|  |  |
| --- | --- |
| Photo by Merlene Goulet on unsplash.com  Development and iteration of an exclusive Java-based android application for the dissemnation of equity linked insights | Abstract  This document presents a thorough and evidenced detailing of the development process I have undergone to plan, build, iterate over and produce an exclusive and secure Java based android application that can be used for the dissemination of Equity-linked insights and information among trusted individuals.  Evan Neale  TM470 |

Contents

[Draft Project Report 3](#_Toc141784712)

[Preparation and planning 3](#_Toc141784713)

[Background 3](#_Toc141784714)

[Problem Description 3](#_Toc141784715)

[My Proposed Solution 4](#_Toc141784716)

[Alternatives 5](#_Toc141784717)

[Major Tasks and Subtasks 6](#_Toc141784718)

[Life cycle model and schedule 8](#_Toc141784719)

[Schedule 10](#_Toc141784720)

[Resources, skills and methods 13](#_Toc141784721)

[Future Plan 18](#_Toc141784722)

[Project Work completed 19](#_Toc141784723)

[Information sources – the literature 19](#_Toc141784724)

[Project work – Increment 1 20](#_Toc141784725)

[Project Work - Increment 2 26](#_Toc141784726)

[Project work - Increment 3 34](#_Toc141784727)

[Review 47](#_Toc141784728)

[Review of current stage of Project Work 47](#_Toc141784729)

[Review of Project Management 48](#_Toc141784730)

[Risks to Project Completion 49](#_Toc141784731)

[Review of Personal Development 50](#_Toc141784732)

[Legal, social, ethical and professional issues and equality, diversity and inclusion concerns 52](#_Toc141784733)

[References 53](#_Toc141784734)

[Appendices 56](#_Toc141784735)

[Appendix 1 – Schedule 56](#_Toc141784736)

[Appendix 1.1 – Tasks and Subtask of he project schedule(Old) 56](#_Toc141784737)

[Appendix 1.2 – Application Preparation and TMA01 58](#_Toc141784738)

[Appendix 1.3 - First increment and TMA02 59](#_Toc141784739)

[Appendix 1.4 – Second increment, Third increment and TMA03 60](#_Toc141784740)

[Appendix 1.5 - Final increment, Iteration and EMA 60](#_Toc141784741)

[Appendix 1.6 – Second increment and TMA03 (Updated) 61](#_Toc141784742)

[Appendix 1.7 – Final increment, Iteration and EMA (Updated) 61](#_Toc141784743)

[Appendix 2 – TMA01 Work completed evidence 62](#_Toc141784744)

[Appendix 2.1 – Old Android Studio file structure 62](#_Toc141784745)

[Appendix 2.2 – Old GitHub file structure 63](#_Toc141784746)

[Appendix 3 – Wireframes 64](#_Toc141784747)

[Appendix 4 – MoSCoW 72](#_Toc141784748)

[Appendix 5 – Java Code Increment 1 73](#_Toc141784749)

[Appendix 5.1 – LoginActivity 73](#_Toc141784750)

[Appendix 5.2 – RegistrationActivity 75](#_Toc141784751)

[Appendix 5.3 – UserProfile 78](#_Toc141784752)

[Appendix 5.4 – UserValidation 79](#_Toc141784753)

[Appendix 5.5 - MainActivity 80](#_Toc141784754)

[Appendix 6 – XML Layout Increment 1 81](#_Toc141784755)

[Appendix 6.1 – activity\_login 81](#_Toc141784756)

[Appendix 6.2 – activity\_registration 83](#_Toc141784757)

[Appendix 6.3 – activity\_main 85](#_Toc141784758)

[Appendix 7 – Java Code Increment 2 86](#_Toc141784759)

[Appendix 7.1 – MainThreadActivity 86](#_Toc141784760)

[Appendix 7.2 – Post 92](#_Toc141784761)

[Appendix 7.3 – PostCreationAcitivity 94](#_Toc141784762)

[Appendix 7.4 – PostDetailsActivity 97](#_Toc141784763)

[Appendix 8 – XML Layout Increment 2 100](#_Toc141784764)

[Appendix 8.1 – activity\_main\_thread 100](#_Toc141784765)

[Appendix 8.2 – activity\_post\_creation 101](#_Toc141784766)

[Appendix 8.3 – activity\_post\_details 104](#_Toc141784767)

[Appendix 9 – Java Code Increment 3 106](#_Toc141784768)

[Appendix 9.1 – AboutActivtiy 106](#_Toc141784769)

[Appendix 9.2 – Comment 110](#_Toc141784770)

[Appendix 9.3 – MainPostDetails 113](#_Toc141784771)

[Appendix 9.4 – ProfileActivity 113](#_Toc141784772)

[Appendix 9.5 – ProfileEditActivity 121](#_Toc141784773)

[Appendix 10 – XML Layout Increment 3 129](#_Toc141784774)

[Appendix 10.1 – activity\_about 129](#_Toc141784775)

[Appendix 10.2 – activity\_main\_post\_details 131](#_Toc141784776)

[Appendix 10.3 – item\_comment 132](#_Toc141784777)

[Appendix 10.4 – activity\_profile 133](#_Toc141784778)

[Appendix 10.5 – activity\_profile\_edit 136](#_Toc141784779)

[Appendix 10.6 – activity\_toolbar 140](#_Toc141784780)

[Appendix 10.7 – activity\_toolbar\_bare 141](#_Toc141784781)

[Appendix 11 – Java code final increment 142](#_Toc141784782)

[Appendix 12 – XML Layout final increment 142](#_Toc141784783)

[Appendix 13 - Communications Log 143](#_Toc141784784)

[Appendix 14 – Project Logs 162](#_Toc141784785)

**DETAILED TODO’S ARE IN THE SEPARATE DOCUEMNT**

# Draft Project Report

## Preparation and planning

### Background

Over the past decade I have developed an interest In trading stocks and shares in my free time. I view it as an enjoyable pass time, but it does require doing a lot of research. I have found that this hobby gives me a much more optimistic outlook on the world, as it provides me with a clearer picture of some of the technological advancements that may help lessen or overcome a number of the world’s key problems.

In recent years many of my friends have gained an interest in investing, attracted by greater accessibility to trading platforms, a parabolic stock market, easier access to trusted information and, also, by having a friend (myself) who can help guide them through the initial steps required. I have also encountered online acquaintances who I regularly converse with regarding stock-based topics.

Both I and my friends do the entirety of our research online where information resources are vast, spread across multiple different websites and apps.

### Problem Description

The key problem we find is that due to the extensive sources and constant stream of new information it can be impossible to filter through everything and it’s difficult for all interested friends to keep abreast of current events alone and especially in a timely manner, which can be very important when investing in newer more speculative companies. Combine this with the added complexity of trying to interpret poorly researched, purposefully misleading articles and having to decipher any hidden motives in some of the public discourse in online forums and it ends up being a very time-consuming enterprise.

Our current system is a group chat, via WhatsApp, which allows for people to post links to resources and any accompanying comments they feel is relevant. This format has an absence of any further functionality and creates a messily structured thread of information that can be troublesome to filter and find old posts. There is a search function but it’s too basic and lacks the ability to refine it any further.

Over multiple years myself and friends have used a long list of different apps and websites that act as useable forums for communication however none have suited us perfectly in their layout, their level of personalisation and their trustworthiness as they are usually home to scammers and dishonest individuals. I will note a few of these applications below in alternatives to my proposed solution and explain why they aren’t ideal.

### My Proposed Solution

At its most basic level my solution will be to create a simple, safe and exclusive java based android application that allows for the dissemination of news, links and rumours within a more trustworthy and customisable application. Initially, to tame its complexity, I would like the application to only be available to a select group of individuals. As I further develop the application, I would like to add more functionality and eventually open it to a greater number of people in my online social circle and beyond.

In more detail, the user will have to sign up and register their details to access the application, the user will have to create a basic user profile and they will be required to enter their login details on each visit. The structure of the application will consist of a central page containing the main thread of user made posts. Each post will have a strict and consistent layout, and will contain identifying tags, a title, a hyperlink, a discussion area for comments and, optionally, an image. The tags will specific to what company, sector and what kind of information is being posted (for example financial results or social media rumours). There will be a forum element attached to each post so that users can discuss each post in a more focused area. The main page will also have a search element which initially will be able to filter the posts in the main thread based on their identifying tags.

Simply this application will be a safer, simpler and free option which is better than all current alternatives. This will make the process of staying current on news a lot more efficient as more individuals are sharing the workload of sourcing and pooling information in an more user friendly and personalised environment.

I believe that developing this application perfectly aligns with the Software focus of my degree route. A second-year module I studied was ‘*Object-Orientated Java programming (M250)*’, and it was the enjoyment I experienced during this module that played a role in my decision to choose a project with a heavy Java element to it. This project builds on the Java and object orientated programming skills acquired during this time as I will be required to further research and understand progressively more complicated programming syntax, data structures and object orientated implementation.

In year three I enrolled in the module ‘*Software Engineering (TM354)’*, this module has played a key role in maturing the approach that I have used to develop the application, and it’s valuable to put the skillset and knowledge I’ve acquired into practice to further solidify this knowledge. Another module I’m studying in year three is ‘*Web, mobile and cloud technologies (TM352)*’ which focuses on application development, from concept to completion, and has helped to develop the development approach, even though it involves using different technologies.

I have also been developing my knowledge and skills outside of the Open University modules, as I have been experimenting with development within android studio, this project will further expand and solidify my foundational skills I’ve previously acquired.

### Alternatives

|  |  |
| --- | --- |
| **Alternative** | **Reasons why they are inappropriate** |
| Continue using WhatsApp | * Not suitable due to the large messages backlog * Search function is limited |
| Use a different messaging application | * Same issues to WhatsApp * Lacks the ability to create separate discussions around individual posts |
| Use stock twits (popular stock based social media app) | * Open and unfilterable userbase can be problematic and misleading * No functionality to create a closed system or group |
| Use a mainstream social media website to create groups | * Lots of non relevant functionality, for example Facebook has the capacity to create events which would be irrelevant * Not popular amongst potential users due to the nature of social media platforms business models |
| Bloomberg terminal | * Overly complex system, more than what is required * Extremely expensive |

### Major Tasks and Subtasks

As I have progressed through the project I have developed a greater understanding of the tasks and as such I have updated the list of tasks and subtask as required.

1. **Read TM470 Module website**
   1. Review study guide
   2. Review 'choosing a project' material
   3. Review forum posts
2. **Decide on project topic**
   1. Review potential topics and post on forum
   2. Reflect on feedback and finalise decision
3. **Work through TMA01**
   1. Review assessment details
   2. Create a skeleton of the document
   3. Work through TMA section 3.1 – preparation and planning
      1. Work through main sub-headings creating a flawed draft
      2. Review the sections that require attention and create a plan of attack
      3. Review content against learning outcomes
      4. Finalise draft
   4. Work through TMA section 3.2 – Project work completed
      1. Work through main sub-headings creating a flawed draft
      2. Review the sections that require attention and create a plan of attack
      3. Review previous module resources to help create diagrams and wireframe
      4. Review content against learning outcomes
      5. Finalise draft
   5. Create initial wireframe and Project files
   6. Work through TMA section 3.3 – Review and Reflection
      1. Work through main sub-headings creating a flawed draft
      2. Review the sections with the biggest issues
      3. Review any module resources needed
      4. Review content against learning outcomes
      5. Finalise draft
   7. Review, finalise and hand in TMA01
4. **First increment of the application**
   1. Build Login and registration page
      1. Review documentation and official guides
      2. Review final authentication choice
      3. Review final login and registration pages design
      4. Build frame of login and registration pages
      5. Embed authentication process into the pages
5. **Work through TMA02**
   1. Work through TMA section 3.1 – preparation and planning
      1. Review Specific Tutor Notes
      2. Review Assignment guidelines and Learning outcomes
      3. Implement Changes
   2. Work through TMA section 3.2 – Project work completed
      1. Review Specific Tutor Notes
      2. Review Assignment guidelines and Learning outcomes
      3. Implement Changes
   3. Work through TMA section 3.3 – Review and Reflection
      1. Review Specific Tutor Notes
      2. Review Assignment guidelines and Learning outcomes
      3. Implement Changes
   4. Review, finalise and hand in TMA02
6. **Second increment of Application**
   1. Build first iteration of main page and post creation page
      1. Review documentation and official guides
      2. Review final main page and post creation page design
      3. Build frame of main page and post creation page
7. **TMA03**
   1. Work through TMA section 3.1 - Draft Project report
   2. Work through TMA section 3.2 - Review
   3. Review, finalise and hand in TMA03
8. **Third increment of application**
   1. Create Profile section
      1. Review requirements and design of profile element
      2. Review Documentation
      3. Create profile and profile edit activities
   2. Iterate on the post creation section to include a comment section for each post
      1. Review Code
      2. Review Documentation on comments inclusion
      3. Develop comments section for posts
   3. Build first iteration of the search functionality
      1. Review documentation and official guides
      2. Add a simplified search element at top of main thread activity
      3. Embed search functionality into search element
9. **EMA**
   1. Work through EMA section 3.1 – General advice
   2. Work through EMA section 3.2 – Developing your report
   3. Review, finalise and hand in EMA
10. **Finalise the application**
    1. Review entire application codebase
       1. Gather and investigate all TODO tasks
       2. Identify elements that require further iteration
       3. Review MoSCoW Document
       4. Construct a plan and update project schedule
    2. Work through finalised tasks

### Life cycle model and schedule

|  |  |  |
| --- | --- | --- |
| **Lifecycle Model** | **Pro’s** | **Con’s** |
| Waterfall lifecycle | * Suits a Project where the numbers of stakeholders are limited (which holds true in this project) * Has an easily understandable sequentially structured approach which makes it easier to break down into sections, which would be appropriate for this project | * Structured in a way that there is no going back, this would not suit my intentions * Requires a lot of prior experience and knowledge, which I lack * Suits a Task that is routine, where there is familiarity with all aspects of the problem, this doesn’t hold true here |
| Iterative | * Can use alongside prototyping, which is a useful tactic when there is uncertainty about a product or a stakeholders desires * Can follow a adjusted waterfall model whereby there is regular movement backwards and forwards trough the stages | * Using a system that implements evolutionary prototyping creates multiple prototypes with limitations imbedded (for example in scope or design) which can bleed into the final model if not managed appropriately * Would require me to release small fully functioning elements of the application at regular intervals which may slow the |
| Incremental | * By developing simpler increments that are less complex is preferable and more manageable as this project is something relatively new to me * Allows the adjustment of requirements and overall deliverables as I develop experience and gain understanding in the project | * Generally used for larger more complex projects. My personal lack of experience means that this project is larger and more complex that anything I’ve done before |
| Agile | * Has a smaller planning section which means that a useful product will be released on a much quicker timeframe * Incredibly flexible to change due to the close cooperation of developer and user | * Hard to calculate the final costs, primarily time in this case but also financial in a business scenario, due to the flexible nature of the development. * Requires a highly skilled developer, which I am not. |

I believe the best lifecycle model choice for my project would be an **Incremental and Iterative** model. I’m relatively new to an undertaking such as this project, using a lifecycle model that is malleable and allows for regular change during development is one of the priorities for me. Following an incremental methodology would allow me to gradually build the application out, starting with some smaller and simpler functionalities while also reiterating over certain elements that I may have gained greater understanding in after my initial increment. By also implementing a iterative methodology I can make notes on the developments that I intend to implement in the future, this allows me to prioritise making the increments functional above all else.

### Schedule

**27/07/23 - Changes made (in project schedule and task list above):**

* **Extended TMA03 hand in date until the 10th July (a week late)**
* **EMA started om the 10th July (a week late)**
* **Added subtasks to 8.1 (iterate on post creation ect to include comments)**
* **Adjusted the search element task s into a more concise structure (just adjusted sub tasks really)**
* **Shortened the length of time for reviewing the application + TODO taks in final increment to 2 days from 1 week**
* Include how you encountered a realisation that you needed to implement a profile firsdt which took longer than expected, slowing the start of comments development
* Have noted down some more in notes app (essientially time took for profile, comments = longer and search = shorter)
* Main changes:
  + Took a week creating profile and strt comemnts (jul 10)
  + Finalised comments (jul17
  + **Reviewed/improved code, comments, cleared up issues ect(jul24)**
  + **Updated EMA(jul31**
  + **Then eesientially carridd on as normal)**

**Main changes**

As the project has progressed I’ve encountered no major set backs however there were considerations I took into account that led me to adjust the project schedule to a larger extent. Due to some life events I initially pushed beginning the second increment of the application onwards by a week. I also increased the number of weeks required to work on this increment, I believed that this would be needed I came to the realisation that it could be a little more complex and time consuming than I first anticipated, when combined with the fact that I will be revising heavily for exams in my other modules.

Upon reviewing and setting out a revision schedule I realised that my time would be better spent prioritizing the revision alone, I calculated there would be sufficient time post exams to fully focus on the this project and still hit the timeframe I had set in the project schedule, as such I adjusted the project schedule appropriately to reflect this. I added an extra activity whereby during the period that I was focusing on the revision I would periodically review the project and my conceptual understanding of the next increment so that it would be a smoother transition when returning to development. I moved the main development work of the increment to a starting date after my exam dates and as a result of this I also moved the start date of the third increment and the finalisation increment to a later date.

At the star of the third increment I realise I would nee to flesh out he Profile element first before moving forward so added those task which pushed some of the intial 3rd inccmrent tasks to a later date, thankfully I also decided that including an extended search leemnt wouldn’t be required and focused on developing an elemnt in the toolbar on the main thread activity alone which meant time was saved and the porigial pace of the scehdulke could be followed.

I also decided the start of final incemnt would take less days than originally planned and adjust the schedule to show that.

Final increment elaboaration here after getting tasks/plans together(and updating schedule to show)

I have also added more detail to the tasks and sub tasks in regards to the third increment of the application. I did this as my understanding of the application has increased throughout the project and now felt I had the ability to elaborate further and make better judgement on these elements.

I have provided a cropped screenshot of the entire task list from the Gantt chart, *Figure 1*, and then proceeded to elaborate on each following phase of the project beneath it. I have provided a series of cropped images of the entire Gantt chart, split into each phase, in [‘Appendix 1 – Schedule’](#_Appendix_1.1_–) which can be accessed by clicking on appendix links in the phase descriptions. For further clarity a more user friendly schedule is available as a PNG in the ‘project schedule’ folder within the ‘supporting documents’ folder of my TMA02 folder, the file itself is called ***TM470\_Project\_schedule(updated).png.***

**Tasks and Sub-Tasks**

This section covers the review of the course details and literature as well as the conception of the initial project idea. The changes I made to the task and sub tasks can be see clearly here, these changes were made during my 13th session of work, evidenced in **Log Sheet no.13 Appendix 8** , and then iterated over during my 14th session of work, evidenced in **Log Sheet no.14 Appendix 8**.

Update the figure below after finalisation of tasks at start of final increment

A picture containing text, screenshot, parallel, number

Description automatically generated

Figure 1 - Task and sub task of the project schedule as well as the first phase of work

**Application Preparation and TMA01**

This section covers the foundational activities, including the preparatory work creating a basic wireframe design of the application as well as the activities required to progress and complete the first assignment, TMA01. The illustration of this can be found in [Appendix 1.1](#_Appendix_1.1_–).

**First increment and TMA02**

This section covers the initial increment of the application which include the activities to build a functioning registration and login page as well as the activities required to progress and complete the second assignment, TMA02. In regards to the change I made regarding pushing back the start of the second increment can be seen illustrated in [Appendix 1.2](#_Appendix_1.2_-).

**Second increment, Third increment and TMA03**

This section covers the second and third increment of the application which will centre around building the functioning main page and post creation page, and all the elements needed to do facilitate that. Included will be the third increment will iterate over the main page post elements to include individual comment sections as well as a separate search page and also a search element at the top of the main page. Additionally included are the activities required to progress and complete the third Assignment, TMA03. It was in this phase of work where I extended the weeks required to finish the second increment and is evidence in [Appendix 1.3](#_Appendix_1.3_-).

**Final increment, Iteration and EMA**

This section covers the final increment which is the finalisation of the application which will involve iterating on previous increments and finalising the EMA. The illustration of this can be found in [Appendix 1.4](#_Appendix_1.4_-).

### Resources, skills and methods

I will primarily be using Java and XML to develop the application within the Android Studio Integrated Development Environment (IDE) which is provided by Android Inc. I chose these elements as I have previous experience with them, gained during previous modules as well as personal study, and they are also the most commonly used constituents in respect to android development and as such have a considerable amount of documentation, guides and a friendly online communities that will prove valuable during development.

I will be utilising the IDE’s emulator during development but have also purchased a cheap Google Pixel device to test the application in a more realistic scenario. Initially during development I will focus o the app’s functionality however on later iterations I will focus purely on the quality of the design and I intend to implement the material 3 design sensibilities within the application to make it appear more modern and fit within Google’s own open source design system.

I will be utilising a series of different note taking applications to help me organise the project. I will be using the Notion application for my project log as it’s available on multiple devices and I have grown familiar to it as it has been used throughput my whole academic career. I will be using GoodNotes application for creating and iterating over the application wireframes as well as reading and annotating PDF’s regarding android application development. I will use notability for scribbling quick and informal notes whenever I have a useful thought, which allows me to review it later on. I also find I regularly simple Todo lists at a more itemised l level which helps to keep me working through tasks bit by bit.

I will be using GitHub as a repository for the project which will allow for version control and also be a useful element in providing access to tutors and assessors if required. I have some experience with Git version control from previous modules and have used GitHub before in personal context. It is a free and popular version control option that has extensive documentation.

In regards to the authentication and database requirements of the application I have decided to implement Firebase authentication and the Firebase Firestore database. I chose these options as they were free use and were designed to integrate within the Android Studio IDE seamlessly, this is likely because they are provided by google which, as mentioned previously, also owns android.

During later iterations of the application I also intend to implement more graphics, as such I’ve decided I will use the ‘Unsplash’ website of free images as well as AI created images via Midjourney. Midjourney has the ability to create free use art based on user prompts in many different sizes and formats.

For creating my project schedule I will be using the smart sheets website to create, review and maintain the ever changing schedule.

