



# CSE6224 Software Requirement ENG

## Project Part 1

**Title: Campus Event Check-in System (CECS)**

**Prepared By:**

Student Name	Student ID	Email	Phone Number
ERIC TEOH WEI XIANG	1221102007	1221102007@student.mmu.edu.my	017-406 3708
PANG TIAN YOU	1231303394	1231303394@student.mmu.edu.my	011-10943832
SIOW YI LING	1211107982	1211107982@student.mmu.edu.my	011-10923772
LIM KAI SHEN	1211110602	1211110602@student.mmu.edu.my	012-2153379

# Table of Contents

Table of Contents.....	2
<b>1. Introduction.....</b>	<b>5</b>
1.1 Purpose.....	5
1.2 Scope.....	6
1.3 Product Overview.....	7
1.3.1 Product Perspective.....	7
1.3.1.1 System Interfaces.....	11
1.3.1.2 User Interfaces.....	12
1.3.1.3 Hardware Interfaces.....	13
1.3.1.4 Software Interfaces.....	14
1.3.1.5 Communication Interfaces.....	15
1.3.1.6 Memory Constraints.....	15
1.3.1.7 Operations.....	16
1.3.1.7.1 Student Operations.....	16
1.3.1.7.2 Admin Operations.....	17
1.3.1.8 Site Adaptation Requirements.....	19
1.3.1.9 Interface with Services.....	20
1.3.2 Product function.....	21
1.3.3 User Characteristics.....	26
1.3.4 Limitations.....	26
1.3.5 Apportioning of requirements.....	28
1.4 Definition.....	29
<b>2 References.....</b>	<b>30</b>
3.1 Functions.....	31
3.1.1 Sequence Diagram.....	31
3.1.1.1 Student Login.....	31
3.1.1.2 Admin Login.....	33
3.1.1.3 View Event List (Student).....	35
3.1.1.4 Register for Event (Student).....	36
3.1.1.5 Make Payment (Student).....	37
3.1.1.6 Make Onsite Payment (Student).....	38
3.1.1.7 Make Online Payment (Student).....	39
3.1.1.8 View Payment Receipt (Student).....	41
3.1.1.9 Request Refund (Student).....	42
3.1.1.10 Check-in Via QR Code (Student).....	43
3.1.1.11 Rate Event (Student).....	44
3.1.1.12 Manage Event (Admin).....	45
3.1.1.13 Create Event (Admin).....	46
3.1.1.14 Generate QR Code (Admin).....	47
3.1.1.15 View Rating or Feedback (Admin).....	48
3.1.1.16 Track Attendance (Admin).....	49

3.1.1.17 View Attendance Analytic (Admin).....	50
3.1.1.18 Generate Payment Report (Admin).....	51
3.1.1.19 Handle Request Refund (Admin).....	52
3.2 Performance Requirements.....	53
3.3 Usability Requirements.....	55
3.4 Interface Requirements.....	57
3.4.1 External Interfaces.....	57
3.4.1.1 SR01 Login Page.....	57
3.4.1.2 SR02 Event List.....	59
3.4.1.3 SR03 Event Payment.....	63
3.4.1.4 SR04 Refund Payment.....	66
3.4.1.5 SR05 Rating.....	68
3.4.1.6 SR06 Scan Attendance.....	70
3.4.1.7 SR07 Admin Dashboard.....	72
3.4.1.8 SR08 Create Events.....	74
3.4.1.9 SR09 Manage Events.....	76
3.4.1.10 SR10 Track Attendance.....	77
3.4.1.11 SR11 View Rating & Feedback.....	80
3.4.1.12 SR12 View Payment Details.....	81
3.4.1.13 SR13 View Request Refund.....	83
3.4.2 System Interfaces.....	84
3.4.3 User Interfaces.....	85
3.4.4 Hardware Interfaces.....	87
3.4.5 Software Interfaces.....	89
3.4.6 Communication Interfaces.....	90
3.5 Logical database requirements.....	91
3.5.1 Student.....	93
3.5.2 Admin.....	93
3.5.3 Event.....	93
3.5.4 Payment.....	94
3.5.5 Feedback.....	94
3.5.6 Receipt.....	95
3.5.7 Refund Request.....	95
3.5.8 QRCode.....	96
3.6 Design Constraints.....	97
3.7 Software System Attributes.....	99
3.7.1 Availability.....	99
3.7.2 Reliability.....	99
3.7.3 Security.....	100
3.7.4 Auditability.....	100
3.7.5 Performance.....	101
3.7.6 Maintainability.....	101
3.7.7 Portability.....	102
3.7.8 Usablity.....	102

3.7.9 Responsiveness.....	103
3.7.10 Scalability.....	103
3.8 Supporting Information.....	104
3.8.1 Questionnaire Analytics.....	108
3.8.2 Prototyping.....	116
3.9 Standard Compliance.....	128
4.1 Verification.....	129
4.1.1 Availability Requirements Verification.....	129
4.1.2 Reliability Requirements Verification.....	129
4.1.3 Security Requirements Verification.....	129
4.1.4 Auditability Requirements Verification.....	130
4.1.5 Performance Requirements Verification.....	130
4.1.6 Maintainability Requirements Verification.....	130
4.1.7 Portability Requirements Verification.....	130
4.1.8 Usability Requirements Verification.....	131
4.1.9 Responsiveness Requirements Verification.....	131
4.1.10 Scalability Requirements Verification.....	131
4.1.11 Accuracy Requirements Verification.....	131
4.2 Verification Criteria.....	132
5 Appendices.....	134
5.1 Assumptions and Dependencies.....	134
5.2 Acronyms and Abbreviations.....	135

# 1. Introduction

## 1.1 Purpose

The purpose of the Campus Event Check-in System (CECS) is to provide a mobile-first solution that simplifies and automates the process of campus event participation for both students and administrators. The current reliance on manual event registration, attendance tracking, and payment processing creates inefficiencies, delays, and potential for errors. This system addresses those challenges by offering an integrated platform accessible via mobile devices.

The system allows students to browse and register for university events through their smartphones, make secure payments using preferred methods such as FPX or Touch 'n Go eWallet, and verify attendance by scanning a QR code generated by the admin. It ensures a smooth experience by sending real-time notifications and enabling students to view receipts, request refunds, and submit feedback or ratings after event participation.

For event organizers (admins), the system provides tools to create and manage events, generate and distribute QR codes, monitor live attendance, process refund approvals, and generate comprehensive reports for attendance and payments—all from a unified admin dashboard.

Requirements for the system were collected through questionnaires and mobile UI prototypes, which were reviewed by key stakeholders including students and administrative staff. The system is designed to integrate with existing authentication and payment servers used by the university, ensuring secure access and reliable transaction handling.

By digitizing these core event processes, the system contributes to the university's broader digital transformation goals—enhancing operational efficiency, reducing administrative workloads, and promoting a more connected and engaging campus culture through mobile accessibility.

## 1.2 Scope

The Campus Event Check-in System (CECS) is a mobile-based software product designed to allow students to view available events and choose which ones to join. The system manages campus event registration, attendance tracking, secure payments, and feedback collection. It integrates with the university's Authentication Server and Payment Server, enabling students to log in via mobile, register for events, and verify their attendance using QR codes alongside their Student ID information.

Students can choose between two payment methods: online payment (via FPX or TNG eWallet) and on-site payment. The system also supports features such as viewing payment receipts, requesting refunds, and receiving real-time notifications after successful registration. After attending an event, students will be prompted to submit ratings and feedback through the mobile interface.

Admins, via their mobile interface, can create and manage events, generate QR codes, track attendance in real-time, view feedback, handle refund requests, and generate payment reports.

The system is only designed to work on mobile devices and can't be used on a computer or through a web browser. It will be applied within the university setting to support a variety of academic, cultural, and extracurricular events.

The goal is to digitize and simplify the event participation process, eliminate manual errors, reduce administrative effort, and enhance the student experience by providing a smooth, mobile-first solution.

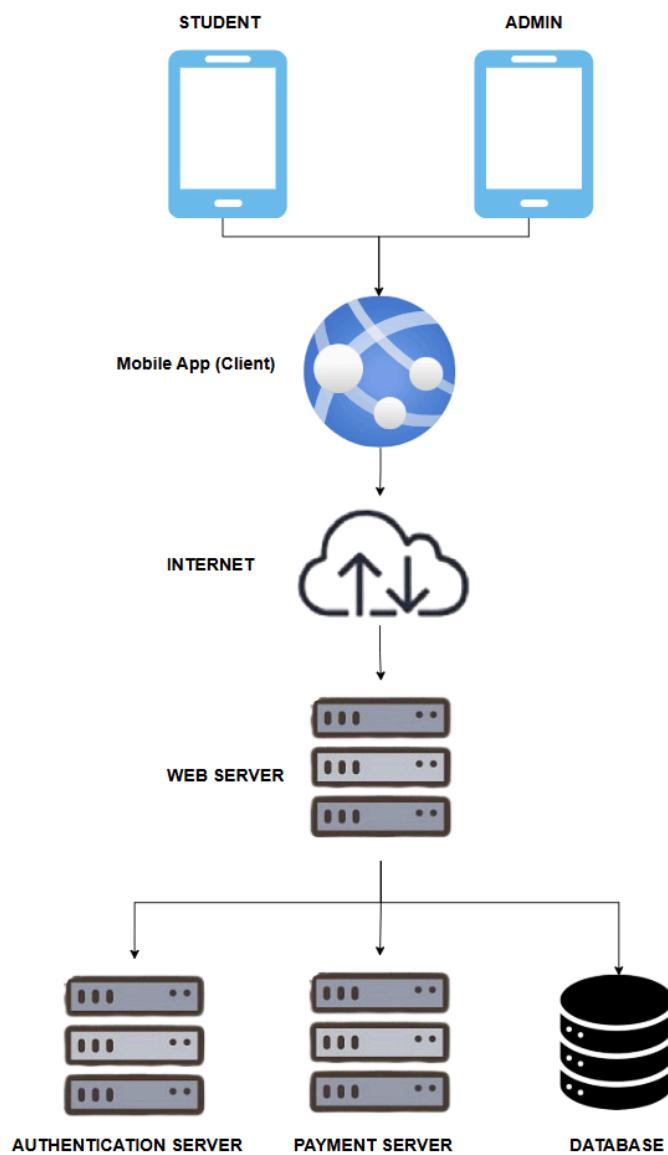
This system aligns with the university's objective to modernize campus operations through mobile technology—improving accessibility, operational efficiency, and overall event engagement.

## 1.3 Product Overview

### 1.3.1 Product Perspective

This section provides an overview of the Campus Event Check-in System (CECS), a mobile-based platform developed to streamline event attendance tracking and integrate securely with the university's student identification system and payment processing gateway. The system is designed to offer a seamless and efficient experience for both students and administrators in managing campus events.

The following diagram illustrates the architecture and key components of the Campus Event Check-in system.



**Figure 1.3.1.1: System Overview Diagram**

The Campus Event Check-in System (CECS) functions as a centralized extension of the university's existing digital ecosystem. It supports secure student authentication, mobile-based check-ins, and integrated payment processing. This infrastructure improves operational efficiency and enhances the student experience during event registration, participation, and post-event interactions.

Students and administrators access the system via a dedicated mobile application, which communicates with the university's Web Server over the Internet. This Web Server coordinates requests to and from external components, including the Authentication Server and Payment Server, ensuring secure and role-based access.

## Core Functionalities

### 1. View Event List

- Students can view a list of all available and upcoming campus events directly on the mobile dashboard.

### 2. Event Registration and Notifications

- Students can browse upcoming events and register through the app.
- Upon successful registration, the system automatically sends event confirmation and reminder notifications.

### 3. Integrated Payment Processing

- The system supports both online (e.g., FPX, TNG eWallet) and on-site payments.
- It connects to the Payment Server to handle secure transactions.
- Students receive digital receipts, and may also request refunds.
- Admins are responsible for managing and approving refund requests.

### 4. QR Code-Based Check-In

- A unique QR code is generated for each event.
- On event day, students scan the QR code using their mobile device and input their Student ID, Name, and Ticket ID.
- This enables fast, accurate, and real-time attendance tracking.

### 5. Event Feedback and Rating

- After the event, students are prompted to submit a rating and feedback.
- This data helps organizers evaluate event quality and improve future events.

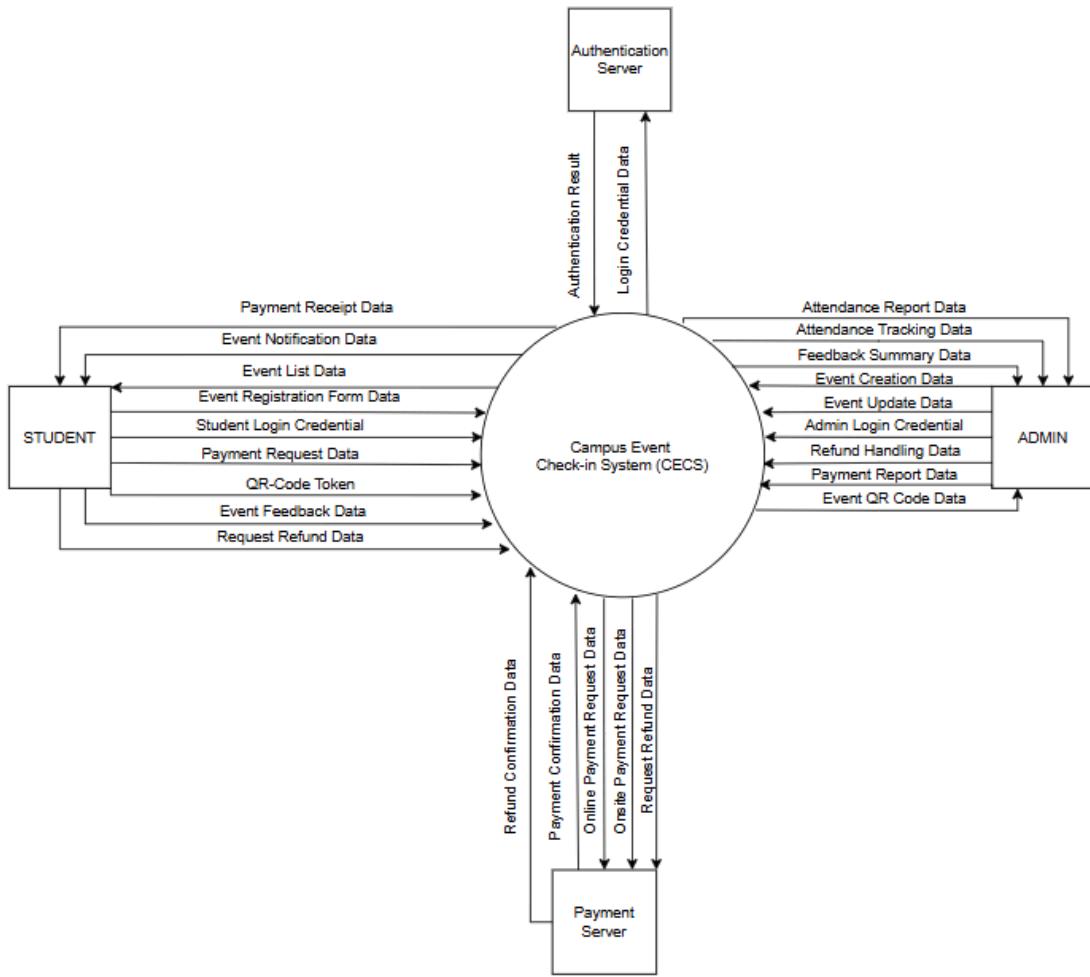
### 6. Admin Event Management

- Admins can create, edit, and delete events.
- Generate QR codes for check-in.
- Track real-time attendance and view student feedback.
- Generate detailed events and financial reports.
- Handle refund approval and monitor payment records.

### 7. Real-Time Data and Reporting

- Admins have access to a reporting dashboard with attendance statistics, payment history, and feedback analysis.
- This supports data-driven decision-making and improves event operations.

The following diagram illustrates the System Context Diagram, showing the interaction between the system, external servers (Authentication and Payment), and user roles (Student and Admin).



**Figure 1.3.1.2: System Context Diagram**

Aligned with the university's mission of enhancing student engagement and streamlining campus operations, the Campus Event Check-in System (CECS) is designed to fulfill a set of key requirements that address the evolving needs of both students and admins.

**Table 1.3.1.1 Functional Requirements for Campus Event Check-in System (CECS)**

Requirement ID	Goals	Author
REQ_CECS_001	The system shall allow students and admins to log in securely using university credentials.	YILING
REQ_CECS_002	The system shall enable admins to create, manage, and update campus events.	ERIC
REQ_CECS_003	The system shall generate QR codes for event check-ins and link them to attendance records.	TIANYOU
REQ_CECS_004	The system shall allow students to register for events and make payments (online or onsite).	KAISHEN
REQ_CECS_005	The system shall integrate with external payment servers for handling transactions and refunds.	KAISHEN
REQ_CECS_006	The system shall provide real-time attendance tracking for admins during event check-ins.	ERIC
REQ_CECS_007	The system shall enable students to check in to events using QR code scanning.	YILING
REQ_CECS_008	The system shall allow students to rate and provide feedback after attending events.	TIANYOU
REQ_CECS_009	The system shall send notifications for event confirmations to students.	ERIC
REQ_CECS_010	The system shall allow admins to generate attendance and feedback reports.	YILING
REQ_CECS_011	The system shall allow students to view a list of upcoming events with relevant details.	KAISHEN
REQ_CECS_012	The system shall verify student ID, name, and ticket ID during QR code check-in.	TIANYOU

By addressing these requirements, the CECS aims to streamline campus event participation, ensure secure and efficient check-ins, and support data-driven event management for both students and administrators.

### **1.3.1.1 System Interfaces**

The Campus Event Check-in System (CECS) works smoothly by connecting with different internal and external systems, making it easier for users and helping manage events more easily. These interfaces are really important for things like logging in safely, handling money transactions, keeping track of who's there in real time, and sending notifications inside the app. The system connects with the university's authentication server to allow students and administrators to log in using their official credentials. It connects to a payment server that handles online payments like FPX and Touch 'n Go eWallet. It also takes care of in-person payments and refunds.

The CECS system also interacts with the university database to validate student IDs, retrieve event information, and store attendance records. To support efficient check-ins, a QR Code Module generates unique codes for each event, allowing students to scan and record their attendance quickly via the mobile app. Besides, the app includes a built-in in-app pop-up notification system, which informs users about event confirmations and updates while they are actively using the app. In addition, a report generator compiles key metrics related to attendance, payments, and user feedback, while a receipt generation API provides digital payment confirmations within the student dashboard. Together, these interfaces create a secure and user-friendly platform aligned with the university's goal of modernizing campus event experiences.

### **1.3.1.2 User Interfaces**

The Campus Event Check-in System (CECS) features a clean and mobile-friendly graphical user interface (GUI) designed to provide a smooth and intuitive experience for both students and administrators. The interface prioritizes ease of use, clarity, and responsiveness across all key modules, supporting a wide range of user interactions within the application.

Students can easily log in using their university credentials through a secure login interface, browse and filter upcoming events, register for events, select payment methods, and view digital receipts. The system also provides a dedicated interface for QR code check-in, where users can input their Student ID, Name, and Ticket ID. After attending an event, students can submit feedback and request refunds through simple, well-structured forms. All actions are accompanied by real-time in-app notifications that confirm successful registration, payments, and event updates.

On the admin side, CECS includes powerful interfaces for event creation, QR code generation, attendance monitoring, and reporting. Admins can manage event details, track live attendance during events, and generate reports based on attendance, payments, and student feedback. By ensuring consistency across these interfaces, the system not only supports efficient event operations but also enhances the overall user experience for all stakeholders.

### **1.3.1.3 Hardware Interfaces**

The Campus Event Check-in System (CECS) is designed to run efficiently on modern mobile devices, with specific hardware requirements to ensure smooth functionality and an optimal user experience. These hardware specifications are essential for supporting key features such as QR code scanning, in-app real-time notifications, event registration, and responsive interface interactions.

To operate the CECS mobile app effectively, devices should be equipped with a 64-bit ARM processor (e.g., Snapdragon, MediaTek, or Apple A-series) and a minimum of 4GB RAM to handle multitasking and maintain smooth UI performance. Sufficient internal storage (at least 650MB) is also necessary for installation and caching purposes. A reliable camera is critical, requiring at least 8MP with autofocus and a scanning range of 10–20 cm to ensure accurate QR code recognition.

Connectivity is another vital aspect, with support for Wi-Fi or mobile data (4G/5G) needed to access real-time event data. The system also requires a touchscreen display with a minimum resolution of 720p for clear interface rendering. To guarantee compatibility and security, the app supports Android 8.0 (Oreo) or iOS 12 and above. Additional hardware considerations include a battery capacity of at least 2000mAh for extended event usage and ambient light sensors for optimal screen visibility in varying lighting conditions.

Together, these hardware requirements ensure that the CECS application performs reliably across different environments, supporting both student and admin users throughout the event lifecycle.

#### **1.3.1.4 Software Interfaces**

The Campus Event Check-in System (CECS) relies on specific software platforms to ensure its mobile application functions smoothly across a wide range of devices. These software interfaces are essential to support compatibility, stability, and access to core system features.

CECS is designed to run on two major mobile operating systems—Android and iOS. For Android devices, the app requires version 8.0 (Oreo) or later, ensuring that it runs on platforms with up-to-date security protocols and native feature support. On Apple devices, the system supports iOS version 12.0 and above, maintaining performance consistency and leveraging the iOS ecosystem's capabilities.

By supporting both Android and iOS operating systems, CECS ensures a broad reach and usability for the majority of students and administrators, allowing the system to function effectively on most modern smartphones and tablets.

### **1.3.1.5 Communication Interfaces**

The Campus Event Check-in System (CECS) is designed to support secure, reliable, and real-time interactions between the mobile application and various backend services. To ensure data integrity and user security, the system primarily relies on HTTPS with TLS 1.2 or higher for all communication between the client and server. Sensitive information is encrypted in transit, in line with university security policies.

The system integrates with Firebase services for authentication and real-time database interactions using Firebase SDK and REST APIs. It utilizes RESTful APIs over HTTPS to access and manage key functionalities such as event details, user records, and payment transactions. JSON is used as the standard data exchange format, allowing consistent and lightweight communication between the client and server.

For notification handling, CECS employs an in-app real-time notification mechanism triggered by backend events, using either periodic API polling or WebSocket-based messaging. This approach while still providing timely alerts to users for key actions such as registration confirmation, payment updates, or event reminders.

Additionally, the system includes support for OAuth 2.0 for potential future integrations with external services. Communication resilience is enhanced with retry and timeout mechanisms using exponential backoff strategies. To support QR-based check-in processes, the mobile app integrates SDKs or external libraries that handle QR code scanning smoothly. These combined communication protocols and methods ensure a secure and responsive user experience across all modules of the CECS platform.

### **1.3.1.6 Memory Constraints**

The memory requirements for the **Campus Event Check-in System (CECS)** are defined to ensure optimal performance and a smooth user experience on supported mobile devices.

**Table 1.3.1.2: Memory Constraints**

<b>Constraint ID</b>	<b>Description</b>	<b>Author</b>
REQ_MC_001	Mobile devices shall have a <b>minimum of 4GB RAM</b> to ensure smooth multitasking, camera usage, and QR scanning.	ERIC
REQ_MC_002	Devices shall maintain <b>at least 500MB</b> of free internal storage for app installation, caching, and data logging.	TIANYOU

### **1.3.1.7 Operations**

This section outlines the normal and special operations performed by users (students and admins) within the **Campus Event Check-in System (CECS)**. The system supports interactive user operations, automated background tasks, real-time data handling, and recovery features to ensure smooth and secure use on mobile platforms.

#### **1.3.1.7.1 Student Operations**

##### **1. User Authentication (Login)**

- Students authenticate securely via the mobile app.
- The system verifies credentials by communicating with the Authentication Server before granting access.

##### **2. View Event List**

- Students can browse a list of all upcoming and available campus events within the app.
- Events are displayed with essential details such as title, date, time, venue, and registration fee to help students decide which to join.

##### **3. Event Registration**

- Students register for selected events through the app and will proceed to make payment..
- Upon successful registration, the system sends confirmation notifications and reminders.

##### **4. Make Payment**

After registration, students pay using either:

- **Online Payment** via integrated Payment Server (e.g., FPX, TNG eWallet)
  - Payment success triggers digital receipt generation viewable within the app.
  - Students may request refunds; refund processing is subject to admin approval.
  - Upon successful payment, the system sends confirmation notifications and reminders.
- **On-site Payment** at the event counter
  - Payment success triggers digital receipt generation viewable within the app.
  - Students may request refunds; refund processing is subject to admin approval.
  - Upon successful payment, the system sends confirmation notifications and reminders.

5. **Check-in via QR Code**
  - During event entry, students scan the unique QR code generated by the admin.
  - They must also input Student ID, Name, and Ticket ID for identity verification.
  - The system validates ownership and marks attendance in real time.
6. **Rate and Provide Feedback**
  - After attending events, students are prompted to submit ratings and optional textual feedback.
  - This feedback is used to improve future events.

