**INTI International College Penang School of Computing**

**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 Koon Teng | |
| CU Student ID Number:  14196313 | |
| Semester:  1 | |
| Session:  **April 2023** | |
| Lecturer:  **Puteri Nursyawati Azzuri (puteri.azzuri@newinti.edu.my)** | |
| Module Code and Title:  **4067CEM Software Design** | |
| Assignment No. / Title:  **Continuous Assessment** | % of Module Mark:  **50** |
| Hand out Date:  **12 May 2023** | Due Date:  **Task 1: 02 June 2023, by 11.59pm.**  **Task 2: 07 July 2023, by 11.59pm**  **Task 3: 23 June 2023, by 11.59pm.**  **Task 4: 23 June 2023, by 11.59pm.**  **Task 5: 23 June 2023, 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. | |
| A close-up of a signature  Description automatically generatedDeclaration: 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 the 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 | 20 |  |
| 2. Setting up a GitHub |  |
| Repository | 10 |
| 3. Creating a Class diagram and |  |
| design pattern selection | 30 |
| 4. Creating a Prototype User |  |
| Interface and Usability Testing | 20 |
| 5. Discuss the ethical issue |  |
| related to the software | 20 |
| Total | 100 |  |

**The 4067CEM assessment should be completed as a full individual work over the course of the module. The assessment output are only judged at the end of the module and not by the expectations during that week. The assessment should be undertaken individually. All submissions will be checked against each other and the internet for possible plagiarism.**

Activities – These activities consist of **50%** of your coursework marks. It will be run throughout the semester and there will be a final submission at the end of the semester. These activities consist of activities that will be done in a software design phase.

# System

Student Business System for College.

# Task 1 – User Story Mapping (20 marks)

The first thing that you need to do is ask the user what they wished for in a system. The user here can be your friends as the system is related to them. Get at least 10 real users to get their feedback. Document their feedback. Use software like Trello to complete this activity.

Output – All the user stories, and backlog with goals, activities, and tasks. In Word format, uploaded it to GitHub.

Due – Week 9 of the semester. 02 June 2023, by 11.59 pm.

# Task 2 – Setting up a GitHub Repository (10 marks)

This is where the output of the tasks will be stored, Make sure you register an account, create a repository and your files are uploaded here and it is in an organized manner and can be easily found.

Output – GitHub Repository with Task 1, Task 3, Task 4 and Task 5 documents. Take note the date of the files will be shown so you must follow the due date of each task.

Due – It will be accessed on Week 14 of the semester. 07 July 2023, by 11.59 pm

# 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 it 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 it to GitHub. Due – Week 12 of the semester. 23 June 2023, by 11.59 pm.

# Task 4 – Creating a Prototype User Interface and Usability Testing (20 marks)

Create a Prototype User Interface (hand drawn/digital) of TWO (2) important functions of the proposed system. Come up with usability testing questions. You don’t have to carry out the test, just prepare the questions. You should indicate what you are testing for in the Usability Testing.

Output – A Prototype and Usability Testing Questions. In Word format, uploaded it to GitHub. Due – Week 12 of the semester. 23 June 2023, by 11.59 pm.

# Task 5 – Discuss the ethical issue related to the software (20 marks)

Discuss and do a critical analysis of your software in these areas, privacy concerns, intellectual property rights, and effects on society.

Output – A report in Word format, uploaded to GitHub.

Due – Week 12 of the semester. 23 June 2023, by 11.59 pm.

# Submission

All tasks needed to be documented in Word format and submitted for SafeAssign checking (Links will be provided before the due date).

Upload the document and the SafeAssign report to your GitHub repository by each task due date. Due – It will be accessed on Week 14 of the semester. 07 July 2023, by 11.59 pm