As I produce the application I will be testing it via the inbuilt emulators in the IDE however since staring the project I have decided to purchase a cheap android device to act as a more realistic testing environment. I will be also be making use of a small number of users, who I will interrogate and include them and their android phones at specific points throughout production.

I will also need to gain better understanding of the formal requirements, methods and technologies required to accurately and fully test the application, a good starting point will be the official developer documentation and the official android studio guide combined with the literature I have gathered. However at this moment I’m aware of and intend to implement unit testing via JUnit which is supported by default in android studio.

A list of the relevant documentation and resources are as follows (complimented by full references in the reference section of this document):

* Android Application developer guides (Android, 2023a)
* Android studio guide (Android, 2023b)
* Material 3 documentation (Google, 2023)
* GitHub documentation (GitHub, 2023)
* Firebase developer documentation (Firebase, 2023a)
* Unsplash, for free images (Unsplash, 2023)
* Midjourney (Midjourney, 2023)
* Smartsheet’s Gantt charting (SmartSheet, 2023)
* JUnit 5 User Guide (Bechtold, S., Brannen, S., de Rancourt, J., Link, J., Merdes, M., Philipp, M & Stein, C. , 2023)
* Notion Application (Notion, 2023)
* GoodNotes Application (GoodNotes, 2023)
* Notability application (Ginger Labs, 2023)

**Risk Assessment Matrix**

By reviewing the risks and using a risk matrix, I can calculate a more specific risk profile and resultingly the overall risk.

**Total risk Key**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Very Low** | **Low** | **Medium** | **High** | **Very high** |
| 0 - 3 | 4 - 8 | 9 - 14 | 15 – 19 | 20 - 25 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RAG Risk Register |  | ***Impact*** | | | | |
|  |  | **Insignificant**  **(1)** | **Minor**  **(2)** | **Significant**  **(3)** | **Major**  **(4)** | **Severe**  **(5)** |
| ***Likelihood*** | **Almost certain**  **(5)** | Medium  (5) | High  (10) | High  (15) | Very High  (20) | Very High  (25) |
| **Likely**  **(4)** | Low  (4) | Medium  (8) | High  (12) | High  (16) | Very High  (20) |
| **Moderate**  **(3)** | Low  (3) | Low  (6) | Medium  (9) | High  (12) | High  (15) |
| **Unlikely**  **(2)** | Very Low  (2) | Low  (4) | Low  (6) | Medium  (8) | High  (10) |
| **Rare**  **(1)** | Very Low  (1) | Very Low  (2) | Low  (3) | Low  (4) | Medium  (5) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Ref | Risk | Likelihood | Impact | Overall Risk |
| Ref.1 | Potential users become uncommunicative | 2 | 1 | 2 |
| Ref.2 | Failure of computing equipment | 1 | 5 | 5 |
| Ref.3 | Miscalculation of planning schedule | 3 | 3 | 9 |
| Ref.4 | Misjudgement of my technical ability and my capacity to follow through in fully developing the project | 2 | 5 | 10 |
| Ref.5 | Scope creep affects the schedule I have created | 3 | 2 | 6 |
| Ref.6 | Other modules take up more time and affect project schedule | 4 | 4 | 16 |

**Mitigation**

I regards to Ref.1 All of the initial potential users have been selected and there are multiple so in the case that some users withdraw this shouldn’t be an issue.

In Ref.2 which focuses on computing equipment failing, I will make regular saves of all my Open university files to a local USB and I will also intend to push regular updates to my GitHub repository so that, in a worst case scenario, there wouldn’t be too much work lost. I would then quickly replace the computing device.

In regards to Ref.3 a miscalculation of the schedule and Ref.4 the misjudgement of my technical ability I intend to allot far more time and effort to the project moving forwards and I intend to expedite my familiarity with the elements of the project and the literature noted in my literature review.

To prevent scope creep, Ref.5, I will make sure to regularly temporarily adjust the project schedule and review feasibility of the increased scope and if it’s impractical to add then it will not be implemented.

In the case of Ref.6 where other modules vie for more time then I will simply allot more of my personal time to them. I won’t let them affect the overall project schedule however I may adjust smaller sections within the schedule. My other modules end a month before I must provide TMA03 so if the other modules do require more time this can be rectified by giving complete focus, and working days, to make up time in the final month.

### Future Plan

My future plan will take me to the end of the third increment and into the fourth, and final, increment where I will primarily be focused on testing and visual appearance of the application. Within this final increment I will also focus on reviewing and working through the lingering list of ‘TODO’ notes within my code.

Once I have completed the TMA03 I intend to review my conceptual understanding of what is required of the next increment and then generate the tasks which need to be completed to conclude the next increment, I will review the documentation to help me achieve this. I will then proceed to create, or modify where relevant, the java and XML files required to add a comments element to each post, build a search function into the application (based on each posts ‘tags’) and also build out the user profile element of the application.

During the third increment I will make sure to regularly test the application via the IDE emulator as well as my personal Google Pixel device, once I have made substantial improvements I will again utilise the user testing of my chosen users but this time in a more formal and structured fashion. Similar to previous increments I intend to focus on getting the core elements functioning so I will not spend a great amount of time labouring over the beautification or extended testing of the application as this will be priority in the fourth and final increment.

Similar to previous increments I intend to create additional *‘TODO’* notes within the code which I will return to in the final increment of the project. Once I reach the final increment I intend to fully review the multitude of ‘TODO’ tasks I have created I the project, I will break down each task into smaller sub tasks and review them against the prevailing schedule. If practical I will implement these tasks into the project schedule.

Once I have completed and fully tested the final increment I will once again engage with the selected users, whom I will to utilise to fully test and obtain feedback from. I intend to ask these users to test the application according to a rigorous test scenarios that I will define. I have been reading books in regards to sufficiently securing and testing android applications however I will need to further develop my knowledge to create formal and complete testing scenarios for the application as a whole.

## Project Work completed

### Information sources – the literature

(change to intext citations here)

During the course of this project there will be certain elements that I don’t have a wealth of experience in and as such I have taken it upon myself to find some literature that would help build upon and fortify my current knowledge. I am currently reading through these books and I believe they will all prove useful to the development of the project.

As such I uncovered **‘Android Studio IDE Quick Reference: a pocket guide to Android Studio Development’** by Ted Hagos that provides some very basic guides for quick reference to each step of the development process from setup of the IDE to preparing the application for release. Most likely this will provide a reference to point me in the right direction for further research or deeper reading of other literature.

I am currently reading ‘**Android design patterns and best practice ’** by Kyle Mew, I believe it will prove most useful as the experience I have within android studio is largely informal and may not conform to best practices. This resource will provide more in-depth directions on certain aspects of development that I have not encountered, and be especially important around the best practice in regards to these components.

I’m continuing to work through ‘**Android Programming for beginners – third edition’** by John Horton, this is a much more exhaustive and informative piece of literature and it covers the development of java applications within android studio in even greater depth, it is also relatively recently released which means that its content should be more up to date than other options.

Another literature source I have discovered and intend to use as both a reference and for more focussed reading is **‘Android Studio 3.0 Development essentials – Android 8 edition’** by Neil Smyth, it is comparable in depth to the book above by John Horton however it is a little older, therefore potentially less relevant, but it does have a more detailed contents list and acts as another beneficial point of reference.

A key literature source I have obtained and have been examining is **‘Testing and securing Android Studio Applications‘** By Belén Cruz Zapata , Antonio H Ninirola. This has been providing a great introduction to the testing methods and technologies that I must begin to implement going forward.

As the project has progressed I felt I needed to access more resources to help me through the more challenging and complex elements of the development.

I came to the realisation that I was essentially only using GitHub as a simple backup for my project when it’s original purpose, for a more professional situation, would involve much more complex scenarios such as multiple branch, merges and multiple users. I realised that I need to increase my knowledge base in regards to fully utilising git and GitHub which would also allow me to operate my repository in a more educated way. I located the literature sources ‘**ProGit’** by Scott Chacon and Ben Straub, which wonderfully was provided free on git’s website, to increase my understanding of Git and it’s functionalities, which GitHub is built around. I then located **‘GitHub Essentials’** by Achilleas Pipinellis to further understand and maximise my use of GitHub for this project.

I also realised that I would benefit from a more advanced and technical literature source so after reviewing a few different resources I discovered one which was aimed at a more professional user with more advanced considerations. I purchased **‘Professional Android’** by Reto Meier and Ian Lake which covered all aspects of the android development in a more advanced manner but also expanded on further aspects. This also included some useful information in regards to testing the application.

I also purchased **‘Android cookbook: Problems and solutions for Android Developers’** by Ian Darwin, this literature was helpful as a resource to check against whenever I encountered a problem, it was a helpful starting point for furthering my understanding of a problem and the gave options for possible solutions to the issue. This has been specifically useful for developing my understanding in regards to the different forms of testing of the application.

### Project work – Increment 1

The work completed in the first phase in the build up towards TMA01 was almost entirely preparatory, the wireframe from this phase of work can be found in [Appendix 3](#_Appendix_3_–) (figures 11 – 18 inclusive). Building on the work completed and referenced in TMA01, I have progressed the project in accordance with the project schedule. In TMA01 I provided screenshots of the files structure in both Android studio and GitHub ([Appendix 2.1](#_Appendix_2.1_–) and [Appendix 2.2](#_Appendix_2.2_–) respectively). In *figure 2* below, have provided the evidence of the updated Android studio file structure . I have chosen not provide any further evidence of the GitHub file structure as it effectively looks similar to the previous iteration.

Text

Description automatically generated

Figure 2 - Android Studio file structure Increment 1

After completion of TMA01 I solely focussed on building the first increment of the application. Within this increment was the login page and the registration page. At this time the application opens to the login activity where users can input their login details to access the main activity, alternatively they can navigate to the registration activity and input their details to register an account before continuing to login.

Graphical user interface, text, application

Description automatically generated

Figure 3 - Login activity

Graphical user interface, application

Description automatically generated

Figure 4 - Registration activity

To implement this I developed multiple java classes, which are as follows:

* LoginActivity
* RegistrationAcitivity
* UserProfile
* UserValidation
* MainActivity

I also further developed the XML files that provide the layout of the activity, they are as follows:

* activity\_login
* activity\_registration
* activity\_main

As the appearance of the GitHub files look largely identical to *figure 1*, Appendix 2.2, I will not provide evidence of that at this point, however I will provide the GitHub hyperlink in my final EMA document.

The various files I have developed I will elaborate on individually below, I have added hyperlinks to the titles which will take you straight to the relevant code snippets located in the appendix, I have removed boilerplate code where possible.

#### JAVA files

**[LoginActivity (Appendix 5.1)](#_Appendix_5.1_–)**

After the setting the overall layout view as the activity\_login XML file I created variables to access the views that take input from the users, whether that was from users inputting text in, the case of inputting an email address and password, or from users clicking on the login or register buttons. I also initialise a Firebase instance for usage in the authentication process.

I set on click listeners for both the login button and the register button. The login button calls the userAccountLogin() Method while the register creates a new intent and then starts that activity, which moves the user to the user registration view.

Once the login button is clicked the inputs the has user made in the edit text views are initialised to variables. The variables are then validated against validation criteria, which are elaborated on in the UserValidation class, they will produce a informative Toast message if there are any issues otherwise the method will continue.

I have used a modified version of the authentication method that I obtained from the Firebase documentation (Firebase, 2023b). It will interrogate the Firebase instance to see if the user credentials are present, if they are there will be a Toast message in the affirmative, it will get the users credentials and initialise them to a key value pair for use in the main activity, and then navigate to the MainActivity. If unsuccessful then the method will convert the returned exception into an informative error message Toast.

**[RegistrationActivity (Appendix 5.2)](#_Appendix_5.2_–)**

After the setting the overall layout view as the activity\_registration XML file I created variables to access the views that take input from the users, whether that was from users inputting text in, the case of inputting a username, email address, password and password confirmation, or from users clicking on the register button. I also initialise a Firebase instance for usage in the authentication process.

I set an on click listener for register button, this calls the registerNewUser() Method.

Once the register button is pressed the inputs the user made into the edit text views are initialised to variables. The variables are then validated against validation criteria, which are elaborated on in the UserValidation class, they will produce a informative Toast message if there are any issues otherwise the method will continue.

I have used a modified version of the registration method that I obtained from the Firebase documentation (Firebase, 2023c). It will then create a new user based on their inputted details, if successful there will be a Toast message in the affirmative, retrieve the UserID created in the Firebase instance and attempt to create a UserProfile object which will be stored I the Firebase Firestore User collection. If successful an affirmative toast message is displayed and the user is navigated back to the login activity to proceed to use their new UserProfile to log into the application. If unsuccessful then the method will convert the returned exception into an informative error message Toast.

**[UserProfile (Appendix 5.3)](#_Appendix_5.3_–)**

The UserProfile class was generated to produce UserProfile objects, which can be used to give individuals users their own unique experience of the application. The user profile class constructs a user profile object based on the supplied parameters of userID, username and userEmail. It contains auto generated getter and setter methods.

**[UserValidation (Appendix 5.4)](#_Appendix_5.4_–)**

I have created the UserValidation class as I wanted to take the depth of the UserValidation elements outside of the individuals classes that implement it. I know that I will expand on some of the methods that are needed across different activities within the application and preferred the idea of having them all in one place as I believe this was a reasonable implementation of OOP principles.

**[MainActivity (Appendix 5.5)](#_Appendix_5.5_-)**

This is almost empty at the moment, it will be key in the next increment of the application as it will be used to stage the main thread of the application. Currently all that is present in this class is setting the content view to the main\_activity XML file (which is also largely empty) and initialising the user details that were linked to the intent in the LoginActivity class, for usage in the implementation of a user profile element within the application.

#### XML Layout files

**[activity\_login (Appendix 6.1)](#_Appendix_6.1_–)**

This file contains the layout information for the login activities, in this iteration it’s a simple login screen with a simple welcome message and two input TextEdit boxes and two buttons, login button and a register button

**[activity\_registration (Appendix 6.2)](#_Appendix_6.2_–)**

This file contains the layout information for the registration activities, in this iteration it’s a simple registration page with a welcome message, four input text edit boxes and a register button.

**[activity\_main (Appendix 6.3)](#_Appendix_6.3_–)**

This file contains the layout information for the main activity, in this increment it’s a almost empty with only a little message that is useful to verify that we have made it to main activity.

#### Other files

Through out the application I have made specific changes and additions to an assorted collection of files.

I’ve updated the AdroidManifest.xml file so that when the application opens it opens to the login activity and not the main activity. I’ve also listed all of the different activities into this manifest.

I’ve updated the strings.xml with all of the text I have used up until now within the application, this is useful as all the text that is used can be accessed and changed via just this single file.

I’ve also added dependencies to the Gradle scripts which are required to make use of the different services related to google and the Firebase services.

#### Testing

As I’ve been working through the development in this increment I’ve been making use of emulator based testing and physical device based testing. When testing I have been making use of a logical, non exhaustive template whereby I have simply tested to check that they function as expected. This has involved me attempting to login with non registered details, registering as a user and reattempting to login with the newly registered details, while checking that the firebase authentication and Firestore database have collected the expected information.

Although this has proved adequate thus far this form of testing is ultimately nowhere near sufficient and I intend to implement a more structured, rigorous and formalised testing in later iterations. Implementing unit and system testing as required.

### Project Work - Increment 2

The second increment of the application has resulted in the file structure shown in *figure 5,* below, (I’ve installed an updated version of android studio which is why the UI looks a little different to *figure 2,* above)

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 5 - Android Studio file structure Increment 2

After completion of TMA02 I began focusing on building the second increment of the application. Within this increment was the main thread activity, the post creation activity and the post details activity. At this stage the application initially opens to the login activity where users can input their login details to access the main activity, or alternatively they can navigate to the registration activity and input their details to register an account before attempting to login. Once a user is logged in they are navigated to the main thread activity where they can see all user posts, from there they can either logout of the application, create a post or click on an individual post to see the post details.

A screenshot of a phone

Description automatically generated with medium confidence

Figure 6 - Main Thread activity

A screenshot of a black box

Description automatically generated with low confidence

Figure 7 - Post Creation activity

A picture containing text, electronics, screenshot, font

Description automatically generated

Figure 8 - Post Details activity

To implement this I developed more Java classes, which are as follows:

* MainThreadActivity *(which currently also holds the class:)*
  + PostViewHolder
* Post
* PostCreationAcitivity
* PostDetailsActivity

The MainActivity which was auto-generated at the start of the project, and is still visible in the file structure, will be removed from the application and I will utilise the MainThreadActivity in it’s place, however first I just need to make I have removed all references to it throughout the project.

I also further developed the XML files that provide the layout of the activities, they are as follows:

* activity\_main\_thread
* activity\_post\_creation
* activity\_post\_details

The various files I have developed I will elaborate on individually below. I have added hyperlinks to the individual titles which will take you straight to the relevant code snippets located in the appendix.

#### JAVA Files

[**MainThreadAcitivity (Appendix 7.1)**](#_Appendix_7.1_–)

This element serves as the main hub of the application, it is the activity that appears once a user has logged in, it provides the facility to view posts as well as navigate to both create posts, post details and logout activities. Further functionalities will be accessible from this activity in future increments.

There are two button variables, and onClickListners, which are the ‘logout’ button and the ‘createPost’ button. When the onClickListners observed a click then they create new intents, navigating the user either to the login activity or to the post creation activity respectively.

I utilised some code from the android documentation which allowed me to implement the Recycler View *(Android, 2023c)* this has essentially allowed me to display each post in a list format that the user can scroll through.

I created a variable which included a reference to the RecyclerView in the XML, I then made further modifications by setting the layout of the RecyclerView as linear, therefore scrollable. I then queried the posts collection, which was created and appended to in the post creation activity, and ordered them by their timestamp in a descending order.

I created an adapter, again with the help of the specific documentation *(Android, 2023c)*, which allowed the creation of overrides of the two methods, onCreateViewholder and onBindViewholder. OnCreateViewholder inflates the layout of each item in the RecyclerView and creates an instance of PostViewHolder, a custom view holder to hold this view. OnBindViewholder binds the data from the post objects to the corresponding PostViewHolder, the post objects being obtained from the firebase query results. There is also an onClickListner built into OnBindViewholder that will navigate to the post details activity when the post is clicked.

There is also the onStart and onStop methods, onStart starts the adapter listening for any changes in the database while onStop stops it from listening. OnStart is actioned when the user has the MainThreadActivity as their current activity and stops when they have navigated away from it. This is done to save resources by stopping the application from listening for changes when it isn’t required.

**PostViewHolder**

The PostViewHolder class is the custom view holder that defines the views that will be displayed I the RecyclerView ViewHolder and it is the BindPost method that binds the data of the post object to the corresponding views in the layout.

[**Post (Appendix 7.2)**](#_Appendix_7.2_–)

This was the class used to create the Post objects which the rest of the application would be built around interacting with. In this increment I created this class with the variables that would be relevant for his stage, there will be further modifications made in later increments. The class includes the fields for a postID which would allow the individual posts to be located and manipulated, a userID which would allow posts to be fond based on the user (important for the next increment), the title, tags and hyperlink as well as the timestamp so that the post could be organised and displayed chronologically.

I also auto populated the class with the relevant getters and setters, however not all have been required but I haven’t removed any that are excess to requirements yet until I finalise the post object and it’s functions in later increments.

[**PostCreationAcitivity (Appendix 7.3)**](#_Appendix_7.3_–)

I created variables to access the views that receive input from the users, whether from text entries, in the case of inputting a title, tags and hyperlink, or from users clicking on the create post button. I also created variables which would create instances of the FirebaseFirestore and FirebaseAuth, which allowed me access the Firestore database and Firestore authentication respectively.

I set an onClickListner to the createPost button, resulting in calling the create post method. This method essentially retrieves the user input in each of the text fields, converts them to a string and stores them in title, tagString and hyperlink variables respectively. These variables are initially validated to ensure they are not empty and subsequently the URL is checked to ensure it is a Valid URL, an informative toast is shown if they are insufficient in any of these regards. A tags array list is created and the tagString is iterated over split at each point where the comma’s occur and placed in an individual position in the array, after being trimmed for trailing space.

The code that followed is centred around adding data to the fire store, I modified it based on the firebase documentation *(Firebase, 2023d).* Essentially it checks for that there is a current user before creating a new post object based on the aforementioned user input and then saving it to the posts collection in the Firestore. There is an on success listener which will make a toast in the affirmative and navigates the user back to the main thread activity if it works as well as a on failure listener which will make a toast to the negative if the process failed.

[**PostDetailsActivity (Appendix 7.4)**](#_Appendix_7.4_–)

I created variables to access the views that will eventually deliver output to the users.

I then created a postID variable by accessing the POST\_ID that was given from the intent that started the post details activity. Utilising code I obtained and modified from the android documentation *(Android, 2023e)* I use the postID variable in a query to obtain the post document that has a matching post ID. If successful the post document is obtained from the database and converted to a post object, the post object is then accessed using it’s getter methods and used to update the relevant views, namely the title view, tags view and hyperlink view.

The hyperlink view also has a onClickListner set to it which, after validating it’s a secure URL, will start a browser intent using the URL, opening the page I a separate browser. In the event of a failure a toast is displayed to show that an invalid URL has been supplied.

I have also provided an on failure listener, in regards to accessing of the posts collection, in the event of a failure it will display a toast to inform the user of this scenario and also enter an error message into the Log.

#### XML Files

[**activity\_main\_thread (Appendix 8.1)**](#_Appendix_8.1_–)

This file holds the layout for the main thread, I have created two buttons to be utilised as a logout and a create post button, they are located in the top left and top right of the view respectively. I have also implement a RecyclerView, which takes up the majority of the view in the central two thirds.

[**activity\_post\_creation (Appendix 8.2)**](#_Appendix_8.2_–)

This file holds the layout for the creation of the posts, I essentially copied and modified the code from my registration page as it had a similar layout to what I required. The appearance has a concise direction in a text view at the top as well as three text inputs for the title, tags and hyperlink. There is also a create post button at the bottom beneath the other inputs.

[**activity\_post\_details (Appendix 8.3)**](#_Appendix_8.3_–)

This file holds the layout of the post details activity, essentially it is the layout of how the posts will appear both in the recycler view in the main thread activity as well as in the separate post details activity. It is comprised of three text views and a button. The text views will be populated by the username (not modified in this in this increment so currently just displays ‘username’ for each post), the title and the tags of the post, the URL will also populate the button which can be clicked by the user.

#### Other Files

Similar to the first increment, throughout the application I have made smaller changes and additions to an assorted collection of files.

I’ve updated the AdroidManifest.xml file so that each activity has a label, which is displayed when that activity is in view.

I’ve updated the strings file with all of the new text that has been used up until now within the application.

I’ve also added dependencies to the Gradle scripts which are required to make use of the different services related to google and the Firebase services.

#### Testing

So far for this increment I have continued to follow the same form of testing that I had previously been utilising which is to mainly focus simple practical testing, this is still not sufficient. I have also been increasing my knowledge by further reading from the security based literature source I found during my literature exploration (Zapata & Ninirola, 2014).

I implemented the usage of a small group of selected users as I wanted to simply test that the application works on other devices, that it allows for uses to both see the posts that have ben created and also create posts themselves that would be stored and displayed.

To send the application to the willing participants I built the APK which there was a facility for in android studio, I then located this APK and transferred it to google drive and the sent the link to the users to allow them to download and install it on their mobiles. I asked them to create accounts, using fabricated details so I didn’t have to worry about storing their data at this point. I then required them to sign in, view the posts that were visible and click on them. I also asked them to attempt to create a few posts. Everything worked as intended and they were able to do what was asked of them. They provided me with some informal feedback which centred around notes on the appearance of the application and the registration button on the login page initially being inaccessible to one of them.

### Project work - Increment 3

SECND DRAFT -- NEEDS WORK ON ALL FRONTS BUT FINE TO RETURN AFTER FINALISATION INCRMENT

The third increment of work has resulted I the file structure in figure 9 below (I’m again using a different version of android studio as I have migrated to a MacOS device).

After reviewing the schedule at the start of this increment, I came to the realisation that I would need to focus on expanding the profile element of the application before continuing with the original scheduled development of the comment sections and the search element.

In short within this increment I have increased navigability by including a custom toolbar, created a profile and profile edit activities, duplicated then modified the post details activity for better harmonisation with the mainthreadactivity recycler view and expanded on the post details activity to include a comments recycler view which included the support of user input.

A screenshot of android studio file structure.

Description automatically generated

Figure 9 - Android Studio file structure Increment 3

A screenshot of a phone

Description automatically generated

Figure 10 - Updated Main Thread Activity

A screenshot of a phone

Description automatically generated

Figure 11 - Main Thread activity utilising the search function

A screenshot of a phone

Description automatically generated

Figure 12 - Updated post details activity including comments

A black and white background

Description automatically generated

Figure 13 - Profile activity

A screenshot of a computer

Description automatically generated

Figure 14 - Profile activity user authentication, before navigation to profile edit activity

A screenshot of a black screen

Description automatically generated

Figure 15 - Profile edit activity

A screenshot of a phone

Description automatically generated

Figure 16 - About activity

>DECIDE HOW TO BEST SHOW THE CHNAGES IN THE TOP 3 BULLETS

>ONCE ABOVE HAS BEEN SORTED THEN HYPERLINK THEM ALL TO THE APPROPRIATE APPENDIX

I have made considerable adjustments to some of the files:

* Custom\_toolbar (in all java files)
* MainThreadActivty - Searchposts() method
* PostDetailsActivtity (comments)

To implement this I have I have developed more java classes:

* AboutActivity
* Comment
* MainPostDetails
* ProfileActivity
* ProfileEditActivity

I’ve also further developed the XML files:

* activity\_about
* activity\_main\_post\_details
* item\_comment
* activity\_profile
* activity\_profile\_edit
* custom\_toolbar
* custom\_toolbar\_bare

N.B. The fragment XML file ss are currently placeholders.

#### Java Files

**Custom Toolbar (currently duplicated in each activity)**

In most the java files I have incorporated the java to make the new custom toolbars functional. For now this code has been duplicated however I intend to abstract it in the final increment.

Every version has a logo and a createpost button which their respective both have onclicklistenrs. The logo navigates back to the main thread activity while the createpost button navigates the user to the postcreationactivity. The toolbars also have a drop down menu which allows the user to navigate to their profile page, the about page or log out. These are actioned by starting the intent for the respective activity .

On the toolbar for the mainthreadactivty there is also a searchview which allows the user to input text, as the text changes the search view takes the text as a query and trims and converts it to lowercase before entering it into the parameter of the method searchposts().

**MainThreadActivity**

The main thread activity has been modified to include the searchpost method, once a user has inputted text into the search view it is modified, trimmed, and converted to lowercase, before being entered into the searchposts method as a parameter.

SearchPosts method will effectively search through the ‘tags’ field of every post to find where there is a match. Should a post be found that has a similar tag then, using a slightly modified version of the original postrecyclerview code the smaller collection of post will be displayed in the recycelrview.

**PostDetialsActivty**

I have further developed this file to include the code for the comments recyclerview. The code for the comments recyclerview is essentially just the same as the posts recyclerview in the mainthreadactivity with some modifications relevant to accessing the specific comments for that post.

**AboutActivtiy (Appendix 9.1)**

This activity is empty and only has the boilerplate code to start the activity and structure it’s layout according to the about\_activty XML file.

**Comment (Appendix 9.2)**

This is the class utilised to create the comments objects. It has the following fields: username, commenttext, timestamp, postID and commentID. It has all getters and setters as wells an override toString method which returns a string of the details of the comment.

The username is to allow the dispay of who the user is who created the comment, the comment text will obviously be the main content which will be displayed, the timestamp is he time at which comment is created, and allows the calculation and display of how long ago the comment was posted for other users who read it. The post ID allows the ID of the post the comment is linked to be stored, this allows for the comments to be searched and displayed all comments on each post, the comment id allows the location and manipulation of that unique comment, which will prove useful if an editing functionality is to be created in the finalisation increment.

**MainPostDetails (Appendix 9.3)**

An issue that I had during development was that I couldn’t get the post details to resize appropriately for the main thread while also providing all the information in the post details activity. I then continued to flesh out this file (and its xml file) before realising that it was only the XML file that I required and as such this file is empty and noted for removal in the finalisation increment.

**ProfileActivity (Appendix 9.5)**

This activity serves as a display of the user details of the user that’s currently logged in. It displays the username and email of the current user and has a delete button, to delete the user account, and a edit profile button.

The user details are accessed by attaining the current users userid and using that to search the ‘users’ collection to access the document linked to that unique id. The document is then converted to a userprofile object and the username and useremail getter methods are utilised to access those fields before a settext method ensures that they are set to the specific views in the XML.

The delete button has an onclicklistner that when pressed will utilise the userID of the current user to find and delete that specific user docuent in the users collection, if successful then the authentication service is also interrogated for the current user which is also deleted. On completion a toast is made to the affirmative and user is navigated back to the login activity, or if failed a log and toast are created to further clarify.

The edit profile button calls the showPasswordDialog method when clicked. This method will create an alert dialog which asks the use to enter their password and saves the input. The dialog interface has a positive button, Confirm, and a negative button, Cancel. When cancel is pressed the dialog interface is simply cancelled. When confirm is pressed he input will be accessed and saved as a string before being entered as a parameter into a checkpasswordandredirect method. The checkpasswordandredirect method accesses the current user’s email before hen attempting to sign in with the accessed email and the inputted password as it’s parameters. If successful it will navigate the user to the profile edit activity.

**ProfileEditActivity (Appendix 9.6)**

This activity lets the user change their username, email and password. It has the respective input boxes for those as well as a save button for saving the changes. It accesses the current users username and email, utilising their current users userid to access the specific document in the users collection, and prefill those respective input boxes.

When the save button is clicked the saveprofileedit method is called, this will initialise the inputs of the input boxes to the respective fields, these fields are validated using some simple validation methods which produce informative toasts if there are issues.

If there are no issues then the username and email strings are added to a HashMap and pushed to the users document in the users collection, if this is successful then the current users email is updated in the authentication records and should that prove successful then the users password is also updated I the authentication records. Finally should all be successful then the user will be navigated back to the profile page which will then display these new details. If at any point there is a failure then there will be an informative negative toast explaining so.

#### XML Files

**activity\_about (Appendix 10.1)**

This file holds the layout for the about activity, and it includes an imagview of the logo, which I created with MidJourney, and a text view which has some details about the project.

**actitiy\_main\_post\_details (Appendix 10.2)**

This file holds the layout for the post details of the posts that will display in the postrecyclerview in the main thread activity. This was required as the initial post details XML file I created in an earlier increment was intended to be more exhaustive. These full information, and comments, each post weren’t required in the mainthread recycelrview and adjusting the layout for use in one activity was affecting the look negatively in the other activity. This layout provides a more condensed and useable post for the mainthread

Essentially displays three textviews: a username, title and tags, as well as the hyperlink in the form of a clickable button.

**item\_comment (Appendix 10.3)**

Essentially this is he layout of the comment that is inflated in the comments recycleviewer, so it is how every comment will look, in the postdetials activity. It includes textviews for the username, comment text and the time that has elaspsed since the comment was created, calculated in the java code using the timestamp of the comment.

**activity\_profile (Appendix 10.4)**

This is the layout file for the profile activity, essentially consists of an empty imageview (for later development into a profile image) and textviews which display the username and the user the users email. At the bottom left is the delete account button and the bottom right is the edit profile button.

**activity\_profile\_edit (Appendix 10.5)**

This is the layout for the profile edit activity, it also has an empty imageview placeholder for the future implementation of an image edit functionality. I repurposed and modified the XML from the postcreation activity, I added an extra text edit so there are four input text edit texts. One for the username, one for the email and then two more for the new password and to confirm the password.

There is also the save changes button at the bottom right.

**custome\_toolbar (Appendix 10.6)**

This is the layout for the toolbar that is included in some of the activities, it consists of a logoview on the left side, which is the logo I’ve create for the app, it has a searchview immediately to the right of the logo and a create post button to the right of that. On pages with the correct java implementation there will also be an options menu on the right hand side of the toolbar

**custome\_toolbar\_bare (Appendix 10.7)**

This toolbar is essentially the same as the toolbar described above but without the searchview element, this is because the searchview is only useful when on the mainthreadactivity.

#### Other Files

**tm470\_app\_logo.png**

This file was a logo that I managed to fashion after a bit of trial and error with the AI image generator MidJourney (MidJourney, 2023)

#### Testing

Probs redo this after you’ve actually done it as will be easier to explain

For this increment I have utilised a much more active schedule of personal testing. I have utilised my personal device via WIFI testing which pushes the updated app to my device when required. Using emulators I have also utilised a couple of different devices, both in size and OS version, to check how the app functions in different environments. I have also utilised my preorganised individuals as beta testers, and I gave them instructions to follow to test the functionality of the core elements of the app. Once they had carried out my instructions they were free to test and attempt to ‘break’ the app and then return their feedback to me.

I have also been developing the my understanding of the formal tests I will include and gathering further resources (must add these to resources section and also make sure to cite here?) to make this happen.

My current intentions are to utilise Junit for the unit tests, espresso for the UI tests and then to test the usability of the app both personally but also using espressos capabilities to do so.

# Review

## Review of current stage of Project Work

I believe the project is going very well so far. I find that the process and the rhythm of work I have adopted to meet the schedule has worked out well and everything I had intended to create up until this point has been completed. I’m extremely happy with how the application functions and happy with myself at how I have managed to get it to the stage where it is close to what I intended it to be from the start. I am a little unhappy with it’s level of polish, especially it’s appearance, and I have to keep reminding myself that it is supposed to be early increments and only a rough version of the final application and doesn’t need to be perfect.

I believe my exploration of the documentation and resources, required to build on my knowledge and guide me through development of the application, has proven invaluable. I also believe my continual iterative development of the activity wireframes has helped me obtain a constant clear picture of how the application will look and this has helped me to reflect and iterate over the application to create a better representation, albeit nowhere near perfect, of my intentions. I do believe this is an area that I could further improve on in future development scenarios.

I believe the less structured layout of this module and the less clearly defined requirements of the final project and assignment expectations slowed me down initially as it took me time to gather my thoughts and start consolidating my intentions for the project. This lead to me elaborating and adjusting the schedule regularly, which admittedly is something I still do even at this stage however I believe that I have improved at adapting to and creating outputs even when there are less structured objectives, hopefully this is evident in my TMA’s and the future increments of my project work.

Throughout the project I think my unwavering focus on a functional application has resulted in me deprioritising key elements such as the thorough and structured testing as well as the visual appearance of the application. Although these aren’t things that I’ve forgotten to do, as they are things I have scheduled for later increments, it has still left me with an overall weaker and less inspiring version of the application, which is a little disheartening.

Throughout the creation of the first TMA and development increment I became aware that I hadn’t done enough in the form of eliciting requirements and creating a conceptual model of the application. This is something I have taken on board and improved on which has been quite important as the increments have become gradually more complex and unfamiliar to me.

In future development scenarios I would spend a lot more time the formal structure of the classes and their relationships with each other as this has been the most challenging thing for me to overcome during this increment. I believe doing this would have lead me to create a more secure, tested and validated product in the first few increments of the application.

## Review of Project Management

I have been pretty happy In regards to the approach I have been taking to progress this project and feel it is sufficient to continue though to the completion of the project. I was studying three other modules alongside TM470 and managed to find a routine which suited me well. I used to spend 2-3 days focussing on TM470, depending on requirements and external circumstances, which was enough to allow me to keep pace with the schedule I had set out. Alongside the primary working days, which I had allotted to the core work of the project, I regularly skimmed parts of the android developer guides and books I have purchased and referenced in the literature sources.

For a period of time I had to adjust my approach and focus on other modules. During this phase I made constant checks on the project schedule and the project files to keep myself current which made sure my return to development would be smooth.

At this stage of development I have finished the other modules and I solely focus on the Project, this has allowed me to be more generous with the time I allot to it however I still must work it around my personal life and the further development of my skillset in other areas that are required for employment, which is of ever increasing priority.

I believe my chosen lifecycle model is supporting my project appropriately, the iterative and incremental model approach allows me to focus my attention on smaller elements of the overall application which stops me from getting too overwhelmed or losing focus from the component I’m working on and results in me producing a better quality and functional component.

The iterative aspect of it also helps me to focus on a the primary elements of the application and not allow issues, which aren’t as important in the initial increment, to distract from finishing the increment. By focusing on these smaller incremental components it is easier to comprehend the application as a whole conceptually and break it into a schedule. This approach also helps with regards to development past the intentions of this project as future increments can be built onto the application and future iterations can perfect the components already built.

I think that my schedule and the granulation of my tasks and subtasks within it has been appropriate. I find that as well as following the cadence and project schedule I also regularly create informal short term plans to help me gather my thoughts on what needs to be done at an even more granular detail as well as check that I going to be able to match the schedule I have set out, this allows me to adjust the time I allot to the project regularly and dynamically and prevents any issues in regards to keeping in pace with the overall project schedule.

As I have worked I have kept a daily record of what I have done and any issues that have arisen in my Project Log. I have included some examples in the appendices at **Appendix 8**. I find that the inclusion of these logs helps me to keep abreast of what I’m doing and intend to do the next working day. I also make use of different notes applications (Notion, GoodNotes and Notability) where I create my project logs, read PDF’s and scribble quick informal reminders. GoodNotes is also where I construct my wireframes and a good storage place for storage place for the PDF’s of books I read and annotate.

I regularly keep in contact with my Tutor via email, both in regards to bi-weekly updates and with any assignment related queries that come up. I’ve found he has been very helpful and incredibly rapid in replying to any and all messages I’ve sent. An example of the regular updates and queries can be found in an exhaustive list of communications in **Appendix 7.**

I have built out the TMA03 document by furthering development of the TMA02 document in relation to TMA03’s requirements. I have also made changes based on Tutor feedback received with the marked TMA02 document. Some of the more significant recommendations which I adhered to were the inclusion of a more exhaustive list of resources, skills and methods that will be utilised during the project, the adjustment of the RAG risk assessment matrix, the further development of the LSEP and ED issues and the inclusion of more literature resources.

## Risks to Project Completion

At this stage of the project I have an increased level of confidence that there will be no risk large enough that would hinder it’s final completion by the date specified in the project schedule (if not earlier potentially). The condensed risk table below identifies the initial risks that I foresaw.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Risk Ref** | **Ref.1** | **Ref.2** | **Ref.3** | **Ref.4** | **Ref.5** | **Ref.6** |
| **Risk** | Potential users become uncommunicative | Failure of computing equipment | Miscalculation of planning schedule | Misjudgement of my technical ability and my capacity to follow through in fully developing the project | Scope creep affects the schedule I have created | Other modules take up more time and affect project schedule |

The risk of uncommunicative users who I intended to utilise as potential testers was an unfounded concern as this issue hasn’t arose thus far, I have been in regular contact and updated them on at what stage of development the project was at and what stage I may need their participation. Although this could still be a risk I have assessed it’s likelihood as increasingly less likely.

The failure of my computing equipment is a constant risk which could be relevant until the end of the project. I have further mitigated this risk by making regular back ups to a USB device as well as regularly pushing all project files to my GitHub repository , therefore if a failure occurred I would have a relatively up to date version of the project available. I have also researched how to access files on the hard drive of a broken laptops, as most likely failure of equipment wouldn’t be related to the hard drive itself so I should still be able to access the files in that scenario. Realistically if any of my computing equipment breaks I have enough money and skills to replace it and would likely be back on track in the space of a few days.

Miscalculation of the schedule was a risk I felt could pose a potential issue at the start of the project however as I have worked through the project I have made constant reviews and changes, as is expected in a normal project. Through these reviews any miscalculations have been reviewed and amended. Another element to this could have been on the miscalculation of the timing of each increment and the final product, however this has been found to be unfounded.

One of the greater risks I anticipated was that the project would be too complex for me , which would cause compounding issues both in the development and the overall outcome. This issue has been largely mitigated by the fact that I have a keen interest topic and have regularly read the literature and documentation to expand my knowledge. Also, as is a common trait, I clearly misjudged my own capabilities and capacity to improve rapidly, as I have experienced no great issues from a technical standpoint.

Another risk I foresaw was scope creep, whereby as my knowledge and ability expanded, the application increments could also expand in scale, although this could affect the application for the better it would also impact the project schedule, and thus the overall outcome, negatively. I found that it was quite easy to focus the main functionality of he application, as I noticed further developments that could be made that would affect he scope I noted them down with the intent to review their possibility after the main functions are built.

One of the biggest risks to the project was that other modules would take up more time, this risk turned out to be a reality and the mitigation I had in place turned out to be unsustainable. My initial mitigation was to allot more and more time to the modules and project, but this had diminishing returns. I overcame this issue by reviewing and adjusting my schedule. For a set period of time I focused entirely on the other modules. Once they were completed I still had ample time, as I had been generous with the timescale of the project schedule, to fully complete the increment and assignment, almost, by the initial dates that were set out.

## Review of Personal Development

Overall I’m very happy with my personal development throughout this project. I’m especially pleased with how I have progressed my knowledge base in the Java, project management and application development.

As I have worked through the project the challenges I have encountered have helped to facilitate the development of technical skills, as well as soft skills that are relevant in all projects.

Through exposure to more complex Java problems that have in turn required complex and unique code solutions I have had to develop my understanding of the Java including the different packages, and their methods, to overcome these problems. This has also improved my problem solving skills which is a fundamental skill that all programmers and software developers must have.

My project management skills have improved through constant review and formal, as well as informal, itemisation and iteration of expected tasks leaves me constantly aware of what is required of me to complete the current increment of work. I also found that developing a greater level of detail in the project logs allowed me to be aware of what I had already done. Combining these elements I found important to keep me present in what I had done and needed to do in the project.

I also found I further developed my own self management as I have learnt more about myself and what works to get the best out of myself. I found the rhythm of work that worked best for me which was generally to have stints of work with regular, but not strict, breaks to break things up but when I was on the cusp of finalising certain elements I would allow myself to carry on until I felt the need to have a break.

**Explain how you have been developing the skills necessary for your project, if you do not already have them.**

In terms of managing a project and planning it’s undertaking I have some skills and knowledge acquired through earlier modules within my degree course, these skills have been further developed in more recent modules such as *‘TM354 software Engineering’*.

Some of the key skills that are required for this project will be java programming and the skills required to fully utilise the IDE. Previously I’ve covered Java within *Object-orientated Java programming M250* , I have also spent my personal time developing my familiarity with Java language using a multitude of online resources, however I haven’t used these skills in such a large scale project or to the level of complexity needed here but I am confident that I will manage to adapt and learn throughout the project. I have gathered a sizeable collection of official documentation and helpful user guides, both provided by android which will help me overcome any issues.

**Indicate to your tutor any issues on which you would like specific feedback.**

The areas of concern where I required feedback in the first TMA was around the scale and complexity of the project. I felt that the project was substantial and complex enough to be a good foundation for this module but I required a little more input. I was aware that there are other elements that I could’ve included in the application but I wasn’t sure on where the line was between having unreasonable aims and creating an application that is too small of a scale. I created a MoSCoW document which has further examples of elements I could add to the application to make it more complex, this is in **appendix 4**.

In TMA02 I asked for feedback in regards to how I had evidenced and explained the work I have done so far. Although it made sense to me I was concerned is it may not be as readable to any tutors who may read it and especially hard to read for tutors who may not be as familiar with Java or android development. In response my tutor mentioned about including more images which I have tried to do but hope to implement in an even greater sense n the final document.

For TMA03 I’m relatively happy that everything is coming together but I would like some feedback as to whether the document is too ‘wordy’? as the longer I have spent reviewing it I find myself thinking that it’s possibly too large a document?

**Briefly summarise how your tutor has influenced your thinking in terms of your project choice and how you have agreed to maintain contact with your tutor.**

When I first contacted my tutor I was still in the process of finalising the project and the scale of it. My initial communication with them made me realis that the scale of my project was initially too small and that it needed to be expanded if it was to be deemed appropriate. I had reviewed some of the elements of the project that I had in my MoSCoW table and moved some from the ‘Should have’ to the ‘must have’ section. It was also brought to my attention that there would be some important legal elements that would need to be considered as the application is stock trading based app that could have legal ramifications if trades were made based off recommendations or based off insider knowledge.

