**Software Design (International Year 1)**

**Coursework Brief**

|  |  |  |
| --- | --- | --- |
| Module Title: | XFX1041-1920 Software Design | IY1 |
| Title of Coursework: | 1. Authentication Manager 2. Game of Tic Tac Toe 3. Point of Sale system 4. \*\*Layout manager (Advanced) | |
| Hand-out date: | 7th May 2020 | |
| Hand-in date: | 5th June 2020 | |
| Weighting within the module | 50% of total module mark | |
| Word limit / presentation criteria | Project deliverables include texts, diagrams and codes | |
| Learning Outcomes to be assessed(from module spec.) | K1. Formulate a range of algorithms for manipulating data structures.  K2. Recognise the concept of correctness and verification and validation in testing computer programs.  K3. Describe the semantic concepts of a variety of abstract modelling techniques relevant to modern computing.  A1. Apply abstract modelling techniques to a problem scenario to build models from a number of perspectives, in order to capture information and to devise appropriate event-driven and algorithmic solutions.  A2. Analyse problems and develop algorithms using a design approach (e.g. functional decomposition or object-oriented design).  A3. Develop a basic object-oriented program, applying designed techniques and algorithms. | |

# Coursework Details

**Purchase and Sales Management System**

The student will be required to choose one out of four system specifications. The fourth project being an advanced project which will be granted to the student based on the tutor’s recommendation. To meet the marking criteria for the course-work all systems must have extensive well written documentation of the following elements:

1. Use of object-oriented analysis, design and implementation of classes obtained
2. Use of use case analysis to determine system requirements
3. Use of activity diagrams to model system behaviour and specific system outcomes.
4. Use of sequence diagram to demonstrate implementation system component interaction
5. Use of a class relationship diagram that reflects outputs of the use case, activity and sequence diagram
6. A text user interface should be implemented for the system using the ncurses library
7. The use of low-level algorithms, programs, functions, pseudocode and flowcharts.
8. Unit tests should be written for a minimum of 5 functions within the program.
9. Working functional and appropriately commented program that demonstrates a direct implementation of the system design

The details of the system specifications are as follows:

**Submission**

An electronic copy of your project (via the assignment submission area on Brightspace). The files to be submit will include

1. C++ source codes compressed into a zip file.
2. The written report using Microsoft word or as a pdf file.

For electronic copy, you should submit your work in .doc or.pdf format. Please name this using the convention YourInitial\_YourSurname\_SD\_Coursework.doc (or .pdf or zip) (so that if your name is Steve Jobs you would name it as S\_Jobs\_SD\_Coursework). If your work cannot be submitted as a single document then you should submit a single zip file containing all your documents. Please make sure that you check your work carefully following submission.

Your solution must be original. Please make sure that you are familiar with the regulations regarding plagiarism and late submissions procedure.

**Marking Criteria**

* **UML Diagrams**: This includes any diagrams or documents that you have produces to plan and understand your system. You may have employed flow chart diagrams for each activity/function of your system, produced a use case, activity and sequence diagrams. These must be well documented to show the reasoning behind each element in the diagram. In order to produce a solution to this problem you are strongly encouraged to apply techniques of analysis in order to break the problem down into manageable and solvable elements. These will involve both high-level systems and subsystems and low-level methods and sub procedures involving pseudocode algorithms and flowcharts. [50%]
* **Functionality**: This criterion focuses on the extent to which your project achieves what it is required to do, and the way by which it goes about doing it. The implementation must be object-oriented involving multiple related classes to implement the software designs and all the functionality which are described in the specification brief. There will need to be unit tests for at least 5 methods having a wide range of inputs and outputs. You are likely to receive extra marks if you illustrate your creativity and extend the system capabilities defined by the project brief. Consistency in design and implementation is also a requirement for this criterion to be met. [40%]
* **Documentation**: This criterion considers the degree to which your assignment gives the impression of a coherent, thoughtful, thorough and sustainable project. Consistency throughout the project in terms of your visual approach, your use of terms and the linkage between the statement of goals in your planning and their final realisation. This also includes in-code commenting and use of a consistent and readable coding style. [10%]