Coding Standards

Purpose of Having Coding Standards

Having coding standards allows the code to become more readable and understandable by following one set of rules.

It also allows for easier maintenance of the code.

It can also speed up the programming process.

C++ Version

Using C++11 with QT version 5.12.8.

Header files

A header file should be used for every class that is made. This will use the standard .h files, apart from the class used to contain global variables which will use a .hpp file. Methods are defined in these header files instead of inside the .cpp files.

Scoping

Variables are scoped as locally as possible. Therefore, if a variable is only needed for an if statement then it should be created inside the statement. Global variables are only used when needed, such as for storing data that is to be used throughout the whole application.

Classes

Classes are used for each of the UI pages that are housed in our application. This is due to being standard practise for QT to use classes for this role.

Functions

Functions are kept as small as possible. If code can be reused, then they should have their own function. If having small functions are not possible, then they will be thoroughly commented to increase code readability.

Other C++ Features

Preincrement and predecrement should be used due to generally being more readable as this form is used in areas such as predefined for loops. Use of these may be unreadable depending on the context, therefore should be followed by a code comment.

Integer types should be used due to being explicit, meaning you are specifying the right size for the variable.

Type deduction should not be used, instead variables should be given an explicit type.

Variable initialisation should be used when the variable is declared, unless the variable has the possibility to be given a value from the user.

Naming Conventions

Using names for variables which relate to what that variable is being used for is very useful for improving readability. Classes will use the upper camel case naming convention, for example “MessagingPage”. Methods will use underscores to separate words for example “set_active_status”. Variable names should use underscores to separate words eg password_entered. We have decided not to use a different naming convention to the methods due to method calling requiring a parenthesis which will differentiate the two.

Comments

Comments should be embedded within the code to make sure that each part of the code can be easily understood by reading the comments. Compatible comments should also be used on methods and important variables when using Doxygen.

Formatting

There should be spaces in-between sections of code to allow for better readability.

Spaces after commas should also be used.

Functions should be clearly separated through spaces.

Indentations should be used when appropriate (eg if statements, for loops etc)

Exception Rules

These would be taken on a case by case basis.