I have taken advantage of the tutorial that was held by my tutor and I have exchanged emails, he has also made me aware of the different avenues of communication open to myself and his availability for more direct and extended contact. For now email communication alone has been sufficient for what has been needed.

## Legal, social, ethical and professional issues and equality, diversity and inclusion concerns

The application will be a conduit of stock based information and this comes with certain legal obligations in regards to how users must be expected to act, how content is moderated and how users must be treated. I will have to develop a legal disclaimer that users of the application will be required to agree to during registration, it will state that the application developer and other users hold no responsibility for loss of money as well as inform the users that they must not post anything potentially illegal, offensive or non-related to the stock trading.

There will be the legal concern in regards to the sensitive data that the user shares when registering in the application, initially this will be the profile information that will include their email and a username (that can sometimes be identifying). This data will have to be managed in the appropriate way and will need to be done so in accordance with GDPR.

The application is primarily a space for user created posts and user discussion, as such without the correct barriers in place there could be offensive or exclusive posts which would harm the inclusive nature that should be imbued into it. To ensure an inclusive space there must be a terms of use that the user must sign up to follow when registering, failure to abide by these terms will result in their account and access to the application being removed.

Previously mentioned there will be a terms of use for users of the application however this alone will not be sufficient and there will have to multiple avenues for moderation imbued into the application to both review content against the terms of use but also to remove content. This must include a way for users to self moderate, by reporting content they deem harmful or not inclusive, in line with their interpretation of the terms of use.

The design of the application needs to be built, or modified in later increments, with accessible attributes imbued throughout. This would be done to make the application more usable for users who are visually or cerebrally impaired. This could be ensured through a clear and consistent structure to the application that makes it clear where one element ends and another begins. It could also include clear and distinctive text of an appropriate font size.

A key element of inclusivity would be to make sure the application is designed, or modified, in such a way that screen readers can reliably be used on it. This includes using descriptive labels for the elements that will be read by the screen reader and alternate text to be used for elements that won’t necessarily be able to be read, such as images or URL links.

# References

Android (2023a) *Developer Guide*. Available at: <https://developer.android.com/guide> (Accessed: January 2023)

Android (2023b) *Android Studio: User Guide* . Available at: <https://developer.android.com/studio/intro> . (Accessed: January 2023)

Android (2023c) *Create dynamic lists with RecyclerView*. Available at: <https://developer.android.com/develop/ui/views/layout/recyclerview#:~:text=RecyclerView%20is%20the%20ViewGroup%20that,by%20a%20view%20holder%20object>. (Accessed: June 2023)

Bechtold, S., Brannen, S., de Rancourt, J., Link, J., Merdes, M., Philipp, M & Stein, C. ( 2023) *JUnit 5 User Guide*. Available at: <https://junit.org/junit5/docs/current/user-guide/#overview> (Accessed: April 2023)

Chacon, S. & Straub, B. (2014) ProGit : 2nd Edition. [Ebook]. Available at: <https://git-scm.com/book/en/v2> (Accessed: June 2023)

Cruz Zapata, B. & Ninirola, A.H. (2014) *Testing and securing Android studio applications*. Packt publishing. [Paperback]

Firebase (2023a) *Developer Documentation for Firebase* . Available at: <https://firebase.google.com/docs?hl=en&authuser=0> (Accessed: March 2023)

Firebase (2023b) *Authenticate with Firebase using Password-Based Accounts on Android*. Available at: <https://firebase.google.com/docs/auth/android/password-auth> (Accessed: March 2023)

Firebase(2023c) *Get Started with Firebase Authentication on Android*. Available at: <https://firebase.google.com/docs/auth/android/start> (Accessed: March 2023)

Firebase (2023d) *Add data to a cloud Firestore*. Available at: <https://firebase.google.com/docs/firestore/manage-data/add-data#java> (Accessed: June 2023)

Firebase (2023e) *Get data with cloud Firestore*. Available at; <https://firebase.google.com/docs/firestore/query-data/get-data> (Accessed: June 2023)

Darwin, Ian F. (2017) *Android cookbook: 2nd Edition*. Oreilly Media, Inc. [Paperback]

GitHub (2023) *GitHub Documentation*. Available at: <https://docs.github.com/en> (Accessed: January 2023)

GoodNotes (2023) GoodNotes. Available at: <https://www.goodnotes.com/> (Accessed: 2023)

Google (2023) *MDC-Android documentation for Material Design 3*. Available at: <https://m3.material.io/develop/android/mdc-android> (Accessed: April 2023)

Goulet, Merlene (2015) *white ceramic mug on top of white ceramic saucer surrounded with coffee beans*. Available at: <https://unsplash.com/photos/ISFopTz7sBo> . Accessed(March 2023)

Hagos, T. (2019) *Android Studio IDE Quick Reference A Pocket Guide to Android Studio Development. 1st ed. 2019. Berkeley, CA: Apress.* Available at: <https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/android-studio-ide/9781484249536/?ar%2F%3Forpq=&email=zy600696&tstamp=1677498675&id=6099F6821F274FA15767A71D9802DD06CD2AFE19> (Accessed: February 2023)

Horton, J. (2021) *Android Programming for Beginners - Third Edition*. Packt Publishing. Available at: <https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/android-programming-for/9781800563438/?ar/?orpq&email=zy600696&tstamp=1675933946&id=4C1885D7C24F4C3123D9CA2A4885516489950457> (Accessed: February 2023)

Meier, R. & Lake, I. (2018) *Professional Android: Fourth Edition*. Wrox [Paperback]

Mew, K. (2016) *Android Design Patterns and Best Practice.* Packt Publishing. Available at: <https://learning-oreilly-com.libezproxy.open.ac.uk/library/view/android-design-patterns/9781786467218/> (Accessed: February 2023)

Midjourney (2023) *Midjourney* . Available at: <https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F> (Accessed: March 2023)

Notability (2023) *Notability*. Available at: <https://notability.com/> (Accessed: 2023)

Notion (2023) *Notion.* Available at: <https://www.notion.so> (Accessed: 2023)

Oracle (2023) *JavaTM Platform, Standard edition 8 API Specification*. Available at: <https://docs.oracle.com/javase/8/docs/api/index.html> . (Accessed: January 2023)

Pipinellis, A. (2018) *GitHub Essentials : Unleash the Power of Collaborative Development Workflows Using GitHub, 2nd Edition*. Pakt Publishing [Ebook]. Available at: <https://ebookcentral.proquest.com/lib/open/detail.action?docID=5446049&pq-origsite=primo> (Accessed: June 2023)

SmartSheet (2023) *TM470\_GanttPRO* . Available at: <https://app.smartsheet.com/sheets/wXxFRC5h4J935mf7hMxxrGG4Rcx79G7rJxJ2CCv1?view=gantt> (Accessed: March 2023)

Smyth, N. (2017) *Android Studio 3.0 Development Essentials - Android 8 Edition.* CreateSpace Independent Publishing Platform. [Paperback]

Unsplash (2023) *Beautiful free images and pictures*. Available at: <https://unsplash.com/explore> (Accessed: March 2023)

# Appendices

## Appendix 1 – Schedule

### Appendix 1.1 – Tasks and Subtask of he project schedule(Old)



### Appendix 1.2 – Application Preparation and TMA01

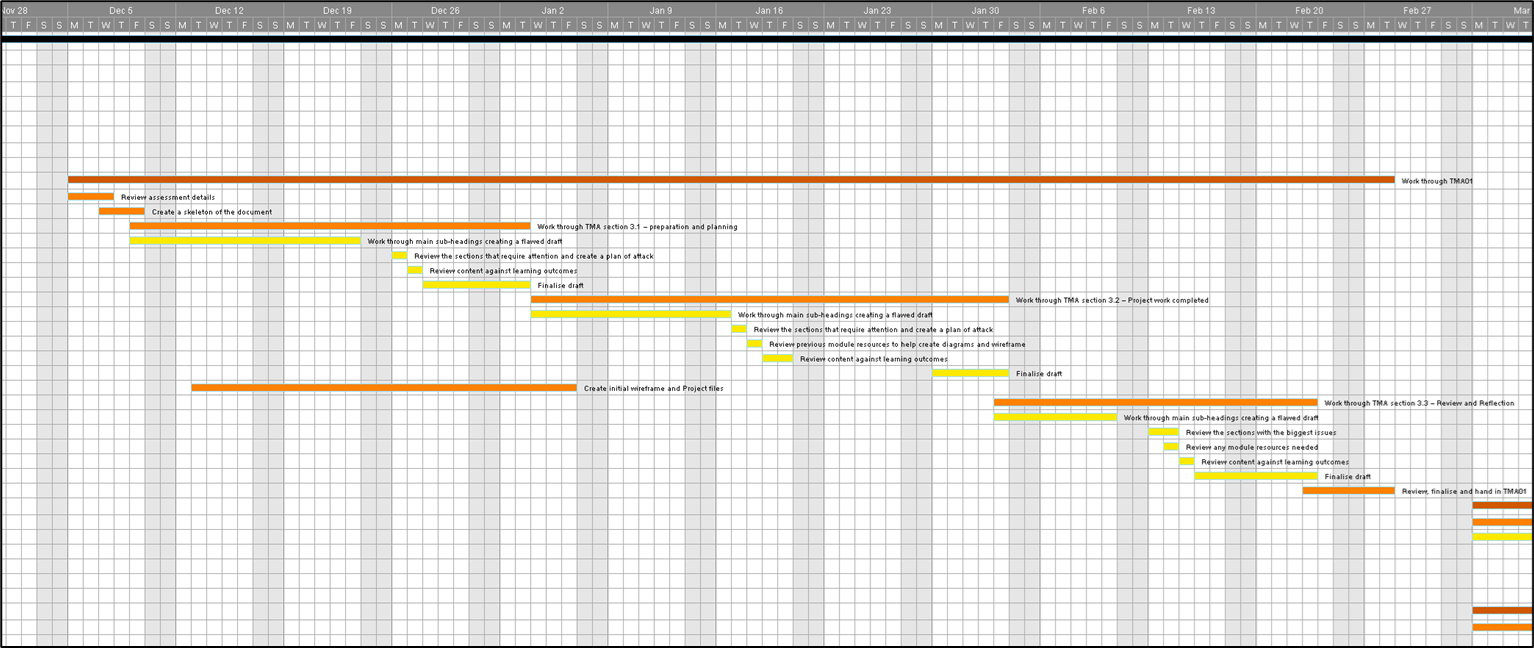


Figure 17 - second stage of Gantt chart

### Appendix 1.3 - First increment and TMA02

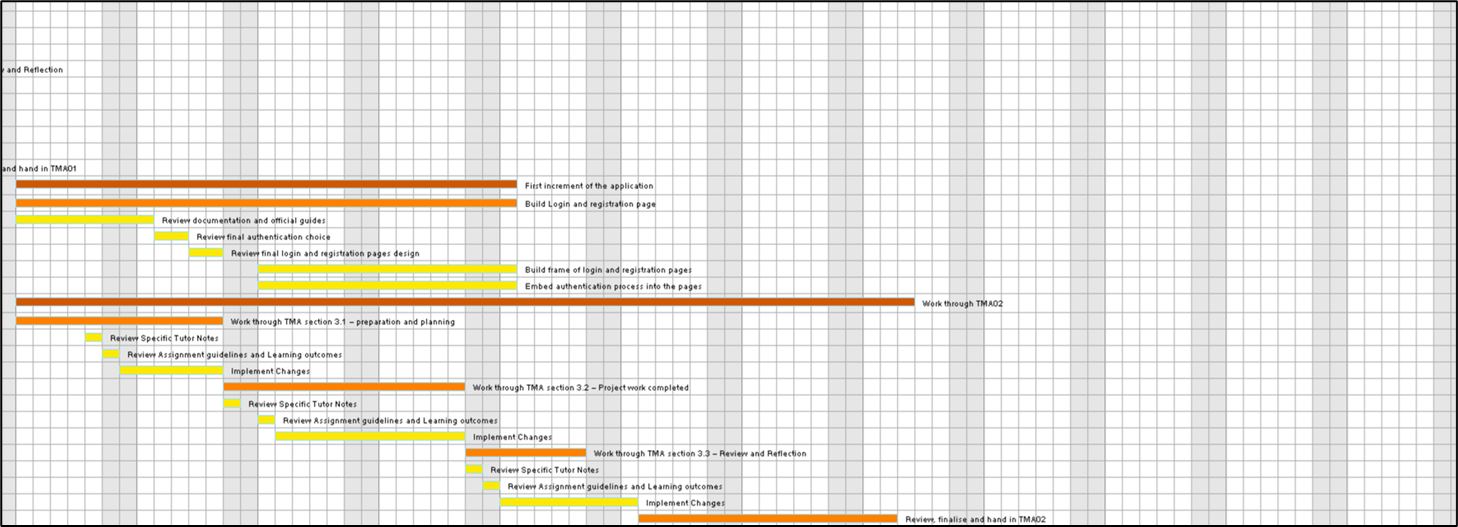


Figure 18 - third stage of Gantt chart

### Appendix 1.4 – Second increment, Third increment and TMA03

Timeline

Description automatically generated

Figure 19 - fourth stage of Gantt chart

### Appendix 1.5 - Final increment, Iteration and EMA

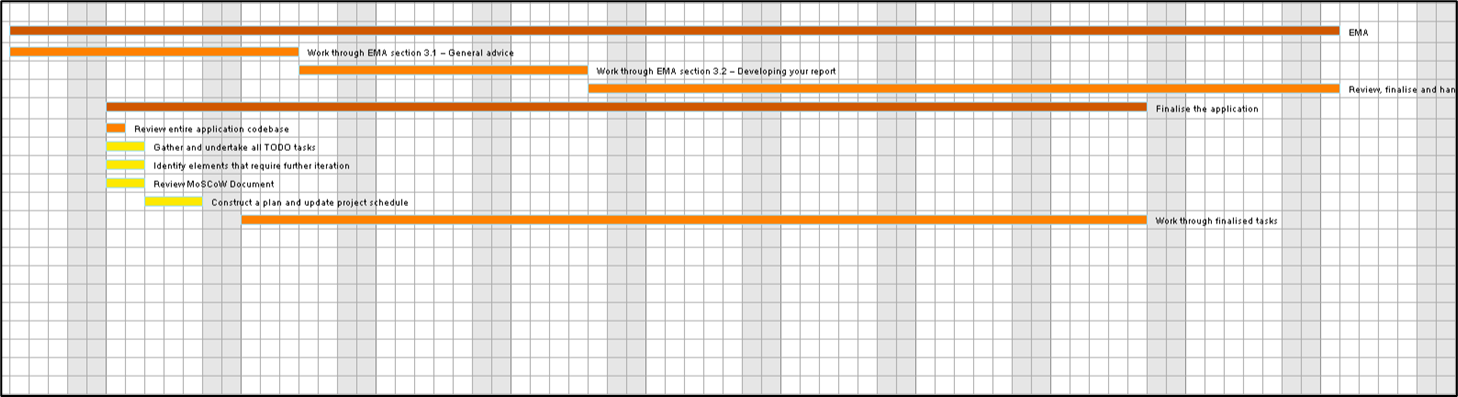
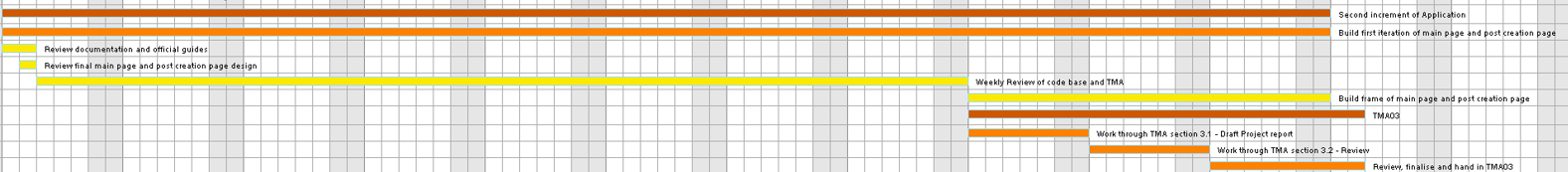


Figure 20 - fifth stage of Gantt chart

### Appendix 1.6 – Second increment and TMA03 (Updated)



### Appendix 1.7 – Final increment, Iteration and EMA (Updated)

A picture containing text, screenshot, line, parallel

Description automatically generated

## Appendix 2 – TMA01 Work completed evidence

### Appendix 2.1 – Old Android Studio file structure

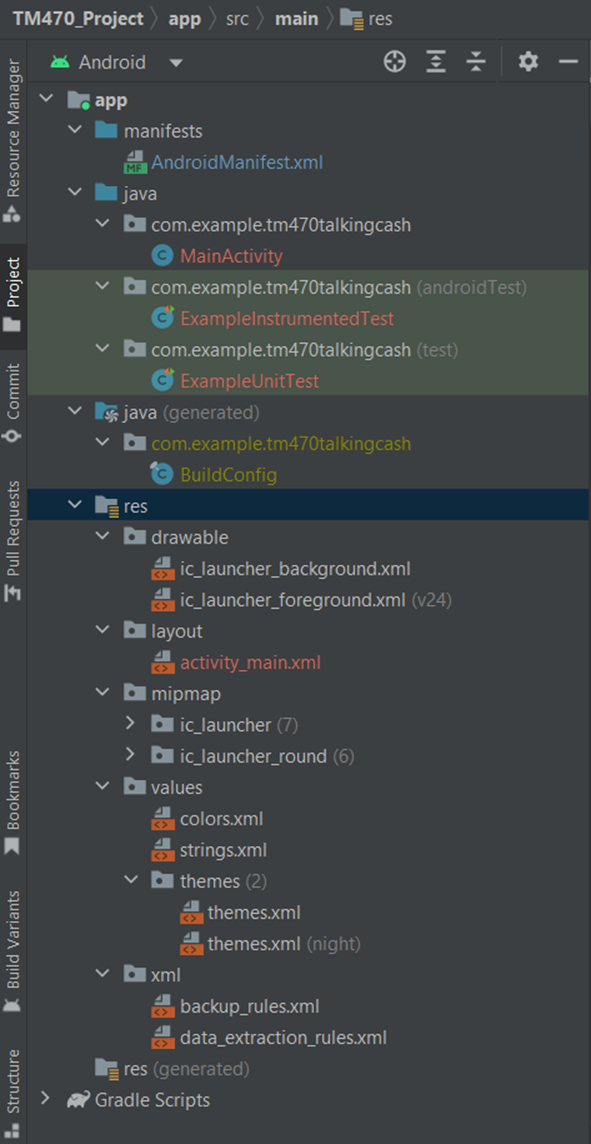


Figure 21 - Android studio file structure

### Appendix 2.2 – Old GitHub file structure

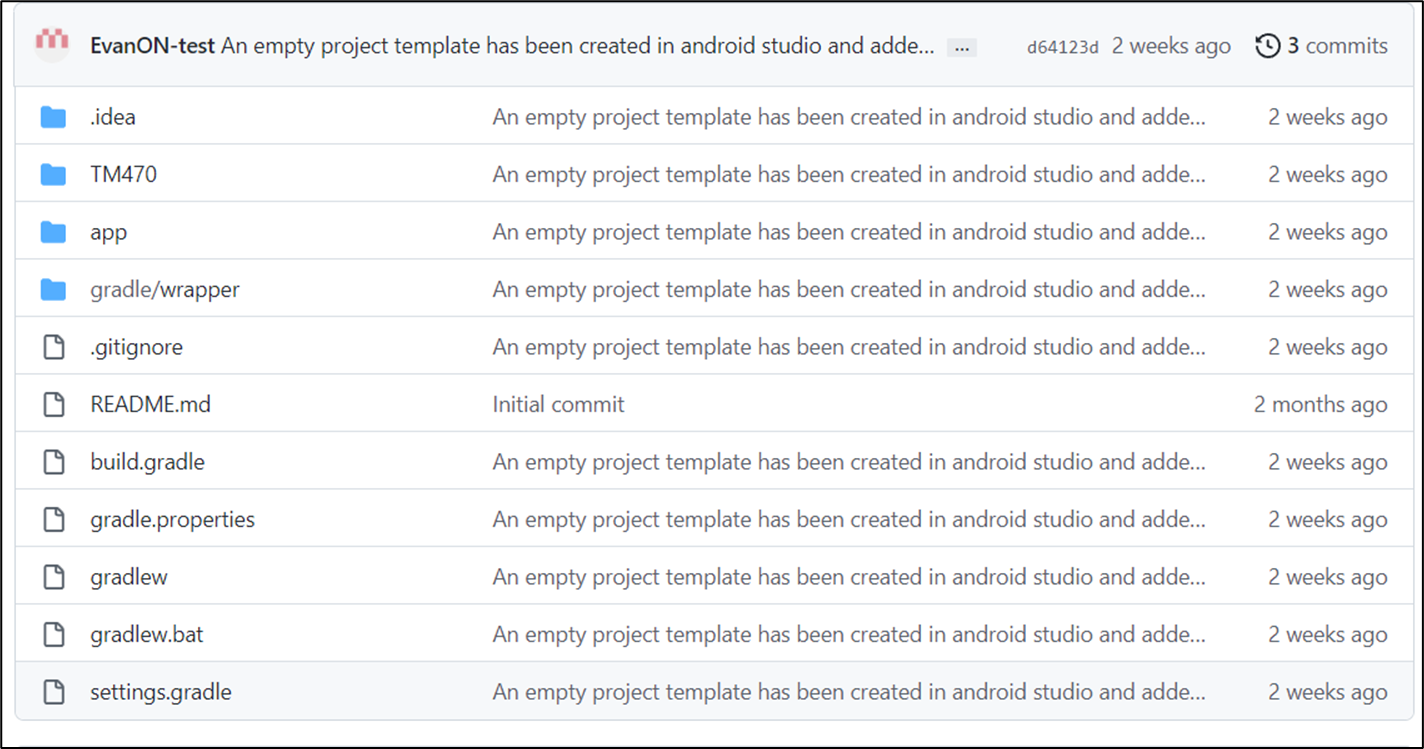


Figure 22 - GitHub file structure

## Appendix 3 – Wireframes

- Incorporating figures 11 - 18

**Diagram

Description automatically generated with medium confidence**

Figure 23 - Login page

**Graphical user interface

Description automatically generated with medium confidence**

Figure 24 - Registration page

Table

Description automatically generated

Figure 25 - Main page

**Diagram

Description automatically generated**

Figure 26 - An expanded view of the post layout as it would appear on the main page

**Graphical user interface

Description automatically generated with low confidence**

Figure 27 - Post creation page

**Diagram

Description automatically generated with medium confidence**

Figure 28 - Discussion page that is linked to each individual post. It will expand from the 'comments' element of a post when clicked

**Diagram

Description automatically generated with medium confidence**

Figure 29 - A view of the menu button when clicked

*Text, letter

Description automatically generated*

Figure 30 - Expanded search page

## Appendix 4 – MoSCoW

A MoSCoW template of some of my thoughts on the initial requirements of the application

**Must Have:**

* A form of login process so that only registered users can use the Application
* A home page consisting of a stream of user sourced posts including article links, website links and links to relevant, trusted social media threads (mainly from twitter and reddit)
* The post creation element must force a common pattern, each post must have a clear and informative title, prominent tags denoting the company and sector as well as the type of information is in the link/article/social media thread (for example financial results, production news or social media based rumors) .
* An enhanced element to the link sharing where by a link preview is shown and not just the simple hyperlink itself
* A simple forum element, attached to each post allowing users to create discussions.
* The ability to filter the main page, either into specific companies, specific sectors (e.g. software, EV, Crypto), and information types.

