|  |  |
| --- | --- |
| **ELECTRONIC ASSIGNMENT COVERSHEET** |  |

|  |  |
| --- | --- |
| Student Number | 33390144 |
| Surname | Kasseran |
| Given name | Manveer |
| Email | 33390144@student.murdoch.edu.au |
|  |  |
| Unit Code | ICT373 |
| Unit name | Software Architecture |
| Enrolment mode |  |
| Date | 15 May 2021 |
| Assignment number | 2 |
| Assignment name | Family Tree Application |
| Tutor | Mr AZM Ehtesham Chowdhury |

|  |
| --- |
| Student’s Declaration:   * Except where indicated, the work I am submitting in this assignment is my own work and has not been submitted for assessment in another unit. * This submission complies with Murdoch University's academic integrity commitments. I am aware that information about plagiarism and associated penalties can be found at http://www.murdoch.edu.au/teach/plagiarism/. If I have any doubts or queries about this, I am further aware that I can contact my Unit Coordinator prior to submitting the assignment. * I acknowledge that the assessor of this assignment may, for the purpose of assessing this assignment:   + reproduce this assignment and provide a copy to another academic staff member; and/or   + submit a copy of this assignment to a plagiarism-checking service. This web-based service may retain a copy of this work for the sole purpose of subsequent plagiarism checking, but has a legal agreement with the University that it will not share or reproduce it in any form. * I have retained a copy of this assignment. * I will retain a copy of the notification of receipt of this assignment. If you have not received a receipt within three days, please check with your Unit Coordinator. |
| I am aware that I am making this declaration by submitting this document electronically and by using my Murdoch ID and password it is deemed equivalent to executing this declaration with my written signature. | |
| Optional Comments to Tutor:  E.g. If this is a group assignment, list group members here | |

*If you can, please insert this completed form into the body of* ***each*** *assignment you submit. Follow the instructions in the Unit Information and Learning Guide about how to submit your file(s) and how to name them, so the Unit Coordinator knows whose work it is.*

Table of Contents

[**Purpose** 3](#_Toc71974919)

[**File Names:** 3](#_Toc71974920)

[**Requirements and Specifications** 3](#_Toc71974921)

[**Requirements** 3](#_Toc71974922)

[**Assumptions** 4](#_Toc71974923)

[**User Guide** 4](#_Toc71974924)

[**Compile and Run the program(using command line)** 4](#_Toc71974925)

[**How to use the program** 4](#_Toc71974926)

[**1.** **Start new family tree by adding the root person** 5](#_Toc71974927)

[**2.** **View details of the root person or subsequent relatives** 5](#_Toc71974928)

[**3.** **Edit details of a selected person** 5](#_Toc71974929)

[**4.** **Add new relatives to root or selected person on the tree** 5](#_Toc71974930)

[**5.** **Save the tree** 5](#_Toc71974931)

[**6.** **Load an existing tree** 5](#_Toc71974932)

[**7.** **Reset and start new family tree** 5](#_Toc71974933)

[**Structure/Design:** 5](#_Toc71974934)

[**Classes(Source listings):** 5](#_Toc71974935)

[**Design Description:** 6](#_Toc71974936)

[**Limitations** 9](#_Toc71974937)

[**Testing** 9](#_Toc71974938)

[**Strategy** 9](#_Toc71974939)

[**Test** 9](#_Toc71974940)

[**Discussion** 20](#_Toc71974941)

Family Tree Application

(ICT373 ASSIGNMENT2 S1 2021)

Manveer Kasseran

33390144

14/05/2021

# Purpose

This is a family tree application which will allow user to create , save, load, view and edit a family tree.

# File Names:

* src folder – contains all the source files for all the classes
* familytreesamples folder- contains sample files of different family tee:

1. Familytree.txt- contains the family tree of Johnny (a big family tree)
2. familytree2.txt – contains family tree of David (can be used to check that you can load a tree and edit/add relatives to the tree)
3. familytree3.txt-it is an empty file to test that the program notify user when trying to load an empty file

* Javadoc folder – contains the Javadoc documentation of all classes

# Requirements and Specifications

## **Requirements**

* Create, save, edit, load and view a family through a GUI built using JavaFX
* Family tree should start with a root person and his immediate relatives :parents, spouse , children and grandchildren.
* The GUI should provide both viewing and editing mode
* In viewing mode ,the GUI must show the root person with his parents ,spouse ,children and grandchildren and the GUI should display the details such as its personal details(first name, surname ,maiden name, gender , life description) and address details (street number , street name , suburb ,postcode) of a selected person from the family tree.
* In editing mode , the user should be able to start a new family tree or edit an existing one. User should be able to add new relatives or change the personal/address details of a person on the tree.

## **Assumptions**

* A person can have zero or one spouse , zero or a mother and a father , zero or more children and zero or more grandchildren.
* The family tree should start with a root person
* Should not display the parents of the root person’s parents or parents of spouse (should display only immediate relatives)
* Should not be able to remove relatives from the tree, can only add

# User Guide

## **Compile and Run the program(using command line)**

Note: make sure you have JDK 8 installed on your device.

1. Open cmd and go to the directory where you have saved the java program
2. To compile: type javac ‘javaprogramname’.java If there is no error in the program the cmd will take you to the next line.
3. To run : type java javaprogramname

## **How to use the program**

Once you got the program running ,you should see the GUI as shown below:

Graphical user interface, application, Word

Description automatically generated

Figure 1:Home of the application

**Now you can do the following actions:**

### **Start new family tree by adding the root person**

To start a new family tree you can click on ‘Add root person’ button .This will open a new separate window in which you can add personal information of the root person and the address details. Then click on the ‘Add’ button. This will add the root person to the tree.

Now you will be able to select an item on the tree to view their details , add new relatives and edit existing details which will be described in following sections.

### **View details of the root person or subsequent relatives**

To view the details of the root person or the added relatives ,you can simply click on the person. The information will appear on the right of the application.( do not click on the labels example .. parents: / spouse : )

### **Edit details of a selected person**

Initially you will not be able to edit the selected person details, you must click on the ‘Edit details’ button , this will enable you to edit the text fields .Then click on the ‘save’ button to save the new data. This will update the tree and the details of the selected person.

### **Add new relatives to root or selected person on the tree**

Once you have added a root person, you can add relatives to the root or add relative to the existing relatives. To do so ,click on a person on the tree, their information will be displayed on the right. Then scroll down you will find the add relatives button. This button will open a new window in which you can select the relationship type(parent, children or spouse) and add the relatives personal details and address. Then click on the ‘add’ button.

**Note:** only the immediate relatives of the root person will be displayed such as the parents, spouse, children and children of the children. But the parent of the parent or parents of spouse will not be shown on the tree.

### **Save the tree**

If you want to save a tree ,you need to click on the ‘save’ button. The file chooser will open and you can select the file to which you want to save the tree. This will save the root person object to the file selected.

### **Load an existing tree**

If you want to load an existing tree ,you need to click on the the ‘load’ button. The file chooser will open and you can select the file of the family you want to load. Initially the file chooser will be open at the location : “directory where you extracted the file”\FamilyTree\familytree

### **Reset and start new family tree**

If you have a tree already loaded and you want to start a new tree ,you can click on the ‘create new tree’ button which will act as a reset button and you can now add a new root person and start a new family tree.

# Structure/Design:

## **Classes(Source listings):**

For designing the family Tree Application I have created and used the following classes:

|  |  |
| --- | --- |
| Class Name | Description |
| FamilyMember.java | The familyMember class is used to represent a person /member in the family tree. It has the first name, last name , full name ,gender ,life description , address ,a static array to store a max of 2 parents , array list of children and a spouse. |
| Address.java | The Address class represent the address details encapsulating details such as the street number , street name , suburb and postcode |
| mainGUI.java | This mainGUI class has a method that will build the first scene of the application (homepage) and it has all the required methods to handles different types of events such as create, load ,save family tree etc.. |
| addRoot.java | This class is used to create a scene to add a root person which will allow user to enter personal details and address details of a root person |
| displayInfo.java | This class is used to create the layout for displaying information when user click on a particular person on the tree |
| addRelative.java | This class is used to create the GUI/scene which will allow user to add a new relative to a person when user click on the add relatives button |
| FamilyTreeApp.java | This is the main class where the static function is defined to start the application |

## **Design Description:**

I decided to use OOP to design and create the classes required to meet the requirements of the application. The FamilyMember will have the personal details (first name ,last name, maiden name, gender, life description) ,a static array of size 2 to store parents since you are allowed to have zero or a maximum of 2 parents, a Spouse of type FamilyMember , an Array List of children of type FamilyMember as a person can have zero or more children (this will allow dynamic growth of the array hence we can add as much children as we want).The FamilyMember class has an address of type Address .This is an aggregation (HAS-A) relationship as they can both exists without each other .This will provide reusability of the address class. These two classes will be used to store the details of a person added to the tree. The FamilyMember class and Address class implements serialization which will be important when saving and loading a family tree to and from a file.

