**1. Introduction**

Multimedia University (MMU) is a large institution with tens of thousands of students from both local and international backgrounds. Due to its vast student population, MMU frequently hosts a wide variety of events. With the huge number of events, our team has decided to develop an application that streamlines how students register for and attend campus events. Furthermore, help event organizers handle check-ins and payments.

Our application, “Tap&Go,” is a comprehensive Campus Event Management System designed to digitize and simplify event check-ins while integrating seamlessly with MMU’s student identification database and payment infrastructure. The system is developed to enhance coordination between event organizers and students by enabling efficient attendance tracking, ticket verification, and on-site transactions.

The platform is primarily divided into two core components. The first is the digital check-in system, which involves multiple stakeholders, including students, event organizers, and the university’s administrative departments. Through access to the student ID database, event organizers can verify student identities quickly and securely during check-in.

The second component focuses on payment integration. “Tap&Go” allows students to make cashless payments for tickets, merchandise, and food at campus events. Acting as a payment intermediary, the system accommodates a variety of digital wallets and banking apps commonly used in Malaysia. This feature ensures compatibility between students’ preferred payment platforms and the vendors' or organizers' accepted methods, by processing and routing transactions accordingly.

In addition, “Tap&Go” provides a centralized event portal where students can browse upcoming events, access event details, and purchase tickets directly within the app. By offering an all-in-one solution for event management, identity verification, and cashless payments, the system aims to deliver a smoother, more efficient experience for all campus stakeholders.

**1.1 Purpose**

The purpose of this document is to specify the details at the specific requirements elicitation by SER Requirements Engineering Team from SER Sdn.Bhd. It addresses the functionality, performance, interface, quality attributes and compliance requirements of Tap&Go system (TGS).

This document is intended for system developers, project managers, configuration managers and clients.

**1.2 Scope**

**Objective**

Implement a digital event check-in system that also allows students to make payments.

**Deadline**

23rd May 2025

**Deliverables**

Student authentication using the university ID system

QR code scanning for event check-ins

Real-time attendance tracking

In-app payment for tickets, food, or merchandise

Digital receipts and transaction history

**Tasks**

Improve event check-in efficiency

Increase student satisfaction

Enable cashless event operations

Centralize attendance data (Basically ticket holder when check-in store that data)

**1.3 Product Overview**

**1.3.1 Product Perspective**

The product is just a small module connected to a much larger university portal. It connects to the university portal through the student identification database. Using an existing student identification database stored by the university, the product can access it to authenticate users. The product is also able to integrate the university’s student identification database into its own database (Tickets).

A diagram of a data flow

AI-generated content may be incorrect.

**1.3.2 Product Functions**

|  |  |  |  |
| --- | --- | --- | --- |
| Actor | Use Case ID | Use Case | Description |
| Student | UC01 | Log In | Allow students to log into system |
| UC02 | Log Out | Allow students to log out of the system |
| UC03 | Check In | Allow students to check in of events |
| UC04 | View Campus Event | Allow students to view campus events |
| UC05 | View Tickets | Allow students to view their tickets |
| UC06 | Make Payment | Allow students to make payments using the application |
| UC07 | View Payment History | Allow students to view their payment history |
| UC08 | Book Tickets | Allow students to book tickets for events. |
| Event Organizer | UC04 | View Campus Event | Allow event organizers to view campus events |
| UC05 | View Tickets | Allow event organizers to view student’s tickets list |
| UC06 | Make Payment | Allow event organizers to make payments using the application |
| UC07 | View Payment History | Allow event organizers to view their payment history |
| UC09 | View Student List | Allow event organizers to view students list. |
| UC10 | Scan Ticket | Allow event organizers to scan tickets. |
| UC11 | Create Event | Allow event organizers to add events into the system |
| UC12 | Edit Event Details | Allow event organizers to edit event details. |
| UC13 | Delete Event | Allow event organizers to delete events. |
| System Admin | UC04 | View Campus Events | Allow system admins to view campus events |
| UC05 | View Tickets | Allow system admins to view tickets. |
| UC07 | View Payment History | Allow system admins to view payment history |
|  | UC09 | View Student List | Allow system admins to view student list. |
| University Administration | UC14 | Authenticate User | Allow university administration to authenticate users |
| UC04 | View Campus Event | Allow university administration to view campus events |
| UC09 | View Student List | Allow university administration to view students list. |
| Payment Gateway | UC07 | View Payment History | Allow Payment Gateway to have access to payment history. |
| UC15 | Verify Payment | Allow Payment Gateway to verify payment. |
| UC16 | Process Payment | Allow Payment Gateway to process payment. |

