**Software Development Lifecycle**

1. **Planning**

The goal is to create a ticket system for a help desk. The tickets should be handled from internal customers only, while tickets will also need help from the help desk as well as staff within the organisation.

All tickets must contain the staff ID, ticket creator name, email, and a description of the issue. (The first two characters of the staff ID, followed by the first three characters of the ticket creator name.) If a ticket’s description reads “password change”, a new password should be generated following the rule. There should also be an option, after the ticket has been submitted, to respond to the ticket by providing a feedback response.

1. **Requirement Gathering/Analysis**

**Statistic Requirements:**

There should be a way to keep track of the number of tickets submitted, resolved, and solved, along with the number of open tickets and a way to display them on the console.

If the staff member has submitted the “password change” request, after the new password is generated and the ticket’s response has been updated, **or** a member of the IT department responds to a ticket, the ticket should close, with the number of closed tickets increased and the number of open tickets decreased by 1 with the ticket’s status changed to “Closed.”

There should also be an option for a ticket to be reopened by the IT department. When this happens, increase the number of open tickets by 1 and reduce the number of closed tickets by 1, and the ticket status should be changed to “Reopened”. The ticket data should be displayed with the common ticket classes, along with a final response from IT department along with the ticket status (open, closed or reopened).

**Technical requirements:**

The ticket class should also have a method allowing the staff to submit a ticket and the IT team to respond to the tickets. (Specifically resolve, reopen and provide a response to the ticket)

The Ticket class should contain a method for resolving password change requests as well as calling the method that would generate the new password, and should set up a response for the ticket and change the ticket’s status to closed. The TicketStats method in Ticket class should contain information of the ticket statistics and should be able to return the statistic information with the main class containing the main method.

Create at least one instance of submitting tickets and include at least one ticket with the request for “Password change”. Once the tickets are created, display the ticket statistics.

Resolve some of the tickets, then display the ticket information and ticket statistics, and reopen some of the resolved tickets, then display the ticket information and ticket statistics.

1. **Design**

How I will design the system is by writing code for the data for each ticket first, then assign them to their variables and have separate sections for ticket actions. Then each ticket will be written out with their variables assigned in the same order as the questions that will be asked in the final product.

I also created the section for password changes, and below this, I then created a section that will count tickets that have been solved, resolved, or created, as well as a section below which prints out the ticket data. Below that, there will be three more sections for the main ticket actions that are viewing, editing and creating.

And finally, for the main menu, the options for each ticket action will be on a separate line for easier viewing. I also separated each main ticket action and section by commenting a title at the top of each section.

1. **Development**

I first assigned the tickets with their associated variables, and then created another section that assigns each class with self., for any ticket action like printing or editing before creating the main menu for the software. There are three options which allow you to either view, create or edit a ticket. I decided to work on a part I found a bit tricky first, which is the password changes, which with some research and little guidance, I finally understood some code examples used this way and applied them to the requirements of the project.

I then made another section that creates the ticket, which requires a few input functions, and then made a section to view the tickets which can select which ticket you want to print, and another for editing a ticket, like deleting or closing a ticket. And finally, I finished off the final section of the development stage by making the main menu.

1. **Testing**

There were a few critical errors that I first encountered that most of which were incorrect indents which I managed to fix. Along the way, there were a few other errors occurring as I progressed through the testing, like variable typos, but with a lot of work and dedication solving these problems, it started to work again, with no errors being found.

1. **Deployment**

The software can be deployed in two different ways. One of ways it can be deployed is by installing it onto a server for companies with IT support divisions so it can be used to prioritise and organise work for the rest of their staff. There is also another option for the software to be deployed as an app or website to use mainly by IT support businesses, especially mobile, who can keep track of tickets anywhere they are.

1. **Maintenance**

A later update for the software can be increasing the efficiency in work, like a better layout of the data shown or the use of a different interface. Another update can be adding admin privileges to provide security and more purposes for authorised users.