**INTI International College Penang School of Engineering and Technology**

**3+0 Bachelor of Science (Hons) in Computer Science, in collaboration with Coventry University, UK**

**3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK**

**Coursework cover sheet**

**Section A - To be completed by the student**

|  |  |
| --- | --- |
| Full Name: Lim Ni Feih | |
| CU Student ID Number: 12672774 | |
| Semester:2 | |
| Session:  **April 2022** | |
| Lecturer:  **Nadhrah Abdul Hadi (nadhrah.abdulhadi@newinti.edu.my)** | |
| Module Code and Title:  **4067CEM Software Design** | |
| Assignment No. / Title:  **Continuous Assessment** | % of Module Mark:  **50** |
| Hand out Date:  **22nd April 2022** | Due Date:  **Task 1: 13 May 2022, by 11.59pm**  **Task 2: 1 July 2022, by 11.59pm**  **Task 3: 17 June 2022, by 11.59pm.**  **Task 4: 17 June 2022, by 11.59pm.**  **Task 5: 17 June 2022, by 11.59pm.** |
| Penalties: No late work will be accepted. If you are unable to submit coursework on time due to extenuating circumstances, you may be eligible for an extension. Please consult the lecturer. | |
| Declaration: I/we the undersigned confirm that I/we have read and agree to abide by the University regulations on plagiarism and cheating and Faculty coursework policies and procedures. I/we confirm that this piece of work is my/our own. I/we consent to appropriate storage of our work for plagiarism checking.  Signature(s): \_\_\_\_\_\_\_\_\_\_\_\_LNF\_\_\_\_\_\_\_\_\_\_\_\_ | |

**Section B - To be completed by the module leader**

|  |  |  |
| --- | --- | --- |
| Intended learning outcomes assessed by this work:  1. Understand and apply appropriate concepts, tools and techniques to each stage of the software development  2. Understand and apply design patterns to software components in developing new software  3. Demonstrate an understanding of project planning and working to agreed deadlines, along with professional, interpersonal skills and effective communication required for software production  5. Demonstrate an awareness of, and ability to apply, social, professional, legal and ethical standards as documented in relevant laws and professional codes of conduct such as that of the Malaysian National Computer Confederation. | | |
| Marking scheme | Max | Mark |
| 1. User Story Mapping 2. Setting up a GitHub Repository 3. Creating a Class diagram and design pattern selection 4. Creating a Prototype User Interface and Usability Testing 5. Discuss the ethical issue related to the software | 20  10  30  20  20 |  |
| Total | 100 |  |

# Class Diagram

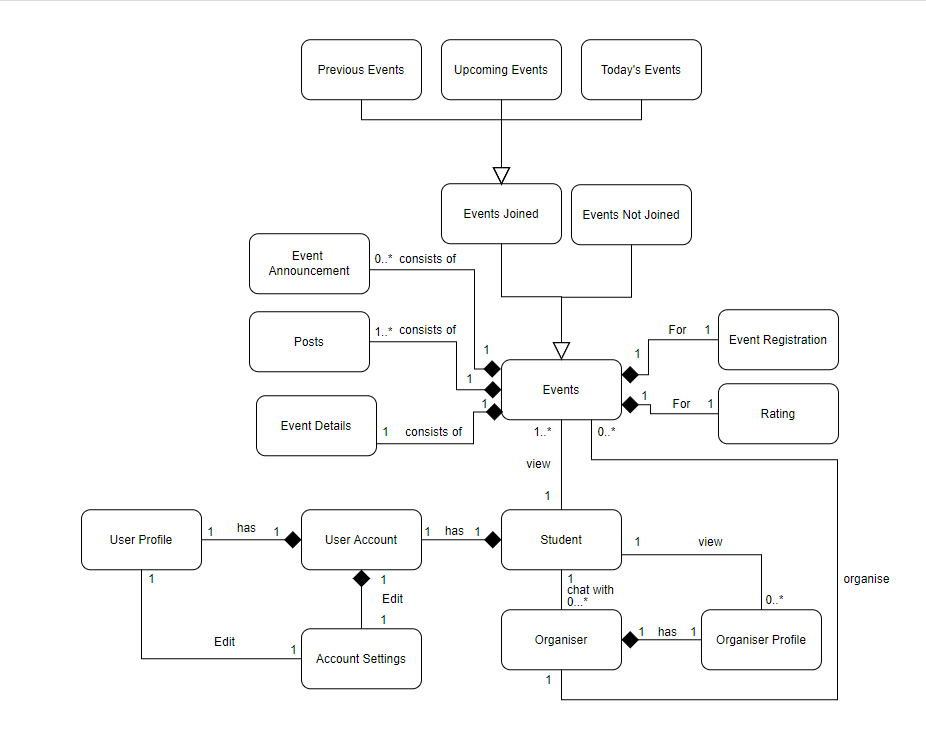


Figure 1

Figure 1 above is the class diagram of my College Event System application. The main function of the application is for students to register for college events, chat with the organizer, and rate the events joined.

**Explanation**

As shown in figure 1, a student has a user account, chat with zero to many organizers, view zero to many organizer profiles, and view 1 to many events in the application.

A user account has a user profile.

One account settings is used to edit one user profile and one user account of a student.

An organizer has one organizer profile and organize zero to many events.

An event in the application consists of one event details, one to many posts, and zero to many event announcements. There is only one registration and one rating for one event per student.

The event in the application is inherited by two subclasses which are events joined and events not joined. The events joined is the inherited by three subclasses which are previous events, upcoming events, and today’s events.

