**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: SATISH A/L PRAKASHAM | |
| CU Student ID Number: P22014510 | |
| Semester: 1 | |
| Session:  **August 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:  **6th September 2022** | Due Date:  **Task 1: 30 September 2022, by 11.59pm.**  **Task 2: 18 November 2022, by 11.59pm**  **Task 3: 18 November 2022, by 11.59pm.**  **Task 4: 18 November 2022, by 11.59pm.**  **Task 5: 18 November 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): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |

**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 (30 marks)

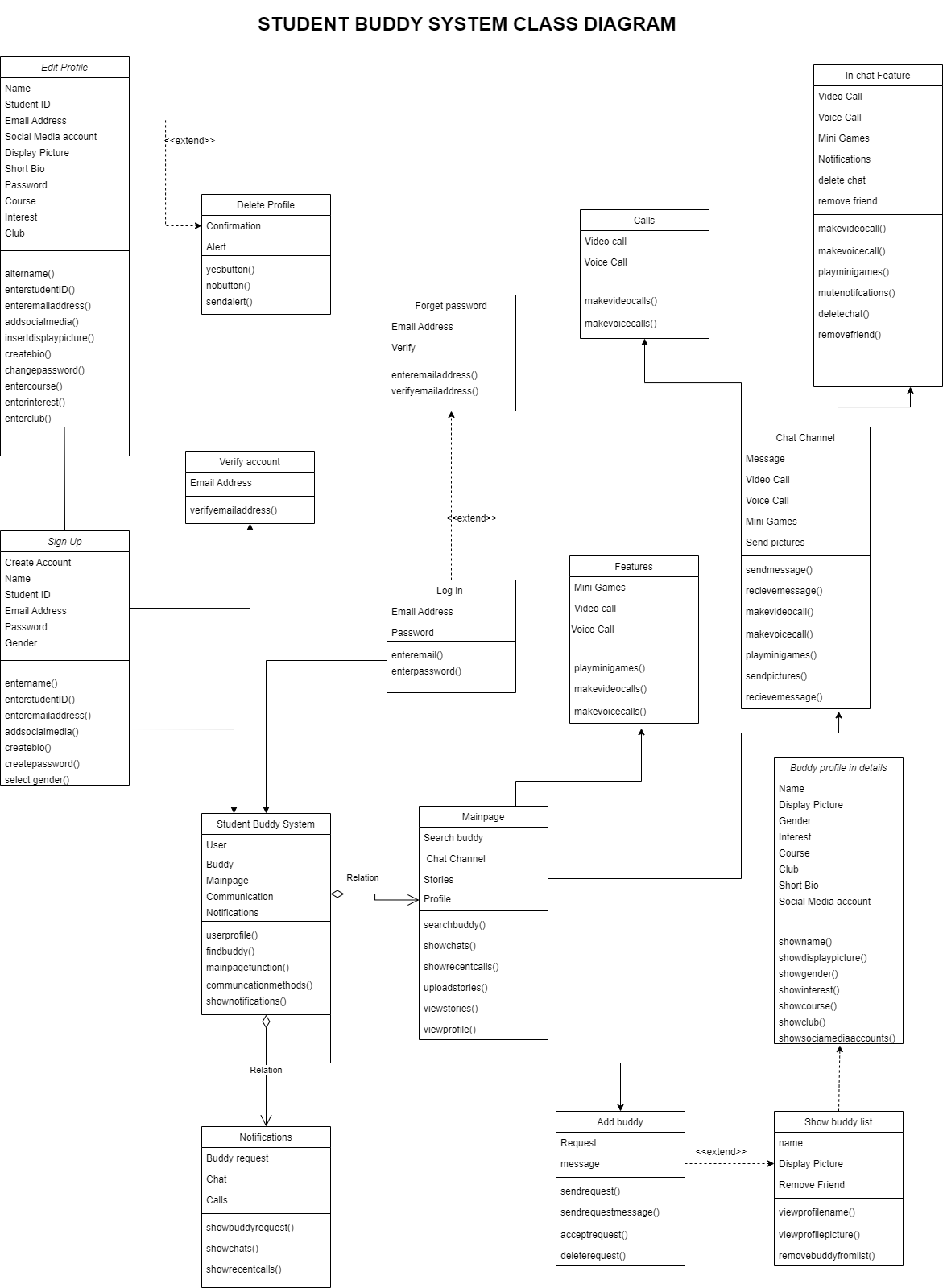
Create a simple Class diagram which should consists of the Classes that might be used to represent the system and the association between them. You don’t have to declare the attributes and operations for this activity. You do have to explain the class responsibility of each class declared. You can use software like StarUML to complete this activity.

Output – A class diagram containing classes and associations. In Word format, uploaded to GitHub.

Consider the problem and select a suitable design pattern that can be implemented on the problem. Give justification on why the design pattern was chosen. Draw the UML diagram representing your class diagram as a design pattern UML. Include all the abstract class/interface, concrete class and inheritance (if any) used to represent the problem.

Output – UML diagram representing the design pattern. In Word format, uploaded to GitHub.

***Task 3***

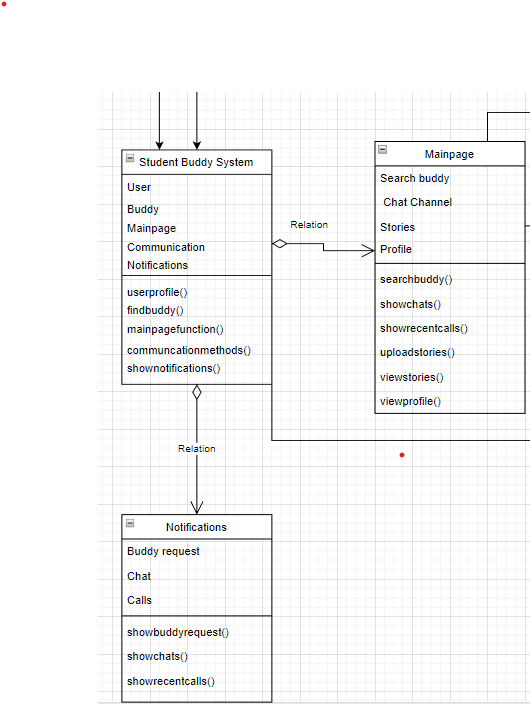


**Figure 1 shows the class diagram of the Student Buddy Application**

Design Pattern Selection

In software design, a design pattern is a generic, repeatable solution to a common issue. A design pattern isn't a finalized product that can be written in code right away. It is a description or model for problem-solving that may be applied in a variety of circumstances. By offering tried-and-true development paradigms, design patterns help hasten the development process. Effective software design entails taking into account problems that might not show up until later in the implementation process. For programmers and architects who are familiar with the patterns, reuse of design patterns helps to eliminate subtle issues that can lead to large difficulties and enhances code readability. There are mainly three types of design patterns, the first one being creational design patterns. The creational design patterns are about class instantiation. Which means, the creational design pattern creates objects in a way that it can suit the situation around it. Secondly, Structural design patterns, this class is more about the class and object attributes. Which means this design can help to identify the relationships between the entities. Last but not least, the behavioral design pattern, this design pattern can aid the communications between objects.

The design pattern chosen to be implemented in the class diagram is the builder design pattern from the creational design patterns. This design pattern delegates the structure of a big object from its representation so that the same construction process can create a alternative representation. Based on Figure 1, the problem is that there are many attributes in the main class which is the Student Buddy System. By implementing the builder design pattern the attributes in class Student Buddy System is delegated into a separate class of its own. Such as the Mainpage class and the Notification class. The Mainpage class and the Notification class now continues the same construction process and the Student Buddy System class. By using this design pattern, it helps to keeps the design much more organized and easier to understand. Figure 2 shows the part where the design pattern was implemented.



**Figure 2 shows the chosen design pattern is applied in the class diagram.**