designated access points (classroom and campus/hostel entry gates) and shall process all the entries made in real time.

-The system should reduce manual effort and waiting time at entry points.

-The system may provide administrative features for monitoring, reporting, and managing campus access data.

- The system is intended for use within the institution only and does not include external authentication or biometric verification mechanisms.

1.5 References

- Few colleges have digitalised id based attendance system
- Mess QR based system
- Library QR based system

1.6 Project Management and Task Tracking

- The project follows an iterative, sprint-based development approach. Task planning, tracking, and coordination are managed using Trello cards.
- Trello is used to organize project tasks into lists representing different stages of progress, including *Backlog, Work in Progress, Testing and review*. Each Trello card represents a specific development task requirement.
- The use of Trello enables clear task ownership, sprint planning, and traceability between requirements, design artifacts, and implementation activities.
- The project is managed by our Alpha Team responsible for approving requirements, validating design decisions, and ensuring consistency (and designing) between the SRS, UML diagrams, and implementation.

- Everyone in the Alpha team has done equal work and there is a flat hierarchy. The project follows a Shared Leadership Model where all members of the Alpha Team act as collective Project Managers. Decisions regarding system architecture, resource allocation, and project direction are made through team consensus. The **Alpha Team** consists of the following members:

  Aarushi –Work & Opportunity
  Archie –Way of Working & Team
  Riddhi – Software System & Stakeholders
  Riya – Requirements

  Approval Statement:
  Version 1.0 of the SRS was reviewed and approved by the Alpha Team. Version 2.0 incorporates review feedback and has been updated and re-approved by the Alpha Team.

2. Overall Description

2.1 Product Perspective

The QR-based Digital ID, Attendance, Library, and Gate Entry Management System is a system designed to replace existing manual and semi-digital record keeping methods used in the institution. The system provides a unified platform for identity verification, attendance tracking, and entry/exit logging using QR codes.

The system follows a client-server architecture with a centralized database server. Users interact with the system through their devices, while all data processing and storage are handled by the backend server and it will also track the records with time stamps.

It does not completely replace the guard and the TA's duties but acts as a support tool to make their work faster, more accurate, and more reliable.

2.2 Product Functions

The product functions include:-

- Generating a QR Code for the digital ID for students.
- QR based identification for students and staff.
- Entry and exit logging at campus gates, hostels, and library.
- QR-based attendance marking for registered courses.
- Display of user details after successful QR scan.
- Course-wise attendance management for faculty members.
- Role-based access control for Student, Guard, Faculty, and Admin.
- Administrative management of users, courses, and schedules.
- Storage and retrieval of attendance records and entry/exit logs.
- Report generation for monitoring and audit purposes.
- Display of error messages for invalid scans, duplicate scans, or system failures.

2.3 User Classes and Characteristics

- Students: Btech, MTech, Phd, B.Sc./B.Ed.
- Guards: Scanner,Qr scan screen
- Faculties:basic software knowledge

Students

- Are regular users of the system: Have basic knowledge of using smartphones and apps.
- Will use the system to: Show digital ID / QR code
- Mark attendance
- Enter/exit hostel, library, and campus gates
- Are not expected to have technical knowledge

Guards / Security Staff

- Are primary operational users at gates and hostels:
- Have basic training to use the scanning interface.
- Are not required to be technically skilled.
- Will mostly use: In/Out buttons, QR scan screen
- Need: Very simple, big-button interface,Clear success/error messages, Faculty / Teaching Staff , Have basic computer and mobile application knowledge.
- Use the system mainly to: View and manage attendance records, Check course-wise reports

Admin / System Administrator

- Are technically proficient.
- Responsible for: User registration, Course mapping, System configuration, Database management, Report generation
- Have: Full access privileges
- Perform: Maintenance, backups, and monitoring

2.4 Operating Environment

- The system will operate in a client-server environment within the institutional campus. The user side application can be accessed using mobile phones, tablets, or desktop computers with a camera for QR code scanning.
- The system will run on standard operating systems such as Windows, Android, iOS, and Linux. A web browser or dedicated application will be used to access the system.
- The backend server will host the application logic and the centralized database. The system requires a stable network connection (campus Wi-Fi or internet) for communication between client devices and the server.

Technology Stack

The system is developed using a client-server architecture with a web-based interface.
The following technologies are used for the development and deployment of the system:

- Frontend:

  HTML, CSS, JavaScript are used to design the user interface for students, guards, faculty, administrators.

- Backend:

  A server-side framework is used to implement QR code validation, attendance management, and entry/exit logging.

- Database:

  A relational database management system is used to store users records, attendance data, entry/exit logs, and system configurations.

- QR code Technology:

  QR codes are used for digital identification and are generated and scanned using standard QR libraries and device cameras.

- Server Environment:

  The application is hosted on a centralized server over a secure network.

- Development Tools:

  Standard development tools and version control systems are used for coding, testing, and maintenance.

2.5 Design and Implementation Constraints

- People can share each others QR code (so we need to verify the identity of the person using which QR code)
- Hardware limitations may happen sometimes
- Timing issue, database retrieval takes time if it's a large database.
- It should be secure.
- Each student is issued a single QR code for their entire degree. The QR code remains unchanged, but the user’s data might change as per the requirements. Upon graduation, the QR code is permanently invalidated. The QR code will refer to the database and the database gets modified based on that.

2.6 User Documentation

- User Manual (PDF Format):

  A detailed user manual describing system features, installation steps, login procedures, and day-to-day usage. Separate sections will be provided for different user roles such as Students, Guards, Faculty, and Admin.

- Quick Start Guide:

  A concise guide with step-by-step instructions and screenshots for common tasks such as QR code scanning, marking entry/exit, and viewing attendance.

- In-App Help and Tooltips:

  Context-sensitive help messages and tooltips embedded within the application interface to guide users during real-time usage.

- Administrator Documentation:

  Technical documentation for system administrators covering user registration, system configuration, backup procedures, cache synchronization, and troubleshooting.

- Online Help / FAQ Section:

  A searchable FAQ section addressing common user issues, errors, and resolutions.

- Documentation Formats and Standards:

  -Documentation will be delivered in PDF and digital in-app formats.

  -All documentation will follow clear, simple language and institutional documentation standards to ensure ease of understanding.

2.7 Assumptions and Dependencies

- Working camera/QR scanning device
- Human supervision is required
- Guards and staff must be trained to use the system.
- All data should be available in digital format.
- The Network should be institution LAN-based.
- If the student records database is down for maintenance then we won't be able to verify the ids so we need to keep backup etc.
- Admin will monitor records.