### **1.3.1.7.2 Admin Operations**

1. **Admin Authentication (Login)**
  - Admins log in securely through the Authentication Server to access administrative functions.
2. **Event Management**
  - Admins can create, update, and delete events.
3. **QR Code Generation**
  - The system auto-generates unique QR codes for each event to enable efficient check-ins.
4. **Real-Time Attendance Tracking**
  - Admins monitor check-ins live as students scan QR codes during event entry.
5. **View Ratings and Feedback**
  - Admins can review post-event feedback for event evaluation and continuous improvement.
6. **Report Generation**
  - Admins generate detailed reports on:
    - Attendance statistics
    - Payment summaries
    - User feedback analysis
7. **Handle Refund Requests**
  - Admins review student refund requests, approve or decline them, and communicate with the Payment Server to process eligible refunds.

### **1.3.1.7.3 Special Operations**

#### **1. Authentication Integration**

- All login attempts are processed through the centralized Authentication Server.
- Login failures or timeouts trigger automatic retries or user-facing error messages.

#### **2. Payment Integration**

- Payment transactions and refunds are securely handled by the external Payment Server.
- All transactions are logged and auditable by both users and administrators.

#### **3. Unattended Data Sync**

- In cases of network disruption, check-in and registration data are cached locally on the mobile device.
- Once connectivity is restored, cached data syncs automatically with the central database.

#### **4. Backup and Recovery**

- The system performs scheduled daily backups of critical data including events, attendance, payments, and refunds.
- Recovery procedures ensure data restoration in case of corruption or system failure.

### **1.3.1.8 Site Adaptation Requirements**

The Site Adaptation Requirements for Campus Event Check-in System (CECS) define specific conditions that need to be considered for successful deployment within the university environment. These include data dependencies, integration constraints, and site-specific configurations.

**Table 1.3.1.3: Site Adaptation Requirements**

<b>Constraint ID</b>	<b>Type</b>	<b>Description</b>	<b>Author</b>
REQ_SAR_001	Student Identity Integration	The system shall integrate with the university's student identification database for login and verification.	TIANYOU
REQ_SAR_002	Event Data Initialization	The system must support importing existing campus event data (titles, venues, fees) from external sources if available.	ERIC
REQ_SAR_003	Payment Gateway Mapping	Configuration with the university's selected payment gateways (e.g., FPX, TNG eWallet) must be completed prior to deployment.	YILING
REQ_SAR_004	Authentication Configuration	The Authentication Server must be accessible via university SSO or directory services.	KAISHEN
REQ_SAR_005	Notification Setup	Setup of in-app notification triggers and backend event polling mechanisms must be configured according to university policies.	YILING

### 1.3.1.9 Interface with Services

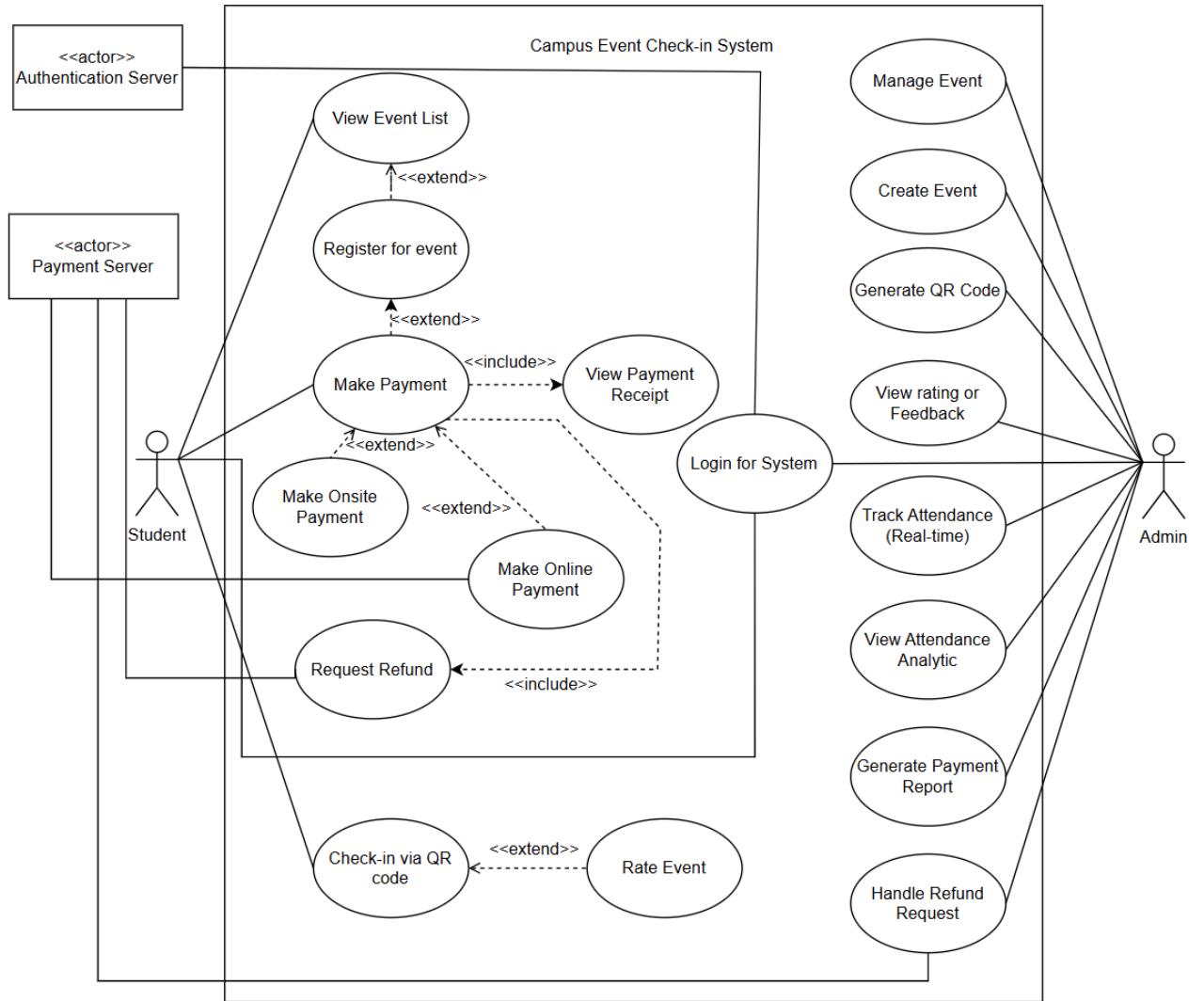
The interfaces with external services required by the Campus Event Check-in System (CECS) ensure secure login, real-time attendance tracking, cloud data storage, and reliable payment and notification operations.

**Table 1.3.1.4: Interface with Services**

Interface ID	Service Name	Description	Author
REQ_IS_001	Authentication Server	The system shall interface with the <b>Authentication Server</b> to validate student and admin login credentials using secure API communication.	ERIC
REQ_IS_002	Payment Gateway	The system shall integrate with a <b>university-approved payment gateway</b> (e.g., <b>FPX, TNG eWallet</b> ) to handle online and on-site event payments and refunds.	YILING
REQ_IS_003	In-App Notification System	The system shall use backend-triggered services or periodic polling mechanisms to deliver real-time in-app notifications and confirmations.	KAISHEN
REQ_IS_004	Cloud Database	The system shall use a <b>cloud-hosted database</b> (e.g., <b>Firebase Realtime Database, Google Firestore</b> ) for storing user, event, and attendance data.	TIANYOU
REQ_IS_005	Analytics & Reporting Tool	The system shall interface with a <b>reporting/analytics platform</b> (e.g., <b>Firebase Analytics, Admin Dashboard</b> ) to help admins monitor events, payments, and feedback.	ERIC

### 1.3.2 Product function

This is the overall use case diagram that shows all use cases for all actors.



**Figure 1.3.2.1: Use Case Diagram Of Campus Event Check-In System**

### 1.3.2.1 Student

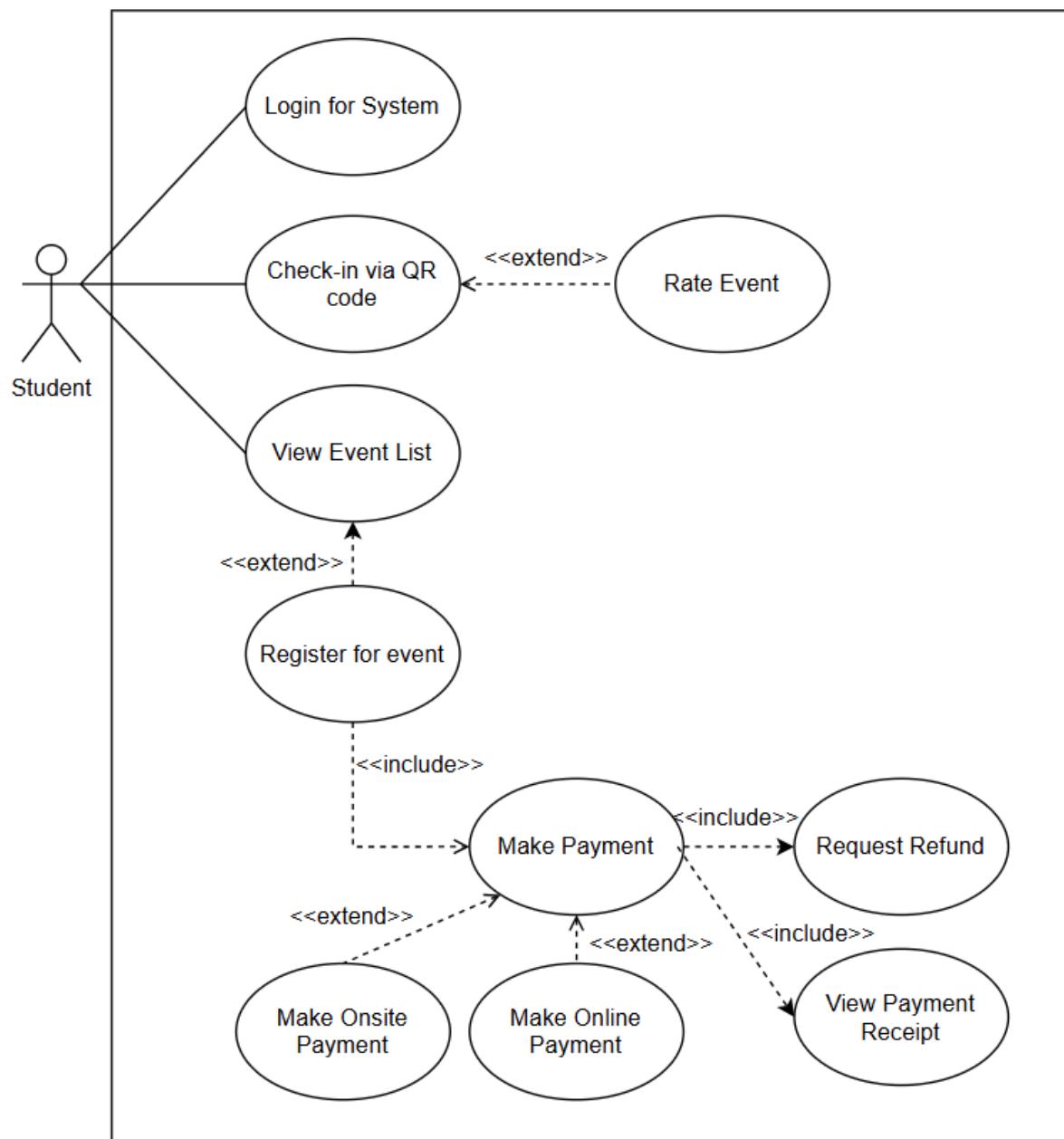


Figure 1.3.2.2: Use Case Diagram Of Student

**Table 1.3.2.1: Use Case Of Student**

Use Case ID	Use Case Name	Description	Author
REQ_UCS_001	Register For Event	Students sign up for an available campus event. This use case includes "Receive Notification" and includes "Make Payment"	KAISHEN
REQ_UCS_002	Rate Event	Students provide feedback or a rating for an event. This use case extends "Check-in via QR Code"	KAISHEN
REQ_UCS_003	Check-in Via QR Code	Students use a QR Code to mark their attendance at an event. This use case is extended by "Rate Event"	YILING
REQ_UCS_004	Make Payment	Student performs a payment action. This use case is included by "Register for event". As drawn, this use case extends "Make Onsite Payment" and also extends "Make Online Payment".	ERIC
REQ_UCS_005	View Payment Receipt	Students view a confirmation of their payment. This use case is included by "Make Payment"	YILING
REQ_UCS_006	Request Refund	Student initiates a request for a refund. As drawn, this use case is a base that can be included by "Make Payment"	TIANYOU
REQ_UCS_007	Make Online Payment	Students make a payment using an online method. As drawn, this use case is a base that can be extended by "Make Payment"	TIANYOU
REQ_UCS_008	Make Onsite Payment	Student makes a payment in person. As drawn, this use case is a base that can be extended by "Make Payment".	KAISHEN
REQ_UCS_009	Login For System	Student authenticates to access the system	ERIC
REQ_UCS_010	View Event List	Student view event that available. This use case is extends "Register for event"	ERIC

### 1.3.2.2 Admin



**Figure 1.3.2.3: Use Case Diagram Of Admin**

**Table 1.3.2.2: Use Case Of Admin**

Use Case ID	Use Case Name	Description	Author
REQ_UCA_001	Create Event	Creates a new event in the system	KAISHEN
REQ_UCA_002	Manage Event	Modifies or updates existing event details	YILING
REQ_UCA_003	Generate QR Code	Generates a QR Code for event check-in	TIANYOU
REQ_UCA_004	View Rating Or Feedback	Reviews student feedback on events	ERIC
REQ_UCA_005	Track Attendance	Monitors real-time attendance for an event	ERIC
REQ_UCA_006	View Attendance Analytics	Analyzes past attendance data and trends	KAISHEN
REQ_UCA_007	Login For System	User authenticates to access the system	YILING
REQ_UCA_008	Generate Payment Report	Creates reports on event payments	TIANYOU
REQ_UCA_009	Handle Refund Request	Processes student refund requests	KAISHEN

### 1.3.3 User Characteristics

This section describes the end users of the system as well as how well their basic understanding and knowledge of the system is and how it will affect the Campus Event Check-in System. The following table illustrates the expected level of knowledge for each role.

**Table 1.3.3.1 User Characteristics**

Role	Description	Expected Knowledge	Author
<b>Student</b>	University students using the mobile app for event participation.	Basic knowledge of using mobile apps: login, view event list, event registration, QR scanning, payment (online/onsite), receipt viewing, and refund requests.	KAISHEN
<b>Admin</b>	Staff member managing all system functionalities via the mobile app.	Comfortable managing events, handling real-time data, processing refunds, and navigating mobile admin tools.	YILING
<b>Authentication Server</b>	External service that validates credentials during mobile login.	Transparent to users; operates in the background.	TIANYOU
<b>Payment Server</b>	Backend service for secure in-app payments.	Interfaced through mobile UI; performs payment tasks securely and seamlessly.	ERIC

#### **1.3.4 Limitations**

The Campus Event Check-in System faces several limitations that could impact its operational reliability, integration capabilities, and overall user experience. These limitations stem from technical constraints, institutional policies, and reliance on existing infrastructure.

##### **1. Authentication Dependency**

- a. Mobile login relies on the Authentication Server. If the server is down, users can't access the app.

##### **2. Limited Payment Flexibility**

- a. Payments are restricted to integrated mobile gateways (e.g., card, digital wallets). Lack of offline or kiosk support may inconvenience some users.

##### **3. Refunds Are Not Automated**

- a. While students can request refunds via the app, Admins must manually review and approve each request, causing potential delays.

##### **4. Device Requirements**

- a. The mobile app assumes a working camera for QR scanning and sufficient performance for real-time features. Older devices may encounter issues.

##### **5. Network Dependency**

- a. Real-time features such as attendance tracking, QR check-in, and payment status updates require stable internet connectivity on mobile devices.

##### **6. No Browser/Desktop Access**

- a. Users without smartphones or who prefer desktop use will not be able to interact with the system. This may limit accessibility for a small portion of users.

##### **7. App Store Approval & Updates**

- a. Initial rollout and future updates depend on app store approvals (e.g., Apple App Store, Google Play), possibly delaying urgent fixes or enhancements.

### 1.3.5 Apportioning of requirements

The Apportioning of requirements of Campus Event Check-in System are described below:

**Table 1.3.5.1: Apportioning of requirements**

Module ID	Functionality	Software Elements	Notes / Version Info	Author
REQ_AR_001	User Authentication	Mobile App Front-End, Authentication API	Enables secure login using mobile credentials.	KAISHEN
REQ_AR_002	Event Registration	Mobile UI, Local Storage, Notification API	Mobile app handles full registration flow with confirmation messages.	YILING
REQ_AR_003	Payment Processing	Mobile Payment API, Payment Server	Supports secure in-app transactions for both onsite and online payment modes.	TIANYOU
REQ_AR_004	Receipt Viewing	In-App Receipt Viewer, Payment Record Storage	Users view receipts inside the mobile app.	ERIC
REQ_AR_005	Refund Request Handling	Mobile UI, Admin Workflow Tools	Request submitted in-app, approved manually by Admin.	ERIC
REQ_AR_006	QR Code Check-In	Camera Access, QR Code Scanner Module	Built-in scanner allows students to check in directly via app.	KAISHEN
REQ_AR_007	Attendance Tracking & Analytics	Mobile Admin Interface, Real-Time Sync	Admins monitor attendance live from mobile dashboard.	YILING

## 1.4 Definition

Below are terms, phrases and words used in the document and its related definition:

**Table 1.4.1: Definition**

Term	Definition
<b>Campus Event Check-in System</b>	A mobile app for students to register for events, check in using QR codes, and complete secure payments.
<b>Student</b>	A user accessing the app to register, pay for, and attend events.
<b>Admin</b>	A mobile user responsible for managing events, processing refunds, and monitoring attendance.
<b>Authentication Server</b>	A backend service validating user credentials during mobile login.
<b>Payment Server</b>	The system that handles all mobile-based transactions securely.
<b>QR Code</b>	A scan-enabled image generated by Admin and scanned via the mobile app to mark attendance.
<b>Payment Receipt</b>	A confirmation record generated in-app after payment is completed.
<b>Refund Request</b>	A feature in the mobile app allowing students to apply for a refund, to be reviewed by Admin.
<b>Event Feedback</b>	Input collected from attendees through the app post-event.
<b>Real-Time Attendance</b>	Mobile-based monitoring of attendees as they check in, updated instantly for Admins.
<b>Notification System</b>	Push or in-app notifications alerting users of registrations, events, or updates.

## 2 References

1. 1.IEEE. (2018). ISO/IEC/IEEE 29148:2018 Systems and software engineering—Life cycle processes—Requirements engineering.  
<https://www.iso.org/standard/72089.html>
2. Event management system for career fair - UTAR Institutional Repository. (n.d.). <http://eprints.utar.edu.my/6877/>
3. Unit Penerbitan UiTM Kelantan. (n.d.). The Development of Mobile Application for College Event Attendance System / Nur Irisya Anis Mohd Akin . . . [et al.] - UiTM Institutional Repository.  
<https://ir.uitm.edu.my/id/eprint/113228/>
4. Bhat, S., R, N., & S, P. (2023, July 8). Enhancing room security and automating class attendance using ID cards. arXiv.org.  
<https://arxiv.org/abs/2307.03926>
5. R Deepika, R Gayathri, T Saravanakumar, K Vigneshwaran, K Vignesh. “Android Application for Event Management System”. International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016: 328-330. Print.  
<https://edlib.net/2016/icsscet/ICSSCCET2016061.pdf>
6. The research on Sports Events Organization and Management Information System based on Process Aware. (2014, October 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/7064014>
7. Study on information management systems for emergency public health events in university. (2011, May 1). IEEE Conference Publication | IEEE Xplore. <https://ieeexplore.ieee.org/document/5882023>

## 3.1 Functions

This section outlines the key functional requirements the system needs to fulfill by the end of development. Each requirement is explained in detail and accompanied by a sequence diagram to provide a clear understanding of how it will work.

### 3.1.1 Sequence Diagram

#### 3.1.1.1 Student Login

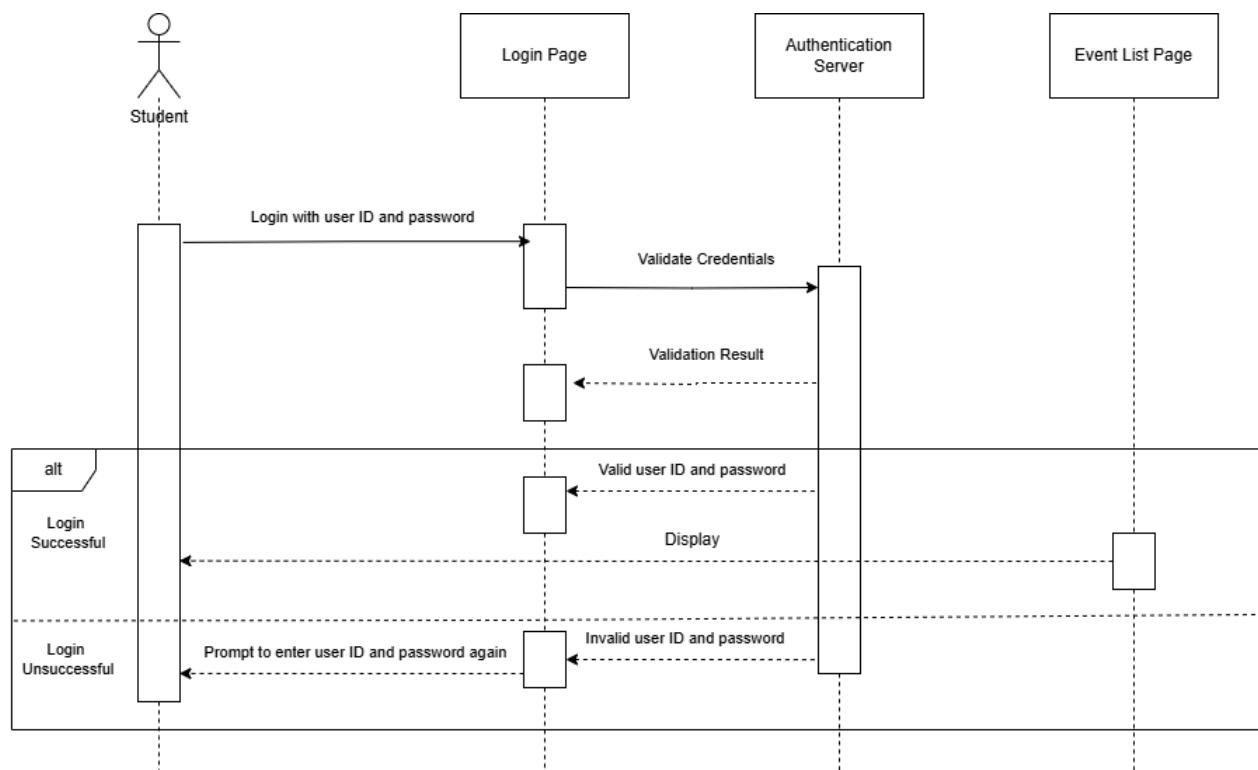


Figure 3.1.1.1: Student Login Sequence Diagram

<b>ID</b>	<b>REQ_SD_001</b>
<b>Feature</b>	Student Login
<b>Purpose</b>	To allow students to log in and authenticate themselves in the system.
<b>Actor</b>	Student
<b>Precondition</b>	The student must enter a valid user ID and password.
<b>Postcondition</b>	If the credentials are valid, the student will be logged in, and the system will display the event list page.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The system displays the login page.</li> <li>2. The Student enters and submits their login credentials (User ID and password).</li> <li>3. The system sends the credentials to the authentication server for validation.</li> <li>4. The authentication server validates the credentials and sends the validation result back to the system.</li> <li>5. If the credentials are valid, the system redirects the student to the event list page.</li> </ol>
<b>Alternate Scenarios</b>	<p><b>1. If the credentials are invalid:</b></p> <ul style="list-style-type: none"> <li>- The student is prompted to re-enter the user ID and password.</li> </ul>
<b>Author</b>	TIANYOU