**Should have:**

* The ability to post PDF files (which are a common format of investor deck).
* An improved search function that will search other aspects not just the tags, for example it will search titles and within comments
* The ability to subscribe to various tags, and a push notification that is initiated any time a post is made about a relevant topic

**Could Have:**

* A simple form of stock watchlist allowing the addition of info such as company name, holdings and money invested and even the extension of a profit/loss element based on the current price sourced only on a limited timescale basis (every hour or every day).

**Will not have:**

* Any form of live share price feed.
* Any form of individual static page for a specific company, although as mentioned the feed can be filtered to show specific companies articles, website links and possibly PDF’s.
* Any further trading information relevant to specific companies outside of the post stream. For example no extra info on analyst expectations or trade volume etc.

## Appendix 5 – Java Code Increment 1

### 

### Appendix 5.1 – LoginActivity

*/\*\*  
 \* Provides the function to sign into the application using their registered email and password, and  
 \* also navigate to the registration activity if required  
 \*/*public class LoginActivity extends AppCompatActivity {  
  
 private TextInputEditText emailText;  
 private TextInputEditText passwordText;  
 private Button loginBtn;  
 private Button registerBtn;  
 private FirebaseAuth fAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 //sets the view to the login activity layout  
 setContentView(R.layout.*activity\_login*);  
 //Gets the firebase instance for the firebase authentication service  
 fAuth = FirebaseAuth.*getInstance*();  
 //Finds the views based on their ID's  
 emailText = findViewById(R.id.*log\_email\_input*);  
 passwordText = findViewById(R.id.*log\_pass\_input*);  
 loginBtn = findViewById(R.id.*log\_button*);  
 registerBtn = findViewById(R.id.*reg\_on\_click*);  
  
 //Sets a click listener for the login button, calls the userAccountLogin method when clicked  
 loginBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
  
 userAccountLogin();  
 }  
 });  
  
 //Sets a click listener for the register button and directs to registration activity when clicked  
 registerBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 Intent intent = new Intent (LoginActivity.this, RegistrationActivity.class);  
 startActivity(intent);  
 }  
 });  
 }  
  
 */\*\*  
 \* This method logs the user into the application if they have a valid email and password  
 \* that is stored in the authentication database  
 \*/* private void userAccountLogin(){  
 String email;  
 String password;  
 Integer minLength;  
  
 //initializes the variables based oin their inputs  
 email = emailText.getText().toString();  
 password = passwordText.getText().toString();  
 minLength = 6;  
  
  
  
  
 //*TODO: review and maybe implement validation techniques in a smarter way. Move toasts to UserValidation?* //Validates the inputs using methods imported from the UserValidation class  
 if (!UserValidation.*isValidEmail*(email)){  
 Toast.*makeText*(getApplicationContext(), "Please enter Valid Email", Toast.*LENGTH\_LONG*).show();  
 return;  
 } else if (!UserValidation.*isValidPassword*(password, minLength)){  
 Toast.*makeText*(getApplicationContext(), "Please enter Valid Password", Toast.*LENGTH\_LONG*).show();  
 return;  
 }  
  
 */\*\*Attempts to sign in to a user account based on their inputted email and password. If successful  
 \* the main activity launches with a affirmative message. If unsuccessful an informative  
 \* error message will be displayed.  
 \*  
 \*/* fAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()){  
 Toast.*makeText*(getApplicationContext(), "Login Successful", Toast.*LENGTH\_LONG*).show();  
 FirebaseUser user = fAuth.getCurrentUser();  
 Intent intent = new Intent(LoginActivity.this, MainActivity.class);  
 intent.putExtra("userEmail", user.getEmail());  
 intent.putExtra("userUid", user.getUid());  
 if (user.getDisplayName()!=null){  
 intent.putExtra("userName", user.getDisplayName());  
 }  
 startActivity(intent);  
 } else {  
 String errorMessage;  
 errorMessage = "Login Failed: ";  
 if (task.getException() != null){  
 errorMessage = errorMessage + task.getException().getMessage();  
 }  
 Toast.*makeText*(getApplicationContext(), errorMessage, Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 });  
 }  
}

### Appendix 5.2 – RegistrationActivity

package com.example.tm470talkingcash;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import androidx.annotation.NonNull;  
import androidx.appcompat.app.AppCompatActivity;  
  
import com.google.android.gms.tasks.OnCompleteListener;  
import com.google.android.gms.tasks.OnSuccessListener;  
import com.google.android.gms.tasks.Task;  
import com.google.android.material.textfield.TextInputEditText;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.firestore.FirebaseFirestore;  
import java.util.Objects;  
  
//*TODO: Make sure the comments are all appropriate i.e. elaborate and make sure they conform to language syntax*//*TODO:In later iterations review other login methods e.g. google accounts/twitter accounts*//*TODO: Also investigate the implementation of an email verification functionality*//*TODO: review the registration process, as in is it worth logging the user in automatically after registration  
  
/\*\*  
 \* provides the functionality to register a new account using a username, email and password.  
 \* Navigates to the login page if successful  
 \*  
 \*/*public class RegistrationActivity extends AppCompatActivity {  
 private TextInputEditText usernameText;  
 private TextInputEditText emailText;  
 private TextInputEditText passwordText;  
  
 private TextInputEditText confirmPasswordText;  
 private Button registerBtn;  
 private FirebaseAuth fAuth;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState){  
 super.onCreate(savedInstanceState);  
 //sets the view to the login activity layout  
 setContentView(R.layout.*activity\_registration*);  
 //Gets the firebase instance for the firebase authentication service  
 fAuth = FirebaseAuth.*getInstance*();  
 //Finds the views based on their ID's  
 usernameText = findViewById(R.id.*reg\_user\_input*);  
 emailText = findViewById(R.id.*reg\_email\_input*);  
 passwordText = findViewById(R.id.*reg\_pass\_input*);  
 confirmPasswordText = findViewById(R.id.*reg\_pass\_confirm\_input*);  
 registerBtn = findViewById(R.id.*reg\_button*);  
  
 //Sets a click listener for the register button, calls the registerNewUser method when clicked  
 registerBtn.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 registerNewUser();  
 }  
 });  
 }  
  
 */\*\*  
 \* This method registers the user into the firebase authentication database and also adds their  
 \* username and email, alongside a generated userId, to the users collection in the Cloud  
 \* firestore database  
 \*/* private void registerNewUser() {  
 String username;  
 String email;  
 String password;  
 String confirmPassword;  
 Integer minLength;  
  
 //initializes the variables based oin their inputs  
 username = usernameText.getText().toString().trim();  
 email = emailText.getText().toString().trim();  
 password = passwordText.getText().toString().trim();  
 confirmPassword = confirmPasswordText.getText().toString().trim();  
 minLength = 6;  
  
 //*TODO: review if more validation techniques are required, restructure into UserValidation class where possible* //Validates the user inputs using methods imported from the UserValidation class  
 if (!UserValidation.*isValidUsername*(username, minLength)){  
 Toast.*makeText*(getApplicationContext(), "Please enter a Valid Username", Toast.*LENGTH\_LONG*).show();  
 return;  
 } else if (!UserValidation.*isValidEmail*(email)){  
 Toast.*makeText*(getApplicationContext(), "Please enter a Valid Email", Toast.*LENGTH\_LONG*).show();  
 return;  
 } else if (!UserValidation.*isValidPassword*(password, minLength)){  
 Toast.*makeText*(getApplicationContext(), "Please enter Valid Password", Toast.*LENGTH\_LONG*).show();  
 return;  
 } else if (!UserValidation.*doPasswordsMatch*(password, confirmPassword)){  
 Toast.*makeText*(getApplicationContext(), "Please make sure that Passwords match", Toast.*LENGTH\_LONG*).show();  
 return;  
 }  
  
  
 */\*\*  
 \* Creates a new user with the email and password combonation, and saves the user profile  
 \* information to 'users' collection in the firestore database  
 \*/* fAuth.createUserWithEmailAndPassword(email, password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 Toast.*makeText*(getApplicationContext(), "Successful Registration", Toast.*LENGTH\_LONG*).show();  
 String userId = Objects.*requireNonNull*(fAuth.getCurrentUser()).getUid();  
 UserProfile userProfile = new UserProfile(userId, username, email);  
 FirebaseFirestore db = FirebaseFirestore.*getInstance*();  
 db.collection("users").document(userId).set(userProfile).addOnSuccessListener(new OnSuccessListener<Void>() {  
 @Override  
 public void onSuccess(Void unused) {  
 Toast.*makeText*(getApplicationContext(), "Registration Details saved to user database", Toast.*LENGTH\_LONG*).show();  
 Intent intent = new Intent(RegistrationActivity.this, LoginActivity.class);  
 startActivity(intent);  
 }  
 });  
 } else {  
 String errorMessage = task.getException().getMessage();  
 Toast.*makeText*(getApplicationContext(), "Registration Failed: " + errorMessage, Toast.*LENGTH\_LONG*).show();  
 }  
 }  
 });  
 }  
}

### Appendix 5.3 – UserProfile

package com.example.tm470talkingcash;  
  
  
  
*/\*\*  
 \* Represents a user with a id, username and email  
 \*/*public class UserProfile {  
 private String userId;  
 private String username;  
 private String userEmail;  
  
 //Constructs a userProfile object  
 public UserProfile(String userId, String username, String userEmail){  
 this.userId = userId;  
 this.username = username;  
 this.userEmail = userEmail;  
 }  
  
 //All the autogenerated getters and setters have been removed   
}

### Appendix 5.4 – UserValidation

package com.example.tm470talkingcash;  
  
  
import android.util.Patterns;  
  
  
  
*/\*\* I thought it would likely comply with OOP by making use of an extra class in which to focus on the  
 \* validation methods  
 \*  
 \*/*public class UserValidation {  
  
//*TODO:Review these techniques again in the next iteration. Could you implement more specific*// *checks and toasts here?* public static boolean isNullOrEmpty(String text){  
 return text == null || text.trim().isEmpty();  
 }  
  
 public static boolean isValidUsername(String username, int minLength){  
 return !*isNullOrEmpty*(username) && username.length() >= minLength;  
 }  
  
 public static boolean isValidEmail(String email){  
 return Patterns.*EMAIL\_ADDRESS*.matcher(email).matches();  
 }  
  
 public static boolean isValidPassword(String password, int minLength){  
 return !*isNullOrEmpty*(password) && password.length() >= minLength;  
 }  
  
 public static boolean doPasswordsMatch(String password, String confirmPassword){  
 return password.equals(confirmPassword);  
 }  
  
  
}

### Appendix 5.5 - MainActivity

package com.example.tm470talkingcash;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
  
  
  
public class MainActivity extends AppCompatActivity {  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 //sets the view to the main activity layout  
 setContentView(R.layout.*activity\_main*);  
  
 //Gets user information from login  
 Intent intent = getIntent();  
 String userEmail = intent.getStringExtra("userEmail");  
 String userUid = intent.getStringExtra("userUid");  
 String userName = intent.getStringExtra("userName");  
 }  
}

## Appendix 6 – XML Layout Increment 1

### 

### Appendix 6.1 – activity\_login

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/root\_constraint\_layout"  
 android:padding="15dp"  
 app:layout\_constraintHeight\_min="600dp"  
 app:layout\_constraintWidth\_min="300dp"  
 tools:context=".LoginActivity">  
  
 <!--*TODO: review yor options in modifying the UI, needs beautifying . Create style.xml*-->  
 <!--*TODO: add background and logo in next iteration*-->  
 <TextView  
 android:id="@+id/appName"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/app\_name"  
 android:textSize="30sp"  
 android:textStyle="bold"  
 android:layout\_marginTop="100dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintVertical\_bias="0.2"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"/>  
  
 <TextView  
 android:id="@+id/welcome"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/log\_welcome\_text"  
 android:layout\_marginTop="30dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintVertical\_bias="0.2"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/appName" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/log\_email\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="250dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/welcome"  
 app:layout\_constraintVertical\_bias="0.5">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/log\_email\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/log\_pass\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="30dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/log\_email\_layout"  
 app:layout\_constraintVertical\_bias="0.5">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/log\_pass\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/pass\_hint"  
 android:inputType="textPassword" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/log\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:text="@string/login"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/log\_pass\_layout"  
 app:layout\_constraintVertical\_bias="0.5"/>  
  
 <Button  
 android:id="@+id/reg\_on\_click"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="50dp"  
 android:clickable="true"  
 android:focusable="true"  
 android:text="@string/register"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/log\_button"  
 app:layout\_constraintVertical\_bias="0.8"/>  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 6.2 – activity\_registration

<?xml version="1.0" encoding="utf-8"?>  
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:id="@+id/root\_constraint\_layout"  
 android:padding="15dp"  
 app:layout\_constraintHeight\_min="600dp"  
 app:layout\_constraintWidth\_min="300dp"  
 tools:context=".RegistrationActivity">  
  
 <!--*TODO: review yor options in modifying the UI, needs beautifying . Create style.xml*-->  
 <!--*TODO: as background and logo in next iteration*-->  
  
 <TextView  
 android:id="@+id/welcome"  
 android:layout\_width="236dp"  
 android:layout\_height="45dp"  
 android:text="@string/reg\_welcome\_text"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="0.5"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintVertical\_bias="0.25" />  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_user\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/welcome">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/reg\_user\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/username"  
 android:inputType="text" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_email\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/reg\_user\_layout">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/reg\_email\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/email\_hint"  
 android:inputType="textEmailAddress" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_pass\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/reg\_email\_layout">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/reg\_pass\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/pass\_hint"  
 android:inputType="textPassword" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/reg\_pass\_confirm\_layout"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="32dp"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@id/reg\_pass\_layout">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/reg\_pass\_confirm\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:hint="@string/confirm\_pass\_hint"  
 android:inputType="textPassword" />  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/reg\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="15dp"  
 android:text="@string/register"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintHorizontal\_bias="1.0"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toBottomOf="@+id/reg\_pass\_confirm\_layout"  
 app:layout\_constraintVertical\_bias="0.1" />  
  
  
  
</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 6.3 – activity\_main

<?xml version="1.0" encoding="utf-8"?>  
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="center"  
 tools:context=".MainActivity">  
  
 <!--the textview and id below are just there as a placeholder so that the main activity  
 isn't blank once logged in-->  
 <TextView  
 android:id="@+id/internalApp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="You're in!! Nothing else here for now though"  
 android:textSize="30sp"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent" />  
  
</ScrollView>

## Appendix 7 – Java Code Increment 2

### 

### Appendix 7.1 – MainThreadActivity

//TODO: Need to implement some kind of unit and integration testing, will need to do some reading on these

//TODO: Remove the old Main activity

//TODO: create a post edit/delete function (this will come after the user profile creation element is flesh out)

//TODO: Also all posts need to have a username/profile associated to the poster so

// create a Profile page whereby users can change their username(which then needs to be linked to posts properly)

// and change emails/passwords ect

//TODO: create a comments page whereby the comments are viewable + can add to the comment thread from the Postdetails view

//TODO: Build a search element into the main page which searches the tags: ideally just a small text thing at top then a search results page?

//TODO: Need to adjust login page so that it is scrollable or is better at adjusting to diffeent phones (josh wasn't able to see register button)

//TODO: think about involving anoter language version, should be simple enough as it's just building of off stuff already in strings XML

/\*\*

\* Displays the user created posts to the users

\*/

public class MainThreadActivity extends AppCompatActivity {

private RecyclerView postRecyclerView;

private FirestoreRecyclerAdapter adapter;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main\_thread);

Button logoutButton = findViewById(R.id.logout\_button);

Button createPostButton = findViewById(R.id.create\_post\_main);

/\*\*

\* Sets click listens for the logout button and create post button respectively. Logging out will also sing out the current user

\*/

logoutButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view){

FirebaseAuth.getInstance().signOut();

Intent intent = new Intent(MainThreadActivity.this, LoginActivity.class);

startActivity(intent);

}

});

createPostButton.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View view){

Intent intent = new Intent(MainThreadActivity.this, PostCreationActivity.class);

startActivity(intent);

}

});

//Reference to the recyclerview layout item in the activity\_main\_thread.xml file

postRecyclerView = findViewById(R.id.post\_recycler\_view);

//Sets layout manager for the recyclerview. This is responsible for arranging the individual elements.

// Linear layout means it's scrollable in a linear way

postRecyclerView.setLayoutManager(new LinearLayoutManager(this));

//Builds FirestoreRecyclerOptions which includes a query to fetch posts and order them by Timestamp in descending order

FirestoreRecyclerOptions<Post> options = new FirestoreRecyclerOptions.Builder<Post>().setQuery(FirebaseFirestore.getInstance().collection("posts").orderBy("timestamp", Query.Direction.DESCENDING), Post.class).build();

//Creates a FirestoreRecyclerAdapter, it listens to chnages in the firebase databse and updates he view accordingly

adapter = new FirestoreRecyclerAdapter<Post, PostViewHolder>(options) {

@NonNull

@Override

//Displays the data at a specific position, updates the contents of the itemview to reflect the post ata given postion

public PostViewHolder onCreateViewHolder(ViewGroup parent, int viewType){

//The LayoutInflater takes the provided xml layout file and inflates it to create the View object.

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.activity\_post\_details, parent, false);

//Returns a new ViewHolder that contains the View for each list item

return new PostViewHolder(view);

}

@Override

public void onBindViewHolder(PostViewHolder holder, int position, Post model){

//Binds the post to the viewholder

holder.bindPost(model);

//sets an onclicklistener to the itemview so that it starts a post details activity when clicked

holder.itemView.setOnClickListener(v -> {

//Creates intent to start post details activity

Intent intent = new Intent(MainThreadActivity.this, PostDetailsActivity.class);

//passes teh post\_id to the post detail activity

intent.putExtra("POST\_ID", model.getPostId());

//starts teh post details activity

startActivity(intent);

});

}

};

postRecyclerView.setAdapter(adapter);

}

/\*\*

\* Sets the adapter to listen (and stop listening) for changes to the Firestore database

\*/

@Override

protected void onStart(){

super.onStart();

if(adapter != null){

adapter.startListening();

}

}

@Override

protected void onStop(){

super.onStop();

if(adapter != null){

adapter.stopListening();

}

}

}

/\*\*

\* Postviewholder represents the layout of each item in the recyclerview

\*/

class PostViewHolder extends RecyclerView.ViewHolder{

TextView titleView;

TextView tagsView;

Button hyperlinkView;

public PostViewHolder(@NonNull View itemView){

super(itemView);

titleView = itemView.findViewById(R.id.post\_deats\_title);

tagsView = itemView.findViewById(R.id.post\_deats\_tags);

hyperlinkView = itemView.findViewById(R.id.post\_deats\_hyperlink);

}

//binds the post to the viewholder

void bindPost (Post post){

titleView.setText(post.getTitle());

tagsView.setText(TextUtils.join(", ", post.getTags()));

hyperlinkView.setText(post.getHyperlink());

}

}

### Appendix 7.2 – Post

package com.example.tm470talkingcash;

import java.util.List;

//TODO: certain setters are not used and could be removed however I intend to review the

// possibility of editing posts and as such do not want to remove them just yet

/\*\*

\* Represents a Post with postid, userid, username, identifying tags, a hyperlink and timestamp.

\* Includes constructors, getters and setters.

