Software Requirements Specification for Software Engineering: A Platform for Event Management

Team 4, EvENGage
Virochaan Ravichandran Gowri
Omar Al-Asfar
Rayyan Suhail
Ibrahim Quraishi
Mohammad Mahdi Mahboob

October 10, 2025

Contents

Revision History

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 Purpose of the Project

The purpose of this project is to create a centralized platform for the McMaster Engineering Society (MES) that simplifies how large-scale events are managed. By consolidating event registration, form creation, attendee tracking, and data analytics into a single system, the platform aims to reduce administrative workload, eliminate fragmented workflows, and enhance the overall experience for both event organizers and attendees.

1.1 User Business

The primary users of this system are members of the McMaster Engineering Society, including executives and event organizers, as well as McMaster engineering students who attend MES events.

- **Administrators (MES Executives and Organizers):** Responsible for creating events, designing registration and feedback forms, managing attendee information, and analyzing data through the dashboard. They will use the platform to reduce manual work and improve event coordination.
- **End Users (Students and Attendees):** Students will use the platform to register for events, complete waivers or feedback forms, and receive event confirmations and updates. The goal is to create a seamless and engaging experience that encourages participation.

By serving both groups through a single platform, the system eliminates redundant workflows and ensures continuity of data across multiple events.

1.2 Goals of the Project

The primary goals of this project are as follows:

1. **Develop a Custom Form Builder:** Build an intuitive tool that lets administrators design and manage event or feedback forms with different question types and conditional logic, without needing third-party software.

2. **Implement a Unified Registration and Feedback System:** Create a simple, centralized process for students to register for events, complete waivers, and share feedback—all in one place.

3. **Create an Attendee Overview Dashboard:** Give administrators a clear view of event participation by displaying key details such as registration status, payments, and waiver completion in real time.

4. **Integrate Backend Analytics:** Include tools that help MES organizers analyze event data, track trends, and use insights to plan and improve future events more effectively.

2 Stakeholders

2.1 Client

The client for this project is the supervisor, Luke Schuurman. He is a member of the MES and has first-hand experience planning and hosting events with the MES. As the supervisor he will play a crucial role by ensuring the project aligns with objective of the MES and integrate the platform seamlessly with existing systems. He will also provide us with feedback and guidance throughout the project and will help define the project requirements in this document.

2.2 Hands-On Users of the Project

MES Executives and Council Members: They will be utilizing this system to create and manage events, configure forms and surveys, monitor event data and generate data analytics reports. They can be characterized as primarily undergraduate students who value their time greatly. They aim to reduce the time taken to do administrative tasks as well as provide a better experience during MES events. Their experience with systems like this can range from Journeyman - Master as they can be experienced in event planning and student engagement. There may be a slight learning curve to utilizing the new technology, but these users have experience performing these functions.

McMaster Engineering Students: They will be utilizing this app to

register for events, purchase tickets, sign waivers, check in at venues, and complete feedback surveys. They want to enjoy their university experience and connect with other students, They are also very busy and value their time greatly so are looking for an intuitive and straightforward user experience. Their general experience with systems like this is Journeyman as they may have used similar systems for other use cases.

Other Students and Guest Event Attendees: This group includes non-engineering students, alumni, and invited guests who participate in large-scale MES events such as the Fireball Formal and Graduation Formal, which extend beyond the core engineering community. It could also include students and guests from other universities as well as industry which attend the engineering conferences that the MES helps host. They are generally looking for an easy and seamless experience registering and attending these events. Their general experience with systems like this is Journeyman as they may have used similar systems for other use cases.

2.3 Personas

1. **Matthew Cruise (Engineering Student)**: Matthew is a 20-year-old second-year Mechanical Engineering student at McMaster University. He lives in a shared house near campus with two close friends. He enjoys attending MES events like pub nights and the Fireball Formal as a way to balance his heavy academic workload with some enjoyment and entertainment. He usually hears about events through word-of-mouth or social media and would prefer an easy and effortless way to find and register for events. Matthew is generally tech-savvy and is comfortable using online platforms but doesn't want to be bothered by too many notifications and forms. He has many ideas on how he can generally improve his university experience but doesn't believe he has an outlet to convey them. For Matthew his priority is convenience as he wants to enjoy himself but doesn't want to spend too much time or effort doing so.

2. **Adam Clooney (MES Executive)**: Adam Clooney is a 22-year-old final-year Civil Engineering student who currently serves as VP Social on the MES. He is responsible for coordinating large-scale events like Fireball Formal, working closely with other council members to handle

logistics, advertising, and student engagement. Adam is outgoing and enjoys bringing people together, but often feels the strain of balancing his role with academic responsibilities. He is proficient with common digital tools such as Google Drive, spreadsheets, and design platforms for promotions, but he's not highly technical. Adam appreciates structure and tools that keep things organized because he dislikes wasting time fixing errors or repeating work. He is motivated by the sense of accomplishment that comes from hosting a successful event and wants tools that help him stay on top of details.

3. **Margot Watson (McMaster Student)**: Margot Watson is a 21-year-old undergraduate student in Political Science at McMaster University. Although she is not part of the engineering faculty, she often attends large MES-hosted events such as the Fireball Formal and other socials because many of her friends are in engineering. Margot lives in an off-campus apartment with two roommates and enjoys being involved in student life across faculties. She has a relaxed attitude toward technology since she uses her phone daily for social media and messaging, but she prefers things to be straightforward and intuitive. She is cautious with her money, balancing tuition and living expenses, but she's willing to spend on experiences with friends. She is motivated by spending time with her friends and having good experiences to have a fulfilling student life.

2.4 Priorities Assigned to Users

Key Users: Luke Schuurman, MES Council Members and Executives, McMaster Engineering Students.
Secondary Users: Other Students and Guest Event Attendees.

2.5 User Participation

Our requirements will primarily be derived through meetings with our supervisor Luke Schuurman. If we have the need to clarify or elicit more requirements we will look to engage other MES members either through our supervisor or directly.

3 Mandated Constraints

3.1 Solution Constraints

Insert your content here.

3.2 Implementation Environment of the Current System

Insert your content here.

3.3 Partner or Collaborative Applications

Insert your content here.

3.4 Off-the-Shelf Software

Insert your content here.

3.5 Anticipated Workplace Environment

Insert your content here.

3.6 Schedule Constraints

Insert your content here.

3.7 Budget Constraints

Insert your content here.

3.8 Enterprise Constraints

Insert your content here.

4 Naming Conventions and Terminology

4.1 Glossary of All Terms, Including Acronyms, Used by Stakeholders involved in the Project

admin A privileged system user affiliated with MES who can access survey, event, and attendee records; track finances; and update event statuses. plural.

attendee An individual who attends an MES event, and uses the system to register, receive event updates, provide feedback, or fill out survey responses. plural.

CFES Canadian Federation of Engineering Students.

MES McMaster Engineering Society.

5 Relevant Facts And Assumptions

5.1 Relevant Facts

- The McMaster Engineering Society currently manages events such as Fireball Formal and the CALE Conference using several disconnected tools like Google Forms and Google Sheets, which creates inefficiencies and limits data analysis.
- Each MES event involves registration, waiver collection, and feedback surveys that must comply with university data privacy and accessibility standards.
- The system will be used by both administrators and students on a wide range of devices and browsers, so responsive and cross-platform design is required.
- The overall Event Management System is being co-developed by three Capstone teams. Project A must maintain compatibility and shared data models with the other two teams' components.

- The project's frontend stack was recently updated from **Next.js** to **Vite + TanStack Router + TanStack Create** to improve development speed, hot reloading performance, and flexibility for building a Single Page Application (SPA) with integrated admin dashboards and authentication.
- PostgreSQL remains the chosen database technology for data storage and analytics.