2.8 Development Plan & sprint strategies

- The project follows an Agile sprint-based architecture. The system is divided into functional modules. Each sprint is aiming for a duration of 1 week and includes design, implementation, testing and review.

  Note: W indicates week

Sprint Name	Key Activities/Modules	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 9	W 10
Blueprint and Foundation	• Database schema design • User data model and relationships • QR code generation rules • User registration and management • Definition of QR lifecycle and status handling	🟨	🟨							

QR Verification	• QR decoding and scanning logic • QR validation APIs • Scanner user interface • Digital ID creation and display upon successful scan • Error handling for invalid or duplicate QR scans			🟩	🟩					
Attendance and entry/exit logging	• Attendance marking using QR scanning • Entry and exit (In/Out) logging • Time-stamp based validation • Course-wise attendance mapping • Prevention of duplicate or invalid logs • Manual entry fallback in case of scan failure					🟥	🟥			
Administration and Reporting	• Admin user management • Course and schedule management • Attendance and entry/exit report generation • Log monitoring and audit features						🟦	🟦		
Integration and Testing	• Module integration • System testing and bug fixing • Performance and reliability checks • Database cleanup and archival logic • Final review and documentation								🟪	🟪

3. External Interface Requirements

3.1 User Interfaces

- The system shall provide a simple In/Out selection interface for guards, allowing them to mark entry and exit with a single tap after scanning a QR code.
- A course-wise attendance interface shall be available, where attendance is automatically coordinated with lecture timings, assigned classrooms, and course schedules.
- Guards shall have access to a scanner screen that displays user details (name, ID, photo) after QR code verification for quick confirmation.
- Faculty members shall be able to view and manage attendance dashboards, showing present/absent status for each lecture in real time.

- Students shall be provided with a digital ID interface displaying their QR code, personal details, and access status.
- The system shall include role-based interfaces (Guard, Student, Faculty, Admin), ensuring users only see features relevant to their role.
- A time-stamp and location indicator (in the current scenario, when a student leaves the campus they have to write the reason for leaving along with their location, that is what we are implying by location indicator, it is NOT a GPS Tracker, so it gets added in the system automatically, the reason for why they are leaving.) shall be shown for every scan to improve transparency and tracking.
- The interface shall support search and filter options for attendance records by date, course, or student ID.
- Admin users shall have access to reports and logs through a graphical interface for monitoring entry/exit and attendance data.
- The user interface shall be designed to be responsive and user-friendly, accessible via mobile devices, tablets, and desktop systems.
- Error and alert messages shall be displayed clearly in cases of invalid QR codes, duplicate scans, or unauthorized access attempts.

3.2 Hardware Interfaces

- A QR code scanner shall be necessary, or phone scanner may be accessed.
- A PC shall be provided to the guard to access the application.

3.3 Software Interfaces

- Everytime a QR is scanned there shall be a data retrieval from the main database in which all the students' records are stored.
- User verification shall be required to access personal data.
- A good and secure network connection shall be used.
- The system shall be integrated with academic schedules.

3.4 Communications Interfaces

- The system shall communicate between user devices and the server using a network connection.
- The system shall use standard internet or campus Wi-Fi for data transmission.
- The system shall use secure communication protocols for transferring data between the client and the server.
- All data exchanged between the client and the server shall be transmitted in a structured digital format.
- The system shall ensure reliable communication for real-time operations such as QR scanning, attendance marking, and entry/exit logging.
- In case of network failure, the system should display an appropriate error message to the user.

4. System Features

4.1 Automated QR Access Validation

4.1.1 Description and Priority

This feature allows students and staff to gain entry by scanning the static QR code printed on their physical ID cards/phones. The system must decode the QR hash and verify the user's status in the database.

Priority: High

4.1.2 Stimulus/Response Sequences

- **Stimulus:** User presents ID card to the CMOS QR scanner.
- **Response:** System decodes the hash, queries the database, and triggers the gate unlock signal if valid.
- **Stimulus:** System detects an invalid or expired QR hash.
- **Response:** System keeps the gate locked and displays a VIOLATION message to the guard.

4.1.3 Functional Requirements

- **FR-1:** The system shall decode QR codes.
- **FR-2:** The validation service must return a response from the database in a given time interval.
- **FR-3:** The system shall display the student's registered photo on the guard's monitor upon a successful scan for visual verification.
- **FR-4:** The system shall assign exactly one QR code to a user for their entire academic tenure.
- **FR-5:** The system shall maintain a user status (Active, Graduated, Blocked) and also allow admin to archive records of graduated students.
- **FR-6:** The system shall invalidate the QR code when the user status becomes Graduated. The database shall maintain the status of users (Active, Graduated, Archived). Records of graduated students shall be archived by the System admin and may be purged after a configurable retention period. This software is just an entry exit mechanism that uses digital id for currently enrolled students. This does not act as a ‘Students’ Records’, hence, will not contain alumni details.

4.2 Anti-Passback Control

4.2.1 Description and Priority

This feature prevents card sharing by tracking the logical state (Inside/Outside) of each user. It ensures a card cannot be used to enter twice without an intervening exit. Priority: High

4.2.2 Stimulus/Response Sequences

- Stimulus: A user already marked as "Inside" scans their card at an entry gate.
- Response: The system denies entry and logs a "Passback Violation" event in the database.

4.2.3 Functional Requirements

- **FR-7:** The system shall maintain a current_status flag for every user in the database.
- **FR-8:** The system shall automatically reset a user's status to "Outside" only after a successful exit scan is recorded.

4.3 Manual Guard Override and Exception Handling

4.3.1 Description and Priority

This feature provides a web-based interface for security personnel to manually trigger the gate in cases of damaged ID cards or system emergencies. Priority: Medium.

4.3.2 Stimulus/Response Sequences

- Stimulus: Guard enters a student's Roll Number into the override terminal.
- Response: System retrieves the user record; Guard clicks "Authorize Entry" to unlock the gate and create a manual log entry.

4.3.3 Functional Requirements

- **FR-9:** The system shall require guard authentication (username/password) before accessing override functions.
- **FR-10:** Every manual override must be logged with the Guard’s ID and a mandatory reason code.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- NFR-01:The system shall be able to manage a considerable number of students with simultaneous access requests within a short time span.
- NFR-03:Scanning of QR codes, identity verification, and entry/exit recording shall be completed within an acceptable response time to prevent congestion at gates, hostels, and access points.
- NFR-03:The system shall support high-throughput operations during peak usage periods such as class start/end times, hostel curfews, and library rush hours.

