FINAL AUDIT REPORT

Group 16:
Mathew Slingsby, Rui Santos, Reuben Sarkar,
Louai Sadraoui, Tomas Sekstela, Sachin Shah

Contents

Definition

The definition of the project scope was largely based upon three factors:

- The project's constraints
- Individual team members' personal development targets and motivation
- Social and social relation to the project and how it would benefit society.

Project Constraints

The project constraints that were given at the beginning of the project were as follows:

- 'Develop, as a team, an Android based mobile application with a remote (server-side or cloud-based) component.
- The application should solve a clearly-identified problem, which presents either a business opportunity, or a potential benefit to society. In either case, you will need to perform an evaluation of the need for your application and its potential for good in order to specify and justify the scope of your project.
    - In addition, your Android-based mobile application must include a data gathering component, along with server-side (or cloud-based) data analytics that relates to your application's main functionality, an could not run on the mobile device.
    - In addition, some technical constraints must be met by your application.
        - run on the tablets that were provided to you in your first year of University in their default configuration (with all system updates applied).
        - make use of data from at least two of the sensors specific to your mobile device (for example, camera, microphone, accelerometer or GPS).
        - fulfil a well-identified, but possibly limited, subset of their core functionality even in the absence of a permanent internet connection.
- Your proposal must be approved by your project sponsor, who will ensure that your proposal indeed fulfils the constraints listed above and will discuss with you the offline functionality constraint.'[1]

With the project constraints defined, our project's scope became clearer as we had objective goals which would have to be met, in order for the application to be considered successful.

[1] (Dupressoir, 2019)

Individual Team Members

All our team members are students at the University of Surrey. The members are listed below:

- Rui Santos (Computer Science Student)
- Reuben Sarkar (Computer Science Student)
- Louai Sadraoui (Computer Science Student)
- Mathew Slingsby (Computer Science Student)
- Sachin Shah (Computer Science Student)
- Tomas Sekstela (CIT Student)

Although we have 5 computer science students, we all have mostly different skill sets and development targets and motivations.

Team Member	Personal Objectives
Rui Santos	• To improve my leadership and teamworking skills. • Develop project management skills
Reuben Sarkar	• Improving my technical ability with firebase and location-based android • Learning to use GitHub to develop an app with multiple developers
Louai Sadraoui	• Improve teamwork with people that previously I didn't know • Working under pressure
Mathew Slingsby	• Learn git/version control • Improve group work
Sachin Shah	• Improve team working and communication skills • Become more familiar with database design and using Firebase
Tomas Sekstela	• Learn how to use android studio, • Be engaged in group work • Develop my design skills

As seen above, all team members have contrasting skillsets and development targets. This is critical in a team project as it allows for the exchange of various ideas and differing perspectives to solve problems. In addition, it allows for the personalization of tasks in the project as each team member will do what they are best at/what their objectives states, with the aim of improving efficiency.

Planning

To decide what application we would create, we had weekly meetings where we discussed project ideas and brainstorm together. All team members had a valid contribution to this brainstorm session and their ideas were added to the ideas document. After meeting, we would add the best ideas to a shortlist to then rank the ideas.[2]

After discussing for two weeks, we decided to vote for the best two ideas and then write the project outline for both. This would enable the group to view exactly how the app would have had to been developed and thus giving us a better understanding of which idea to pick. The two final ideas were to:

- Produce a meal delivery app where the user would request which food he/she wanted, and a delivery driver would be linked to him to bring him these foods.[3]
- Produce a fitness app which would track users' runs and would include a feature which enabled users to become friends and run together. [4]

After producing a project outline brief, the decision was clear to the group as only one fully met the project constraints and was feasible to do with the current team members. As a team, we voted in the next meeting and the second idea of producing a fitness app was voted for with 5 votes in favour and only 1 against.

Final Project Definition

As defined in the project definition document, the project scope states that the "app will notify users of other users within a selected radius. Users will be able to add friends and chat with other users. Users will be able to count their steps walked for the day and compare statistics with other users."[5]

[2] Retrieved from Appendix A – Idea's document
[3] Retrieved from Appendix B – Louai's final idea project brief
[4] Retrieved from Appendix C – Mat's final project brief
[5] Retrieved from our project's definition document.

Communication

As a group, we all exchanged phone numbers in order to create a WhatsApp group. This was used to notify the group to changes to planned meetings. We did this as we felt that it was not always necessary to hold a full meeting every week, especially during the implementation period. If a meeting was to be postponed or cancelled everyone would be notified through these means of communication. Another useful feature that we used this for was to create an event when a meeting was to occur, so that members could respond 'going' or 'can't go'. It also reminded them nearer the event. This as a whole made the team productivity increase dramatically.

Development Environment and Version Control

As we are all 2nd year students with no prior work experience in professional software development, debate took place about which platform to use. Looking at the benefits of GitHub, being able to support different versions within the same repository with tracking changes[6] [7], it was clear that this platform is successful and well respected by big organisations such as Google and Microsoft[8]. Therefore, it would be a great benefit not only for our project but also in the long term for every user to get familiar with source control.

After discussing about the best way of updating code, we started to discuss about what development environment to use to create our mobile application. The main programming language that the whole group was most proficient in was Java, thus we decided to use it as our programming language. None of our group members had experience with alternative programming languages that can be used for Android development (such as Kotlin), so Java was the clear choice.

The advantages that Android Studio provides was the most suitable for our needs and our mindset in coding, due to its gradle integration with being able to download libraries through modifying the gradle app file, which was not used in Eclipse[9]. Android studio is dynamic at assisting code and being able to correct it and report errors more dynamically. Android Studio has a very suitable UI which allows us to see real time updates to our code and design. Android Studio is more organised and simpler in terms of organising the files using module, whereas for example in Eclipse, the workspace must be merged. Furthermore, Android Studio allows developers to update the code periodically leaving out the effort to save which can be forgotten in important and long development process. A dedicated emulator is provided if the developers do not have access to a physical Android device. Finally, Android Studio has integration with useful third parties such as Google Cloud and others[10].

Relevant technology to use

A long discussion was made in how to process the backend and what technology to use. We chose to use Firebase due to a couple of key factors. Firstly, this software encapsulates all the necessary backend features for a mobile application. Secondly, Firebase is capable to provide a real-time database management environment, as well as security encryption, server-side project, hosting, cloud storage and an authentication system[11].

[6] (Bradford, 2019)
[7] (Brown, 2017)
[8] (Metz, 2015)
[9] (Rajput, 2015)
[10] (Alurkar, 2018)
[11] (Aicardi, 2016)

General, Social and Ethical Issues and Benefits

Firstly, greatest social and ethical issue related to our application is health. Our application's main purpose is to benefit society's overall health with the continued usage of our app. Currently in society, there is a problem with obesity/overall fitness[12] and loneliness[13]. Obesity and loneliness are a negative to society because they are related to various problems, both mentally and physically which can harm human life.

The reason why our app is beneficial to both these issues as it is a great way of exercising while making new friends. When you sign up to the app, it displays new people around you that are interested in having exercise. This solves the first problem of obesity as the more exercise the person does, the lower the risk of obesity. Also, this solves the second problem as well, as you keep meeting new people around you with the common interest of exercise. After meeting them, the app allows you to add them as friends, and it enables for the communication between both. This helps the loneliness problem as people will develop new friendships.

Secondly, another social and ethical issue regarding our app is security and authenticity. As users are inputting their sensitive data such as passwords and weight/height, we need to be encrypting the data and to make it secure. This will be done by encrypting all user data in our application's database.

Thirdly, there is the problem of privacy and anonymity as well as surveillance. Our application must be made in a way that other users can't see your private data without the user's consent. Furthermore, the user should be aware of all the data that our app stores about them. Both issues will be solved with our terms and conditions. When you first sign up for the application, you will be required to agree to our terms and conditions. This is done for legal and moral issues as it is critical that the user knows exactly what the app is doing and what it is storing.

When designing our app, we took into careful consideration these issues. When it came to data, we asked ourselves two questions, "Is the data accurate?" and "Is the data complete?"[14]. We had to ensure our app only stored the relevant information from the user necessary for the app to function to their needs. As such in the register activity, we only captured the following information from the user:

- Email Address – Used to authenticate and uniquely identify each user. Used to login and send reset password instructions.
- First Name – Basic information to make it easier to identify the user
- Last Name – Gives the user more of an identity, especially if different users have the same first name.
- Weight (Lbs) - Metric used in the calculation of calories burned.
- Password – Used to verify the user during login.
- Confirm Password – Necessary, to catch typos by prompting users to type their password twice

All this sensitive information is stored safely and securely on firebase. Passwords are encrypted automatically by firebase. Furthermore, in compliance with GDPR, we took explicit consent (through

[12] (Obesity and overweight, 2018)

[13] (Schulze, 2018)

[14] (Kotorov, 2015)

a check box) from the user to have their data collected. In addition, we made it like this so consent can be withdrawn at any time, through the option of allowing user to delete their account and the data that is associated with that account. "The right to be forgotten," meaning that people have the right to have their data deleted[15].

Another piece of highly sensitive information we had to think carefully about, was the location of users, and the personnel that would have access to this data. The current location of a user accessible by anyone is huge breach in user's privacy. We decide to store location coordinates (latitude and longitude) on Firebase and restrict access to Firebase to only the members of the group.

GDPR and Safety Concerns

As the application we designed is heavily based on a person's location at a given time we had to think seriously about our biggest issue, which was that we are collecting highly sensitive data from a user that could have severe consequences if it was stolen or used incorrectly. We disused the fact that users of our app, may potentially feel unsafe about providing their real time location data in this environment, however, we agreed that should it be carefully managed and controlled the application could still succeed.

We also researched this issue in detail so that we were confident that our solution to the issue met UK law requirements, especially the new GDPR regulations which are an extension to the Data Privacy Act. Essentially the GDPR regulations fall in to the following categories:

- **Right to be Informed** – This states that the user must be aware of 'fair processing information' including how and why we collect data and when we do. It also includes what we do with it. We agreed that we would make a privacy notice that a user must agree to before using the app. It will contain this information making us GDPR compliant in this area.
- **Right of access –** This states that a user must have or be able to gain access to their data. It also states that they can receive confirmation that we process their data. We agreed that a designated IO (Information Officer) was to be delegated and named in the privacy notice. This means that if a user wishes to make a SAR (Subject Access Request), which they are entitled to by law, we had a specific person in place to deal with it.
- **Right to Rectification –** This states that if the information we store, or use is inaccurate or out of date, the user has the right to have it changed. Similarly, to above, if a user wishes to make a request under this, the IO can handle the request by updating the data.
- **Right to Erasure** This allows the user to have a 'right to be forgotten' in which all data that could potentially identify the user or identify that the user used the application must be destroyed completely upon request if there is no legitimate reason for its continued collection or use. We also meet these requirements by having an IO and additionally a 'Delete Account' option which deletes all user data from the app and the database.
- **Right to Object** This states that the user must 'opt in' for us to collect and use their data, and if they object to us doing it, then we cannot collect or use any data from them as this would be in breach of the regulations and Illegal. The privacy notice also solves this issue.

Our application does not send any data to third parties, therefore regulations relating to this do not apply. The information officer (IO) for this project was Mathew Slingsby.

[15] (Falbe, 2018)

Security Issues

The main security issues we may face are data breaches. We had to ensure strict access to Firebase, to prevent unwanted authorization to our app. Like mentioned above, we only granted Firebase access to our group only. Any newly registered user was incapable of accessing the database and leaking information.

Viruses, and computer malware were also a great issue, that can cause a devastating social impact on our users. We made sure during development; we have antivirus programs installed on our PC's. We ran frequent scans to detect any virus or malware. Cyber risks such as these can cause a huge breach in data, and corruption of data. Which are reasons for huge penalties according to GDPR regulations.

Security of data

We do value the safety of the users of our app. For that user information will be stored in a secured database using Firebase.

Calories, number of steps, and weight might be considered as sensitive in terms of information about health, however this information will only be calculated with permission from the user. The user must also allow other users to see it.

Firebase is known to be GDPR-ready and trusted as it is made by Google. Furthermore, it is certified with many security standards such that ISO 27001 as an example, for every feature as Authentication, Cloud Storage. The software makes the use of IP addresses and unique IDs for data processing.

Therefore, anything which is not shared to the other users will not be shared anywhere thanks to how sophisticated Firebase is, facilitating the implementation and allowing us to focus on other parts of our software project[16].

We will ask the user to input important data twice as the email address and the password. Also, if the email is invalid, it will be known thanks to Firebase, also if the email address is already in use on the application.

A forgotten password and reset password system will also be implemented to facilitate the use of the application for users. The software allows us to say the users have nothing to worry about using our application.

Using authentication requirement, no part outside the app will be able to access private data of our users.

[16] (Firebase - Privacy, 2019)

Design

MVC for architectural design pattern

Our team chose waterfall as an approach to project management and software design development. Waterfall is commonly used in small group software engineering projects. Since the scope of our project isn’t that huge compared with real world companies’ applications, it was easy to predict the following steps and methods we need to apply.

Advantages of waterfall development techniques:

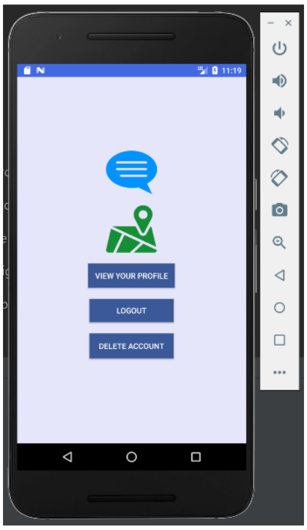
- It uses clear structure
- Determines how the finalized program should look like
- It is easily understandable for the customer.