### 3.1.1.2 Admin Login

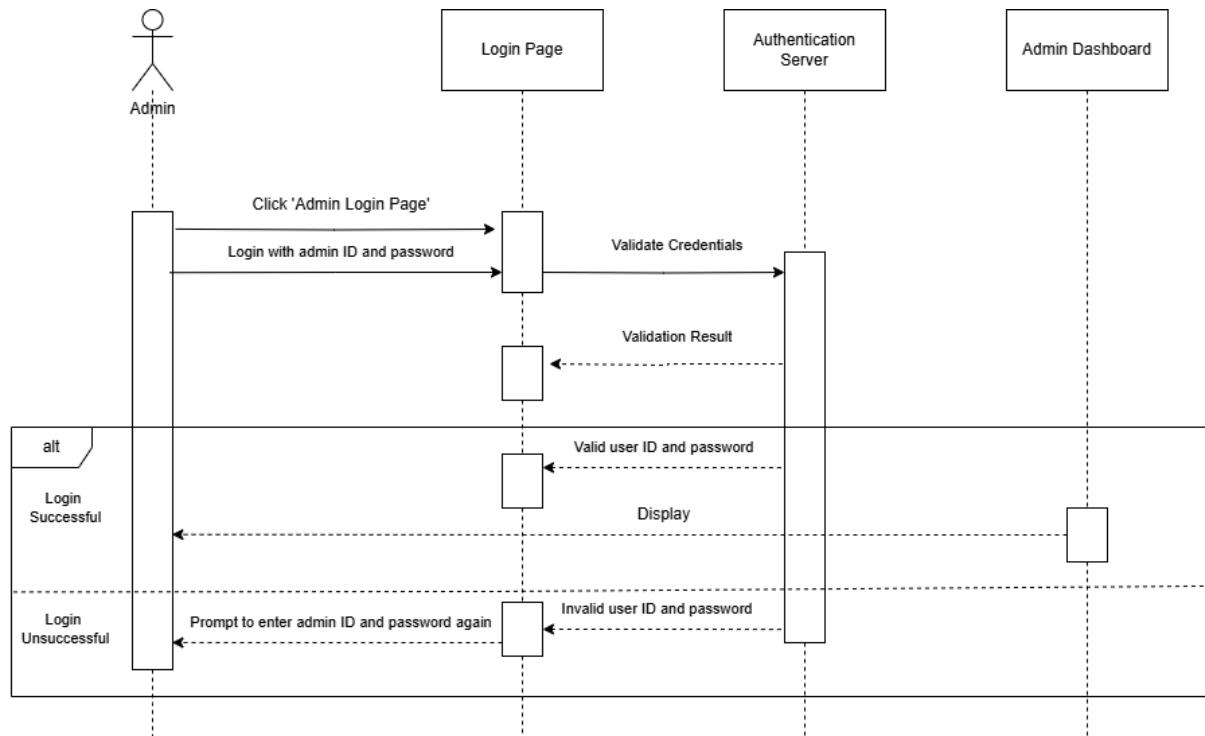


Figure 3.1.1.2: Admin Login Sequence Diagram

<b>ID</b>	REQ_SQ_002
<b>Feature</b>	Admin Login
<b>Purpose</b>	To allow administrators to log in and access the admin dashboard.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin must navigate to the admin login page and enter valid credentials.
<b>Postcondition</b>	If credentials are valid, the admin is logged in, and the admin dashboard is displayed.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin clicks on "Admin Login Page" to navigate to the login page.</li> <li>2. The system displays the admin login page.</li> <li>3. Admin enters and submits their login credentials (Admin ID and password).</li> <li>4. The system sends the credentials to the authentication server for validation.</li> <li>5. The authentication server validates the credentials and sends the validation result back to the system.</li> </ol>

	6. If the credentials are valid, the system redirects the admin to the admin dashboard.
<b>Alternate Scenarios</b>	1. If the credentials are invalid: - The admin is prompted to re-enter the Admin ID and password.
<b>Author</b>	TIANYOU

### 3.1.1.3 View Event List (Student)

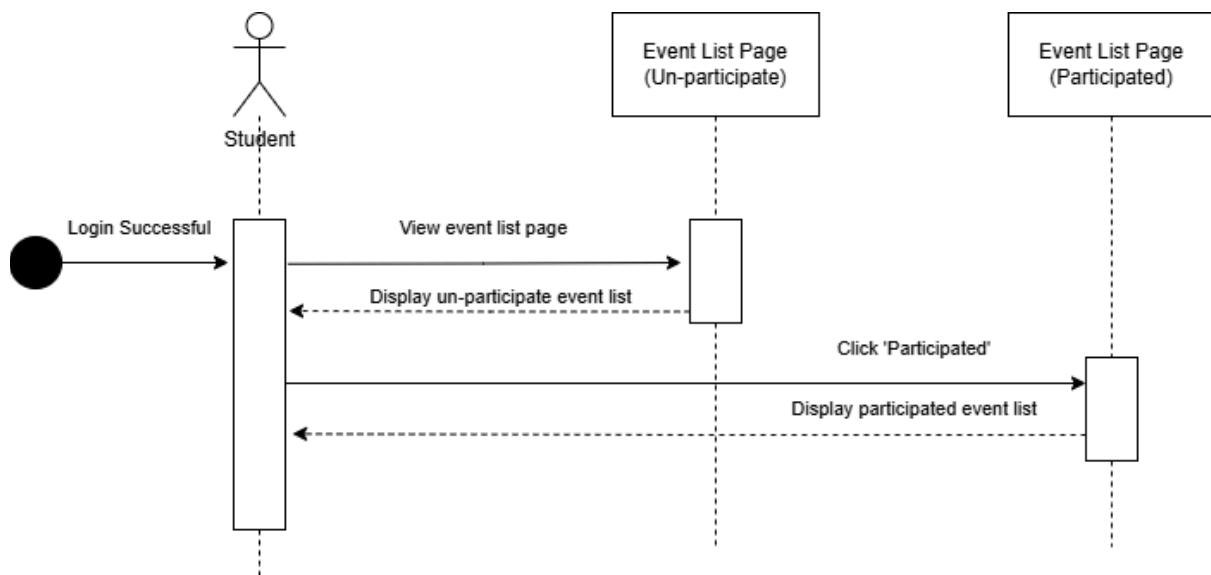


Figure 3.1.1.3: View Event List Sequence Diagram

<b>ID</b>	REQ_SD_003
<b>Feature</b>	View Event List
<b>Purpose</b>	To allow students to view both un-participated and participated events.
<b>Actor</b>	Student
<b>Precondition</b>	The student must be logged in successfully.
<b>Postcondition</b>	The student can view un-participated events and participated events separately.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. The student navigates to the event list page.</li> <li>2. The system retrieves and displays the list of un-participated events.</li> </ol>
<b>Alternate Scenario</b>	<ol style="list-style-type: none"> <li>1. If the student wants to view participating events, they click the "Participated" button.</li> <li>2. The system retrieves and displays the list of participating events.</li> </ol>
<b>Author</b>	ERIC

### 3.1.1.4 Register for Event (Student)

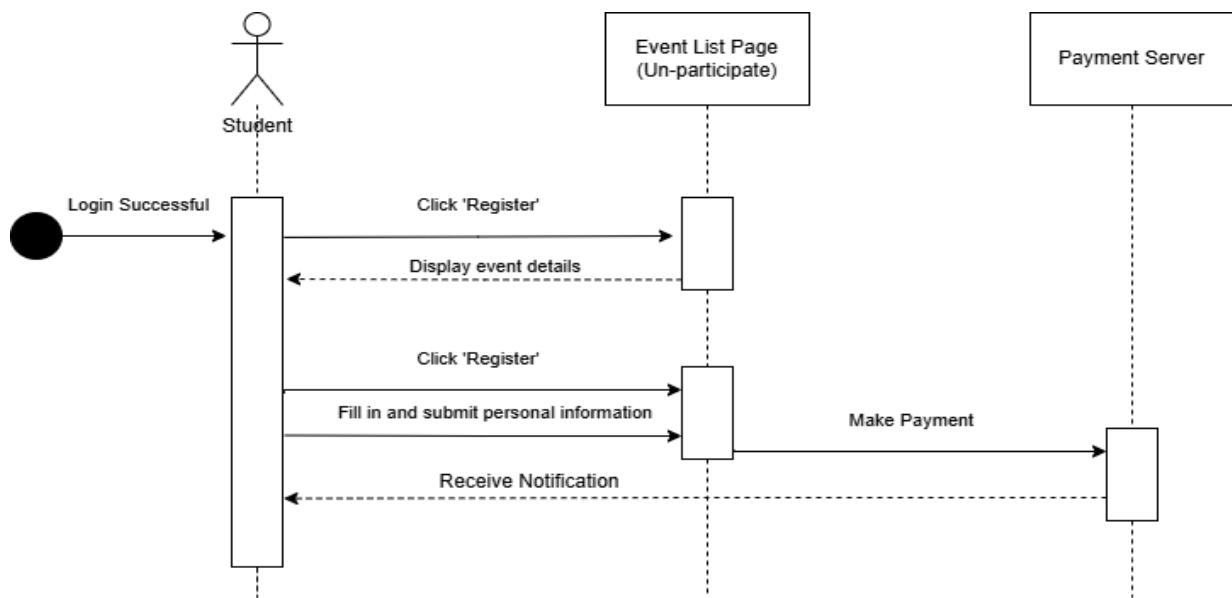


Figure 3.1.1.4: Register for Event Sequence Diagram

<b>ID</b>	REQ_SD_004
<b>Feature</b>	Event Registration
<b>Purpose</b>	To allow students to register for events and complete payment for registration.
<b>Actor</b>	Student
<b>Precondition</b>	The student has successfully logged in and is viewing the un-participated event list.
<b>Postcondition</b>	The student is redirected to the participating event list page with the newly registered event displayed.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>Students click on the 'Register' button in the un-participated event list page.</li> <li>The system displays the event details page.</li> <li>Students click on the 'Register' button on the event details page.</li> <li>The system displays a form for personal information.</li> <li>Student fills in and submits the personal information form.</li> <li>The system initiates the payment process.</li> <li>Student completes the payment process.</li> <li>The system displays a notification of successful registration.</li> <li>Student clicks 'Done' and is redirected to the participated event list page, where the registered event is displayed.</li> </ol>
<b>Alternate Flow</b>	<ol style="list-style-type: none"> <li>If the payment fails, the system displays an error message and allows the student to retry the payment.</li> </ol>
<b>Author</b>	ERIC

### 3.1.1.5 Make Payment (Student)

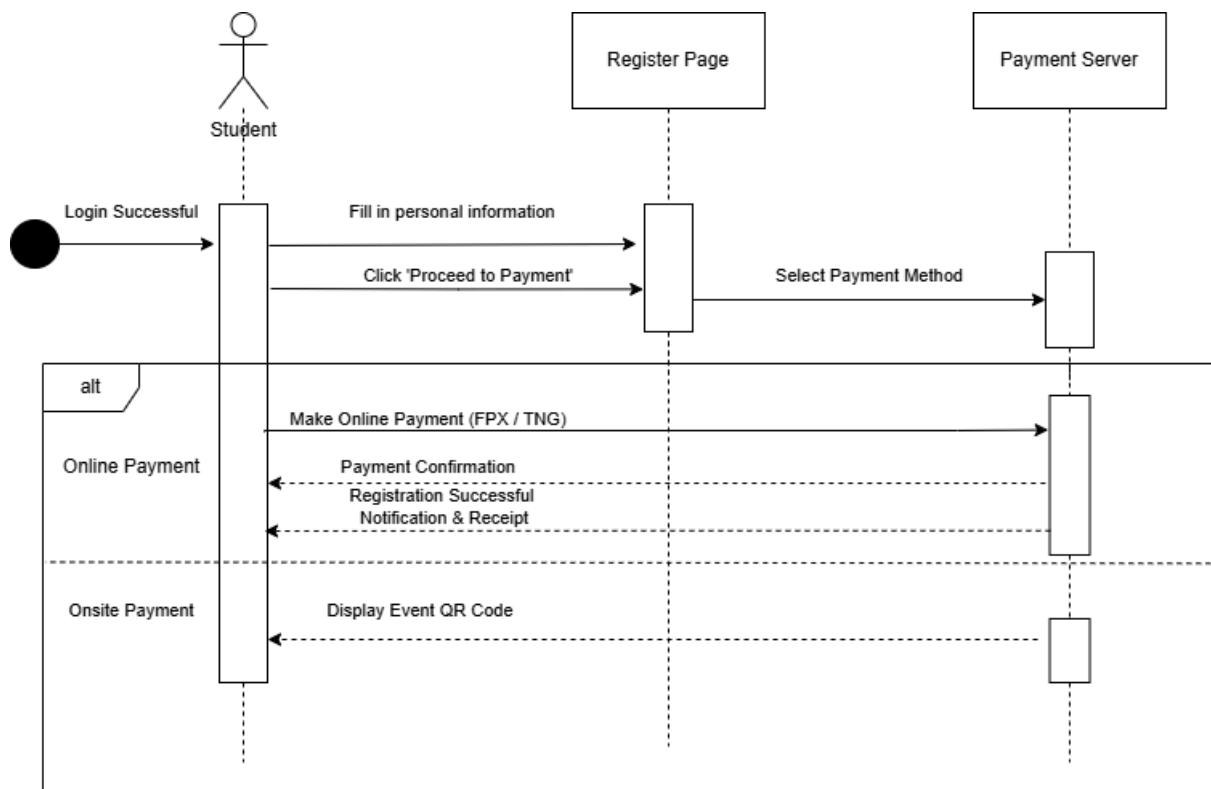


Figure 3.1.1.5: Make Payment Sequence Diagram

<b>ID</b>	REQ_SD_005
<b>Feature</b>	Make Payment
<b>Purpose</b>	To enable students to complete payment for event registration.
<b>Actor</b>	Student
<b>Precondition</b>	Student has filled in personal information and clicked "Proceed to Payment."
<b>Postcondition</b>	Payment is completed successfully, and the student is redirected to the participated event list page.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>Students fill in personal information on the registration page.</li> <li>Students click "Proceed to Payment" to select a payment method.</li> <li>Students choose between online payment or onsite payment.</li> <li>For online payment:             <ul style="list-style-type: none"> <li>Students complete the payment through the payment server (FPX/TNG). - System displays a confirmation message with a registration notification and receipt.</li> </ul> </li> <li>For onsite payment:             <ul style="list-style-type: none"> <li>System displays the event QR code for verification.</li> </ul> </li> <li>Students click "Done" and are redirected to the participated event list page with the registered event displayed.</li> </ol>

<b>Alternate Flow</b>	If payment fails, the system displays an error message and allows the students to retry payment.
<b>Author</b>	YILING

### 3.1.1.6 Make Onsite Payment (Student)

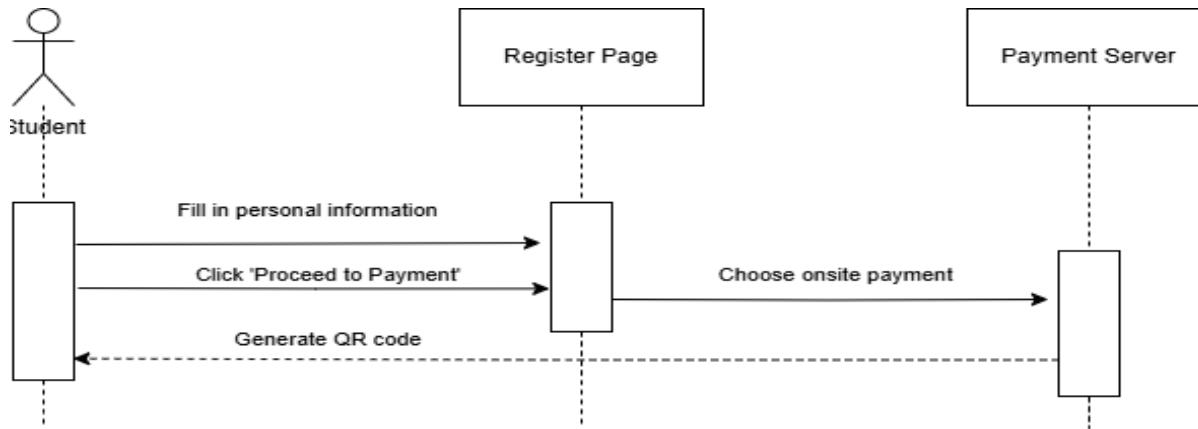


Figure 3.1.1.6: Make Onsite Payment Sequence Diagram

<b>ID</b>	REQ_SD_006
<b>Feature</b>	Make Onsite Payment
<b>Purpose</b>	To enable students to complete event registration by selecting onsite payment and receiving a QR code for verification.
<b>Actor</b>	Student
<b>Precondition</b>	User has filled in personal information and click "Proceed to Payment".
<b>Postcondition</b>	A QR code is generated for onsite verification, and the user is redirected to the participated event list page.
<b>Main Flow</b>	1. Students fill in personal information on the registration page. 2. Student clicks "Proceed to Payment". 3. System sends a request to the payment server for an onsite payment option. 4. System generates a QR code for the event. 5. Students receive the QR code and use it for verification. 6. Student clicks "Done" and is redirected to the participated event list page with the registered event displayed.
<b>Alternate Flow</b>	If QR code generation fails, the system displays an error message and prompts the student to retry.
<b>Author</b>	KAISHEN

### 3.1.1.7 Make Online Payment (Student)

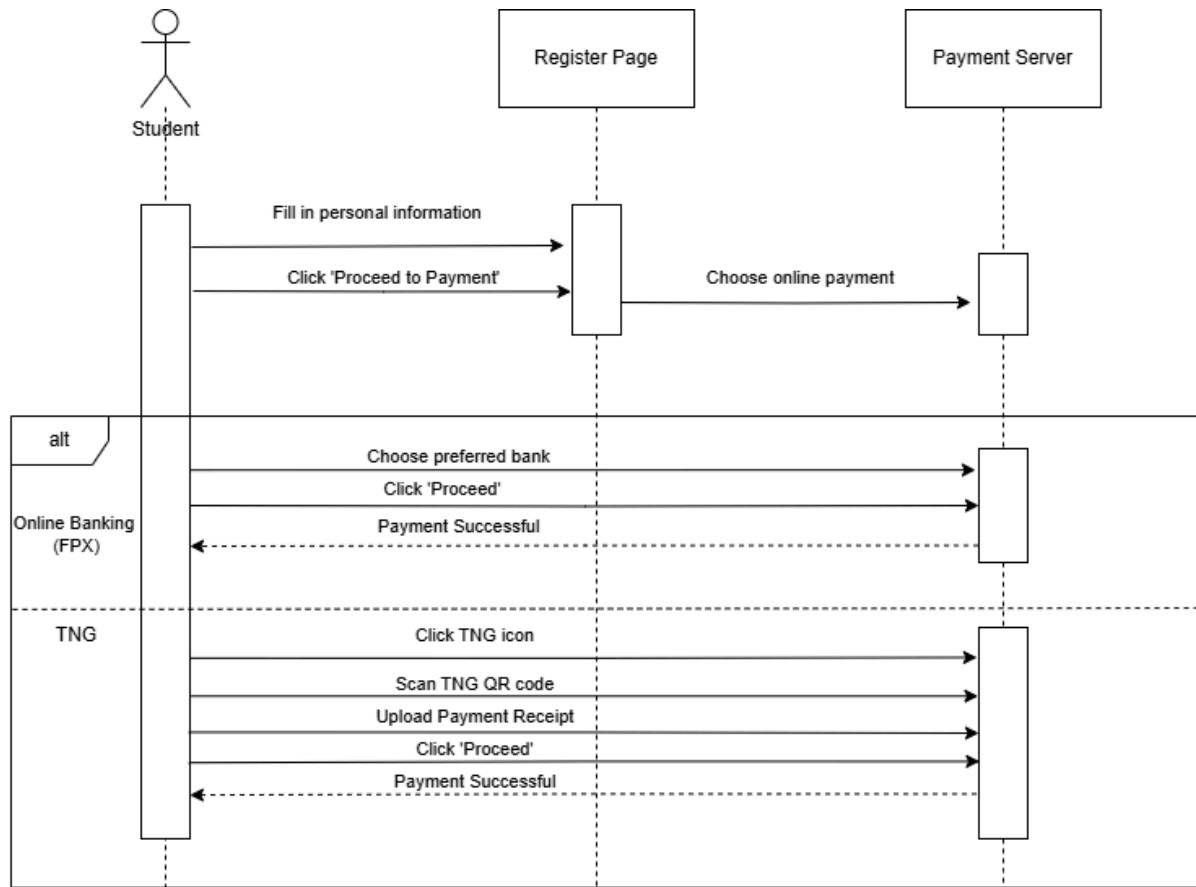
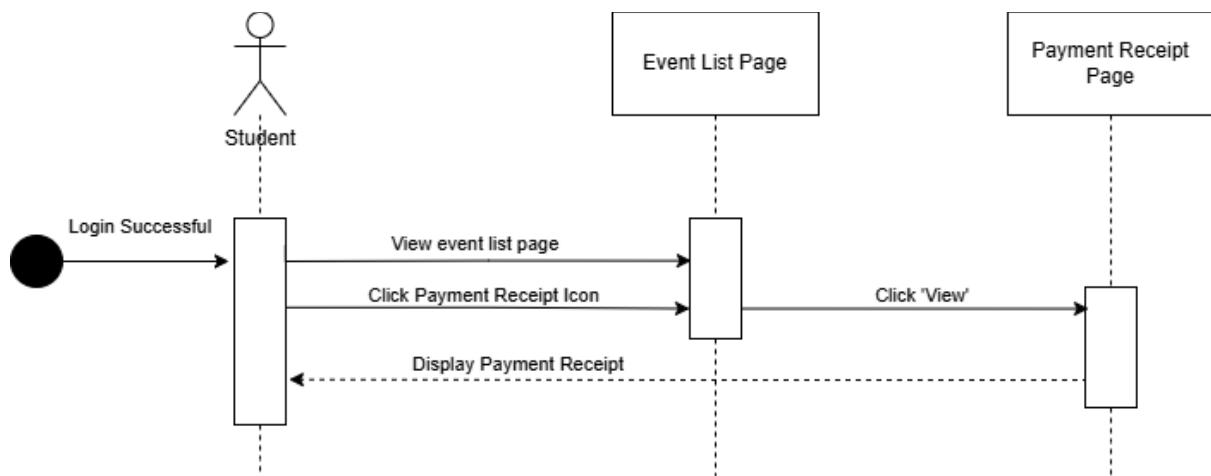


Figure 3.1.1.7: Make Online Payment Sequence Diagram

<b>ID</b>	REQ_SD_007
<b>Feature</b>	Make Online Payment
<b>Purpose</b>	To allow students to complete event registration by making an online payment using FPX or TNG.
<b>Actor</b>	Student
<b>Precondition</b>	User has filled in personal information and click "Proceed to Payment".
<b>Postcondition</b>	Payment is completed successfully and the user is redirected to the participated event list page.