5.2 Safety Requirement

- NFR-04:The system shall guarantee that no incorrect or unauthorized access is granted due to system failure, network issues, or data corruption.
- NFR-08:In the event of central server downtime, the system shall operate safely using locally cached authorized IDs to prevent disruption or unsafe crowd formation.The system shall ensure that no entry or exit records are lost by temporarily storing them locally until synchronization with the central server is restored.
- NFR-09:Manual verification by guards shall always be available as a fallback mechanism in case of system or scanner failure to ensure student safety.

5.3 Security Requirements

- NFR-01:The system shall provide a secure local cache of authorized student IDs accessible only to authorized system components.All locally stored data and cached logs shall be protected from unauthorized access and tampering.The system shall maintain accurate student identity verification even during offline operation.
- NFR-07:Upon restoration of network connectivity, the system shall synchronize locally stored logs with the central database without loss or duplication of data.

5.4 Software Quality Attributes

- NFR-04 (Reliability):The system shall continue operating during central server downtime using local caching and backup mechanisms.
- NFR-04 (Availability):The system shall be highly available, particularly during peak operational periods.
- NFR-05 (Scalability):The system shall scale to accommodate increasing numbers of students and access requests.
- NFR-06 (Robustness):System crashes or data loss shall not occur due to temporary network disconnections.
- NFR-03 (Performance Efficiency):The system shall prioritize fast QR scanning and verification over non-critical background processes.
- NFR-07 (Maintainability):Caching and synchronization components shall be modular and easy to maintain or debug.

5.5 Business Rules

- NFR-01:Only students whose IDs exist in either the central database or the locally cached authorized list shall be permitted entry or exit.
- NFR-08:During central server unavailability, the system shall permit access only to previously verified and cached student IDs.
- NFR-07:All entry/exit transactions recorded during offline operation shall be synchronized with the central database once connectivity is restored.
- NFR-10:Guards and administrators shall not directly modify cached authorization data except through approved administrative procedures.

6. Other Requirements

- Use Cases
    - It can be used in attendance systems during lectures
    - Entry/exit at main gate+hostels
    - Mess in/out
    - Library in/out

- Logical Database Requirements-
    - takes data from ERP profile of students
    - central database
    - Stores all user information
    - Stores courses registered
    - entry/exit logs
    - QR scan timestamps

- Design and Coding Standards
    - Function naming: camelCase()
    - Class naming: PascalCase
    - Constants: UPPER_CASE
    - FR naming: REQ-1
    - NFR naming: NFR-1
    - API naming pattern: /api/scanQR, /api/markAttendance

- **Traceability Matrix**

Item	**In Version 1 (V1)?**	**In Version 2 (V2)?**	**Change Description**	**Reason**
QR Scan Flow	Yes	Yes	Guards need not do manual entry unless the QR scanning system fails.	Instructor feedback (provided Clarification)
QR Lifecycle	No	Yes(Added)	Defined QR validity across academic years	Missing in V1
Database Cleanup/flushing	Yes (in the use-case diagram)	Yes (added in the srs template)	Added graduation data purge requirement	Instructor Feedback
Location Indicator	Yes	Modified	Made optional / derived	Physical scan makes it redundant
Sprint Plan	No	Yes	Added	Required in V2
Templates	No	Yes	Added	Required in V2

- **Traceability Matrix Functional Requirements**

FR ID	**Functional Requirement Description**	**Components (based on component uml diagram in appendix B)**	**Responsibility holder**	**Source/Feature**	**UML Diagram Mapping**
FR-1	Decode QR codes	Unified App (Student/Guard View), QR Processor	Riya	QR Processing	Use Case, Sequence
FR-2	Validation service returns DB response within a time interval	API Gateway, QR Processor	Aarushi	Validation and Performance	Sequence, Activity
FR-3	Display student's registered photo on	Unified App (Guard View), Data Access	Archie	Identity Verificati	Sequence

	guard’s screen after successful scan	Layer		on	
FR-4	Assign exactly one QR code per user for entire academic tenure	Auth Service, Data Access Layer	Riddhi	QR Lifecycle	Class, Use Case
FR-5	Maintain user status (Active, Graduated, Blocked) and allow archiving	Auth Service, Data Access Layer	Archie	User Status Management	Class, Component
FR-6	Invalidate QR on Graduation; archive/purge graduated records; no alumni storage	Data Sync Service, Data Access Layer	Riddhi	Data Lifecycle & Retention	Class, Sequence, Component
FR-7	Maintain current_status flag for every user	Campus Actors Database, Data Access Layer	Riya	Presence Tracking	Class, Database
FR-8	Reset user status to "Outside" only after successful exit scan	QR Processor, Data Access Layer	Aarushi	Entry/Exit Logic	Sequence, Activity
FR-9	Require guard authentication before override functions	Auth Service, Unified App (Guard View)	Riya, Aarushi	Security & Access Control	Use Case, Component
FR-10	Log every manual override with Guard ID and reason code	Auth Service, Data Access Layer	Archie, Riddhi	Audit Logging	Sequence, Activity

- **Traceability Matrix Non-Functional Requirements**

NFR ID	Non-Functional Requirement	Components (based on component uml diagram in appendix b)	Responsibility holder	Category	Source	System Features	Design /UML Artifacts	Version Update
NFR-1	The system shall authenticate users before allowing access to	API Gateway, Auth Service	Archie	Security	Prevent misuse of shared QR codes	Digital ID, QR Scan	Use-Case, Sequence Diagra	V2.0-classified

	personal data.						m	
NFR-2	The system shall ensure secure data transmission between client and server.	API Gateway, Unified App	Riya	Security	Protect sensitive student records	All modules	Component, Deployment Diagram	V2.0-unchanged
NFR-3	The system shall respond to a QR scan request within 2 seconds under normal network conditions.	QR Processor, Data Access Layer	Riddhi	Performance	Smooth workflow of guard & attendance	Gate Entry, Attendance	Timing Diagram	V2.0 – added
NFR-4	The system shall be available during institutional working hours with minimal downtime.	API Gateway, Backend System (All Services)	Aarushi	Availability	Daily academic usage	All modules	Deployment Diagram	V2.0 – unchanged
NFR-5	The system shall support mobile, tablet, and desktop devices.	Unified App, rest api	Riya	Compatibility	Multi-device access	Digital ID, Admin Dashboard	Deployment Diagram	V2.0 – unchanged
NFR-6	The system should be easy to use by guards with minimal training.	Unified App (Guard View)	Archie	Usability	Guard efficiency	Gate Entry Module	Activity Diagram	V2.0 – refined
NFR-7	The system shall store attendance and entry logs in a structured and searchable format.	Data Access Layer, Campus Actors Database	Riddhi	Maintainability	Reporting & audits	Attendance, Gate Entry	Logical DB Design	V2.0 – unchanged
NFR-8	The system shall automatically deactivate records of graduated students.	Data Sync Service, Data Access Layer	Aarushi	Data Management	Instructor feedback	Student Management	Class Diagram	V2.0 – added
NFR	The system	Unified App	Riya, Archie,	Reliabilit	Instructo	Gate	Use-Ca	V2.0 –