A diagram of a process

AI-generated content may be incorrect.

Student Activity Diagram

A diagram of a function

AI-generated content may be incorrect.

Event Organizer Activity Diagram

A diagram of a process

AI-generated content may be incorrect.

Payment Gateway Activity Diagram

A diagram of a flowchart

AI-generated content may be incorrect.

System Admin Activity Diagram

A diagram of a flowchart

AI-generated content may be incorrect.

University Administration Activity Diagram

**1.3.3 User Characteristics**

Students are expected to have a basic understanding of the application as to be able to use the application properly.

Event Organizers are expected to have a basic understanding of the application as to be able to use the application properly.

System Admins and University Administration needs to have a thorough understanding of the system as they will be the one managing the system.

**1.3.4 Limitations**

Certain functions are unavailable when used on a computer, especially the payment function since it uses a QR scanner. A computer usually does not have a QR scanner.

**1.4 Definitions**

Application: A set of software programs designed to perform a specific function for the user. In this context, it refers to the Tap&Go System that helps with event check-ins and on-site purchases.

**2 References**

The following documents are referred to in the text in such a way that some or all of their content constitutes the requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC/IEEE 29148:2018 Systems and software engineering — Life cycle processes Requirements engineering https://www.iso.org/obp/ui/en/#iso:std:iso-iec-ieee:29148:ed-2:v1:en

**3 Requirements**

**3.1 Functions**

This section details the functional requirements of TGS, starting with overall requirements followed by the requirements of each feature of the system.

A diagram of a network

AI-generated content may be incorrect.

**3.1.1 F001 Login**

The functional requirements for Start Up are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F101 | Version | 1.0 |
| Description | System shall start up when a user starts up the application | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F102 | Version | 1.0 |
| Description | System shall show login page when start up occurs. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC01 | | Version | 1.0 |
| Feature | F001 Login | | | |
| Purpose | To allow students to log into their accounts. | | | |
| Actor | Students | | | |
| Trigger | Students insert correct student id and password and press “login” button. | | | |
| Preconditions | TGS has started up | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students start up the application. | | |
|  | 2 | Students input their student id and password. | | |
|  | 3 | Students pressed on “Login” button. | | |
|  | 4 | Student id and password will be authenticated by University Administration. | | |
|  | 5 | If the student id and password are correct, students will be able to log into their account. | | |
| Alternate Flow | 5.1 | Student id and password are incorrect. | | |
|  | 5.2 | System display message “Student ID or Password is incorrect. Please try again.”. | | |
| Rules | Student id and password textbox cannot be left empty. | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.2 F002 Log Out**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F201 | Version | 1.0 |
| Description | System shall log users out when “Log Out” is pressed | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC02 | | Version | 1.0 |
| Feature | F002 Log Out | | | |
| Purpose | To allow students to log out of their accounts. | | | |
| Actor | Students | | | |
| Trigger | Students press “Log Out” button. | | | |
| Preconditions | Students are already logged into TGS. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press on “Log Out” button. | | |
| Alternate Flow | 1.1 | If log out fails, the system will show a message indicating so and redirects back to menu. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.3 F003 Check In**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F301 | Version | 1.0 |
| Description | The system allows students to check into events if tickets are purchased. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC03 | | Version | 1.0 |
| Feature | F003 Check In | | | |
| Purpose | To allow students to check into campus events. | | | |
| Actor | Students | | | |
| Trigger | Students press “Check In” button. | | | |
| Preconditions | Students are already logged into TGS.  Students already booked a ticket. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press “View Ticket” button. | | |
| 2 | System redirects to View Ticket page. | | |
| 3 | The system shows a list of booked tickets. | | |
| 4 | Students select tickets that they booked. | | |
| 5 | Students press “Check In” button. | | |
| 6 | System shows QR code for Event Organizers to scan. | | |
| 7 | If successfully scanned, Students will be checked in. | | |
| Alternate Flow | 3.1 | If no tickets are booked, the system will show a message indicate so. | | |
|  | 7.1 | If failed to scan, system will show a message indicating so any shows QR again. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.4 F004 View Campus Event**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F401 | Version | 1.0 |
| Description | System shall let users view past, current, and future campus events. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC04 | | Version | 1.0 |
| Feature | F004 View Campus Event | | | |
| Purpose | To allow students to view campus events. | | | |
| Actor | Students, Event Organizers, System Admin, University Administration | | | |
| Trigger | Students press “Campus Event” button. | | | |
| Preconditions | Students are already logged into TGS. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press on “Campus Event” button. | | |
| 2 | System redirects to Campus Event page. | | |
| 3 | The system shows a list of campus events. | | |
| Alternate Flow | 3.1 | If there are no campus event exists, the system will show a message indicating so | | |
|  | 3.2 | Users are able to sort by past, current, and future events. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a diagram