# Marking Rubric for Continuous Assessment

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Marks Below 40%** | **Marks in the range 40 – 49%** | **Marks in the range 50 – 59%** | **Marks in the range 60 – 69%** | **Marks 70% and above** |
| **User Story** | User Story Mapping | User Story Mapping | User Story Mapping | User Story Mapping | User Story Mapping done and does capture most important activities of the system. The breakdown of the user story mapping is excellent and uses software that can assist that process (For example Trello compared to Ms.  Word). |
| **Mapping** | not done or User | done at a minimum | done and does | done and does |
| **(20 marks)** | Story copied/does  not match the exact | level and does not  capture the | capture several  important activities of | capture several  important activities of |
|  | system. | important activities of | the system. The | the system. The |
|  |  | the system. | breakdown of the | breakdown of the user |
|  |  |  | user story mapping | story mapping is good |
|  |  |  | can be improved. | and uses software that |
|  |  |  |  | can assist that |
|  |  |  |  | process (For example |
|  |  |  |  | Trello compared to |
|  |  |  |  | Ms. Word). |
| **Setting up a** | GitHub repository | GitHub repository | GitHub repository | GitHub repository exist | GitHub repository |
| **GitHub** | does not exist or | exist and some of | exist and most of the | and all of the required | exist and all of the |
| **Repository** | cannot be accessed | the required files are | required files are | files are available at | required files are |
| **(10 marks)** | or the required files  are not available at | not available at the  time of access. | available at the time  of access. However | the time of access.  However the dates for | available at the time  of access. The dates |
|  | the time of access. |  | the dates does not | some files does not | on the files follows |
|  |  |  | follow the required | follow the required | the required |
|  |  |  | deadline. | deadline. | deadline. |
| **Creating a** | The Class diagram | The Class diagram | The Class diagram | The Class diagram | The Class diagram |
| **Class diagram** | does not represent | and design pattern | and design pattern | and design pattern | and design pattern |
| **and design pattern selection (30 marks)** | the required solution (contains generic or non- related classes  such as admin), the design pattern | represent the required solution but in a very general and incomplete way.  Required classes in | represent the required solution in a partial way. A few  required classes in the design are not | represent the required solution in a satisfactory way. Most  required classes are declared. | represent the required solution in an excellent way. All  required classes are declared. |
|  | suggested is not | the design are not | declared. |  |  |
|  | suitable for the given | declared. |  |  |  |
|  | problem. |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Creating a Prototype User Interface and Usability Testing**  **(20 marks)** | No prototype were available or the measurement for the usability testing is not clear. | The prototype cover minimalist and trivial design (such as login) and the measurements for the usability testing are not clear. | The prototype cover adequate design and several measurements for the usability testing are not clear. | The prototype cover good design and most measurements for the usability testing are clear. | The prototype cover excellent design and all measurements for the usability testing are clear. |
| **Discuss the ethical issue related to the software**  **(20 marks)** | There is no discussion on the ethical issue or only the theories are pasted back for this component. | There is an attempt to discuss on the ethical issue but no critical  analysis was done | There is an attempt to discuss on the ethical issue with some critical analysis was done | There is an attempt to discuss on the ethical issue with good critical analysis. | There is an attempt to discuss on the ethical issue with excellent critical analysis. |

# Task 3 – Creating a Class diagram and design pattern selection (30 marks)

# Introduction

# The focus of this chapter is creating a class diagram and choosing the design pattern for the student business system for college.

# Class diagram

# The class diagram is one of the few diagrams of UML diagram. It shows the classes, attributes, and methods of the system’s structure. The class diagram also shows the relationships of the classes. The main purpose of this diagram is to help developers to understand the software requirements and describe detailed designs. For instance, during the designing phase, the developers can create a class diagram to understand the class structure and the relationships between the classes. (Ijaz, n.d)

# 

# This is the class diagram of the student business system for college. From the diagram the relation between each class. First, the CollegeStudent class has an association relationship with the website. While the website class has an aggregation relationship with the HelpService, Notification, CareerHub, Exam, Semester, OnlineEnrollment, Lecturer, StudentForms, Activity, and VirtualTour class. Next, from the class diagram, the Notification class has an association relationship with the CollegeStudent class. The Exam class has a composition relationship with the ExamTimetable class. The same goes for the Semester class which has a composition relationship with SemesterCalendar class. Not forgetting that the Activity class also has a composition relationship with the Organizer class. Finally, the VirtualTour has a composition relationship with the Facility class.

# Design pattern

# A design pattern is a solution to solve problems that software engineers faced during the process of designing an application. A design pattern is more of a description compared to a code. The description will mainly describe how the software engineers are going to tackle the problem. For readability purposes, practicing design patterns will help the final code to be more readable, as the solution’s design has been thoroughly evaluated. (Rahman, 2019)

# There are quite a few types of design patterns, but all these types can be categorized into 3 types. The three types of design patterns are Creational, Structural, and Behavioral. The goal of creational patterns is for class instantiation. These patterns can be used to create classes or objects. In designing structural patterns, one must consider the class’s structure and composition. The goal of these patterns is to improve the functionality of the class(es) without altering much of its composition. As for behavioral patterns, it is designed based on how the class(es) communicates with each other. (Rahman, 2019)

# 

# In this diagram, a more detailed version of the class diagram has been shown. From the diagram, it can be seen that all the classes have the attributes, data types, and operations included. In this section, a description of the operation will be discussed.

# In the CollegeStudent class, EnrollCourse() operation is used to enroll students into a course, while the getNotifications() operation is used to retrieve the notifications for the student. The getExamTimetable() operation is to retrieve the exam timetable for the student. The getSemTimetable() operation is to retrieve the semester calendar for the student. The getStudentForm() operation is to retrieve the student form for the student. Lastly, the SubmitEnrollForm() operation is to submit the student enrollment form.

# For the Websites class, the getWebTitle() operation is used to retrieve the Title of the website. The getWebDescription() operation is used to retrieve the description of the website. The getWebLink() operation is used to retrieve all the links of the website. Last, the getWebImg() operation is used to retrieve all the images of the website.