<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Students fill in personal information on the registration page.</li> <li>2. Student clicks "Proceed to Payment".</li> <li>3. System sends a request to the payment server for online payment.</li> <li>4. Students choose between FPX or TNG.</li> </ol> <p><b>For FPX:</b></p> <ol style="list-style-type: none"> <li>a. Students choose their preferred bank.</li> <li>b. Student clicks "Proceed".</li> <li>c. System processes payment and returns a success message.</li> </ol> <p><b>For TNG:</b></p> <ol style="list-style-type: none"> <li>a. Students click the TNG icon.</li> <li>b. Student scans the TNG QR code.</li> <li>c. Student uploads payment receipt.</li> <li>d. Student clicks "Proceed".</li> <li>e. System processes payment and returns a success message.</li> </ol> <ol style="list-style-type: none"> <li>5. Students are redirected to the participated event list page.</li> </ol>
<b>Alternate Flow</b>	If payment fails, the system displays an error message and allows the student to retry.
<b>Author</b>	YILING

### 3.1.1.8 View Payment Receipt (Student)



**Figure 3.1.1.8: Make Online Payment Sequence Diagram**

<b>ID</b>	REQ_SD_008
<b>Feature</b>	View Payment Receipt
<b>Purpose</b>	To allow students to view the payment receipt for an event they have registered for.
<b>Actor</b>	Student
<b>Precondition</b>	The student has successfully logged in and is on the event list page.
<b>Postcondition</b>	The student is shown the payment receipt for a selected event.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Students successfully log into the system.</li> <li>2. The system redirects the student to the event list page.</li> <li>3. Students click the payment receipt icon next to a registered event.</li> <li>4. The system navigates to the payment receipt page.</li> <li>5. Student clicks 'View' to see the receipt.</li> <li>6. The system displays the payment receipt.</li> </ol>
<b>Alternate Flow</b>	None specified.
<b>Author</b>	KAISHEN

### 3.1.1.9 Request Refund (Student)

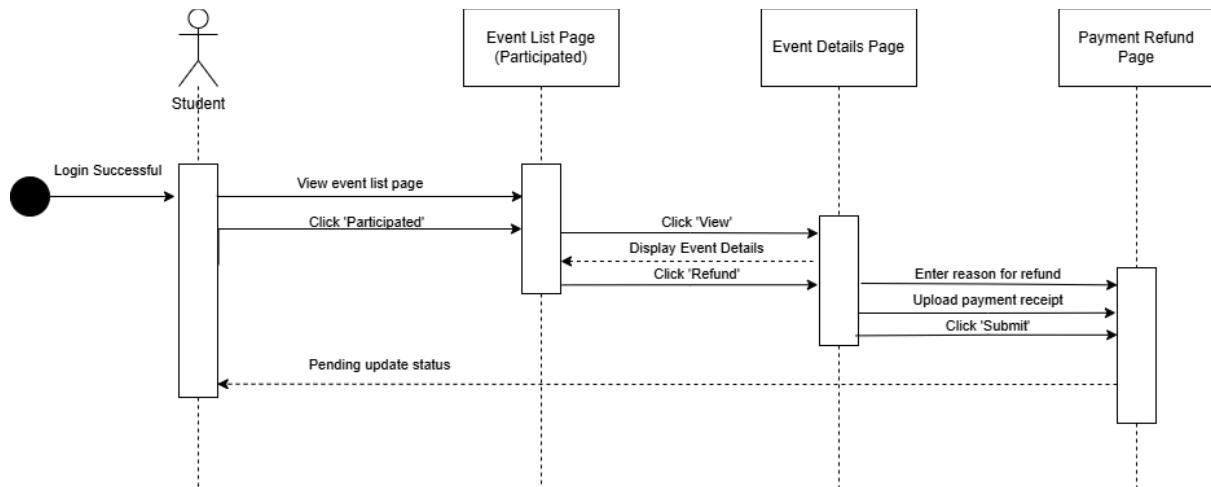


Figure 3.1.1.9: Request Refund Sequence Diagram

<b>ID</b>	REQ_SD_009
<b>Feature</b>	Request Refund
<b>Purpose</b>	To allow students to request a refund for a registered event by submitting a reason and payment receipt.
<b>Actor</b>	Student
<b>Precondition</b>	Student is logged in and has participated in an event.
<b>Postcondition</b>	Refund request is submitted and status is updated to "Pending".
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Student logs in successfully.</li> <li>2. Students view the participated event list page.</li> <li>3. Student clicks "Participated".</li> <li>4. System displays participated events.</li> <li>5. Students click "View" to open event details.</li> <li>6. System displays event details.</li> <li>7. Student clicks "Refund".</li> <li>8. System navigates to the refund page.</li> <li>9. Students enter the reason for refund.</li> <li>10. Student uploads the payment receipt.</li> <li>11. Students click "Submit".</li> <li>12. System updates refund status to "Pending".</li> </ol>
<b>Alternate Flow</b>	If the refund request submission fails (e.g., missing fields or upload error), the system displays an error message and prompts the student to retry.
<b>Author</b>	TIANYOU

### 3.1.1.10 Check-in Via QR Code (Student)

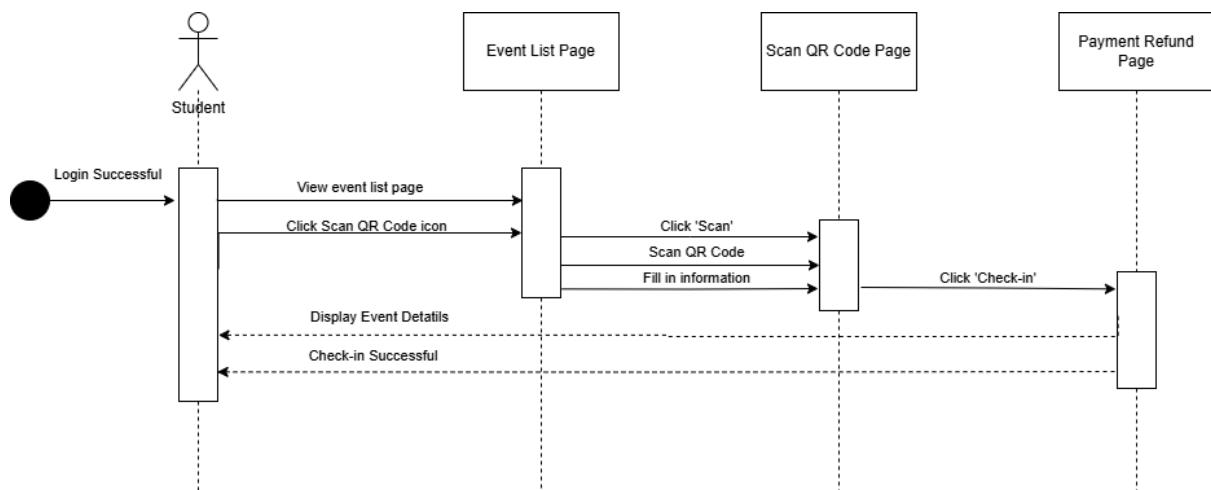
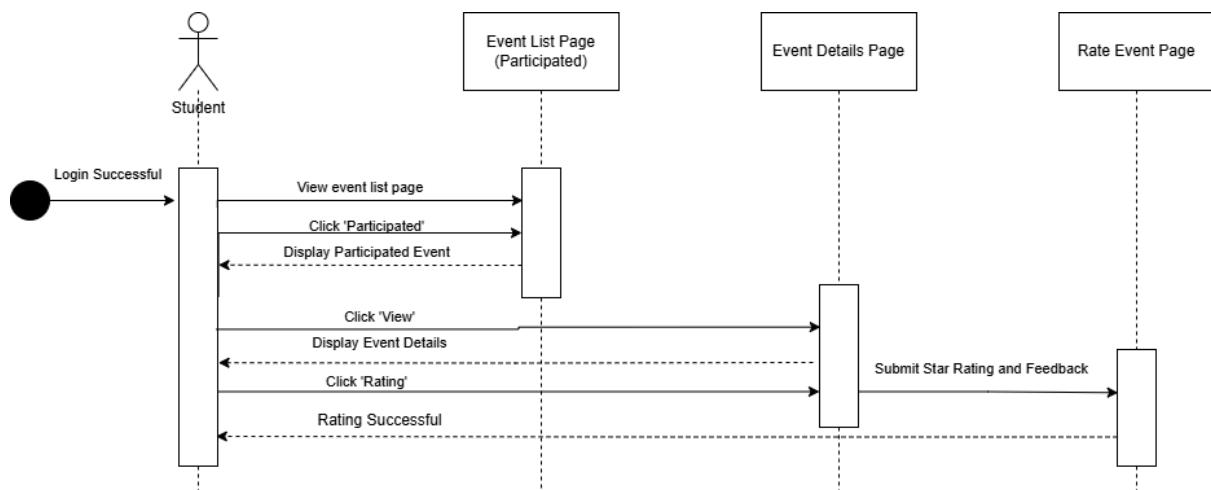


Figure 3.1.1.10: Check-in Via QR Code Sequence Diagram

<b>ID</b>	REQ_SD_010
<b>Feature</b>	Check-in via QR Code
<b>Purpose</b>	To allow students to check in to an event by scanning a QR code.
<b>Actor</b>	Student
<b>Precondition</b>	Student is logged in successfully.
<b>Postcondition</b>	Students are successfully checked in to the selected event.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Student logs in successfully.</li> <li>2. Students view the event list page.</li> <li>3. Students click the "Scan QR Code" icon.</li> <li>4. System navigates to the Scan QR Code page.</li> <li>5. Students click "Scan".</li> <li>6. Student scans the QR code.</li> <li>7. Students fill in required information.</li> <li>8. Students click "Check-in".</li> <li>9. System verifies and processes the check-in.</li> <li>10. System displays event details and confirms check-in success.</li> </ol>
<b>Alternate Flow</b>	If the QR code scan fails or required information is missing, the system displays an error message and prompts the student to retry.
<b>Author</b>	ERIC

### 3.1.1.11 Rate Event (Student)



**Figure 3.1.1.11: Rate Event Sequence Diagram**

ID	REQ_SD_011
Feature	Rate Event
Purpose	To allow students to provide a star rating and feedback for events they have participated in.
Actor	Student
Precondition	The student has successfully logged in and is viewing the list of participated events.
Postcondition	The student's rating and feedback are successfully submitted and stored.
Main Flow	<ol style="list-style-type: none"> <li>Students successfully log into the system.</li> <li>The system redirects the student to the event list page.</li> <li>Students click 'Participated' to view participated events.</li> <li>The system displays the participated events.</li> <li>Students click 'View' on an event.</li> <li>The system shows the event details page.</li> <li>Student clicks 'Rating'.</li> <li>The system opens the rate event page.</li> <li>Students submit a star rating and feedback.</li> <li>The system confirms that the rating was successful.</li> </ol>
Alternate Flow	None specified.
Author	KAISHEN

### 3.1.1.12 Manage Event (Admin)

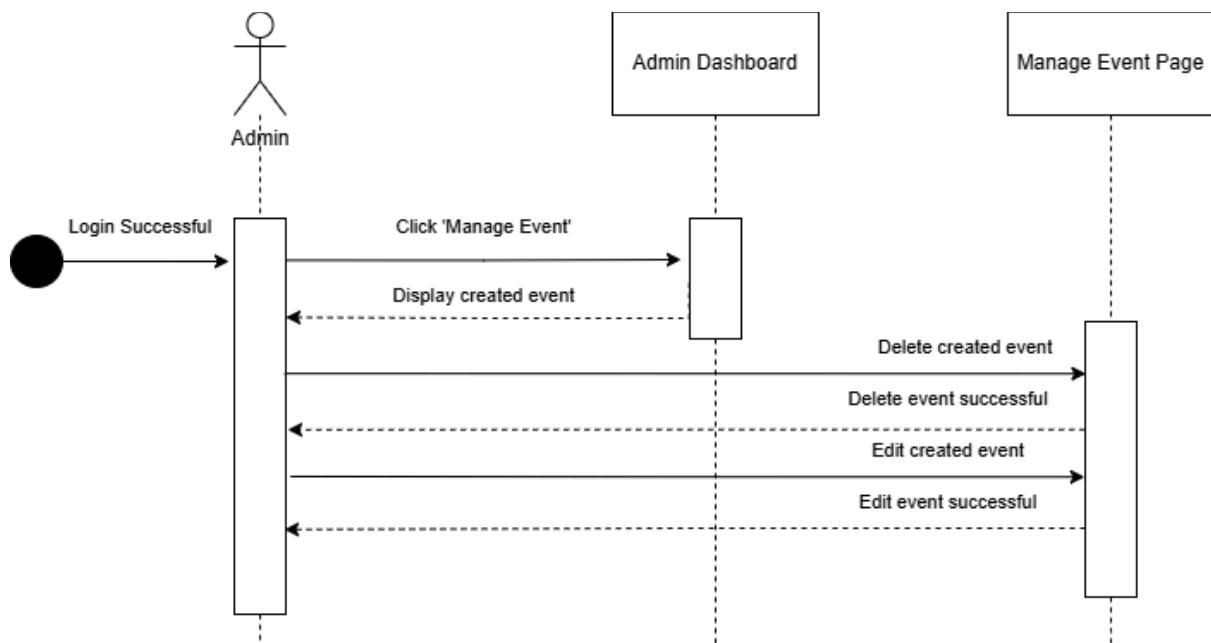


Figure 3.1.1.12: Manage Event Sequence Diagram

<b>ID</b>	REQ_SD_012
<b>Feature</b>	Manage Event
<b>Purpose</b>	To allow admin to manage events by editing or deleting created events.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in successfully.
<b>Postcondition</b>	Selected events are either updated or removed successfully.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "Manage Event".</li> <li>3. System displays created events.</li> <li>4. Admin selects to delete a created event.</li> <li>5. System confirms event deletion.</li> <li>6. Admin selects to edit a created event.</li> <li>7. System confirms event update.</li> </ol>
<b>Alternate Flow</b>	If edit or delete operation fails (e.g., server error, validation fail), the system displays an error message and prompts retry.
<b>Author</b>	YILING

### 3.1.1.13 Create Event (Admin)

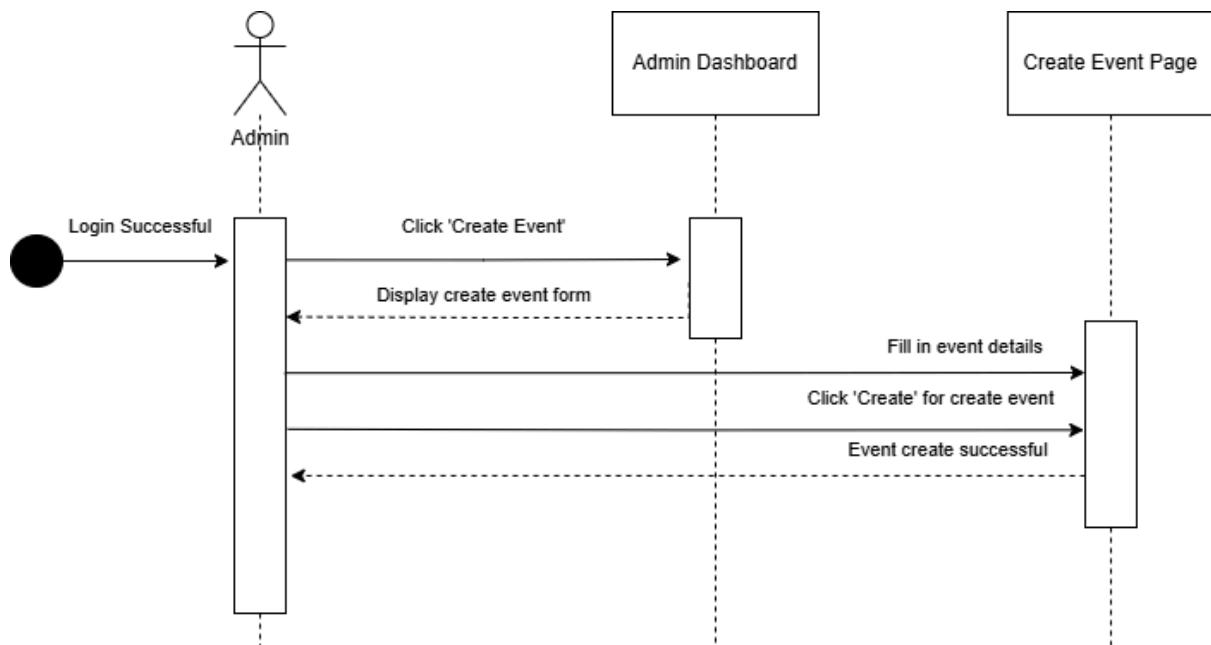


Figure 3.1.1.13: Create Event Sequence Diagram

<b>ID</b>	REQ_SD_013
<b>Feature</b>	Create Event
<b>Purpose</b>	To allow admin to create a new event by filling in event details.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in successfully.
<b>Postcondition</b>	A new event is created and stored in the system.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "Create Event".</li> <li>3. System displays the create event form.</li> <li>4. Admin fills in event details.</li> <li>5. Admin clicks "Create" to submit the form.</li> <li>6. System confirms the event has been successfully created.</li> </ol>
<b>Alternate Flow</b>	If event creation fails (e.g., missing details, invalid input), the system shows an error message and requests correction.
<b>Author</b>	ERIC

### 3.1.1.14 Generate QR Code (Admin)

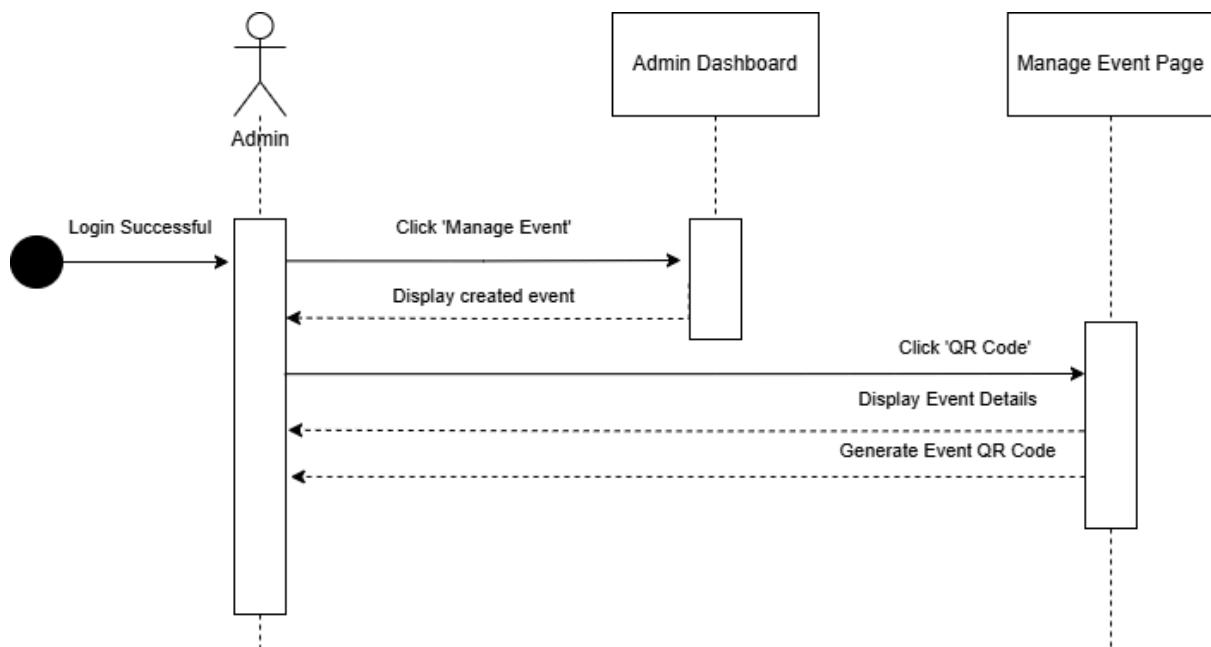


Figure 3.1.1.14: Create Event Sequence Diagram

<b>ID</b>	REQ_SD_014
<b>Feature</b>	Generate QR Code
<b>Purpose</b>	To allow admin to generate a QR code for an existing event.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in and has created at least one event.
<b>Postcondition</b>	A QR code for the selected event is generated and ready for use.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "Manage Event".</li> <li>3. System displays the list of created events.</li> <li>4. Admin clicks "QR Code" on the desired event.</li> <li>5. System displays event details.</li> <li>6. System generates a QR code for the event.</li> </ol>
<b>Alternate Flow</b>	If the event QR code generation fails (e.g., system error), an error message is displayed and the admin is prompted to retry.
<b>Author</b>	TIANYOU

### 3.1.1.15 View Rating or Feedback (Admin)

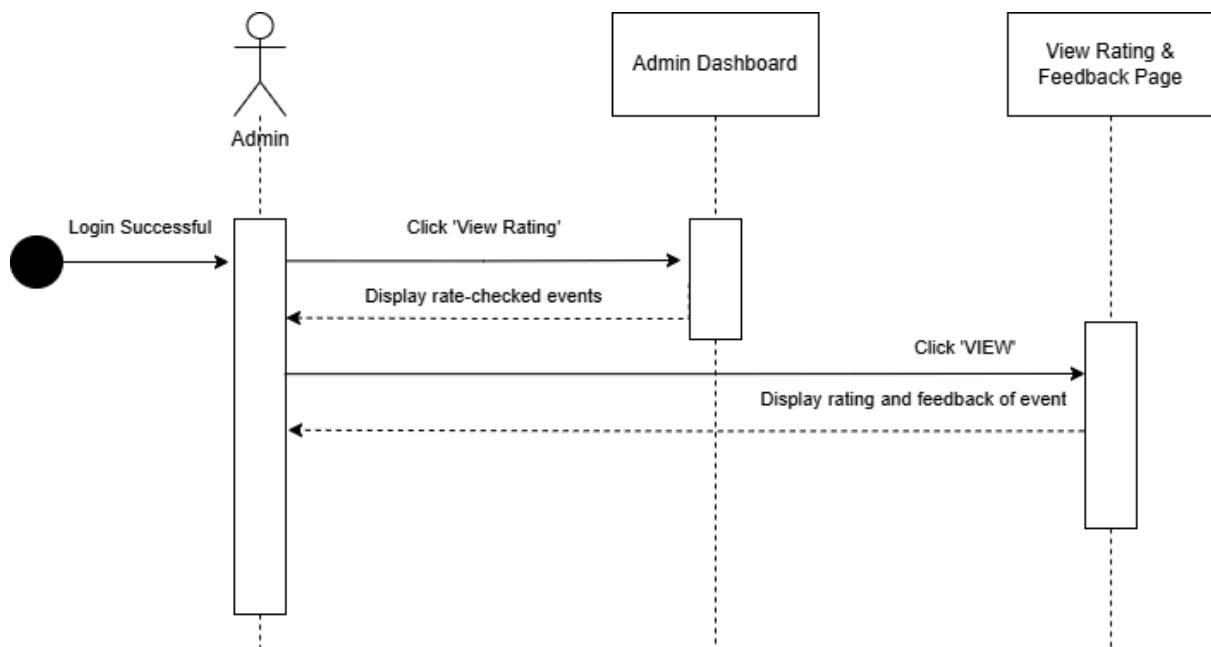


Figure 3.1.1.15: View Rating or Feedback Sequence Diagram

<b>ID</b>	REQ_SD_015
<b>Feature</b>	View Rating or Feedback
<b>Purpose</b>	To allow admin to view ratings and feedback provided by participants for events.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in and has access to rated events.
<b>Postcondition</b>	Admin successfully views the rating and feedback of selected events.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "View Rating".</li> <li>3. System displays the list of events that have received ratings.</li> <li>4. Admin clicks "VIEW" for a specific event.</li> <li>5. System displays the ratings and feedback for that event.</li> </ol>
<b>Alternate Flow</b>	If no rating or feedback is available, the system displays a message such as "No feedback available for this event."
<b>Author</b>	KAISHEN

### 3.1.1.16 Track Attendance (Admin)

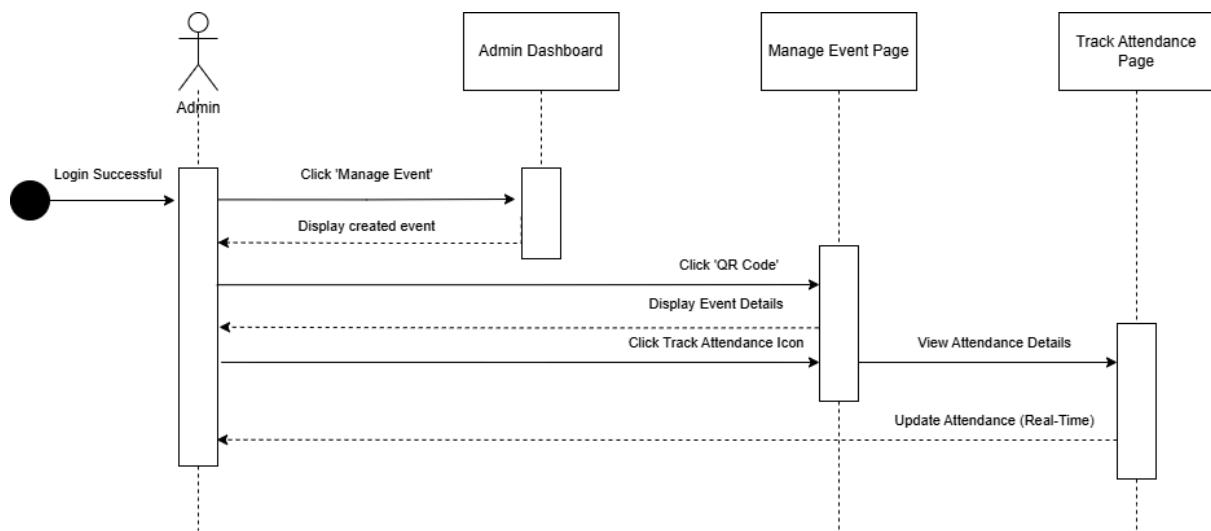


Figure 3.1.1.16: Track Attendance Sequence Diagram

<b>ID</b>	REQ_SD_016
<b>Feature</b>	Track Attendance
<b>Purpose</b>	To allow admin to track and update the attendance of event participants in real-time.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in and has access to the created events.
<b>Postcondition</b>	Attendance is viewed and updated in real-time.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "Manage Event".</li> <li>3. System displays the list of created events.</li> <li>4. Admin clicks "QR Code" for the selected event.</li> <li>5. System displays event details.</li> <li>6. Admin clicks the "Track Attendance" icon.</li> <li>7. System displays attendance details.</li> <li>8. Attendance is updated in real-time.</li> </ol>
<b>Alternate Flow</b>	If attendance tracking fails (e.g., data loading issue), the system displays an error message and suggests retry.
<b>Author</b>	YILING

### 3.1.1.17 View Attendance Analytic (Admin)

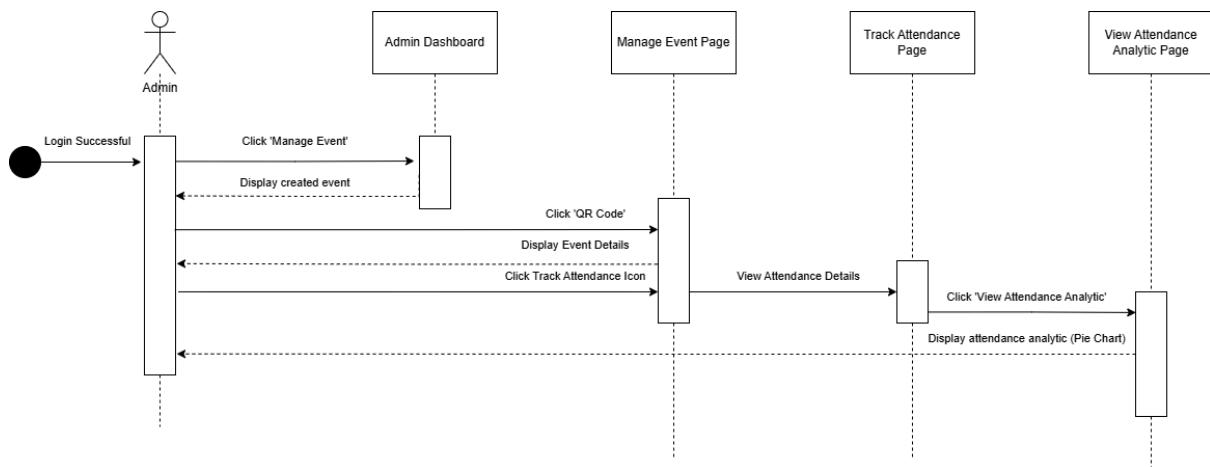


Figure 3.1.1.17: View Attendance Analytic Sequence Diagram

ID	REQ_SD_017
Feature	View Attendance Analytic
Purpose	To allow admin to view visual attendance analysis (e.g., pie chart) of an event.
Actor	Admin
Precondition	Admin is logged in and has access to an event's attendance data.
Postcondition	Attendance analytics are successfully displayed in a visual format.
Main Flow	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "Manage Event".</li> <li>3. System displays the list of created events.</li> <li>4. Admin clicks "QR Code" for a specific event.</li> <li>5. System displays event details.</li> <li>6. Admin clicks the "Track Attendance" icon.</li> <li>7. System shows attendance details.</li> <li>8. Admin clicks "View Attendance Analytic".</li> <li>9. System displays attendance analytics (e.g., pie chart).</li> </ol>
Alternate Flow	If analytics fail to load or no data is available, the system displays an appropriate message such as "No attendance data available."
Author	YILING