-9	shall allow manual guard entry as a fallback when QR scanning fails.	(Guard View), Auth Service	Riddhi, Aarushi	y	r Feedback	Entry Module	se Diagram	clarified

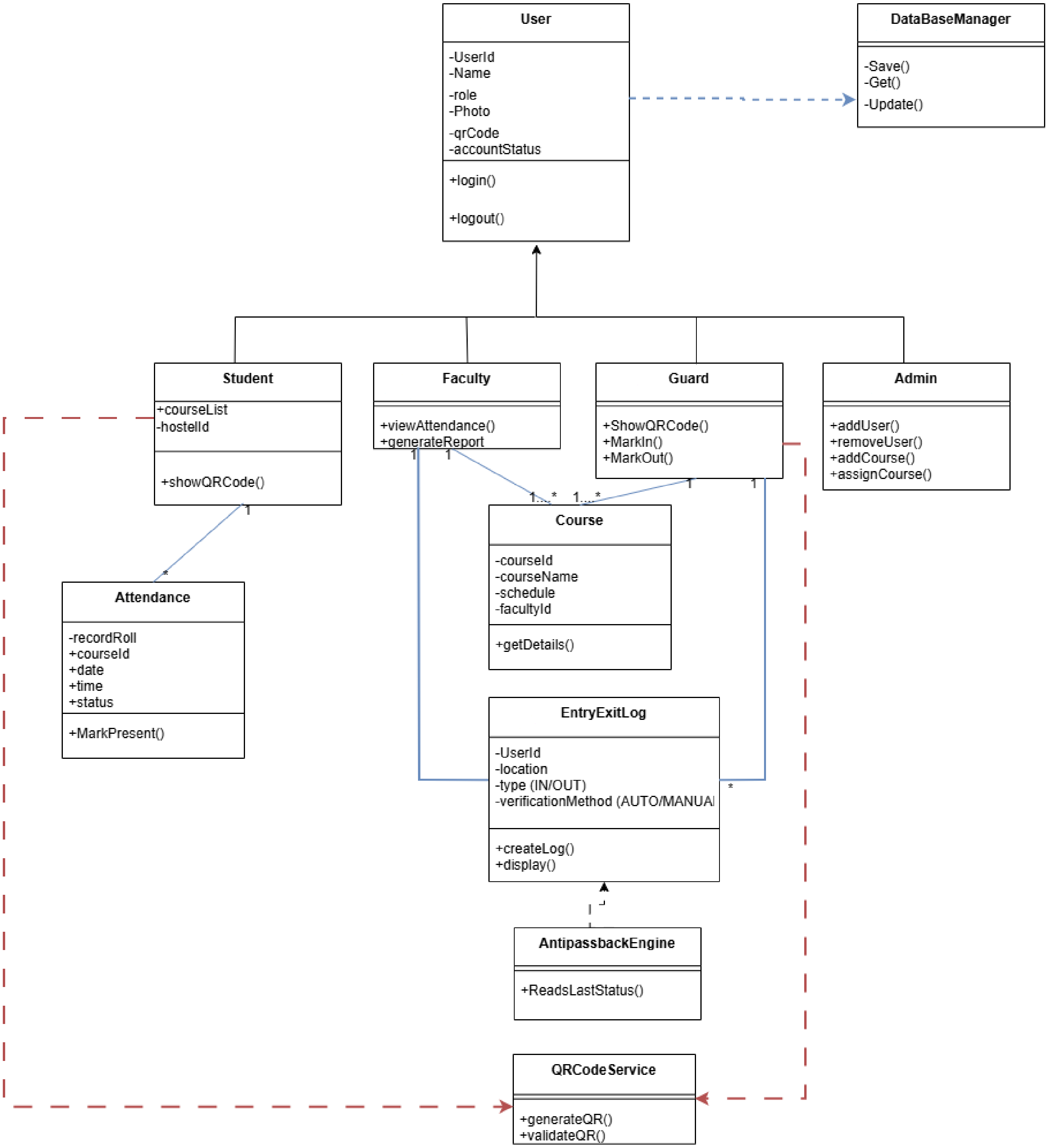
Appendix A: Glossary

- QR- Quick Response
- ERP- Enterprise Resource Planning
- Anti-Passback system- is a security measure that aims to prevent consecutive entries from one qr code and prevent multiple people from using the same qr code.

