|  |  |
| --- | --- |
| 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

[Preparation and planning 3](#_Toc131769736)

[Background 3](#_Toc131769737)

[Problem Description 4](#_Toc131769738)

[My Proposed Solution 4](#_Toc131769739)

[Alternatives 5](#_Toc131769740)

[Major Tasks and Subtasks 6](#_Toc131769741)

[Life cycle model and schedule 8](#_Toc131769742)

[Schedule 10](#_Toc131769743)

[Resources, skills and methods 14](#_Toc131769744)

[Future Plan 16](#_Toc131769745)

[Project Work completed 17](#_Toc131769746)

[Information sources – the literature 17](#_Toc131769747)

[Project work 17](#_Toc131769748)

[JAVA files 22](#_Toc131769749)

[XML Layout files 23](#_Toc131769750)

[Other files 24](#_Toc131769751)

[Review and reflection 24](#_Toc131769752)

[References 26](#_Toc131769753)

[Appendix 27](#_Toc131769754)

[Appendix 1 – Schedule 27](#_Toc131769755)

[Appendix 1.1 – Application Preparation and TMA01 28](#_Toc131769756)

[Appendix 1.2 - First increment and TMA02 29](#_Toc131769757)

[Appendix 1.3 - Second increment and TMA03 30](#_Toc131769758)

[Appendix 1.4 - Final increment, Iteration and EMA 30](#_Toc131769759)

[Appendix 2 - Wireframes 31](#_Toc131769760)

[Appendix 3 – MoSCoW 39](#_Toc131769761)

[Appendix 4 – Java Code 40](#_Toc131769762)

[Appendix 4.1 – LoginActivity 40](#_Toc131769763)

[Appendix 4.2 – RegistrationActivity 42](#_Toc131769764)

[Appendix 4.3 – UserProfile 46](#_Toc131769765)

[Appendix 4.4 – UserValidation 46](#_Toc131769766)

[Appendix 4.5 - MainActivity 47](#_Toc131769767)

[Appendix 5 – XML Layout 48](#_Toc131769768)

[Appendix 5.1 – activity\_login 48](#_Toc131769769)

[Appendix 5.2 – activity\_registration 50](#_Toc131769770)

[Appendix 5.3 – activity\_main 52](#_Toc131769771)

[Appendix 6 – Other files 53](#_Toc131769772)

**>Review contents page to make sure it’s correct/good enough**

**>FYI skim marked example one more time and see if you missed anything**

**>Make sure to updaet the task and subtasks below to match the updated schedule sections**

**>do you need the boys to review? Maybe get their thoughts on it? Or at least mention somewhere that you’ll get them involved in the next iteration when you build out the main body**

**> add a copy of all the java code to the additional files? Or add a GitHub hyperlink somewhere?**

**Some of tutors comments:**

**It is worth noting that every reference list item, in your alphabetically ordered reference list, should correspond to one or more citations in the main body of the project report**. (If you read or review sources that you dont cite, then include those in a separate 'bibliography' listing). As a rough guideline, for every 1000 words of academic work there should be around 8 to 12 citations.

**Three areas for develoment:  
  
A project log showing weekly progress and decsion making linking to GANNT chart iterations  
A prescriptive resources / skills & activities list  
A 'RAG' rated Risk Register with a key showing numeric values and mitigating activities**

**\*\*Review feedback form and use that to help make changes\*\***

* Perhaps consider including the title description included here as part of a ‘summary’ on the title page (telling the reader what is contained within the report) with a more succinct title for the project ?
* Another thing to be aware of is I think code needs to be referenced if you’ve borrowed some e.g. for firebase stuff for authentication (which has to be exact). Also in general link resources that helped in the code creation 😊 maybe even ask turtor about this

# 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 worlds 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 its 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. This makes it less efficient to for some busier individuals. I will note a few of these applications below my proposed solution and explain why they aren’t ideal alternatives.

## My Proposed Solution

At it’s most basic principle my solution will be to create a relatively simple, safe, invite only java based android application that allows for the dissemination of news, links and rumours within a more trustworthy and customisable application. Initially, to tame it’s 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 alternative to 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.