### 3.1.1.18 Generate Payment Report (Admin)

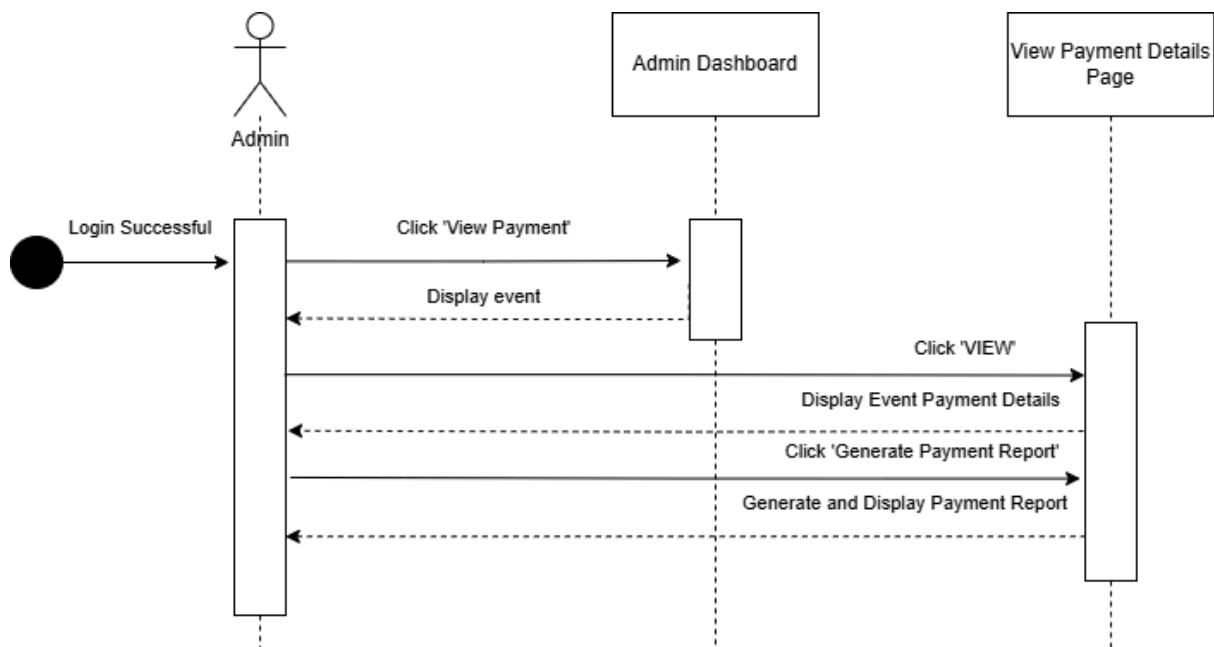


Figure 3.1.1.18: Generate Payment Report Sequence Diagram

<b>ID</b>	REQ_SD_018
<b>Feature</b>	Generate Payment Report
<b>Purpose</b>	To allow admin to generate and view detailed payment reports for events.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in and has access to event payment details.
<b>Postcondition</b>	A payment report is successfully generated and displayed.
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "View Payment".</li> <li>3. System displays the list of events.</li> <li>4. Admin clicks "VIEW" on a specific event.</li> <li>5. System displays event payment details.</li> <li>6. Admin clicks "Generate Payment Report".</li> <li>7. System generates and displays the payment report.</li> </ol>
<b>Alternate Flow</b>	If the report generation fails (e.g., missing data, system error), the system shows an error message and prompts the admin to try again.
<b>Author</b>	ERIC

### 3.1.1.19 Handle Request Refund (Admin)

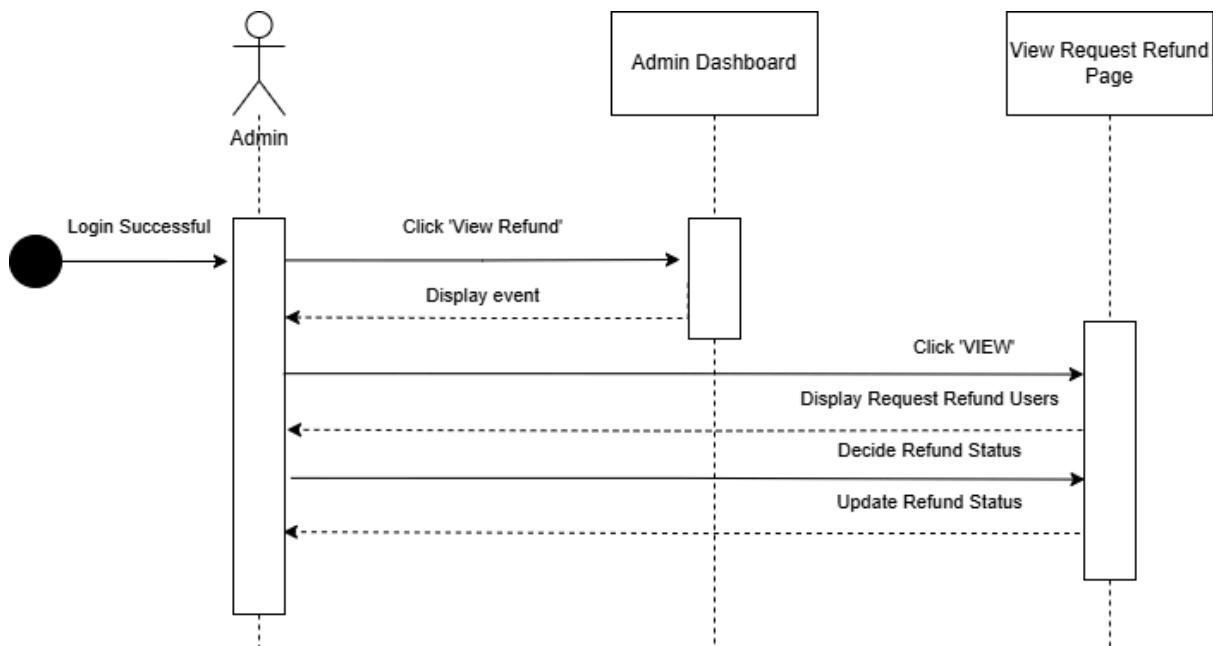


Figure 3.1.1.19: Handle Request Refund Sequence Diagram

<b>ID</b>	REQ_SD_019
<b>Feature</b>	Handle Request Refund
<b>Purpose</b>	To allow admin to view, review, and update the status of refund requests.
<b>Actor</b>	Admin
<b>Precondition</b>	Admin is logged in and users have submitted refund requests.
<b>Postcondition</b>	Refund status is updated (e.g., approved or rejected).
<b>Main Flow</b>	<ol style="list-style-type: none"> <li>1. Admin logs in successfully.</li> <li>2. Admin clicks "View Refund".</li> <li>3. System displays a list of events.</li> <li>4. Admin clicks "VIEW" on a selected event.</li> <li>5. System displays the list of users requesting refunds.</li> <li>6. Admin reviews and decides refund status.</li> <li>7. System updates and confirms the refund status.</li> </ol>
<b>Alternate Flow</b>	If the refund update fails (e.g., server error, missing info), the system shows an error message and prompts retry.
<b>Author</b>	TIANYOU

## 3.2 Performance Requirements

This section defines the quality requirements that the **Campus Event Check-in System (CECS)** must achieve in the final product. These requirements outline the desired performance and characteristics that ensure a good user experience, system stability, and high efficiency. The table below describes the performance requirements in detail to guide the system's implementation and ensure it meets stakeholder expectations.

**Table 3.2: Performance Requirements**

Requirement ID	Description	Priority	Author
REQ_P_001	The system shall respond to login and event-related user requests within 1 to 3 seconds.	High	ERIC
REQ_P_002	The platform shall support up to 5,000 concurrent users without performance degradation.	High	TIANYOU
REQ_P_003	The system shall ensure 99.9% uptime, excluding scheduled maintenance.	High	KAOSHEN
REQ_P_004	The platform shall process event registration and check-in requests within 2 seconds.	High	YILING
REQ_P_005	The system shall handle up to 100,000 transactions per day without performance loss.	High	YILING
REQ_P_006	The platform shall load event lists and user profiles within 1 second.	Medium	KAISHEN
REQ_P_007	The system shall generate QR codes for events within 1 second of request.	High	KAISHEN
REQ_P_008	The platform shall maintain a latency of less than 100ms for all database transactions.	Medium	TIANYOU
REQ_P_009	The system shall scale horizontally to accommodate increases in user load and transaction volume.	High	TIANYOU
REQ_P_010	The platform shall ensure no more than one crash per 10,000 sessions.	Medium	ERIC
REQ_P_011	The system shall notify users of successful event registration or payment within 1 second of transaction.	High	ERIC
REQ_P_012	The platform shall ensure secure data	High	YILING

	transmission with an encryption/decryption delay of less than 50ms.		
REQ_P_013	The system shall perform backups every hour with a maximum data loss window of 30 minutes.	High	ERIC
REQ_P_014	The platform shall send real-time event updates and notifications with a delay of no more than 1 second.	High	TIANYOU
REQ_P_015	The system shall use load balancing to efficiently distribute requests across servers.	High	KAISHEN

### 3.3 Usability Requirements

In this section, we detail the usability requirements the **Campus Event Check-in System (CECS)** aims to achieve by the project's completion. These requirements are designed to enhance the user experience and overall satisfaction by focusing on intuitive interfaces, accessibility, and user-friendly functionalities. The table below outlines the usability requirements in detail.

**Table 3.3: Usability Requirements**

Requirement ID	Description	Priority	Author
REQ_U_001	The platform shall provide a simple and intuitive login process for students and admins.	High	ERIC
REQ_U_002	The interface shall display event lists and details in an organized and user-friendly manner.	High	TIANYOU
REQ_U_003	The platform shall provide step-by-step guidance for users registering for events or making payments.	Medium	KAOSHEN
REQ_U_004	The system shall allow users to access their payment receipts and event history with minimal effort.	Medium	YILING
REQ_U_005	The platform shall feature a clean and minimalistic design to reduce cognitive load for users.	High	YILING
REQ_U_006	The system shall include a mobile-friendly interface to ensure usability on smartphones and tablets.	High	KAISHEN
REQ_U_007	The platform shall support clear instructions and error messages to assist users in case of issues.	High	KAISHEN
REQ_U_008	Users shall have access to a help section or tutorial on how to navigate the platform.	Medium	TIANYOU
REQ_U_009	The platform shall allow users to customize notification settings for event updates and reminders.	Medium	TIANYOU
REQ_U_010	The system shall provide accessibility features to accommodate users with disabilities.	High	ERIC
REQ_U_011	The platform shall allow users to provide	Medium	ERIC

	event ratings and feedback via an intuitive interface.		
REQ_U_012	The platform shall ensure secure and straightforward check-in for events using QR code scanning.	High	YILING
REQ_U_013	The system shall include a feedback mechanism for users to report issues or suggest improvements.	Medium	ERIC
REQ_U_014	The platform shall display real-time attendance status for admins in an easy-to-read format.	High	TIANYOU
REQ_U_015	The system shall provide periodic updates and ensure compatibility with the latest devices and browsers.	Medium	KAISHEN

## 3.4 Interface Requirements

### 3.4.1 External Interfaces

#### 3.4.1.1 SR01 Login Page

**Table 3.4.1.1: Login Button**

Requirement ID	REQ_SR0101	Version	1.0
Item	Login Button		
Item Description	A button with the word “Login”		
Item Purpose	To submit login credentials to be authenticated		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0102, REQ_SR0103 (Fill in both fields to login)		
Author	KAISHEN		

**Table 3.4.1.2: User ID Field**

Requirement ID	REQ_SR0102	Version	1.0
Item	User ID Field (Input)		
Item Description	A text field with the label “User ID”		
Item Purpose	To enter user ID to be submitted		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	KAISHEN		

**Table 3.4.1.3: Password Field**

Requirement ID	REQ_SR0103	Version	1.0
Item	Password Field (Input)		
Item Description	A text field with the label “Password”		
Item Purpose	To enter password to be submitted		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	KAISHEN		

**Table 3.4.1.4: Admin Login Page Click**

Requirement ID	REQ_SR0104	Version	1.0
Item	Admin Login Page Click (Input)		
Item Description	A button with the word “Admin Login Page”		
Item Purpose	Click to login for the admin page		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0102, REQ_SR0103(Fill in both fields to login)		
Author	KAISHEN		

### 3.4.1.2 SR02 Event List

**Table 3.4.2.1: Unparticipated Button**

Requirement ID	REQ_SR0201	Version	1.0
Item	Unparticipated Button (input)		
Item Description	A button with the word “Unparticipated”		
Item Purpose	To view Event that user unparticipated		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.2.2: Register Button**

Requirement ID	REQ_SR0202	Version	1.0
Item	Register Button (Input)		
Item Description	A button with the word “Register”		
Item Purpose	Resgister a event		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.2.3: Name Field**

Requirement ID	REQ_SR0203	Version	1.0
Item	Name Field (Input)		
Item Description	A text field with the label "Full Name"		
Item Purpose	To enter user's name to be register		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.2.4: Age Field**

Requirement ID	REQ_SR0204	Version	1.0
Item	Age Field (Input)		
Item Description	A text field with the label "Age"		
Item Purpose	To enter user's age to be register		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.2.5: Phone Number Field**

Requirement ID	REQ_SR0205	Version	1.0
Item	Phone Number Field (Input)		
Item Description	A text field with the label "Phone Number"		
Item Purpose	To enter user's phone number to be register		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.2.6: Email Field**

Requirement ID	REQ_SR0206	Version	1.0
Item	Email Field (Input)		
Item Description	A text field with the label "Email"		
Item Purpose	To enter user's email to be register		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.2.7: Category Field**

Requirement ID	REQ_SR0207	Version	1.0
Item	Category Field (Input)		
Item Description	A text field with the label “Enter Category”		
Item Purpose	To enter category to be register		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.2.8: Proceed To Payment Button**

Requirement ID	REQ_SR0208	Version	1.0
Item	Proceed To Payment Button (Input)		
Item Description	A button with the word “Proceed To Payment”		
Item Purpose	To proceed the payment for an event		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0203, REQ_SR0204, REQ_SR0205, REQ_SR0206, REQ_SR0207 (All fields must be field in and proceed to payment)		
Author	YILING		

**Table 3.4.2.9: Cancel Button**

Requirement ID	REQ_SR0209	Version	1.0
Item	cancel Button (Input)		
Item Description	A button with the word “Cancel”		
Item Purpose	Cancel for register an event		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.2.10: Participated Button**

Requirement ID	REQ_SR0210	Version	1.0
Item	Participated Button (Input)		
Item Description	A button with the word “Participated”		
Item Purpose	To view the event that user register		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.2.11: View Button**

Requirement ID	REQ_SR0211	Version	1.0
Item	View Button (Input)		
Item Description	A button with the word “View”		
Item Purpose	To view the event that user register		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

### 3.4.1.3 SR03 Event Payment

**Table 3.4.3.1: Online Payment Button**

Requirement ID	REQ_SR0301	Version	1.0
Item	Online Payment Button (Input)		
Item Description	A button with the word “Online Payment”		
Item Purpose	To proceed online payment		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.3.2: Online Banking Button**

Requirement ID	REQ_SR0302	Version	1.0
Item	Online Banking Button (Input)		
Item Description	A button with the word “FPX(Online Banking)”		
Item Purpose	To proceed online banking		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.3.3: TNG Button**

Requirement ID	REQ_SR0303	Version	1.0
Item	TNG Button (Input)		
Item Description	A button with the word “TNG”		
Item Purpose	To proceed TNG		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.3.4: Choose Bank Button**

Requirement ID	REQ_SR0304	Version	1.0
Item	Choose Bank Button (Input)		
Item Description	A button with the word “Choose Bank”		
Item Purpose	To choose bank that user want to proceed payment		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.3.5: Upload Payment Receipt Button**

Requirement ID	REQ_SR0305	Version	1.0
Item	Upload Payment Receipt Button (Input)		
Item Description	A button with the word “Upload Payment Receipt”		
Item Purpose	To Upload the proof for TNG payment		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.3.6: Proceed Button**

Requirement ID	REQ_SR0306	Version	1.0
Item	Proceed Button (Input)		
Item Description	A button with the word “Proceed”		
Item Purpose	After make payment click to proceed		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0302, REQ_SR0303, REQ_SR0304, REQ_SR0305 (The payment must be paid, then can proceed)		
Author	ERIC		

**Table 3.4.3.7: Done Button**

Requirement ID	REQ_SR0307	Version	1.0
Item	Done Button (Input)		
Item Description	A button with the word “Done”		
Item Purpose	After make payment click Done		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0306 (After successful to payment, then successful for registration)		
Author	ERIC		

**Table 3.4.3.8: Success Registration Message**

Requirement ID	REQ_SR0308	Version	1.0
Item	Success registration Message (Output)		
Item Description	A toast message with the success for registration message		
Item Purpose	User is success for register an event		
Input Format	-	Valid Input	-
Related L/O	REQ_SR0307 (After success for the payment)		
Author	ERIC		

**Table 3.4.3.9: Payment Successful Message**

Requirement ID	REQ_SR0308	Version	1.0
Item	Payment Successful Message (Output)		
Item Description	A toast message with the success payment		
Item Purpose	To notify user success to the payment		
Input Format	-	Valid Input	-
Related L/O	-		
Author	ERIC		

### 3.4.1.4 SR04 Refund Payment

**Table 3.4.4.1: Refund Button Button**

Requirement ID	REQ_SR0401	Version	1.0
Item	Refund Button (Input)		
Item Description	A button with the word “Refund”		
Item Purpose	Request refund for an event		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.4.2: Select Event Button**

Requirement ID	REQ_SR0402	Version	1.0
Item	Select Event Button (Input)		
Item Description	A button to choose event user register		
Item Purpose	After make payment click Done		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.4.3: Text field for “Reason For Refund”**

Requirement ID	REQ_SR0403	Version	1.0
Item	Text field for “Reason For Refund” (Input)		
Item Description	A text field for reason		
Item Purpose	To write refund why refund the events		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	TIANYOU		

**Table 3.4.4.4: Upload Payment Receipt Button (Input)**

Requirement ID	REQ_SR0404	Version	1.0
Item	Upload Payment Receipt Button (Input)		
Item Description	A button with the word “Upload Payment Receipt”		
Item Purpose	To upload the proof for the payment		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.4.5: Submit Button**

Requirement ID	REQ_SR0405	Version	1.0
Item	Submit Button (Input)		
Item Description	A button with the word “Submit”		
Item Purpose	After fill up all the details and submit		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0309, REQ_SR0310, REQ_SR0311 (After fill up details and submit)		
Author	TIANYOU		

**Table 3.4.4.6: Refund Status**

Requirement ID	REQ_SR0406	Version	1.0
Item	Refund Status (Input)		
Item Description	A button for Refund		
Item Purpose	View the refund is pending or approved		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

### 3.4.1.5 SR05 Rating

**Table 3.4.5.1: Rating Button**

Requirement ID	REQ_SR0501	Version	1.0
Item	Rating Button (Input)		
Item Description	A button with the word “Rating”		
Item Purpose	Go to the rating page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.5.2: Star Rating Button**

Requirement ID	REQ_SR0502	Version	1.0
Item	Star Rating Button (Input)		
Item Description	5 star button with the word “Star Rating”		
Item Purpose	Give the event's experience with clicking star		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.5.3: Feedback Field**

Requirement ID	REQ_SR0503	Version	1.0
Item	Feedback Field (Input)		
Item Description	A text field with the label “Feedback”		
Item Purpose	Give the event's feedback		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.5.4: Submit Button**

Requirement ID	REQ_SR0504	Version	1.0
Item	Submit Button (Input)		
Item Description	A button with the word "Submit"		
Item Purpose	To submit the feedback		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0502, REQ_SR0503 (Fill up the rating and submit)		
Author	KAISHEN		

**Table 3.4.5.5: Success Message**

Requirement ID	REQ_SR0505	Version	1.0
Item	Success Message (Output)		
Item Description	A toast message with the "Rating Success" message		
Item Purpose	To notify the user that the rating was successful		
Input Format	-	Valid Input	-
Related L/O	REQ_SR0504 (After finish the rating)		
Author	KAISHEN		

### 3.4.1.6 SR06 Scan Attendance

**Table 3.4.6.1: Scan Button**

Requirement ID	REQ_SR0601	Version	1.0
Item	Scan Button (Input)		
Item Description	A button with the word “Scan”		
Item Purpose	Scan the QR Code to take attendance		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.6.2: Full Name Field**

Requirement ID	REQ_SR0602	Version	1.0
Item	Full Name Field (Input)		
Item Description	A text field with the label “Full Name”		
Item Purpose	To enter user’s name to be check-in		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.6.3: Student ID Field**

Requirement ID	REQ_SR0603	Version	1.0
Item	Student ID Field (Input)		
Item Description	A text field with the label “Student ID”		
Item Purpose	To enter user’s Id to be check-in		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.6.4: Ticket ID Field**

Requirement ID	REQ_SR0604	Version	1.0
Item	Ticket ID Field (Input)		
Item Description	A text field with the label “Ticket ID”		
Item Purpose	To enter user's ticket ID to be check-in		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	YILING		

**Table 3.4.6.5: Check In Button**

Requirement ID	REQ_SR0605	Version	1.0
Item	Check in Button (Input)		
Item Description	A button with the word “Check-in”		
Item Purpose	To Check-in for an events		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0502, REQ_SR0503, REQ_SR0504 (After finish fill up, and check-in)		
Author	YILING		

**Table 3.4.6.6: Check-in Successful Message**

Requirement ID	REQ_SR0606	Version	1.0
Item	Check-in Successful Message (Output)		
Item Description	A toast message with the success check-in		
Item Purpose	To notify user success to check-in		
Input Format	-	Valid Input	-
Related L/O	REQ_SR0505 (After click check-in)		
Author	YILING		

### 3.4.1.7 SR07 Admin Dashboard

**Table 3.4.7.1: Create Event Button**

Requirement ID	REQ_SR0701	Version	1.0
Item	Create Event Button (Input)		
Item Description	A button with the word “Create Event”		
Item Purpose	To go to create event page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.7.2: Manage Event Button**

Requirement ID	REQ_SR0702	Version	1.0
Item	Manage Event Button (Input)		
Item Description	A button with the word “Manage Event”		
Item Purpose	To go to Manage Event page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.7.3: View Rating Button**

Requirement ID	REQ_SR0703	Version	1.0
Item	View Rating Button (Input)		
Item Description	A button with the word “View Rating”		
Item Purpose	To go to View Rating page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.7.4: View Payment Button**

Requirement ID	REQ_SR0704	Version	1.0
Item	View Payment Button (Input)		
Item Description	A button with the word “View Payment”		
Item Purpose	To go to View Payment page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.7.5: View Refund Button**

Requirement ID	REQ_SR0705	Version	1.0
Item	View Refund Button (Input)		
Item Description	A button with the word “View Refund”		
Item Purpose	To go to View Refund page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

### 3.4.1.8 SR08 Create Events

**Table 3.4.8.1: Venue Field**

Requirement ID	REQ_SR0801	Version	1.0
Item	Venue Field (Input)		
Item Description	A text field with the label “Venue”		
Item Purpose	To enter the Venue		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	TIANYOU		

**Table 3.4.8.2: Date Field**

Requirement ID	REQ_SR0802	Version	1.0
Item	Date Field (Input)		
Item Description	A text field with the label “Date”		
Item Purpose	To enter the Date		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	TIANYOU		

**Table 3.4.8.3: Time Field**

Requirement ID	REQ_SR0803	Version	1.0
Item	Time Field (Input)		
Item Description	A text field with the label “Time”		
Item Purpose	To enter the Time		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	TIANYOU		

**Table 3.4.8.4: Registration Details Field**

Requirement ID	REQ_SR0804	Version	1.0
Item	Registration Details Field (Input)		
Item Description	A text field with the label “Registration Details”		
Item Purpose	To enter the registration details		
Input Format	String	Valid Input	ASCII code from decimal 32 to 126
Related L/O	-		
Author	TIANYOU		

