THE UNIVERSITY OF NORTHAMPTON

MODULE: Systems Design & Development **2019-2020**

Assignment Brief

Assignment title:	CSY2030 Individual Project to create a GUI based National Property Sales System
Weighting:	50%
Deadline:	Sunday, 3rd May 2020, by 23h59.
Resit Date	TBA

Brief:
Design, implement and test a stand-alone Property Sales 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:

Assume a new *national property sales company* would like a java system for their branches across the country to keep track of properties for sale in their areas through a graphical user interface. An administrator adds new branches to the system and branch secretaries maintain the details of all the properties in his/her branch. An administrator has access to the system through a single user name and password. Likewise, branch secretaries access the system through a different user name and password which is stored with each branch (see below).

An administrator will add new branches to the system. Each branch has a name (e.g *Northampton, Glasgow, Manchester* etc), address, phone number, email address and a web address. For each branch we also store a user name and password to allow a branch secretary to log in to maintain the properties at their branch. Administrators can also modify and delete branches.

When a branch secretary logs on to the system given the user name and password provided by the administrator then he/she will be able to add/modify/delete the properties for their branch as well as view them and record them as sold when a purchase is made.

There are 2 distinct types of properties that are recorded for each branch by their secretary:
1. *houses* – each house has an address, number of rooms, selling price, sold price, number of floors, whether it has a garden or not, and whether it as a garage or not
2. *flats* – each flat has an address, number of rooms, selling price, sold price, which floor it is on, and its monthly charge

As well as adding properties, secretaries can also modify and delete properties. Whenever a property is sold then the sale price will be recorded for the property and removed from the sales list. All sold properties and their sold price can be viewed by the system.

Minimum System Requirements (for grades upto D-):
The system must, using a GUI, allow administrators to do the following:
1. Log onto the system with a user name and password
2. Exit the system
3. Add a new branch
4. Change details of a branch
5. Delete a branch

The system must, using a GUI, also allow branch secretaries to do the following:
6. Log onto the system with a user name and password that is stored with each branch
7. Exit the system
8. Add properties
9. Modify properties
10. Delete properties

11. Show details of all the houses
12. Show details of all flats
13. Query a property by their address
14. Show details of all the details of all the properties (including their type) in their branch
15. Record the sales prices of property when it is sold and remove from sales list
16. View all sold properties

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

17. Exit the system and write all objects to file so they can be reloaded when the system is run again
18. An intuitive GUI
19. Appropriate exception handling
20. Use the Model View Controller
21. Use of Collections

Note that object serialisation must be used and no marks to be given for using databases and SQL queries.

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:

- Your name, student identification number and web link to your demonstration
- Design of the system expressed in the form of a UML use case diagram and a class diagram
- A description of the system (main components and functionality) in the form of a user manual. 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
  - White Box tests of 2 methods which have at least 2 decision points
  - 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.
- 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*

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 - -zip-file.zip e.g fred-smith-12345678-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 --java-files.doc e.g fred-smith-12345678-java-files.doc

C) 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. Provide the link to your video demonstration in your technical report.

Use Kaltura for recording your video demonstration. 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

It is your responsibility to ensure that the permissions of your video are correct otherwise you will lose 10% which is the allocation for this part of the submission.

Below is a link to "how to upload video to NILE".

https://mypad.northampton.ac.uk/nilefaq/2015/07/28/what-is-my-media-and-the-media-gallery/

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.

Assessment Breakdown

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

Marking Criteria

The grade for this assignment will form 50% of the overall assignment grade for the module. The Standard Front Sheet of the assignment gives an indication of how the marks are split. In general, the following criteria will act as a guide to what you should expect:

A **bare pass (D)** will require you to produce a working system that incorporates all of the basic requirements. You must also include all the deliverables (design, evidence of testing, etc.) stated in the assignment.

A **good pass (B to C)** will require you to produce a working system that incorporates all the basic requirements and some additional features. The code should be well documented (code layout, useful comments, etc.) with evidence of testing, error handling and error recovery. Adherence to object-oriented principles is a must.

A **very good pass (A)** will require you to produce a working system that incorporates all the basic requirements with a significant number of additional features. Any new innovative features that are useful will be considered. The code should be well documented (code layout, useful comments, etc.) with evidence of testing, error handling and error recovery. Additionally, the design and implementation of the system should be of good quality and efficiency with adherence to object-oriented principles.