Please reword below into something more professional and probably from a ‘before’ POV || Also reread exactly what they’re asking for here

I believe that developing this application perfectly aligns with the Software focus of the my degree route. One of my year two modules was ‘Object-Orientated Java programming (M250)’, my enjoyment of this module absolutely played a role in my decision to select a project with a large java element to it. This project builds on my java and object orientated programming skills, it requires me to build on my foundational skillset, researching and understanding progressively more complicated coding and object implementation.

In year three I’m 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 useful to put the skillset and knowledge I’ve acquired into practice to further solidify this knowledge. Also in year three I’m covering the module ‘Web, mobile and cloud technologies (TM352)’, during this module, amongst other things, we covered application development which hac been useful

In every single module alongside increasing my skill and knowledge base I’ve also been developing my soft skills which will be useful and also further developed within this project.

I have also been developing my skillset outside of the Open University modules, and have been experimenting with development within android studio, this project will further expand and solidify my learning in this respect.

## 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. Iterate on the post creation section to include a comment section for each post
   2. Build first iteration of the separate search page
      1. Review documentation and official guides
      2. Review final search page design
      3. Build frame of search page
      4. Embed search functionality into search element
   3. Add a simplified search element at top of main page
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 undertake 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** model. I’m relatively new to an undertaking such as this project, by 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.

## Schedule



There will likely be a number of iterations of your GANNT chart – **each linking to an entry in your project log** where you make an alteration to your plan due to the inevitable factors that crop up over the life of a project. These iterations will show evidence of decision making, review and corrective action – essential components of project management.

**Try if possible to include the GANNT charts (perhaps cropped to allow a chart to be presented in a number of ‘parts’) in the appendices as the final EMA must be a ‘self contained’ report.**

**I think I should put the extended (cropped) schedule in the appnedixe. Maybe have a title (with link to appendix ) then a lil written bit below the title?**

**Rewords and paragraph nicer**

As the project has progressed I’ve encountered no major set backs however there are small issues and considerations that have led me to adjust the project schedule. Due to some life events I’ve pushed starting the second increment of the application onwards by a week. I have also extended the amount of time that I will prescribe to work on this increment, I believe this will be needed as while I’ve been prepping myself in anticipation for this section of the application I have come to the realisation that this could be a little more complex and time consuming that first anticipated and mix this with the fact that I will be revising heavily for exams in my other modules.

I have provided a cropped screenshot of the entire task list and then spoke a little bit on each phase of the project. I have also provided a series of cropped images of the entire Gantt chart, split into each phase, in the appendix which can be accessed by clicking on the hyperlinked titles below. For clarity more readable full project 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**

Figure

Table

Description automatically generated with medium confidence

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

**[Application Preparation and TMA01](#_Appendix_1.1_–)**

**[First increment and TMA02](#_Appendix_1.2_-)**

**[Second increment and TMA03](#_Appendix_1.3_-)**

[**Final increment, Iteration and EMA**](#_Appendix_1.4_-)

A definitive list of resources skills and activities needed for successful project completion was required here Evan. (LO3

From evan: Go over everything you’ve done so far and write a pretty thorough list of resources skills and activities? Needed – **pronbs doneish here?**

## Resources, skills and methods

I will primarily be using the Java and XML to develop the application within the Android Studio Integrate 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 phone 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 using GitHub as a repository for the project which will allow for version control and also be a useful element to supply 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 decided on these options as they both were freely available and designed to integrate within 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.

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

I will be making use of a small number of potential users, who I will interrogate at specific points throughout production to help further develop the applications requirements.

Maybe have these below have in text citations and link to references in the reference section further below

A list of the respective links to the relevant documentation and resources are as follows:

* Android Application developer guides - <https://developer.android.com/docs>
* Android studio guide - <https://developer.android.com/studio/intro?utm_source=android-studio>
* Material 3 documentation - <https://m3.material.io/develop/android/mdc-android>
* GitHub documentation - <https://docs.github.com/en>
* Firebase developer documentation - <https://firebase.google.com/docs?hl=en&authuser=0>
* Unsplash, for free images - <https://unsplash.com/explore>
* Midjourney - <https://www.midjourney.com/home/?callbackUrl=%2Fapp%2F>
* Smartsheet’s Gantt charting - <https://www.smartsheet.com/welcome-customers-home>

**Probs give below a once over – maybe a lil convoluted could simplify?**

**Risk Matrix – turn this to a RAG rated risk register (google) *this needs sorting out***

By feeding the risks into a risk matrix, where there have representative numeric values we can calculate a more specific risk profile and resultingly the overall risk.

