# Chapter 1: Introduction

## Purpose

The purpose of the project is to develop a digital check-in platform for campus events that seamlessly integrates with the university's student identification database and payment system.  
The system aims to automate and streamline the process of event attendance tracking, ticket verification, and on-site transactions, ensuring an efficient, secure, and user-friendly experience for students and event organizers.

Integration of these components will enable real-time monitoring of attendance records and payments, reducing administrative burdens and enhancing overall event management capabilities within the campus environment.

## Scope

Campus Event Check-in System with Student ID and Payment Integration is created in order to streamline the process of campus event attendance tracking, ticket verification, and payment processing for students and event organizers.

Campus Event Check-in System shall facilitate the following operations:

1. Check-in to campus events using student ID verification.
2. Verify event tickets digitally.
3. Process on-site payments through integrated payment gateways.
4. Provide real-time attendance and transaction records to event organizers.

## Product Perspective

The Campus Event Check-in System is an integrated digital platform that operates as an extension of the university's event management and student information infrastructure. It interacts with the university's student database to authenticate users and with a payment gateway to process event-related transactions. The system supports two main user roles: students and event organizers. Students use the system to browse available events, register, make payments, and check in using QR code. Organizers create events, manage attendee data, and monitor attendance status.

### System Interface

The system interface requirements describe how the Campus Event Check-in System interacts with external systems such as the university's student database, payment gateway, event management modules, and notification services **Table ‎1.3.1.1**.

Table ‎1.3.1.1 System Interfaces Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | **Description** | **Priority** | **Author** |
| **REQ\_SI001** | The system shall integrate with the university student database to authenticate student identities and retrieve relevant information during registration and check-in. | High | Azhar |
| **REQ\_SI002** | The system shall connect to a third-party payment gateway to process event payments securely, providing real-time confirmation of transaction status. | High | Ainee |
| **REQ\_SI003** | The system shall interface with device cameras or scanners to scan student IDs or QR codes for efficient check-in and attendance tracking. | Medium | Alkatheri |
| **REQ\_SI004** | The system shall be accessible via the latest versions of Chrome and Firefox on Windows, Linux, and Mac to ensure broad compatibility and usability. | High | Yousef |
| **REQ\_SI005** | The system may optionally integrate with an email/SMS service to send notifications such as registration confirmations or event reminders to students. | Low | Yousef |

### User Interface

The user interface (UI) requirements define the visual and interactive characteristics of the Campus Event Check-in System. These specifications aim to ensure consistency, clarity, and accessibility for end users across various devices **Table ‎1.3.2.1.**

Table ‎1.3.2.1 User Interface Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface ID** | **Description** | **Priority** | **Author** |
| REQ\_UI001 | The GUI for the Campus Event Check-in System will use three background colors: White (RGB Hex: #FFFFFF) as the primary color, Navy Blue (RGB Hex: #001F3F) as the secondary color, and Emerald Green (RGB Hex: #2ECC71) for emphasis elements such as confirmation banners or successful check-in indicators. | High | Yousef |
| REQ\_UI002 | The system will use two main font colors: Black (RGB Hex: #000000) on light backgrounds and White (RGB Hex: #FFFFFF) on dark backgrounds to ensure readability and accessibility for all user types. | High | Yousef |
| REQ\_UI003 | The font family used will be "Poppins" for all headings and titles, and "Roboto" for body text across both desktop and mobile versions of the application, to ensure modern and clean visual presentation. | Medium | Yousef |
| REQ\_UI004 | The minimum base font size will be set to 16pt to ensure clarity on both mobile and desktop devices. Font resizing through accessibility settings will also be supported. | High | Yousef |

### Hardware Interface

The Campus Check-in application will be compatible with any desktop or mobile device with the following specifications **Table ‎1.3.3.1**.

Table ‎1.3.3.1 Required Hardware Interfaces

|  |  |  |  |
| --- | --- | --- | --- |
| **Interface ID** | **Description** | **Priority** | **Author** |
| REQ\_H1001 | The device must be equipped with a processor that is a 32-bit, ARM or x86, processor regardless of brand or manufacturer. | High | Azhar |
| REQ\_H1002 | The device must be equipped with at least 2GB of Random Access Memory (RAM). | High | Azhar |
| REQ\_H1003 | The device must include a secondary storage drive with a minimum of 1GB of free space. | High | Azhar |
| REQ\_H1004 | The device must be equipped with a network adapter that is compliant with the IEEE 802 family of standards. | High | Azhar |
| REQ\_H1005 | The device must be equipped with a display. | High | Azhar |

### Software Interface

The Campus Check-in application requires other software products to function properly. The interfaces between The Campus Check-in application and other software products are described in **Table ‎1.3.4.1**.

