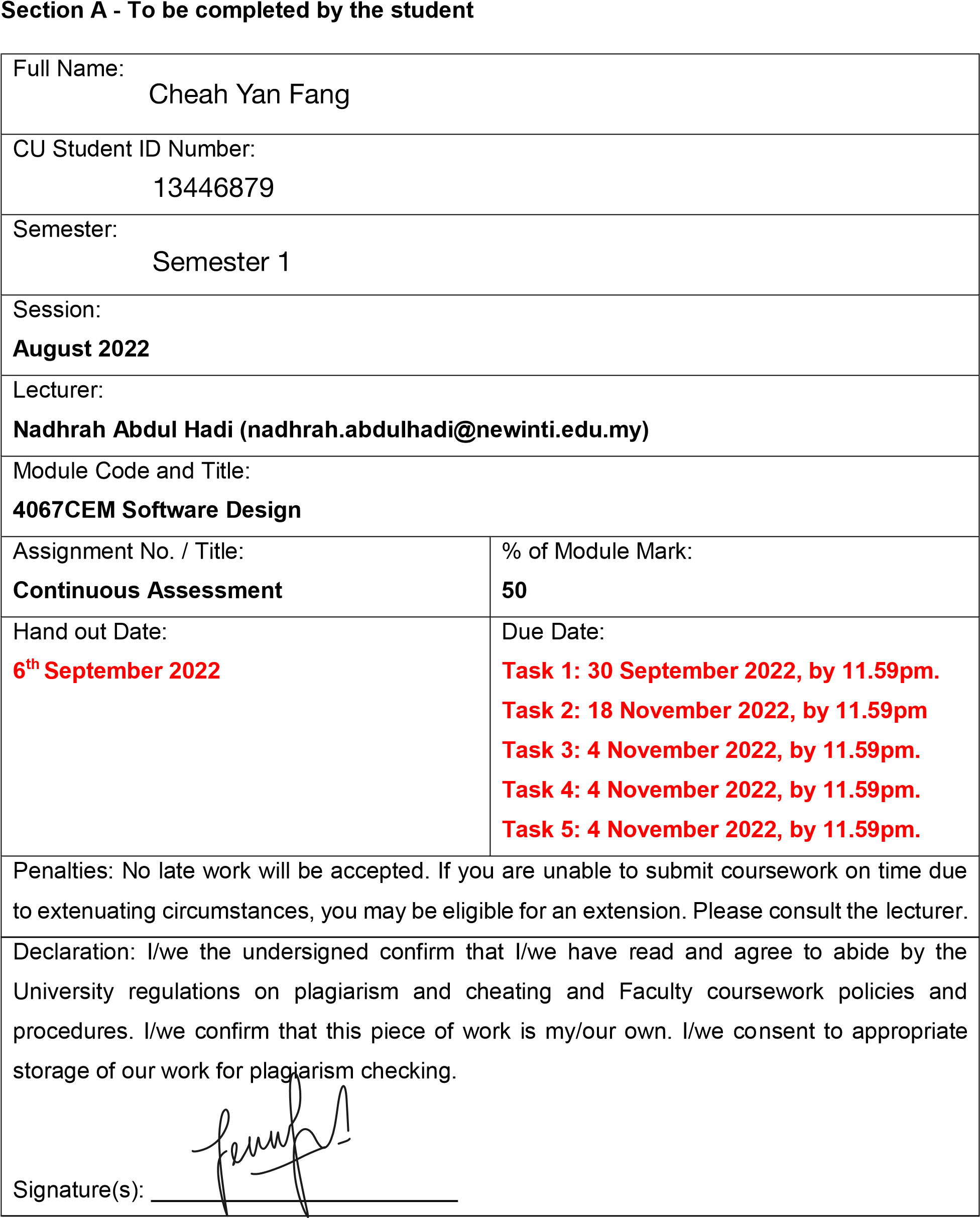
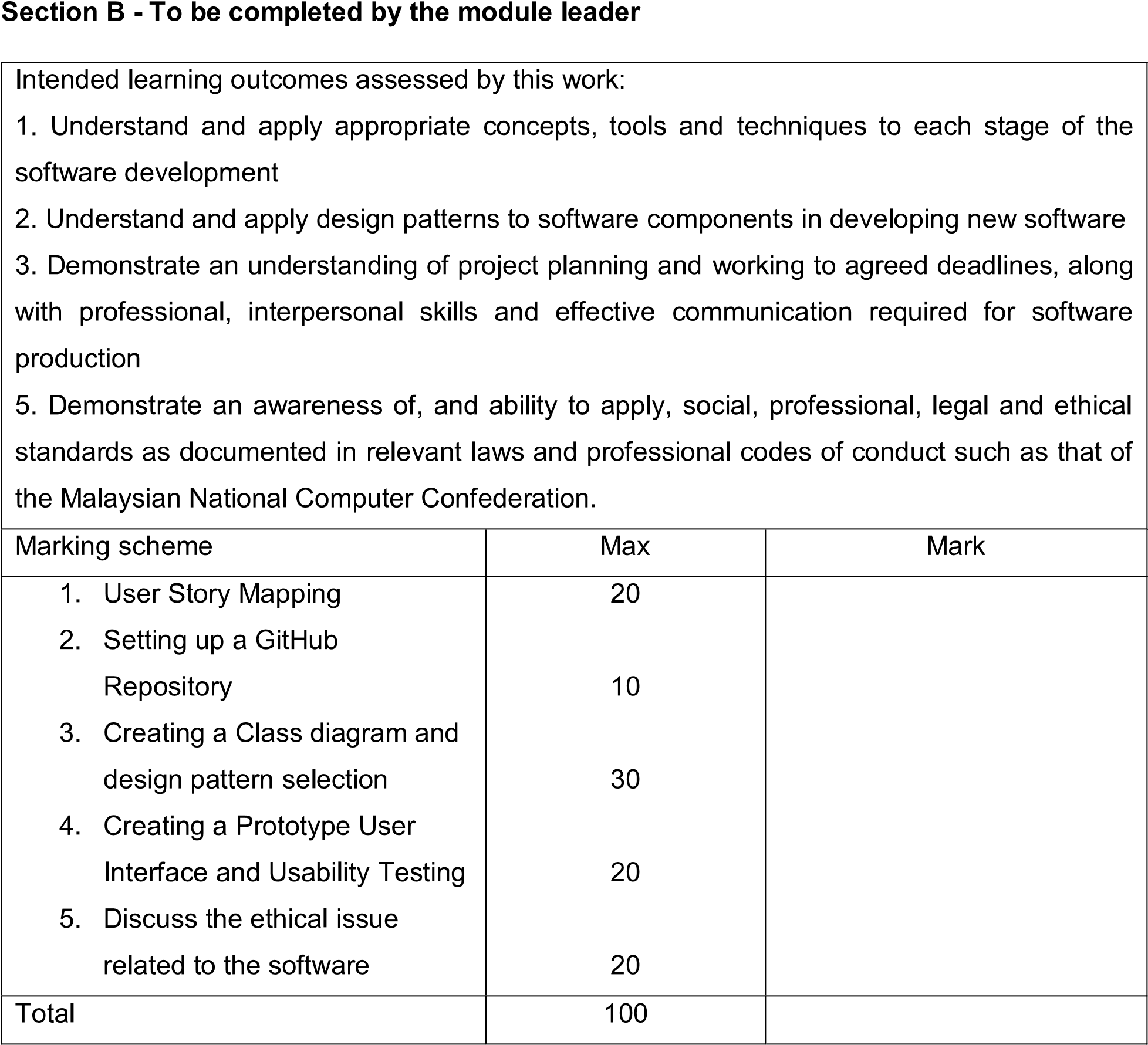
**INTI International College Penang School of Engineering and Technology**

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

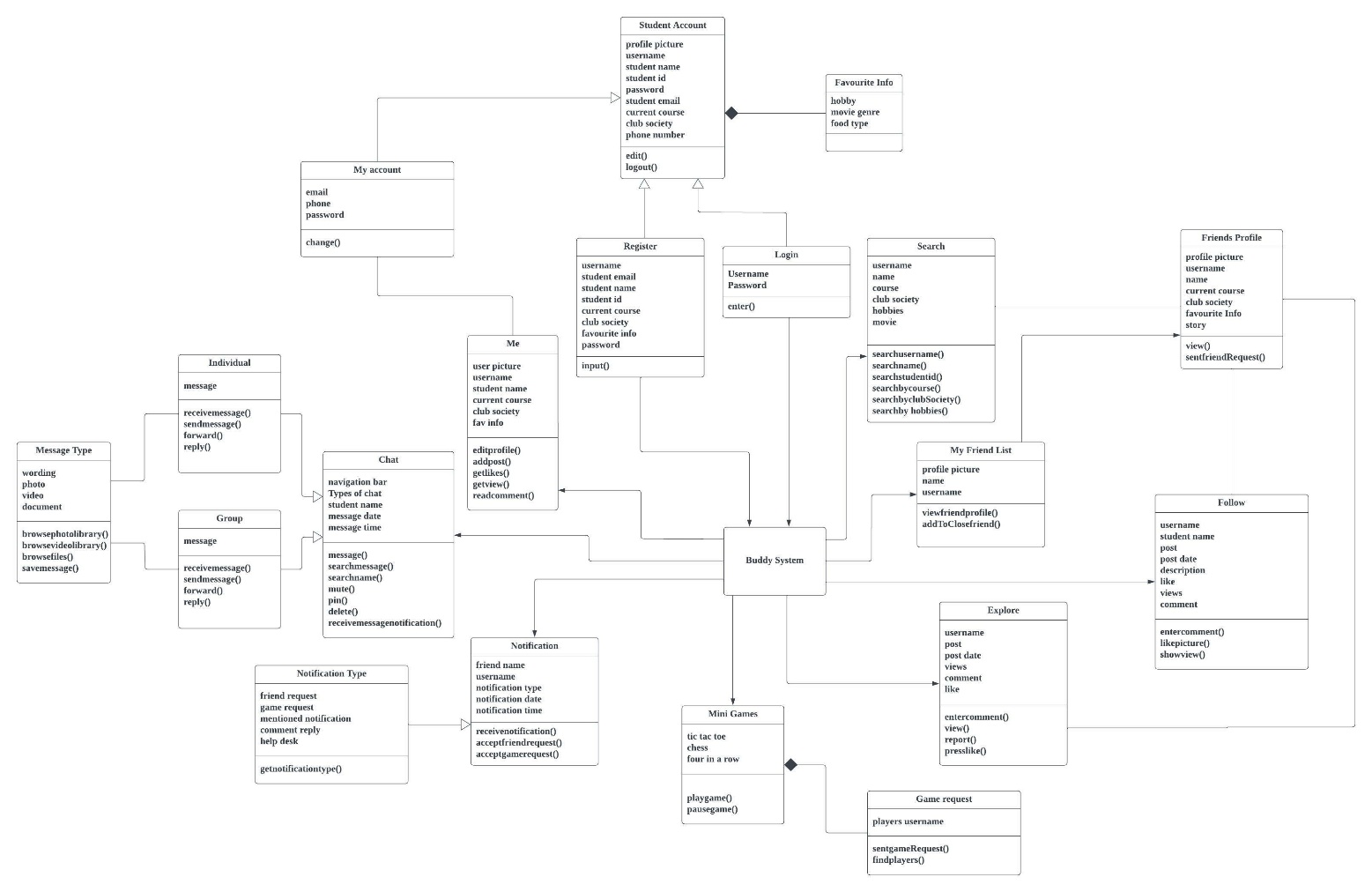
**3+0 Bachelor of Science (Hons) in Computing, in collaboration with Coventry University, UK**

**Coursework cover sheet**





**Task 3 – Buddy System Class Diagram**

**3.1 Class diagram of Buddy System**

A class diagram shows how the relationships and source code interconnections between classes. A class diagram looks like a flowchart where classes are represented by boxes with three rectangles inside each box. The class name appears at the top of each box, followed by a list of its attributes in the middle and operations at the bottom.

This class diagram is created based on the Buddy System application. The diagram can represent how data is converted into information and processes when the user uses the Buddy System. It also provides the overview of how the system works through its operation on finding a buddy and others technical function as well. There are 19 classes included in this class diagram. The main superclass in the class diagram is consist of the search function, My Friend List, Explore, Mini games, Notification, Chat, and Me, whereas the subclass is remaining ones which is the derivation of the super class. The relationship between classes is denoted by the different types of arrows in between. Association, direct associations, composition, inheritance is the relationship used to illustrates the Buddy System class diagram.

**3.2 Design Pattern**

Diagram

Description automatically generated

**Creational Pattern – Abstract Factory**

In software design, a design pattern is a basic, improvement of work to significant problems. A design pattern isn't a finalized product that can be written in code right away. It is a concept or model for problem-solving that may be applied in a variety of circumstances.

The figure above shows a part of the class diagram. The abstract factory belongs to the creational design pattern that is applied to the diagram. The Chat is the user interface widget's abstract class represented by an abstract factory class, which contains methods for creating instances of each abstract class. The individual and group represent the concrete factories, which are a concrete subclass of an abstract factory that uses their methods to build widgets instances for the same platform. Concisely, the abstract factory class and its concrete subclasses are structural sets of concrete classes that operate with various but similar products.

The abstract factory isolates concrete classes, making it easy to manage the types of objects an application builds. Clients are isolated from implementation classes because a factory is responsible for producing product objects. Moreover, it encourages uniformity among products. It's crucial that an application use only one family at a time of product objects when those objects are made to function together as a family. It is simple to implement this using an abstract factory pattern.