\*/

public class Post {

private String postId;

private String userID;

private String title;

private List<String> tags;

private String hyperlink;

private long timestamp;

public Post(){}

public Post(String postId, String userID, String title, List<String> tags, String hyperlink, long timestamp) {

this.postId = postId;

this.userID = userID;

this.title = title;

this.tags = tags;

this.hyperlink = hyperlink;

this.timestamp = timestamp;

}

public String getPostId() {

return postId;

}

public void setPostId(String postId) {

this.postId = postId;

}

public String getUserID() {

return userID;

}

public void setUserID(String userID) {

this.userID = userID;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public List<String> getTags() {

return tags;

}

public void setTags(List<String> tags) {

this.tags = tags;

}

public String getHyperlink() {

return hyperlink;

}

public void setHyperlink(String hyperlink) {

this.hyperlink = hyperlink;

}

public long getTimestamp() {

return timestamp;

}

public void setTimestamp(long timestamp) {

this.timestamp = timestamp;

}

}

### Appendix 7.3 – PostCreationAcitivity

/\*\*

\* TODO: Review XML against your code a final time before testing

\*/

public class PostCreationActivity extends AppCompatActivity {

private TextInputEditText titleText, tagText, hyperlinkText;

private Button postBtn;

private FirebaseFirestore db;

private FirebaseAuth mAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_post\_creation);

titleText = findViewById(R.id.pc\_title\_input);

tagText = findViewById(R.id.pc\_tag\_input);

hyperlinkText = findViewById(R.id.pc\_link\_input);

postBtn = findViewById(R.id.create\_post\_button);

db = FirebaseFirestore.getInstance();

mAuth = FirebaseAuth.getInstance();

postBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

createPost();

}

});

}

/\*\*

\* Creates a post based on the user inputs into title, tag and hyperlink inputs. Some simple validation

\* is implemented. Once validated a post is created based on the user inputs and user details

\*/

private void createPost(){

String title = titleText.getText().toString();

String tagsString = tagText.getText().toString();

String hyperlink = hyperlinkText.getText().toString();

//TODO: review and increase validation here or in a validation class

/\*\*

\* validates that inputs aren't empty and that url is valid

\*/

if (title.isEmpty() || tagsString.isEmpty() || hyperlink.isEmpty()){

Toast.makeText(PostCreationActivity.this, "All fields are required. Please review content ", Toast.LENGTH\_SHORT).show();

return;

}

if(!isValidUrl(hyperlink)){

Toast.makeText(PostCreationActivity.this, "Invalid URL", Toast.LENGTH\_SHORT).show();

return;

}

//TODO: Need to implement validation for this

//Split and trim the list of tags

List<String> tags = new ArrayList<>();

for (String tag : tagsString.split(",")){

tags.add(tag.trim());

}

/\*\*

\* Creates a new post after assessing current user and getting their details as well as their inputs.

\* Adds post to the Posts database. If successful creates a toast in the affirmative and starts MainThreadActivity else

\* creates toast in the negative.

\*/

//TODO: Review if it's a better scenario to implement this in a separate class

if(mAuth.getCurrentUser() != null){

Post post = new Post(UUID.randomUUID().toString(), mAuth.getCurrentUser().getUid(), title, tags, hyperlink, new Date().getTime());

db.collection("posts").document(post.getPostId()).set(post)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void unused) {

Toast.makeText(PostCreationActivity.this, "Post created", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(PostCreationActivity.this, MainThreadActivity.class);

startActivity(intent);

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Toast.makeText(PostCreationActivity.this, "Failed to create Post", Toast.LENGTH\_SHORT).show();

}

});

} else {

Toast.makeText(PostCreationActivity.this, "You are not recognised as logged in", Toast.LENGTH\_SHORT).show();

}

}

}

### Appendix 7.4 – PostDetailsActivity

package com.example.tm470talkingcash;

import static androidx.constraintlayout.helper.widget.MotionEffect.TAG;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.widget.Button;

import android.widget.TextView;

import android.widget.Toast;

import com.google.firebase.firestore.FirebaseFirestore;

public class PostDetailsActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_post\_details);

TextView postTitleTextView = findViewById(R.id.post\_deats\_title);

TextView postTagsTextView = findViewById(R.id.post\_deats\_tags);

Button postHyperlinkButton = findViewById(R.id.post\_deats\_hyperlink);

//Retrieves teh postID that was passed in the intent from the main thread activity

String postID = getIntent().getStringExtra("POST\_ID");

//Fetches the document from the 'posts' firestore collection using the postID

FirebaseFirestore.getInstance().collection("posts").document(postID).get().addOnSuccessListener(documentSnapshot -> {

//converts the firestore document ot a post object

Post post = documentSnapshot.toObject(Post.class);

//Sets the text of the postTitleTextView to the title of the post.

postTitleTextView.setText(post.getTitle());

//Joins the list of tags from the post inot a string seprated buy a comma ','

String tags = TextUtils.join(", ", post.getTags());

postTagsTextView.setText(tags);

//Sets teh text of the posthyperlinkbutton to the hyperlink of the post

postHyperlinkButton.setText(post.getHyperlink());

//Sets an onclick listener on the posthyperlink button that opens the hyperlink in the devices browser when the button is clicked

postHyperlinkButton.setOnClickListener(view -> {

String url = post.getHyperlink();

//TODO: Review if this validation is needed as it's already been validatd in creation. Maybe make sure is implemented fully elsewhere and just have a simple check here

if (url != null && (url.startsWith("http://")) || (url.startsWith("https://"))){

Intent browserIntent = new Intent(Intent.ACTION\_VIEW, Uri.parse(post.getHyperlink()));

startActivity(browserIntent);

} else {

Toast.makeText(this, "Invalid URL", Toast.LENGTH\_SHORT).show();

}

});

})

//If there is an error getting the document then a error toast is shown and entered into the log

.addOnFailureListener(e -> {

Toast.makeText(this, "Error getting document", Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error getting document", e);

});

}

}

## Appendix 8 – XML Layout Increment 2

### 

### Appendix 8.1 – activity\_main\_thread

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainThreadActivity">

<Button

android:id="@+id/logout\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/logout"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<Button

android:id="@+id/create\_post\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/create\_post"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/post\_recycler\_view"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_marginStart="16dp"

android:layout\_marginEnd="16dp"

app:layout\_constraintTop\_toBottomOf="@+id/logout\_button"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintVertical\_bias="0.5"

tools:listitem="@layout/activity\_post\_details"/>

</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 8.2 – activity\_post\_creation

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/root\_constraint\_layout"

android:padding="15dp"

app:layout\_constraintHeight\_min="600dp"

app:layout\_constraintWidth\_min="300dp"

tools:context=".PostCreationActivity">

<TextView

android:id="@+id/welcome"

android:layout\_width="236dp"

android:layout\_height="45dp"

android:text="@string/pc\_welcome"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_bias="0.25" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/pc\_title\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/welcome">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/pc\_title\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/pc\_title"

android:inputType="text" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/pc\_tag\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/pc\_title\_layout">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/pc\_tag\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/pc\_tag"

android:inputType="text" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/pc\_link\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/pc\_tag\_layout">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/pc\_link\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/pc\_link"

android:inputType="textUri" /><!--TODO: not too sure about this, see what it does could always change to 'text'-->

</com.google.android.material.textfield.TextInputLayout>

<Button

android:id="@+id/create\_post\_button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="15dp"

android:text="@string/create\_post"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="1.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/pc\_link\_layout"

app:layout\_constraintVertical\_bias="0.1" />

</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 8.3 – activity\_post\_details

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

tools:context=".PostDetailsActivity">

<!--TODO: This needs modification to make it appear better, must be easy to read what the post is about and see where one post ends and the next begins -->

<TextView

android:id="@+id/post\_deats\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="14sp"

android:layout\_margin="16dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

android:text="@string/post\_details\_username"/>

<TextView

android:id="@+id/post\_deats\_title"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:textSize="24sp"

android:layout\_margin="16dp"

app:layout\_constraintTop\_toBottomOf="@id/post\_deats\_username"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:text="@string/post\_details\_title"/>

<TextView

android:id="@+id/post\_deats\_tags"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:textSize="16sp"

android:layout\_margin="12dp"

app:layout\_constraintTop\_toBottomOf="@id/post\_deats\_title"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:text="@string/post\_details\_tags"/>

<Button

android:id="@+id/post\_deats\_hyperlink"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:textSize="16sp"

android:layout\_margin="12dp"

app:layout\_constraintTop\_toBottomOf="@id/post\_deats\_tags"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:text="@string/post\_details\_content"/>

</androidx.constraintlayout.widget.ConstraintLayout>

## Appendix 9 – Java Code Increment 3

### Appendix 9.1 – AboutActivtiy

package com.example.tm470talkingcash;

import androidx.appcompat.app.AppCompatActivity;

import androidx.appcompat.widget.Toolbar;

import android.content.Intent;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuInflater;

import android.view.MenuItem;

import android.view.View;

import android.widget.Button;

import android.widget.ImageView;

import android.widget.Toast;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

public class AboutActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_about);

//Initialises the toolbar - same in every activity where i hav included a custom toolbar

Toolbar toolbar = (Toolbar) findViewById(R.id.custom\_toolbar);

setSupportActionBar(toolbar);

//Finds the views within the layout to attach onclick listenrs to them

ImageView logoView = findViewById(R.id.logo\_view);

Button createPostButton = findViewById(R.id.create\_post\_main);

/\*\*

\* Sets on click listens for the logo to naviagte to the Mainthreadactivity and the create post button .

\*

\*/

//Refreshed main thread to show whole list of posts

logoView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(AboutActivity.this, MainThreadActivity.class);

startActivity(intent);

}

});

//Navigates to create post activity

createPostButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(AboutActivity.this, PostCreationActivity.class);

startActivity(intent);

}

});

}

/\*\*

\* Create the options menu in the toolbar which currently allows for the navigation to the profile and about pages as well as

\* the logout function

\*

\* @param menu

\* @return

\*/

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_profile:

//Will navigate to the profile activity, puts the users UserID so that the Profile views can populate based on current the users details

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user != null) {

Intent intent = new Intent(AboutActivity.this, ProfileActivity.class);

intent.putExtra("USER\_ID", user.getUid());

startActivity(intent);

} else {

Toast.makeText(this, "ERROR: You're not recognised as logged in", Toast.LENGTH\_SHORT).show();

}

return true;

case R.id.action\_about:

//Navigates to the about page - same for all users

Intent intentTwo = new Intent(AboutActivity.this, AboutActivity.class);

startActivity(intentTwo);

return true;

case R.id.action\_logout:

//Logs users out of the app & instance

FirebaseAuth.getInstance().signOut();

Intent intentThree = new Intent(AboutActivity.this, LoginActivity.class);

startActivity(intentThree);

return true;

default:

return super.onOptionsItemSelected(item);

}

}

}

### Appendix 9.2 – Comment

package com.example.tm470talkingcash;

public class Comment {

private String username;

private String commentText;

private Long timestamp;

private String PostID;

private String commentID;

public Comment(){

}

public Comment(String username, String commentText, long timestamp, String postID, String commentID) {

this.username = username;

this.commentText = commentText;

this.timestamp = timestamp;

this.PostID = postID;

this.commentID = commentID;

}

//TODO: Review unused methods after finalisation - remove the unused

public String getUsername() {

return username;

}

public void setUsername(String username) {

this.username = username;

}

public String getCommentText() {

return commentText;

}

public void setCommentText(String commentText) {

this.commentText = commentText;

}

public Long getTimestamp() {

return timestamp;

}

public void setTimestamp(Long timestamp) {

this.timestamp = timestamp;

}

public String getPostID() {

return PostID;

}

public void setPostID(String postID) {

PostID = postID;

}

public String getCommentID() {

return commentID;

}

public void setCommentID(String commentID) {

this.commentID = commentID;

}

@Override

public String toString() {

return "Comment{" +

"username='" + username + '\'' +

", commentText='" + commentText + '\'' +

", timestamp=" + timestamp +

", PostID='" + PostID + '\'' +

", commentID='" + commentID + '\'' +

'}';

}

}

### Appendix 9.3 – MainPostDetails

/\*\*Currently Empty\*\*/

### Appendix 9.4 – ProfileActivity

public class ProfileActivity extends AppCompatActivity {

private FirebaseFirestore db;

private FirebaseAuth mAuth;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_profile);

Toolbar toolbar = (Toolbar) findViewById(R.id.custom\_toolbar);

setSupportActionBar(toolbar);

ImageView logoView = findViewById(R.id.logo\_view);

Button createPostButton = findViewById(R.id.create\_post\_main);

TextView usernameTextView = findViewById(R.id.text\_username);

TextView emailTextView = findViewById(R.id.text\_email);

Button editProfileButton = findViewById(R.id.edit\_user\_profile);

Button deleteProfileButton = findViewById(R.id.delete\_user\_profile);

db = FirebaseFirestore.getInstance();

mAuth = FirebaseAuth.getInstance();

String userID = mAuth.getCurrentUser().getUid();

/\*\*

\* Sets click listens for the logo (home) button, create post button and edit profile button respectively.

\*

\*/

//Navigates to mainthreadactivity

logoView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(ProfileActivity.this, MainThreadActivity.class);

startActivity(intent);

}

});

//Navigates to create post activity

createPostButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(ProfileActivity.this, PostCreationActivity.class);

startActivity(intent);

}

});

editProfileButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

showPasswordDialog();

}

});

/\*\*

\* Fetches information from the users document, specific to users ID, to populate the text views with their details

\*/

//TODO: removed - String userId = mAuth.getCurrentUser().getUid();

//Fetches the document from the 'posts' firestore collection using the userID

FirebaseFirestore.getInstance().collection("users").document(userID).get().addOnSuccessListener(documentSnapshot -> {

//converts the firestore document ot a userprofile object

UserProfile userProfile = documentSnapshot.toObject(UserProfile.class);

//Sets the text of the postTitleTextView to the title of the post.

usernameTextView.setText(userProfile.getUsername());

emailTextView.setText(userProfile.getUserEmail());

})

//If there is an error getting the document then a error toast is shown and entered into the log

.addOnFailureListener(e -> {

Toast.makeText(this, "Error getting document", Toast.LENGTH\_SHORT).show();

Log.w(TAG, "Error getting document", e);

});

//TODO: Add a password requirement for deleting the profile

/\*\*

\* Deletes document in users collection bsed on the current userID. If succesful will then attempt to dlete the user in Authentication database aswell,

\* Will flash up toasts and tags in the epsonse to teh outcome

\*/

deleteProfileButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

//TODO: removed - String userId = mAuth.getCurrentUser().getUid();

db.collection("users").document(userID).delete().addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

Log.d(TAG, "Document deleted");

mAuth.getCurrentUser().delete().addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()){

Log.d(TAG, "User Account Deleted");

Toast.makeText(getApplicationContext(), "Account deleted", Toast.LENGTH\_LONG).show();

Intent intent = new Intent(ProfileActivity.this, LoginActivity.class);

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(intent);

finish();

} else {

Log.w(TAG, "Error deleting user Authentication account", task.getException());

Toast.makeText(getApplicationContext(), "Error deleting user Authentication account", Toast.LENGTH\_LONG).show();

}

}

});

}

})

.addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.w(TAG, "Error deleting document", e);

Toast.makeText(getApplicationContext(), "Error deleting document", Toast.LENGTH\_LONG).show();

}

});

}

});

}

/\*\*

\* Creates the options menu in the toolbar which currently allows for the navigation to the profile and about pages as well as

\* the logout function

\*

\* @param menu

\* @return

\*/

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_profile:

//navigates to the profile identified by the currewnt user's ID

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user != null) {

Intent intent = new Intent(ProfileActivity.this, ProfileActivity.class);

intent.putExtra("USER\_ID", user.getUid());

startActivity(intent);

} else {

Toast.makeText(this, "ERROR: You're not recognised as logged in", Toast.LENGTH\_SHORT).show();

}

return true;

case R.id.action\_about:

//Navigates tto the about page

Intent intentTwo = new Intent(ProfileActivity.this, AboutActivity.class);

startActivity(intentTwo);

return true;

case R.id.action\_logout:

//Logs out the user

FirebaseAuth.getInstance().signOut();

Intent intentThree = new Intent(ProfileActivity.this, LoginActivity.class);

startActivity(intentThree);

return true;

default:

return super.onOptionsItemSelected(item);

}

}

/\*\*

\* Creates a password check before navigation to profile edit activity

\*/

private void showPasswordDialog(){

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setTitle("Enter Password");

final EditText input = new EditText(this);

input.setInputType(InputType.TYPE\_CLASS\_TEXT | InputType.TYPE\_TEXT\_VARIATION\_PASSWORD);

builder.setView(input);

//Actions taken when confirm button in alert dialog pressed. Essentially uses the check password method further below

builder.setPositiveButton("Confirm", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

String enteredPassword = input.getText().toString();

checkPasswordAndRedirect(enteredPassword);

}

});

//Actions taken when cancel button in alert dialog pressed. Cancels the event

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

dialogInterface.cancel();

}

});

builder.show();

}

//TODO: need to provide more actions in the event of a wrong password here

//Gets curent users email and actions a another sign in event

private void checkPasswordAndRedirect(String enteredPassword){

String emailCheck = FirebaseAuth.getInstance().getCurrentUser().getEmail();

FirebaseAuth.getInstance().signInWithEmailAndPassword(emailCheck, enteredPassword).addOnCompleteListener(task -> {

if (task.isSuccessful()){

Intent intent = new Intent(ProfileActivity.this, ProfileEditActivity.class);

startActivity(intent);

}

});

}

}

### Appendix 9.5 – ProfileEditActivity

public class ProfileEditActivity extends AppCompatActivity {

private TextInputEditText usernameText;

private TextInputEditText emailText;

private TextInputEditText confirmNewPasswordText;

private TextInputEditText newPasswordText;

private Button saveBtn;

private FirebaseAuth fAuth;

private FirebaseFirestore db;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_profile\_edit);

ImageView logoView = findViewById(R.id.logo\_view);

Button createPostButton = findViewById(R.id.create\_post\_main);

db = FirebaseFirestore.getInstance();

fAuth = FirebaseAuth.getInstance();

//Finds the views based on their ID's

usernameText = findViewById(R.id.edit\_username\_input);

emailText = findViewById(R.id.edit\_email\_input);

newPasswordText = findViewById(R.id.edit\_password\_input);

confirmNewPasswordText = findViewById(R.id.confirm\_edit\_password\_input);

saveBtn = findViewById(R.id.save\_user\_profile);

/\*\*

\* Sets click listens for the logo (home) button, create post button and save profile edit button respectively.

\*

\*/

//Refreshed main thread to show whole list of posts

logoView.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(ProfileEditActivity.this, MainThreadActivity.class);

startActivity(intent);

}

});

//Navigates to create post activity

createPostButton.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent intent = new Intent(ProfileEditActivity.this, PostCreationActivity.class);

startActivity(intent);

}

});

//Saves inputs when clicked

saveBtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

saveProfileEdit();

}

});

String currentUserID = fAuth.getCurrentUser().getUid();

DocumentReference docRef = db.collection("users").document(currentUserID);

docRef.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {

@Override

public void onComplete(@NonNull Task<DocumentSnapshot> task) {

if (task.isSuccessful()) {

DocumentSnapshot document = task.getResult();

if (document.exists()) {

String username = document.getString("username");

String email = document.getString("userEmail");

usernameText.setText(username);

emailText.setText(email);

} else {

Log.d(TAG, "No such document");

Toast.makeText(getApplicationContext(), "No such document", Toast.LENGTH\_SHORT).show();

}

} else {

Log.d(TAG, "Get failed", task.getException());

Toast.makeText(getApplicationContext(), "Get failed", Toast.LENGTH\_SHORT).show();

}

}

});

}

/\*\*

\* Creates the options menu in the toolbar which currently allows for the navigation to the profile and about pages as well as

\* the logout function

\*

\* @param menu

\* @return

\*/

@Override

public boolean onCreateOptionsMenu(Menu menu) {

MenuInflater inflater = getMenuInflater();

inflater.inflate(R.menu.menu\_main, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_profile:

//navigates to the profile identified by the currewnt user's ID

FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();

if (user != null) {

Intent intent = new Intent(ProfileEditActivity.this, ProfileActivity.class);

intent.putExtra("USER\_ID", user.getUid());

startActivity(intent);

} else {

Toast.makeText(this, "ERROR: You're not recognised as logged in", Toast.LENGTH\_SHORT).show();

}

return true;

case R.id.action\_about:

//Navigate to about page

Intent intentTwo = new Intent(ProfileEditActivity.this, AboutActivity.class);

startActivity(intentTwo);

return true;

case R.id.action\_logout:

//Logs user out

FirebaseAuth.getInstance().signOut();

Intent intentThree = new Intent(ProfileEditActivity.this, LoginActivity.class);

startActivity(intentThree);

return true;

default:

return super.onOptionsItemSelected(item);

}

}

/\*\*

\* This method updaets the users details based on their inputs

\* \*\*/

private void saveProfileEdit(){

String username;

String email;

String confirmNewPassword;

String newPassword;

Integer minLength;

//initializes the variables based oin their inputs

username = usernameText.getText().toString().trim();

email = emailText.getText().toString().trim();

newPassword = newPasswordText.getText().toString().trim();

confirmNewPassword = confirmNewPasswordText.getText().toString().trim();

minLength = 6;

//TODO: review if more validation techniques are required, restructure into UserValidation class where possible. Maybe change this to Case structure?