**Key**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Likelihood:** | Rare | Unlikely | Moderate | Likely | Almost certain |
| **Impact:** | Insignificant | Minor | Significant | Major | Severe |
| Numeric equivalent | 1 | 2 | 3 | 4 | 5 |

**Total risk**

Low = 1 - 3 Medium = 4 - 7 High = 8 – 10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Ref | Risk | Likelihood | Impact | Overall Risk |
| Ref.1 | Potential users become uncommunicative | 2 | 1 | 3 = Low |
| Ref.2 | Failure of computing equipment | 1 | 5 | 6 = Medium |
| Ref.3 | Miscalculation of planning schedule | 3 | 3 | 6 = Medium |
| Ref.4 | Misjudgement of my technical ability and my capacity to follow through in fully developing the project | 2 | 5 | 7 = Medium |
| Ref.5 | Scope creep affects the schedule I have created | 3 | 2 | 5 = Medium |
| Ref.6 | Other modules take up more time and affect project schedule | 4 | 4 | 8 = High |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | ***Impact*** | | | | |
|  |  | Insignificant | Minor | Significant | Major | Severe |
| ***Likelihood*** | Almost certain | Medium | High | High | High | High |
| Likely | Medium | 4 | 6 |  |  |
| Moderate | 3 | 6 |  |  |  |
| Unlikely | 4 | 8 |  |  |  |
| Rare | 5 |  |  |  |  |

**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 project schedule.

## Future Plan

Mention intentions of sched change a lil, focus on the increments and iterate over all elements in the time after they’ve all been completed

From the point I’m currently at in the project I believe I have a clear picture of what I need to do and in what schedule I ned to carry it out. Initially upon the completion of TMA02

**I think you should finalise comments in the code before going HAM on theis section**

**with below I think you just need to build onwards from the work already referenced from before**

# Project Work completed

## Information sources – the literature

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 have also initially skimmed and I’m currently working through in more depth ‘**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.

ADD more here – maybe find some more official literature (ie not documentation) relevant to the sections you will be adding in the next increment

## Project work

MAKE SURE THE FIG REFRENCES ARE CORRECT – **MUST ONLY ADD CODE YOU HAVE MADE….ANY AUTOGENERATED CODE WILL BE FLEAGGED BY TURNITIN\_maybe delete the old project files an github files pics ?**

The work completed I the build up towards TMA01 was almost entirely preparatory, setting the foundations for the second phase of work, were I start to create some of the elements of the application. 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 (file structure/appendex citation needed here), below I have provided the evidence of the updated file structures:

Text

Description automatically generated

Figure - updated Android Studio file structure

At this stage of the project the majority of the work I have done has been exploratory and preparatory, as such the evidence I provide in this section will reflect this. In regards to the actual body of the project work I have done little more than create the initial skeleton of the project files, ***figure 2***, and created a repository on GitHub, ***figure 3***.

I have spent thinking through and iterating over the initial wireframes which I’ve provided in the appendix as ***figures 4 to 11*** inclusive, I placed them there as there are multiple and it seemed more appropriate. I have also provided these wireframes in a PDF in the ‘supplementary documents’ file alongside this TMA. The wireframes act a visual representation of my design intentions and include rudimentary layouts of all of the pages that the application shall have.

ALL BITS OFWORK FOR THE FIRST INCREMENT I WILL ADD BELOW – DELETE THESE NOTES AFTER COMPLETION

In the prior phase of m work I had initially just been mainly preparatory, setting the foundations for me to undergo the practical tasks needed the build the application. The evidence I provided for theose

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. Conceptually the app 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.