The following classes were used to build and provide functionality of the GUI using JavaFX.

The addRoot class extends VBox class which is used to create the GUI for adding a root person to a new familyTree. This class encapsulates its components which prevent other classes from directly manipulating the variables .For example ,other class can access the add Button using the getAddButton() but cannot directly manipulate the button.

Graphical user interface, text, application, email

Description automatically generated

Figure 2 Add Root person GUI

The displayInfo class extends the VBox class which is used to build the layout for displaying information of a selected person on the tree. It consists of a sets of textfield and textarea in which the details( personal information ,address details and relatives) will be shown. It will also have the edit details button (this will be used to allow user to edit existing details) , add relative button(this will allow user to add a new relative) and a save button(which will appear when clicked on edit details button and use to save new edited details).The data of the components of this GUI can only be accessed through getter methods.

Graphical user interface, text, application, email

Description automatically generatedGraphical user interface, application

Description automatically generated

Figure 3 Display information of person

The save button appears only after you clicked on editdetails button.When you click on save the button ,it will disappear and save the new data entered or edited.

Graphical user interface, text, application, email

Description automatically generated

Figure 4 Save Button

The addRelative class inherits some GUI components and methods from the addRoot class .This will create the GUI to add a new relative to a person.I have used inheritance here since the addRelative GUI will be composed of almost the same set of components as in the addRoot GUI .This will avoid duplication of methods and attributes.You can choose the type of relative (parent,spouse or children) using the combo box.

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated

Figure 5 Add relative GUI

The mainGUI class extends the VBox class and it contains the method to build the scene of the application when first started. It also contains a set of methods to control the states of the GUI. It is encapsulates of a set of components such the buttons ,title , treeview of type family member which will be used to display the family tree. I have also used Border pane to handle the nodes of the mainGUI. It is easier to manipulate the nodes using Boderpane compared to other layout available.(see Javadoc to see documentation of the mainGUI class).Some methods also includes exception handling example in case of saving and loading a file.

Graphical user interface, application

Description automatically generated

Figure 6 MAIN GUI

## **FAMILY TREE APP UML**

Diagram, schematic

Description automatically generated

## **State Transition Diagrams**

Diagram

Description automatically generated

# Limitations

1. The program does not check for empty fields .for example: The user must at least enter the first name in the add root GUI but even though they do not enter the first name, the program will accept the empty field .
2. No regex check for the fields for validation of inputs
3. I forgot to display the grandchildren of an existing person. I can add children to children but the information is not displayed in the root person details.
4. Since I have not included icons for different gender ,it is difficult to identify who is the mother and father .To check who is the mother and father of the person, user will need to click on the person and the information will be displayed (for example ; the father first name will appear in the father textfield and same for mother or other relatives.)
5. The program does not check the file extension.
6. The user will need to manually create a txt file in the file chooser if they want to save the tree to a new file and then select the file, else they can overwrite an existing file.

# Testing

## **Strategy**

Testing is done after adding each feature required. If the feature does not function per requirement or does not give the output expected then review code and fixed the issue until the testing is passed .In case ,the error could not be fixed ,it will be mentioned in the limitations.