//Validates the user inputs using methods imported from the UserValidation class

if (!UserValidation.isValidUsername(username, minLength)){

Toast.makeText(getApplicationContext(), "Please enter a Valid Username (greater than 6 characters)", Toast.LENGTH\_LONG).show();

return;

} else if (!UserValidation.isValidEmail(email)){

Toast.makeText(getApplicationContext(), "Please enter a Valid Email", Toast.LENGTH\_LONG).show();

return;

} else if (!UserValidation.isValidPassword(newPassword, minLength)){

Toast.makeText(getApplicationContext(), "Please enter Valid Password (greater than 6 characters)", Toast.LENGTH\_LONG).show();

return;

} else if (!UserValidation.doPasswordsMatch(newPassword, confirmNewPassword)){

Toast.makeText(getApplicationContext(), "Please ensure you have confirmed passwords appropriately", Toast.LENGTH\_LONG).show();

return;

}

//Gets userID and creates a docref to the specific user document based on that ID

String currentUserID = fAuth.getCurrentUser().getUid();

DocumentReference docRef = db.collection("users").document(currentUserID);

Map<String, Object> updates = new HashMap<>();

updates.put("username", username);

updates.put("userEmail", email);

//Updates the specific docuemnt based on the Hashmap. On success will also update the auth with hte new details

docRef.update(updates).addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void aVoid) {

// Updates Firebase Authentication

fAuth.getCurrentUser().updateEmail(email).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

fAuth.getCurrentUser().updatePassword(newPassword).addOnCompleteListener(new OnCompleteListener<Void>() {

@Override

public void onComplete(@NonNull Task<Void> task) {

if (task.isSuccessful()) {

Toast.makeText(getApplicationContext(), "Profile updated successfully", Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(ProfileEditActivity.this, ProfileActivity.class);

intent.setFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

startActivity(intent);

} else {

Log.w(TAG, "Error updating password", task.getException());

Toast.makeText(getApplicationContext(), "Error updating password", Toast.LENGTH\_SHORT).show();

}

}

});

} else {

Log.w(TAG, "Error updating email", task.getException());

Toast.makeText(getApplicationContext(), "Error updating email", Toast.LENGTH\_SHORT).show();

}

}

});

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

Log.w(TAG, "Error updating Firestore document", e);

Toast.makeText(getApplicationContext(), "Error updating Firestore document", Toast.LENGTH\_SHORT).show();

}

});

}

}

## Appendix 10 – XML Layout Increment 3

### Appendix 10.1 – activity\_about

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".AboutActivity">

<include

layout="@layout/custom\_toolbar\_bare"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"/>

<androidx.constraintlayout.widget.Guideline

android:id="@+id/guideline"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"/>

<ImageView

android:id="@+id/imageView"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:src="@drawable/tm470\_app\_logo"

android:scaleType="fitCenter"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintDimensionRatio="H,2:1" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:layout\_margin="8dp"

android:text="@string/about\_extended"

android:textAlignment="center"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/imageView" />

</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 10.2 – activity\_main\_post\_details

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".AboutActivity">

<include

layout="@layout/custom\_toolbar\_bare"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"/>

<androidx.constraintlayout.widget.Guideline

android:id="@+id/guideline"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"/>

<ImageView

android:id="@+id/imageView"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:src="@drawable/tm470\_app\_logo"

android:scaleType="fitCenter"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintDimensionRatio="H,2:1" />

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:padding="10dp"

android:layout\_margin="8dp"

android:text="@string/about\_extended"

android:textAlignment="center"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/imageView" />

</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 10.3 – item\_comment

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:padding="8dp">

<TextView

android:id="@+id/comment\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textStyle="bold" />

<TextView

android:id="@+id/comment\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp" />

<TextView

android:id="@+id/timestamp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp" />

</LinearLayout>

### Appendix 10.4 – activity\_profile

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ProfileActivity">

<include

layout="@layout/custom\_toolbar\_bare"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"/>

<!--TODO: this imageview is empty and a placeholder for now-->

<ImageView

android:id="@+id/profile\_image"

android:contentDescription="@string/profile\_image"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:padding="50dp"/>

<TextView

android:id="@+id/text\_username\_label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/post\_details\_username"

android:textAllCaps="true"

android:textSize="20sp"

android:layout\_marginTop="100dp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/profile\_image"

app:layout\_constraintBottom\_toTopOf="@id/text\_username"/>

<TextView

android:id="@+id/text\_username"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="15sp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/text\_username\_label"/>

<TextView

android:id="@+id/text\_email\_label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/email\_hint"

android:textAllCaps="true"

android:textSize="20sp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/text\_username"/>

<TextView

android:id="@+id/text\_email"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textSize="15sp"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/text\_email\_label"/>

<Button

android:id="@+id/delete\_user\_profile"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="16dp"

android:padding="15dp"

android:text="@string/delete\_account"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/edit\_user\_profile"

app:layout\_constraintStart\_toStartOf="parent"/>

<Button

android:id="@+id/edit\_user\_profile"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="16dp"

android:padding="15dp"

android:text="@string/edit\_profile"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/delete\_user\_profile"/>

</androidx.constraintlayout.widget.ConstraintLayout>

### Appendix 10.5 – activity\_profile\_edit

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schema

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".ProfileActivity">

<include

layout="@layout/custom\_toolbar\_bare"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="0dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"/>

<!--TODO: this imageview is empty and a placeholder for now-->

<ImageView

android:id="@+id/profile\_image"

android:contentDescription="@string/profile\_image"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toBottomOf="@id/custom\_toolbar"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

android:padding="50dp"/>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/edit\_username\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:padding="15dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/profile\_image">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/edit\_username\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/username"

android:inputType="text" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/edit\_email\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:padding="15dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/edit\_username\_layout">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/edit\_email\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/email\_hint"

android:inputType="text" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/edit\_password\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:padding="15dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/edit\_email\_layout">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/edit\_password\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/pass\_hint"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/confirm\_edit\_password\_layout"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:padding="15dp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/edit\_password\_layout">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/confirm\_edit\_password\_input"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:hint="@string/confirm\_pass\_hint"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

### Appendix 10.6 – activity\_toolbar

<?xml version="1.0" encoding="utf-8"?>

<androidx.appcompat.widget.Toolbar

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal">

<ImageView

android:id="@+id/logo\_view"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:src="@drawable/tm470\_app\_logo"

android:scaleType="fitCenter"

android:layout\_weight="1"

/>

<androidx.appcompat.widget.SearchView

android:id="@+id/toolbar\_searchview"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="3"

android:layout\_gravity="center\_vertical"/>

<Button

android:id="@+id/create\_post\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:layout\_gravity="center\_vertical"

android:text="@string/create\_post"/>

</LinearLayout>

</androidx.appcompat.widget.Toolbar>

### Appendix 10.7 – activity\_toolbar\_bare

<?xml version="1.0" encoding="utf-8"?>

<androidx.appcompat.widget.Toolbar

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/custom\_toolbar"

android:layout\_width="match\_parent"

android:layout\_height="?attr/actionBarSize"

android:background="?attr/colorPrimary">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="horizontal">

<ImageView

android:id="@+id/logo\_view"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:src="@drawable/tm470\_app\_logo"

android:scaleType="fitCenter"

android:layout\_weight="1"/>

<Button

android:id="@+id/create\_post\_main"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="3"

android:layout\_gravity="center\_vertical"

android:text="@string/create\_post"/>

</LinearLayout>

</androidx.appcompat.widget.Toolbar>

## Appendix 11 – Java code final increment

## Appendix 12 – XML Layout final increment

## Appendix 13 - Communications Log

\*As of 08/07/2023\*

|  |
| --- |
| **Welcome to TM470B for 2023 Ref: 263095981** 4 messages |

|  |  |
| --- | --- |
| **darren.bolton@open.ac.uk**<darren.bolton@open.ac.uk> | 25 January 2023 at 19:38 |
| To: e22neale22@gmail.com | |
| |  | | --- | | Hi, my name is Darren Bolton and I am your TM470B tutor for 2023 !  I have posted an initial welcome post on our tutor group Forum - please have a read of that and reply to that post to confirm that you are seeing the forum. (don't worry if you cant see it - I will paste a copy of the post at the end of this email, but please let me know if you cant access the tutor group forum.)  For now, just reply with the details as indicated below so that we can get a line of communication established.  Just to put your minds at rest from the start - my job is to help you get the best pass grade possible and I will do my best to help you in whatever way I can !  Text of post on Tutor group forum...  "Hi everyone and welcome to TM470B for 2023 !  My name is Darren Bolton and I will be your module tutor this year.  I will shortly be sending everyone an introductory email, so if you could just reply to that email to ensure we have good, established, private communication, via email.  It would be useful if everyone could include in that email the full thread of their posting on the preparation forum. (if you haven't been able to do that, have a look at the posts that have been made and send that detail to me in an email). We can then start discussing your ideas for your project and hopefully come up with an achievable project for you.  I am going to propose holding an initial 'live' group tutorial on Sunday morning (5th February 2023) at 10:00hrs (UK time) where I will run through an introductory powerpoint, which gives an overview of the module and some specific information about TMA 01. I will also give you up to date information from last year's TM470 module in terms of "do's and dont's" as you head into this year's project.  Right at the outset - If I can make two suggestions, which will pay dividends as you near the EMA hand in date in September...  1. Start a project journal and record everything you do on your project, everything from initial planning and design to decision making to coding. (decisions in your journal may well correspond to amended versions of your GANNT charts [more on that in the tutorial] )  2. Keep a communications log - record all interactions with me, including emails, online tutorials and also your TMA feedback.  The headline here is - help me to help you to get the best grade possible by providing me with the evidence to justify it (remember the final EMA is second marked by a different TM470 tutor) !!!  Let me know if Sunday 5th February works for you. If not, let me know whether mid week evenings would be better.  Don't worry if you cant make date of the initial tutorial, I will upload the introductory powerpoint and a second powerpoint which gives you some background information about myself, such as my employment history and qualifications.  If there is demand for it, I can run a second introductory group tutorial on a different date, as getting all of my students together at the same time is not always possible due to many different factors.  I am going to run these live tutorials on the OU's preferred web based application, 'Adobe Connect'. More details on that once we have a confirmed date(s) for our first tutorial.  There is no need to have a camera for video if you don't want to, but speakers are needed as a minimum and a microphone will be required if you wish to ask any questions.  From the 1st of February (the official course start date), I will be making myself available for 30 minute, 1:1, tutorials (using Adobe Connect) on weekend or midweek evenings. To book a time slot, please email me with preferred date and time and I will get back to you to confirm the booking, or to suggest an alternative, if I can't make your suggested date & time.  I look forward to seeing some of you at our first tutorial and I anticipate your emails in due course.  Please reply to this post with a quick confirmation so that I can be assured that everyone has sight of our tutor group forum  Regards,  Darren"  -- The Open University is incorporated by Royal Charter (RC 000391), an exempt charity in England & Wales and a charity registered in Scotland (SC 038302). The Open University is authorised and regulated by the Financial Conduct Authority in relation to its secondary activity of credit broking. | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 26 January 2023 at 09:32 |
| To: darren.bolton@open.ac.uk | |
| |  | | --- | | Hi Darren,  I've seen both your Email and Forum post (It hasn't opened for replies just yet otherwise I'd have confirmed within the forum) so we're all good on those fronts. I'll paste my posts from the project preparation forum below. Hopefully it all looks ok and I'm interested to get your thoughts on the project idea.  **Post 1**  **Which L3 course?** TM352 - Web, Mobile and Technologies  **What is the Problem?** Myself and a few friends have a keen interest in investing. We all share relevant information (articles/company releases/financial statements) in a group chat which, although convenient, means that it is just a untidy stream of links with no deeper functionality or ability to further personalise the information.  **The stakeholders:** Myself  **The user:**  Myself, my friends and family.  **I will be able to find out more by talking to:** My friends and other interested parties within my investment social media circle.  **The skills I have from my L3 study include these which are relevant to the problem:** TM352 covers the development of apps which will be most relevant. TM354 covers the processes to be used in software development which will be most useful when developing the application.  **The ways in which I will be moving my skills and knowledge beyond the L3 module(s)  are:**  Developing a functioning, java based android app will require deepening my java knowledge and familiarity with the frameworks, technologies and development software which isn't covered in the module.  **Any legal or ethical ramifications that might occur:** There is a multitude of legal ramifications in the world of investing. Making clear that this is only a simple app for dissemination of information from official sources and light discussion and not a place for official investing advice. Also the legal and ethical ramifications with dealing with members personal data.  **The end product I will deliver will be:** A java based android app that acts as a hub of investing information and discussion for friends and family. The App will have multiple functionalities that are only unlocked when registered as a member. The app will incorporate streams of information posted by both myself (the creator) and registered members alongside forum elements and other elements which will be decided during development.  **Post 2**  Hi Ju,  Yeah I'll be able to get others to test it.  Ok I'll have a look at that and have a think on what would be realistically achievable. Will come back to this after a bit of other TMA work.  **Post 3**  Hi Ju,  At the moment I’ve knocked up a few thoughts and I’d like to get your opinion on what you think so far?  I think this is something I’ll review again in a few days and add to a little more with a fresh mind. I’ll be more specific at that point, removing the ‘ect.’ And ‘e.g.’, and I’ll also expand the list a little.    **Must Have:**   * A form of login process so that only registered users can use the Application * A home page consisting of a stream of user sourced posts including article links, website links and links to relevant social media threads (e.g. twitter and reddit) * The posts must be In a common pattern, e.g. they must have a title and prominent tags denoting the company and sector as well as what type of information is in the link/article/social media thread (for example financial results, production news or social media based rumours ) . * A forum element, allowing users to create discussions.   **Should have:**   * The ability to post PDF files (which are a common format of investor deck). * The ability to filter the main page, either into specific companies, specific sectors (e.g. software, EV, Crypto), and information types.   **Could Have:**   * A simple form of stock watchlist allowing the addition of info such as company name, holdings and money spent  and even the possible extension of a profit or loss element based on the current price. * A comment section specific to each article.   **Will not have:**   * Any form of live share price feed. * Any form of individual static page for a specific company, although as mentioned the feed can be filtered to show specific companies articles, website links and possibly PDF’s. * Any further trading information relevant to specific companies outside of the news stream. For example no extra info on analyst expectations or trade volume etc.     Evan | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 29 January 2023 at 12:53 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Hi Evan, thanks for getting back to me.  I think the idea is a great starting point if it is relevant to you personally, which will make the development and management of the project much more interesting than picking some arbitrary topic.  However, I am perhaps not quite seeing the main 'functionality' of the app you are proposing to develop.  If I have understood correctly, the information gathered (and comments) by yourself and your investment media circle would be made available within the developed app for other users to search and gain information about a specific company ?  I would initially suggest that the coding for this might not be sufficiently demanding enough to give the project a wide enough scope to attract a high grade.  The scope of a project is always difficult to find the right balance for. Between a project that is massively out of scope (e.g. developing an electric car recharging App for the whole of the United Kingdom) and a project that is too simplistic and very easy to implement in a very short space of time (e.g. developing a noughts and crosses app) and therefore wouldn't have enough 'substance' to gain a high grade for the EMA.  Could I suggest that you review existing stock investment apps and try and identify something unique that your app will introduce that is not available in other existing ones ?    The development phase might be piloted in your own environment but the aspiration should perhaps be to make the app available for other users.  You will most definitely want to include consideration of Legal and Ethical issues as there is a specific Learning outcome for TMA2 which you will need to provide evidence of:  **LO10: Identify and address the legal, social, ethical and professional issues (LSEPIs) and the equality, diversity and inclusion (EDI) concerns that may arise during the development and use of computing and IT systems.**  As an example, you may want to consider the possibility of someone gaining access to information on which they purchase shares. If those shares drop in value, would that user be able to challenge the owner/developer of the app legally for compensation ?  I strongly recommend reviewing the Learning Outcomes for the first TMA and I have attached a useful document (the information is also available on the TM470 website) which contains the Learning Outcomes against which I have to mark all the TMA's and the final EMA.  The development of an 'artifact' (the final coded application) is quite important if you are wanting to use the TM470 project as part of an application for accreditation to the British Computer Society but the main bulk of the marks for TM470 are awarded for project planning and management !  I hope this helps. Please come back to me if you have any further questions and I hope to see you at our first tutorial on Sunday 5th February at 10:00 if you can make it ?  Regards,  Darren  TM470 Tutor | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 2 February 2023 at 10:11 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Thanks for such an informative response and apologies in advance for such a long winded reply.  My previous email contained my initial thoughts on what the project should be. I've been working through the first TMA and had time to think on my reasons why it's needed and how it should look ect and I'll try to elaborate and see if that sounds better.  Yeah I believe your understanding is correct.  The App would require users to have to register as a member to use the app, and only I would be responsible for accepting or denying these registration requests. Once accepted users can make posts containing website links to News articles, company websites and social media comment threads (to name a few); All of these posts would initially appear as a stream of chronological posts; when each post is made it would have to be created in a set format, to help readability, and it would require tags (I've moved this from the 'should have' to 'must have' MoSCoW group); the user would then be able to filter the thread as required looking for posts about specific companies, sectors or geographic areas (to name a few example tags) and even filter using multiple tags. It would initially be for only a select group of friends but with the aim to make it available to extended acquaintances. I would aim to have a relatively controlled user base though some form of invite only set up.  This is a topic that interests me so Ideally I'd like to build out this project and expand on it to make sure it's elaborate enough. Would moving some of the other elements from the 'should', 'could' and 'wouldn't have' sections of Moscow make it elaborate enough a project? Or should I think of some more things that I could add to give it a wider scope?  I'll have to review a few of the more mainstream investing apps out there and see if I can think up something to make it a little more unique.  Yeah there will definitely be a considerable legal element I'd need to thoroughly investigate and I'd need to make sure to create the app with some sort of legal disclaimer to negate the issues that could arise with the example you mentioned (although thinking about it this would probably require some legal expertise). Ultimately all the information being exchanged would be publicly available and any user discussion would just be 'friendly conversation' around this information, by keeping it between close friends this shouldn't be a major issue until it is opened up to a larger audience.  I've actually rarely attended Tutorials during my studies, I tend to prefer using the prerecorded ones that become available as I find them more convenient however I will definitely make attending the one for this module my priority.  Sincerely,  Evan  On Sun, 29 Jan 2023 at 12:53, Darren.Bolton <[darren.bolton@open.ac.uk](mailto:darren.bolton@open.ac.uk)> wrote:  Hi Evan, thanks for getting back to me.  I think the idea is a great starting point if it is relevant to you personally, which will make the development and management of the project much more interesting than picking some arbitrary topic.  However, I am perhaps not quite seeing the main 'functionality' of the app you are proposing to develop.  If I have understood correctly, the information gathered (and comments) by yourself and your investment media circle would be made available within the developed app for other users to search and gain information about a specific company ?  I would initially suggest that the coding for this might not be sufficiently demanding enough to give the project a wide enough scope to attract a high grade.  The scope of a project is always difficult to find the right balance for. Between a project that is massively out of scope (e.g. developing an electric car recharging App for the whole of the United Kingdom) and a project that is too simplistic and very easy to implement in a very short space of time (e.g. developing a noughts and crosses app) and therefore wouldn't have enough 'substance' to gain a high grade for the EMA.  Could I suggest that you review existing stock investment apps and try and identify something unique that your app will introduce that is not available in other existing ones ?    The development phase might be piloted in your own environment but the aspiration should perhaps be to make the app available for other users.  You will most definitely want to include consideration of Legal and Ethical issues as there is a specific Learning outcome for TMA2 which you will need to provide evidence of:  **LO10: Identify and address the legal, social, ethical and professional issues (LSEPIs) and the equality, diversity and inclusion (EDI) concerns that may arise during the development and use of computing and IT systems.**  As an example, you may want to consider the possibility of someone gaining access to information on which they purchase shares. If those shares drop in value, would that user be able to challenge the owner/developer of the app legally for compensation ?  I strongly recommend reviewing the Learning Outcomes for the first TMA and I have attached a useful document (the information is also available on the TM470 website) which contains the Learning Outcomes against which I have to mark all the TMA's and the final EMA.  The development of an 'artifact' (the final coded application) is quite important if you are wanting to use the TM470 project as part of an application for accreditation to the British Computer Society but the main bulk of the marks for TM470 are awarded for project planning and management !  I hope this helps. Please come back to me if you have any further questions and I hope to see you at our first tutorial on Sunday 5th February at 10:00 if you can make it ?  Regards,  Darren  TM470 Tutor | | |