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 2** above I will provide a link so they can be accessed at <https://github.com/EvanON-test/TM470_Project> (CURRENTLY PRIVATE MAKE PUBLIC WHEN TMA02 IS HANDED IN).The updated android studio file layout looks like the following:

REMEMEBR ANY CODE YOU ADD HAS TO BE NON AUTO GENERATED OTHERWISE IT FLAGS TURNITIN

From here break down all the changes that you’ve made class by class or file by file – must include all the smaller changes in the manifest/styles files ect . ADD CODE SNIPPETS IN THE APPENDIX || ALSO MAYBE SITE ANY CODE USED HERE AND PUT THE EXTENDED REFERNCES IN THE REFERENCE SECTION || TURN THE VARIABLE NAMES INTO THERE CODE EQUIVELANTS IN THE EXPLANATION BELOW? Then make em bold? || Also reread to make sure it makes sense to someone who’s not totally familiar with the application || Also can you make titles link to appendix elememt?

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.

### JAVA files

**[LoginActivity](#_Appendix_3.1_–)**

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 loginBtn() and the registerBtn(). The login 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 loginBtn is pressed the inputs the user made into the edit text views are initialised to variables. The variables are then validated against some 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 login ciatation) . It will interrogate the Firebase instance to see if the user credentials are present, if they are there will be a Toast message I the affirmative, it will get the users credentials and initialise them to a key value pair for access 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_3.2_–)**

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 the registerBtn(). The this calls the registerNewUser() Method.

Once the registerBtn is pressed the inputs the user made into the edit text views are initialised to variables. The variables are then validated against some 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 reg ciatation) . 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_3.3_–)

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_3.4_–)

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_3.5_-)

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 implantation of a user profile element within the application.

### XML Layout files

[**activity\_login**](#_Appendix_4.1_–)

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_4.2_–)

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_4.3_–)**

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

This is where you can mention the smaller files and their changes like android manifest, srings ect

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 t eh 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.

FYI you will be staring the secod increment while doing this TMA so just bear that in mind

# Review and reflection

As of this moment the majority of the work I have completed for the project has been preparatory, I have concentrated on gathering resources, creating a clear foundation for the project and iterating over the application in my mind and on paper to get a clearer image of what I hope to create and achieve.

I believe my exploration of the documentation and resources required to build on my knowledge and guide me through aspects of development within the IDE that I am unfamiliar with has proven beneficial. I have resourced the required documentation and found interesting books through which I can further develop my knowledge. I also believe my initial creation of the wireframes has helped me obtain a clearer picture of how the application will look and this has helped me to reflect and iterate over it to create a better, albeit not perfect, representation of my intentions.

I believe the less structured layout of this module and the less clearly defined elements to the final project and assignment expectations slowed me down initially as it took me time to gather my thoughts as well as start piecing all the elements of the project together. This also lead to me elaborating and adjusting the schedule regularly. I believe through the process of this TMA I have mostly recovered from this issue and this should hopefully prove evident in the next TMA and future presentations of my actual project work.

Throughout the creation of this TMA it has made me realise I haven’t done enough in the form of eliciting requirements and creating a more useful conceptual model of the application which would prove more useful during development.

**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 Open University Modules, 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 which I would like some feedback on would be around the scale and complexity of the project. I feel that the project is substantial enough and complex enough to be a good foundation for this module but I would appreciate any input. I’m aware that there are other elements that I could add to the application but I’m not sure on where the line is 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 2.

**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.

**Answer the following:**

Identify those elements of your plan that have gone well and (more importantly) those that have proved more problematic. Adjust your plan as necessary.

fff

Provide a brief account of the legal, social, ethical and professional issues raised by your project and describe any adjustments to the project made in light of them. You should consider these as if your project were to be fully implemented and/or your report presented to its target audience.

fff

Indicate how you have responded to your tutor feedback on TMA 01. Briefly indicate to your tutor any specific issues on which you would like feedback.

dfdfdf

Review progress of project work outlining what you have done, identifying any obstacles or problems that have emerged and explaining either how you addressed these or propose to address these in the near future.

sdfsdf

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

dsfsdf

# References

**I THINK YOU NEED TO ADD CITATIONS TO THE ACTUAL DOC SO THAT IT LINKS TO THE SPECIFIC REFERENCE HERE 😊 || also try using the in built citation thing under the references option in word 😊**

Goulet, Merlene (2015) *white ceramic mug on top of white ceramic saucer surrounded with coffee beans*. Available at: <https://unsplash.com/photos/ISFopTz7sBo> . Accessed(25/03/2023)\*\*double check this is a good ref for photos