**Table 3.4.8.5: Create Button**

Requirement ID	REQ_SR0805	Version	1.0
Item	Create Button (Input)		
Item Description	A button with the word “Create”		
Item Purpose	To create the events		
Input Format	Button	Valid Input	-
Related L/O	REQ_SR0801, REQ_SR0802, REQ_SR0803, REQ_SR0804( After fill up the information and create)		
Author	TIANYOU		

### 3.4.1.9 SR09 Manage Events

**Table 3.4.9.1: Edit Button**

Requirement ID	REQ_SR0901	Version	1.0
Item	Edit Button (Input)		
Item Description	A button with the word “Edit”		
Item Purpose	To edit the events		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.9.2: QR Code Button**

Requirement ID	REQ_SR0902	Version	1.0
Item	Qr Code Button (Input)		
Item Description	A button with the word “Qr Code”		
Item Purpose	To go to the Qr Code Page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.9.3: Generate Button**

Requirement ID	REQ_SR0903	Version	1.0
Item	Generate Button (Input)		
Item Description	A button with the word “Generate”		
Item Purpose	To generate Qr Code give user to check-in		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

### 3.4.1.10 SR10 Track Attendance

**Table 3.4.10.1: Track Attendance Button**

Requirement ID	REQ_SR1001	Version	1.0
Item	Track Attendance Button (Input)		
Item Description	A button with the icon		
Item Purpose	To go to the track attendance page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.2: Total Register**

Requirement ID	REQ_SR1002	Version	1.0
Item	Total Register (Output)		
Item Description	A toast message with the total register		
Item Purpose	To show the total register for admin		
Input Format	-	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.3: Total Present**

Requirement ID	REQ_SR1003	Version	1.0
Item	Total Present (Output)		
Item Description	A toast message with the total Present		
Item Purpose	To show the total Present for admin		
Input Format	-	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.4: Total Absent**

Requirement ID	REQ_SR1004	Version	1.0
Item	Total Absent (Output)		
Item Description	A toast message with the total absent		
Item Purpose	To show the total absent for admin		
Input Format	-	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.5: Total Late**

Requirement ID	REQ_SR1005	Version	1.0
Item	Total Late (Output)		
Item Description	A toast message with the total late		
Item Purpose	To show the total late for admin		
Input Format	-	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.6: View Attendance Analytic Button**

Requirement ID	REQ_SR1006	Version	1.0
Item	View Attendance Analytic Button (Input)		
Item Description	A button with the word “View Attendance Analytic”		
Item Purpose	To go to the Analytic page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	YILING		

**Table 3.4.10.7: Pie Chart**

Requirement ID	REQ_SR1007	Version	1.0
Item	Pie Chart (Output)		
Item Description	A Pie Chart		
Item Purpose	To show the percentage for admin		
Input Format	-	Valid Input	-
Related L/O	-		
Author	YILING		

### **3.4.1.11 SR11 View Rating & Feedback**

**Table 3.4.11.1: View Button**

Requirement ID	REQ_SR1101	Version	1.0
Item	View Button (Input)		
Item Description	A button with the word “View”		
Item Purpose	To view the feedback and rating		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	ERIC		

**Table 3.4.11.2: Rating & Feedback**

Requirement ID	REQ_SR1102	Version	1.0
Item	Rating & Feedback (Output)		
Item Description	A rating and feedback from the events		
Item Purpose	To display feedback and rating		
Input Format	-	Valid Input	-
Related L/O	-		
Author	ERIC		

### 3.4.1.12 SR12 View Payment Details

**Table 3.4.12.1: View Button**

Requirement ID	REQ_SR1201	Version	1.0
Item	View Button (Input)		
Item Description	A button with the word “View”		
Item Purpose	To view the payment details		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.12.2: Payment Details**

Requirement ID	REQ_SR1202	Version	1.0
Item	Payment Details (Output)		
Item Description	A payment details from the events		
Item Purpose	To display payment details		
Input Format	-	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.12.3: Generate Payment Report Button**

Requirement ID	REQ_SR1203	Version	1.0
Item	Generate Payment Report Button (Input)		
Item Description	A button with the word “Generate Payment Report”		
Item Purpose	To go to the payment report page		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	TIANYOU		

**Table 3.4.12.4: Payment Report**

Requirement ID	REQ_SR1204	Version	1.0
Item	Payment Report (Output)		
Item Description	A payment Report from the events		
Item Purpose	To display payment report		
Input Format	-	Valid Input	-
Related L/O	-		
Author	TIANYOU		

### 3.4.1.13 SR13 View Request Refund

**Table 3.4.13.1: View Button**

Requirement ID	REQ_SR1301	Version	1.0
Item	View button (Input)		
Item Description	A button with the word “View”		
Item Purpose	To view the user who request refund		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.13.2: Approve Button**

Requirement ID	REQ_SR1302	Version	1.0
Item	Approve button (Input)		
Item Description	A button with the word “Approve”		
Item Purpose	To approve the user's refund		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

**Table 3.4.13.3: Declined Button**

Requirement ID	REQ_SR1303	Version	1.0
Item	Declined button (Input)		
Item Description	A button with the word “Declined”		
Item Purpose	To declined the user's refund		
Input Format	Button	Valid Input	-
Related L/O	-		
Author	KAISHEN		

### 3.4.2 System Interfaces

The Campus Event Check-in System (CECS) interfaces with several external systems to support secure user authentication, real-time event attendance tracking, digital payment processing, and mobile-based user notifications. The table below outlines the key system interfaces that interact with the CECS mobile application.

**Table 3.4.2.1: System Interfaces**

Interface ID	System Name	Description	Details	Author
REQ_SI_001	Authentication Server	Interfaces with university login service	Enables secure login for students and admins via OAuth 2.0 or LDAP	ERIC
REQ_SI_002	Payment Server	External gateway for financial transactions	Supports online (FPX, TNG eWallet) and onsite payments, refund APIs	ERIC
REQ_SI_003	University Database	Central student and event records	Validates student IDs, fetches event info, and logs attendance records	YILING
REQ_SI_004	Notification Service	In-app messaging and alert tool	Displays simple pop-up notifications within the CECS mobile app.	TIANYOU
REQ_SI_005	QR Code Module	QR generation and scanning tool	Generates unique QR codes for events; uses mobile camera for scanning	YILING
REQ_SI_006	Report Generator	Analytics and reporting module	Generates event-wise reports for attendance, payment, and feedback	KAISHEN
REQ_SI_007	Receipt Generation API	Confirms successful payments	Provides digital receipts viewable in student dashboard	TIANYOU

### 3.4.3 User Interfaces

The Campus Event Check-in System (CECS) provides a mobile-friendly graphical user interface (GUI) designed for ease of use, readability, and accessibility. The system ensures consistency in design across modules .

The following table details the key user interface elements:

**Table 3.4.3.1: User Interfaces**

Module ID	Module Name	Description	Priority	Author
REQ_UI_001	Login Interface	Allows students and admins to securely log in using university credentials with a simple UI.	High	TIAN YOU
REQ_UI_002	Event List Interface	Displays a scrollable list of upcoming events with filters (e.g., category, date, fee).	High	ERIC
REQ_UI_003	Event Registration Form	Allows students to register for selected events, select payment methods, and submit registration.	High	ERIC
REQ_UI_004	Payment Interface	Interface for selecting payment options (online via FPX/TNG or on-site) and viewing payment receipt.	High	TIANYOU
REQ_UI_005	QR Check-in Interface	Enables students to scan QR codes and input Student ID, Name, and Ticket ID for attendance.	High	YILING
REQ_UI_006	Feedback Submission Form	Allows students to rate events and submit optional comments post-event.	Medium	KAISHEN
REQ_UI_007	Admin Event Dashboard	Admin interface to create/edit events, generate QR codes, and view event details.	High	KAISHEN
REQ_UI_008	Attendance Tracking View	Enables real-time monitoring of check-in status during events for admins.	High	YILING
REQ_UI_009	Refund Request Page	Allows students to submit refund requests linked to their ticket/payment.	Medium	ERIC

REQ_UI010	Admin Reporting Interface	Generates attendance, payment, and feedback reports for admin analysis.	Medium	TIANYOU
REQ_UI011	Notification Panel	Displays in-app notifications for successful registration and payment confirmation..	Medium	KAISHEN

### 3.4.4 Hardware Interfaces

The Campus Event Check-in System (CECS) is designed primarily for mobile devices. To ensure optimal performance, compatibility, and full access to all features such as QR scanning, event registration, and in-app real-time notifications, devices should meet the following minimum hardware specifications.

**Table 3.4.4.1: Hardware Interface Requirements**

Interface ID	Description	Author
REQ_HI_001	The mobile device shall have a <b>64-bit ARM</b> processor (e.g., Snapdragon, MediaTek, or Apple A-series) to ensure smooth operation and camera access.	ERIC
REQ_HI_002	The mobile device shall have <b>at least 4GB of RAM</b> to support multitasking and responsive UI interactions.	TIANYOU
REQ_HI_003	The mobile device shall have a <b>minimum of 650MB</b> free internal storage to install the application and handle cached data such as scanned tickets and images.	YILING
REQ_HI_004	The mobile device shall include a <b>camera</b> with autofocus ( <b>minimum 8MP</b> ) to enable accurate QR code scanning.	KAISHEN
REQ_HI_005	The mobile device shall support <b>Wi-Fi and/or mobile data (4G/5G)</b> for real-time access to event and user data.	KAISHEN
REQ_HI_006	The mobile device shall have a touchscreen with a minimum resolution of <b>720p (1280x720)</b> for proper GUI rendering.	YILING
REQ_HI_007	The operating system shall be <b>Android 8.0 (Oreo) or iOS 12</b> and above to ensure compatibility with the app's native functions and security features.	TIANYOU

REQ_HI_008	The device shall have a battery capacity of <b>at least 2000mAh</b> to support extended event usage without disruption.	ERIC
REQ_HI_009	The mobile device should support <b>ambient light sensors</b> for brightness adjustments in different event environments.	ERIC
REQ_HI_010	The camera should support a scanning distance of <b>at least 10–20 cm</b> for accurate QR code recognition.	YILING

### **3.4.5 Software Interfaces**

The Campus Event Check-in System requires other supporting software components for its core functionalities. These software interfaces ensure the proper operation of features.

**Table 3.4.5.1: Software Interfaces**

<b>Interface ID</b>	<b>Category</b>	<b>Name</b>	<b>Version Number</b>	<b>Purpose</b>	<b>Reference</b>	<b>Author</b>
REQ_SI_001	Operating System	Android	Android 8.0 (Oreo) or later	Required platform for running the CECS mobile app.	Android Developer Docs	ERIC
REQ_SI_002	Operating System	iOS	iOS 12.0 or later	Required platform for running the CECS mobile app on Apple devices.	Apple iOS Developer	TIANYOU

### 3.4.6 Communication Interfaces

This section outlines the communication protocols and methods the **Campus Event Check-in System (CECS)** employs to securely interact with backend servers, authentication services, payment gateways, databases, and notification systems, ensuring smooth, reliable, and real-time operation.

**Table 3.4.6.1: Communication Interfaces**

Interface ID	Description	Protocols/Methods	Priority	Author
REQ_CI_001	The mobile app shall use HTTPS for secure communication with the university's web server.	HTTPS (TLS 1.2 or higher)	High	ERIC
REQ_CI_002	The system shall communicate with Firebase Authentication and Realtime Database services.	Firebase REST API, Firebase SDK	High	ERIC
REQ_CI_003	The system shall support in-app real-time notifications triggered by backend events or polling.	RESTful API Polling or WebSocket	Medium	TIANYOU
REQ_CI_004	The app shall integrate QR code scanning features using embedded SDK or external libraries.	Mobile Integration	Medium	YILING
REQ_CI_005	The system shall use JSON as the data format for all client-server communication.	JSON over HTTPS	High	KAISHEN
REQ_CI_006	The mobile app shall connect to backend services using RESTful APIs for event, user, and payment data.	RESTful API over HTTPS	High	KAISHEN
REQ_CI_007	The system shall support OAuth 2.0 for user authentication and authorization with external services if integrated in future.	OAuth 2.0	Medium	YILING
REQ_CI_008	The system shall handle retry and timeout mechanisms for communication failures.	Exponential backoff, timeouts	Medium	TIANYOU
REQ_CI_009	The app shall encrypt sensitive data in transit to comply with university security policies.	TLS Encryption	High	ERIC

### 3.5 Logical database requirements

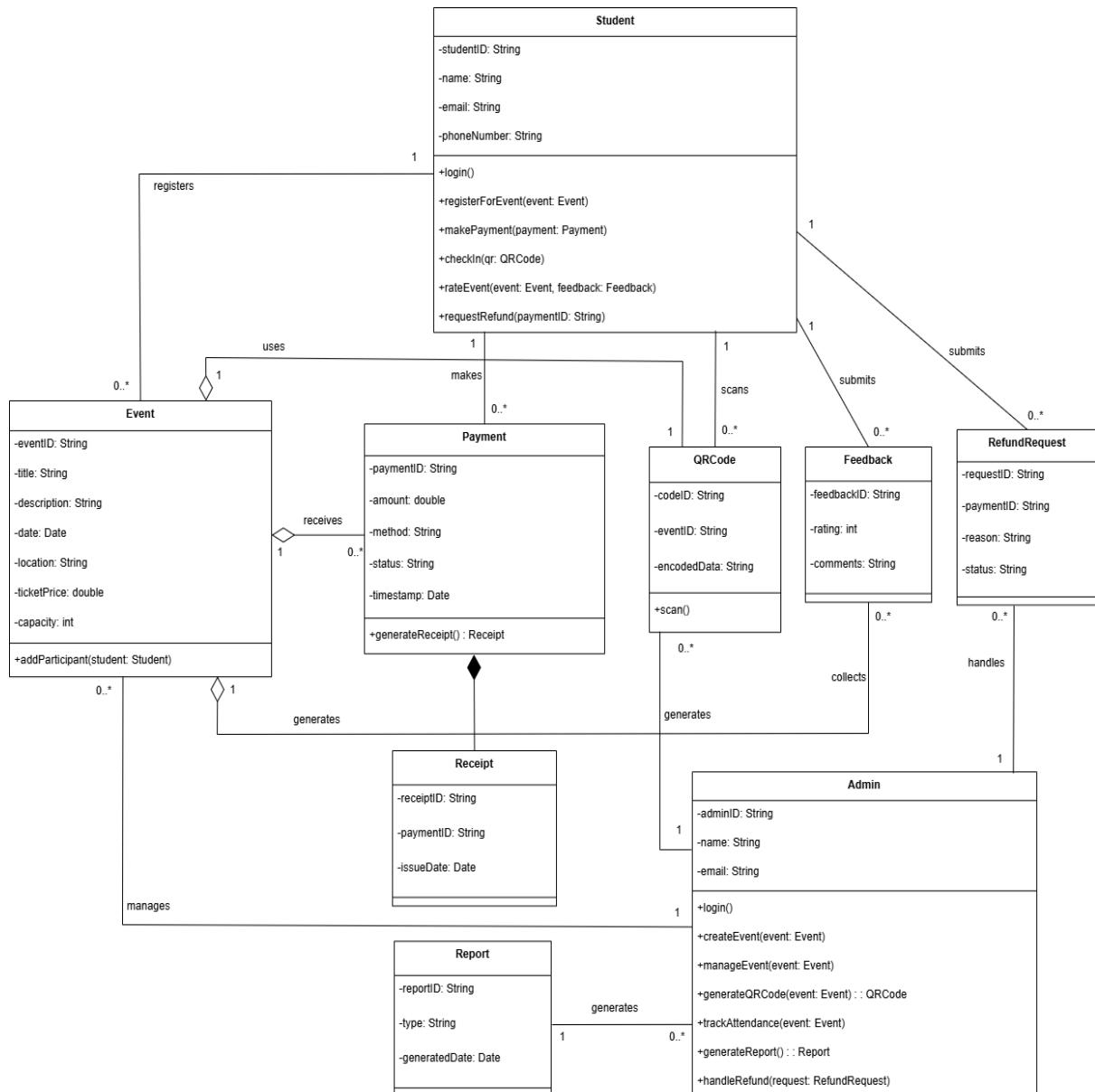
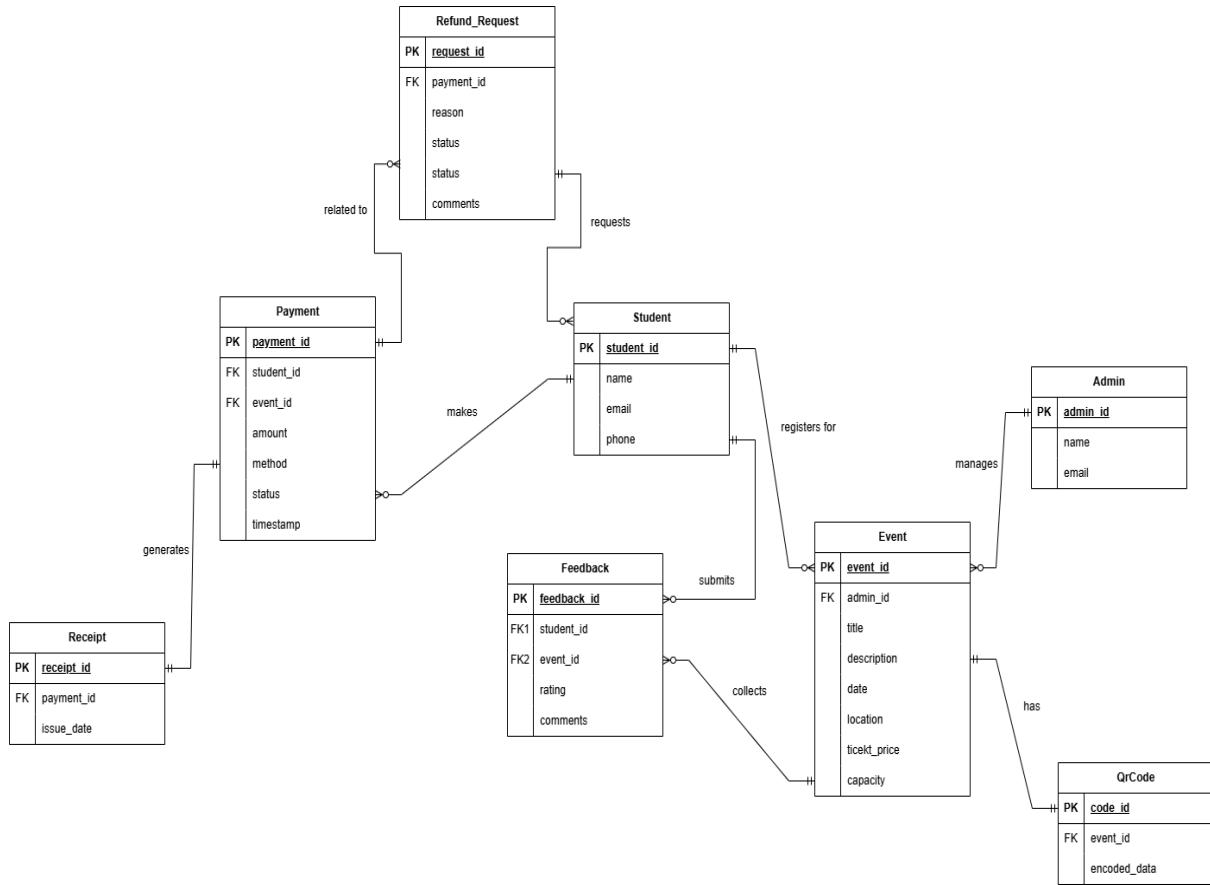


Figure 3.5.1: Class Diagram



**Figure 3.5.1.2: System Entity-Relationship Diagram**

The Entity-Relationship Diagram (ERD) represents the structure and relationships of entities within the Campus Event Check-in System. The key entities include Student, Admin, Event, QRCode, Payment, Receipt, RefundRequest, and Feedback. This model supports the core functions of student event registration, attendance tracking through QR codes, payment handling, feedback collection, and administrative reporting.

### 3.5.1 Student

**Table 3.5.1.1: Student Data Dictionary**

Field Name	Description	Data Type	Constraints	Extra Notes
student_id	Unique ID of student	varchar(50)	Primary Key, NOT NULL, Unique	Usually university-assigned
name	Student's full name	varchar(50)	NOT NULL	Used as login credentials
email	Email used for login and notifications	varchar(50)	Unique, NOT NULL	Used for login and notifications
phone	Contact number	varchar(20)	NULLABLE	Optional field

### 3.5.2 Admin

**Table 3.5.2.1: Admin Data Dictionary**

Field Name	Description	Data Type	Constraints	Extra Notes
admin_id	Unique ID of admin	varchar(50)	Primary Key, NOT NULL, Unique	Assigned by system
name	Admin's full name	varchar(50)	NOT NULL	Used as login credentials
email	Email used for login	varchar(50)	Unique, NOT NULL	Used for login

### 3.5.3 Event

**Table 3.5.3.1: Event Data Dictionary**

Field Name	Description	Data Type	Constraints	Extra Notes
event_id	Unique event ID	varchar(50)	Primary Key, NOT NULL, Unique	Auto-generated
title	Event name	varchar(100)	NOT NULL	-
description	Details about the event	varchar(255)	NULLABLE	-
date	Event date	date	NOT NULL	-

location	Venue of the event	varchar(100)	NOT NULL	-
ticket_price	Cost per ticket (can be free)	double	DEFAULT 0.0	0 means free
capacity	Maximum participants	int	NOT NULL	-
admin_id	Event organizer	varchar(100)	FK → ADMIN.admin_id	-

### 3.5.4 Payment

Table 3.5.4.1: Payment Data Dictionary

Field Name	Description	Data Type	Constraints	Extra Notes
payment_id	Unique payment ID	varchar(50)	Primary Key, NOT NULL, Unique	-
amount	Amount paid	double	NOT NULL	-
method	Payment method	varchar(20)	NOT NULL	E.g., "online", "onsite"
status	Payment status	varchar(20)	NOT NULL	E.g., "completed", "pending", "refunded"
timestamp	Time of transaction	datetime	DEFAULT current_timestamp	-
student_id	Student who made the payment	varchar(50)	FK → STUDENT.student_id	-
event_id	Event paid for	varchar(50)	FK → EVENT.event_id	-

### 3.5.5 Feedback

Table 3.5.5.1: Feedback Data Dictionary

Field Name	Description	Data Type	Constraints	Extra Notes
feedback_id	Unique feedback ID	varchar(50)	Primary Key, NOT NULL, Unique	-

student_id	Student who gave the feedback	varchar(50)	FK → STUDENT.student_id	-
event_id	Event being rated	varchar(50)	FK → EVENT.event_id	-
rating	Rating out of 5	int	CHECK (rating BETWEEN 1 AND 5)	Required
comments	Optional comments	varchar(100)	NULLABLE	Optional

### 3.5.6 Receipt

Table 3.5.6.1: Receipt Data Dictionary

Field Name	Description	Data Type	Constraints	Extra Notes
receipt_id	Unique receipt ID	varchar(50)	Primary Key, NOT NULL, Unique	-
payment_id	Related payment	varchar(50)	FK → PAYMENT.payment_id	One-to-one relationship
issue_date	Date when the receipt was issued	date	NOT NULL	Could default to payment date

### 3.5.7 Refund Request

Table 3.5.7.1: Refund Request Data Dictionary

Field Name	Description	Data Type	Constraints	Extra Notes
request_id	Unique refund request	varchar(50)	Primary Key, NOT NULL, Unique	-
payment_id	Related payment	varchar(50)	FK → PAYMENT.payment_id	-
reason	Reason for refund	varchar(100)	NULLABLE	Provided by student
status	Request status	string	DEFAULT 'pending'	E.g., "approved", "declined"

### 3.5.8 QRCode

Table 3.5.8.1: QRCode Data Dictionary

Field Name	Description	Data Type	Constraints	Extra Notes
code_id	Unique QR code ID	varchar(50)	Primary Key, NOT NULL, Not Null	Auto-generated
event_id	Related event	varchar(50)	FK → EVENT.event_id	One QR per event
encoded_data	Encoded check-in data	varchar(255)	NOT NULL	Data used for check-in

### 3.6 Design Constraints

This section outlines the design constraints and limitations that must be observed during the development of the **Campus Event Check-in System (CECS)**. These constraints may arise from technical requirements, regulatory compliance, organizational standards, or mobile platform limitations.