Table ‎1.3.4.1 Required Software Interfaces

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category** | **Name** | **Version Number** | **Purpose** | **Reference** |
| **Database** | phpMyAdmin | 5.2.1 | Used to create, store, modify, and delete event, user, and registration data. | phpMyAdmin official page |
| **Operating System** | Microsoft Windows | Windows XP or later | Runs browser-based application for admin and student users. | Chrome Browser system requirement |
| macOS | High Sierra 10.13 or later | Supports browser access on Apple devices. | macOS official site |
| Linux | Ubuntu 18.04+, Debian 10+, openSUSE 15.2+, Fedora 32+ | Runs the system via browser on Linux environments. | Linux distributions’ sites |
| Android | Android 7.0 Nougat or later | Accesses system via mobile browsers. | Android system requirements |
| iOS | iOS 14.0 or later | Accesses system via Safari or Chrome on iPhones/iPads. | iOS system requirements |
| **Browser** | Google Chrome | 113.0.5672.64 | Primary browser interface used to access the system. | Chrome official page |
| Microsoft Edge | 112.0.1722.48 | Alternative browser to access the system. | Microsoft Edge official page |
| Safari | 16.4.1 | For iOS/macOS users to access the system. | Safari official page |
| Opera | 105.0.4970.63 | Optional browser supported for accessing the platform. | Opera official page |
| **Screen Reader** | Speechify | 9.35.0 | Assists visually impaired users by reading on-screen content aloud. | Speechify official page |

### Communication Interface

**Table ‎1.3.5.1** shows an overview of the communication interfaces used in the Campus Event Check-in System. It lists each interface or protocol along with its purpose, type of communication, and the users involved.

Table ‎1.3.5.1 Communication Interface Requirements

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement ID** | **Interface/Protocol** | **Purpose** | **Type** | **Users** |
| REQ\_CI001 | HTTP/HTTPS | Allows secure access to the web-based system via modern browsers on any device. | System-to-User | Students, Event Organizers, Admin Staff |
| REQ\_CI002 | RESTful API (JSON) | Enables secure interaction between frontend and backend for registration, login, and payments. | System-to-System | System Frontend, System Backend |
| REQ\_CI003 | WebRTC | Supports real-time access to device cameras for QR code scanning during check-in. | System-to-Device | Students, Event Organizers |
| REQ\_CI004 | SMTP / Email API | Sends registration confirmations, payment receipts, and event reminders via email. | System-to-User | Students |
| REQ\_CI005 | TLS/SSL | Encrypts all communication between users and the system to ensure privacy and data integrity. | Security (System-to-User) | All System Users |
| REQ\_CI006 | TCP/IP | Provides reliable network connectivity across campus systems for accessing the application. | Local Network | Administrative Staff, IT Infrastructure |

### Product Functions

**Table ‎1.3.6.1** shows an overview of the core features to be implemented in the Campus Event Check-in System. Each feature is identified by a unique ID and includes a brief description along with the user roles that can access it.

Table ‎1.3.6.1 Campus Event Check-in System Function Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature ID** | **Feature** | **Description** | **Accessible Roles** |
| REQ\_F001 | Login | Allows users to securely log into the system. | Student, Event Organizer |
| REQ\_F002 | View Available Event | Displays a list of upcoming events open for registration. | Student |
| REQ\_F003 | View Event Detail | Provides detailed information about a selected event. | Student |
| REQ\_F004 | Register for Event | Enables students to register for a selected event. | Student |
| REQ\_F005 | Make Payment | Allows students to pay for event tickets. | Student |
| REQ\_F006 | Receive e-Ticket/QR Code | Generates a QR code for the student upon successful registration and payment. | Student (System) |
| REQ\_F007 | Show QR Attendance for Event | Displays the QR code used for scanning during check-in. | Student |
| REQ\_F008 | View Registration History | Shows a history of past registered events. | Student |
| REQ\_F009 | Create New Event | Allows organizers to create a new event. | Event Organizer |
| REQ\_F010 | Set Event Detail | Enables organizers to enter event name, date, time, and location. | Event Organizer |
| REQ\_F011 | View Events | Displays a list of events created by the organizer. | Event Organizer |
| REQ\_F012 | View Registrations with Attendance Status | Allows organizers to view who has registered and their attendance status. | Event Organizer |
| REQ\_F013 | Check-in Upon Arrival | Enables organizers to scan QR codes or student IDs to verify attendance. | Event Organizer |

### Use Case Diagram

A diagram of a person's process

AI-generated content may be incorrect.