AI-generated content may be incorrect.

**3.1.5 F005 View Tickets**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F501 | Version | 1.0 |
| Description | The system should let users view their booked tickets. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC05 | | Version | 1.0 |
| Feature | F005 View Tickets | | | |
| Purpose | To allow students to view campus events. | | | |
| Actor | Students, Event Organizers, System Admin, University Administration | | | |
| Trigger | Students press “My Tickets” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Users press on “View Ticket” button. | | |
| 2 | System redirects to View ticket page. | | |
| 3 | The system shows a list of students booked tickets. | | |
| Alternate Flow | 3.1 | If no booked tickets exist, the system will show a message indicating so | | |
| 3.2 | If a student is accessing this function, they can only view their own tickets. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.6 F006 Make Payment**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F601 | Version | 1.0 |
| Description | System should let users make payments. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC06 | | Version | 1.0 |
| Feature | F006 Make Payment | | | |
| Purpose | To allow students to make payments on campus | | | |
| Actor | Students, Event Organizers | | | |
| Trigger | Students press “Pay” button. | | | |
| Preconditions | Students are already logged into TGS. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Users press on “Pay” button. | | |
| 2 | System redirects to Payment page. | | |
| 3 | The system will open a QR scanner. | | |
| 4 | User scans receiver’s QR | | |
| 5 | Once scanned, students can choose their bank of choice for payment. | | |
| 6 | Payment will then be verified and processed by the payment gateway. | | |
| 7 | Users will receive a digital receipt (record) for their payment. | | |
| Alternate Flow | 1.1 | Students make payment when booking tickets. (F008 Book Tickets). | | |
| 4.1 | If QR scanning fails, system will show a message indicating so. | | |
| 6.1 | If verification and processing fail, the system will show a message indicating so. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a payment method

AI-generated content may be incorrect.