**Table 3.6.1: Design Constraints**

Requirement ID	Description	Priority	Author
REQ_DC_001	The mobile app shall comply with university branding guidelines, including color schemes, logos, and typography.	High	YILING
REQ_DC_002	The system shall comply with the Personal Data Protection Act (PDPA) in handling student information.	High	KAISHEN
REQ_DC_003	The application shall support Android 8.0+ and iOS 12+ mobile operating systems.	High	TIANYOU
REQ_DC_004	The platform shall be developed using Firebase as the backend to ensure real-time data processing and push notification support.	Medium	ERIC
REQ_DC_005	The design shall prioritize responsiveness across various mobile screen sizes and resolutions.	High	TIANYOU
REQ_DC_006	QR code scanning shall be optimized for camera modules with autofocus and minimum 8MP resolution.	Medium	YILING
REQ_DC_007	The system must integrate with existing university services, including Authentication Server and Payment Gateway, without requiring major changes.	High	KAISHEN
REQ_DC_008	The system shall restrict access to admin features using role-based access control (RBAC).	High	KAISHEN
REQ_DC_009	The app shall be developed and delivered within one academic semester (e.g., 14–16 weeks).	High	YILING
REQ_DC_010	All payment transactions shall follow secure protocols .	High	YILING

REQ_DC_011	The system shall store data securely in cloud storage and must support daily backups.	Medium	ERIC
REQ_DC_012	The app shall avoid using any proprietary SDKs or paid third-party plugins. Open-source tools must be prioritized.	Medium	TIANYOU

## 3.7 Software System Attributes

This section outlines the non-functional requirements of the Campus Event Check-in System (CECS), specifying the essential system qualities related to availability, performance, security, usability, and more. These attributes are crucial to ensure the system delivers a reliable, secure, and user-friendly experience for both students and administrators.

### 3.7.1 Availability

**Table 3.7.1.1: Availability**

Requirement ID	Description	Priority	Author
REQ_SA_001	The system shall be operational 24/7 with minimal downtime, especially during active registration and event periods.	High	KAISHEN
REQ_SA_002	Firebase backend must support real-time updates, ensuring continuous access to registration and check-in services.	High	KAISHEN

### 3.7.2 Reliability

**Table 3.7.2.1: Reliability**

Requirement ID	Description	Priority	Author
REQ_SA_003	Core functionalities such as login, QR check-in, payment, and refund processing must execute without failure under normal usage.	High	YILING
REQ_SA_004	The system shall be fault-tolerant, automatically recovering from transient errors during scanning or network interruptions.	Medium	YILING

### 3.7.3 Security

**Table 3.7.3.1: Security**

Requirement ID	Description	Priority	Author
REQ_SA_005	All data transmission shall be encrypted using HTTPS with TLS to protect user credentials, payment details, and personal information.	High	TIANYOU
REQ_SA_006	The system shall integrate with University OAuth authentication (LDAP) to ensure secure and verifiable login.	High	TIANYOU
REQ_SA_007	Role-Based Access Control (RBAC) shall be implemented to ensure only authorized users access sensitive admin functions like refund approval and report generation.	High	TIANYOU

### 3.7.4 Auditability

**Table 3.7.4.1: Auditability**

Requirement ID	Description	Priority	Author
REQ_SA_008	The system shall maintain audit logs for sensitive operations (e.g., login, payment, refunds, attendance edits) with timestamp, user ID, and action type.	High	ERIC

### 3.7.5 Performance

**Table 3.7.5.1: Performance**

Requirement ID	Description	Priority	Author
REQ_SA_009	The system shall support concurrent check-ins for high-volume events without delay or lag in data sync.	High	ERIC
REQ_SA_010	User actions (login, QR scan, event listing, payment) shall complete within 3 seconds under standard operating conditions.	High	ERIC

### 3.7.6 Maintainability

**Table 3.7.6.1: Maintainability**

Requirement ID	Description	Priority	Author
REQ_SA_011	The system backend shall use modular Firebase functions to enable easier debugging and independent updates to QR, payment, and event modules.	Medium	YILING
REQ_SA_012	Frontend code shall follow consistent styling, naming conventions, and component reuse to simplify future UI enhancements.	Medium	KAISHEN

### 3.7.7 Portability

**Table 3.7.7.1: Portability**

Requirement ID	Description	Priority	Author
REQ_SA_013	The system shall support Android 8.0+ and iOS 12+ platforms, with full functionality on both device types.	High	TIANYOU
REQ_SA_014	The app shall scale across devices of varying screen sizes and resolutions, ensuring consistent layout and interaction.	High	ERIC

### 3.7.8 Usability

**Table 3.7.8.1: Usability**

Requirement ID	Description	Priority	Author
REQ_SA_015	The interface shall be intuitive and easy to navigate for students and admins, requiring minimal training or technical knowledge.	High	KAISHEN
REQ_SA_016	Key workflows (e.g., event registration, check-in, refund request) shall require no more than 3 taps from the home screen.	High	YILING

### **3.7.9 Responsiveness**

**Table 3.7.9.1: Responsiveness**

Requirement ID	Description	Priority	Author
REQ_SA_017	The user interface shall dynamically adapt to device orientation and screen size without breaking layout or interactivity.	High	ERIC

### **3.7.10 Scalability**

**Table 3.7.10.1: Scalability**

Requirement ID	Description	Priority	Author
REQ_SA_018	The Firebase backend shall automatically scale to accommodate increased student participation during university-wide events.	Medium	ERIC
REQ_SA_019	Admin reporting and analytics shall remain performant even when generating data for large-scale multi-day events.	Medium	TIANYOU

### 3.8 Supporting Information

**Table 3.8.1: Kano Categories**

Identifier	Requirement	Kano Category	Elicitation Method	Description
REQ_001	View Event List	Dissatisfier	Prototype + Questionnaire	Students can browse available events to join.
REQ_002	Register for Event	Dissatisfier	Prototype + Questionnaire	Allows students to sign up for events via the system.
REQ_003	Make Online Payment	Dissatisfier	Prototype + Questionnaire	Users can choose an online method (e.g., FPX, Touch 'n Go) to complete payment for events.
REQ_004	Make Onsite Payment	Dissatisfier	Prototype + Questionnaire	Students can opt to pay physically at the event counter using QR code verification.
REQ_005	Check-In via QR Code	Dissatisfier	Prototype + Questionnaire	Students scan a QR code during the event to mark their attendance.
REQ_006	Create Event	Dissatisfier	Prototype + Questionnaire	Enables admin to input event information to

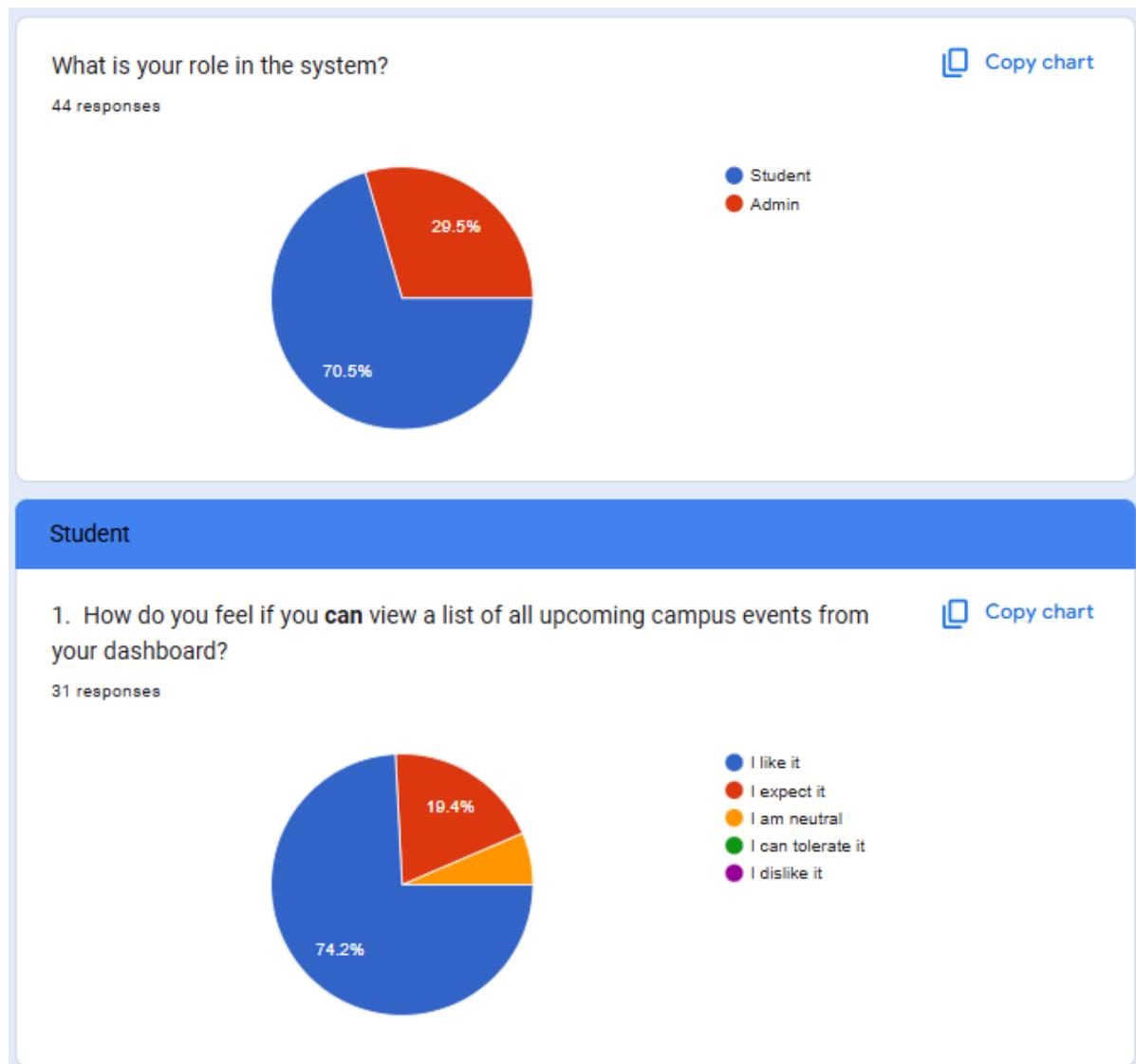
				publish new events on the platform.
REQ_007	Manage Event	Dissatisfier	Prototype + Questionnaire	Lets admins edit, update, or remove event details from the system.
REQ_008	Generate QR Code	Dissatisfier	Prototype + Questionnaire	Allows admins to create QR codes used for student event check-in.
REQ_009	Generate Payment Report	Dissatisfier	Prototype + Questionnaire	Admins can generate a financial report for event payments received.
REQ_010	Receive Notification	Satisfier	Prototype + Questionnaire	Students can receive notification when an event successfully register.
REQ_011	Request Refund	Satisfier	Prototype + Questionnaire	Students can submit a refund request by providing a reason and payment receipt.
REQ_012	View Payment Receipt	Satisfier	Prototype + Questionnaire	Students can view or download receipts for past event payments.
REQ_013	Rate Event	Satisfier	Prototype + Questionnaire	Allows users to submit a rating after attending an event.

REQ_014	View Rating or Feedback	Satisfier	Prototype + Questionnaire	Displays ratings and comments submitted by students for each event.
REQ_015	Track Attendance	Satisfier	Prototype + Questionnaire	Allows real-time tracking of students who check in for events, helping admins monitor attendance.
REQ_016	View Attendance Analytic	Satisfier	Prototype + Questionnaire	Admins can view attendance data visualizations to assess participation rates.
REQ_017	Handle Refund Request	Satisfier	Prototype + Questionnaire	Enables users to request refunds and allows finance/admin to manage and respond to those requests.
REQ_018	Event Attendance Rewards	Delighter	Questionnaire	Students can earn digital badges or points for attending multiple events.
REQ_019	Friend Participation Visibility	Delighter	Questionnaire	Allows students to see which friends have joined an event.
REQ_020	Event Calendar Sync	Delighter	Questionnaire	Syncs registered events with a student's personal calendar (e.g., Google Calendar).

REQ_021	Shared Event With Friends	Delighter	Questionnaire	Students can share events via link or social media with friends.
REQ_022	Auto Close Registration	Delighter	Questionnaire	Automatically stops event registration when capacity is reached.
REQ_023	Check-In Heatmap	Delighter	Questionnaire	Provides a visual heatmap of check-in data, showing where and when most attendees checked in.

### 3.8.1 Questionnaire Analytics

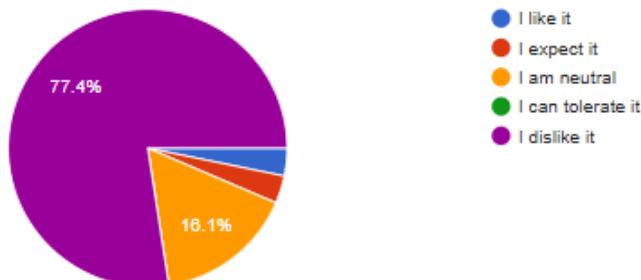
The charts below present the survey findings, with responses categorized according to the Kano model to distinguish must-have features, standard expectations, and delightful bonuses. While a detailed analysis and requirements are outlined in Section 3, this section emphasizes the visual data underpinning those classifications.



3. How do you feel if you **cannot** make online payments and must pay cash on-site?

 [Copy chart](#)

31 responses

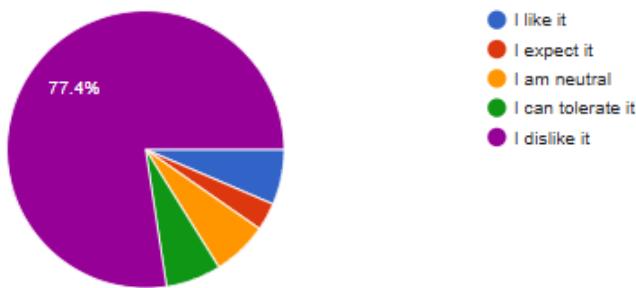


- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

4. How do you feel if there is **no** QR code and you must manually write your name during check-in?

 [Copy chart](#)

31 responses

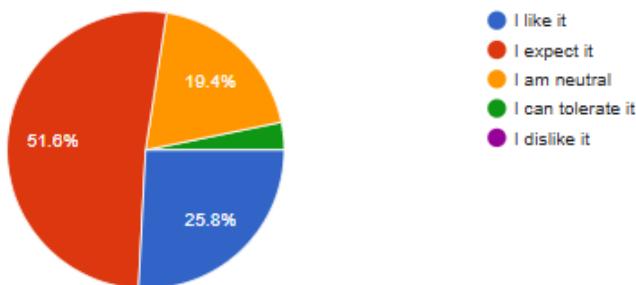


- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

5. How do you feel if you **can** request a refund through the system for paid events if you cannot attend the event?

 [Copy chart](#)

31 responses

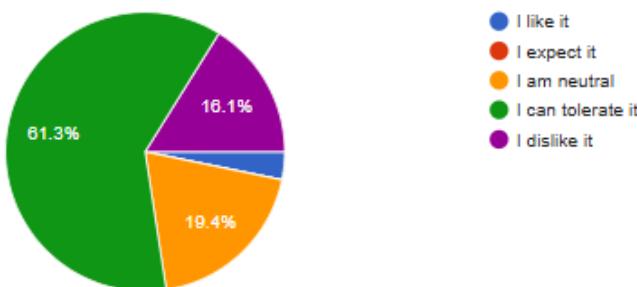


- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

6. How do you feel if there are **no** payment receipts available?

 [Copy chart](#)

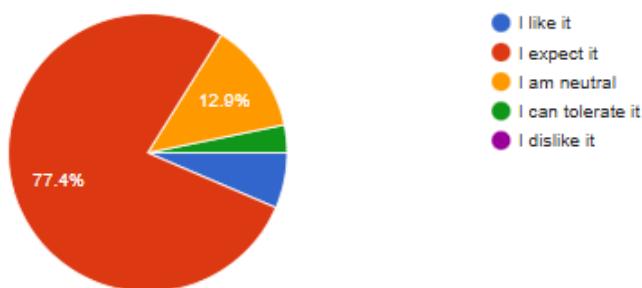
31 responses



7. How do you feel if you **can** give ratings and feedback after an event?

 [Copy chart](#)

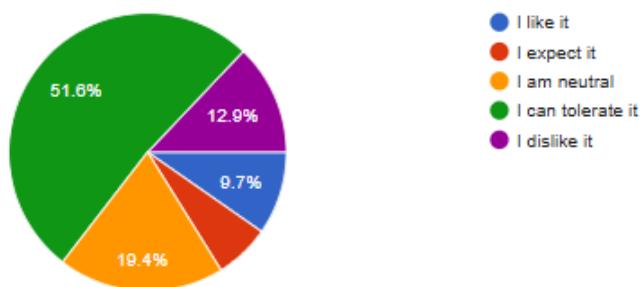
31 responses



8. How do you feel if you receive **no** notifications at all?

 [Copy chart](#)

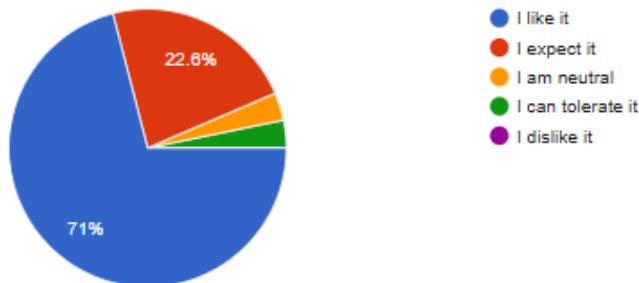
31 responses



9. How do you feel if you **can** earn points or badges for attending events?

 Copy chart

31 responses



I like it

I expect it

I am neutral

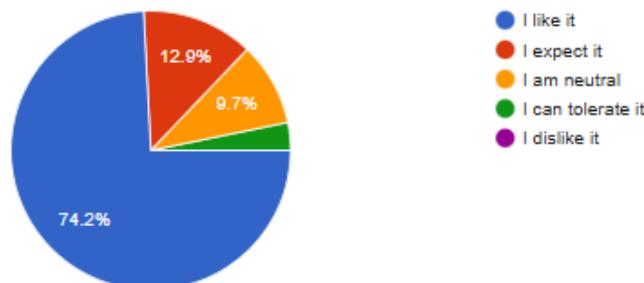
I can tolerate it

I dislike it

10. How do you feel if you **can** see which of your friends (from the same university) registered for the same event?

 Copy chart

31 responses



I like it

I expect it

I am neutral

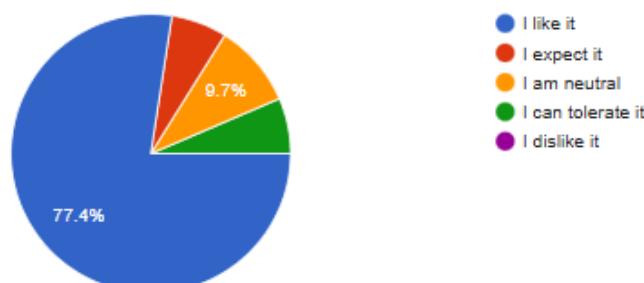
I can tolerate it

I dislike it

11. How do you feel if you **can** export events to your personal calendar (Google Calendar/Outlook)?

 Copy chart

31 responses



I like it

I expect it

I am neutral

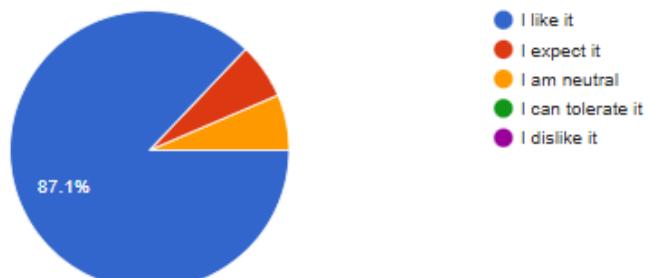
I can tolerate it

I dislike it

12. How would you feel if you **can** share event info with your friends via social media or messaging apps?

 Copy chart

31 responses



Admin

1. How do you feel if the system **allows** you to create events directly from your mobile device?

 Copy chart

13 responses

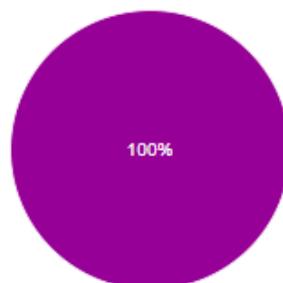


- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

2. How do you feel if the system does **not** generate QR codes for check-in?

 Copy chart

13 responses

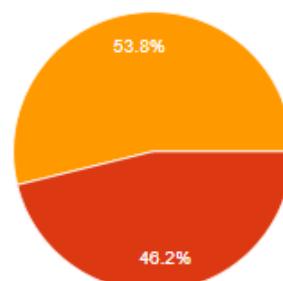


- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

3. How do you feel if you **can** view real-time attendance during an event?

 Copy chart

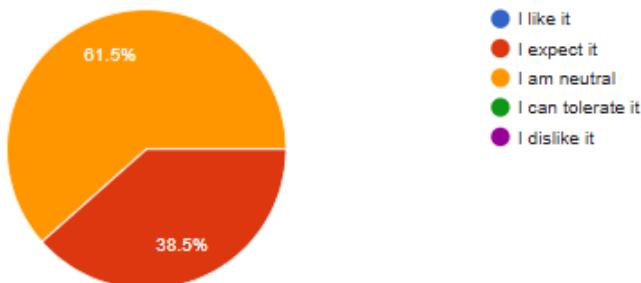
13 responses



- I like it
- I expect it
- I am neutral
- I can tolerate it
- I dislike it

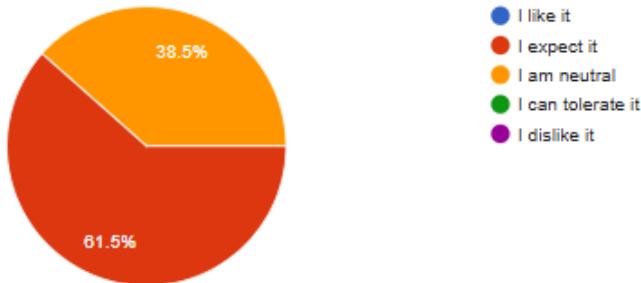
4. How do you feel if you **can** view student ratings and feedback after the event? [Copy chart](#)

13 responses



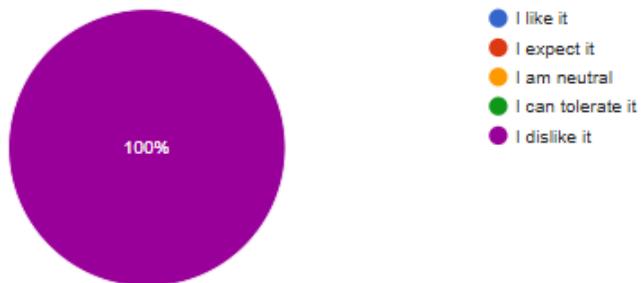
5. How do you feel if you **can** view and approve refund requests from the app? [Copy chart](#)

13 responses



6. How do you feel if the system **does not** generate financial reports? [Copy chart](#)

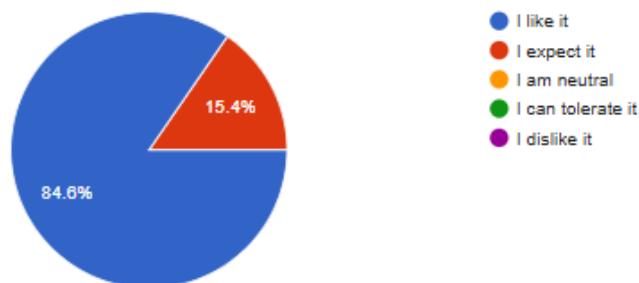
13 responses



7. How would you feel if the system **could** automatically close event registration once the limit is reached?

[Copy chart](#)

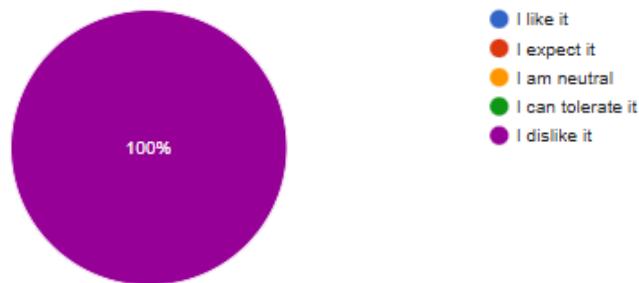
13 responses



8. How would you feel if **no** heatmap or check-in trends were shown during events?

[Copy chart](#)

13 responses



### 3.8.2 Prototyping

[Click Here Direct To Our Prototype](#)



**Figure 3.8.2.1: Login Page (User)**

This is the sign in page for users of Campus Event Check-in System (CECS). In the sign in page, there are User ID input and password input that enable users to enter to the next page. Click the 'Admin Login page' can directly move to the admin login page. Users need to click the login button to enter to the next page.



**Figure 3.8.2.2: Login Page (Admin)**

This is the sign in page for admin of Campus Event Check-in System (CECS). In the sign in page, there are Admin ID input and password input that enable users to enter to the next page. Click the 'User Login page' can directly move to the user login page. Admin needs to click the login button to enter to the next page.