5.2 Business Rules

- Only verified MES administrators can create, edit, or delete forms and view backend analytics.
- Event data, form responses, and attendee details must be securely stored and retrievable for future event reviews.
- All user input will be validated to prevent incomplete or invalid submissions before being stored in the database.
- A consistent data format must be maintained across all events to support accurate analytics and report generation.
- Administrators can export data such as attendee lists or analytics summaries in standard formats (e.g., CSV, PDF).

5.3 Assumptions

- The MES will provide sample event and registration data to support development and testing.
- Users will have stable internet access when using the platform, as it relies on real-time connectivity.
- Collaborating teams (Projects B and C) will follow the agreed API and database specifications to ensure system integration works smoothly.
- Authentication and authorization will be handled securely within the shared platform using standardized methods.

- Future MES administrators and developers will continue maintaining the project using the provided GitHub repository, documentation, and CI/CD workflows.

6 The Scope of the Work

The purpose of the section is to define the expected scope of the project, including specifications on how the system is partitioned, business data models, and business use cases.

6.1 The Current Situation

Currently, the MES uses plethora of different platforms to host events and conduct surveys such as tools for registration, signing of waivers, and checking in attendees. The main platforms used are a combination of Google Forms and Google Sheets. There are several pain points to address with the current system.

- **Decentralized System Components** Registration and surveying is split across a plethora of tools and software which may not be compatible with each other. For example, Google Forms is used for registration forms and surveys, Google Sheets stores form data and LinkTree holds links to all the registration forms, and event updates is done by email or social media.
- **Overly Complex Form Logic** The current Canadian Federation of Engineering Students (CFES) survey consists of 70 pages of questions linked together through complex branching logic. The Google Forms form building UI makes this very complicated as all form elements are displayed as a linear list of sections making it very hard to track paths through the form.
- **Disorganized Data Visualization** Form response data is currently stored using Google Sheets. While Google Form data is easily imported into Google Sheets, any analytics on the data must be done manually through equations and macros.

- **Lack of Reusability** Google Forms comes with a few templates that provide an initial starting point for many types of forms such as registrant information. However, after the first section of the form, each subsequent section must be made manually. Sections may be imported from other forms, but this requires the user to have access to a form with the wanted section and to scour through an unorganized list of forms and sections.

- **Low Response Rates on Surveys** The response rates on the annual CFES survey of undergraduate engineering students have been decreasing due to the long length of the Google Form and the lack of ability to submit a partially completed form.

- **Manual Registration Scheduling** Event registration is managed through a combination of Instagram, Google Forms, and LinkTree. Events are advertised on Instagram, and a link to the signup Google Form is posted on the MES LinkTree. This solution lacks automation since a new Google Form must be made for every event and links have to be manually added and removed for each form when registration is opened or closed.

6.2 The Context of the Work

This section provides an overview of the high-level inputs and outputs between the system and external systems or actors. The system is split into two perspectives, the attendee perspective shown in Figure 1, and the admin perspective shown in Figure 2.

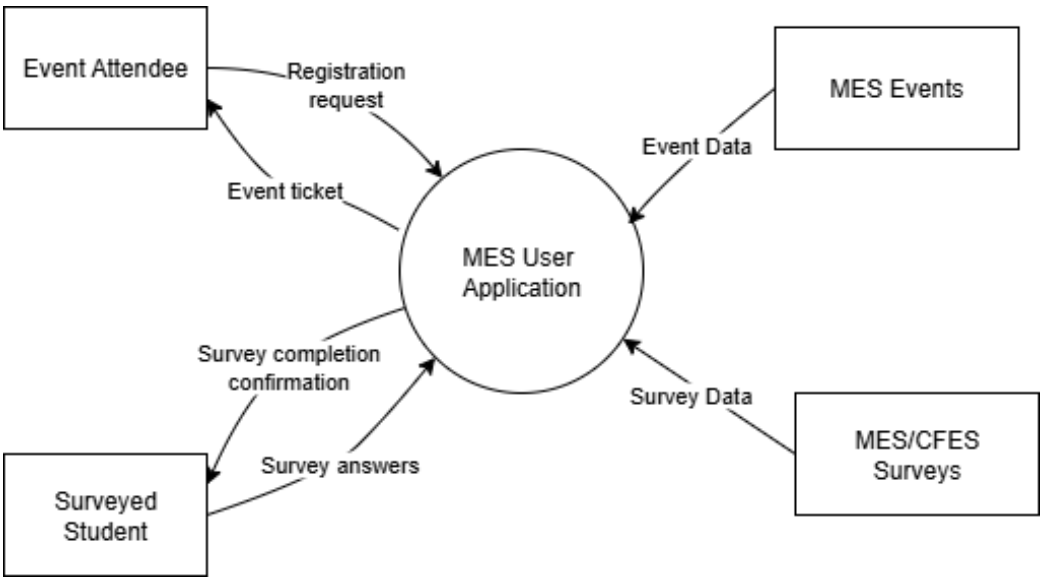

Figure 1: Work context diagram of the attendee side application

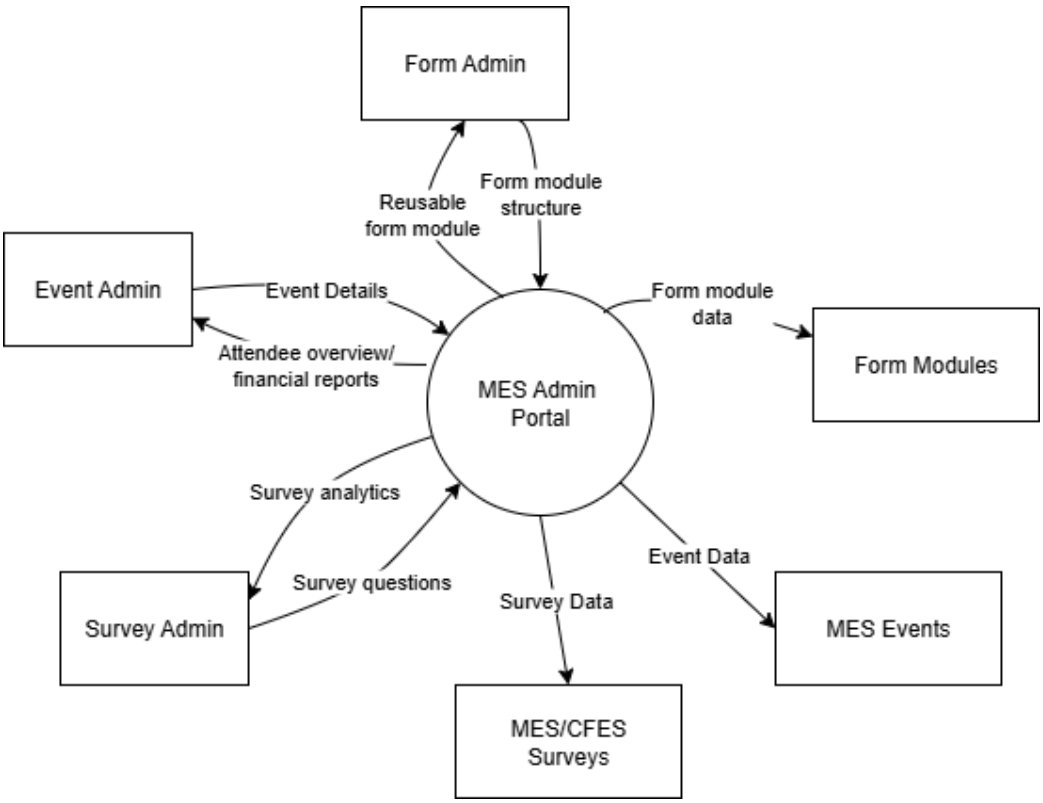

Figure 2: Work context diagram of the admin side application

6.3 Work Partitioning

This section describes how the work done by the system can be partitioned into smaller and more manageable workflows. Below is a table describing each of the sub-workflows of the proposed systems. Note that event creation and survey creation have been grouped as a single workflow since they are nearly identical.

No.	Event	Input	Output	Requirements
1	Attendee registers for event	Registration data	Event ticket	
2	Attendee fills out a survey	Survey responses	Survey completion confirmation	
3	Admin creates a form module	Module question data	Reusable form module	
4	Admin creates an event/survey	Event/survey data	Event/survey creation confirmation	
5	Admin views survey/event statistics	Event/survey to view	Event/survey registration/response reports	

Table 1: Partitioning of system workflows

6.4 Specifying a Business Use Case (BUC)

BUC 1: User registers for an event

Input: Registration data
Output: Event tickets
Pre-condition: User has downloaded the application and has created an account
Scenario:

1. The user application receives the registration data
2. The user application verifies the data is filled correctly
3. The user application sends the registration data to the backend server
4. The backend server verifies the event is not full, and the deadline has not passed
5. The backend server adds the user to the list of attendees

6. The backend server generates an event ticket and sends it to the user application
7. The user application confirms with the user that the registration was successful and presents the user with the ticket

Sub variations:

2a. The submitted registration data has errors: the user application prompts the user to fix the errors and resubmit.

4a. The event is full, or the deadline has passed: the backend sends an error code to the user application.

4b. The user application alerts the user of the error.

BUC 2: User fills out a survey module

Input: Survey responses
Output: Survey completion confirmation
Pre-condition: Attendee has downloaded the application and has created an account
Scenario:

1. The user application receives the survey data
2. The user application verifies the data is filled correctly
3. The user application sends the survey data to the backend server
4. The backend server updates the survey database with the attendee's data
5. The backend server sends a confirmation message to the user application
6. The user application confirms with the attendee that the survey data has been submitted

Sub variations:

2a. The submitted data has errors (i.e. mandatory fields not filled), the user application prompts the attendee to fix the errors and resubmit

BUC 3: Admin creates a form module

Input: Form fields
Output: Reusable form module
Pre-condition: Admin has access to create custom form modules
Scenario:

1. The admin portal receives the list of form fields and questions from the admin user for the custom module
2. The admin portal verifies the custom module has been created correctly
3. The admin portal sends the custom module data to the backend server
4. The backend server authenticates the admin user
5. The backend server saves the custom module data to the template database
6. The backend server sends a confirmation message to the admin portal
7. The admin portal updates the list of custom modules with the completed module
8. The admin portal confirms with the admin user that the custom module has been created

Sub variations:
2a. The submitted form module has errors (i.e. unfinished fields), the admin user is prompted to fix these errors before resubmitting
4a. Authentication of the admin fails, the admin portal is notified of the request denial

BUC 4: Admin creates an event/survey

Input: Event details
Output: Event creation confirmation
Pre-condition: Admin has access to create events
Scenario:

1. The admin portal receives the event/survey details
2. The admin portal verifies the event/survey details are correct
3. The admin portal sends the event/survey data to the backend server
4. The backend server authenticates the admin user
5. The backend server saves the event/survey data to the database of events/surveys
6. The backend server sends a message to the user application to notify users of the new event/survey
7. The backend server sends a confirmation message to the admin portal
8. The admin portal adds the created event/survey to the event/survey dashboard
9. The admin portal confirms with the admin user that the event/survey has been created

Sub variations:
2a. The submitted details have errors (i.e. event date has already passed), the admin user is prompted to fix these errors before resubmitting
4a. Authentication of the admin fails, the admin portal is notified of the request denial

BUC 5: Admin views event/survey statistics

Input: Event/survey to view
Output: Event registration/survey response report
Pre-condition: Event/survey has been created, and attendees have registered/responded
Scenario:

1. The admin portal receives the request to view event/survey statistics
2. The admin portal sends a request to the backend server containing the identification for the event/survey to view

3. The backend server receives the request and authenticates the admin
4. The backend server retrieves the registrant/response data for the requested event/survey from the database
5. The backend server generates a statistical report of all the data
6. The data is sent back to the admin portal
7. The admin portal formats all the data into a readable format
8. The admin is presented with the event/survey statistics

Sub variations:

3a. Authentication of the admin fails, the admin portal is notified of the request denial

7 Business Data Model and Data Dictionary

The purpose of this section is to illustrate the flow of data throughout the system.

7.1 Business Data Model

Figure 3 splits the system into into well defined subsystems and outlines the interactions between subsystems.

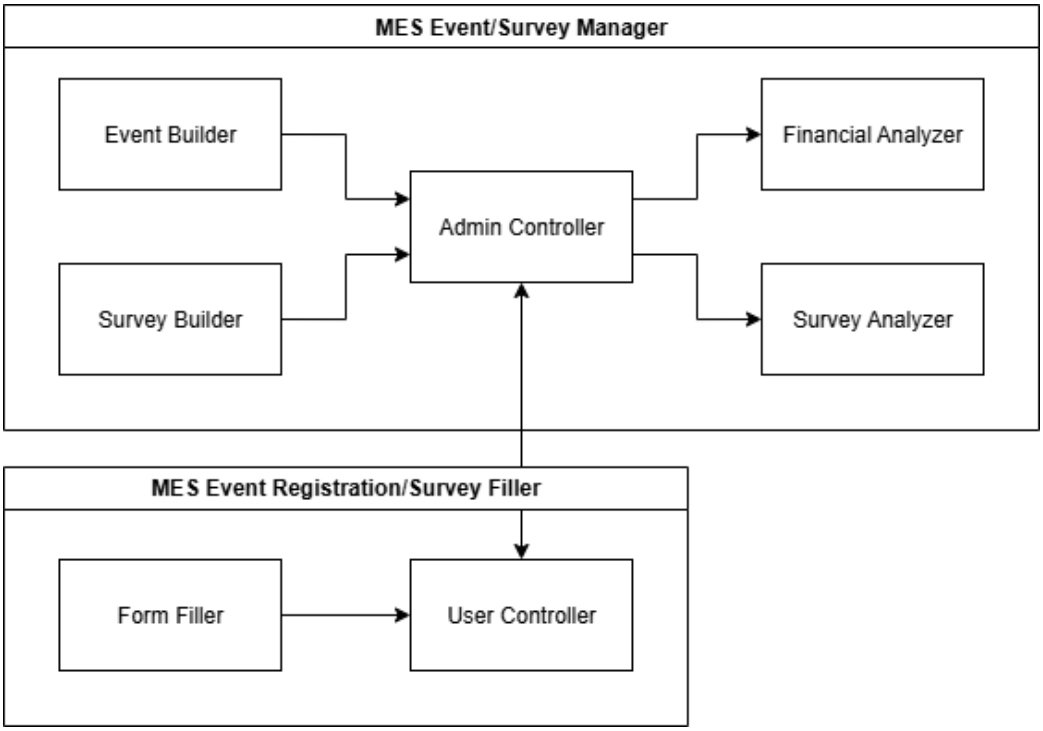

Figure 3: Diagram of the interactions between main system components

7.2 Data Dictionary

Table 2 defines each of the system components shown in Figure 3.

Component Name	Description	Type
MES Event/Survey Manager	The application handling the admin side of the system, including event/survey creation, and data analytics	Application
Event Builder	Contains tools for creating, publishing, and managing events	Module
Survey Builder	Contains tools for creating, publishing, and managing surveys	Module
Admin Controller	Handles the flow of execution of MES Event/Survey Manager modules and data flow between the event/survey builders, data analyzers, and the user controller	Module
Financial Analyzer	Automates analysis and visualization of finances from event attendee data	Module
Survey Analyzer	Automates analysis and visualization of survey response data	Module
MES Event Registration/Survey Filler	The application handling the attendee side of the system, including event registration and survey responses	Application
Form Filler	Handles input and validation of form responses	Module
User Controller	Controls the flow of execution of the form filler and handles communication with the admin controller	Module

Table 2: Data dictionary table

8 The Scope of the Product

The purpose of this section is to define the scope of the product by defining the boundaries of each major component of the system as well as their functionalities, interfaces for external actors, and interactions with each other.

8.1 Product Boundary

Figure 4 outlines the main components of the system, how they interact with each other, and their related use cases.