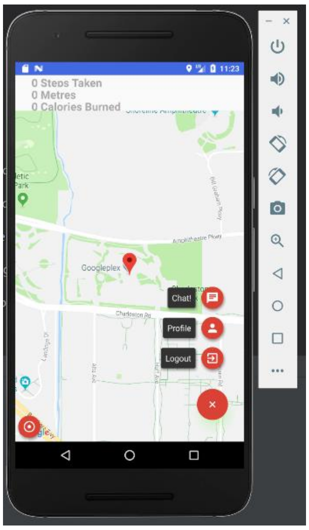
Disadvantages of waterfall development techniques:

- Difficult to make changes during development phase
- There is high risk that the customer expectations won’t be met, because it is hard to make changes to the project and customer might change its opinion of how the completed project should look like.
- More documentation and less code
- Testing is only being held after completing the coding part

Abstraction

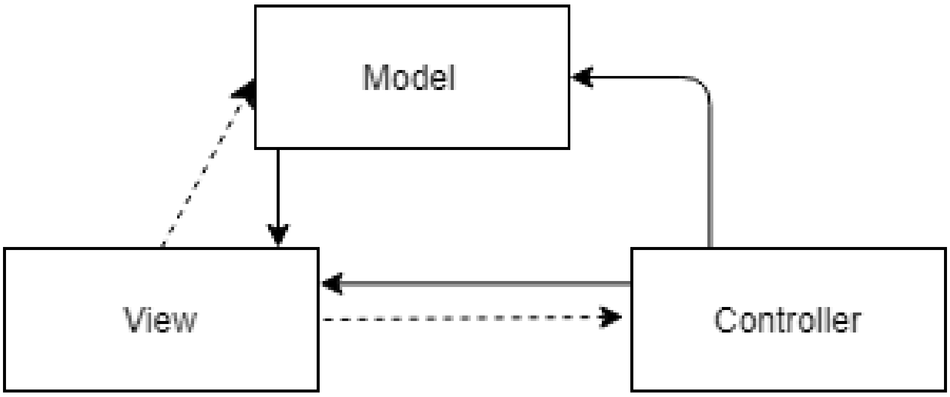
While developing the application the team tried to simplify the program so that it could be intuitive and easily understood by users. We minimized the extra steps the user needs to take to achieve the same outcome. For example, instead of having the main menu as an additional activity we implemented the main menu alongside the map. The first image shows the design with the extra step for accessing the map wherein the second image main menu is integrated into the map.





Architecture

For architectural pattern, our group used a model view controller (MVC) pattern. MVC helps to develop programs with clearly distinct aspects of the application (business logic, input logic and UI logic) while providing strong interconnection between these components.

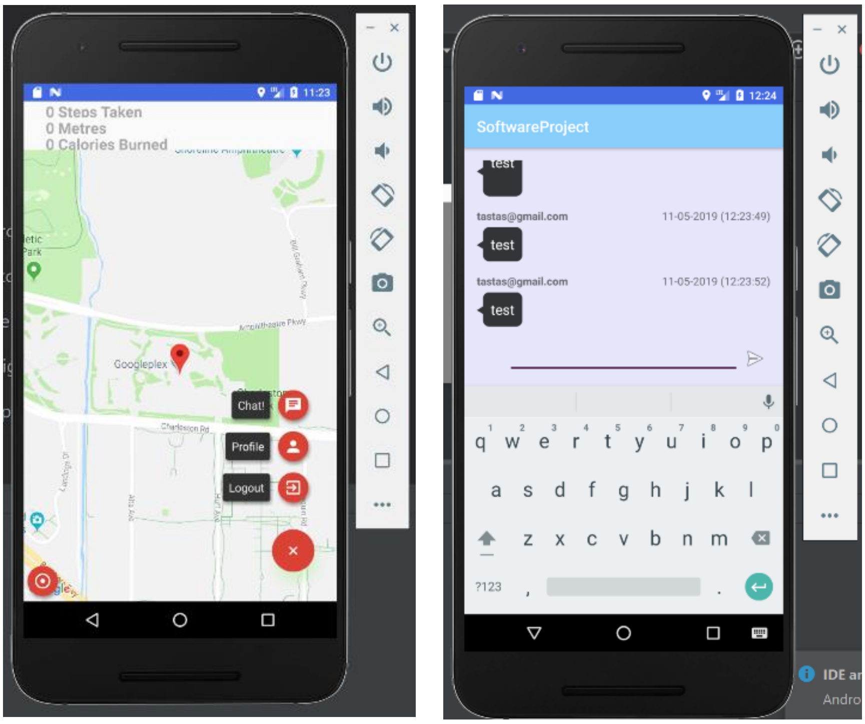


The model performs as data storage, encapsulates logic and data, performs validation. In our case Firebase is the main component of a model, it responds to applications enquiries, notifies view of any changes. Also, firebase provides us an integrated security system, all data is encrypted.

View sends user input to the controller and shows any change that was made in the model. In our application view is presented as UI.

The controller defines application behaviour, it initiates the changes in the model accordingly with users input. Backend of the program acts as a controller in our app.

For example, when a user is logged in the system he can see a man and main menu, he chooses to click on the chat icon. This is the action of input, then the user's input is being sent to the controller where it asks server the data of previous chats and the finally it uploads the chat view with all the text messages.


0 Steps Taken
0 Metres
0 Calories Burned

test

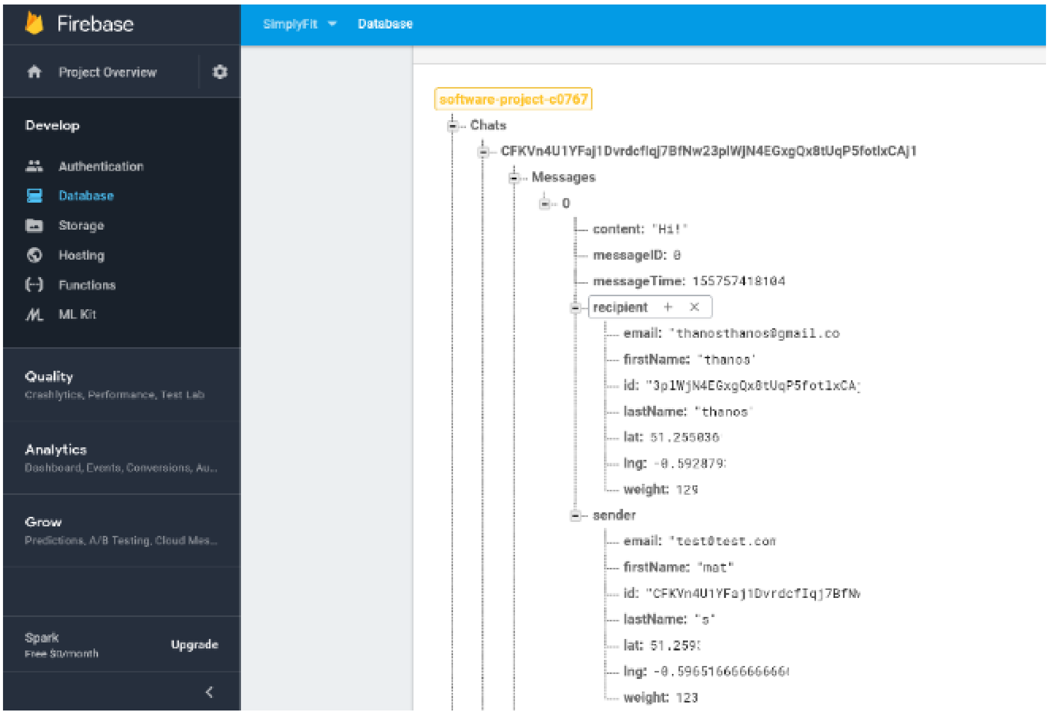
tastas@gmail.com 11-05-2019 (12:23:49)

test

tastas@gmail.com 11-05-2019 (12:23:52)

test

User's input (clicks on icon) -> Controller asks the model to give the data and loads it