**Figure 3.8.2.3: Unparticipated Event List Page (User)**

This is the unparticipated event page for users of Campus Event Check-in System (CECS). In this page, users can scroll down to view the event lists and choose events to register. Click the 'Participated' can directly view the participated event lists. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



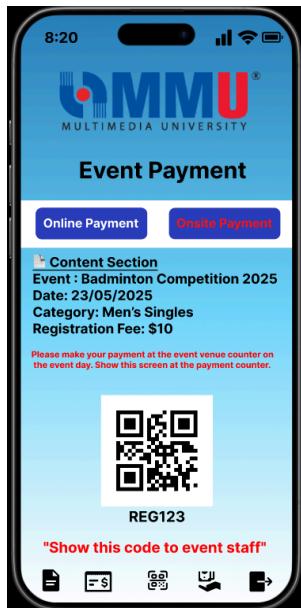
**Figure 3.8.2.4: Event Details Page (User)**

This is the event details page for users of Campus Event Check-in System (CECS). This page shows the details of the event that users selected to register. Users can click the 'Register' button to enter the register event page. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



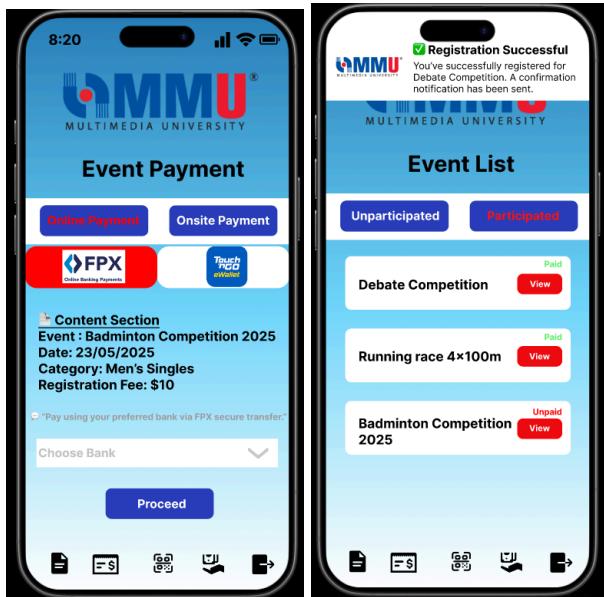
**Figure 3.8.2.5: Register Event Page (User)**

This is the register event page for users of Campus Event Check-in System (CECS). In this page, users need to insert the full name, age, phone number, email and category of the event. After filling the form, users can click the 'Proceed to payment' button to enter the next page. Users can click the 'Cancel' button to decline the process.



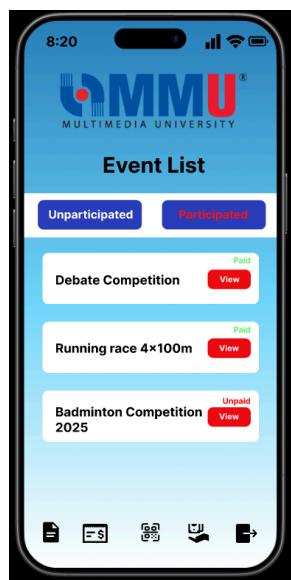
**Figure 3.8.2.6: Event Onsite Payment Page (User)**

This is the event onsite payment page for users of Campus Event Check-in System (CECS). In this page, users can choose online payment or onsite payment to proceed their payment. Users need to show the QR Code at the counter to do the onsite payment. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



**Figure 3.8.2.7: Event Online Payment Page (User)**

This is the event online payment page for users of Campus Event Check-in System (CECS). In this page, users can choose online payment or onsite payment to proceed their payment. Users need to choose the method of payment (FPX or Touch N Go) to do the online payment. If users choose FPX, users have to choose the bank. Click the 'Proceed' button to finish the payment and it will show the registration successful after done the payment. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



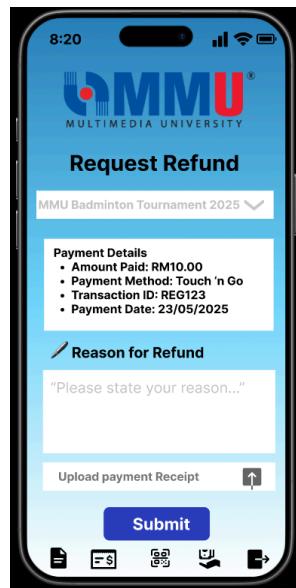
**Figure 3.8.2.8: Participated Event List Page (User)**

This is the participated event page for users of Campus Event Check-in System (CECS). In this page, it shows the event that has the status 'Paid' or 'Unpaid'. Users can choose the event to view more information. Click the 'Unparticipated' can directly view the participated event lists. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



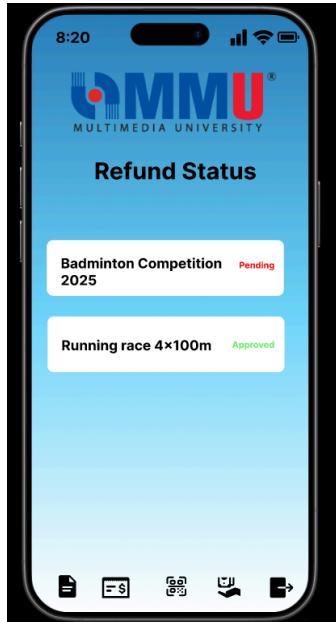
**Figure 3.8.2.9: Participated Event Description Page (User)**

This is the participated event description page for users of Campus Event Check-in System (CECS). In this page, it shows the event information and the buttons 'Refund' and 'Rating'. If users want to request a refund then click the 'Refund' button; If users want to rate the event then click the 'Rating' button. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



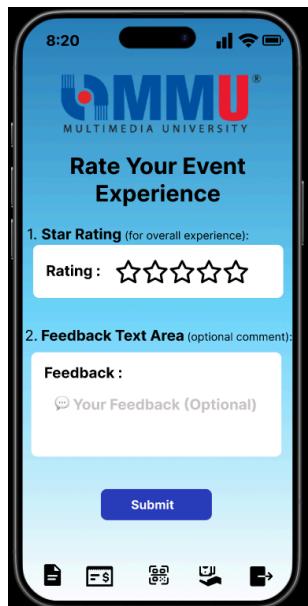
**Figure 3.8.2.10: Request Refund Page (User)**

This is the request refund page for users of Campus Event Check-in System (CECS). In this page, it shows the event title, payment details, the reason for refund input and requires users to upload their payment receipt. Users need to insert the reasons for refund and upload the payment receipt. Click 'Submit' to enter the next page. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



**Figure 3.8.2.11: Refund Status Page (User)**

This is the refund status page for users of Campus Event Check-in System (CECS). In this page, it shows the event that the status is ‘Pending’ or ‘Approved’. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



**Figure 3.8.2.12: Rating Page (User)**

This is the rating page for users of Campus Event Check-in System (CECS). In this page, it requires users to rate the stars regarding the experience and insert comment in the feedback area(optional). Click ‘Submit’ to enter the next page. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.

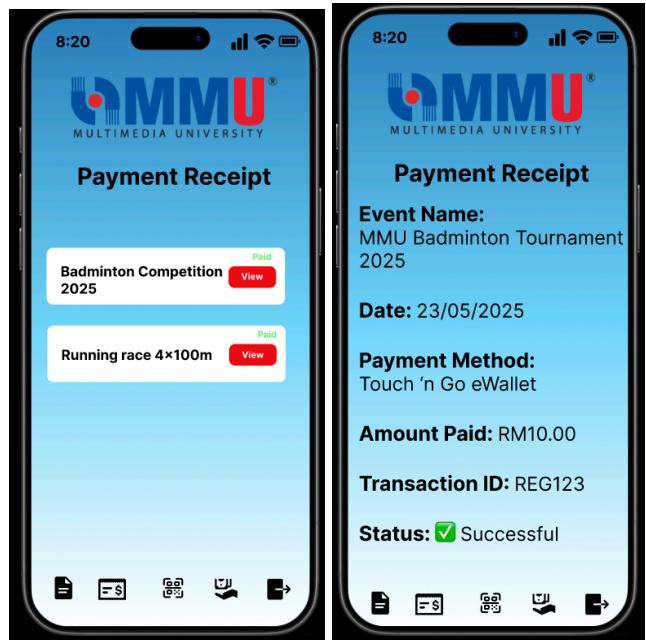


Figure 3.8.2.13: Payment Receipt Page (User)

This is the payment receipt page for users of Campus Event Check-in System (CECS). Users can click the icon below the page and it will move to the payment receipt page. Users can click the 'View' button and it will show the details of the receipt. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.

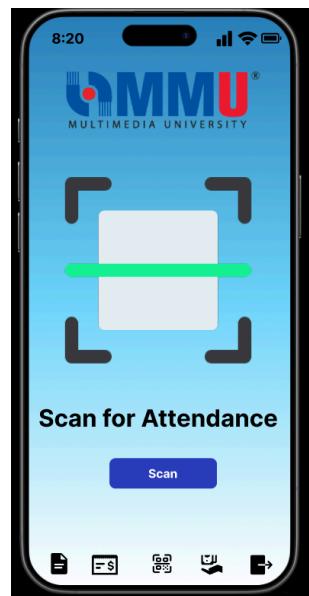


Figure 3.8.2.14: Scan Attendance Page (User)

This is the scan attendance page for users of Campus Event Check-in System (CECS). Users can click the icon in the middle to enter this page. Users can click the 'Scan' button to scan the QR Code. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.



Figure 3.8.2.15: Verification Page (User)

This is the verification page for users of Campus Event Check-in System (CECS). After users scan the attendance, they can fill in the name, student id and ticket id to check in the event. Users can click the 'Check-in' button to finish it. After finishing the verification, it will move to the next page that shows the check-in successfully. Users can choose the icon below that represents the function of event lists, payment receipt, scan QR code, refund status and logout.

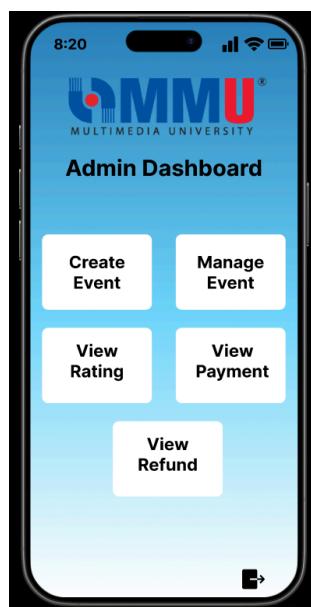
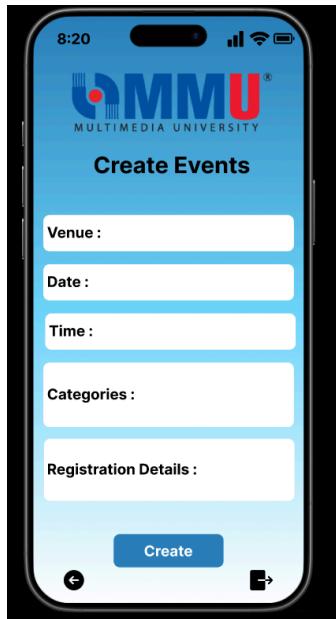


Figure 3.8.2.16: Dashboard Page (Admin)

This is the dashboard page for admin of Campus Event Check-in System (CECS). In this page, there are five functions to choose from which are 'Create Event', 'Manage Event', 'View Rating', 'View Payment', 'View Refund'. Admins can choose the icon to logout the system.



**Figure 3.8.2.17: Create Event Page (Admin)**

This is the create event page for admin of Campus Event Check-in System(CECS). In this page, there are venue input, date input, time input, categories input and registration details input that enable admins to enter to create the event. Admins can choose the icon below to back to the next page or logout the system.



**Figure 3.8.2.18: Manage Page (Admin)**

This is the manage event page for the admin of Campus Event Check-in System (CECS). In this page, there are event lists to choose to edit, delete and generate the QR Code.



Figure 3.8.2.19: Event QR Code Page (Admin)

This is the event QR Code page for admin of Campus Event Check-in System (CECS). In this page, there are the details of the event selected and the OR Code of that event. Click the 'Generate' button to generate the QR Code. Admins can choose the icon below that represents 'Back', 'Track attendance' and 'Logout'.

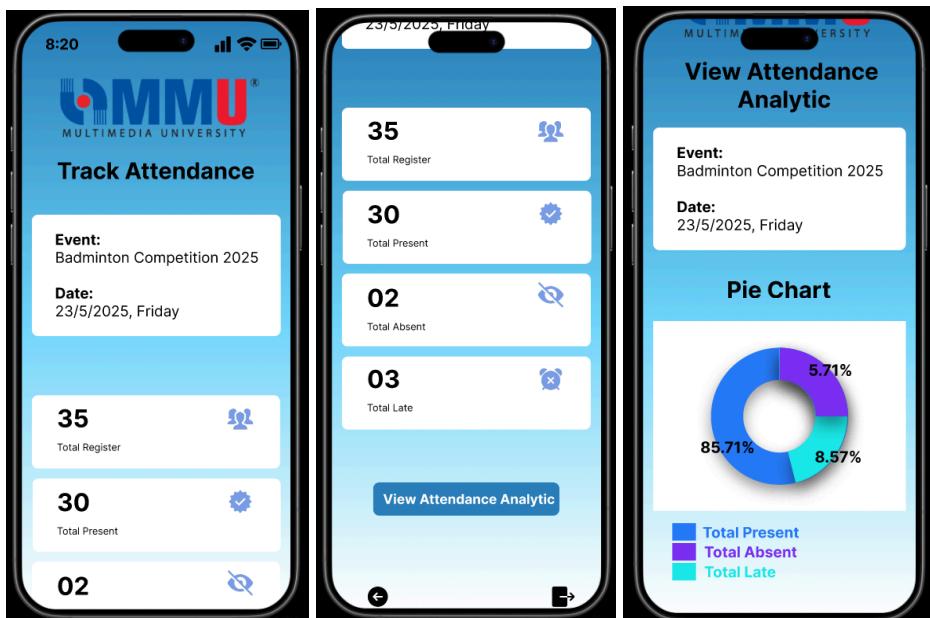


Figure 3.8.2.20: Track Attendance Page (Admin)

This is the track attendance page for admin of Campus Event Check-in System (CECS). In this page, there are the event title, event date, total register student, total present students, total absent students and total late students. Click the 'View Attendance Analytic' button to view the pie chart that shows the percentage of the students' attendance. Admins can choose the icon below that represents 'Back', 'Track attendance' and 'Logout'.



Figure 3.8.2.21: Event Rating Page (Admin)

This is the event rating page for admin of Campus Event Check-in System (CECS). In this page, admins choose the events to view and move to the next page. The next page will show the rating and the feedback of the selected event. Admins can choose the icon below that represents 'Back' and 'Logout'.



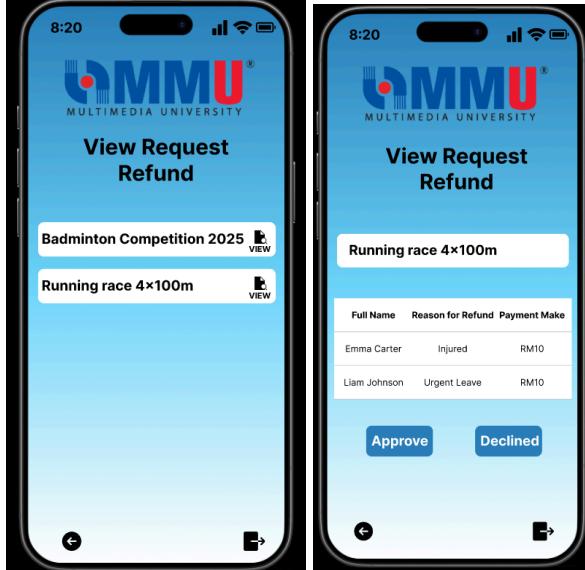
Figure 3.8.2.22: View Event Payment Page (Admin)

This is the view event payment page for the admin of Campus Event Check-in System (CECS). In this page, admins choose the events to view the payment details of each event. The next page will show the rating and the feedback of the selected event. Admins can choose the icon below that represents 'Back' and 'Logout'.



**Figure 3.8.2.23: View Payment Details Page (Admin)**

This is the view payment details page for admin of Campus Event Check-in System (CECS). In this page, there are the details (total register, total received payment and total request on-site payment) shown. Click the 'Generate Payment Report'. The next page will show the full name, payment method and the payment made by the students. Admins can choose the icon below that represents 'Back' and 'Logout'.



**Figure 3.8.2.24: View Event Request Refund Page (Admin)**

This is the view request refund page for the admin of Campus Event Check-in System (CECS). In this page, admins can choose the event to view the request refund. Admins can view the full name, reason for refund and payment made on the next page. Click the button 'Approve' or 'Declined' to decide the request. Admins can choose the icon below that represents 'Back' and 'Logout'.

### 3.9 Standard Compliance

This section outlines the standard compliance requirements that must be considered throughout the development of the **Campus Event Check-in System (CECS)**. These requirements ensure that the system conforms to relevant regulatory, security, and operational standards, supporting secure user data handling, payment integrity, and traceability of system actions.

**Table 3.9.1: Standard Compliance**

Requirement ID	Description	Priority	Author
REQ_SC_001	The system shall comply with the Personal Data Protection Act (PDPA) to ensure secure handling and storage of student and admin personal information.	High	TIANYOU
REQ_SC_002	All payment-related activities shall comply with PCI-DSS standards to ensure secure processing of financial transactions through integrated payment gateways.	High	ERIC
REQ_SC_003	The system shall generate audit trails for sensitive operations, such as login, payment, refund handling, and attendance changes, with timestamp, user ID, and action details.	High	YILING
REQ_SC_004	The system shall ensure compliance with ISO/IEC 27001 for managing sensitive data and maintaining information security controls throughout the mobile application.	Medium	KAISHEN
REQ_SC_005	The system shall follow standard naming conventions for event data, student records, and transaction logs to ensure consistency and interoperability with other university systems.	Medium	ERIC
REQ_SC_006	Any reporting features (e.g., attendance or financial summaries) shall follow university reporting formats for ease of audit and review by administrative departments.	Medium	TIANYOU

## 4.1 Verification

The verification of software system attributes ensures that the Campus Event Check-in System not only functions correctly but also meets essential quality standards such as availability, security, usability, and scalability. Each attribute has been mapped to a specific verification method, responsible party, timing, and test environment to ensure systematic evaluation.

To improve clarity and traceability, the attributes are classified into functional categories such as Availability, Security, Performance, Maintainability, and Usability. This structure enables focused verification across different quality dimensions.

### 4.1.1 Availability Requirements Verification

**Table 4.1.1.1: Availability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_001	Uptime monitoring	Developer / QA team	During active event dates	Firebase dashboard
REQ_SA_002	Real-time sync test (Firebase updates)	QA team	Registration and check-in periods	Firebase test environment

### 4.1.2 Reliability Requirements Verification

**Table 4.1.2.1: Reliability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_003	Normal + stress load testing	QA team	Pre-launch and post-deployment	Android/iOS devices
REQ_SA_004	Fault injection (e.g., network drop, retry logic)	QA team	During offline/retry testing	Simulated low network

### 4.1.3 Security Requirements Verification

**Table 4.1.3.1: Security Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_005	HTTPS inspection	Developer / QA	During API integration	Wireshark / browser inspection
REQ_SA_006	OAuth login test	QA team	At login screen	Linked to university auth system
REQ_SA_007	Role-based access control testing	QA team	Admin/restricted function execution	Admin panel

#### 4.1.4 Auditability Requirements Verification

**Table 4.1.4.1: Auditability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_008	Log inspection (timestamp, user ID, action)	Developer / Security team	After sensitive operations	Firebase Logs / Console Viewer

#### 4.1.5 Performance Requirements Verification

**Table 4.1.5.1: Performance Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_009	Stopwatch-based task timing	QA team	Login, scan, and list view	Mobile phones
REQ_SA_010	High concurrency test for QR check-in	Developer / QA	Event peak times	Firebase + simulation tools

#### 4.1.6 Maintainability Requirements Verification

**Table 4.1.6.1: Maintainability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_011	Code modularity review	Developer	During development	VS Code / GitHub
REQ_SA_012	Frontend code inspection (naming, reuse)	Tech lead / Developer	During sprint reviews	Codebase / CI pipeline

#### 4.1.7 Portability Requirements Verification

**Table 4.1.7.1: Portability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_013	Compatibility test on devices	QA team	During UAT phase	Android 8+ / iOS 12+
REQ_SA_014	Responsive layout test on screens	QA team	Post-UI design completion	Mobile / tablet screens

#### 4.1.8 Usability Requirements Verification

**Table 4.1.8.1: Usability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_015	Walkthrough with test users	UX / QA team	Pre-deployment	Figma / App prototype
REQ_SA_016	Action flow inspection ( $\leq 3$ steps for key tasks)	UX team	After workflow implementation	App navigation screens

#### 4.1.9 Responsiveness Requirements Verification

**Table 4.1.9.1: Responsiveness Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_017	Orientation/screen resizing tests	QA team	During UI testing	Real devices / browser preview

#### 4.1.10 Scalability Requirements Verification

**Table 4.1.10.1: Scalability Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_018	Load simulation with high concurrent users	QA team	University-wide events	Firebase emulator / JMeter
REQ_SA_019	Report generation performance with bulk data	Developer / QA	After report module implementation	Admin panel / Firebase backend

#### 4.1.11 Accuracy Requirements Verification

**Table 4.1.11.1: Accuracy Requirements Verification**

Requirement ID	Method	Responsibility	Event-Based Timing	Venue/Environment
REQ_SA_020	Cross-check of input/output (e.g., refunds, check-ins, payments)	QA team	Post-registration / transaction	Firebase DB / App interface

## 4.2 Verification Criteria

The following criteria define how the system's functionalities and performance will be verified. These criteria align with the functional, performance, and usability requirements to ensure the software meets the desired outcomes:

**Table 4.2.1: Verification Criteria**

Criteria ID	Requirement Description	Verification Method
REQ_VC_001	The system shall allow students and admins to log in within 3 seconds under normal load.	Conduct performance testing with up to 1,000 concurrent login attempts.
REQ_VC_002	Users shall be able to view and register for events successfully, and a confirmation notification should be sent within 1 second.	Execute functional tests for event registration and confirm notification delivery.
REQ_VC_003	Payments (online and on-site) shall be processed securely with a maximum delay of 5 seconds.	Perform end-to-end payment processing tests and verify transaction success logs.
REQ_VC_004	The system shall generate event-specific QR codes within 1 second of request.	Test QR code generation for multiple events under normal load conditions.
REQ_VC_005	Admins shall be able to view attendance in real time with updates occurring every 2 seconds.	Simulate a check-in process and verify attendance updates on the admin dashboard.
REQ_VC_006	The system shall maintain 99.9% uptime, excluding scheduled maintenance.	Monitor system logs over a 30-day period.
REQ_VC_007	All functionalities, including event registration and check-in, shall be accessible on mobile devices with no display or usability issues.	Conduct usability testing across various mobile devices and screen resolutions.
REQ_VC_008	The platform shall comply with WCAG 2.1 AA standards for accessibility, including screen reader support and keyboard navigation.	Perform accessibility testing with tools like Axe or WAVE.
REQ_VC_009	All database transactions shall have a latency of less than 100ms.	Conduct database performance testing under load conditions.

REQ_VC_010	All user data shall be encrypted during transmission with an encryption delay of no more than 50ms.	Perform penetration testing and verify encryption performance logs.
REQ_VC_011	Backups shall occur hourly, with a maximum data loss window of 30 minutes.	Verify scheduled backup operations and conduct recovery tests.

# 5 Appendices

## 5.1 Assumptions and Dependencies

This section outlines the assumptions and dependencies that influence the development and functionality of the Campus Event Check-in System (CECS).

### Assumptions:

1. The system assumes that the university's Authentication Server will remain stable and accessible throughout development and operation for verifying student and admin credentials.
2. It is assumed that students and admins have mobile devices capable of running the app (Android 8.0 / iOS 12 or later), with necessary features like a camera and stable internet access.
3. The Payment Server or gateway (e.g., FPX, TNG eWallet) used for processing online payments and refunds will be available and compliant with integration requirements.
4. It is assumed that the event-related data (e.g., event title, date, venue, fee) will be maintained and updated accurately by admins through the system.
5. Users (students and admins) are assumed to be familiar with basic mobile application navigation and willing to follow system instructions for actions like check-ins and feedback submission.
6. It is assumed that QR code scanning functionality will work correctly within the supported mobile environments and will not be restricted by device-level permissions or third-party limitations.

### Dependencies:

1. Integration with the university's Authentication Server is required to validate login credentials securely before allowing access to student/admin features.
2. The system depends on Payment Gateway APIs (e.g., FPX, TNG) for secure, real-time processing of event registration payments and refunds.
3. The app is dependent on Firebase Cloud Messaging (FCM) or equivalent notification services for sending alerts like registration confirmations.
4. The system depends on the availability and reliability of the university's backend database to manage user, event, attendance, and payment data consistently.
5. Successful implementation also relies on admin participation, including timely creation of events, QR code generation, and handling of refund approvals.
6. The project is dependent on the university IT team or system administrators for configuration, maintenance, and hosting of the backend services and servers.

## 5.2 Acronyms and Abbreviations

**Table 5.2.1: Acronyms and Abbreviations**

<b>Acronym / Abbreviation</b>	<b>Definition</b>
CMS	Campus Management System
SMS	Short Message Service
QR Code	Quick Response Code
GPA	Grade Point Average
MUET	Malaysian University English Test
ID	Identification Number
FPX	Financial Process Exchange (Malaysia's online payment gateway)
TNG	Touch 'n Go (Malaysia's digital payment platform)
PDF	Portable Document Format
UI	User Interface
DB	Database
CRUD	Create, Read, Update, Delete (standard database operations)
OTP	One-Time Password
API	Application Programming Interface
SMS Gateway	A service allowing systems to send SMS messages programmatically