**3.1.7 F007 View Payment History**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F701 | Version | 1.0 |
| Description | System should let users view payment history | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC07 | | Version | 1.0 |
| Feature | F007 View Payment History | | | |
| Purpose | To allow students to make payments on campus | | | |
| Actor | Students, Event Organizers, System Admins, University Administration | | | |
| Trigger | Users press “View Payment History” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press on “View Payment History” button. | | |
| 2 | System redirects to Payment History page. | | |
| 3 | The system shows all payment made by students. | | |
| Alternate Flow | 3.1 | For students, the system will only show their own payment history. | | |
| 3.2 | If no payments are made before, system will show a message indicating so. | | |
| Rules | Users must have access to TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.8 F008 Book Tickets**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F801 | Version | 1.0 |
| Description | System shall allow students to book campus event tickets | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC08 | | Version | 1.0 |
| Feature | F008 Book Tickets | | | |
| Purpose | To allow students to book tickets for campus events. | | | |
| Actor | Students | | | |
| Trigger | Students press “Book Ticket” button. | | | |
| Preconditions | Students are already logged into TGS. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press on “Campus Events” | | |
| 2 | System redirects to Campus Event page | | |
| 3 | The system shows a list of campus events. | | |
| 4 | Students view an event they are interested in. | | |
| 5 | The system redirects to that event details page. | | |
| 6 | Students press on “Book Ticket” button. | | |
| 7 | The system redirects to payment page. | | |
| 8 | Students can choose their preferred bank of choice. | | |
| 9 | Students will accept fund transfer from their bank of choice. | | |
| 10 | The payment gateway will verify and process the payment. | | |
| 11 | The system will save the ticket in the student’s ticket list. | | |
| Alternate Flow | 2.1 | If no campus exists, the system will show a message indicating so. | | |
| 10.1 | If the payment system fails to verify or process the payment, the payment will be denied and a message indicating so will be shown. | | |
| Rules | Users must be logged into TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a flowchart

AI-generated content may be incorrect.

**3.1.9 F009 View Student List**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F901 | Version | 1.0 |
| Description | System should let users view student list | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC09 | | Version | 1.0 |
| Feature | F009 View Student List. | | | |
| Purpose | To allow users to view student list. | | | |
| Actor | System Admins, University Administration | | | |
| Trigger | Users press “View Student List” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students press on “View Student List” button. | | |
| 2 | System redirects to Student List page. | | |
| 3 | The system shows a list of students that are using the application. | | |
| Alternate Flow | 3.1 | If no students exist, the system will show a message indicating so. | | |
| Rules | Users must have access to TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a student list

AI-generated content may be incorrect.

**3.1.10 F0010 Scan Ticket**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1001 | Version | 1.0 |
| Description | System should let users scan tickets | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC010 | | Version | 1.0 |
| Feature | F0010 Scan Tickets. | | | |
| Purpose | To allow users to scan tickets. | | | |
| Actor | Event Organizers. | | | |
| Trigger | Users press “Scan Ticket” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Event Organizers press on “Scan Ticket” button. | | |
| 2 | System redirects to a QR scanner. | | |
| 3 | When successfully scanned, the system will show a message indicating its success. | | |
| Alternate Flow | 3.1 | If scanned unsuccessfully, the system will show a message indicating its failure. | | |
| Rules | Users must have access to TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a scanning process

AI-generated content may be incorrect.

**3.1.11 F0011 Create Event**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1101 | Version | 1.0 |
| Description | System should let users create event | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC011 | | Version | 1.0 |
| Feature | F0011 Create Event | | | |
| Purpose | To allow users to create events | | | |
| Actor | Event Organizers. | | | |
| Trigger | Users press “Create Event” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Event Organizers press on “Create Event” button. | | |
| 2 | System redirects to create event page. | | |
| 3 | System will ask to input events such as event name, date, description, etc. | | |
|  | 4 | Event Organizers then press on “Create” button. | | |
|  | 5 | The system will then save the event in the database. | | |
| Alternate Flow | 5.1 | If event is not saved, the system will show a message indicating its failure. | | |
| Rules | Users must have access to TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a event

AI-generated content may be incorrect.

**3.2.12 F0012 Edit Event Details.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement ID** | REQ\_F1301 | **Version** | 1.0 |
| **Description** | System should let users edit event details. | | |
| **Author** | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID** | UC012 | | **Version** | 1.0 |
| **Feature** | F0012 Edit Event Details | | | |
| **Purpose** | To allow users to edit event details. | | | |
| **Actor** | Event Organizers | | | |
| **Trigger** | Users press “Edit” button. | | | |
| **Preconditions** | Users have access to the system. | | | |
| **Scenario Name** | **Step** | **Action** | | |
| **Main Flow** | 1 | Event Organizers press on “Campus Events” button. | | |
| 2 | System redirects to Campus Events page. | | |
| 3 | Event Organizers then select the event they wish to edit. | | |
|  | 4 | The system will redirect to the event details page. | | |
|  | 5 | Event Organizers then press on “Edit” button. | | |
|  | 6 | The system will redirect to the edit page. | | |
|  | 7 | Event Organizers will then input the new details and press save. | | |
|  | 8 | The system will then save the new event details in the database. | | |
| **Alternate Flow** | 2.1 | If no campus events exist, the system will show a message indicating so. | | |
| 8.1 | If event details are not saved, the system will show a message indicating its failure. | | |
| **Rules** | Users must have access to TGS | | | |
| **Author** | Bryan Leow Man Uu | | | |

A diagram of a flowchart

AI-generated content may be incorrect.

**3.1.13 F0013 Delete Event.**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1301 | Version | 1.0 |
| Description | System should let users delete events. | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC013 | | Version | 1.0 |
| Feature | F0013 Delete Events | | | |
| Purpose | To allow users to delete events. | | | |
| Actor | Event Organizers | | | |
| Trigger | Users press “Delete” button. | | | |
| Preconditions | Users have access to the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Event Organizers press on “Campus Events” button. | | |
| 2 | System redirects to Campus Events page. | | |
| 3 | Event Organizers then select the event they wish to delete. | | |
|  | 4 | The system will redirect to the event details page. | | |
|  | 5 | Event Organizers then press on “Delete” button. | | |
|  | 6 | The system will show a confirmation message. | | |
|  | 7 | Event Organizers will then press on “Yes” button. | | |
|  | 8 | The system will then delete the event from the database. | | |
| Alternate Flow | 2.1 | If no campus events exist, the system will show a message indicating so. | | |
| 8.1 | If event is not deleted, the system will show a message indicating its failure. | | |
| Rules | Users must have access to TGS | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.14 F0014 Authenticate User.**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1401 | Version | 1.0 |
| Description | System should let user authenticate users (students). | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC014 | | Version | 1.0 |
| Feature | F0014 Authenticate User | | | |
| Purpose | To allow users to authenticate users. | | | |
| Actor | University Administration | | | |
| Trigger | Every time a student logs into their account. | | | |
| Preconditions | Students must have an account in the system. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students log into their account. | | |
| 2 | The system will then take student id and password. | | |
| 3 | The system will compare the login information with the information stored in the database. | | |
|  | 4 | If information matches, students will be allowed access into the system | | |
| Alternate Flow | 4.1 | If login information does not match, the system will deny students access to the system and request them to try again. | | |
| Rules | Students must have an account in the system. | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a system

AI-generated content may be incorrect.

**3.1.15 F0015 Verify Payment**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1501 | Version | 1.0 |
| Description | Payment Gateway is able to verify payments | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC015 | | Version | 1.0 |
| Feature | F0015 Verify Payment | | | |
| Purpose | To allow payment gateways to verify payments | | | |
| Actor | Payment Gateway | | | |
| Trigger | Every time payment is made. | | | |
| Preconditions | User makes a payment. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Students make a payment. | | |
| 2 | Payment gateway will verify contents of the payment. | | |
| 3 | If verification is successful, payment gateway will proceed with the payment. | | |
| Alternate Flow | 3.1 | If verification failed, payment will be denied, and a message will be shown by the system. | | |
| Rules | Students must have an account in the system. | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of payment process

AI-generated content may be incorrect.

