Faculty of Engineering, Environment and Computing

Assignment Brief 2018/19

Module Title Software Engineering	Ind/Group Individual	Cohort (Sept/Jan/May) Sept	Module Code 260CT
Coursework Title (e.g. CWK1) Lab assignment			Hand out date: 25/01/2019
Lecturer Yih-Ling Hedley			Due date: 04/04/2019 18:00
Estimated Time (hrs): 50 Word Limit*: 1,000	Coursework type: Assignment		% of Module Mark 50%
Submission arrangement online via CUMoodle: File types and method of recording: A PDF file and online submission Mark and Feedback date: 08/05/2019 Mark and Feedback method: A mark and feedback sheet on Moodle			

Module Learning Outcomes Assessed:

2. Describe and apply appropriate concepts, tools and techniques to each stage of the software development
3. Describe and apply design patterns and software components in developing new software
4. Use a Computer Aided Software Engineering (CASE) tool to produce design specifications and code for a software solution

Task and Mark distribution:

Task (based on the case study given on page 4)	**% of Coursework Mark allocated**
1. **System design:** To produce a four- layered architectural design model (in UML class diagrams using CASE tools) and report on ONE GRASP and ONE GoF patterns of your choice applied to the model *You should produce a short formal report with a maximum of 1,000 words and Coventry University Harvard referencing style. Note: GRASP: General Responsibility Assignment Software Patterns; GoF: Gang of Four	40%
2. **Software prototyping and testing:** a. To produce a prototype graphical user interface design for your chosen functionality (in screen shots with annotation) to explain the user requirements captured from the client (i.e. tutor) – 10% b. To provide the evidence of automated unit testing for your chosen functionality with annotation to explain the testing and the data sets used (which should include valid values and invalid values with boundary values where applicable) – 20% c. To produce the final version prototype system (in screen shots with annotation) to explain the working of your chosen functionality and the tutor's feedback – 30%	60%
Total:	**100%**

Additional notes on submission:

You need to choose ONE of the following functionality to design, implement and testing for the case study on page 4:

- membership management
- sports facility and class bookings and management
- sports facility and class maintenance and management
- schedule generation and management
- staff management
- invoices generation and management
- or any other new features …

For Task 1: The report should include the below:

- The final design class diagram (produced using a CASE tool, e.g. visual paradigm) for your chosen functionality as below:
  - each of the four layers with necessary design details based on the layered architectural design style
  - necessary design details for your chosen GRASP pattern
  - necessary design details for your chosen GoF pattern
- The discussion about the design:
  - research on the GRASP and GoF patterns used in your design model supported by referencing in Coventry University Harvard referencing style
  - as a formal academic piece of work (in a third-person style) with a maximum of 1,000 words

For Task 2:

- Screen shots of the prototype graphical user interface design for your chosen functionality with annotation including client's feedback to explain the requirements captured from the client (i.e. tutor). You are required to use the Graphical User Interface (GUI) features of an Integrated Development Environment (IDE, e.g. Netbeans) to produce your interface design only. **Please note that you are required to demonstrate your prototype to a tutor for feedback on the user requirements ONCE only in the lab sessions around mid-semester, starting from week 5. The design is required at this stage only.**
- Automated unit testing for a key method from your chosen functionality (in screen shots with annotation) to explain the testing and the data set used (including valid and invalid data values, with boundary data values where applicable). You must use an automated tool with assert statements (e.g. Junit in Netbeans) to carry out the testing.
- Screen shots of the final prototype system for your chosen functionality with annotation to explain the working of your functionality based on the tutor client's feedback. **Please note that you are required to demonstrate your working system prototype to a tutor for feedback ONCE only in the lab sessions before the submission deadline.** You should implement the chosen functionality in Java based on the design from Task 1 (i.e. the four- layers architecture style, the GRASP and GoF pattern of your choice). Your source code will be checked in the demo by the tutor.

Please label your full name, SID and tasks (i.e. Task 1, Task 2) clearly and save the document into ONE PDF document for submission. Please save your document in the format of **yourLastName_yourFirstName_yourSID**.pdf

(Please see the assignment supporting information and updates on Moodle page.)

Case Study: Sports and Leisure Village Registration and Booking System (VRB)

Sports and Leisure Village is a nation-wide club that offers a variety of facilities for sports (such as swimming and indoor tennis) and classes (such as yoga and fitness training) to members. The classes are run by qualified instructors working on a full-time or part-time basis. The club requires a new computer system to manage their members' registration and bookings.

New customers who wish to book the club facilities or classes should firstly be registered by a club advisor. The club offers different types of membership, including silver, gold and platinum. The silver membership is free to a new customer, who can then be upgraded to a gold membership if the customer has more than 10 bookings each year. The gold membership status for a member can only be maintained if the member has at least 10 bookings made with the club. However, the platinum membership will require a fee of £100 a year, but members will receive discounts on bookings and other benefits.

A club member can book a scheduled class (led by an instructor) or the use of facility with a club advisor. Prior to a booking, the advisor would need to be able to check the details of the classes, facilities and their availability. The advisor would also be able to access to the above information as and when required. The member can change or cancel a booking. The club facilities or classes can also be booked by individuals or as a group. The booking process will produce the schedules for classes and facilities booked. The copies of the schedules can be sent to the instructors and club managers on request.

