**COMSATS University Islamabad, Abbottabad Campus**

**Department of Computer Science**

**Project Proposal**

**University Events Management System**

**CSC392 Object Oriented Software Engineering**

Submitted on: Date: 5/April/23

Group Members:

Maaz Latif (SP21-BSE-050)

Sayed Mujtaba Safi (SP21-BSE-091)

Mian Habib (SP21-BSE-042)

Table of Contents

[CHAPTER 1 PROJECT PROPOSAL 3](#_Toc118288358)

[Introduction 3](#_Toc118288359)

[Vision and Business Case 3](#_Toc118288360)

[Use-Case Model 3](#_Toc118288361)

[Supplementary Specification 3](#_Toc118288362)

[Glossary 3](#_Toc118288363)

[Risk List & Risk Management Plan 3](#_Toc118288364)

[CHAPTER 2 USE CASES 4](#_Toc118288365)

[Use Case Diagram 4](#_Toc118288366)

[Use Cases Distribution 4](#_Toc118288367)

[Brief Level Use Cases 4](#_Toc118288368)

[Student Name 1 (Maaz Latif) 4](#_Toc118288369)

[Use Case: Process Sale 4](#_Toc118288370)

[Fully Dressed Use Cases 5](#_Toc118288371)

[Student Name 1 (Mian Habib ) 5](#_Toc118288372)

[Use Case UC1: Process Sale 5](#_Toc118288373)

# CHAPTER 1 PROJECT PROPOSAL

## Introduction

The University Event Management System is a web-based platform designed to simplify the process of planning, organizing and managing various events within a university. The system offers a range of features which allow university administrators, event planners and other stakeholders to effectively plan and manage events, such as workshops, seminars, ceremonies, and conferences. Through this system, users can easily create, manage and share event schedules, track attendance, manage budgets, and receive feedback from attendees. The system is designed to streamline the event management process, making it easier to plan and execute successful events. By utilizing the University Event Management System, university administrators can save time, reduce costs, and improve the overall experience of the events they organize. The system is user-friendly, simple to use, and easy to access - making it an ideal solution for busy administrators and event planners in the university environment.

# 

## Vision and Business Case

The vision of the Event Management System is to be the go-to platform for universities and their stakeholders to manage all aspects of event planning, organization and execution. The system aims to streamline the event management process, from initial planning stages to follow up after the event's completion. The ultimate goal is to create a platform that empowers administrators and event planners to deliver successful, engaging and memorable events.

The business case for the Event Management System is built on a number of key factors. Firstly, the system offers significant cost savings through reduced expenses associated with event planning and management. By automating administrative tasks and streamlining workflows, the system reduces the time and resources required to plan events, freeing up university stakeholders to focus on other important responsibilities.

Secondly, the system improves the overall event experience for all stakeholders involved. By offering a single platform for event planning and management, the system creates a more efficient, streamlined process, allowing organizers to spend more time on creating engaging and memorable events. Increased engagement and satisfaction with events will help to build stronger connections with stakeholders and promote positive reputation for the university.

Finally, the Event Management System offers valuable insights into the success of events, which can be used to improve future planning and execution. Through collecting feedback and evaluating performance metrics, the system helps universities to identify areas for improvement and make data-driven decisions when planning future events.

Overall, the combination of cost savings, improved event experiences and valuable insights make the Event Management System an essential tool for universities looking to create successful, engaging and memorable events.

## Use-Case Model (Functional Requirements)

The use case model for the Event Management System includes various actors and their interactions with the system.

1. Create Event: This use case involves the administrator or event planner creating a new event in the system. They provide details such as event name, date, time, venue, and other relevant information.

2. Edit Event: This use case involves the administrator or event planner making changes to an existing event in the system. They can modify event details such as date, time, venue, and other relevant information.

3. Delete Event: This use case involves the administrator or event planner deleting an event from the system.

4. Manage Registrations: This use case involves managing the event registration process, where attendees can register for an event. Attendees can register themselves, and the system will capture attendee information and send confirmation notifications.

5. Manage Attendees: This use case involves managing the list of attendees that have registered for an event. The system can generate attendee lists, manage waitlists, and track attendance.

6. Manage Payments: This use case involves managing payments for an event. The system can process and track payments, generate invoices, and issue refunds.

7. Manage Feedback: This use case allows attendees to provide feedback about the event. The system captures feedback and presents it to the administrator or event planner for review.

8. Generate Reports: This use case involves generating reports about the event, including attendance, feedback, financial information, and other relevant details.