Appendix B: Analysis Models

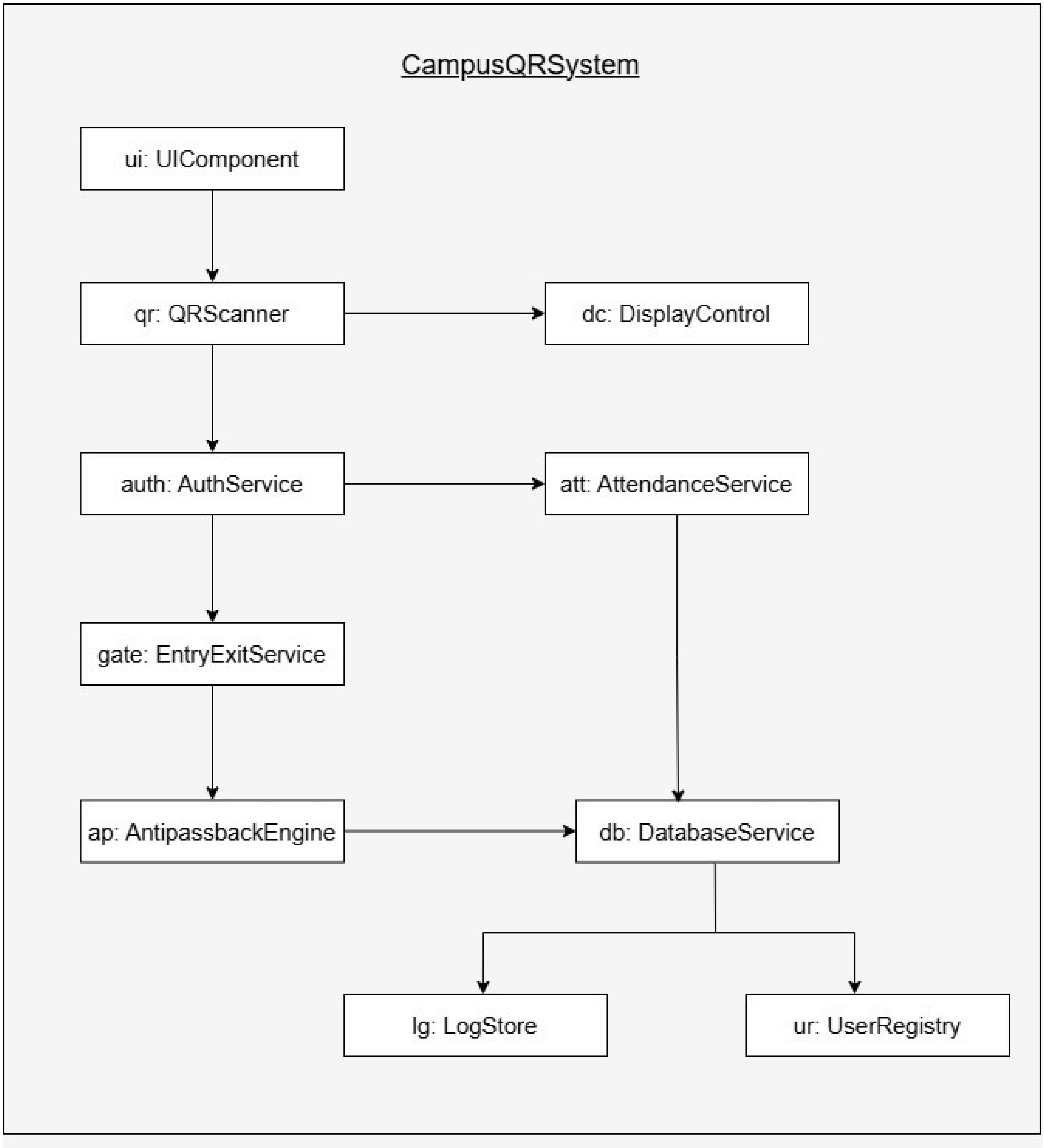
PFA UML diagrams

CLASS DIAGRAM

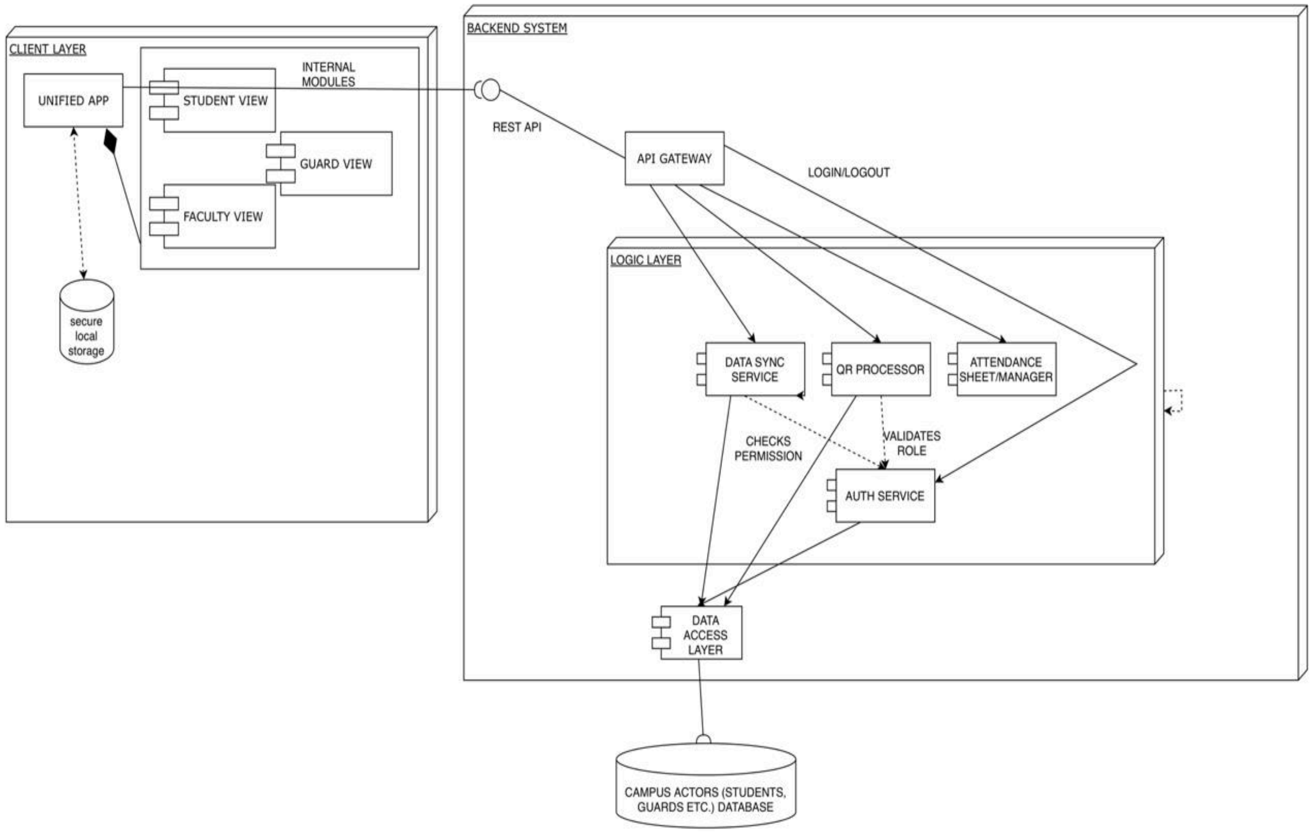


.

