**Table Of Contents**

1- Problem Definition

2- How was the project built

2i- How to add records and how it was achieved

2ii- How to edit records and how it was achieved

2iii- How to search records and how it was achieved

2iv- How view records and how it was achieved

2v- How to delete records and how it was achieved

3- Task Sheet

4 Algorithms

**1 Problem Definition:**  
 We were to told build An Employee Recorder, this is a software built with java, and it takes details from employee's and store them in a text file, the software should be able to add records, edit records, search records, view records and delete records.

**2 How Was the Project Built:**  
 We built the employee recorder software with java and javaFx, to work on the programming and Graphical User Interface (GUI), We fulfilled all we were required to achieve which includes add records, edit records, search records, view records and delete records.

**(i) Add Records:**  
 The add records was built with GUI and the fxml file to load the GUI is stored in src, under resources' folder, under com.example.employeerecords folder a file called add-records.fxml, and the java file controller to run the functionality of adding records is stored in src, under main folder, under java folder, under com.example.employeerecords a file called AppRecords.java. With GUI using fxml we designed javaFX TextFields to receive inputs from users, we processed those inputs in our AddRecords.java class file using array lists and java objects.  
 To begin to add a record you will land on the home page with two buttons Add Record button and View Records button, the Add Record button is responsible for pushing us to the add-records.fxml file and calling its controller.  
 This button is created in hello-view.fxml, and its function addRecord() is created in the AppController.java.  
 We used EmployeeItems.java class as our object to store and get the employee details.

We used RandomNumber.java class as our object to store and get the employee number that is automatically generated by a random number generating algorithm in our AddRecords.java controller class.  
 Array Lists of objects were used to along these objects mentioned above to Iterate, set, add, remove and sort the information gotten from our TextFields holding the employee details you can see that happening in the AddRecords.java controller class.  
 When you click on the submit button in the add records page a file employee.txt is created and all the details gotten from the TextFields are written into the file using ObjectOutputStream and array list of Employee Items object, a random.txt file is also created to store the random generated number that will be the user employee number

**(ii) Edit Records:**  
 Records can be edited once you have added a record, you can see the button to edit a record once you have clicked the display records button, there you will see a list of all the records you have added and a green button that says edit, click it and a modal will show containing labels and TextFields and two buttons, one is the saveBtn responsible for saving the new updated changes, and secondly the closeBtn responsible for closing the modal.  
 The functionality for the edit button is called in a setOnAction event listener at AppController.java controller class, the event listener calls a function editBtnFunction() which accepts two parameters, the EmployeeItems object needed for processing records details and the parent element of all elements in my GUI which is the HBoxroot.  
 The function editBtnFunction() is created in ButtonFunctions.java class, and is responsible for running the algorithm that edits a record. uses the object EmployeeItems ek to process and iterate through the employee information.

**(iii) Search Records:**  
 Once the app has been loaded you will see a search bar located at the top of the page, the GUI is a TextField designed with a javaFX and is found in a fxml file in the resources folder called hello-view.fxml, the controller for handling the functionality search() is in the java folder at AppController.java controller class.  
 The search() function runs the algorithm responsible for search for a record, it reads some information from a record from the employee.txt file, and check if it matches with what was inputted in the search TextField.  
 The search option only searches people by their first name or last name or department or the combination of either two. Though if no record is found it will either display a blank white screen or do nothing at all.

**(iv) View Records:**  
 Records can be viewed when you click the View Records Button when the app has loaded although nothing will happen if you have no records to display, the GUI is designed with javaFX and is found in a fxml file in the resources folder called hello-view.fxml, the controller for handling the functionality which is view() is in the java folder at AppController.java controller class.  
 This view() function displays just some primary information and 3 other buttons, among those buttons is the view button which is white the button has an SetOnAction event listener that calls a function viewBtnFunction() which accepts EmployeeItems object as its parameter, the viewBtnFunction() is created in ButtonFunctions.java class and runs the algorithm which displays all the that record information which was gotten from employee.txt file.

**(v) Delete Records:**  
 When the app has loaded, and you have added a record, click the Display Records button, and you will see a list of your records, and a red delete button responsible for deleting that record.  
 This red delete button's GUI was designed in hello-view.fxml, and it has a setOnAction which calls a function deleteBtnFunction() that accepts two parameters EmployeeItems object and the HBox root which is the parent element of all the elements in our app.  
 The function deleteBtnFunction() is created in ButtonFunctions.java and its algorithm is similar to  
editBtnFunction(), only that we did not set what was in our array list, instead we removed it and wrote the array list into the employee.txt file.

**3 Task Sheet**

This project took us a month to build but most of that time was just us realizing and fixing mistakes, learning smarter ways to achieve a problem.

|  |  |  |
| --- | --- | --- |
| **Start Date** | **End Date** | **Task** |
| 16/07/2022 | 18/07/2022 | Working on the GUI |
| 19/07/2022 | 21/07/2022 | Working on adding records |
| 22/07/2022 | 23/07/2022 | Working on view records |
| 24/07/2022 | 26/07/2022 | Working on search records |
| 28/07/2022 | 29/07/2022 | Working on edit records |
| 30/07/2022 | 30/07/2022 | Working on delete records |
| 02/08/2022 | 03/08/2022 | Checking for errors and writing documentation |

**4 Algorithms**

1 Using javaFx and FXML to design the GUI

2 Using text files as databases to store and receive information using java file handling

3 Using array list of objects to iterate through every information, and performing CRUD operations using the array list functions like add() to add and information, set() to update and information, remove() to remove an information for the array list which contained everything from our text file database.

**Note:**

The main file where the main function is store is the HelloApplication.java