## **Test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Description | Expected Output | Output | Passed(Yes/No) |
| 1 | Check if the main GUI is displayed correctly with the correct layout and features | The main GUI with title, load, save, create new button and add root person button | Graphical user interface, application, Word  Description automatically generated | Yes |
| 2 | Check if the Add Root GUI open when user click on Add root person button | The Add Root GUI with all the correct  Text fields and add root button | Graphical user interface, text, application, email  Description automatically generated | Yes |
| 3 | Check if the root is created and added to the tree using the first name entered in the Add Root GUI | Root is created and added to treeview.(displayed using the first name) | Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, application  Description automatically generated | Yes |
| 4 | Check if the display information node appears on the right of the window when user select an item on the tree | The display info node appears on the right of the stage with the correct information about the selected person .  (edit buttons and add relative button show also be seen at the bottom) | Graphical user interface, text, application, email  Description automatically generatedGraphical user interface, application  Description automatically generated | Yes |
| 5 | Check if the save button appears after clicking on edit details button | Save button appears next to add relatives button | Graphical user interface, text, application  Description automatically generated | Yes |
| 6 | Check if user can edit details of the person selected and the edited data is saved and the tree is updated with new change.  (compare the changes to the data entered in test 4) | If user edit the first name of the selected person, the tree will display the new name entered after clicking the save button)  The save button will disappear after saving is done | Graphical user interface, application, email  Description automatically generatedGraphical user interface, application  Description automatically generated | Yes |
| 7 | Check if the add relative GUI is displayed correctly | The Add new relative GUI is displayed correctly with the correct fields and add button .There is a combo box which allow user to choose the relationship type(parent, child or spouse) | Graphical user interface, text, application, email  Description automatically generatedGraphical user interface, text, application, email  Description automatically generated | Yes |
| 8 | Check if user can add a parent to the tree | The parent is added correctly to the root person and to the correct label(Parents:)  If it the parent is a male it is the father of the root else it is the mother of the root person.(this will be show in the display information node of the root person).The root person will be the children of the parent added. | Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, text, application, website  Description automatically generated | Yes |
| 9 | Check if you can add a spouse to the root person of the tree | Spouse is added to the tree.(the spouse of the spouse will be the root person) | Graphical user interface, text, application, email  Description automatically generated  Graphical user interface, application  Description automatically generated  Graphical user interface, application  Description automatically generated | Yes |
| 10 | Check if you can add a child to the root person | Children is added to the family tree.(  The parents of the child will be the root person and spouse of the root person | Graphical user interface, text, application, email  Description automatically generated  Graphical user interface  Description automatically generated with medium confidence  Table  Description automatically generated | Yes |
| 11 | Check if you can add multiple children to the root | Children are added to the tree. | Graphical user interface  Description automatically generated |  |
| 12 | Check if you can add children to the children of the root | Children are added to the children of the root | Graphical user interface, table  Description automatically generated with medium confidenceGraphical user interface, application, email  Description automatically generated | Yes |
| 13 | Check if you can save the tree to a selected file from file chooser. | File is saved to the file selected. | A screenshot of a computer  Description automatically generated | Yes |
| 14 | Check if the saved tree in test 13 is loaded correctly with all the immediate relatives | The tree saved in test 13 is loaded with all the correct relatives . | A screenshot of a computer  Description automatically generated  Graphical user interface, application, email  Description automatically generated | Yes |
| 15 | Check if the window is reset back to the main GUI when the user click on Create New Tree .this acts like a reset button where user will need to click on the add root person to start new tree | The GUI is reset back to main GUI | Graphical user interface, application, email  Description automatically generated  After clicking on create new tree:  Graphical user interface, application, Word  Description automatically generated | Yes |
| 16 | Check if you can edit/add relatives to a tree loaded from a file | The tree is loaded with the root person (and if any other relatives) . | Whenloaded:  Graphical user interface, application  Description automatically generated  After adding relatives:  Graphical user interface, application  Description automatically generated  After editing person:  Graphical user interface, application  Description automatically generated | Yes |
| 17 | check if the program notify the user if they try to load from an empty file | Alert the user the file is empty  (use familytree/familytree3.txt file to test) | Graphical user interface, text, application, chat or text message  Description automatically generated | Yes |
| 18 | Check if the program notify the user if they are trying to add more parents and spouse than allowed | Alert the user when they trying to add more than 2 parents or more than 1 spouse | Graphical user interface, text, application, chat or text message  Description automatically generatedGraphical user interface, text, application, chat or text message  Description automatically generated | Yes |

Figure 7:Testing Table

## **Discussion**

These tests were carried out to ensure that the main functionality were satisfied. Based on the requirements of the program ,all the features works correctly excepts there are a few shortfalls (refer to Limitations section) .But more tests can be done to tests each class for example the setters and getters of the familyMember class and the Address class.