**Testing and Evidence Log**

|  |  |  |
| --- | --- | --- |
| Component 1 – Setting up Entry Frame | | |
| **DATE** | **WHAT I DID** | **EVIDENCE** |
| 20/5/21 | Test plan for this component:  All the aspects of the job should print into the shell when submit is pressed.  Other than that it should behave (on expected input only) as the final program, so the final test plan (specifically everything but the next and back display testing).  These screenshots show the testing plan, but with the boundary and exceptional tests crossed out because they are not relevant at this point. |  |
| 20/5/21 | For the very first version of my program (just the entry frame) I created all the widgets that I would need for this frame and gridded them where they will be placed. I have not added any variables or functionality to any of them, and there still is a lot of formatting to do, but this will be the base for my program. |  |
| 20/5/21 | Next, I got the customer name and distance travelled to print to the shell. These were grouped together because they are quite easy to get working. I added variables for them, linked the variables to the widgets, then set a command to the submit button that prints the value of these variables when pressed. |  |
| 20/5/21 | Next, I got the check buttons to also work and be printed when submit is pressed. When the submit button is pressed, if the variable for their respective variable is set to the on value, it will print out that service. When only virus was ticked it only printed that, vice versa for WOF, when neither were ticked it did not print anything, and when they were both ticked it printed both. |  |
| 20/5/21 | Finally, the last thing I did in this version was make it so that the minutes entry printed. I added a command to the virus checkbox so that you can only type in the entry box when that checkbox has been ticked. If the checkbox is unticked but there is still a value in the entry box, it will clear the entry box, so it does not print anyway. At the moment for it to work with the entry box the minutes variable is a string var. This will need to be changed when it is time to do calculations with the minutes value. |  |
| 20/5/21 | In this version I was saving the inputs into an object and calculating the charge. As I was testing that it was working I came across an error message. I had both WOF and virus ticked, but when I unticked WOF so only virus was selected this error popped up. I traced the error back to its line and realised it was because I had put the wrong variable (the virus one, when it should be the WOF one) in the else statement. I fixed it and the error went away.  Here is the code I changed and a screenshot showing a few of the tests I did. One of the charge numbers has a lot of decimal places but python does that sometimes and it won’t affect it when it is rounded to 2dp (because it is money). I added some constants to do with the charge amounts, a Job class, a method for calculating the cost, and changed the submit method so that it creates a job object and prints the variables directly from that object, so I know it has saved properly.  Next I need to get the job number to iterate. |  |
| 20/5/21 | I added code to the end of the submit method so that the job number would iterate. Shown are screenshots of some testing examples with the job number iterating, and also one of the GUI showing that it also updates on the label. |  |
| 20/5/21 | Finally, before testing, I made some minor formatting changes to the frame. I spaced everything out a little bit, so it is easier for the user to read and see which label corresponds to which widget. I also made it so that the minutes spent description label greys out whenever virus protection isn’t ticked so the user knows they can’t enter values into it. |  |
| 20/5/21 | I tested that everything in this component was working before I moved on to the next component. I tried a variety of distances, minutes, and combinations of WOF/virus, and nothing broke. This means that I can continue and start on the next component. | Evidence is in the testing videos folder called “testing component 1” . |
| 20/5/21 | With that, component 1 was finished! I was now running a couple days ahead of schedule. Here are screenshots of my trello board / calendar that show my progress so far. |  |
| Component 2 – Setting up Display Frame | | |
| 21/5/21 | Test plan for this component:  For this component I want a display frame to work on one hard-coded example job. | Test plan:   * Title should say 1/1 * Job number should be 1 * Customer name will be Olivia * Distance will be 20, virus yes, minutes 38, and WOF yes, so **charge will display as $147.90** * Logo image should appear at the bottom * Buttons don’t need to be working yet |
| 21/5/21 | I set up the GUI for the display frame and put in the hard-coded details. There were no problems with the actual creation of the GUI or the displaying of the details on the labels, however there was a problem with the image of the logo. The logo is much wider than I want it to be and is stretching out the window for the program. My next step is to figure out how to resize that logo. |  |
| 21/5/21 | First, I tried to change the width in the label directly. However, this ended up cutting off the image.  After some research online, the only easy way to resize an image inside of python was to import another plug-in. So, I instead opted to resizing the actual file to a smaller image. This also meant the file size was not as big. |  |
| 23/5/21 | Before I tested this component, I wanted to do some formatting to make the output more readable, and also just to make things nice and balanced and more like what I envisioned when I drew the layout diagram. I added padding and changed the “sticky” alignment of a few widgets. |  |
| 23/5/21 | For testing, there was no need to take a video because it is just one lot of output that does not change (yet), and the buttons do not do anything (yet). I went through the test plan and ticked off all the things I had said I wanted to display. Everything was as expected, meaning I can now move on to the next component!  The next thing to work on will be making the display frame show multiple jobs and scrolling through them with back and next. As seen on the calendar, I am still running ahead of schedule. |  |
| Component 3 – Display Frame to show Multiple Jobs | | |
| 24/5/21 | Test plan for this component:  I will add three more example jobs into the program and get the next and back buttons to cycle through them.  This test plan demonstrates how I want it to work – a next and back button that change what is displayed on the labels, and they disable when you cannot go any further. | * Press next button   + Display labels should change   + Title label should say 2/4   + Description labels, image, buttons, and general layout should stay consistent * Press back button   + Should go back to the first and be identical to how it was before * Press the next button until you reach the end of the list   + Next button should disable until you press the back button   + Each time the labels should update and show the unique job * Press the back button until you reach the start of the list   + Back button should disable until you press the next button   + Going back over a job you have scrolled through, nothing should have changed on an individual job. |
| 24/5/21 | First, I added the other hard coded jobs. Then, I set up the functionality for the next button. The next button itself now has the next method as a command. When it is pressed the position, counter is added to and the labels updates. However, I have not yet made it so that it disables when you reach the end of the list. When I pressed next after reaching 4/4, I got error messages, and the job display counter kept on going up without the number of jobs changing (e.g. 8/4). |  |
| 24/5/21 | Next, I added a method that would check the position variable and disable/enable the next or back button depending on where you are in the list. I also added back button functionality, which was the same as the next button, but the position subtracts. It worked, but I found a few problems with this. Firstly, the back button only disables when you press the next button and then go back. When you first start the program, you can just keep on going back. I need to make the button disable from the start. Also, there is a lot of repeated code (back method is almost identical to next method. I will fix this by adding the repeated code in the next and back button methods to the check position method, which is already called in both of them. |  |
| 24/5/21 | I fixed all the issues outlined in the previous log entry. I set the back button to be disabled by default and moved the repeated code to the check position method (and renamed it to include the updating labels part). Now the buttons were working as I wanted them to (as shown in the screenshots). |  |
| 24/5/21 | I went through and tested everything that was in the test plan on my program. Everything worked as I wanted it to.  With that, I was finished with this component. The next component is to combine the entry and display frames into one program (right now they are on separate python files). According to the calendar, I am still ahead of schedule. | Testing evidence is in the testing videos folder, called “testing component 3”. |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |