Faculty of Engineering, Environment and Computing

Assignment Brief 2017/18

Module Title	Ind/Group	Cohort (Sept/Jan/May)	Module Code
Software Engineering	Individual	Jan	M18COM
Coursework Title (e.g. CWK1) UML Assignment			Hand out date: 26/01/2018
Lecturer Yih-Ling Hedley			Due date: 27/04/2018
Estimated Time (hrs): 50 Word Limit*: 2,000	Coursework type: Assignment		% of Module Mark 100%
Submission arrangement online via CUMoodle: File types and method of recording: A PDF file and online submission Mark and Feedback date: 15/05/2018 Mark and Feedback method: A mark and feedback sheet on Moodle			

Module Learning Outcomes Assessed:

1. Critically discuss different paradigms and process models to achieve and support the goals of software engineering for modern software development (Task 7)
2. Apply tools and techniques appropriate for each stage of the software process, from requirements to testing (Task 1-6)
3. Critically evaluate and apply general principles and patterns of software design in modeling software based solutions (Task 3)
4. Use a Computer Aided Software Engineering (CASE) tool to produce requirements models, analysis or conceptual domain models, software design models and automatically generate code for implementation and testing (Task 1-6)
5. Critically assess the role of software quality assurance with associated software quality attributes and metrics (Task 7)
6. Critically evaluate commercial risk and risk management along with social, legal, ethical and professional issues based on a given case study scenario (Task 7)

Task and Mark distribution:

Task (based on the case study given on page 3)	**Marks allocated**
1. To provide the detailed description (in a tabular format) for the requirement of ONE main use case (i.e. functionality) of your choice. • Use case description	10
2. To produce an analysis or domain model for your use case description. • Diagram: class diagram	10
3. To produce a design model that applies a layered architectural design style for your use case description. • Diagram: class diagram	10
4. To produce commented source code examples (e.g. in screen shots) with annotation to explain the code in relation to the design in Task 3, based on your preferred object oriented language (e.g. Java or C++), for your chosen use case description. • Source code	10
5. To produce program output (in screen shots) with annotation to explain the working of the system in relation to your use case description. • Working program	10
6. To produce a test plan for testing the methods of data classes only using black box testing. A minimum of 5 methods with different data values are required. • Test plan and testing results	10
7. To report on the following areas in relation to the given case study in a report: (500-750 words for each area, a minimum of 2,000 words in total) • A chosen software process model for its development and management and how the goals of software engineering are supported • Commercial risks and their risk management • Social, legal, ethical and professional issues • The quality of the software system based on two quality attributes of your choice from the ISO/IEC 25010 product quality model	40 (10 marks for each topic area)
Total:	**100**

Case Study: Activ Sports Clothing Sales and Stock System

Activ Sports Clothing (ASC), a retailer selling men's and women's sports clothing, needs a new software system to handle sales as well as their stocks. In addition to clothing, they also sell footwear and other accessories. The company currently employs sales assistants and a manager to manage their business.

When a customer checks out with their items, a sales assistant uses the system to process the sale. The process includes scanning each of the items, calculating their total amount (based on the price of each item stored on the system), debiting the stock level for each item, adding the sale to the sales ledger.

The store accepts cash, debit or credit cards, but card payments are handled by a separate payment and credit checking system. The customer can request a gift wrap service with additional charge on the check-out, otherwise, no charge for a standard store bag.

The manager can access and view staff activities on tills and the sales activities, which can be generated for any day or weeks. The manager can also check the total value of sales for a given period and acquire a breakdown of sales by item and category of items. The manager is responsible for altering the pricing information for all items in the system, adding new stock or deleting a discontinued stock line from the system.

At the end of each business day, prior to requesting the software system to generate orders for new stock, the manager needs to view the sales activities of the day. The new orders are entered and copied directly to the online ordering systems of the supplier(s) concerned which a confirmation of orders received is required.

The sales information is also sent to a separate company account system for an update on the accounts. An account report from the system can be produced on request by the store manager.

The company would welcome any additional features from the software developer to help them improve the running of their business.

*Please note that you may make assumptions on the above case study where necessary.

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	Use Case Requirement (Task 1)	Domain Model (Task 2)	Design (Task 3)
≥70%	An excellent and detailed description (in a tabular format) for the requirement of ONE main use case (i.e. functionality) of your choice.	An excellent domain model for the chosen functionality including the details and complexity	An excellent architectural design model for the chosen functionality including a layered architectural style and a GRASP pattern with excellent design details and complexity
60-69%	An appropriate description (in a tabular format) for the requirement of ONE main use case (i.e. functionality) of your choice, which includes major steps and alternative flows.	An appropriate domain model for the chosen functionality including the main attributes, associations and multiplicities (without the design details)	An appropriate architectural design model for the chosen functionality including a layered architectural style and a GRASP pattern with key design elements but missing some minor details in very good level of complexity
40-59%	A satisfactory description (in a tabular format) for the requirement of ONE main use case (i.e. functionality) of your choice, which includes some steps but lacking details or missing alternative flows with system responses.	A satisfactory domain model for the chosen functionality including some of the attributes, associations and multiplicities (or may include the design details)	An appropriate architectural design model for the chosen functionality with some design elements (but missing a few elements from a layered architectural style and a GRASP pattern, in a reasonable level of complexity
<40%	No information or few description for the requirement of ONE main use case (i.e. functionality) of your choice.	No information or few details for the attributes, associations and multiplicities of the chosen functionality in the domain model	No information or little design details for a layered architectural style and a GRASP pattern for the chosen functionality.

GRADE	Source Code (Task 4)	Testing (Task 5)	Program Output (Task 6)	Reports (Task 7)
≥70%	Commented source code with excellent annotation to explain the code in relation to the chosen architectural design and excellently presented	A test plan for a variety of data sets with details and excellently executed with results recorded	Program output screen shots with excellent implementation and annotation in relation to the chosen functionality fully	A clear, consistent in-depth critical and evaluative argument from a range of sources. Engagement with theoretical and conceptual analysis.
60-69%	Commented source code with very good annotation to explain the code in relation to the chosen architectural design for the key elements with a very good presentation	A test plan with a very good set of data values and but not in full with results recorded	Program output screen shots with very good implementation and annotation in relation to the chosen functionality for major parts	A generally clear line of critical and evaluative argument. Relationships between statements and sections are easy to follow, and there is a sound, coherent structure.
40-59%	Commented source code with good annotation to explain the code in relation to the chosen architectural design but missing some key elements of the design	A test plan with some data values but lacking of results recorded	Program output screen shots with good implementation and annotation in relation to the chosen functionality but missing important parts	Some critical discussion, but the argument is not always convincing, and the work is descriptive in places, with over-reliance on the work of others.
<40%	Commented source code with no or few annotation to explain the code in relation to the chosen architectural design	No test plan or little testing results	No or few program output screen shots with no annotation in relation to the chosen functionality	Descriptive with no evidence of theoretical engagement, critical discussion or theoretical engagement. At the lower end displays a minimal level of understanding.