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

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

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

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: Feb 2023)

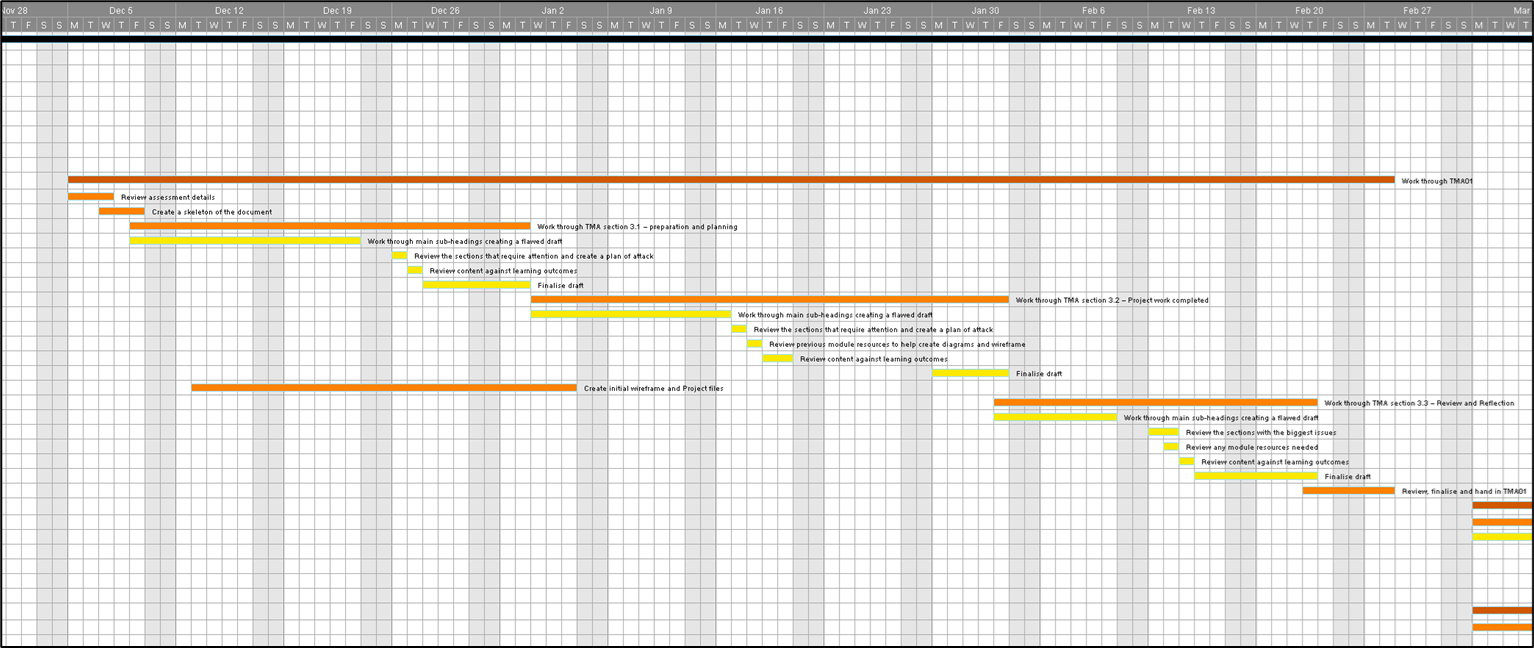
Firebase authentication (login) - <https://firebase.google.com/docs/auth/android/password-auth>

