Software Development Life Cycle

1.Planning and 2. Requirements Analysis

These were both done by my senior developer who then gave me the set of requirements and specifications I had to follow while designing my solution. Because of this I didn’t need to do the planning and requirement analysis myself.

For context the planning stage is getting what the objective is from the employer and making the goal. Whereas the requirements analysis is about putting research into what requirements need to be met and which ones are optional. This is mainly to get the specifications and requirements that the solution must follow.

3.Solution Design

I created a couple of design ideas about how we could make the ticketing system. The first was to have all the functions run in the main class and have the ticket class only be for storing and moving information. The second being that most of the functions would be in the ticket class and that the main class was where the inputs and outputs would occur.

Then after getting it looked at and seeing that the client was ok with ether, I decided to go with the second one because there is less chance of user error and because it becomes more robust and a lot easier for people to understand.

Solution design is where you make multiple new system solutions and then you select a small handful to get checked by the client if approved then they make the System Architecture and the Software Architecture so that the detailed design stage is easier.

4.Detailed Design

For the detailed Design I first split it into 2 categories the ticket class and the main class. The ticket class was composed of the tickets information and all functions that store and change the information of the tickets. The main class was composed of all the inputs and outputs and display format.

I then branched them into systematic chunks for example for the ticket system some of the chunks where the storage variables, the functions to change those variables and the output functions. Whereas I split the main class into the chunk’s user input, error prevention, output data collection and display.

Once I had gotten this far, I started coding because I am the kind of person who is better at developing something when testing then from purely planning.

Detailed design is where you split your task into individual chunks and design in further detail for each of these chunks. This is also where you think about the design of the interface and the error prevention.

5.Construction

For the construction I started by following my framework and did the ticket class and only when I had done all the functions, I had planned did I move onto the main class. This is because I like to be able to test the results altogether but do a lot of tests at that time and then slowly perfect the program. So, once I got to the main class section, I did tests and got others to also test it to make sure it worked as intended. Then I refined the display to look more formal and visually appealing.

Construction is about building the code to be like the design and about testing it to make sure it is like that design.

6.System Testing

I did a lot of unit testing and did a good amount of user acceptance testing but didn’t make a test table because I didn’t have enough time. If I had more time, I would have made a test table to record all my tests.

Firstly, some of the unit testing I did was when I finished making the interface, I tested what happens when you close a ticket that was already closed multiple times and the statistic counters would go into the negatives. I also did this for the reopening of tickets then I coded it so that it checks that first.

Then for the integration testing I did the email checking tests to see what I could put for an email. At first you could even put h for the email, After a lot of trial error I finally got it so that all emails need @ and . to be allowed.

Then for the user acceptance testing I got over 4 other people to use my program and multiple errors were found and I did error prevention code for most of them. One of the errors that where found was that if you try to interact with a specific before any tickets where created you get stuck being asked for ticket ID.

System testing is about testing that all your components work, that the exchange between the sections work and lastly getting the users to test that it works. Then getting the errors and improving it from those errors.

7. Deployment, 8. Operation and 9. Maintenance

These are all done after the program is done and are outside of what I do for this project so I can’t do them for my life cycle.