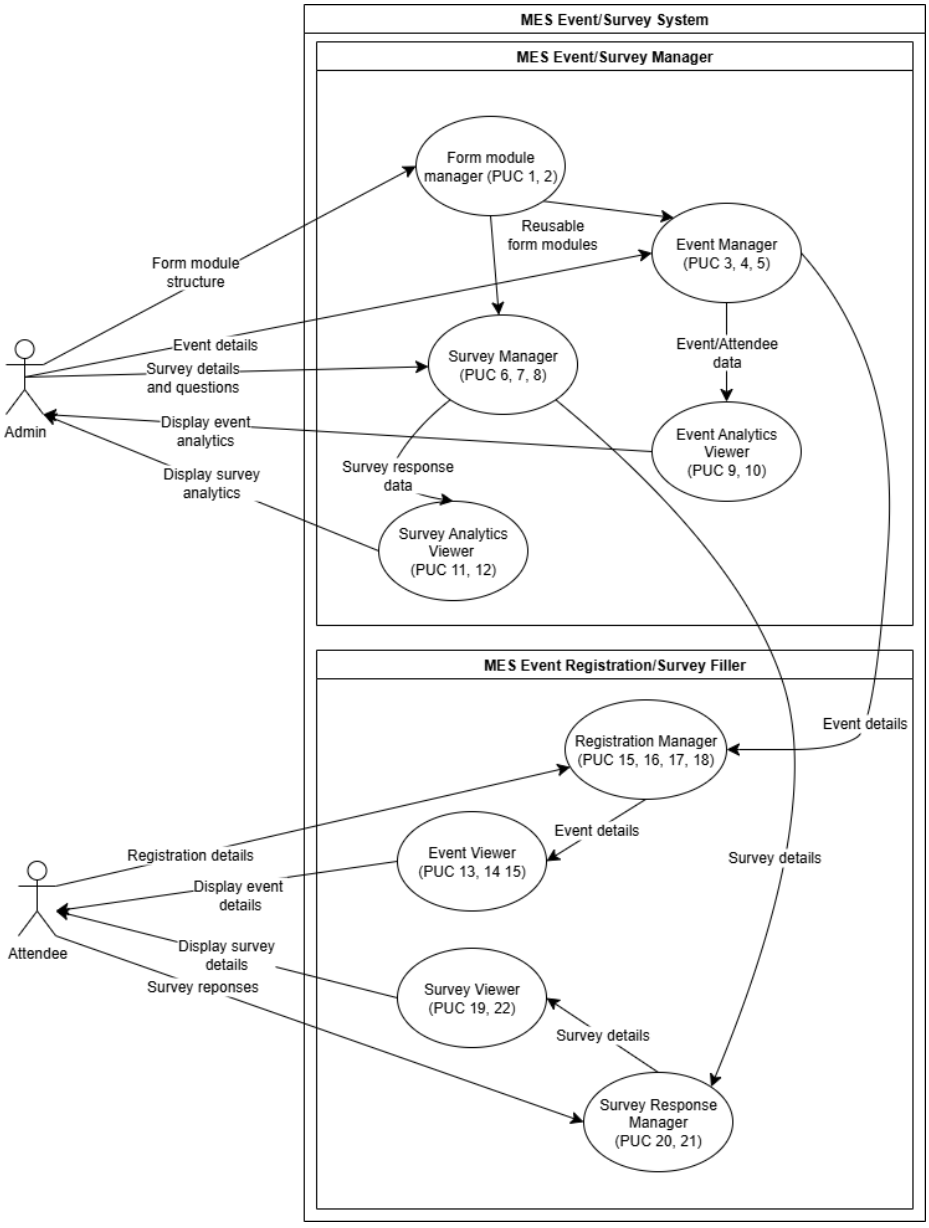

Figure 4: Product use case diagram

8.2 Product Use Case Table

Table 3 further defines each of the individual use cases of the system, outlining the specific inputs and outputs and related requirements.

#	Name	Actor	Input/Output	Requirements
1	Create a custom form module	Admin	Form module name and fields (in) Reusable form module (out)	
2	Edit an existing module	Admin	Selected form module and updated fields (in) Edited module (out)	
3	Create an event	Admin	Event details (in) Scheduled event (out)	
4	Edit an event	Admin	Selected event and updated details (in) Edited event (out)	
5	Cancel an event	Admin	Selected event (in) Cancelled event (out)	
6	Create a survey	Admin	Survey details and questions (in) Fillable survey (out)	
7	Edit a survey	Admin	Selected survey and updated details (in) Edited survey (out)	
8	Close a survey	Admin	Selected survey (in) Closed survey (out)	
9	View financial reports	Admin	Selected event (in) Financial report (out)	

10	View event attendee overview	Admin	Selected event (in) Attendee data (out)
11	View survey response report	Admin	Selected Survey (in) Survey response report (out)
12	Filter survey response data	Admin	Filter Criterion (in) Filtered response data (out)
13	View upcoming events	Attendee	View request (in) Upcoming events list (out)
14	View past events	Attendee	View request (in) Past events list (out)
15	View registered events	Attendee	View request (in) Registered events list (out)
16	Register for upcoming event	Attendee	Selected event and registration details (in) Event admission ticket (out)
17	Update event registration	Attendee	Selected event and updated details (in) Update confirmation (out)
18	Cancel event registration	Attendee	Selected event (in) Cancellation confirmation (out)
19	View fillable surveys	Attendee	View request (in) List of surveys (out)
20	Complete a survey	Attendee	Selected survey and question responses (in) Survey completion confirmation (out)

21	Edit a survey response	Attendee	Selected survey and edited answers (in) Survey update confirmation (out)
22	View previous survey responses	Attendee	Selected survey (in) Survey responses (out)

Table 3: Product use case table

8.3 Individual Product Use Cases (PUC's)

This section further expands on the use cases defined in Table 3 by defining specific triggers, preconditions, and outcomes of each use case.

PUC 1: Create a custom form module
Trigger: The admin requests to create a new form module
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to create form modules

Actors: Admin
Outcome: A form module is created which admins can use in any event registration/survey form
Input: Form module name and fields
Output: Reusable form module

PUC 2: Edit an existing form module
Trigger: The admin selects a form module to edit
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to edit form modules
- At least one form module is available for editing

Actors: Admin
Outcome: The selected form module is updated with the admin's changes
Input: Selected form module and updated fields
Output: Edited form module

PUC 3: Create an event
Trigger: The admin requests to create a new event
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to create events

Actors: Admin
Outcome: An event is created and attendees are notified of a new upcoming event
Input: Event details (e.g. title, date, location, cost)
Output: Scheduled event

PUC 4: Edit an event
Trigger: The admin requests to edit an event
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to edit events
- At least one upcoming event is scheduled

Actors: Admin
Outcome: The event details are updated and attendees are notified of a change
Input: Selected event and edited details
Output: Edited event

PUC 5: Cancel an event
Trigger: The admin requests to cancel an event
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to cancel events
- At least one upcoming event is scheduled

Actors: Admin
Outcome: The event is cancelled and the attendees are notified of the change
Input: Selected event
Output: Cancelled event

PUC 6: Create a survey
Trigger: The admin requests to create a new survey
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to create surveys

Actors: Admin
Outcome: The survey is created and attendees are notified of a new survey
Input: Survey details (e.g. title, questions)
Output: Fillable survey

PUC 7: Edit a survey
Trigger: The admin requests to edit a survey
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to edit surveys
- At least one survey is open to responses

Actors: Admin
Outcome: The survey is updated and attendees are notified of the change
Input: Survey to update and updated survey details
Output: Updated survey

PUC 8: Close a survey
Trigger: The admin requests to close a survey
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to close surveys
- At least one survey is open to responses

Actors: Admin
Outcome: The survey is closed
Input: Selected survey
Output: Cancelled event

PUC 9: View financial reports
Trigger: The admin requests to view a financial report
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to view financial reports
- At least one event has been scheduled or completed