9. Manage Communication: This use case involves managing communication related to the event. The system can send notifications to attendees, update them about changes, and provide follow-up information.

Overall, the use case model for the Event Management System covers various aspects of event planning, management, and execution. The system is designed to streamline the entire event process, from planning to follow-up, and enhance the overall event experience for all stakeholders.

## Supplementary Specification (Non Functional)

1. User Accounts

The system should allow for the creation of user accounts for faculty, staff, students, and external users. Each account should have a unique username and password combination that allows them to access their account information and perform certain tasks.

2. Event Creation

The system should provide an interface for administrators to create and schedule events. This interface should allow users to input event details, such as the type of event, date, time, location, and any special requirements such as audiovisual equipment or catering. The system should also support the ability to manage multiple events at the same time.

3. Event Registration

The system should allow users to register for events. This should include features such as choosing which events to attend, adding additional attendees, and providing special requirements or needs. The system should also confirm their registration and send them an email confirmation of their registration details.

4. Event Payment

The system should allow users to make payments for events, either online or offline. For online payments, the system should integrate with common payment gateway providers such as PayPal or Stripe. For offline payments, the system should provide options for users to pay by check or in person.

5. Event Management

The system should provide event managers with features to manage events such as checking attendance, sending event reminders to attendees, and updating event details. This should include support for managing multiple events at the same time.

6. Reporting

The system should provide administrators with features to generate reports on events, attendance, and revenue. This should include the ability to customize reports based on specific criteria, such as event type or date range. The system should also allow for exporting data in commonly used file formats, such as CSV or Excel.

7. Security

The system should have robust security features such as encryption and user authentication to ensure the privacy of user data. User access to the system should be restricted based on their user role, and permissions should be granted only to users who need it.

8. Availability

The system should be available 24/7 on any device. The system should also be scalable to accommodate a large number of users and multiple ongoing events.

9. Customer Support

The system should provide customer support via email, phone, or chat to help users with any issues they may have. Customer support should be available during business hours and have a response time of no more than 24 hours.

## Risk List & Risk Management Plan

Risk List:

1. Technical issues - Failure of the event management system due to technical issues such as power outages, hardware failure or software bugs causing delays or cancellations of events.

2. Cybersecurity attacks - The event management system may be vulnerable to cyber attacks, leading to theft or loss of personal data, and disruption of the system's functions.

3. Data privacy breaches - Unauthorised access to personal data of individuals, resulting in loss or misuse of data, and damage to the university's reputation.

4. Inadequate user training - User error resulting from a lack of training on the event management system, resulting in errors that can impact the success of events.

5. Ineffective communication - Failure to effectively communicate event details to attendees or staff leading to misunderstandings and confusion.

6. Conflicts with other university events - Overlapping or conflicting event schedules, leading to reduced attendance and lower revenue.

7. Student conflicts of interest - Due to conflicts of interest between student organisations, it is possible that some events are favoured more than others, resulting in an unequal distribution of resources.

8. Revenue loss - Ineffective monetisation strategies leading to low revenue generation or no revenue at all, hampering future event planning.

Risk Management Plan:

1. Technical issues can be minimised through regular maintenance and backup procedures of the event management system. Technical staff should be on standby to fix any technical issues that arise quickly.

2. Use strong security measures and protocols to prevent cybersecurity attacks. Continuous security testing and training should also be provided to staff.

3. Implement strict data privacy policies to prevent personal data breaches, encryption of data, and minimising access to personal data.