Firebase auth (register) - <https://firebase.google.com/docs/auth/android/start>

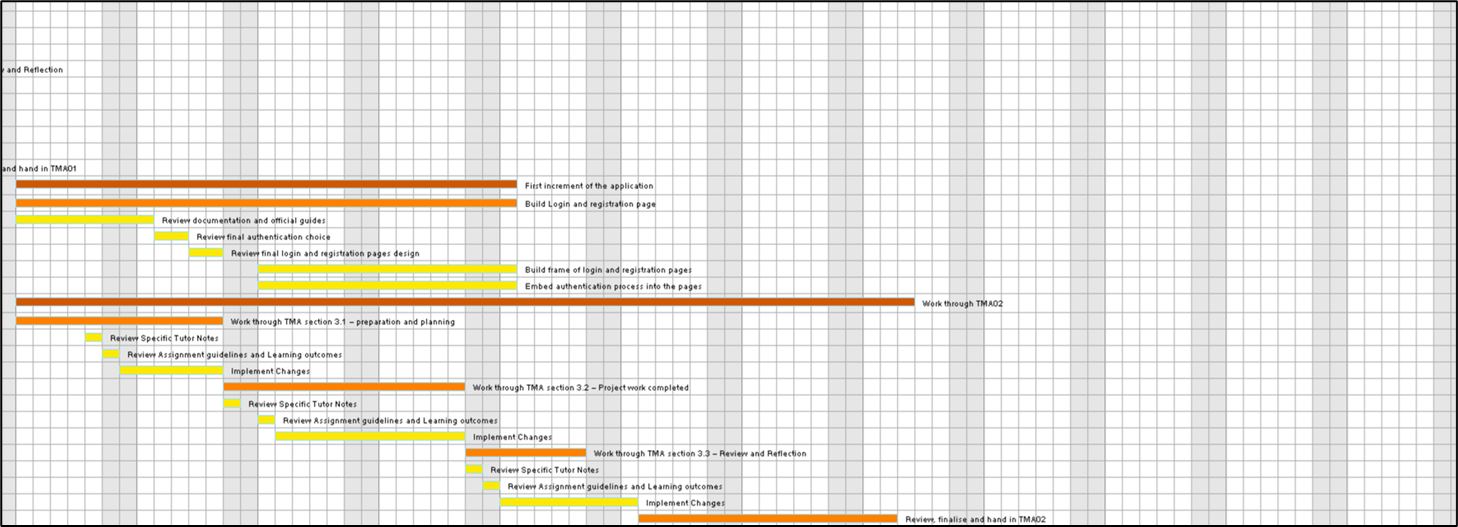
# Appendix

Appendix 1 – Schedule

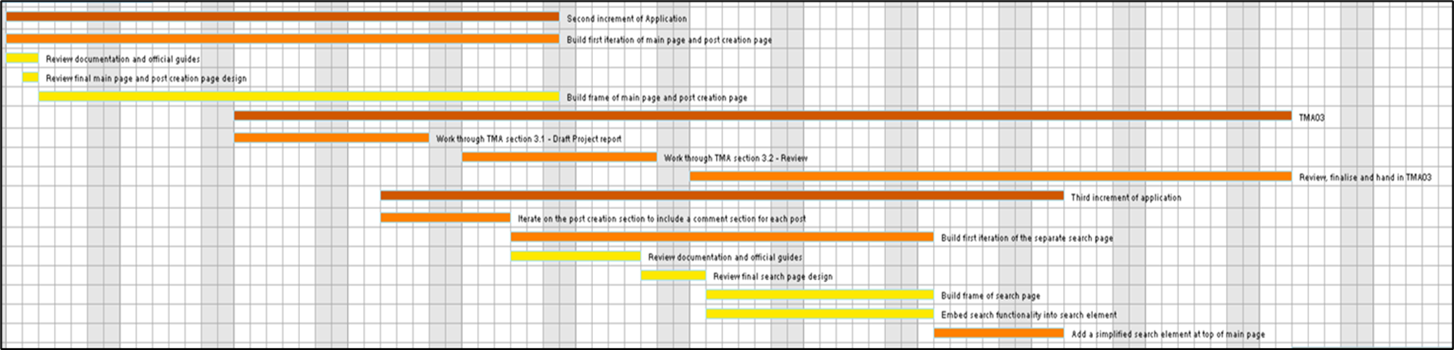
### Appendix 1.1 – Application Preparation and TMA01



