****

**School of Computing Science and Digital Media**

Faculty of Design and Technology

**Coursework Assignment**

|  |  |
| --- | --- |
| Surname | Tait |
| First name | Andrew |
| Matriculation Number | 1504693 |
| Contact phone number | 07768454608 |
| Course + Year | Computing: Application Software Development – Year 3 |
| Module Co-ordinator | Garry Brindley |
| Module Number + Name | CM3008 OOP / CM2015 OOSD |
| Coursework Title | Mail Service GUI |
| Coursework Part | 1 |
| Due Date | 22nd January 2016 at 12 noon |

**Declaration** \*\* *This* ***must*** *be affirmed by signing below*

**I confirm**

* **that the work undertaken for this assignment is entirely my own and that I have not made use of any unauthorised assistance**
* **that the sources of all reference material has been properly acknowledged.**

|  |  |
| --- | --- |
| Student Signature | C:\Users\Andrew\Downloads\DSC_0658.JPG - Andrew Tait |
| Date Submitted | 22/01/2016 |

|  |  |
| --- | --- |
| **Marker's Comments** | |
| **Marker** | **Grade** |

Table of Contents

[Statement of Compliance 3](#_Toc441209407)

[Design and Documentation 3](#_Toc441209408)

[Implementation 3](#_Toc441209409)

[Testing 3](#_Toc441209410)

[Design + Documentation 4](#_Toc441209411)

[Testing 6](#_Toc441209412)

Table of Figures

[Figure 1 Wireframe: Main Frame GUI 4](#_Toc441209414)

[Figure 2 Wireframe: Input Option Pane 4](#_Toc441209415)

[Figure 3 Wireframe: Add Van to fleet 4](#_Toc441209416)

[Figure 4 Wireframe: Select Mail Item 5](#_Toc441209417)

[Figure 5 Wireframe: Add/Edit Mail Item 5](#_Toc441209418)

# Statement of Compliance

## Design and Documentation

All the designs such as wireframes have been created in Microsoft Visio and are present within this document. Each wireframe has got a description of its purpose and why the components were chosen. Exported Images can be found within the documentation folder if they are unclear within this word document.

## Implementation

* All Implementation tasks have been attempted and completed (Task1 to Task 10).
* NetBeans IDE version 8.0.2 was used for implementation, running on Windows 10.
* Input validation is present when necessary including error checking and handling.
* Each task is commented and documented clearly within the GUI framework.
* Any provided code/classes have not been altered except the main runnable class.
* However there was an error with the loading system later on in the development when loading an incompatible file as it would display an error on the NetBeans output terminal but would not affect the GUI itself.
* The Van details sorted functions work s but only in descending order. It also works without altering the order of any other functions.
* Saving a fleet to a file can be a bit temperamental as in most test cases it works but sometimes the new saved file cannot be read back in. I wasn’t able to replicate these errors frequently enough to produce a possible solution to the issue.

## Testing

All the tasks and their corresponding functionality have been tested through a variety of test cases.

The test cases for each task have been validated and verified by normal, extreme and exceptional inputs and conditions.

Any unexpected results are documented.

Each test case includes:

* + Test Number
  + Its Purpose
  + Inputs
  + Excepted Outputs
  + Final Result

# Design + Documentation

For the graphical user interface designs, I used Microsoft Visio to create wireframes. These wireframes formed the basis of the Frame and Dialog windows when designing in NetBeans graphical editor.

|  |  |
| --- | --- |
| Wireframe Design | Purpose |
| Figure Wireframe: Main Frame GUI | **The main form for the program’s GUI.**  I used dropdown menus are the main method of navigation for the GUI.  The output from each function is displayed in a text area which automatically scrolls based on the output’s total length. |
| Figure Wireframe: Input Option Pane | **Input option pane:**  These are built into the Swing library itself but made a diagram for layout purposes.  I used Input option panes for getting the van id from the user in searching, adding and editing functions. |
| C:\Users\Andrew\Documents\RGU\CM3 Computing (Application Software Development)\CM2015 Object Oriented Software Design\Coursework\AddVantofleet.PNG  Figure Wireframe: Add Van to fleet | **Add van to fleet dialog:**  This dialog was the most simple to design and implement and it only required the driver details (first + surname) from the user. |
|  |  |
| Figure Wireframe: Select Mail Item | **Select Mail Item Dialog:**  For editing a mail item the user would first had to enter a van id in a popup window. After they will brought to this Dialog window.  All the mailitems will be populated in the combobox.   The selected mailitem’s index number will be sent to the edit mail item dialog when the user presses the OK button. |
| Figure Wireframe: Add/Edit Mail Item | **Add & Edit Mail Item Panel:**  The adding and editing of mail items were fairly similar in terms of design as they had the same fields to deal with.  On the adding form the user enters the address details and chooses the mail item type (parcel, express parcel, letter) and that mail item is then added the selected van.  The editing works by populating the mail item data in the textboxs and updates the mailitem when the user presses OK. |

# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test  Number | Reason | Inputs | Excepted Outputs | Results |
| 1 | Load Supported File | .txt file | File Loads Successfully | File Loads Successfully with confirmation message |
| 2 | Load Unsupported File | .png file | Program Crash | No effect on GUI but error messages displays on netbeans output terminal |
| 3 | Save Fleet to file | Fleet data | .txt file Saved | Text file saved successfully with confirmation message |
| 4 | Save Empty fleet to file | Fleet data | .txt file Saved | File saved successfully |
| 5 | Press yes on exit sub menu item dialog | User input | Program Exits | Program exits gracefully |
| 6 | Edit mail item function without any vans | User input, fleet data, van data | Error Message | Error Message popup stating: add a van |
| 7 | Edit mail item without any mail items | User input, fleet data, van data, Mail Items data | Error Message | Error Message popup stating: add a mail item to van |
| 8 | Edit mail item function with van & mail item data present and all forms filled in | User input, fleet data, van data, Mail Items data | Mail Item Successfully | Mail Item Successfully with popup confirmation message |
| 9 | Edit mail item function with van & mail item data present and forms left incomplete | User input, fleet data, van data, Mail Items data | Error Message | Popup message saying all forms need filled in. |
| 10 | Display Van Details with data from file | User input, fleet data, van data,  Mail Items data | Van Details displayed with Mail Items | Van Details displayed Successfully in textarea. |
| 11 | Display Van Details with no fleet data | User input, fleet data, van data,  Mail Items data | No details displayed | No details but blank fleet manager: displayed in textarea |
| 12 | Display Van Details sorted with data from file | User input, fleet data, van data,  Mail Items data | Van Details sorted by Delivery cost | Van Details sorted by Delivery cost in ascending order in textArea |
| 13 | Display Van Details Summary with data from file | User input, fleet data, van data,  Mail Items data | Van details summary displayed | Van details summary displayed along with number of items in text area |
| 14 | Add new van and filling all forms in | User input, fleet data, van data | Van added to fleet successfully | Van added to fleet successfully with confirmation popup message |
| 15 | Add new van and leaving forms empty | User input, fleet data, van data | Error message | Error message popup stating to fill out all forms |
| 16 | Add new mail item with no vans present | User input, fleet data, van data, Mail Items data | Error message | Error message popup stating to add a van to fleet first |
| 17 | Add new mail item with vans present in fleet, select parcel and fill in all fields | User input, fleet data, van data, Mail Items data | Parcel Successfully Added | Parcel Successfully Added with confirmation popup message |
| 18 | Add new mail item with vans present in fleet, select express parcel and fill in all fields | User input, fleet data, van data, Mail Items data | Express Parcel Successfully Added | Express Parcel Successfully Added with confirmation popup message |
| 19 | Add new mail item with vans present in fleet, select first Class letter and fill in all fields | User input, fleet data, van data, Mail Items data | First Class Successfully Added | First Class letter Successfully Added with confirmation popup message |
| 20 | Add new mail item with vans present in fleet, select second Class letter and fill in all fields | User input, fleet data, van data, Mail Items data | Second Class Successfully Added | First Class letter Successfully Added with confirmation popup message |
| 21 | Add new mail item with vans present in fleet but input a string as parcel weight | User input, fleet data, van data, Mail Items data | Error Message | Error message stating that’s an invalid value |
| 22 | Searching for a van id but without any fleet data loaded | User input, fleet data, van data | Error Message | Error message popup stating to add a least one van to fleet |
| 23 | Searching for a van id with fleet data load and within range | User input, fleet data, van data | Van details displayed | Van details displayed in textArea successfully |
| 24 | Searching for a van id with fleet data load and out of range | User input, fleet data, van data | Error Message | Error Message pop up stating van id is out of range |
| 25 | Searching for a van id with a character | User input, fleet data, van data | Error Message | Error Message pop up stating input is invalid |