4. Provide adequate training to users of the system promote understanding and usage best practices.

5. Establish effective communication strategies to ensure event details are communicated accurately and effectively.

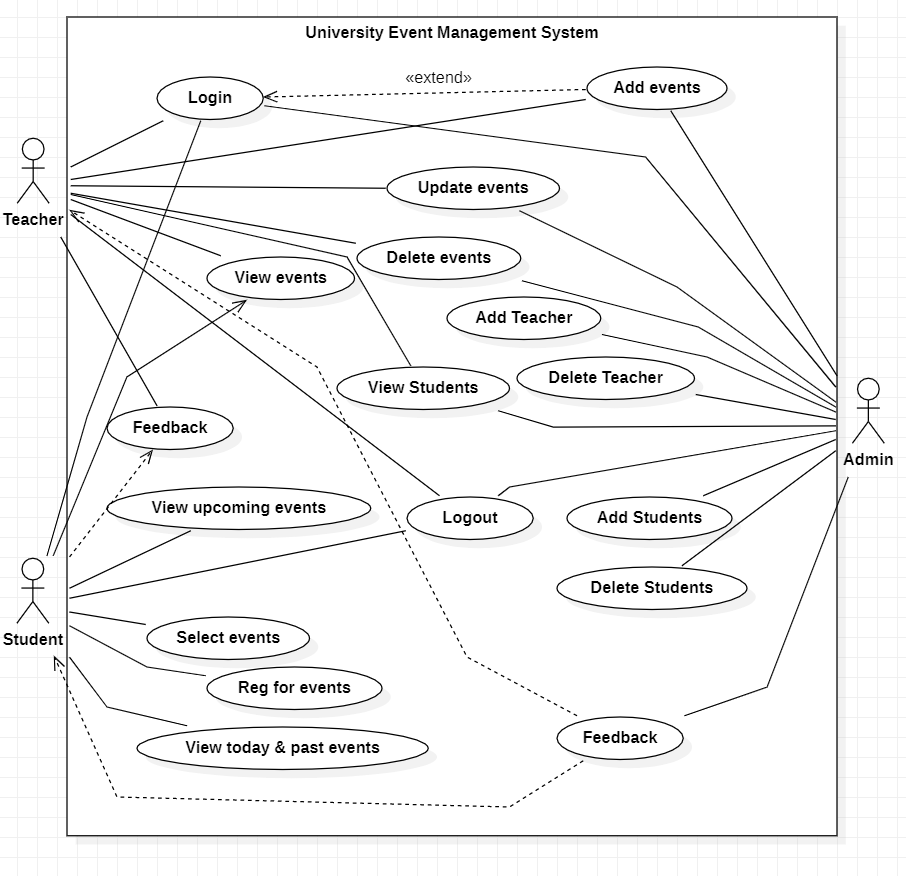
6. Plan well in advance to avoid scheduling conflicts, ensuring events are appropriately spaced apart.

7. Monitor any potential conflicts between student groups and manage resources equitably.

8. Develop monetisation strategies that generate sustainable revenue streams, optimising revenue generation by adopting creative event marketing campaigns, sponsorship opportunities etc.

# CHAPTER 2 USE CASES

## USE CASE Diagram

****

## Use Cases Distribution

|  |  |  |
| --- | --- | --- |
| S# | Group Member | Assigned Use Cases |
| 1 | <Sayed Mujtaba Safi>  <SP21-BSE-091> | UC 1: View upcoming Events  UC 2: Select Events  UC 3: Reg for Events  UC 4: View today / past event  UC 5: Login  UC 6: Logout |
| 2 | <Maaz Latif>  <SP21-BSE-050> | UC 7: View Events  UC 8: Update Events  UC 9: Delete Events  UC 10: Feedback Event |
| 3 | <Mian Habib>  <SP21-BSE-042> | UC 11: Add Events  UC 12: Delete Teacher  UC 13: View Students  UC 14: Add Students  UC 15: Delete Students  UC 16: Add Teacher  UC 17: Feedback |

## Brief level of Use Cases

### Mian Habib

#### Add Event:

User adds new events to the system by filling a form with required fields and clicking "Submit" button.

#### 2.Delete Teacher:

Administrator removes a teacher from the system by selecting the teacher to delete, confirming the deletion, and clicking "Delete" button.

#### 3.Add Teacher:

Administrator adds a new teacher to the system by filling a form with required details and clicking "Submit" button.

#### 4.View Students:

Teacher views a list of students in a class by selecting the class and viewing student information such as name, contact details, and attendance.

#### 5.Add Students:

Administrator adds new students to the system by filling a form with required details and clicking "Submit" button.

*6.Delete Student:*

Administrator removes a student from the system by selecting the student to delete, confirming the deletion, and clicking "Delete" button.

*7.Feedback:*

User provides feedback about the system by filling a form with their feedback and clicking "Submit" button.