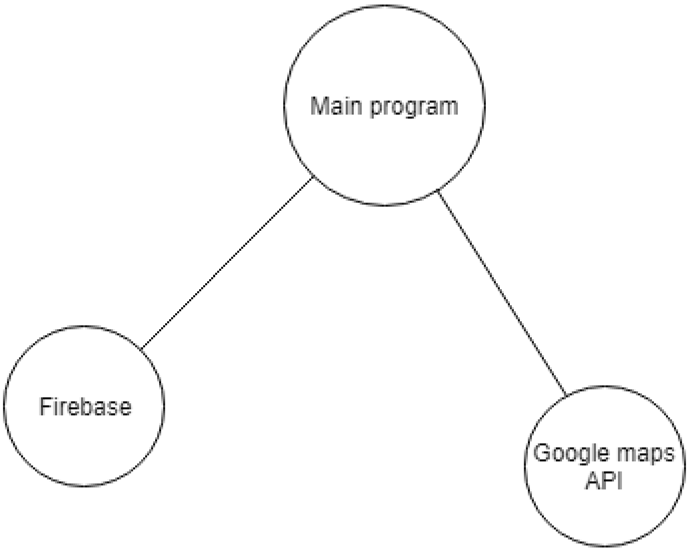


```
software-project-c0767
  Chats
    CFKVn4U1YFaj1DvrdcfIqj7BfNw23plWjN4EGxgQx8tUqP5fotlxCAj1
      Messages
        0
          content: "Hi!"
          messageID: 0
          messageTime: 155757418104
          recipient  +  ×
            email: "thanosthanos@gmail.co
            firstName: "thanos"
            id: "3plWjN4EGxgQx8tUqP5fotlxCA;
            lastName: "thanos"
            lat: 51.255036
            lng: -0.592879:
            weight: 129
          sender
            email: "test@test.con
            firstName: "mat"
            id: "CFKVn4U1YFaj1DvrdcfIqj7BfNv
            lastName: "s"
            lat: 51.259:
            lng: -0.5965166666666(
            weight: 123
```

Modularity

Modularity could be defined as software division into unique components which are known as models. The concept of modularity is highly effective when dealing with large applications. Thus, the different modules could be divided, and developers could work on different parts of it. Modularizing the design is useful because it helps to plan development of a program, also do the testing and make changes more easily. However, it could be quite difficult to merge the modules into one system.
Our application could be divided into three distinctive modules:

- Firebase (which acts as a database)
- Google maps API
- Main program

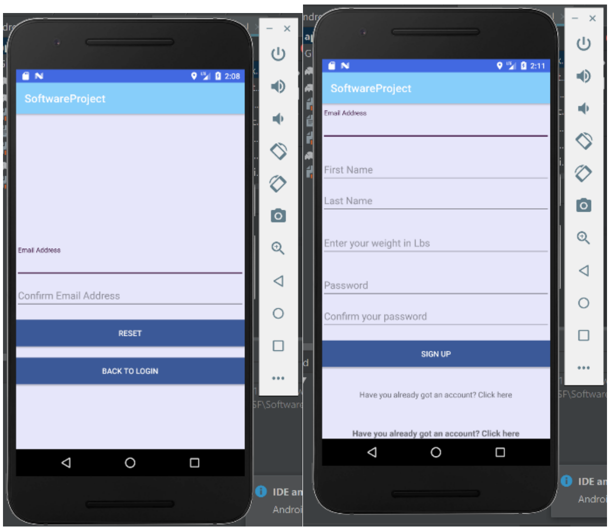


Even though modularization is simple and useful in large systems, it brings some challenges to our program. The main issue is compatibility, Google maps API can change or be updated, firebase also can update itself. Thus, we must update our program in accordance with other components updates.

Security

Never has security been so important as it is today, that is why a core aspect of all of our design processes was focusing on properly implementing security features to minimise the security risks mentioned in the planning stage.

SimplyFit app uses FiresBase as its main data storing component. Firebase is a secure real-time database that encrypts the data, so nobody can steal it from the user. Our application also authorizes the user before it lets the user in. We are using simple registration and logging in the mechanism. In a registration part user is asked to provide its email address, first and last names, the weight I Lbs, password and confirm the password. This information is used to identify a user, for example, there can’t be two accounts with the same email address thus we are using an email address as an id in the firebase data system. Weight is used to track users progress while using the app. Confirmation of password is used to prevent a typo error that might occur. Login window asks the user to only input the email address and password. Lastly, if the user forgets his password, he can reset his password by typing in his email and confirming it, then the email is sent to the provided email address which helps to change the password.



Have you already got an account? Click here

Have you already got an account? Click here

Creating an app which enables users to communicate brings a lot of different challenges. One of them is dealing with hate language. Since we are not going to publish this app to the public we don’t have any measures to prevent the issue or to investigate users’ messages. However, if the program was to be released to public use we would provide complain centre to solve these issues here users could notify us about each case, from there our support team would start the investigation and confirm or deny the action of hate speech. People involved in a such as bullying action would have to take responsibility.

We have set out rules and conditions which user must accept to use the app. These rules conform and integrate all legal constraints including the GDPR and Data Protection Act.

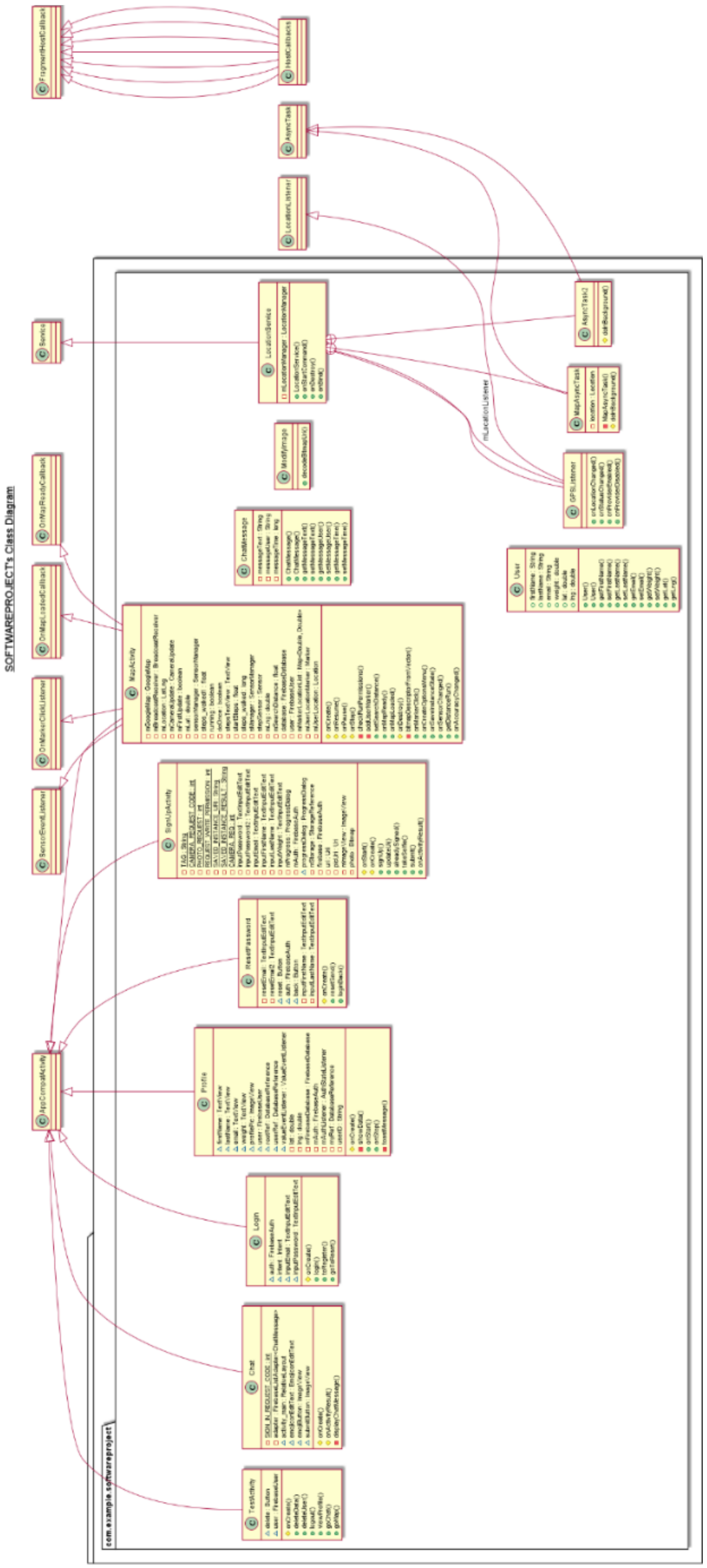
Colours

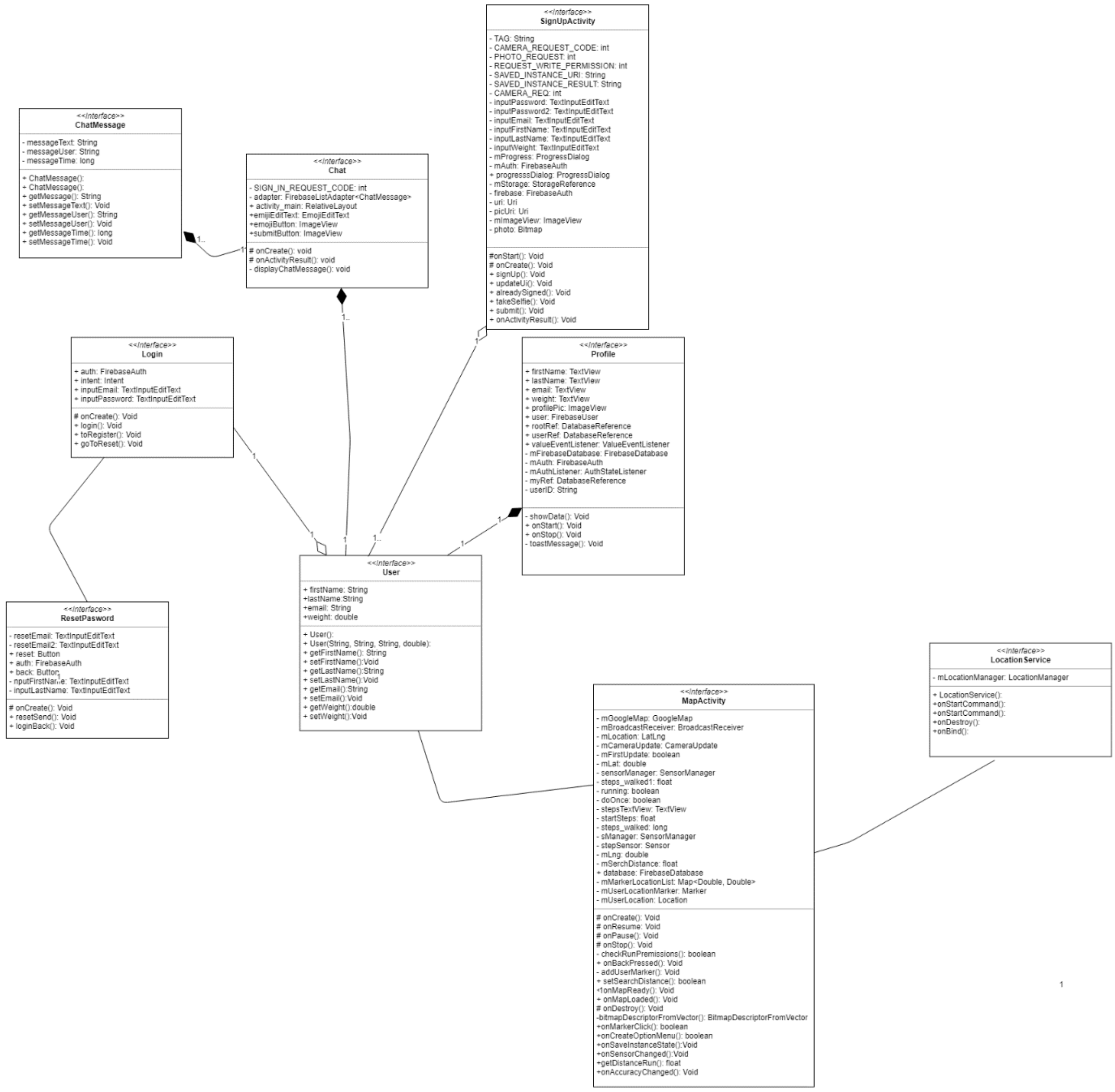
As we can see that the app’s main colour is red, the logo, layouts of the program uses the colour red and its shades. The colour was chosen for the psychological aspect, it is effective to attract user attention, we can see the same analogy with traffic lights. Thus, when the user unlocks his phone the red icon will stand out. The app purpose is to encourage, motivate and enable people to walk more, colour aspect plays a role here as well. Recent research suggests that 76% of people associates colour red with speed.

UML

Unified model language is a visual representation of software components. By these representations, we can better understand how each component communicates and what attributes and relationships it has with the rest of them. For our program, we used a backward design approach, where UML diagram changes were followed by code changes. For UML diagrams we used different android studio plugins like graphviz or sketh_it to better understand the structure of the whole app. These tools were used to automatically generate the diagram (first screenshot). However, to better understand relationships and connectivity between different classes we had to use web drawing tool draw.io to manually draw the UML sketch (second photo).

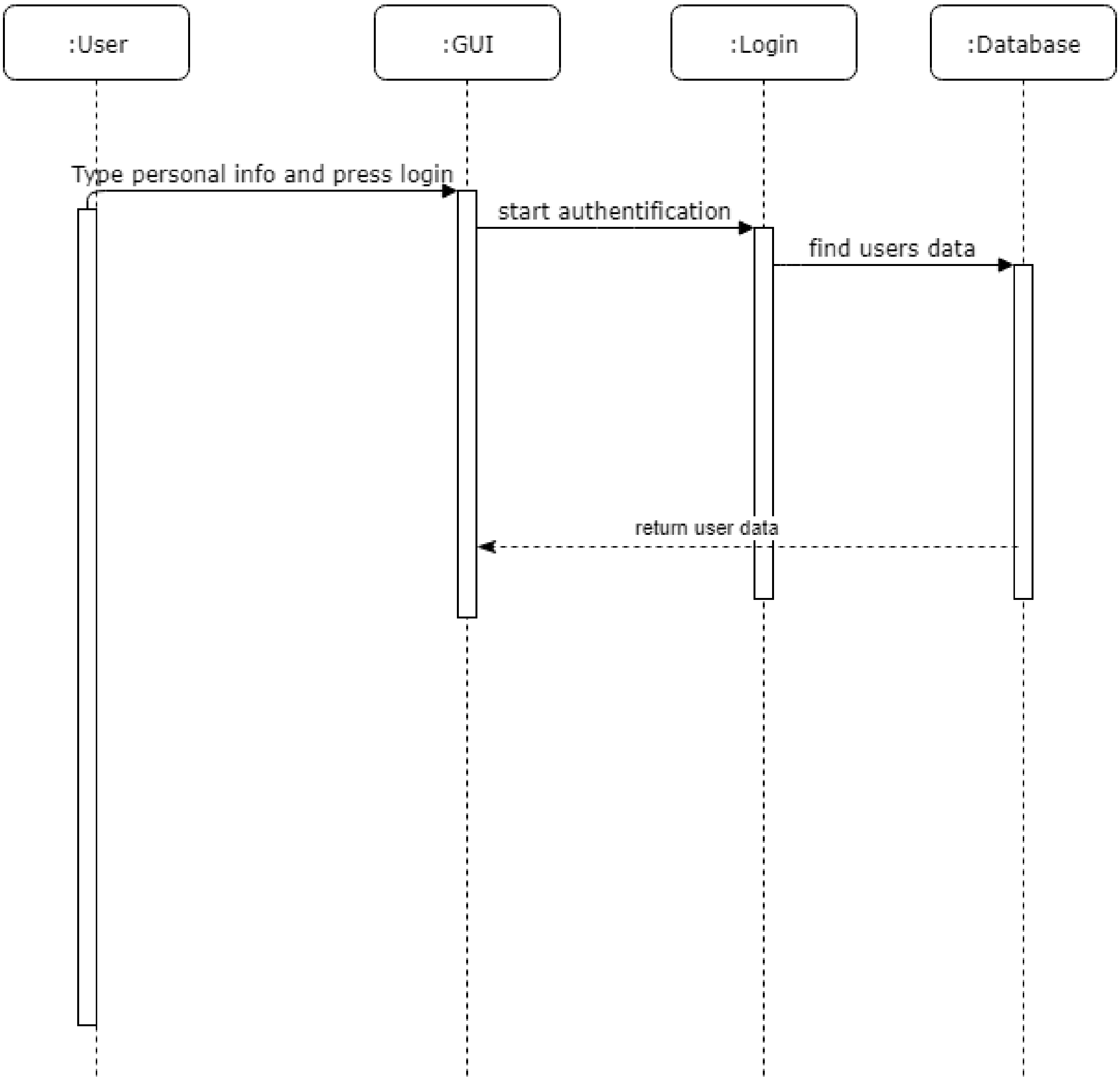
A copy of the files is included in the Appendix section for easier viewing.


SOFTWAREPROJECT's Class Diagram
com.example.softwareproject
FragmentHostCallback
HostCallbacks
AsyncTask
LocationListener
Service
OnMapReadyCallback
OnMapLoadedCallback
OnMarkerClickListener
SensorEventListener
AppCompatActivity
LocationService
ModifyImage
ChatMessage
MapActivity
SignUpActivity
ResetPassword
Profile
Login
Chat
TestActivity
AsyncTask2
MapAsyncTask
GPSListener
User
mLocationListener



In the second photo, we tried to simplify the UML to emphasize its classes relationships. Class Login can work without any users thus it has a relationship of aggregation with the user class, we could describe Sign up Class relationship with User Class in the same way. However, the class profile can only exist if there is class user thus they have a relationship called composition, moreover, it has one to one relationship which means that one user can only have one profile to view his details. Class Message has also the composition connection with the Class Chat, messages couldn't exist without class Chat, it is also one to many relationships thus one chat can have many messages but not vice-versa. MapActivity class is interconnected with user class, it notifies the user class of any changes, it mostly works with Google services.

The design diagram below shows how client-server multi-layer architecture is used in our app.



Firstly, the user gives some input that can be login details in this case, then the input data is sent to the controller (MVC) which asks firebase database to confirm or deny the users input to authenticate the user. If user details are correct and the user exists in the database, then the login class (controller in MVC) receives a positive response. User is logged to his profile; all profile details are loaded from the database.

Implementation

We started the implementation with a basic login and register activity for the user. This allowed new users to register, login and use our app. All relevant data was collected from the user with privacy issues in mind. Firebase was used to authenticate users and store the relevant user registration details.

We decided to use Google Maps API to implement the entire map functionality. The API automatically handles access to Google Maps servers, data downloading, map display, and response to map gestures[17]. We used the API to add current location markers for each user, and to update the marker locations as the app detects the user is moving. We have an overlay, displaying information such as steps walked, metres travelled, and calories burnt to the user.

We decided we only needed the step value on Firebase. Metres travelled was calculated using a function using the formula: distance = (steps*78)/100 (since 78cm is the average stride length of a male / divide by 100 to get answer in metres). Calories burnt was also calculated using a function and a formula; involving the weight of the user, and the distance travelled in metres.

We used 3 sensors in our implementation. The Android platform provides two sensors that let you determine the position of a device: the geomagnetic field sensor and the accelerometer[18].

We used these sensors to track the movement of each user. We understand the privacy concerns this may raise, so we made sure to ask for location permissions from the user, before displaying their location on the map. The tracking of movement was important in finding and communicating with other users through our chat. The accelerometer was also used to track step movements for each user. This allowed for the implementation of leader boards, where users can competently compare their steps with others. Camera was used to allow the user, to optionally add a picture to their profile.

We also had to cater for offline functionality. We understood that the user may not always have an internet connection present. We wanted to keep track of steps even when the user is offline.

When designing the database on Firebase we took great care to maintain good database design practices. Since data is structured on Firebase as a JSON tree, we weren't used to the traditional SQL database, with tables and records. Following best practices, when designing we took care to avoid nesting data; We kept our data structure as flat as possible. We instead split the data into separate paths (denormalization), so it can be downloaded efficiently in separate calls. Following the creation of the database, we used many child references, and event listeners to capture the relevant data from the database; mainly tracking location and steps.

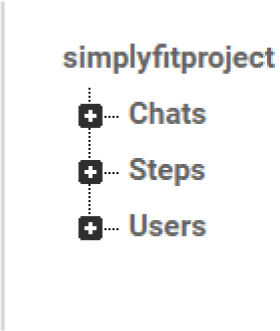
[17] (Android SDK Documentation, n.d.)

[18] (Sensors Position, n.d.)

Firebase

We chose to use Firebase as our database as members in our group had previous experience with it, and certain members expressed a desire to develop their skills with Firebase. Firebase also has the benefit of being a NoSQL database that is easy to implement and learn[19]. This was beneficial to use over alternatives such as MySQL, since these alternatives need a dedicated host. By using Firebase, we can render the effective cost of implementing a server to 0.

Firebase is a real-time database with a simple structure to build and understand. Since some of the group members were not familiar with database languages such as SQL, Firebase, being a NoSQL database, allowed members to become familiar with working around databases which was a personal motivation for a lot of members, and reduced the need for understanding complex queries. This enabled the server-side implementation to be completed efficiently so that other areas may be focused on.

