CSY2030: Systems Design & Development

Agreed Date for late submission:	**Sunday 30 April 2017**	Module Tutor:	Apkar Salatian
Student ID:			
Video Link:			

Aspect (& weighting)	Excellent A	Good B	Satisfactory C	Needs much more work F
Design of System (Use Case Diagram, Class Diagram) (15%)				
Functionality (45%)				
Testing (black box and white box testing) (15%)				
Code Layout and Documentation (5%)				
Code Quality and Efficiency (10%)				
Demonstration (10%)				

Specific aspects of the assignment that the marker likes:	Specific aspects of the assignment that need more work:

Tutor's Signature:		Date:		Grade:	

DRAFT

The University of Northampton Policy on Plagiarism & Mitigating Circumstances will be strictly implemented. By submitting this assignment you are asserting that this submission is entirely your own/individual work.

THE UNIVERSITY OF NORTHAMPTON

MODULE: Systems Design & Development **2016-2017**

MODULE CODE	LEVEL	CREDIT VALUE	CO-ORDINATOR
CSY2030	5	20	Apkar Salatian

DRAFT

Assignment Brief

Assignment title:	CSY2030 Individual Project to create a GUI based Banking System
Weighting:	50%
Deadline:	23:59 on 30 April 2017
Resit Date	TBA

Brief:
Design, implement and test a stand-alone banking system using object-oriented principles in Java and accessed using a graphical user interface. Design should include use cases and a class diagram while testing should include black box and white box testing

Here are the specific requirements:

The *Bank of Northampton* would like a system for their staff to open various bank accounts for their customers and service them using a graphical user interface. Staff can access the graphical user interface using a user name and password.

The *Bank of Northampton* has the following 3 distinct bank accounts with associated features:

1. *Current Account* – each current account has a sort-code, account number, the date it was opened, overdraft amount and a balance. Customers can deposit to and withdraw monies from a current account and will incur a daily charge of £10 if they exceed their overdraft limit
2. *Deposit Account* – each deposit account has a sort-code, account number, the date it was opened, monthly interest rate and a balance. Customers can deposit to and withdraw monies from a deposit account and interest can be calculated on the account based on the average balance for the month.
3. Individual Savings Account (ISA) - each ISA account has a sort-code, account number, the date it was opened, annual interest rate, term of account (e.g 1 year, 2 year etc) and a balance. Customers can deposit monies to an ISA account to a limit of £15,240 but no withdrawals can be made. The interest earned can be calculated from the date it was opened.

For each customer we record their name, address, phone number, email address and date of birth. We also record their user name and password so they can log into the system to query their accounts. A customer can have at most 2 current account, 1 deposit account and 1 ISA account.

Customers can access their accounts and query them through the same graphical user interface using their user name and password.

Minimum System Requirements (for grades upto D-):
The system must, using a GUI, allow staff to do the following by logging on with their user name and password:

1. Open a new account for a customer. When a customer opens a new account a check is made to see if they are an existing customer or not. If they are an existing customer then it is just a matter of opening the account they want. If they are not an existing customer then a new customer has to be created as well as opening the account they want.
2. Service all the accounts in the following ways:
    a. Deposit monies
    b. Withdraw monies
    c. Query account balances (note account balances will be based on the type of account e.g if it is a current account then the balance will reflect any charges the customer incurred and in the case of the deposit and ISA accounts it will be based on any interest earned)
    d. Query account details based on customer details if the customer has forgotten his/her account number

The system must also, using a GUI, allow customer to access and query their accounts using their user name and password.

DRAFT

Additional System Requirements (for grades C to A+):

3. Use of user name and passwords
4. Exit the system and write all objects to file so they can be reloaded when the system is run again
5. An intuitive GUI
6. Allow staff to query the option to search all current accounts to see if the balance is over £15,240 and list email addresses of these customers so they can be contacted to open an ISA
7. Appropriate exception handling
8. Use the Model View Controller
9. Use of Collections

Deliverables

All requirements (A, B and C below) **MUST** be delivered to achieve a passing grade for this assignment:

A) Technical Report

The report should consist of the following:

- Standard Front Sheet (Attached to this assignment [copy and paste the 1st page of this document] - DO NOT include whole assignment brief).
- Table of contents.
    - Introduction/ Problem statement for the task.
    - Analysis of requirements and assumptions for the task.
    - Design of the system expressed in the form of UML class diagrams and use cases.
        - Brief explanation of the overall design (classes, responsibilities and collaborations)
        - A description of the system (main components and functionality). Include screenshots of the system in different modes of operation. Also, give clear instructions on how to run/use your system (user guide).
- Evidence of Testing:
    - Test logs providing information of all the tests carried out (including any failed tests for functionality not implemented, screenshots, unit tests, etc.) – this is black box testing
    - List of any bugs and/or weaknesses in your system (if you do not think there are any, then say so). Bugs that are declared in this list will lose you fewer marks than ones that you do not declare.
    - List of any bugs that were discovered and, if fixed, what was done to fix them.
    - White Box tests of 2 methods which have at least 2 decision points
- Conclusion/Recommendations (list of additional features you would have liked to implement)
- References (use Harvard referencing)
  [If you have borrowed some code from elsewhere (e.g. from a book or some resource on the web you **must** indicate clearly what they are and include references).

The technical report should be saved as a word file in the following format *<your-name>-<your student id>-technical-report.doc* e.g *fred-smith-12345678-technical-report.doc*

DRAFT

B) Source Code

The source code must be well documented with necessary comments. Consistent and clear indentation of the code is also important. Source code needs to be submitted in two forms:

(i) As a single ZIP archive (.zip file consisting of all ".java" files, unit tests, data files, executable jar). It must contain all the code and files required to run the application. The zip file should be saved in the following format *<your-name>-<your student id>-java-zip-file.zip* e.g *fred-smith-12345678-java-zip-file.zip*

(ii) A commented full listing of each java file in a separate Word document. The word file should be saved in the following format *<your-name>-<your student id>-java-files.doc* e.g *fred-smith-12345678-java-files.doc*

B) Video Demonstration

In addition to the report, you **must** provide a video demo of your assignment. The demo should be 10-15 minutes long (no longer than 15 minutes), and should cover all of your work in a logical way. Your voice needs to be clear for the marker to hear. Please include a walkthrough of using the software and emphasise the key features. You may be called in for a viva-voce should there be any doubts on the originality (plagiarism aspects) of your submission.

Submission Procedure:

- E-Submission of documents through Turnitin on NILE as TWO separate WORD documents. [Document 1 is your Technical Report and Document 2 is your Java code listing]
  To do this, go to the NILE site for this module and use the link labelled 'Submit your work'.

- E-Submission of a single ZIP archive that contains all the source code files (.java), unit tests, data files and executable (jar). To do this, go to to the NILE site for this module and use the link labelled 'Submit your work'. Clicking on the link (*SourceCodeEsubmission*), will take you into the submission form, where you can upload your ZIP archive using the 'Attach File' button (*Browse for Local File*). Finally, click the *Submit* button.

- When submitting your video demonstration, please use Kaltura or YouTube to upload the video. If you use YouTube please put a link on the Standard Front Sheet of your report. Kaltura is integrated in NILE and is very easy to use. For instructions on using Kaltura, there is a PDF document
  here: *https://nile.northampton.ac.uk/bbcswebdav/orgs/Help/KalturaMediaspace/MediaSpace%20Student%20Guide%202015%20-%20Version%2013.pdf*.
  To record your demonstration you can use Kaltura's software available from here: *https://northampton.mediaspace.kaltura.com/capturespace/launch/create*
  To record your screen once you have the software you can go to http://video.northampton.ac.uk/ and select Add New > Record a Lecture. This will open the Kaltura program and you can choose to record the screen. In the options you can choose to record sound as well. Once you have recorded your video, press "Done" and you can insert the video using the "Mashups" option in the submit your work area as detailed
  here: *https://nile.northampton.ac.uk/bbcswebdav/orgs/Help/KalturaMediaspace/MediaSpace%20Student%20Guide%202015%20-%20Version%2013.pdf*

DRAFT

Assessment Breakdown

Assessment Criteria:
Design of System (Use Case Diagram, Class Diagram) - 15%
Functionality - 45%
Testing (black box and white box testing) - 15%
Code Layout and Documentation (5%)
Code Quality and Efficiency (10%)
Demonstration (10%)

DRAFT