**3.1.16 F0016 Process Payment**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_F1601 | Version | 1.0 |
| Description | Payment Gateway is able to verify payments | | |
| Author | Bryan Leow Man Uu | | |

The following table shows the details of this feature.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Use Case ID | UC016 | | Version | 1.0 |
| Feature | F0016 Process Payment | | | |
| Purpose | To allow payment gateways to process payments | | | |
| Actor | Payment Gateway | | | |
| Trigger | Every time payment is made. | | | |
| Preconditions | Students make a payment. | | | |
| Scenario Name | Step | Action | | |
| Main Flow | 1 | Payment has gone through the verification stage | | |
| 2 | Payment will then go through processing stageF | | |
| 3 | Funds will then be transferred to one of the various bank accounts owned by the university. | | |
|  | 4 | Funds will then be transferred to bank accounts preferred by the vendor. (still university bank account) | | |
|  | 5 | Funds will finally be transferred to the vendor or event organizers | | |
| Alternate Flow | 3.1 | If payment processing fails, payment will be rejected and will not continue. A message will be shown by the system indicating so. | | |
| Rules | Students must have an account in the system. | | | |
| Author | Bryan Leow Man Uu | | | |

A diagram of a flowchart

AI-generated content may be incorrect.

**3.2 Performance Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_PR001 | The system shall process and confirm a check-in within **5 seconds** for **95%** of users during peak event times (up to 500 check-ins per hour). | High | Bryan Leow Man Uu |
| REQ\_PR002 | The system shall support at least **200 concurrent users** performing check-ins without system slowing down or failure. | High | Bryan Leow Man Uu |
| REQ\_PR003 | The system shall maintain **99.9% uptime** during scheduled event periods. | High | Bryan Leow Man Uu |
| REQ\_PR004 | The system shall maintain **99.9% uptime** for the payment system. | High | Bryan Leow Man Uu |

**3.3 Usability Requirements**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_UR001 | The system should provide a **user interface** that allows a first-time user to complete the check-in process **without prior training** in under **30 seconds**. | High | Bryan Leow Man Uu |
| REQ\_UR002 | The system shall provide **clear, actionable error messages** (e.g., “Student ID not recognized”) within **1 second** of an error occurring. | High | Bryan Leow Man Uu |
| REQ\_UR003 | The app shall maintain a **consistent layout and design** across all screens, following a defined style guide. | High | Bryan Leow Man Uu |

**3.4 Interface Requirements**

**3.4.1 System Interface**

**3.4.1.1 Login Interface**

The system will interact with the University Digital ID System for user authentication and identity verification during login.

The system will interact with external payment systems for ticket payments and on-campus purchases.

**3.4.2 User Interface**

**3.4.2.1 Login Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_1001 | Version | 1.0 |
| Item | Login Button (input) | | |
| Description | “Login” button on starting screen | | |
| Purpose | To allow users to log into their Tap&Go accounts | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O | REQ\_1002 (Student ID and Password must match the information stored in Student Information Database) | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_1002 | Version | 1.0 |
| Item | Student ID and Password (input) | | |
| Description | “Student ID and Password” textbox on starting screen | | |
| Purpose | To allow users to input their student id and password | | |
| Format | Textbox | Valid Range |  |
| Related I/O | None | | |
| Author | Bryan Leow Man Uu | | |

