CS-230 Software Engineering
Assignment Two (A2)
Implementation of Software (2020)

Liam O'Reilly

Number of Credits: 45% of 15 credit module.
Deadlines:
Week 1's Minutes and Contribution Breakdown:
Monday 16th November at 11am
Week 2's Minutes and Contribution Breakdown:
Monday 23th November at 11am
Week 3's Minutes and Contribution Breakdown:
Monday 30th November at 11am
Week 4's Minutes and Contribution Breakdown:
Monday 7th December at 11am
Final Submission:
Monday 7th December at 11am.
Demo Take Place Between:
Monday 7th to Friday 11th December.
Learning Outcome(s): To gain experience in implementing larger software as part of a team.

1 Overview

In this assignment, each group will develop the application detailed in the separate CS-230 *Functional Specification* document. The system you develop as part of this assignment shall adhere to the detailed list of the functionality and requirement specifications provided in that document. Please read the *Functional Specification* document before continuing.

This assignment requires a substantial implementation in Java (using JavaFX for the GUI). *Please, start early.*

2 A2 Tasks

Your team is asked to implement the first version of the application. Each team member is required to contribute to at least one class in the implementation. The designer of the class (from A1) does not necessarily need to implement the class in A2.

2.1 Implementation

You are tasked with implementing Version 1.0 of the application. This initial version will be a partial implementation of the requirements detailed in CS-230 *Functional Specification* document and designed in A1.

All features described in the CS-230 *Functional Specification* must be implemented, with the exception of:

- Level Editor (aka Game Board Editor). This includes all content within Section 11 of the CS-230 *Functional Specification* document.

Do not implement the feature(s) listed in the exceptions above as they may form part of the assessment for CS-235.

All other features should be implemented (including Extra Features – see Section 12 of CS-230 *Functional Specification* document).

Your code must follow the coding conventions from Lecture 9 (Coding Conventions). You must also comment your methods appropriately using normal Java comments (this is in addition to Javadoc). If your code is written well (and is self documenting) then you may need only a few normal (i.e., non-Javadoc) Java comments.

2.2 Application Documentation

Your application should be fully documented with Javadoc as per the coding conventions in Lecture 9 (Coding Conventions). Please also see Lecture 11 (Javadoc).

2.3 Zoom Demo

You must prepare for and participate in a live Zoom session where you will be required to demonstrate **all** of the implemented features of your application. Markers will be present who will note which features are working (and those which are not) and also the quality of the these features.

You should prepare as a group a demonstration which must take a maximum of 10 minutes. Please show all features even if only partly implemented / working as you may be awarded partial marks. When preparing you should put together a running-order with clear explanations of what you're demonstrating. 10 minutes is not a lot of time and you do not want to waste it faffing around!

Note: You must clearly show the features are working as intended. For example, if you are attempting to show the ability to create a user profile you might do the following: navigate to the create user screen, fill it in, press a confirm button then display the newly created user profile. If you do not show the newly created user profile then you have not demonstrated that your program can actually create user profiles.

The demo will be used to mark the functionality of the application. If the demo does not demonstrate a feature working, then you will not get the marks for that feature.

All members of the group should attended the demo however not all members need to provide input into the demo i.e. if you want one person can do the entire demo.

The application which you demonstrate must be the code submitted to Canvas. To demonstrate code which is different to that submitted will count as Academic Misconduct!

The demos will take place between Monday 7th to Friday 11th December (see Canvas for your exact date and time). **Please do not book any travel before your demo date. If you miss the demo, you may get a heavy deduction on A2.**

2.4 Deliverables

There are multiple deliverables during this assignment:

- **Weekly Contribution Breakdowns and Minutes – 10% of A2**

  You must submit weekly Contribution Breakdowns (see CS-230 Assignment Overview document). These will include the Contribution Breakdown itself and the minutes of the weekly meeting where you create the Contribution Breakdown. These will help document the contribution of members and the progress over the duration of the assignment.

  The Contribution Breakdown is submitted according to the Assignment Overview document. The minutes that accompany them will be submitted to Canvas.

- **Final Submission – 90% of A2 (in total)**

  The final submission will take place digitally via Canvas. The deadlines can be found at the start of this document. The final submission to Canvas consists of:

  – Implementation – **75% of A2** – see Sections 2.1 and 2.3

  – Application Documentation – **10% of A2** – see Section 2.2

  – Contributions Report – **5% of A2** – see Section 2.5

All group members must review the final documents prior to submission. Each individual group member is responsible for understanding the entire contents of the submission. Submission indicates that all group members have read and approved the content unless a conversation that has been documented by your Academic Mentor indicates otherwise.

One member of each group, the **Secretary**, should lead the submission of the weekly Contribution Breakdowns and the final submission on behalf of the group. Points will be deducted for those submissions that do not follow the file naming conventions and required file formats.

2.4.1 Weekly Minutes

Each week you will write up the minutes of the meetings where you discuss and create the Contribution Breakdown for that week. The minutes will be submitted **each week via Canvas as a PDF**, along with the Contribution Breakdowns, see CS-230 *Assignment Overview* document). The deadlines can be found at the start of this document.

Your minutes files must be in PDF format and be named *"GroupGN-yyyy-mm-dd-minutes.pdf"* where *GN* is replaced by your group number and *yyyy-mm-dd* is replaced by the calendar date of the deadline for the minutes (with *yyyy* replaced by the year, *mm* with the month, and *dd* by the day), e.g., *Group03-2020-11-16-minutes.pdf.*

The minutes format should follow the standard as set out in lectures.

