Task 2 UML Designs and OOP Considerations

1 Introduction

Before starting on the project, various object-oriented programming principles and designs were considered. Without first considering the various techniques and principles, the direction of the project may have been unclear. WinForms was decided upon to be used for the user interface as it is an easy .NET graphical library with many tools to use to create windows applications.

2 Decision of features (MoSCoW)

To decide upon what features to implement within the program, the MoSCoW method was used. This method defines different features which could possibly go into a product and categorise them in four categories depending on if the features are needed or not. This meant that the team could decide on which direction to go with how to design the program.

3 MVP (Model View Presenter)

The very first thing which was discussed was how the project would be laid out. A common pattern to use with WinForms is the MVP pattern as it allows the decoupling of layers within the project; The model layer has no reference to either the view or the presenter layer. This pattern also allows the project to be split into two main work flows, allowing the two members of the team to work concurrently on the application. The use of interfaces is also key to the pattern as it means the forms are decoupled and unit testing is also made simpler.

4 Facade Pattern

The decision of the facade pattern was made due to the amount of times the various objects in the database are required to interact with each other. In doing this, the system is decoupled and a more simplified way of accessing the data within the database is given. Therefore, having one class to access or edit information from the four different types of objects within the database is crucial in order to increase workflow and make the program more manageable.

5 Singleton

As the database needs to hold an image as an array of bytes, a class was needed which converts image objects into the format which can be stored. Since the class which converts the image back and forth only has two methods, it was decided that the class would just be a singleton. If the class were to be bigger, another method would be used as apposed to the singleton method to avoid debugging issues later in the products lifespan. However due to the time scale of the development of the project, a singleton is used for this class.