**3.4.2.2 Main Interface (Main Menu Screen / Dashboard)**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2001 | Version | 1.0 |
| Item | Log Out button (input) | | |
| Description | “Log Out” button on menu screen | | |
| Purpose | To allow users to log out of their Tap&Go accounts. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2002 | Version | 1.0 |
| Item | Check In button (input) | | |
| Description | “Check In” button on menu screen | | |
| Purpose | To allow users to check into events they booked tickets for. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2003 | Version | 1.0 |
| Item | Book Ticket button (input) | | |
| Description | “Book Ticket” button on menu screen | | |
| Purpose | To allow users to book tickets to events they are interested in | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2004 | Version | 1.0 |
| Item | Campus Events button (input) | | |
| Description | “Campus Events” button on menu screen | | |
| Purpose | To allow users to view past, current and future campus events. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2005 | Version | 1.0 |
| Item | My Tickets button (input) | | |
| Description | “My Tickets” button on menu screen | | |
| Purpose | To allow users to view tickets that they booked. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2006 | Version | 1.0 |
| Item | Pay button (input). | | |
| Description | “Pay” button on menu screen. | | |
| Purpose | To allow users to pay for on-site purchases on the campus. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_2007 | Version | 1.0 |
| Item | Payment History button (input). | | |
| Description | “Payment History” button on menu screen. | | |
| Purpose | To allow users to view their payment history. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.1.2.3 Campus Event List Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_3001 | Version | 1.0 |
| Item | More Details (input) | | |
| Description | “More Details” button | | |
| Purpose | To allow users to view more details about the event. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_3002 | Version | 1.0 |
| Item | Purchase Ticket (input). | | |
| Description | “Purchase Ticket” button. | | |
| Purpose | To allow users to purchase tickets for events they are interested in. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_3003 | Version | 1.0 |
| Item | Sort by Past, Current, Future (input). | | |
| Description | “Past”, “Current”, “Future” button. | | |
| Purpose | To allow users to view past, current, and future campus events. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.4.2.4 My Ticket Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_4001 | Version | 1.0 |
| Item | View Details (input). | | |
| Description | “View Details” button. | | |
| Purpose | To allow users to view details about the tickets they purchased. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_5001 | Version | 1.0 |
| Item | Ticket Status (input) | | |
| Description | “Ticket Status” button. | | |
| Purpose | To allow users to view the status of their tickets. (Checked In, cancelled, etc) | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.4.2.5 Payment Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_5001 | Version | 1.0 |
| Item | QR scanner (input) | | |
| Description | “Scan” button | | |
| Purpose | To allow users to scan QR codes for payment. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.4.2.6 Check in Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_6001 | Version | 1.0 |
| Item | Show QR (input) | | |
| Description | “Show QR” button. | | |
| Purpose | To allow users to show their ticket QR. | | |
| Format | Virtual Button | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.4.2.7 Payment History Interface**

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | REQ\_7001 | Version | 1.0 |
| Item | Payment Date sorting (input) | | |
| Description | “Sort by date” | | |
| Purpose | To allow users to view their payments with a range (days, weeks, months) | | |
| Format | Virtual Calendar | Valid Range |  |
| Related I/O |  | | |
| Author | Bryan Leow Man Uu | | |

**3.4.3 Hardware Interfaces**

The minimum system requirements for TGS’s system must be windows 7 and above or Mac 10.7 and above for computers.

The devices used must include a camera for QR scanning.

Minimum system requirements for TGS’s system for mobile devices shall be above android KitKat 4.4 or IOS 8 for apple devices.

**3.4.4 Software Interfaces**

The system will interact with a third-party payment gateway for transaction handling (e.g.,

Maybank, RHB Bank CIMB Bank, etc.).

The system will interact with Multimedia University for data integration (Student Information Database).

**3.4.5 Communication Interfaces**

The system will use HTTPS for secure communication between users and

server.

**3.5 Logical Database Requirements**

**A diagram of a payment terminal

AI-generated content may be incorrect.**

**3.6 Design Constraints**

The user interface must comply with the university’s branding guidelines.

**3.7 Software System Attributes**

The following section is used to specify attributes that are expected from TGS and its implementation.

**3.7.1 Security**

The security requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q001 | TGS shall have student credential validation via an encrypted connection to the university database | High | Lovin A/L K Balasubramaniam |
| REQ\_Q002 | TGS shall have secure payment processing by being PCI DSS complaint, use HTTPs and tokenization | Very High | Lovin A/L K Balasubramaniam |
| REQ\_Q003 | TGS shall have access split based on roles for students, admins and event organizers. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q004 | TGS shall have AES data encryption for personal and payment information. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q005 | TGS should have audit logging to ensure traceability for check-ins, refunds and admin logins. | High | Lovin A/L K Balasubramaniam |

**3.7.2 Accuracy**

The accuracy requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q006 | TGS shall have real time event data synchronization to prevent duplication and maintain data integrity. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q006 | TGS shall have accurate student identification via proper lookups from university database | Medium | Lovin A/L K Balasubramaniam |
| REQ\_Q007 | TGS shall have accurate payment tracking by linking payments with unique event IDs and user IDs. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q008 | TGS shall have accurate ticket availability. Availability will be updated every 10 minutes. | Medium | Lovin A/L K Balasubramaniam |

**3.7.3 Performance**

The performance requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q008 | TGS shall not have a response time of more than 2 seconds at any given time during check-ins. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q009 | TGS shall have a scalable backend to handle peak traffic loads during events to ensure smoothness. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q009 | TGS shall have proper load balancing and efficient coaching during high traffic sessions to ensure smoothness. | High | Lovin A/L K Balasubramaniam |

**3.7.4 Usability**

The usability requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q010 | TGS shall have simple and clean user interface for all roles of users | Medium | Lovin A/L K Balasubramaniam |
| REQ\_Q011 | TGS shall be accessible across multiple devices like laptops, tablets and smartphones. | Very High | Lovin A/L K Balasubramaniam |
| REQ\_Q012 | TGS should be functional on web browsers on both windows devices and macOS devices. | Very High | Lovin A/L K Balasubramaniam |

**3.7.5 Reliability**

The reliability requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q013 | TGS shall always maintain system availability of 99.9%. | Very High | Lovin A/L K Balasubramaniam |
| REQ\_Q014 | TGS shall implement retry payment/failure mechanism | High | Lovin A/L K Balasubramaniam |
| REQ\_Q015 | TGS shall implement a backup for all logs and history to ensure a recovery plan can be executed to ensure data protection in case of system failure or breach. | High | Lovin A/L K Balasubramaniam |

**3.7.6 Maintainability**

The maintainability requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q016 | TGS shall contain a modular code structure to ensure system updates are easily integrated. | Medium | Lovin A/L K Balasubramaniam |
| REQ\_Q017 | TGS shall provide event organizers with the ability to edit event details like adding a new event, edit event timing or change of venue directly onto the system without the need of a admin or a developer. | Medium | Lovin A/L K Balasubramaniam |
| REQ\_Q018 | TGS shall implement proper error and failure logging with Syslog using Splunk | Very High | Lovin A/L K Balasubramaniam |

**3.7.7 Interoperability**

The Interoperability requirements for TGS are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement ID | Description | Priority | Author |
| REQ\_Q019 | TGS shall include integration with university databases to access and authenticate student credentials with university student records. | High | Lovin A/L K Balasubramaniam |
| REQ\_Q020 | TGS shall include integration with an external payment gateway system, iPay88, to conduct payments. | Very High | Lovin A/L K Balasubramaniam |

**3.8 Supporting Information**

**4 Verification**

**4.1 Verification Approach**

|  |  |
| --- | --- |
| Approach | Description |
| Stress Test | TGS should be run in a virtual machine to test the limits on the maximum operating capacity to check on how good the system is at handling high traffic. Stress Testing also allows us to evaluate graceful degradation of TGS. |
| Transaction Test | Multiple transactions will be made before the application is released to the public. The purpose of this test is to validate the accuracy, completeness and integrity of transactions. It also allows us to measure the response time of the transactions. |
| Black Box Test | All requirements and specifications are tested to determine the inputs of outputs of TGS. Developers will test the system without focusing on its internal logic or code. |
| White Box Test | The test will mainly focus on control structures, logical paths, functions, data flow and exception handling. Developers will test the system based on the design documents (data flow diagrams, state diagrams, etc.). |

**4.2 Verification Criteria**

* TGS should be able to handle 5,000 concurrent users.
* Payments made using TGS should be processed without issues 99.9% of the time
* TGS can handle heavy load during peak hours (Does not crash).
* TGS can always receive new data from Student Identification Database whenever there is an update.
* All functions of TGS are usable and work as intended.

**5 Appendices**

**5.1 Assumptions and Dependencies**

The system depends on the availability of the university's student database for user authentication.

System Admin should have full access to the system.

Every Students should have a Student ID.

**5.2 Acronyms and Abbreviations**

|  |  |
| --- | --- |
| TGS | Tap&Go System |
| TG | Tap&Go |
|  |  |
|  |  |
|  |  |
|  |  |