## Fully Dressed Use Cases

### Mia Habib (FAsadkj)

#### Add Event

Title: Add Event

Primary Actor: User

Goal: To add new events to the system

Main Flow:

1. User selects the "Add Event" option from the main menu

2. System presents a form to the user with the required fields to fill (such as event title, date, time, location, description, and attendees)

3. User enters the event details and clicks on the "Submit" button

4. System validates the entered data and saves the event into the database

5. System displays a success message to the user

Prototype

#### Delete Teacher

Title: Delete Teacher

Primary Actor: Administrator

Goal: To remove a teacher from the system

Main flow:

1. Administrator selects the "Delete Teacher" option from the main menu

2. System displays a list of all the available teachers in the system

3. Administrator selects the teacher to delete

4. System prompts the administrator to confirm the deletion

5. Administrator clicks on the "Delete" button

6. System removes the teacher from the database and also deletes any associated data such as classes, schedules, and grades

7. System displays a success message to the administrator

**3.** **Add Teacher use case:**

Title: Add Teacher

Primary Actor: Administrator

Goal: To add a new teacher to the system

Main flow:

1. Administrator selects the "Add Teacher" option from the main menu

2. System displays a form to the administrator with the required fields to fill (such as name, email, phone number, and subject)

3. Administrator enters the teacher's details and clicks on the "Submit" button

4. System validates the entered data, generates a unique teacher ID, and saves the teacher into the database

5. System displays a success message to the administrator

#### View Students

Title: View Students

Primary Actor: Teacher

Goal: To view a list of students in a class

Main Flow:

1. Teacher selects the "View Students" option from the main menu

2. System displays a list of available classes

3. Teacher selects the class to view

4. System displays a list of students enrolled in the selected class

5. Teacher can view detailed information about a particular student by clicking on their name

6. System displays the student's information such as name, contact details, parents' information, and attendance

#### Add Students

Title: Add Students

Primary Actor: Administrator

Goal: To add new students to the system

Main Flow:

1. Administrator selects the "Add Students" option from the main menu

2. System displays a form to the administrator with the required fields to fill (such as student name, date of birth, address, contact details, and parents' information)

3. Administrator enters the student's details and clicks on the "Submit" button

4. System validates the entered data, generates a unique student ID, and saves the student into the database

5. System displays a success message to the administrator

**6. Delete Student use case:**

Title: Delete Student

Primary Actor: Administrator

Goal: To remove a student from the system

Main Flow:

1. Administrator selects the "Delete Student" option from the main menu

2. System displays a list of all the available students in the system

3. Administrator selects the student to delete

4. System prompts the administrator to confirm the deletion

5. Administrator clicks on the "Delete" button

6. System removes the student from the database and also deletes any associated data such as classes, schedules, and grades

7. System displays a success message to the administrator

**7. Feedback use case:**

Title: Feedback

Primary Actor: User

Goal: To provide feedback about the system

Main Flow:

1. User selects the "Feedback" option from the main menu

2. System presents a form to the user to fill with their feedback

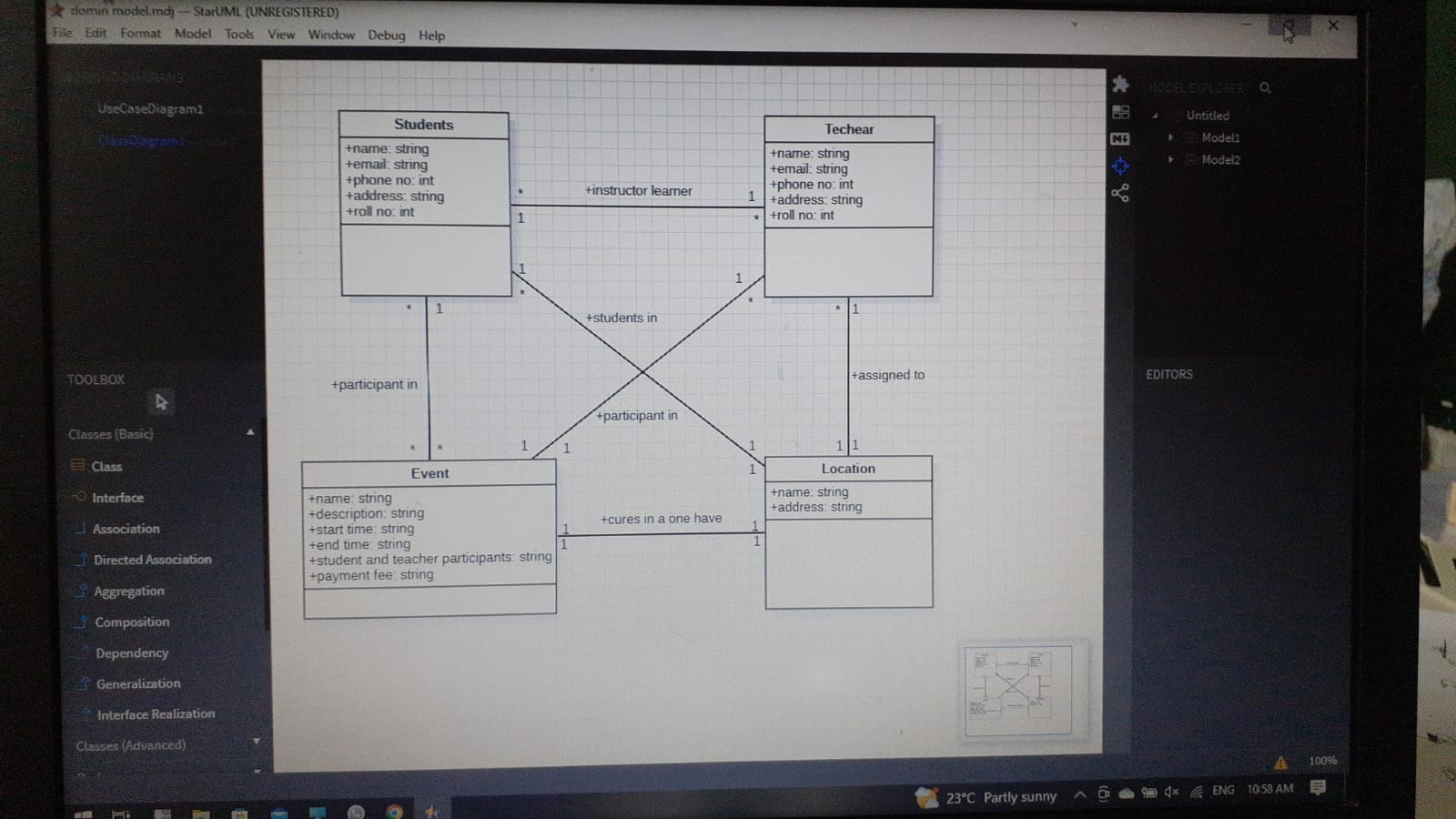
3. User enters their feedback and clicks on the "Submit" button

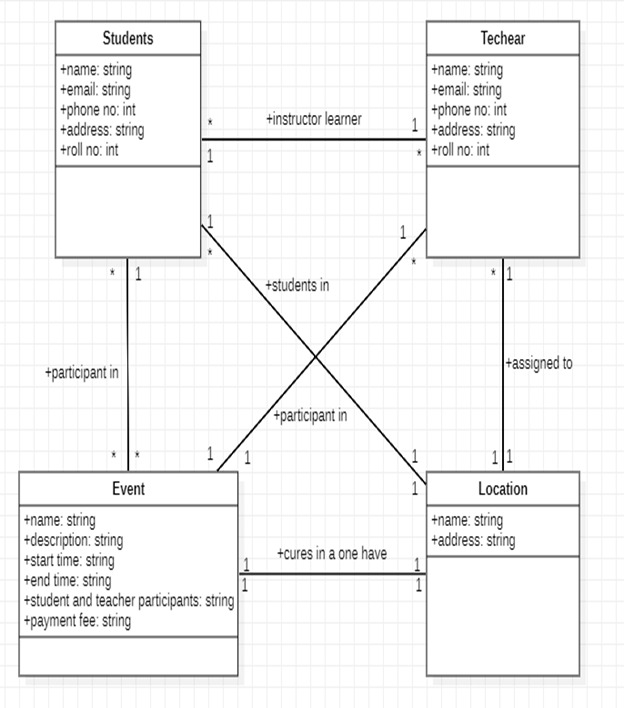
4. System saves the feedback into the database

5. System displays a success message to the user

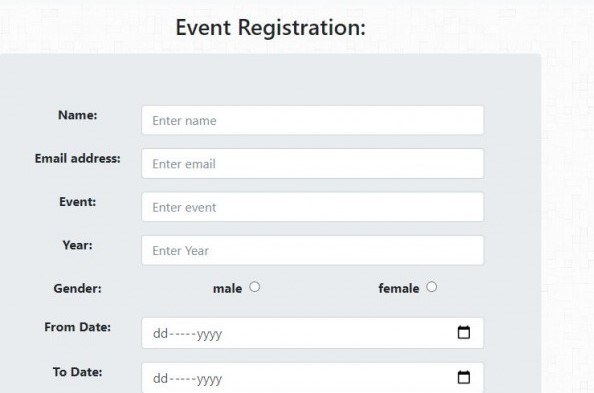
# CHAPTER 3 Domain Model:

**Domain Model**

****

****

**Prototype**

****