Actors: Admin
Outcome: A financial report of the selected event is generated and displayed
Input: Selected event
Output: Financial report of the selected event

PUC 10: View event attendee overview
Trigger: The admin requests to view an event attendee overview
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to view attendee overviews
- At least one event has been scheduled or completed

Actors: Admin
Outcome: An overview of registered attendees is generated and displayed
Input: Selected event
Output: Attendee overview of the selected event

PUC 11: View survey response reports
Trigger: The admin requests to view a survey response report
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to view survey responses
- At least one survey has been released and at least one attendee has responded

Actors: Admin
Outcome: An overview of all survey responses is generated and displayed
Input: Selected survey
Output: Response report of the survey

PUC 12: Filter survey responses
Trigger: The admin selects criterion to filter survey data by
Preconditions:

- The admin is logged in and authenticated
- The admin has permission to view survey responses
- At least one survey has been released and at least one attendee has responded
- The admin has a survey response report open

Actors: Admin
Outcome: The survey data is filtered and the updated report is generated
Input: Criteria to filter by
Output: Filtered survey response data

PUC 13: View Upcoming Events
Trigger: The attendee requests to view a list of upcoming events
Preconditions:

- The attendee is logged in and authenticated
- There is at least one upcoming event

Actors: Attendee
Outcome: The attendee is shown a list of all future events
Input: View request
Output: Upcoming events list

PUC 14: View Past Events
Trigger: The attendee requests to view a list of past events
Preconditions:

- The attendee is logged in and authenticated
- At least one event has completed

Actors: Attendee
Outcome: The attendee is presented with a list of past events
Input: View request
Output: List of past events

PUC 15: View Registered Events
Trigger: The attendee requests to see events they are registered for
Preconditions:

- The attendee is logged in and authenticated
- The attendee is registered for at least one event that has not completed

Actors: Attendee
Outcome: The system displays all events the attendee has registered for
Input: View request
Output: Registered events list

PUC 16: Register for Upcoming Event
Trigger: The attendee selects an event and submits registration details
Preconditions:

- The attendee is logged in and authenticated
- The selected event is open for registration

Actors: Attendee
Outcome: The attendee's registration is confirmed, and an admission ticket is issued
Input: Selected event and registration details
Output: Event admission ticket

PUC 17: Update Event Registration
Trigger: The attendee requests to update their registration information
Preconditions:

- The attendee is logged in and authenticated
- The attendee has registered for at least one event that has not completed

Actors: Attendee
Outcome: The registration details are successfully updated
Input: Selected event and updated details
Output: Update confirmation

PUC 18: Cancel Event Registration
Trigger: The attendee requests to cancel a registration
Preconditions:

- The attendee is logged in and authenticated
- The attendee is registered for at least one event which has not completed

Actors: Attendee
Outcome: The registration is cancelled and the system confirms the cancellation

Input: Selected event
Output: Cancellation confirmation

PUC 19: View Fillable Surveys
Trigger: The attendee requests to view available surveys
Preconditions:

- The attendee is logged in and authenticated
- At least one survey is open for responses

Actors: Attendee
Outcome: The attendee sees a list of surveys they can complete
Input: View request
Output: List of surveys

PUC 20: Complete a Survey
Trigger: The attendee requests to fill out a survey
Preconditions:

- The attendee is logged in and authenticated
- At least one survey is open for responses

Actors: Attendee
Outcome: The attendee's responses are submitted and the application confirms the submission
Input: Selected survey and question responses
Output: Survey completion confirmation

PUC 21: Edit a Survey Response
Trigger: The attendee requests to edit a survey response
Preconditions:

- The attendee is logged in and authenticated
- The attendee has completed at least one survey which is still open for responses

Actors: Attendee
Outcome: The survey response is updated successfully
Input: Selected survey and edited answers
Output: Survey update confirmation

PUC 22: View Previous Survey Responses
Trigger: The attendee requests to view their past survey responses
Preconditions:

- The attendee is logged in and authenticated
- The attendee has completed at least one survey

Actors: Attendee
Outcome: The system displays the attendee's responses for the selected survey
Input: Selected survey
Output: Survey responses

8.4 State Diagrams

Below are formal state diagrams showing the different states of the system and the transitions between them. [fig:stateadmin]Figure 5 shows the state diagram of the admin side of the system and [fig:stateattendee]Figure 6 shows the attendee side.

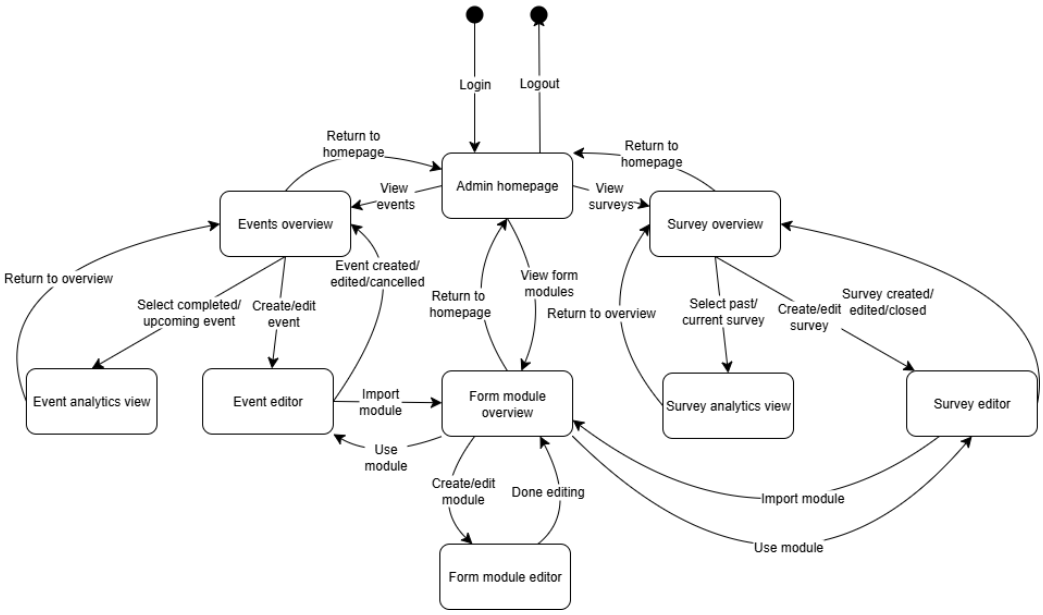

Figure 5: State diagram of the admin side of the system

8.5 Data Dictionary

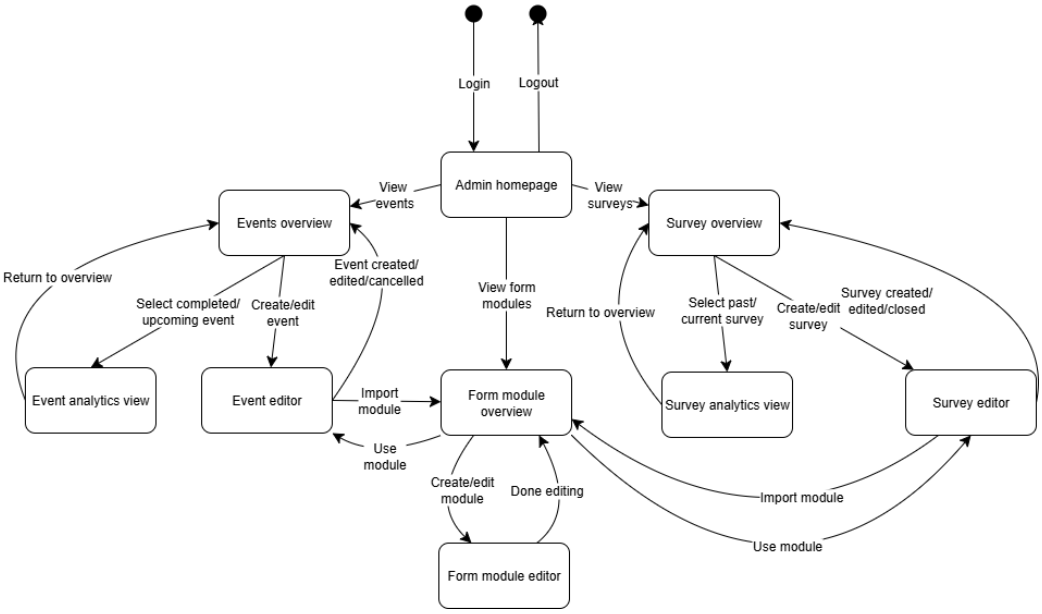

Figure 6: State diagram of the attendee side of the system

8.6 Data Dictionary

9 Functional Requirements

9.1 Functional Requirements

FR-1: The system shall allow admins to create forms with multiple field types.

