**Testing**

This sample of tests is primarily focused on user interaction rather than the underlying processes and algorithms that respond to this input, as it is the user giving unexpected input that is the most common source of errors and crashes in a program.

|  |  |  |  |
| --- | --- | --- | --- |
| What’s being tested | Input/steps taken | Expected output | Actual output |
| An employee can be added | Click ‘Add employee’ in the Edit menu’ | An employee will appear in the list with the default details |  |
| An employee can be removed | Select an employee and click ‘Remove employee’ | The employee will disappear from the list |  |
| If there are no employees, clicking ‘remove’ will not crash the program | Remove all employees, and then click ‘remove employee again’ | None | None |
| If an employee is not selected, clicking ‘remove’ won’t crash the program. | Deselect an employee and click ‘remove employee’ | None | None |
| An employee’s details can be edited | Select an employee and click ‘Edit’, add some details and then click ‘done’. | The Edit button will be hidden, four text boxes will appear over the details, and the ‘done button will appear. |  |
| If the user edits an employee when none are selected, the first one will be selected by default. | Deselect an employee and click ‘Edit’. | The edit will be applied to the first employee in the list | The changes applied to the first employee. |
| When a different employee is selected, the details and tasks update | Select an employee. | The details will update to the corresponding employee | Selecting a different employee causes the details to update |
| If the user attempts to edit an employee when none exist, they will be notified | Click ‘Edit’ without adding an employee first. | A message box will pop up notifying the user they must add an employee first. |  |
| A task can be assigned to an employee | Select an employee and click ‘Add task’ | A task with default details will appear in the task list. |  |
| If the user attempts to assign a task to an employee when none exist, they will be notified | Click ‘Add Task’ without adding any employees. | A message box will pop up notifying the user they must add an employee first. | The program allowed the task to be added. |
| **(This test is a result of unexpected behaviour in last one)**  Editing a task not assigned to an employee (i.e a task added when there are no employees) will result in a popup asking the user to add an employee. | Add and edit a task without adding any employees first. | A message box will pop up informing the user they need to add a task. | The message is generic, but the code does check if there are any employees added before allowing the user to edit a task. |
| If the user attempts to edit a task when none exist, they will be notified | Click ‘Edit’ without adding a task | A message box will pop up notifying the user they must add a task first. |  |
| Tasks can be edited | Select a task and click ‘Edit’. | The Edit button will be hidden, three text boxes and a dropdown menu will appear over the details, and the ‘done’ button will appear. |  |
| When a different task is selected the details will be updated. | Select a task. | The details will update to the corresponding task |  |
| Tasks can be removed. | Select a task and click ‘remove task’ | The task will disappear from the task list |  |
| When no tasks exist, clicking ‘remove task will not crash the program. | Click ‘remove task’ without adding one first. | none | None |
| When a user is removed, their file is deleted and the remaining files are correctly renamed so tasks stay assigned to the correct employee. | Remove one of multiple employees. | The tasks will correspond to the correct employees when another employee is selected. | These screenshots show that the first employee has one task and the second has two. Below, when the second one is removed, the new second employee (formerly the third) has three and does not inherit the deleted employees tasks. |