Figure ‎1.3.7.1 Use Case Diagram

### ****User Characteristics****

This section describes the end users of the Campus Event Check-in System and their expected level of knowledge. These characteristics help determine how the system should be designed for usability and accessibility. **Table ‎1.3.8.1** summarizes the intended users and their required understanding of the system.

Table ‎1.3.8.1 Intended Users and Expected Knowledge

|  |  |  |
| --- | --- | --- |
| **Role** | **Description** | **Expected Knowledge** |
| **Student** | University students who will use the system to view events, register, make payments, and check in. | Basic understanding of web browsing, online form usage, and scanning QR codes. |
| **Event Organizer** | Authorized university staff or student committees managing events, registrations, and check-ins. | Familiarity with web forms, participant tracking, and scanning devices for check-in. |

### Limitations

Several limitations may affect the design, development, and usage of the system. These limitations must be considered to ensure that the system remains secure, scalable, and compliant.

* The system must comply with regional data protection laws to protect student privacy.
* Compatibility with assistive technologies (such as screen readers) is required for accessibility.
* Support for multiple languages may be needed in the future to cater to a diverse user base.
* The system must include secure login procedures and encryption for data in transit and at rest.
* User authentication must be robust to prevent unauthorized access.
* Measures must be in place to ensure data integrity and prevent corruption or loss.
* The system should be scalable to support growing numbers of users and data without performance loss.
* Key features must remain responsive, especially for time-sensitive operations like check-in.
* Regular system maintenance is required to address bugs, vulnerabilities, and browser compatibility.
* Compatibility issues may occur when integrated platforms are updated.
* Training resources and user documentation should be available to support users with limited technical expertise.

### Assumptions and Dependencies

The development and deployment of the system are based on the following assumptions and dependencies:

* The university will provide access to its student information database for identity verification.
* A secure and reliable internet connection is available at event locations.
* Third-party services like payment gateway APIs will remain stable and available.
* Users (students and organizers) will have access to compatible browsers and devices.
* The university will assign responsible personnel for ongoing maintenance and updates.
* External systems used for integration (student database, payment gateway) must maintain backward compatibility.

### Definitions

**Table 7** provides the definitions of terms, acronyms, and keywords used throughout this document.

|  |  |
| --- | --- |
| **Terms, Acronyms and Word** | **Definition** |
| Campus Event Check-in System | The proposed web-based application to manage student event participation, registration, and attendance. |
| Student | A university-registered individual who can register and check in to events. |
| Event Organizer | A staff or authorized user responsible for creating events and managing check-ins. |
| QR Code | A unique code generated after registration used to check in at the event venue. |
| Payment Gateway | A third-party service used to process online event payments securely. |
| Check-in | The process of confirming a student’s attendance using their ID or QR code. |
| REST API | An interface using HTTP methods for client-server communication in the system. |

# Chapter 2: References

# Chapter 2: Requirements

Requirements are the functionalities, behaviors, qualities, and constraints that the final system must fulfill. This chapter outlines the functional and non-functional requirements necessary for the successful implementation of the Campus Event Check-in System. Each requirement is described in detail, and diagrams are provided when necessary to illustrate system behavior, user interactions, and data flows. This section also includes any supporting information that aids in understanding or validating the system’s expected operations and performance.

## External Interfaces

This section details the external interface requirements of the Campus Event Check-in System. It outlines the actions available in the user interfaces and specifies the expected inputs and outputs for each interaction. These interfaces are designed to ensure a smooth and intuitive experience for users while supporting secure and efficient communication with external systems such as student databases, payment gateways, and scanning devices. The interfaces described here serve as the bridge between users and system functionalities.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_IO(\_ \_ \_ \_) | Version | 1.0 |
| Item |  | | |
| Item description |  | | |
| Item purpose |  | | |
| Input format |  | Valid Input |  |
| Related I/O |  | | |
| Author |  | | |