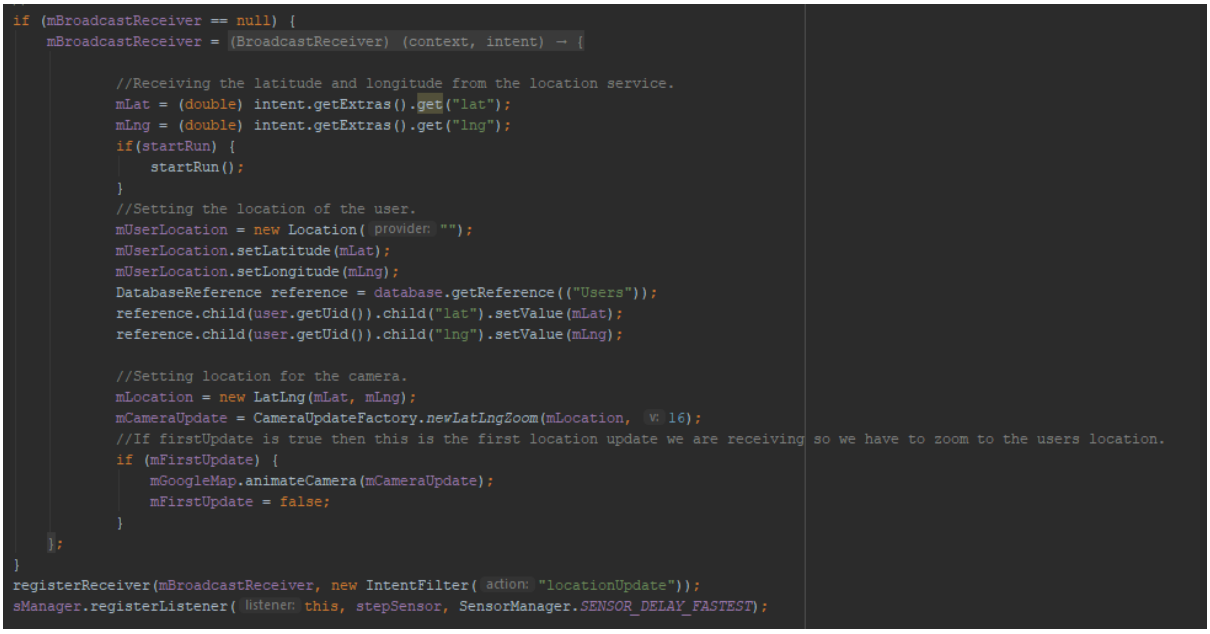


simplyfitproject
- Chats
- Steps
- Users

Map

We chose to use the Google Maps API to design our map activity, since it is readily available, has all the features we needed, and the development team had previous experience with the API. We use the devices GPS functionality to get the users latitude and longitude on every location update. The latitude and longitudes for each user are saved in the Firebase database and updated on every location change. We then update the markers in real-time to display users as they are moving and have the app open.

[19] https://firebase.google.com/use-cases



```
if (mBroadcastReceiver == null) {
    mBroadcastReceiver = (BroadcastReceiver) (context, intent) → {

            //Receiving the latitude and longitude from the location service.
            mLat = (double) intent.getExtras().get("lat");
            mLng = (double) intent.getExtras().get("lng");
            if(startRun) {
                startRun();
            }
            //Setting the location of the user.
            mUserLocation = new Location( provider: "");
            mUserLocation.setLatitude(mLat);
            mUserLocation.setLongitude(mLng);
            DatabaseReference reference = database.getReference(("Users"));
            reference.child(user.getUid()).child("lat").setValue(mLat);
            reference.child(user.getUid()).child("lng").setValue(mLng);

            //Setting location for the camera.
            mLocation = new LatLng(mLat, mLng);
            mCameraUpdate = CameraUpdateFactory.newLatLngZoom(mLocation,  v: 16);
            //If firstUpdate is true then this is the first location update we are receiving so we have to zoom to the users location.
            if (mFirstUpdate) {
                mGoogleMap.animateCamera(mCameraUpdate);
                mFirstUpdate = false;
            }
    };
}
registerReceiver(mBroadcastReceiver, new IntentFilter( action: "locationUpdate"));
sManager.registerListener( listener: this, stepSensor, SensorManager.SENSOR_DELAY_FASTEST);
```

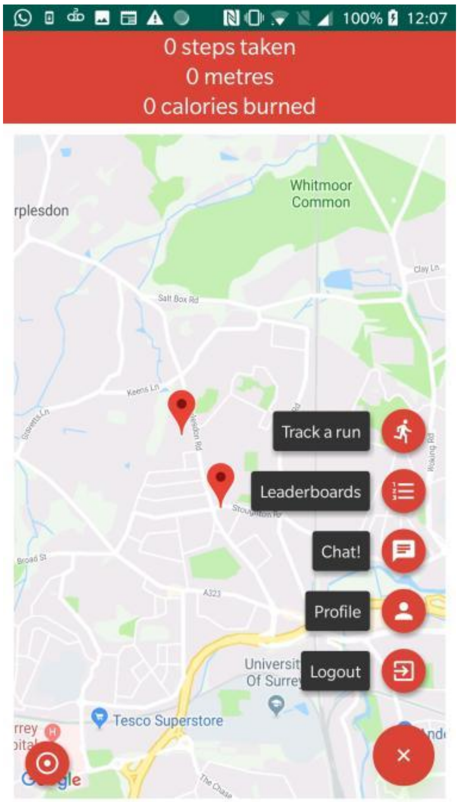
The map displays the markers of all users within 10 miles of the user's radius. This is done through calculating the distance between the LatLng objects of users. If the distance between the users is less than 10 miles, the marker is displayed to the user's phone.

Displaying the markers of all users brings about a sense of community since it encourages the more social aspects of fitness, especially when you can see other users running/exercising through the app.

GUI

The bulk of the design of the app is the map screen, which contains several features, so it is crucial the map is attractive and intuitive to use. To ensure this, we chose a modern and fluid design with a minimalistic colour palette, which better suited the current trends in mobile applications[20].

[20] https://blog.prototypr.io/the-rise-of-minimalism-in-app-design-91524d3fdf4b


0 steps taken
0 metres
0 calories burned

One of the major design choices was the use of a floating action button menu instead of a traditional appbar menu. We felt that this was a more modern choice and fit the aesthetic of the application better.

We aimed to make the app as easy to use with minimal instructions. Since we did not choose to use an app bar for the menu, we could use that space to display all the essential information the user would need, such as the steps taken, distance travelled, and calories burned. This is always visible to the user and the user does not need to leave the map screen for general use with the app.

Sensors

Camera

The camera sensor is used upon user registration where the user can choose to take a profile picture that is then associated with their account. This profile picture can be viewed by other users if they choose to click on the marker.

Accelerometer

The second sensor we make use of is the accelerometer. This is used to accurately determine the steps walked by a user since it is based off a change in acceleration in the sensor as opposed to the alternative method which is to calculate distance changed in the location.

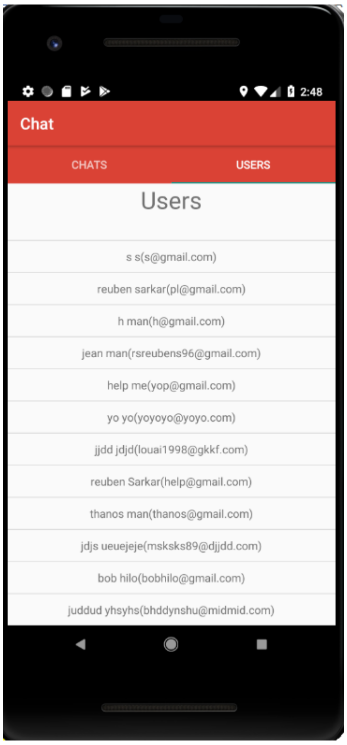
If we were to calculate the steps using distance changes, then this could potentially be inaccurate, such as the user being in a vehicle. By using the accelerometer, we reduce the potential of inaccurate leader board scores.

GPS

Simply Fit uses the GPS sensor of the phone to track the user. This sensor is also used in conjunction with the accelerometer to track the steps that a user takes. The GPS is used to receive the latitude and longitude of users, which is then stored on the database, allowing all users to have markers to display the location of other users.

