**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: **TAN YI JIA** | |
| CU Student ID Number: **12672752** | |
| 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): Yi Jia Tan | |

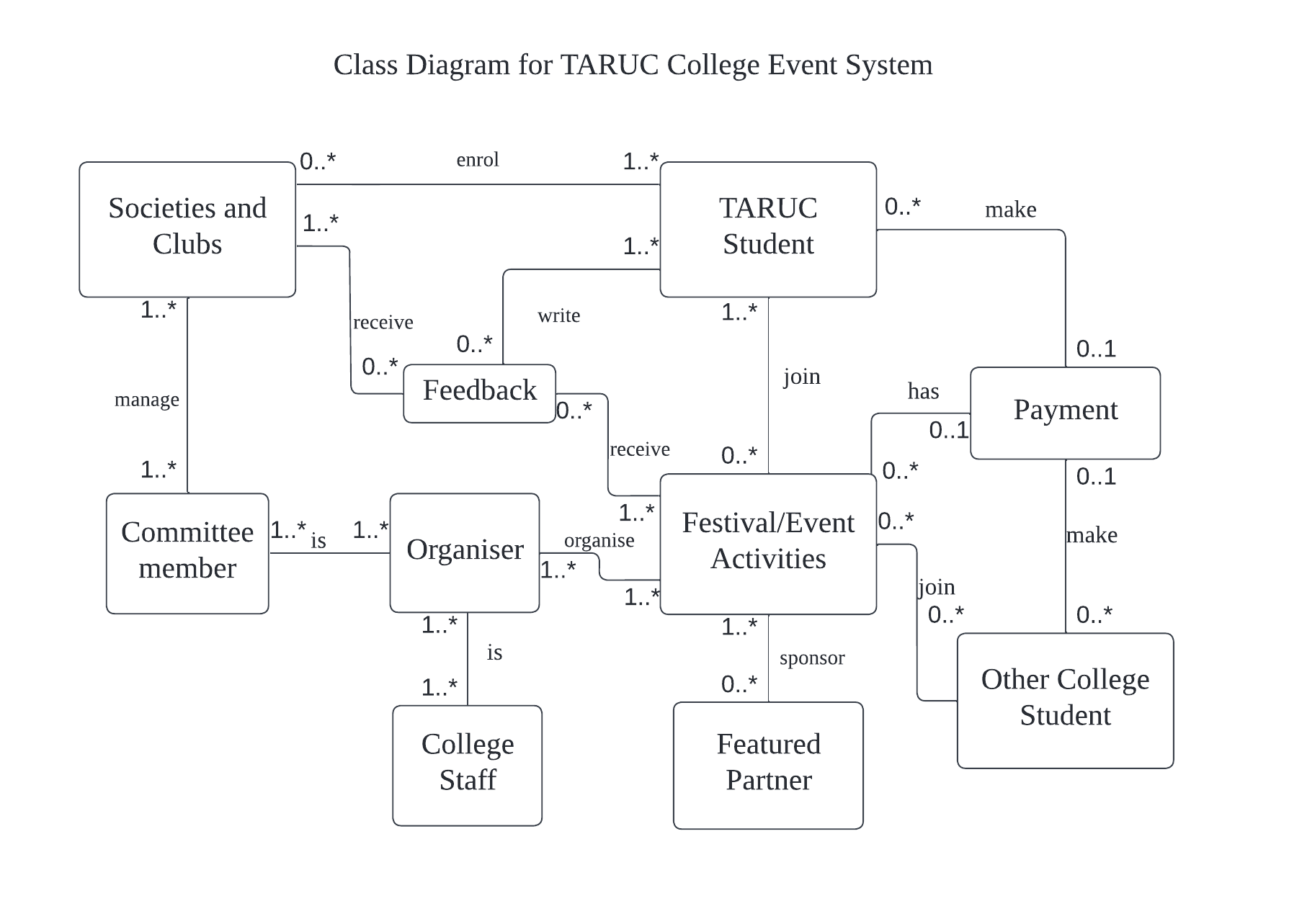
## **Section B - To be completed by the module leader**

|  |  |  |
| --- | --- | --- |
| 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 |  |

# **Task 3 – Creating a Class diagram and design pattern selection**

Lucid Chart is used in this task to draw the Class Diagram and UML Diagram to represent the TARUC college system.

### **Class Diagram**



The relationship between **TARUC Student** and **Societies and Clubs** shows that a student may enrol in zero or many societies and clubs and that the same condition may be associated with several students.

One to many **TARUC Student** may write zero or many **Feedback** to the societies and clubs or the festival/event activities while the **Societies and Clubs** or **Festival/Event Activities** may receive zero to many **Feedback** from the students.

One to many **Committee member** will manage a **Societies and Clubs** whereas the committee member is also can manage many others societies and clubs.

The relationship between **TARUC Student** and the **Festival/Event Activities** describe that a student may join zero or many events and there may have the same condition for others student.

The relationship between **Other College Student** and the **Festival/Event Activities** describe that the students may join zero or many TARUC events that are opened to all students.

The relationship of **Committee member** with the **Organiser** illustrates that many committee members in the club can represent as an organiser to organise one to many **Festival/Event Activities**. The committee member can also become others organiser that show one or many committee member can be one or many organiser.

The example of college staffin this system is the lecturers and head of programme. One or many **College Staff** can act as an **Organiser** to organise one to many **Festival/Event Activities** whereas the college staff can become another organiser to organise festivals and events.

A **Festival/Event Activities** may have zero or many **Featured Partner** to sponsor the festival/event while the **Featured Partner** can sponsor to many other **Festival/Event Activities**.

A **Festival/Event Activities** may have zero or one **payment** while the other festival/event activities will also have the same condition.

The **TARUC Student** and **Other College Student** can make **Payment** to those festival/event activities which have the fees, for instance, registration fee, camp t-shirt fee, and others. The **Payment** may consist of one or many **TARUC Student** and the **Other College Student** while the **TARUC Student** and **Other College Student** can make zero or one **Payment** only.

### **Design Pattern and UML Diagram**

The problems in the system are how the TARUC students enrol in clubs how TAR UC student and other college students register in festival/event activities with college events website. The suitable design pattern that can be implemented on the problem is facade. The reason choose Facade in this system is to provide a simple interface which are user-friendly for users to access in the complex system. Furthermore, facade to tightly joined the abstractions and operations of a subsystem. Facade is a structural design pattern that deliver the complex system in simplified interface. The purpose of facade is to hide the complication of the internal system behind a single interface that looks simple from exterior but not reduce the complexity of the system. Facade also can minimize the dependencies on the sub-system. One of the examples for facade pattern in real life is STM machine, for instance, we want to withdraw RM200 from the ATM machine, we just need to insert the bank card into the ATM machine and enter the PIN of the card. Next, we choose to withdraw and key in the amount then the money will come out. It is very easy and simple for the user. Nonetheless we did not know about how the system run on the background, for example, how the system needs to validate the account, check the PIN, check the amount, and others.

Based on the class diagram, there are many classes that are connect with *Festival/Event Activities* whereas some of the class have the same relationship. Besides that, the *committee member* *and college staff* both are the *organiser* but they are in two different class while the *TARUC Student* and *Other College Student* are also have the same condition which both have the same dependencies which is join the *festival/event activities* but they are in different class. All of these make the system seemed very complex. Thus, one of the solutions is combined the committee members, college staff, TARUC students and other college student in a same class which is a superclass that known as ‘user’. This is to minimalize the dependencies on the sub-class.