Motivation: {**G-1**, **G-5**}. MES wants to be able to create forms for their surveys, especially CFES surveys, with different response option types.
Fit Criterion: Forms can be created with different response styles (e.g., single choice, multiple choice, numeric scales) and response data is correctly stored in databases.

FR-2: The system shall allow admins to organize forms for analytics, and specify how the analysis should be done.

Motivation: {**G-1**, **G-3**}. MES wants to use different metrics to view survey data, and wants to combine multiple forms in a singular analysis.
Fit Criterion: Admins can combine results from multiple forms together in one analysis query, and can compare data points as they wish.

FR-3: The system shall provide attendees a method to register for events and view event information.

Motivation: {**G-2**}. Attendees must have a way to register for events and view event information.
Fit Criterion: Attendees can fill out event registration forms and receive and view notifications about the events, and their registration is recorded in the database.

FR-4: The system shall generate QR codes for confirmation and check-in integration.

Motivation: {**G-2**}. Attendees should be able to provide proof of registration to have easy access to events.
Fit Criterion: A QR code is generated and stored in the dashboard for the attendee.

FR-5: The system shall provide an event dashboard for admins with filters for status updates on payments, waivers, event sign-ins, etc.

Motivation: {**G-4**}. Admins must have a way to track the event and relay updates as needed.
Fit Criterion: The admin dashboard provides trackers for sign-ins, and provides a medium to notify attendees of any changes.

FR-6: The system shall store all event data in a secure, centralized database.

Motivation: {**G-6**, **G-5**, **G-2**}. Event and survey data can contain sensitive data from attendees, and should only be accessed by authorized users.
Fit Criterion: Only relevant admins can access stored user data unrelated to their account.

FR-7: The system shall store all questionnaire data in a secure, centralized database.

Motivation: {**G-1**, **G-3**, **G-6**}. See **FR-6**.
Fit Criterion: See **FR-6**.

FR-8: The system shall allow admins to generate analytics visualizations and export CSV/Excel reports.

Motivation: {**G-3**}. Admins should have a way of viewing and exporting survey data.
Fit Criterion: The admin can export queried data as CSVs or Excel-compatible files, and can view metrics of their choosing in the admin dashboard after specifying which metrics to plot.

FR-9: The system shall provide event attendees with a medium to relay feedback for the event.

Motivation: {**G-5**}. MES wants to gather feedback from events and incorporate them in future events.
Fit Criterion: Admins can create feedback forms which attendees are provided during the event which they can fill out, and their responses are stored in the database and are available for admins to access during or after the event.

FR-10: The system shall provide admins with a medium to view all event feedback once an event has completed.

Motivation: {**G-5**, **G-4**, **G-3**}. See **FR-9**.
Fit Criterion: See **FR-9**.

FR-11: The system shall implement a roles-based feature access system which ensures that only relevant functionality and information is made available to the user.

Motivation: {**G-4**, **G-6**, **G-2**}. MES wants access to be restricted and ensure that only authorized personnel can view relevant information.
Fit Criterion: Users assigned roles can access information and features specified by those roles, and cannot see irrelevant features or information.

10 Look and Feel Requirements

10.1 Appearance Requirements

LFR-AP.1 The interface shall comply with MES branding criteria (logo, colour scheme).

Motivation: Ensures that the product is recognizable as an official MES event registration platform.
Fit Criterion: The MES representative shall certify that the product complies with the current standards.

LFR-AP.2 The tool shall have a clean, minimalist layout prioritizing the functionality.

Motivation: Increases accessibility and user productivity, and allows for consistency across different events.
Fit Criterion: During usability testing across the MES events, at least 80% of students rate the forums ease of use a 4+.

LFR-AP.3 Admin dashboard shall clearly display relevant analytics, and distinct charts.

Motivation: Improves organizer accessibility and readability, allowing for easier management during planning operations.
Fit Criterion: The representative shall approve of the analytics layout.

10.2 Style Requirements

LFR-S.1 The design shall be professional but approachable, accommodating all types of events.

Motivation: The app should have a friendly vibe for casual student use, but maintain a sense of maturity for the more formal uses.
Fit Criterion: After use in some MES events, 70% of users shall agree that they trust and respect the product, and 70% shall agree that it was not boring.

LFR-S.2 The admin dashboard shall emphasize functionality and clarity, avoiding distractions just for the sake of aesthetics.

Motivation: Admin and organizers require clear data and analytics, they have no use for stylistic components.
Fit Criterion: A sample of event organizers and club executives approve and are able to interpret all analytics without prior explanation.

11 Usability and Humanity Requirements

11.1 Ease of Use Requirements

UHR-EoU.1 Registration and feedback forms shall be easily and quickly filled out for anyone with no instruction.

Motivation: The app should have a friendly vibe for casual student use, but maintain a sense of maturity for the more formal uses.
Fit Criterion: After use in some MES events, 70% of users shall agree that they trust and respect the product, and 70% shall agree that it was not boring.

UHR-EoU.2 Users shall sign up/sign-in to the app with no instruction.

Motivation: The registration process is the most important aspect of the platform, so the app should have a clear and simple layout.
Fit Criterion: A sample of users shall register/login within 2 minutes without instruction.

UHR-EoU.3 The admin dashboard shall emphasize functionality and clarity, avoiding distractions just for the sake of aesthetics.

Motivation: Admin and organizers require clear data and analytics, they have no use for stylistic components.
Fit Criterion: A sample of event organizers and club executives approve and are able to interpret all analytics without prior explanation.

UHR-EoU.4 The system shall provide survey progress indicators to reduce form abandonment rates.

11.2 Personalization and Internationalization Requirements

UHR-PIR.1 The custom form builder shall allow admin to create personalized forms with event-specific fields.

Motivation: Flexibility is essential to accommodate the diverse set of events as per the scope.
Fit Criterion: The forum is used for 2 events successfully without developer assistance.

UHR-PIR.2 Admin dashboard shall support sorting and filtering of attendees by predefined metrics.

Motivation: Streamlines the attendee organizing process, allowing admin to categorize attendees as per the event requirements.
Fit Criterion: Organizers shall filter attendees within 1 minute of use without instruction.

11.3 Learning Requirements

UHR-LR.1 The app shall be easy for anyone with an intermediate level of English.

Motivation: Users should expect intuitive, concise interactions.
Fit Criterion: 80% of students successfully complete a form with no assistance.

UHR-LR.2 Admin shall be able to build a form with the custom builder with no prior instruction.

Motivation: Event organizers are constantly changing, so it is important to reduce dependency on technical training.
Fit Criterion: 80% of organizers shall complete a basic test form on their first attempt.

11.4 Understandability and Politeness Requirements

UHR-UPR.1 Basic, non-technical language will be used (Sign up instead of RSVP).

Motivation: Increases accessibility among non-technical users, or non-fluent English speakers.
Fit Criterion: 80% of students understand all wording without clarification.

UHR-UPR.2 The app shall communicate errors clearly and politely.

Motivation: Clear error messages reduce frustration and improve user experience.
Fit Criterion: Feedback suggests that 80% of students found the experience satisfying.