# Design Pattern

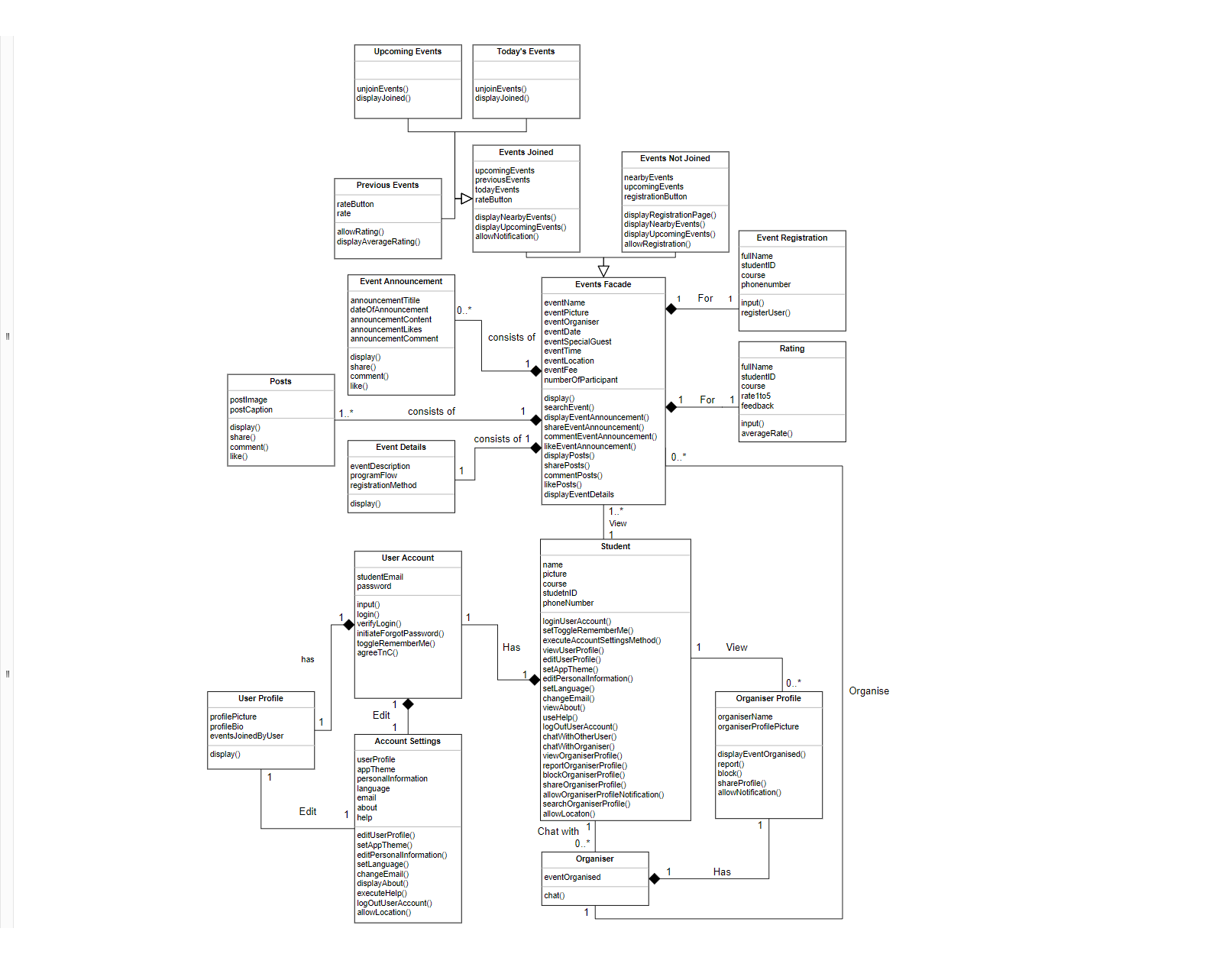


Figure 2

Figure 2 shows the UML diagram that represents the class diagram of my College Event System as a design pattern UML.

## Problem

**How do a few students register for an event in the system?**

This problem can be solved by implementing a login function for the students so that they have their own user account. This way, the student can be recognized by the system as a user. The student can then view the events organized by the organizers. The events in the system consists of and display the event announcement, posts, and details. With this, the student can view the contents of the events posted by the organizer. Then, the student can decide whether to join the events or not. If he or she decides to do so, he or she can tap on the join button and fill in the details required to join.

## Facade

The design pattern chosen to represent the class diagram of my College Event System is Facade.

**What is Facade?**

Facade pattern hides the complexities of a system and provides an interface to the client using which can be used by the client to access the system. This design pattern is considered a structural pattern as this pattern adds an interface to existing system to hide its complexities.

This pattern involves a single class which provides simplified methods required by the client and delegates calls to methods of existing subsystem classes. A representation of the client’s relation with the Facade class is shown in Figure 3 below.

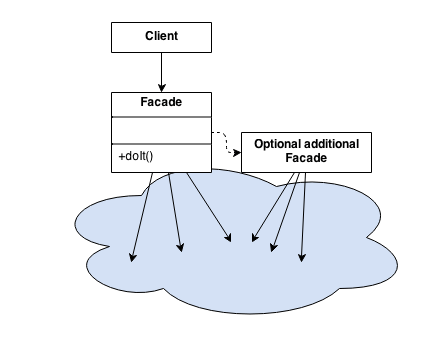


Figure 3

### Justification

Facade consists of the following class:

**Wrapper**

A wrapper class is used to encapsulate the subsystems. The wrapper or also known as facade delegates the client’s requests to the appropriate subsystems and know which classes are responsible for the request. The facade class can be considered as a manager managing the requests of its boss which is the client and delegating the tasks to its subordinates which are the subsystems.

The wrapper/facade in the UML diagram of Figure 2 is Events. This is because it encapsulates the subsystem which are event announcement, posts, event details, rating, and registration. It also delegates the methods to the subsystems based on the functions called by the client. For example, if the student wants to rate an event, the event has to first be joined by the student and is a previous event. Then, the facade class will allow the student to rate the event by delegating the task to the rating subsystem. The same goes for registration as the event has to first be an event that is not joined by the student before the facade will delegate the task to the registration subsystem class and allow the student to register for the event.

**Subsystem**

The subsystem classes implement its functionality and handle the work assigned by the Façade object.

Event announcement, posts, event details, rating, and registration are the subsystems of the system as per the UML diagram shown in Figure 2. These subsystems will carry out the requests of the clients through the Facade class.

**Client**

The client in this design pattern is coupled to Facade only instead of the subsystems to simplify the process.

The client in the UML diagram (Figure2) of my college event system is the student. For instance, the student can use the application to view event details which will cause the system to send the requests to the facade class which is events. The facade class will then return the student’s requests and allow the student to make use of the application. The student can also carry out other tasks such as search event, view event announcement, share event announcement, comment on event announcement, view posts, share posts, like posts, register event, allow notification for events, rate event, view average rating, and unjoin event.