William Dallas
Dooho Shin

COMP211P: Deliverable 1

COMPILING AND RUNNING THE CODE:

All the java source code files are stored in the */coursework/project* folder. To compile the program, navigate to this folder by typing into Terminal the commands:

```
cd project
javac Main.java
```

This produces class files in the same folder. To then run the program, type in to terminal the command:

```
java Main
```

JAVA STYLE GUIDE:

We chose the Google Java Style Guide for the project. The Integrated Development Environment we used for the project (IntelliJ IDEA) used this style guide by default, so it made sense to choose this for simplicity.

LIST OF CLASSES:

We structured the classes based on the objectives of the entire project, rather than solely this deliverable. Some classes therefore have significantly less code than others, as the first deliverable was more focussed on a specific areas of the program (e.g. user management) than others.

Main

The main class simply contains the *main()* method, which launches the rest of the program by calling a method in the *MenuAndAbout* class, which launches the menu.

MenuAndAbout

This class prints the menu text with the *menu()* method, and calls various methods depending on user input. It also contains, the *about()* method, which prints the game's instructions.

UserManagement

This class handles the registration of new users and login, using several methods to do this. The class also informs other classes which is the last user to have logged in, using the *getLastUserLoggedIn()* method. This class creates instances of the *User* class. It is the longest class in the program at the moment, due to the nature of the first deliverable's focus on registering/logging in users.

User

Currently, this class stores username and password variables for each user object. These variables can be changed, or obtained using *get/set* methods. This class is relatively limited at the moment, as we have not started developing other areas of the project which will require more functionality from the *User* class.

GameManagement

This class checks if any user has logged in, and if so, creates an instance of the *Game* class corresponding to the most recently logged in user. If no user has logged in, it returns to the menu, printing a prompt for the user to log in.

Game

This class, at the moment, just prints a placeholder for the game, which has yet to be developed. The class is not very useful for Deliverable 1, but was included for use in the future deliverables. The instance variable *gameNumber* performs no function currently, but is a potentially useful variable to have in the future, when developing the game.

MiscFunctions

This class is designed to store methods that perform miscellaneous functions in the app that are useful to multiple classes. There is only one method at the moment, the *clearScreen(String s)* method, which clears the screen then prints string *s*. This method is used by several of the other classes, so it makes sense to have a class for similar functions, which will likely be expanded over the course of the project.

TESTING THE APP:

To test the app, we made use of the IDE's debugging functionality, including using breakpoints to stop the program at troublesome sections to see the values stored in variables etc. in order to work out the logical errors.

We tried to test the app systematically, for example testing the outcome when a user registration is attempted after the *userObjects* array is full, or when the return button is pressed in the menu without entering a character, to test whether the app functioned as expected, and whether any exceptions occurred.

CONTRIBUTION MARKS:

We wish to distribute the contribution marks evenly: 10 each.