

# CS-230 Software Engineering

## Assignment Two (A2)

### Partial Implementation of Software

Liam O'Reilly

**Number of Credits:** 60% of 15 credit module.

**Deadlines:**

Week 1's Minutes and Contribution Breakdown:  
Monday 19th November at 11am

Week 2's Minutes and Contribution Breakdown:  
Monday 26th November at 11am

Week 3's Minutes and Contribution Breakdown:  
Monday 3rd December at 11am

Week 4's Minutes and Contribution Breakdown:  
Monday 10th December at 11am

Final Submission:

Monday 10th December at 11am.

**Viva Dates:** Vivas will take place between Monday 10th and Friday 14th of December 2018(see Blackboard for your exact date and time).

**Learning Outcome(s):** To gain experience in implementing larger software as part of a team.

## 1 Overview

In this assignment, each group will build a partial application as outlined in the requirements specification document of the previous assignment (A1). A detailed list of the requirements specifications is given in the Functional Specification Section of A1.

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

You and your team are now in the partial implementation phase.

This assignment is broken down into 70% group work and 30% individual viva.

## 2 A2 Tasks

Your team is asked to implement the first version of Tawe-Lib. 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 Partial Implementation

You are tasked with implementing Tawe-Lib Version 1.0. This initial version will be a partial implementation of the requirements discussed and designed in A1.

All features described in A1 must be implemented, except for:

- Statistics. This includes all content within Section 2.13 of A1. You should not collect and store the statistical information either.

**Do not implement the features exempted above.**

All other features should be implemented (including your extensions – see Section 2.15 of A1). The displays should appear as specified as well. All other features described in the specification (A1) are to be completed as part of CS-235.

Tawe-Lib should operate on a single machine. For example, a user might log in to the program, perform some actions which are saved locally to disk on that machine, and then log out. When the next user logs in (or the same user or even if the program is restarted) then the data in the system will be the same as when the first user logged out.

The `main` method must be in a class named `"Main.java"`. This class should contain no other method.

Your code must follow the coding conventions from Lecture 13 (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 13 (Coding Conventions). Please also see Lecture 15 (Javadoc).

## 2.3 Video Demos

You must create a video (or videos) that demonstrate each and every feature of your application working. You should use screen capturing software to do this. The audio or subtitles should clearly explain what is happening in the video.

If using multiple videos, then the video files must be named according to the feature that is being demonstrated. The video file(s) are to be saved in MPEG or MP4 format. You may use as many video files as necessary to demonstrate the features of your application. Showing multiple feature in one video also works well. One video that captures all the features is ideal.

**The total length of video you may submit is 10 minutes. Video content beyond the 10 minute mark will not be watched.**

You may crop your video(s) to cut out uninteresting parts. If you chose to speed up your video(s) (something I would deem quite advanced) then make sure the audio is not sped up.

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

The videos will be used to mark the functionality of the product. If the videos do not demonstrate a feature working, then you will not get the marks for that feature.

## 2.4 Contributions Report

A Contribution Report is to be included in your final submission to Blackboard. This document contains a description of what and how each group member contributed (overall) to the project.

*Each group member is obliged to contribute at least one class.* The intention is that the classes each student chose to design (in A1) are also the classes they implement (in A2). However, students can change between assignments if necessary.

The Contributions Report, no more than 5 pages, describes who contributed to classes, implementations, and other contributions, e.g., the minutes etc. The Contributions 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 didn't 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 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.

## 2.5 Deliverables and Viva

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 a physical paper submission. The minutes that accompany them will be submitted to Blackboard.

- **Final Submission – 60% of A2 (in total)**

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

- **Partial Implementation (Source Code and Video Demo) – 45% of A2** – see Sections 2.1 and 2.3.
- **Application Documentation – 10% of A2** – see Section 2.2
- **Contributions Report (PDF) – 5% of A2** – see Section 2.4

- **Viva (Individual) – 30% of A2** – see Section 2.5.3

All group members must review the entire submission prior to submission. Submission indicates that all group members have read/watched 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.

### 2.5.1 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 Blackboard as a PDF**, along with the Contribution Breakdowns via a **paper submission**, 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 “*XXGroupGNMinutes-yyyy-mm-dd.pdf*” where *XX* is replaced by either “CS” or “SE” depending on if you are a Computer Science Group or a Software Engineering Group, *GN* is replaced by your group number, *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., *CSGroup3minutes-2018-11-19.pdf*.

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

### 2.5.2 Preparing the Final Submission

The final submission will be a single zip file named “*XXGroupGN-A2.zip*” where “*XX*” is replaced by either “CS” or “SE” depending on if you are a Computer Science Group or a Software Engineering Group and “*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 “*XXGroupGN-A2*” where “*XX*” is replaced by either “CS” or “SE” depending on if you are a Computer Science Group or a Software Engineering Group and “*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 “*XXGroupGN-A2.zip*” and submit this single file to Blackboard.

The folder must be structured as follows:

```
XXGroupGN-A2/
  demo/
  doc/
  source/
  contributions-report.pdf
```

where

- **demo** contains all your video demos (one video is ideal) that demonstrate each feature of your application. See Section 2.3.
- **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.

- **contributions-report.pdf** is your Contribution Report. See Section 2.4.

### 2.5.3 Viva (Individual)

The individual viva component is primary purpose is to evaluate the understanding of and contribution to the project by each group member; and also their knowledge of CS-230. The viva will take place as a group, but you will answer questions individually.

The vivas will take place between Monday 10th and Friday 14th of December 2018 (see Blackboard for your exact date and time). **Please do not book any travel before your viva date. If you miss your viva, at best you will get a heavy deduction and at worst a zero mark on A2.**

Each group member will be asked to individually define their contribution to both the **design** and **implementation** of the software product. Each group member will be asked some questions on their contributions, some on the design of the software, and some on coding in Java. Questions will also be asked to evaluate individual's knowledge of the product and of the content of CS-230.

More specifically, you (as an individual) will have no more than 5 minutes to clearly explain three things:

1. How you designed/implemented your technical part of the system.
2. How your part of the system fits into the larger software product. You must spend at least a minute talking about the code implemented by another group member and demonstrate to us that you understand it.
3. That you understand various topics covered in CS-230.

Students are responsible for the full course material during their viva. We can ask any question from any lecture of the term. You must demonstrate sufficient knowledge of this material in the viva.

### 2.5.4 Overall Marks

The Weekly Minutes and Final Submission (worth a total of 70% for A2) 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 and then adding on their individual viva score (worth 30% for A2).

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 and also at the end of term when the groups are interviewed and each member has the opportunity to demonstrate their understanding of the course material in reference to the project.

Under normal circumstances, the Contribution Breakdowns and the interview 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.

## 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 13 (Coding Conventions).
  2. Code Documentation: Your team required to use Javadoc to document your source code. See Lecture 13 (Coding Conventions) and Lecture 15 (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 don't manage to finish. Allow extra time for teamwork.
- Start early. This cannot be over emphasised.
- Producing the video(s) will take considerable time. Consider dedicating a day or two 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 better at it. *However, all group members are responsible for understanding the full submission. During the vivas, this will be tested.*
- 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 Blackboard forum where possible.

- You could use Discord ([discordapp.com](https://discordapp.com)) or Slack ([slack.com](https://slack.com)) for communication as a group. Many software development teams and companies use systems such as these a main form of communication.
- You could Skype for some of your group meetings?

## 5 A2: Last Assignment of CS-230

A2 is the last assignment of CS-230. Software Engineers will be taking the follow-up module CS-235 next term. CS-235 is also 100% coursework based and will have three assignments which will be based on Tawe-Lib.