Membership registration and bookings will be generally handled by the club advisors. However, the club managers have the responsibility to oversee and authorise the matters regarding staff and the club, which includes advisors, instructors, classes or facilities that are needed to be added to or updated on the system.

From the bookings of class activities, the club plans to use the information to generate invoices. The club advisor will request the invoices with a confirmation to be sent via a post or email once the payment is made. At the end of each month, a copy of the all invoices is transferred to the club's separate accounts system to be further processed.

The club would also welcome any new features recommended by the software developer.

(*Please note you can make any necessary assumptions or include additional details for the above requirements. You may also use dummy data where required to demonstrate the working program.)

Notes:

1. You are expected to use the CUHarvard referencing format. For support and advice on how this students can contact Centre for Academic Writing (CAW).
2. Please notify your registry course support team and module leader for disability support.
3. Any student requiring an extension or deferral should follow the university process as outlined here.
4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system.
5. If there are technical or performance issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours or the next working day if the deadline falls on a Friday or over the weekend period. This will be communicated via email and as a CUMoodle announcement.

Overall mark guidelines to students

0-39	40-49	50-59	60-69	70+	80+
Work mainly incomplete and /or weaknesses in most areas	Most elements completed; weaknesses outweigh strengths	Most elements are strong, minor weaknesses	Strengths in all elements	Most work exceeds the standard expected	All work substantially exceeds the standard expected

Marking Rubric		
GRADE	**System design (Task 1)**	**Software Prototyping and Testing (Task 2)**
First **≥70%**	An excellent architectural design model for the chosen functionality including four-layers, GRASP and GoF pattern with excellent design details and complexity A clear, consistent in-depth critical and evaluative argument for the design concepts and patterns applied to the project from a range of sources. Engagement with theoretical and conceptual analysis. Excellent presentation.	Prototype graphical user interface design using an IDE tool in screen shots with excellent annotation to explain the requirements captured from the tutor's feedback for the chosen functionality fully. Innovative design solutions. Automated unit testing using an automated tool with assertion statements and data sets, in detail and excellently executed and annotated Program output screen shots with excellent annotation of the final prototype based on the tutor's feedback to explain the chosen functionality fully, based on the design in Task 1. Innovative solutions.
Upper Second **60-69%**	An appropriate architectural design model for the chosen functionality including four-layers, GRASP and GoF pattern with key design elements but missing some minor details in very good level of complexity A generally clear line of critical and evaluative argument for the design concepts and patterns applied to the project is presented. Relationships between statements and sections are easy to follow, and there is a sound, coherent structure. Very good presentation.	Prototype graphical user interface design using an IDE tool in screen shots with very good annotation explain the requirements captured from the tutor's feedback for most of the chosen functionality Automated unit testing using an automated tool with assertion statements for a very good data set but not in full, very good annotation Program output screen shots with very good annotation of the final prototype based on the tutor's feedback to explain the chosen functionality for major parts of the design in Task 1
Lower Second **50-59%**	An appropriate architectural design model for the chosen functionality with some design elements (but missing a few elements from four-layers, GRASP and GoF pattern), in a reasonable level of complexity Some critical discussion, but the argument for the design concepts and patterns applied to the project is not always convincing, and the work is descriptive in places, with over-reliance on the work of others. Good presentation.	Prototype graphical user interface design in screen shots with good annotation to explain the requirements captured from the tutor's feedback for main functionality of your choice but not using an IDE tool Automated unit testing using an automated tool with limited data values and lacking of good annotation Program output screen shots with good annotation of the final prototype only to explain the chosen functionality for some parts of the design in Task 1, but not based on the tutor's feedback
Third **40-49%**	A simple architectural design model for the chosen functionality with a basic design (but missing most of the elements for four-layers, GRASP and GoF pattern), in a low level of complexity The work is descriptive with minimal critical discussion on the design concepts and patterns applied to the project and limited theoretical engagement. Acceptable presentation.	Prototype graphical user interface design in screen shots with some annotation to explain the requirements for the chosen functionality but no tutor's feedback captured or not using an IDE tool Automated unit testing using an automated tool with few data values and no annotation Program output screen shots with brief annotation of the final prototype only to explain the chosen functionality for some parts of the design in Task 1 but lacking details or no tutor's feedback
Fail **<40%**	No information or little design details for an architectural design model for the chosen functionality for four-layers, GRASP and GoF pattern. Descriptive with no evidence of theoretical engagement, critical discussion on the design concepts and patterns applied to the project or theoretical engagement. At the lower end displays a minimal level of understanding. Poorly presented.	Little evidence of prototype graphical user interface design in screen shots with no or few annotation, no tutor's feedback nor using an IDE tool to explain the requirements for the chosen functionality No Automated unit testing, not in an automated tool or with few data values with no annotation No or few program output screen shots with no annotation for final prototype, tutor's feedback nor based on the design required by Task 1
0%	No submission or late submission.	No submission or late submission.