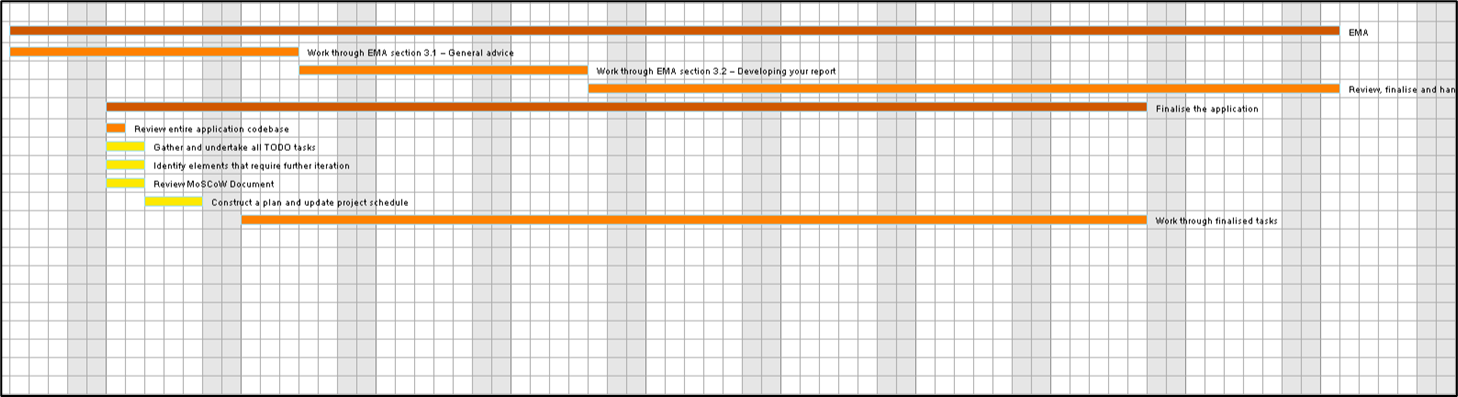
### Appendix 1.2 - First increment and TMA02



