|  |  |  |  |
| --- | --- | --- | --- |
| **CSY2030: Systems Design & Development** | | | |
| Agreed Date for late submission: | **Sunday 22 April 2018** | Module Tutor: |  |
| Student ID: |  | | |
| Video Link: |  | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aspect (& weighting) | | Excellent  A | | Good  B | | | Satisfactory  C | | Unsatisfactory  D | | 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%) | |  | |  | | |  | |  | |  | | |
| Report (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: | |  |  |

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  **2017-2018**

|  |  |  |  |
| --- | --- | --- | --- |
| **MODULE CODE** | **LEVEL** | **CREDIT VALUE** | **CO-ORDINATOR** |
| CSY2030 | 5 |  |  |

**Assignment Brief**

|  |  |
| --- | --- |
| **Assignment title:** | CSY2030 Individual Project to create a GUI based Vehicle Hire System |
| **Weighting:** | 50% |
| **Deadline:** | 22 April 2018 |
| **Resit Date** | TBA |

**Brief:**

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

Here are the specific requirements:

Assume the *Northampton* *Vehicle Hire Company* would like a system for their staff to open various accounts for their customers and hire them out various types of vehicles (cars, mini-buses and lorries) using a graphical user interface. *Northampton* *Vehicle Hire Company* would also allow their customers to view their accounts and see what vehicles are available for hire.

The *Northampton* *Vehicle Hire Company* has the following 3 distinct types of vehicles with associated features in their fleet for hire:

1. *Car* – for each car we store their make, model, top speed, registration number, daily hire rate, fuel type (diesel or petrol) and number of doors.
2. *Mini-Bus* – for each mini-bus we store their make, model, top speed, registration number and daily hire rate. Mini-buses also have a seating capacity. For mini-buses we need to know the intended number of passengers the customer will want – if this figure exceeds the seating capacity then the mini-bus will not be hired out.
3. *Lorry* – for each lorry we store their make, model, top speed, registration number and daily hire rate. Lorries also have a loading capacity. For lorries we need to know the intended load weight the customer will want the lorry for – if this figure exceeds the loading capacity then the lorry will not be hired out.

For each customer we record their customer identification number, name, address, phone number, email address and date of birth, user name and password. A customer can have any number of cars, mini-buses and lorries out on hire.

**Minimum System Requirements (for grades upto D-):**

The system must, using a GUI, allow staff to do the following:

1. Log onto the system with a user name and password
2. Exit the system
3. Add vehicles to the system as the fleet grows
4. Remove vehicles from the system when the fleet decreases
5. Add new customers to the system
6. Service all the customer accounts in the following ways:
   1. Hire out vehicles from the fleet to customers
   2. Return vehicles from customers to the fleet
   3. Query vehicles i.e view all details of a specified vehicle
   4. Query customer accounts i.e view details of a specified customer and details of the vehicles (if any) he/she currently has hired out

The system, must, also using a GUI allow customers to

1. Log onto the system with their user name and password
2. Exit the system
3. Query his/her account i.e see his/her details and all the vehicles he/she has out on loan
4. See what vehicles are available for hire (this list should exclude the vehicles that are out on loan to the customer and other customers)

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

1. Exit the system and write all objects to file so they can be reloaded when the system is run

again

1. An intuitive GUI
2. Ensure vehicles that out on loan cannot be hired out
3. Appropriate exception handling
4. Use of the Model View Controller
5. Use of Collections

**Deliverables**

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

1. **Technical Report**

The report should consist of the following:

* Standard Front Sheet (Attached to this assignment-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*

**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:

1. 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>-zip-file.zip* e.g *fred-smith-12345678-zip-file.zip*
2. 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*

**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.

**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%2 0Student%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%2 0Student%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/](https://webmail.northampton.ac.uk/owa/redir.aspx?C=g2vuQ3A8GDDMpUneWcJFtHCB1bhGbq1UVpgaHTTFmucsPRJcE53UCA..&URL=https%3A%2F%2Fmypad.northampton.ac.uk%2Fnilefaq%2F2015%2F07%2F28%2Fwhat-is-my-media-and-the-media-gallery%2F)

## 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 – 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.