# For the HelpService class, the accessChat() operation is used to initiate the chat functionality with the help service. While the getHelpContactInfo() is used to retrieve the contact information of the help service staff.

# Moving on to the Notification class, the getNoTitle operation is to retrieve the title of the notification. The getContent() operation is to retrieve the content of the notification. The getDate() operation is to retrieve the date of the notification.

# Next, in the CareerHub class, the getJobName operation is to retrieve the available job’s name. The getJobDescription() operation is to retrieve the description of the job. The getAdvice() is to retrieve the advice for the job. The getListOfInternship() operation is to retrieve the list of available internships. The getResume() operation is to retrieve the resume template.

# For the Notification class, the getNoTitle() operation is to retrieve the title of the notification. The getContent() operation will retrieve the content of the notification and the getDate() operation will retrieve the date of the notification.

# In the Exam class, the getCourseName() operation is to retrieve the course name. The getCourse() operation is to retrieve the list of courses.

# For the Exam timetable class, the getCourseName() is to retrieve the name of the course. The getExamDate() operation is to retrieve the course exam date and the getExamLocation() operation is to retrieve the venue of the exam.

# Moving to the semester class, the getSemProgrammeName() operation retrieves the name of the program. The getSemProgramme() is to retrieve the list of programs.

# The semester calendar class has the getSemProgrammeName() operation that retrieves the name of the program. It also has the getSemCalendar() operation which retrieves the semester calendar of the program.

# The online enrollment class only has one operation which is the getLecEnrollForm() operation which retrieves the online enrollment form.

# For the lecturer class, the getLecName() operation is to retrieve the name of the lecturer, and the getLecEmail() operation is to retrieve the lecturer’s email. The getLecConsultationHour() operation is to retrieve the lecturer’s consultation hour and the getLecContactInfo() operation is to retrieve the lecturer’s contact information.

# The student form class has the getFormName() operation which retrieves the name of the form. Not only that, but it also has the getStudForm() operation which will retrieve the list of student forms.

# The activity class has the getActTitle() operation which retrieves the title of the activity and the getActDescription() operation which retrieves the activity description. Not only that, but it also has the getOrganizer() operation which retrieves the link to contact the organizer, and the getActPoster() operation which retrieves the poster of the activity.

# The organizer class contains the getOrgName() operation which helps retrieves the organizer’s name and the getOrgContact() operation which helps retrieves the organizer’s contact information.

# The virtual tour class consists of the getFacMap() operation which retrieves the image of the map of the campus and the getFacImg() operation which retrieves the image of the facilities. Not only that, but it also has the getFacName() operation that retrieves the facility’s name and the getFacLink() operation that fetches all the links.

# Lastly, the facility class contains the getFacImg() operation which retrieves the image of the facility, and the getFacName() operation which retrieves the name of the facility. The getFacDescription() operation retrieves the description of the facility.

# Design pattern selection of the project

# Based on the user story mapping of the student business system for college, in task 1 a design pattern is created. In this design pattern, there are three types of patterns that are going to be discussed.

# First, is the observer pattern which can be categorized as a creational pattern. The observer pattern is used to fulfill the need of receiving notifications. The notification system serves as the subject of the application's subject-observer connection, while college students serve as the observers. Students can sign up to get notifications and stay informed about crucial information. Ensuring that notifications can be provided to multiple observers without the requirement for direct dependencies, the Observer design enables a decoupled system.

# Secondly, is the Model-View-Controller (MVC) pattern which can be categorized as a structural pattern. The use of the MVC pattern is to structure the application and deal with multiple requirements. In this application model, view, and controller are the three interconnected parts used. For model, it holds the necessary information, such as academic schedules, enrollment information, activity details, lecturer information, student forms, facility locations, and exam timetables. As for the view, it is in charge of providing information to the students. On the other hand, the controller will manage the user interactions and update the model and view accordingly. The MVC pattern encourages the separation of concerns, improving the organization, reuse, and maintainability of code.

# Lastly, is the Chain of Responsibility pattern which can be categorized as a behavioral pattern. The Chain of responsibility design is applied to meet the requirement for the help service that can answer questions and provide guidance. The application creates a chain of objects, with each object in charge of responding to a certain class of inquiries or requests. A question submitted by the student will travel through the chain until it reaches an entity that can answer the question. This pattern provides flexibility in processing various query types and permits the installation or alteration of handlers without affecting the system.

# References

Rahman, S. (2019) *The 3 types of design patterns all developers should know (with code examples of each)*, *freeCodeCamp.org*. Available at: https://www.freecodecamp.org/news/the-basic-design-patterns-all-developers-need-to-know/ (Accessed: 19 June 2023).

Ijaz, U. (n.d) *What is a class diagram?* Available at: https://www.educative.io/answers/what-is-a-class-diagram (Accessed: 19 June 2023).