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. Automated Teller Machine (ATM) simulation 4. Puzzle game 5. Any System of interest	
Hand-out date:	10th July 2020	
Hand-in date:	7th August 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

Assessment Criteria

The student will be required to choose one out of four system specifications or any system of interest and must be discussed with tutor. 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. Use of wireframe models and/or package diagrams
7. A text user interface should be implemented for the system using the ncurses library
8. The use of low-level algorithms, programs, functions, pseudocode and flowcharts.
9. Unit tests should be written for a minimum of 5 functions within the program.
10. Working functional and appropriately commented program that demonstrates a direct implementation of the system design

The details of the system specifications are as follows:

1.   Authentication Manager

The authentication manager system to be developed allows the user to login into the system by authenticating the user name and password against a list of registered members saved on a file. The system will be able to determine if the login was successful or failed based on the accurate result of the login check. The authentication manager will also allow users to register and add to the list of registered members on the file system. Finally the system will permit the user to reset his password should he choose to do so while logging in using a 'forgot password' button. While registering, the user must also provide his mother's maiden name and the system must store this information that will be required in order for the user to change their password.

2.   Tic Tac Toe game

The program should allow a user to play a game of tic tac toe against the computer. The user should be able to select 3 modes of game play easy, intermediate and hard. For the easy mode, the computer will place moves randomly on the board. For the intermediate mode the computer will also place moves randomly on the board but prioritize blocking the users win moves. For hard the computer will play with the aim to place winning moves on the board based on a random selection of winning sequences.

3. Automated Teller Machine (ATM) simulation

Given 3 trials, the user is able to see his balance by entering a four-digit pin that must NEVER be displayed on screen but masked by the Asterix (*) character. A list of pins stored on the file system must be loaded to verify the pin. The user should be able to withdraw funds below a set limit and cannot redraw more than 3 times within a space of 5 minutes. The balance from each pin should be determined each time the program is run and is a random value between 0 and 100. The user should also be allowed to change his pin at any time which should be reflected in the file system.

4. Puzzle game

The puzzle game is a board comprising a 3x3 matrix of numbers that the player is asked to arrange from left to right then top to bottom in order. The board, at the beginning has numbers scrambled from 1 to 8. There is a free space on the board that the user can swap an adjacent piece horizontally or vertically in order to rearrange the numbers on the board in the diagram below.
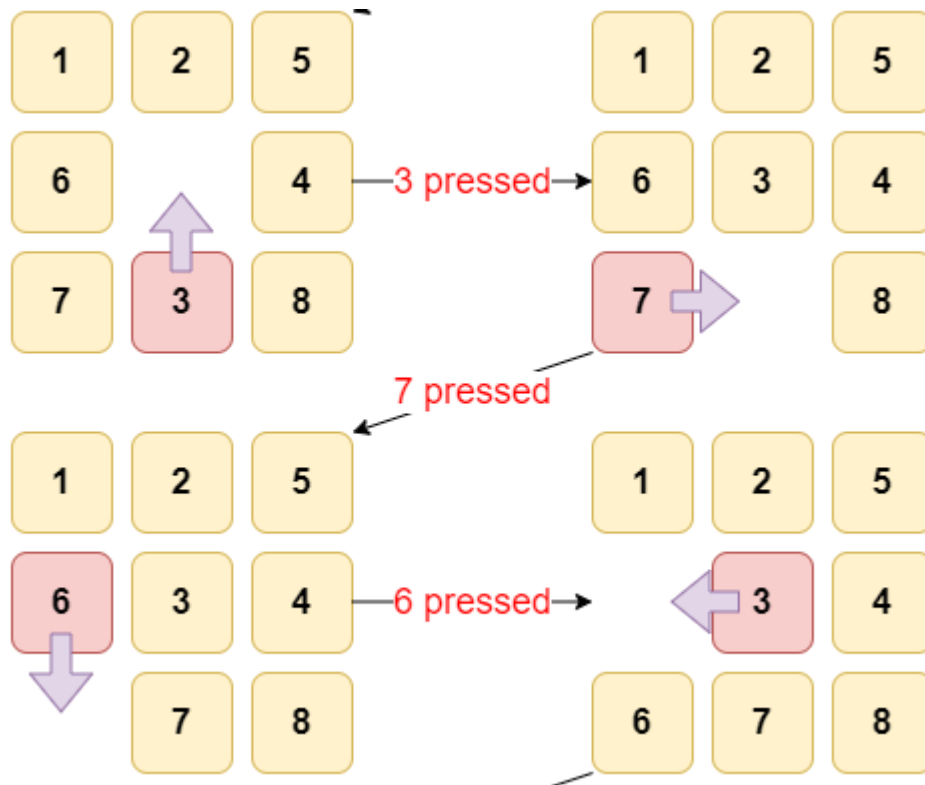
Your task is to design and implement this game to ensure that the player can only move in the said directions to achieve the winning rearrangement.
Ensure that you meet all the criteria mentioned above

5. Any system of interest
You can choose any system of your interest provided such system meets all the criteria. This must be discussed and agreed with you tutor.

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%]