Quality Manual
Team Knuth

Introduction

This quality manual denotes the qualities which must be adhered to throughout the development of the project. Quality is an important aspect of any product, and it is therefore important that all team members follow the quality manual to ensure that the product is produced to a high standard.

Qualities

1. Backup Policies

Each team member is responsible for creating weekly backups of their own work. They should ensure that they create backups both locally on their own system and on the teams GitHub Repository.

2. Documentation

All code and independent work should be fully documented. This helps to ensure that any other team member who wishes to view your work will be able to understand it with ease.

3. File naming conventions

All files should be named clearly with the first letter of each word being capitalised. Version numbers should also be included where applicable in the following format:

[File Name] v[Version].[file type] – Example: 'Team Knuth Gantt v1.xlsx'

4. Java Documentation

All code should be fully documented. Each class should have a description of what the class does, the author and also the version number. Any alterations to other code should be commented with the editors initials alongside with what has been changed.

5. Document Styling

All documentation produced for the project should be written in the 'Calibri (Body)' Font, with a font size of 11. Each document should have a centralised title using the 'Microsoft Word Title Style' of size 28, which is found in the styles section. Sub headings should be of size 14 and will use the 'Microsoft Word Heading 1 Style'. All text will be coloured black unless otherwise required. If needed, use this document as a guide on how documents should look when styling your own.

6. Variable Names

All Java Variables should use full descriptive names so that it easy for developers to understand what each variable does. An example of this would be having a variable named 'customerFirstName' rather than 'custFName'.

7. Class File Organisation

All class files should be split up between presentational and functional, with all class names having a similar structure. Each mission of the assignment will nave its own relevant folder with GUI interface classes having the style '[FileName].fxml', while functional classes should have the style '[FileName]Controller.java'.

Signed

| Ryan Pickering | Matthew Randle | Matthew Pritchard | Ethan Roe |