UHR-UPR.3 The app shall hide sensitive and confidential information from users.

Motivation: Enhances security and confidence of organizations in the product.
Fit Criterion: Approval from MES representative.

11.5 Accessibility Requirements

UHR-AR.1 The app shall conform to WCAG 2.1 AA standards, improving visibility for impaired users.

Motivation: Ensures accessibility for users with visual disabilities and impairments.
Fit Criterion: Approval from a sample of students with impairments, and evaluation tools (WAVE Web Accessibility).

12 Performance Requirements

12.1 Speed and Latency Requirements

SL-1: The system shall have a maximum 2-second latency for form loading, optimized for mobile devices.

12.2 Safety-Critical Requirements

Insert your content here.

12.3 Precision or Accuracy Requirements

Insert your content here.

12.4 Robustness or Fault-Tolerance Requirements

RF-1: The system shall support 2047 concurrent users during peak event times.

RF-2: The system shall auto-save partially completed surveys every 30 seconds.

RF-3: The system shall recover from a server crash within 60 seconds without data loss.

RF-4: The system shall allow new form modules to be added without any downtime.

RF-5: The system shall allow admins to update event templates without disruption during ongoing events.

12.5 Capacity Requirements

Insert your content here.

12.6 Scalability or Extensibility Requirements

SE-1: The system shall support up to 2047 registrations per event.

12.7 Longevity Requirements

Insert your content here.

13 Operational and Environmental Requirements

13.1 Expected Physical Environment

Insert your content here.

13.2 Wider Environment Requirements

Insert your content here.

13.3 Requirements for Interfacing with Adjacent Systems

Insert your content here.

13.4 Productization Requirements

Insert your content here.

13.5 Release Requirements

Insert your content here.

14 Maintainability and Support Requirements

14.1 Maintenance Requirements

MT-1: The system shall provide updated documentation upon every release and be accessible to all developers

Motivation: Allows for developers to understand the changes made on each release and will be easily be able to modify the code to fix errors or make updates.
Fit Criterion: Updated documentation will be included with the release of each version.

14.2 Supportability Requirements

SU-1: The system shall include a FAQ section to help clarify or answer common questions.

Motivation: Will allow for a better user experience as they can solve their own problems quickly and will make it easier for MES support teams.
Fit Criterion: Ensure a FAQ with common questions obtained

from stakeholders (supervisor and MES council members) is included within the system.

14.3 Adaptability Requirements

AD-1: The system shall run as a web app accessible through Chromium, Firefox, and Safari browsers.

Motivation: This will allow users to use whatever browser they want and will make the application more accessible.
Fit Criterion: Test for compatibility of the application to ensure it runs on all specified browsers.

AD-2: The mobile app components will work on both IOS and Android platforms.

Motivation: Allows users on both mobile ecosystems to use the app making the application more accessible.
Fit Criterion: Test the application in both the IOS and Android Environment ensuring correct functionality of all components.

15 Security Requirements

15.1 Access Requirements

AC-1: The system shall require all users to authenticate using their McMaster University Email before accessing the system

Motivation: Ensure that only the expected users of the system are using and prevent misuse from external actors.
Fit Criterion: Allow only users who have authenticated with their emails to access the system. Ensure logs track who accesses the system and that they are authenticated.

AC-2: Admin level features and sensitive data are only available to users who are assigned admin roles.

Motivation: Prevents unauthorized users from viewing sensitive event and user data.
Fit Criterion: Perform testing to ensure that features deemed to

be admin-level are only accessible to the users who are given those roles. Ensure that data is only accessible to admins

15.2 Integrity Requirements

IG-1: The system shall provide input validation for all user submitted data

Motivation: Ensures the safety of our application by ensuring only the required input types are entering our database. Protects us from SQL injection and data corruption in our database.
Fit Criterion: All inputs will include checks to ensure only the correct type is entering our database. Perform testing to ensure that all inputs have validation checks.

IG-2: The system shall only use HTTPS for communication between clients and servers

Motivation: Ensures a secure connection between our client and server and protects against the interception or tampering of data during data transmission.
Fit Criterion: All requests will be done through HTTPS and if an HTTP request is done we will redirect it to HTTPS.

15.3 Privacy Requirements

PV-1: Users shall have the ability to request for the deletion of their stored personal data

Motivation: Increases transparency and control over their own data for users. Provides reassurances for users that their data can be removed if neccesary.
Fit Criterion: Ensure a mechanism exists for users to delete their data. Peform testing so that the mechanism actually removes all related data from the database.

PV-2: The system shall only store sensitive data after obtaining the consent of the user

Motivation: Keeps us legally sound as we are not storing user data

without first obtaining consent. Also provides users with information on what their data is actually being used for.
Fit Criterion: Ensure that all sections which require inputting sensitive information have consent checks where we explicitly explain what is being stored and how it may be used.

PV-3: All sensitive user information such as student numbers, names and emails shall be encrypted in storage using AES-256

Motivation: Protects user's information from unauthorized viewing and ensures that even if a breach were to occur sensitive information will not be leaked.
Fit Criterion: Inspect database to show that the required fields are stored encrypted and not in plaintext.

15.4 Audit Requirements

AU-1: The system shall track admin action inside an audit log

Motivation: Ensures traceability of admin actions and allows other admins to know who performed what.
Fit Criterion: 100% of actions will be in a log with tracking of user who did the action, timestamp and action commited. Testing will be done to ensure all actions are being logged and logs should be exportable for review.

16 Cultural Requirements

16.1 Cultural Requirements

CL-1: The system shall use Canadian English spelling.

Motivation: MES is an organization located in Canada.
Fit Criterion: All rendered text provided by the system passes a Canadian English spell-checker.

17 Compliance Requirements

17.1 Legal Requirements

LG-1: The system shall store and handle all sensitive personal data in compliance with PIPEDA regulations.

Motivation: The system will store private personal information, which must be done responsibly.
Fit Criterion: The system meets all PIPEDA regulation requirements.

17.2 Standards Compliance Requirements

ST-1: The system code shall conform to the *Airbnb JavaScript Style Guide* and *Airbnb React/JSX Style Guide*.

Motivation: Specified in Development Plan § 11.
Fit Criterion: The system code passes all CI/CD checks related to formatting compliance.

18 Open Issues

There are currently no open issues which need to be considered at this stage of the development cycle.

19 Off-the-Shelf Solutions

19.1 Ready-Made Products

There are some products that exist on the market which solve certain components of this product but no product exists that combines these components into one. For event management there are various tools such as EventBrite and Stubhub which can allow for the selling of tickets for events. The problem with these products are that they don't integrate well with current MES Systems and don't provide all the features as required such as collecting

waivers and ensuring McMaster students attend university specific events. For general data collection Google Forms is the most used tool, and it allows for the creation of complex forms and stores data in spreadsheets. For further data analysis it requires effort on the part of the form creator to actually go and analyze the data to gain actual insights. Also, the forms can become hard to handle as they become complex and can become confusing for respondents. As mentioned before there is no tool that has a combination of the two components.

19.2 Reusable Components

There are various components that we can use to help develop our application:

- **Vite**: Provides us with a framework for developing web apps in **React** and provides built in routing behaviour. It also handles many built in libraries and packages that we can use for front-end styling such as **tailwind-css**. We can also integrate it with other libraries for authentication and for database management
- **TanStack**: Provides us with a powerful middleware to connect backend and front-end services. Will primarily be used for routing but also offers data visualization capabilities. Offers many libraries for working with data and state management that we can leverage.
- **SurveyJS**: This is a potential tool we can use to develop the form builder as it provides us with some pre-built components we can use to develop and customize our own forms.
- **Recharts**: This could be potential charting tool we can use to develop our admin dashboard. Though there are other charting tools the benefit of Recharts is that it provides customization on top of the pre-built charts.
- **Drizzle**: We can use Drizzle as an ORM to map from our database to our web-app. Drizzle is useful since it is lightweight and is code first. Drizzle's simplicity allows for faster development and is perfect for our use case.

These components will allow us to develop our application faster and more efficiently as many of these products are already optimized. All the above products are free to use so do not incur any extra costs or require the handling of licenses.

19.3 Products That Can Be Copied

There don't seem to be any products that can be copied which contain all the required components within a single product. Moreover, the MES wants a product that doesn't require an external license and wants to be fully built in house, so they have control over future development and usage.

20 New Problems

20.1 Effects on the Current Environment

Insert your content here.

20.2 Effects on the Installed Systems

Insert your content here.

20.3 Potential User Problems

Insert your content here.

20.4 Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Insert your content here.

20.5 Follow-Up Problems

Insert your content here.

21 Tasks

21.1 Project Planning

- **Task Assignment:** Tasks are divided based on each member's technical strengths and defined roles. Rayyan leads the admin web UI, Virochaan handles full-stack web and DevOps, Ibrahim focuses on backend and mobile integration, Omar develops mobile UI components, and Mahdi manages backend logic and database design.

- **Collaboration:** Weekly meetings are held to review progress, plan upcoming work, and prepare for supervisor syncs. Daily communication occurs over Discord, while GitHub Issues serve as the main hub for technical updates.

- **Quality Control:** All new code must pass automated linting and testing via GitHub Actions before merging. Peer reviews on pull requests ensure consistency and prevent regressions.

21.2 Planning of Development Phases

Development is organized into five phases, each with clear goals and deliverables. The primary objective is to have a fully functional **Minimum Viable Product (MVP)** by the end of **December 2025**, where all four core features work at a basic level before refinement begins.

1. **Phase 1 – Requirements and Design (September–October 2025)** Finalize system requirements, architecture, and UI mockups for the Admin Portal and Custom Form Builder. Deliverables include the SRS, Hazard Analysis, and shared API specifications with Projects B and C.

2. **Phase 2 – Core Functionality and MVP (Mid October–December 2025)** Implement the main system features for a working MVP:

    - Basic **Custom Form Builder** for creating and saving modular forms.
    - Functional **Registration and Feedback Forms** linked to the backend.

- Simple **Attendee Overview Dashboard** showing registration data.
- Initial **Backend Analytics** for event and submission counts.

The MVP will demonstrate a complete flow—from form creation to submission and viewing results.

3. **Phase 3 – Integration and Refinement (January 2026)** Improve performance, UI/UX, and backend data handling. Begin integration testing with Projects B and C to ensure smooth database and API interaction.

4. **Phase 4 – Analytics and Dashboard Expansion (February–March 2026)** Enhance analytics and administrative tools with filtering, charts, and export options (CSV/PDF). The dashboard will evolve into a production-ready tool for MES event organizers.

5. **Phase 5 – Testing and Deployment (March–April 2026)** Conduct final testing, bug fixes, and documentation updates. Complete cross-team integration and deploy the system for real MES events such as Fireball Formal or CALE Conference. Prepare final deliverables and the Capstone Expo presentation.

22 Migration to the New Product

22.1 Requirements for Migration to the New Product

Insert your content here.

22.2 Data That Has to be Modified or Translated for the New System

Insert your content here.

23 Costs

We don't expect any costs to arise during the development of the project. We will be using free tools and libraries for development and for hosting we will

look to host databases and the back-end locally. In production the hosting will be done by the MES, so it is out of the scope of this Capstone.

24 User Documentation and Training

24.1 User Documentation Requirements

No separate written documentation or manuals are planned for this system. Instead, all user guidance will be built directly into the application to create a more natural and seamless learning experience. This approach ensures that both attendees and administrators can quickly understand how to use the platform without leaving the interface.

The system will include:

- **Interactive Onboarding:** Step-by-step tutorial pop-ups and tooltips that appear during first-time use to introduce users to key features such as event registration, check-in, and analytics.
- **Guided Walkthroughs:** Contextual highlights that show users where to click and how to complete common actions directly within the interface.
- **Short Visual Aids:** Optional embedded videos or animations explaining how to perform important tasks, such as creating an event or submitting a form.

These interactive elements replace the need for formal documentation and are designed to make the system self-explanatory and easy to learn. Users will be able to replay the onboarding sequence or revisit tips at any time from within the application.

24.2 Training Requirements

No formal training sessions or written instructions are required for this system. The design prioritizes simplicity, discoverability, and self-learning through interactive guidance.

- **End Users (Attendees):** Will learn the system through in-app pop-ups and short guided tutorials shown during their first login. These

tutorials will walk users through tasks such as signing up for an event, filling out a form, or checking in.

- **Administrators and Event Organizers:** Will be guided through core management workflows using the same interactive approach, supported by optional short videos or animations that explain advanced features such as creating forms or viewing analytics.

Overall, training and documentation will be fully integrated into the user experience. The application itself will serve as the guide, helping users learn by doing rather than reading external instructions.

25 Waiting Room

Insert your content here.

26 Ideas for Solution

Insert your content here.

Appendix — Reflection

The purpose of reflection questions is to give you a chance to assess your own learning and that of your group as a whole, and to find ways to improve in the future. Reflection is an important part of the learning process. Reflection is also an essential component of a successful software development process.

Reflections are most interesting and useful when they're honest, even if the stories they tell are imperfect. You will be marked based on your depth of thought and analysis, and not based on the content of the reflections themselves. Thus, for full marks we encourage you to answer openly and honestly and to avoid simply writing "what you think the evaluator wants to hear."

Please answer the following questions. Some questions can be answered on the team level, but where appropriate, each team member should write their own response:

Virochaan Ravichandran Gowri Reflection:

1. **What went well while writing this deliverable? During the writing**
   During the writing of this deliverable we all had a clear understanding of the goals of this project and the general plan which we did in the past deliverable. From this through conversations with our supervisors we were easily able to obtain requirements that were relevant to our project. The best part was the communication between the members of our team and with the supervisor. This allowed us to

2. **What pain points did you experience during this deliverable, and how did you resolve them?**
   We needed to make sure that we remained flexible in our requirements if we had to make changes in the future. This was especially crucial because we knew that there could be a chance that there could be additional requirements or changes based on what the client wanted. Also the client wanted a certain stack so we didn't really have flexibility regarding that so it was some new technologies that we need to become accustomed with.

Group Reflection

1. How many of your requirements were inspired by speaking to your client(s) or their proxies (e.g. your peers, stakeholders, potential users)? Most of our functional requirements were obtained directly through conversations with out supervisor (Luke) or through todcumentation provided by him. This provided us with a good starting point on what his vision of the system would look like and from there we built upon and added requiremnts based on what we throught the system needed.

2. Which of the courses you have taken, or are currently taking, will help your team to be successful with your capstone project.

    - SFWRENG 4HC3: This course will help us design the frontend components and create a platform which will provide a positive user experience.
    - SFWRENG 3DB3: Help us design effective database schema and queries.
    - SFWRENG 3A04: This will help us design our architecture for the system. Also gave us experience working with git control and designing sytem with diagrams.
    - SFWRENG 2AA4: Designing full scale software and using github project management.
    - ENG 3PX3 + 2PX3: This course gave us some basic project and team management skills as we had to work on semester long projects similar to what is being done in this course.
    - SFWRENG 3RA3: Gave us help writing this document and will help us refine and update our requirements in the future.

3. **What knowledge and skills will the team collectively need to acquire to successfully complete this capstone project? Examples of possible knowledge to acquire include domain specific knowledge from the domain of your application, or software engineering knowledge, mechatronics knowledge or computer science knowledge. Skills may be related to technology, or writing, or presentation, or team management, etc. You should look to identify at least one item for each team member.**

- Full-Stack Web Development: Some skills in this category that will be needed to be learned include requiring the development of front-end

4. **For each of the knowledge areas and skills identified in the previous question, what are at least two approaches to acquiring the knowledge or mastering the skill? Of the identified approaches, which will each team member pursue, and why did they make this choice?**