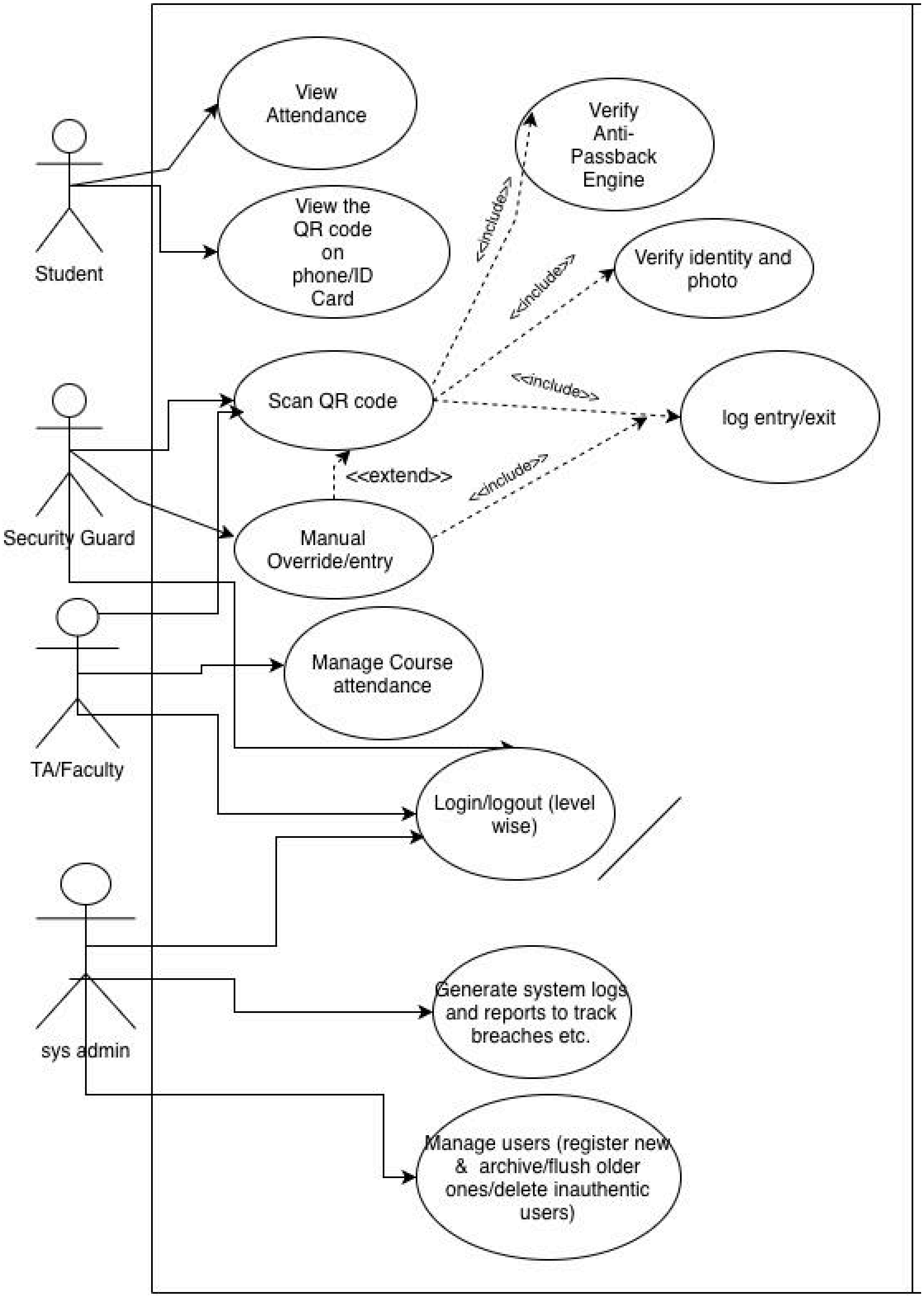
COMPOSITE DIAGRAM



COMPONENT DIAGRAM

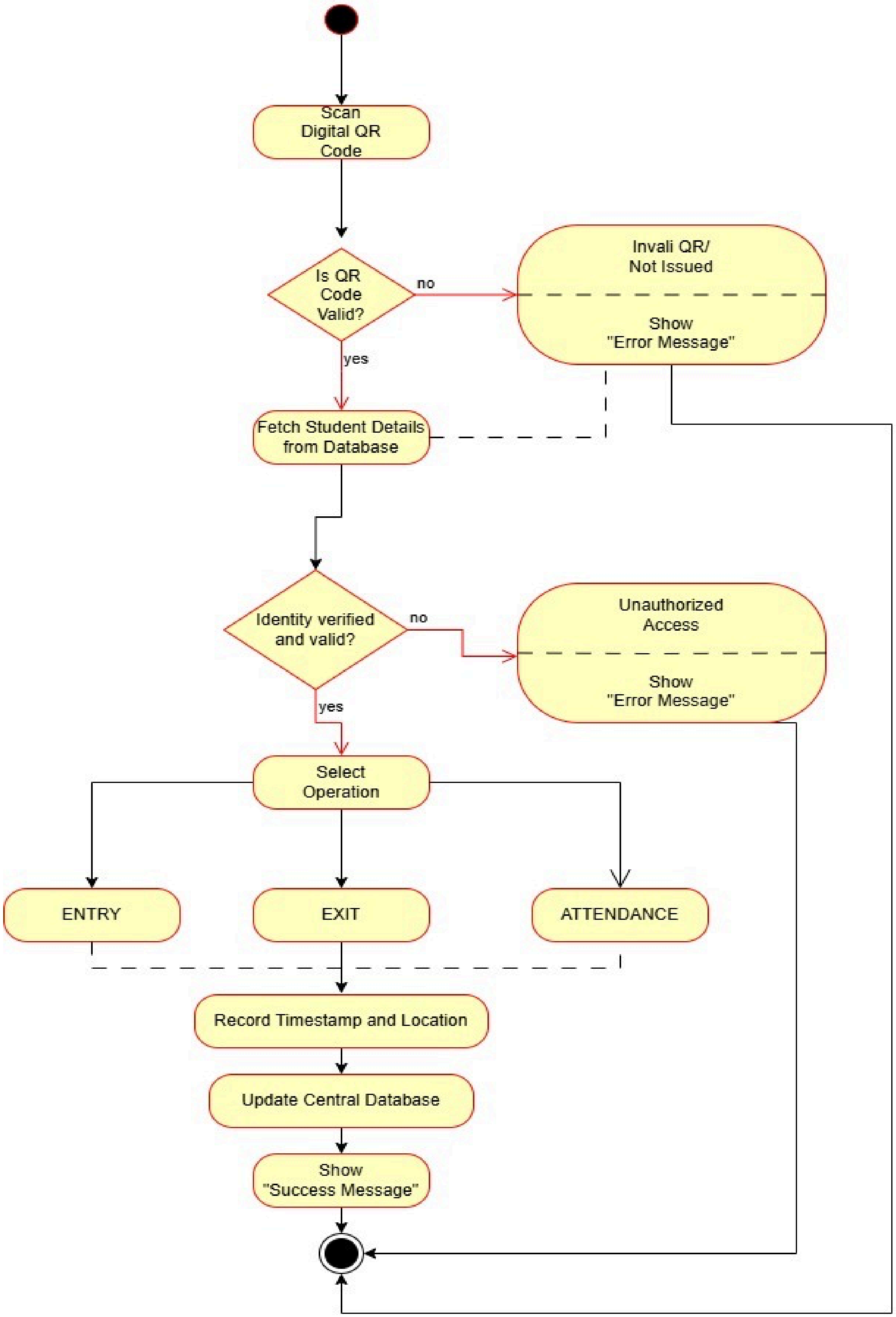


USE-CASE DIAGRAM

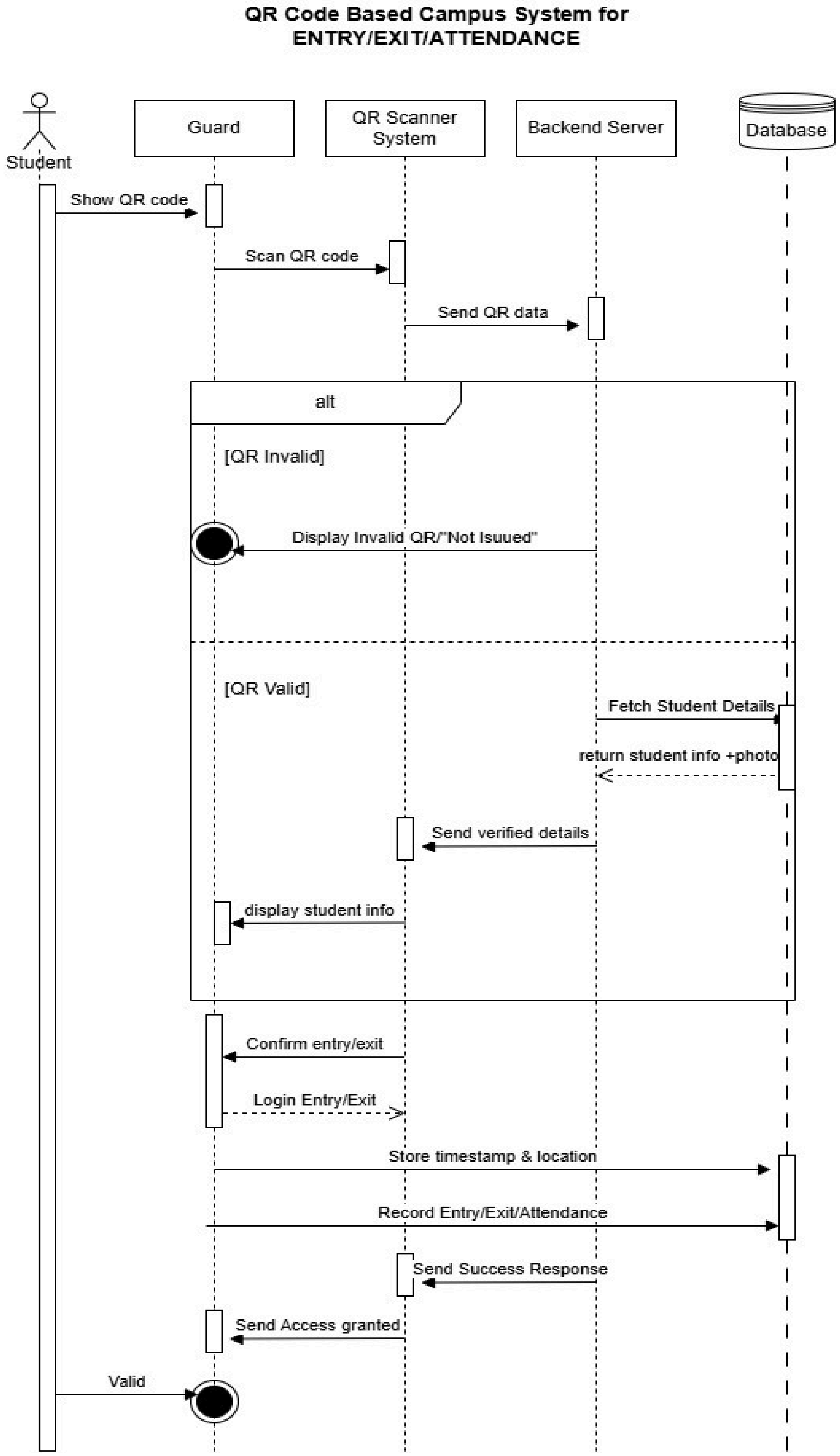

View Attendance
View the QR code on phone/ID Card
Verify Anti-Passback Engine
Verify identity and photo
Student
<<include>>
<<include>>
<<include>>
Scan QR code
log entry/exit
<<extend>>
<<include>>
Security Guard
Manual Override/entry
Manage Course attendance
TA/Faculty
Login/logout (level wise)
Generate system logs and reports to track breaches etc.
sys admin
Manage users (register new & archive/flush older ones/delete inauthentic users)