### Appendix 1.3 - Second increment and TMA03

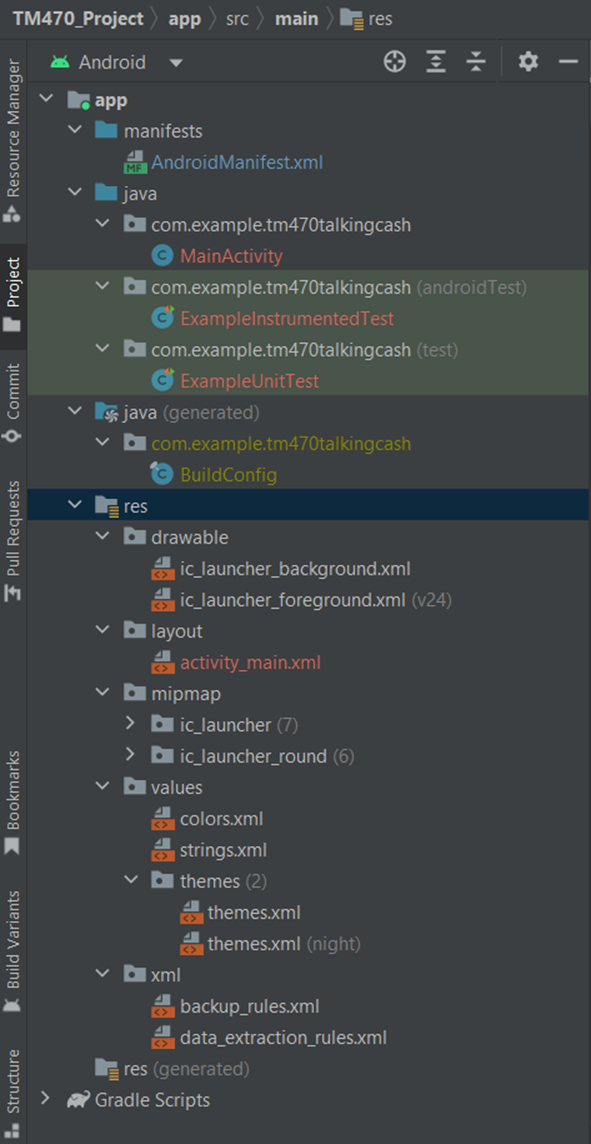


### Appendix 1.4 - Final increment, Iteration and EMA

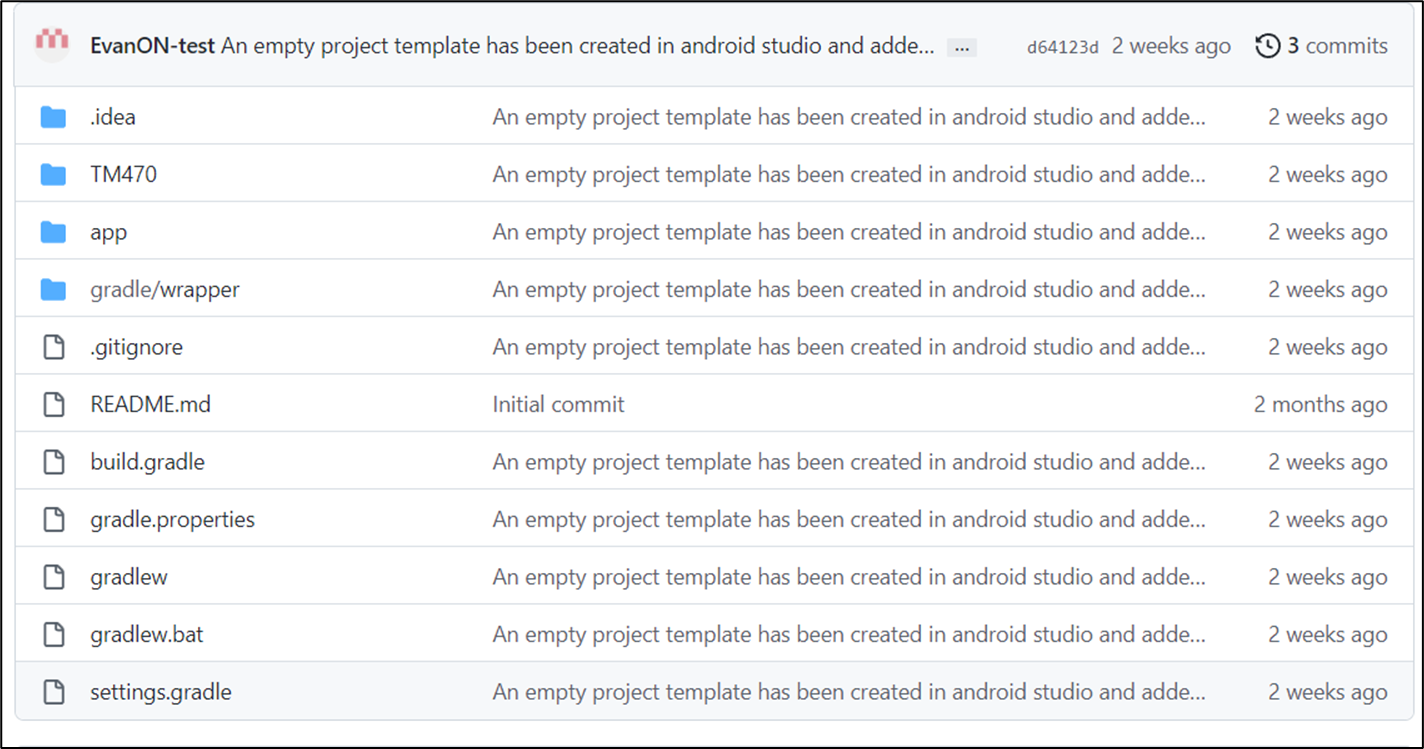


Appendix 2 – Previous Android studio and GitHub File structure

### Appendix 2.1 –



### Appendix 2.2 –



Appendix 3 - Wireframes

- Incorporating figures 3- 10 illustrating Work Completed

**Diagram

Description automatically generated with medium confidence**

Figure - Login page

**Graphical user interface

Description automatically generated with medium confidence**

Figure - Registration page

Table

Description automatically generated

Figure - Main page

**Diagram

Description automatically generated**

Figure - 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 - Post creation page

**Diagram

Description automatically generated with medium confidence**

Figure - 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 - A view of the menu button when clicked

*Text, letter

Description automatically generated*

Figure - 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

Put the extended java code for each element here??

### Appendix 5.1 – LoginActivity

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.Task;  
import com.google.android.material.textfield.TextInputEditText;  
import com.google.firebase.auth.AuthResult;  
import com.google.firebase.auth.FirebaseAuth;  
import com.google.firebase.auth.FirebaseUser;  
  
*/\*\*  
 \* 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

### 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 – Other files