|  |
| --- |
| **Evan Neale (F2724895) - TM470 - TMA02** 10 messages |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 23 March 2023 at 09:25 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  I hope all is well with you. Thanks for all the feedback you provided in TMA01 it's proved very helpful and I've already planned the changes I need to make.  So far everything is on schedule, even a little ahead of it. I've managed to create the registration XML and Java files as well as the XML for the login. I've also set up a firebase account which I'm using for authentication, and I've integrated it into the Java file (not tested yet though). I still need to work on the Java element for the login and tie everything together then test it all works.  To make it easier to mark, would you like me to highlight or use different text colours for all the extra text/changes I've added in this TMA02?  Am I right in thinking I need to keep in contact with you every few weeks to let you know where I'm at and if everythings going ok regarding the project?  Thanks,  Evan | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 23 March 2023 at 18:07 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Hi Evan, thank you for the email and yes, please just a short summary of progress every few weeks or so which gives me documentary evidence of communication.  Dont worry about the colouring for changes - I will be able to pick out where you have made changes in line with the TMA01 feedback.  Good to see your progression towards TMA02.  Regards,  Darren  TM470 Tutor  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 23 March 2023 09:25 **To:** Darren.Bolton <[darren.bolton@open.ac.uk](mailto:darren.bolton@open.ac.uk)> **Subject:** Evan Neale (F2724895) - TM470 - TMA02    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 6 April 2023 at 13:59 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Just a few queries regarding the TMA.  I'm currently putting together the 'work done' section and at the moment I've decided that in regards to the main Java and XML files I'll outline what code I've implemented and then maybe add the extended code into appendix's. Does this sound reasonable?  When I paste code into the TMA02 document I'll take out any of the boilerplate code however there are some bits (like the authentication and database bits) which have a set structure, that I've taken and adjusted, from the firebase documentation. Assuming I have referenced these there shouldn't be any issues there?  Thanks,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 6 April 2023 at 14:05 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Evan, that sounds the right approach - placing code 'snippets' in the appendices and referring to them in the main body of the report.  There is never a problem, with regards to plagiarism, when you cite and reference correctly, as you are acknowledging the author.  There can be problems with copyright if the code used is not Open Source, so you may need to consider this aspect.  Regards,  Darren  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 06 April 2023 13:59 **To:** Darren.Bolton <[darren.bolton@open.ac.uk](mailto:darren.bolton@open.ac.uk)> **Subject:** Re: Evan Neale (F2724895) - TM470 - TMA02    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 6 April 2023 at 14:33 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | That's great.  Thanks for such a speedy reply.  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 13 April 2023 at 13:07 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  I've got another query for you :)  So in my TMA02 document I've iterated over my Gantt chart and placed it into the 'schedule' section, replacing the old one.  However I've just got to the 'future plan' section of the document and it mentions:  Future plan  ***You should give an updated schedule of future activity for the project. You may use a format of your choosing for the schedule, as long as it is clear and can be inserted as text or a table in your TMA document. Your tutor will make an assessment of whether your approach is justified in relation to the nature of your project***  As I've already put a lengthy, fully updated schedule in my document covering all the elements of the project to completion, would it be more appropriate to have a smaller plan here, maybe in the form of a table or bullet points that cover the phases of the project from now until the next TMA hand in date?  Thanks,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 14 April 2023 at 09:20 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Just another query.  In relation to the **'Review and reflection**' section, should I remove all the writing I added in TMA01 and fill that space following only the TMA02 specific guidelines? or should I adjust the TMA01 writing slightly instead?  Regards,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 14 April 2023 at 13:40 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Hi Evan, rather than replace previous GANNT charts, include them all showing their evolution. Each change should correspond to a 'decision point' in your reflective journal.  An updated GANNT chart covering the project through to completion will meet the requirements of a future plan and you can make a referal to the appendices where you have placed it.  Regards,  Darren  TM470 Tutor  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 13 April 2023 13:07  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 14 April 2023 at 13:43 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Evan, no don't replace everything submitted for TMA01. Make any required adjustments in line with new work or TMA feedback and then develop each section from the previous TMA. It is important that there should be evidence of development in each TMA.  The project should be viewed as an evolving, growing document which culminates in the production of the final EMA.  Hope this helps.  Regards,  Darren  TM470 Tutor  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 14 April 2023 09:20  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 14 April 2023 at 15:05 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Great,  I’ll take all that on board.  Thanks,  Evan  [Quoted text hidden] | | |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 3 May 2023 at 20:58 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Hi Evan, I just wanted to keep you in the picture with regard to the marking of your TMA02 assignment.  Tutors are requested to try and mark all assignments within 10 working days of hand in.  I do try and get work turned around in the quickest time possible, to give students the maximum time I can, to review the feedback ahead of the next assignment.  Owing to a number of factors, including having to take on a student whose previous tutor has fallen ill, I have not turned your assignment around as quickly as I would normally like to.  To keep within the 10 day timescale, I should have your assignment marked and returned to you by NLT 09 May 2023.  I will be marking your assignment over the weekend and am confident of meeting the marking 'deadline'.  I just wanted to explain the deadline and assure you that marking has not been overlooked.  Thank you for your understanding.  Regards,  Darren  TM470 Tutor  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 14 April 2023 15:05  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 3 May 2023 at 21:04 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Thanks for keeping me informed.  I just want to say on my end there is no pressure to get it returned asap. Whether it finds its way to me next week or the week after it doesn’t matter.  All the best,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 4 May 2023 at 17:44 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Evan, thanks for your understanding.  Regards,  Darren  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 03 May 2023 21:04  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 14 May 2023 at 13:53 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Just thought I'd give you an update on where I'm at.  At the start of the month I started putting in some revision for two other modules as well as some work for an EMA for another module. After assessing things I realised my time would be better spent focusing entirely on these modules, as such I intend to do no work to progress the TM470 project until about the 7th of June. I will reflect this change of project schedule in the Gantt chart as well as the document itself.  I intend to review the project at least once a week just to stay familiar enough with it that I can pick it back up in June easily.  Come June the 7th I will still have about a month to fully prioritise the project and TMA03, which has a due date of  the 4th July, so I'm not concerned with it affecting the work I was expecting to have completed by this point.  regards,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 14 May 2023 at 15:25 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Evan, I fully understand the need to prioritise when you are studying a number of concurrent modules.  It is good that you are going to reflect the absence of progression in your schedule, as this shows an honest appraisal of the constraints on your time. It would be useful to include this in your reflection.  Regards,  Darren  TM470 Tutor  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 14 May 2023 13:53  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 15 June 2023 at 11:03 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Just thought I'd get in touch and let you know I've returned to focusing on the project full time now.  I've started developing the Java and xml files required and I've also adjusted the TMA slightly in respect to what is required for TMA03.  I envision I should have a basic implementation of the 'post creation' and 'post display' activities (as was my initial aim) by the end of next week. I should then have time to finalise TMA03 and clear up any issues in the week preceding the hand in date.  Regards,  Evan.  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 15 June 2023 at 18:33 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Evan, thanks for keeping me updated and remember you can get an extension on TMA03, if you need it.  Regards,  Darren  **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 15 June 2023 11:03  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 27 June 2023 at 09:41 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  I've managed to complete the increment for the App as planned and now I'm spending time on TMA03.  My intentions over the next few days are to iterate over the TMA based on your TMA02 feedback, build out the work done section to reflect this increment and also finish the 'Review' section.  I just wanted to query how you think I should structure the 'Review' section? At the moment I've got about 1800 words from previous TMA's covering things like 'Review and reflection', 'ways of working' and 'Legal, social ethical and professional issues'. Some of the sections I've just mentioned are similar to what is being asked in the 'review' section of this TMA, so would it be appropriate to chop, adjust and build those previous sections under the new headings of this section? or should I leave them as is and continue below them?  Regards,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 29 June 2023 at 12:39 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Any chance of getting a week long extension for TMA03? I feel like I could create a better, more polished report with a little extra time.  Regards,  Evan  [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Darren.Bolton**<darren.bolton@open.ac.uk> | 29 June 2023 at 15:11 |
| To: Evan Neale <e22neale22@gmail.com> | |
| |  | | --- | | Hi Evan, I would recommend adjusting, in line with TMA02 feedback, and then developing these sections.  I have applied a week's extension for you for TMA03 on the system.  TMA03 now due 11/7/2023 by 12:00.  Regards,  Darren  TM470 Tutor    **From:** Evan Neale <[e22neale22@gmail.com](mailto:e22neale22@gmail.com)> **Sent:** 27 June 2023 09:41  [Quoted text hidden]    [Quoted text hidden] | | |

|  |  |
| --- | --- |
| **Evan Neale**<e22neale22@gmail.com> | 29 June 2023 at 15:37 |
| To: "Darren.Bolton" <darren.bolton@open.ac.uk> | |
| |  | | --- | | Hi Darren,  Yeah I thought that might be the case but just thought I’d enquire.  That’s ideal, thank you.  Regards,  Evan  [Quoted text hidden] | | |

## Appendix 14 – Project Logs

\*As of 08/07/2023\*

**Log sheet no.** 1

**TMA 01 date:** 28th February

**Time spent:** 2 hours

**Work -** Read through the TMA information as well a the learning outcomes to clarify how to fulfil the requirements and attain the highest marks. I started to complete the initial preparation and planning sections of the TMA . I reiterated over it a few times to further clarify my reasoning for the solution as well as to get a clearer image of the solution itself. I also started looking at what software would be best to plan the project, as I intended to use a Gantt chart.

**Problems –** No major problems at this stage, although it took more time than I’d liked to clarify the solution. Choosing the software for the Gannt chart took more time than I would have liked as it took a little trial and error of using the application to make sure it would be adequate.

**Comments –** Feels good to have actually gotten the module underway and to have started creating a picture of what needs to be done. My priority needs to be breaking down the project into a tasks so that I have a clearer view of the steps I need to take to create the project.

**Next work planned –** Continue working through the Planning and preparation section of the TMA as well as prioritising the development of the project plan in the form of the Gantt chart.

**Log sheet no.** 2

**TMA 01 date:** 28th February

**Time spent:** 3

**Work -** Continued through the process of setting out the skeleton of the TMA document and working through it. I also reflected on the practical elements of the project and used a brainstorm structure to set out the key elements of the project. I spent time deliberating over and reiterating this section until I was relatively happy and started to develop a Gantt chart that combined these elements into a time orientated plan.

**Problems –** No major problems to mention, took a little longer than expected to iterate over and identify the key elements of the project

**Comments –** Feels much better having started to produce and fill out the Project plan as I can now see a route to the end of the project whereas previously it wasn’t as clear.

**Next work planned –** Continue working through the TMA and actually created the initial Project files in Android Studio

**Log sheet no.** 3

**TMA 01 date:** 28th February

**Time spent:** 4

**Work -** Continued working through the first section of he TMA. Reflected on the different project lifecycles and decided on the one that best for my scenario. Laid out the

**Problems –** No major problems at this stage. Although it did take some understanding and further thought on which lifecycle model would this project.

**Comments –** I have noticed that my ability to carry out the tasks related to the project is already taking much longer than first anticipated. I believe that because it is my own personal project I send more time deliberating on things than I should.

**Next work planned –** Work through the second section of the TMA. Research the literature I believe I will need to help me carry out the project and also create the body of the project in android studio

**Log sheet no.** 4

**TMA 01 date:** 28th February

**Time spent: 5**

**Work -** Made a draft of the section 3.3 – review and reflection and made some notes on how to progress it. Further iterated over the description and scope (section 3.1) to finalise these parts and make sure they were clear for the reader. Built out the tasks and sub tasks in section 3.1.

**Problems –** No major problems at this point just gradually working through the elements, no issues appeared.

**Comments –** Went smoothly today nothing of note to speak of

**Next work planned –** Continue on and reiterate over the task, sub task and their conversion to a Gantt chart. Then continue through 3.1

**Log sheet no.** 5

**TMA 01 date:** 28th February

**Time spent: 5**

**Work -** Finished most of the tasks and sub tasks. Adjusted the Gantt chart to match. Reiterated over the lifecycle and fleshed out the risks section

**Problems –** No major problems at this point just gradually working through the elements, no issues appeared.

**Comments –** Everything will still need review before finalising and handing in

**Next work planned –** Concentrate on the work done section and focus on the diagram you’d like to use

**Log sheet no. 6**

**TMA 02 date:** 25th April

**Time spent:  5**

**Work -**  I reviewed the marked TMA01 and copied the feedback from the marked document into a fresh TMA02 document to act a reminder of what to change. I then reviewed the more thorough feedback form and made a more thorough list of TODO actions to develop the TMA further. I then reviewed all the resources I’ve accumulated and decided upon using and gathered them together for easy access. I then Thoroughly reviewed authentication options and the final design of registration and login pages.

**Problems -** No immediate problems however as I was reading the feedback from the previous TMA I realised that I hadn’t implemented time into the project schedule for making these changes, Initially I will just front load these tasks into the time I’ve allotted for TMA02, I will review the time scale after another 2 sessions to see if an issue has been caused.

**Comments -** Pretty easy session as it was just reviewing feedback and resources. This helped me to build a short term plan for how to move forward.

**Next work planned -** On Thursday 16th April, using the finalised design and guides/ references which were finalised in this session, I will begin to build out the initial registration page and login page. Concentrating on the body of it initially and implementing the authentication after.

**Log sheet no. 7**

**TMA 01 date: 25th April**

**Time spent:  5 hours**

**Work -**  Created the login and registration XML files and java classes. Set out the basic skeleton of both the XML files, the registration one needs some work still though.

**Problems -** I realised that I spent too much time trying to beautify the application, to no avail. At this stage in my project I can’t be too focused on this aspect otherwise it will affect my schedule and eventually the overall outcome of the project.

**Comments -** I have realised at the stage of creation I will focus on expending very little time iterating over how the application looks. My main priority will be making sure that the application is functional.

**Next work planned -** Tomorrow. Finish up the registration page and connect up the authentication feature of firebase API.

**Log sheet no.** 8

**TMA 01 date: 25th April**

**Time spent:  4**

**Work -**  Finalised the layout of the registration XML file. Then reviewed my resources and fleshed out the java required to access users input, once register has been clicked, and register a new user . All in RegistrationActivity.Java

**Problems -** No major issues occurred yet however I have focussed getting this element functioning and not focusing on some of the finer details which may hold me back. This has ended up with me creating a couple of TODO notations within the XML and Java files which I will need to return to.

**Comments - A**s I haven't technically created anything functional yet I haven't been able to test anything, I have a slight concern that there may be an issue I’ve overlooked which is why I’m focussing on getting it built and not deliberating over the smaller issues yet.

**Next work planned** - I will have a go at doing some of the notes that were made by the tutor on the TMA either tomorrow of Sunday

**Log sheet no.** 9

**TMA02 date: 25th April**

**Time spent:  5**

**Work -  I implemented the java code needed for the login process. This included making the onclick listeners for the login and register buttons and the actions that clicking them entailed. This also involved extremely basic form of error checking.**

**Problems - Had a few issues regarding the testing of the login process. It took a little time finding the mistakes I’d made.**

**Comments - The whole app still looks a little bland. Will spend some time beautifying it by changing the constraints and the theme/style at a later date**

**Next work planned - Tomorrow. Will focus on registration validation. May possibly add an email verification API or at the very least add it as a TODO for later iterations. Will look into user profile creation and storage on firebase.**

**Log sheet no.** 10

**TMA02 date: 25th April**

**Time spent:  4**

**Work -  I reviewed the validation code I used and decided, for better or worse, to create a separate class for them, which I then made use of in the login and registration classes. I further developed the registration java file so the registration would create a input to a collection of the firebase store whereby it can be tied to a profile for usage in other activities of the project.**

**Problems - Labouring over the most appropriate validation methods took time, will probably require more formulaic thought about what’s appropriate later. Also the creation of the collection in the database to a lot of trial and error to get right.**

**Comments - No major comments**

**Next work planned - Tomorrow, I will focus on the TMA and tidy up regarding the notes given in TMA01**

**Log sheet no.** 11

**Session no.** n

**TMA02 date: 25th April**

**Time spent:  5**

**Work -  I reviewed some of my code removing elements that weren’t needed, improved some of the TODO comments that I have added to be done at a later iteration and fiddled with some of the views constraints to make it look nicer. I then spent some time reviewing and working through the TMA02 trying to implement some of the notes made from TMA01 as well as adding further to-do notes within the assignment.**

**Problems - None**

**Comments - None**

**Next work planned - tomorrow - I would like to finalise all the comments in my code for clearer understanding and then evidence them into my ‘work done’ section of the TMA02**

**Log sheet no.** 12

**TMA02 date: 25th April**

**Time spent:  2**

**Work -  Restructured the code making sure that all comments were of better quality and more informative. I also reviewed some of the TODO elements which I’ve added for later iterations. Also started building out the ‘work done’ part of the TMA02 ready for filling in next session**

**Problems - No problems here in regards to the work, some other stuff came up so I couldn’t give a full day to carrying out the work**

**Comments -** n

**Next work planned - Tomorrow. I intend to try complete the work done section of the TMA, adding in all code an referencing any sources i borrowed code from (e.g. auth stuff)**

**Log sheet no.** 13

**TMA02 date: 25th April**

**Time spent:  3**

**Work -  Remade the project log in SmartSheet’s web app. Made a short list of changes I’ve made in the document which needs to be explained and expanded on.**

**Problems - Took a bit longer to trial and error the best Gantt chart application**

**Comments - Still needs finalising of the project schedule**

**Next work planned - 06/04/23 - continue working through TMA**

**Log sheet no.** 14

**TMA02 date: 25th April**

**Time spent:  4**

**Work -  Fiddled with schedule and fiddled with other bits related to the schedule as well as the work done section, specifically the information resources.**

**Problems - None**

**Comments - None today**

**Next work planned -** n

**Log sheet no.** 15

**TMA02 date: 25th April**

**Time spent:  5**

**Work -  Tightening up the entire TMA02 document, rewording sections, filling out other sections, adjusting citations and references. making notes on what else needs to be done**

**Problems - Sent a query to Darren about schedule section in the future plan ,whether a smaller limited reference to the schedule was fine**

**Comments - None**

**Next work planned - 14/04/2023 - continue TMA**

**Log sheet no.** 16

**TMA02 date: 25th April**

**Time spent:  5**

**Work -  Tightened up TMA02 centring around the future plan and review and reflection sections.**

**Problems - Found myself jumping around the sections which ultimately led to a slightly disjointed day of work.**

**Comments - A lot has been done but still not happy. Need to add project logs and communication logs and tighten them up**

**Next work planned - 15/04 - probably start reviewing design of the next increment and maybe create a basic class diagram**

**Log sheet no.** 17

**TMA02 date: 25th April**

**Time spent:  6**

**Work -  Finalised multiple sections across the whole of the TMA document. Added the communications log and project log into the appendix and referenced them in the appropriately. Reviewed the wireframes of the pages/view I intend to create in the next increment of the application. Iterated and improved them as well as thought through an noted down the the classes and files that I will need to create to implement them.**

**Problems - No problems, only solutions.**

**Comments - Still need to finalise the review section , more specifically the evaluation of project management and the covering of the Legal, social, ethical and professional issues and equality, diversity and inclusion concerns**

**Next work planned - 21/04/2023 - Finalise the entire TMA before having a day off and returning on another day for a final review**

**Log sheet no. 18**

**TMA 02 date: 25th April**

**Time spent:  4**

**Work -  Finished the review section specifically the section regarding the LSEI and DEI. Returned to the schedule section and implemented references to project logs regarding schedule changes. Finalised the short term task list of tasks needed to be completed to finish the TMA.**

**Problems - None**

**Comments - Not much left but want to stop and return when mind is refresh and can cast another eye at the work**

**Next work planned - 23/04/2023**

**Log sheet no.** 19

**Session no.** n

**TMA 03 date:** 4th July

**Time spent:  4**

**Work -**  Reviewed the entire project and most recent TMA as well as the information for the next TMA. Assessed the project schedule and made changes to it based on my updated timeframe

**Problems -** No problems at this point

**Comments -** Getting back into the rhythm of project work.

**Next work planned** - 13/06 - Anther review of the project files and begin building out the application

**Log sheet no.** 20

**TMA 03 date: 4th July**

**Time spent:  6**

**Work -  Started to work through the classes that i required to implement the activities. Began making the post class and the post creation activity.**

**Problems - As I was working I realised I hadn’t updated anything in a while and decided to do so however it added some kotlin sections into the project … i tried to remove this and accidentally caused an issue that stopped android studio from being able to open. Even when uninstalled and reinstalled it wouldn’t open. Through lengthy trial and error and trying out different versions i managed to fully overcome the issues but it sucked up a lot of time**

**Comments - Wasted a lot of the day trying to fix android studio**

**Next work planned - 14/ 06 - continuing to working on the classes**

**Log sheet no.** 21

**TMA 03 date: 4th July**

**Time spent:  4**

**Work -  Continued to flesh out the post creation java file and created the main thread activity, changed the intents in the login activity so it will take the user to the main thread after logging in**

**Problems - None today**

**Comments - Made notes to review the validation of the post creation at a later date, also to review the post creation details after fully creatin the main thread activity**

**Next work planned - 15/06/2023 - Flesh out the Main thread activity**

**Log sheet no.** 22

**Session no.** n

**TMA 03 date:** 4th July

**Time spent:  4 hours**

**Work -  Filled out post details XML and iterated over post creation Java.**

**Problems - No major problems however it does seem a little more challenging to work through this stage**

**Comments -** n

**Next work planned - 20/06//2023**

**Log sheet no.** 23

**TMA 03 date:** 4th July

**Time spent:  4**

**Work -  Iterated over code cleared any warnings and issues. Reviewed conceptually and then tested the app as it was**

**Problems - View of the posts takes up entire page. URL’s causing the app to close once clicked**

**Comments - Honestly went better than expected, I was not confident in the app functioning so well at this point.**

**Next work planned - 22/06/2023**

**Log sheet no.** 24

**TMA 03 date:** 4th July

**Time spent:  5**

**Work -  Iterated over post design, made it fit to content rather than fit to screen as it was originally. Changed some of the text in post creation. Moved username to the top of the post (will just say username for now as i haven’t made the user profile edit section yet). Sorted out the issue which was stopping the URL form being clickable**

**Problems - None yet**

**Comments -** n

**Next work planned - 23/06/2023 - Hopefully aim to just smarten up the code, streamline the comments and finalise my TODO’s for the next increment. This should leave me a in a good space for finalising TMA03 next week.**

**Log sheet no.** 25

**TMA 03 date:** 4th July

**Time spent:  2**

**Work -  Just started to review the ‘review’ section of the assignment. Created a template and started writing out some bits.**

**Problems - Wasn’t sure how to approach the removal or integration of the old sections and the review element od this TMA. Have enquired with the tutor but ploughing on for now and will adjust based on tutors opinion**

**Comments - Only a short one today.**

**Next work planned - 28/06/2023**

**Log sheet no.** 26

**TMA 03 date:** 4th July

**Time spent:  6**

**Work -  Fleshed out the entire first draft of the review section.**

**Problems - None as of yet**

**Comments - Draft will need review, trim and rewording then will have to see how to deal with it in regards to the old section**

**Next work planned - 29/06/2023 - Update the work done/ future plan section**

**Log sheet no.** 27

**TMA 03 date:** 11th July

**Time spent:  3**

**Work -  Created and began to flesh out the work done in regards to the second increment of work. Very raw and will need review and rewording**

**Problems - None, just thinking about he best way to word and present the work I’ve completed was a small hurdle however I ended up largely following the frame of the precious increment**

**Comments -** n

**Next work planned - 30/06/2023**

**Log sheet no.** 28

**TMA 03 date:** 4th July

**Time spent:  4**

**Work -  Filled out all the work done section and pasted the created code into the appendix.**

**Problems -** n

**Comments - Have made a note that I’d like to review this a final time to work out any issues with a fresh view of it before being happy**

**Next work planned - 04/07/2023 - finalise the review section and amalgamate it into the older review section**

**Log sheet no.** 29

**TMA 03 date:** 11th July

**Time spent:  5**

**Work -  Reviewed and fleshed out the review section, this included fixing all grammatical error, rewording bits and then amalgamating with the elements form previous TMA’s**

**Problems -** n

**Comments -** n

**Next work planned -** n

**Log sheet no.** 30

**TMA 03 date:** 11th July

**Time spent:  5**

**Work -  Worked through from the top iterating over sections, rewording sections and tightening the whole document up. Have made a thorough to do list at the top of the document that must be worked through to finalise the assignment**

**Problems -** n

**Comments -**

**Next work planned - continue working through the to do elements from the top downwards**

**Log sheet no.** 31

**TMA 03 date:** 11th July

**Time spent:  5**

**Work -  Worked through and finalised the end of he work done element. Worked through the review element and merged bits together as well as reworded and trimmed bits. Review whole document and made a final to do list of what needs to be completed**

**Problems -** n

**Comments - Essentially just need to add a few more of the resources/literature sources that I’ve sued that are relevant going forward before checking supporting docs, updating communication an project logs and reviewing whole doc**

**Next work planned -** n

**Log sheet no.** 32

**TMA 03 date:** 11th July

**Time spent:  3**

**Work -  Reviewed and titivated different elements of the document. Also include the update literature resources**

**Problems -** n

**Comments -** n

**Next work planned -** n

**Log sheet no.** 33

**TMA 03 date:** 11th July

**Time spent:  4**

**Work -  Added in all the project logs and comm logs. Finalised the document.**

**Problems -** n

**Comments - Have finalised the document however will have a break so I can come back to it with a fresh mind on Monday to review it one final time before handing in.**

**Next work planned –** Monday 10th July – Final Review