ACTIVITY DIAGRAM

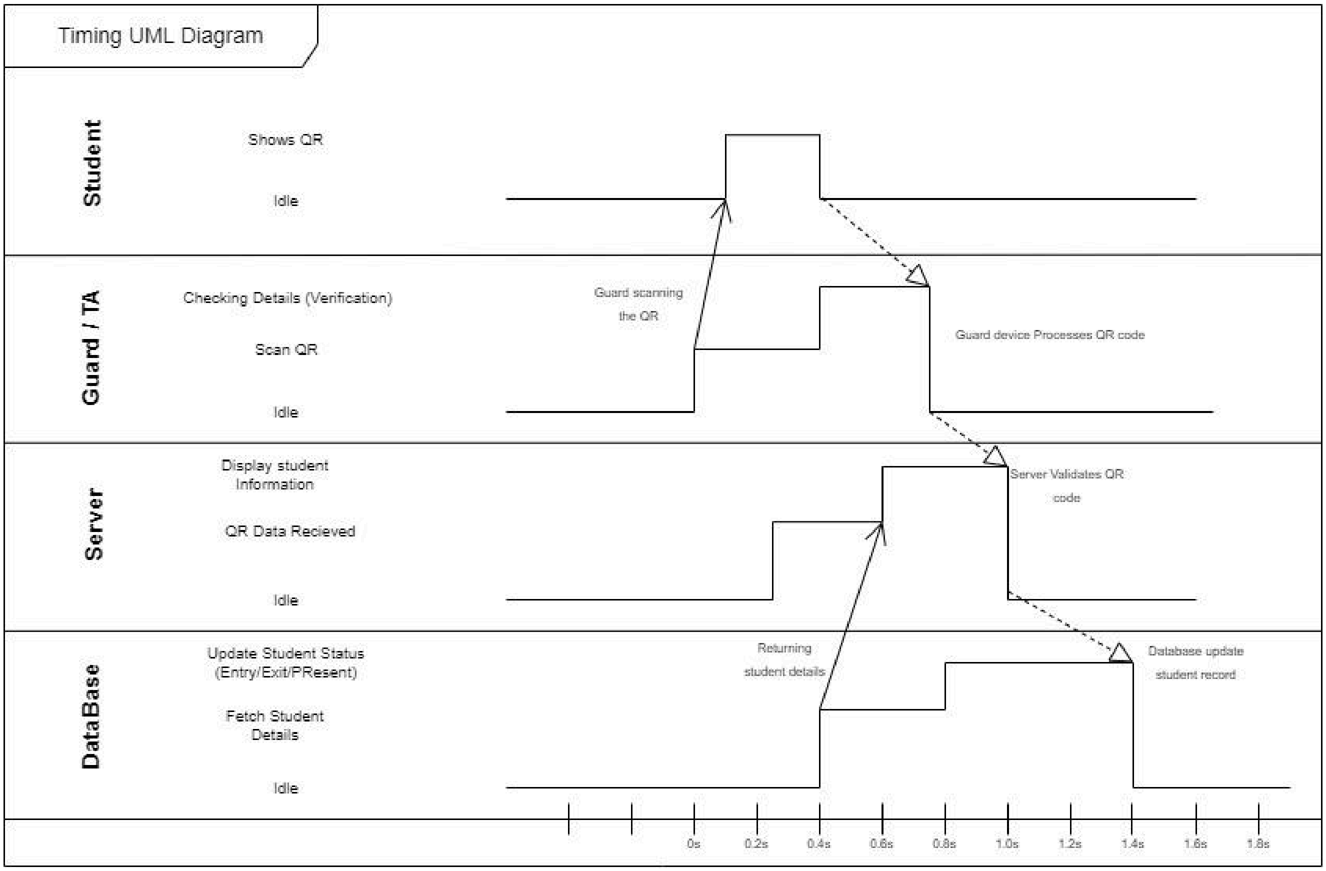
QR Code Based Campus System for ENTRY/EXIT/ATTENDANCE



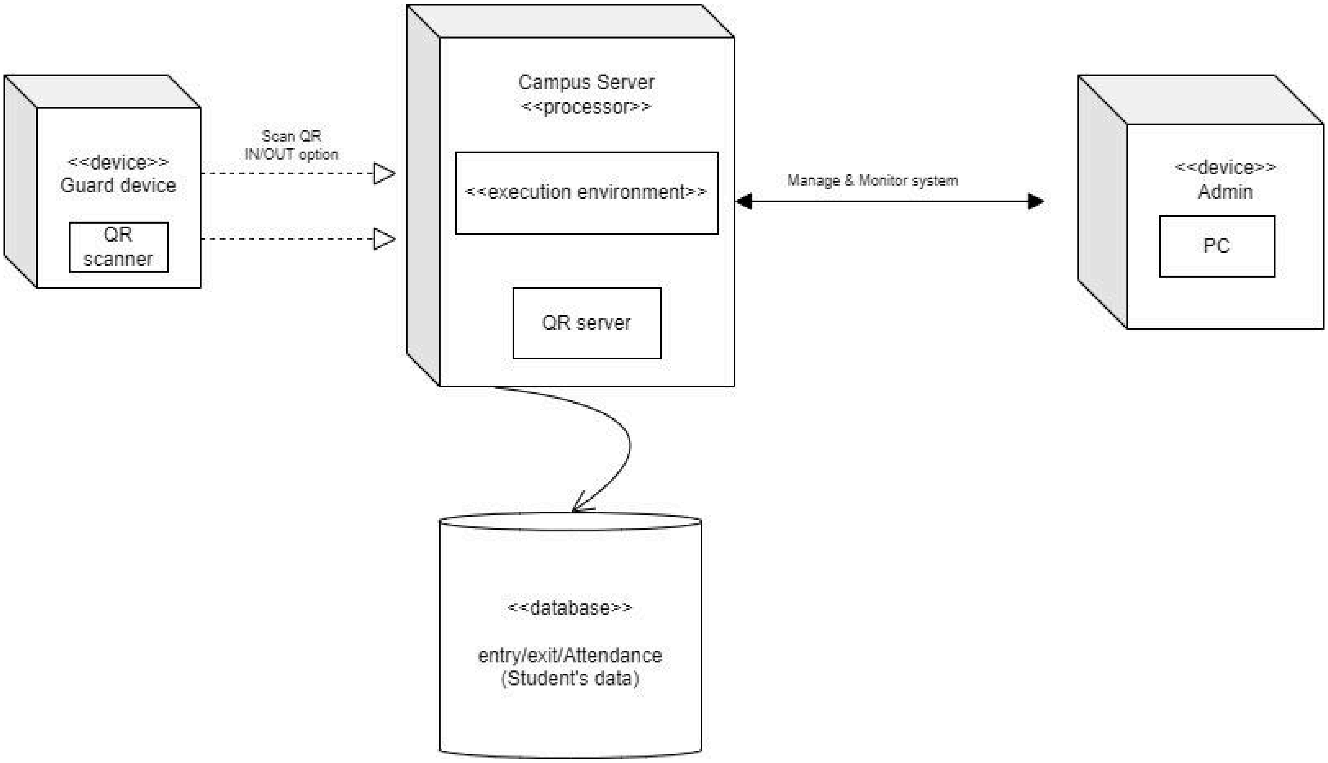
SEQUENCE DIAGRAM



TIMING DIAGRAM



DEPLOYMENT DIAGRAM



Appendix C: To Be Determined List

- Documentation for the project.
- And making the project itself.