2.5 Contributions Report

A Contribution Report is to be included in your final submission to Canvas. This document contains a description of what and how each group member contributed (overall) to the project implementation. The file must be named *"GroupGN-MemberContributions.pdf"* where *GN* is replaced by your group number.

Each group member is obliged to contribute at least one class. The intention is that the classes each student chooses to design are also the classes they contribute to (in A2). However, students can change between assignments if necessary.

The Contribution Report, no more than 5 pages, describes who contributed to which classes and other contributions, e.g., the minutes etc.. The Contribution Report is also written collectively. In other words, each group member describes their respective contribution to the project.

The Contribution Report also informs the reader if any unexpected problems arose during the course of the assignment. For example, if a group member skipped too many lectures, and as a result did not have the background necessary to contribute, this should be stated in the Contribution Report. Likewise, if the group experienced success in some areas, both expected and unexpected, this should be included.

Finally, this document should summarise any significant changes you made to the design during the implementation phase. It is normal to change the design when implementing, but please write a few sentences to explain how you have changed it and why.

2.5.1 Preparing the Final Submission

The final submission will be a single zip file named *"GroupGN-A2.zip"* where *"GN"* is replaced by your group number. Marks will be deducted for those submissions that do not follow the file naming conventions and required file formats.

To create this zip file follow these instructions.

Create a folder named *"GroupGN-A2"* where *"GN"* is replaced by your group number. This folder will contain all files and folders that you submit (see below). Once this folder is ready, zip it up to create a file named *"GroupGN-A2.zip"* and submit this single file to Canvas.

The folder must be structured as follows:

```
GroupGN-A2/
      doc/
      source/
      contribution-report.pdf
```

where

- **doc** contains a full copy of the website produced by Javadoc (all files produced). See Section 2.2.
- **source** contains your application source code (the .java files and other resources). Please also include a README file (named *"README.txt"* with basic instructions on how to compile and run your code. See Section 2.1.
- **contribution-report.pdf** is your Contribution Report. See Section 2.5.

2.5.2 Overall Marks

The 4 components above (Weekly Minutes, Implementation Source Code, Application Documentation, and Contribution Report) will be used to produce a Group Mark. You will submit 4 Contribution Breakdowns over the duration of this assignment. The average of the weekly contribution scores will form each student's own Contribution Score. The final individual mark (for A2) for each student will be calculated by weighting the Group Mark with their Contribution Score.

Academic staff can overrule marks and change the process of peer assessment for a group in conjunction with the Year Head in atypical situations.

3 Issues with Contribution

Assignment One (A1) and Assignment Two (A2) assess the performance of the group when delivering a non-trivial piece of software. As mentioned in the course, real software is developed in teams and therefore group work is a necessary skill that needs to be developed. Individual contribution levels will be taken in to account via the Contribution Breakdown system. Under normal circumstances, the Contribution Breakdowns will be used to adjust marks and determine the level of contribution. However, we realise that abnormal circumstances during group work may occur.

In the event of any non-contribution, please refer to the CS-230 *Assignment Overview* document and also follow this procedure:

- As early as possible, discuss the situation with your Academic Mentor.
- Work with your Academic Mentor to try and get the non-contributor participating.
- Keep your Academic Mentor posted on the situation.

Likewise, if any particular group member disruptively dominates group meetings and the assignment outputs in a way that is damaging to the team, the Academic Mentor should be notified immediately.

3.1 Academic Integrity and Academic Misconduct

By submitting coursework, electronically and/or hardcopy, you state that you fully understand and are complying with the university's policy on Academic Integrity and Academic Misconduct. The policy can be found at `https://myuni.swansea.ac.uk/academic-life/academic-misconduct`.

In the event that you think you think you witness plagiarism you should get in touch with your Academic Mentor straight away and explain the issue to them.

4 Project Hints

- The implementation must adhere to some rules:
  1. Coding Conventions: Your team is required to follow the Coding Conventions presented in Lecture 9 (Coding Conventions).
  2. Code Documentation: Your team is required to use Javadoc to document your source code. See Lecture 9 (Coding Conventions) and Lecture 11 (Javadoc).

  Teams very often lose marks because they do not follow the coding conventions or they do not submit fully-completed Javadoc documentation.
- This coursework is very time consuming. Some teams usually lose marks because they do not manage to finish. Allow extra time for teamwork.
- **Start EARLY!?!** This cannot be over emphasised.
- Preparing for the demo will take some time. Consider dedicating a day for this.
- All group members are expected to contribute to both design and implementation. Group members who are weak at programming may get help from other members who are stronger at it. *However, all group members are responsible for understanding the full submission.*
- It is best to have a robust and stable application with fewer features than an unstable program with all the features.
- Test your application thoroughly. Your program should be able to handle random input without crashing. Strange inputs will be tested.
- Allow your group time to learn how to use Javadoc and git (should you decide to use it).
- Ask questions in your Academic Mentoring sessions.
- Do not email the module lecturer with questions, use the Canvas forum/discussion board where possible.
- You could use Discord (`discordapp.com`) or Slack (`slack.com`) for communication as a group. Many software development teams and companies use systems such as these a main form of communication.
- You should use Zoom to conduct all virtual group meetings.
- Do not leave submission until the last moment. Submit early to avoid technical problems with

uploading to Canvas.