Employee Management System

Test Cases

Alicia Pan

Functional Units

1. Add an employee (pg. 2)
2. Search for an employee (pg. 4)
3. Edit an employee (pg. 5)
4. Remove an employee (pg. 6)
5. Save current state of employee data (pg. 7)
6. Load most recent state of employee data saved

This order was chosen because:

- Adding an employee is the starting point of the entire database
- Searching for an employee immediately detects whether or not adding the employee worked
- Editing an existing employee makes changes to an item already in the database
- Removing an employee can only be done if no further tests are being done on the removed employee
- Saving and loading from text files can be, in part, tested externally by looking at the text files themselves
- The functional units closer to the beginning of the testing regime generally have more possible scenarios as subsequent functionalities build on top of what was already given to the database

Add An Employee

Test ID	Description	Initial	Interaction	Criteria
1.1	Add employee to empty database, no text file pre-made	No employees in system yet, no pre-existing txtfile.txt	Launch program; add employee (with valid input)	Txtfile.txt is created; contains data of first employee added
1.2	Add employee to empty database, text file pre-made	No employees in system yet, pre-existing txtfile.txt	Launch program; add employee (with valid input)	Txtfile.txt contains data of first employee added
1.3	Add employee to already-populated database, no text file pre-made	At least 1 employee in the system already, no pre-existing txtfile.txt	Launch program; add employee (with valid input)	Txtfile.txt is created; contains data of all employee added
1.4	Add employee to already-populated database, text file pre-made	At least 1 employee in the system already, pre-existing txtfile.txt	Launch program; add employee (with valid input)	Txtfile.txt contains data of all employee added
1.5	Add employee to database with invalid values, text file not pre-made and database not already populated	No employees in system yet, no pre-existing txtfile.txt	Launch program; add employee (with invalid input)	Interface tells the user that the changes are invalid, gives parameters for valid values, and prompts the user to enter valid values; no invalid data

				enters the text file
1.6	Add employee to database with invalid values, text file pre-made and database already populated	At least 1 employee in the system already, pre-existing txtfile.txt	Launch program; add employee (with invalid input)	Interface tells the user that the changes are invalid, gives parameters for valid values, and prompts the user to enter valid values; no invalid data enters the text file

This order was chosen so that adding an employee could be tested progressing from the least set-up work to the most set-up work (from no employees and no text file to employees and a text file already there).

Search for An Employee

Test ID	Description	Initial	Interaction	Criteria
2.1	Search for a non-existent employee in an empty database	No employees in system yet	Launch program; search for an employee by employee number that is not in use	Interface displays that the employee that was searched is not in the system
2.2	Search for a non-existent employee in a populated database	Database contains employees, but not the one whose number is going to be used to search	Launch program; search for an employee by employee number that is not in use	Interface displays that the employee that was searched is not in the system
2.3	Search for an existent employee in a populated database	Database contains employees, including the one whose number is going to be used to search	Launch program; search for an employee by employee number that is in use	Interface displays that the employee that was searched is in the system and offers further action to remove or edit the employee's information

This order was chosen so that searching for an employee could be tested progressing from the least set-up work to the most set-up work (from no employees and a non-existent employee search to employees in the database and an existent employee search).

Edit An Employee

Test ID	Description	Initial	Interaction	Criteria
3.1	Open the edit window for an employee and exit with no changes made	Database contains the employee and allows user to edit	Launch program; search for an existing employee, open the edit window, exit without changing anything	None of the employee's information is altered and its record is still in the text file
3.2	Open the edit window for an employee and make invalid changes to save	Database contains the employee and allows user to edit	Launch program; search for an existing employee, open the edit window, exit without changing anything	Interface tells the user that the changes are invalid, gives parameters for valid values, and prompts the user to enter valid values
3.3	Open the edit window for an employee and make valid changes to save	Database contains the employee and allows user to edit	Launch program; search for an existing employee, open the edit window, exit without changing anything	Employee's information is changed and saved to the text file

This order was chosen so that searching for an employee could be tested progressing from the smallest difference made to the greatest difference made (no changes made to changes made and saved).

Remove An Employee

Test ID	Description	Initial	Interaction	Criteria
4.1	Open the view window for an employee and then do not confirm removal	Database contains the employee and allows user to remove	Launch program; search for an existing employee, select remove, select no when asked if user wants to confirm removal	None of the employee's information is altered or deleted and its record is still in the text file
4.2	Open the view window for an employee and then confirm removal	Database contains the employee and allows user to remove	Launch program; search for an existing employee, select remove, select yes when asked if user wants to confirm removal	Employee's information is deleted and no longer in the text file; new employees under the deleted employee's number can be validly added

This order was chosen so that searching for an employee could be tested progressing from the smallest difference made to the greatest difference made (no removal of employee to removal of employee).

Save Current State of Employee Data

Test ID	Description	Initial	Interaction	Criteria
5.1	Add a new employee	Text file exists	Launch program; add an employee, check text file in the EMS folder to see if data has been added	Employee's information is in the txtfile.txt file as a new line with attributes separated by "%", any other employees were unaffected and their data is still present, only one text file named "txtfile.txt" is in the folder
5.2	Edit an existing employee	Text file exists, employee to be edited exists	Launch program; search for an existing employee, select edit, select save	Employee's information with any changes made are in the txtfile.txt file as a new line with attributes separated by "%", any other employees were unaffected and their data is still present, only one text file named

				“txtfile.txt” is in the folder
5.3	Remove an existing employee	Text file exists, employee to be removed exists	Launch program; search for an existing employee, select remove, select yes when asked if user wants to confirm removal	Employee’s information is removed from txtfile.txt, any other employees were unaffected and their data is still present, only one text file named “txtfile.txt” is in the folder

This order was chosen so that searching for an employee could be tested progressing from the order of the test cases that relate to each of the save functionalities being tested (adding to editing to removing).

Load Most Recent State of Employee Data Saved

Test ID	Description	Initial	Interaction	Criteria
6.1	Make changes and check hash table through Master List window in the same session	Text file exists	Launch program; add/edit/remove employee and save; check if txtfile.txt reflects changes, select View Master List button to check if data is updated	All data in the text file and jTable are up to date; no duplicate entries are made
6.2	Make changes and check hash table through Master List window in a subsequent session	Text file exists	Launch program; add/edit/remove employee and save; check if txtfile.txt reflects changes, exit program and run it again, select View Master List button to check if data is updated from last session	All data in the text file and jTable are up to date; no duplicate entries are made

This order was chosen so that the testing within the same session can take place before the program is ended.