Chat

The chat feature allows users to chat with another user of the application. To start a chat with someone, you must indicate the user in which you would like to chat to. This is done on the 'Users' tab shown below. The current user is omitted from the list for obvious reasons. The app also checks if a chat already exists and prevents multiple from being made. This feature will work in both directions. (e.g. if user 1 creates a chat with user 2, when user 2 tries to create a chat with user 1, it will fail as a chat between them exists)



Users

s s(s@gmail.com)

reuben sarkar(pl@gmail.com)

h man(h@gmail.com)

jean man(rsreubens96@gmail.com)

help me(yop@gmail.com)

yo yo(yoyoyo@yoyo.com)

jjdd jdjd(louai1998@gkkf.com)

reuben Sarkar(help@gmail.com)

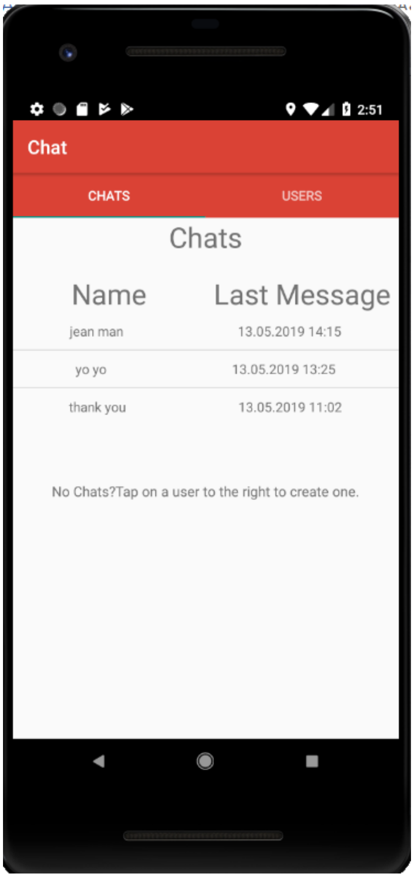
thanos man(thanos@gmail.com)

jdjs ueuejeje(msksks89@djjdd.com)

bob hilo(bobhilo@gmail.com)

juddud yhsyhs(bhddynshu@midmid.com)

Once a chat has been started, a chat session is created consisting of a list of 2 users and a list of messages. This will then be displayed on the 'Chats' tab.

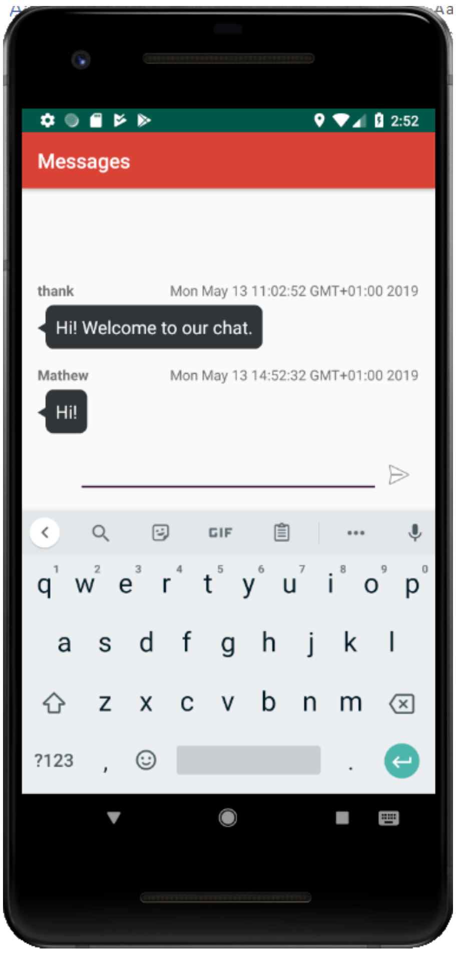


Chats

Name	Last Message
jean man	13.05.2019 14:15
yo yo	13.05.2019 13:25
thank you	13.05.2019 11:02

No Chats?Tap on a user to the right to create one.

When a chat is opened, a messaging activity is then displayed. This is updated in real time. This is shown below.



thank Mon May 13 11:02:52 GMT+01:00 2019

Hi! Welcome to our chat.

Mathew Mon May 13 14:52:32 GMT+01:00 2019

Hi!

Sever and Offline Functionality

Offline functionality

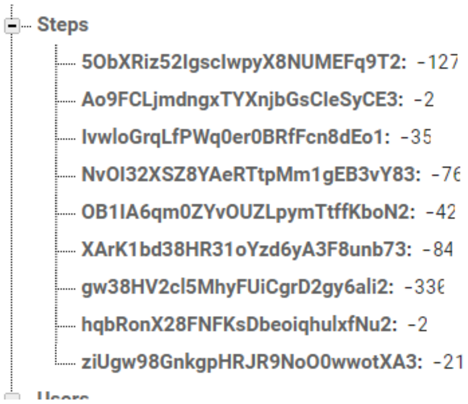
The constraint for offline functionality for the app is achieved by keeping a local field for the number of steps a user has walked on the phone. This is updated as well as the database, however if internet connection is lost, the local field is still updated with the steps since this is activated by the accelerometer on the phone. Upon reconnection with the database, if there is a discrepancy with the value of the steps on the phone and on the database, the database is updated to the new value stored on the phone.

The same functionality is used for the latitude and longitude of the client so that upon reconnection, other users can be notified of the new location of the client.

Server-Side Processing

To fulfil the constraint of server-side processing prevalent in the application, we implemented a leader boards feature which will display the user with the highest number of steps taken in the database.

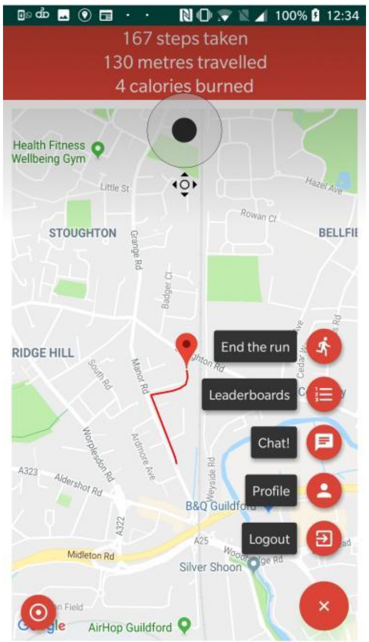
The data is received by the client as a sorted array from Firebase by using a public method and is then displayed on the client in the order received. A limitation we discovered is that Firebase can only sort data in ascending order, however we needed the steps in descending order. To overcome this, the step count is stored negatively in the database, so that the highest steps are of the lowest value, and when the data is sorted ascendingly, it is in the correct order. This saves computational cost on the client since no more processing is needed.



```
Steps
    5ObXRiz52IgsclwpyX8NUMEFq9T2: -127
    Ao9FCLjmdngxTYXnjbGsCleSyCE3: -2
    IvwloGrqLfPWq0er0BRfFcn8dEo1: -35
    NvOI32XSZ8YAeRTtpMm1gEB3vY83: -76
    OB1IA6qm0ZYvOUZLpymTtffKboN2: -42
    XArK1bd38HR31oYzd6yA3F8unb73: -84
    gw38HV2cl5MhyFUiCgrD2gy6ali2: -336
    hqbRonX28FNFKsDbeoiqhulxfNu2: -2
    ziUgw98GnkgpHRJR9NoO0wwotXA3: -21
Users
```

Tracking Runs

Simply Fit allows the ability for a user to start tracking their run. This is done by pressing a button on the menu which then begins the drawing of a polyline which is updated to every new location update. This results in a line that accurately tracks the route the user has taken.



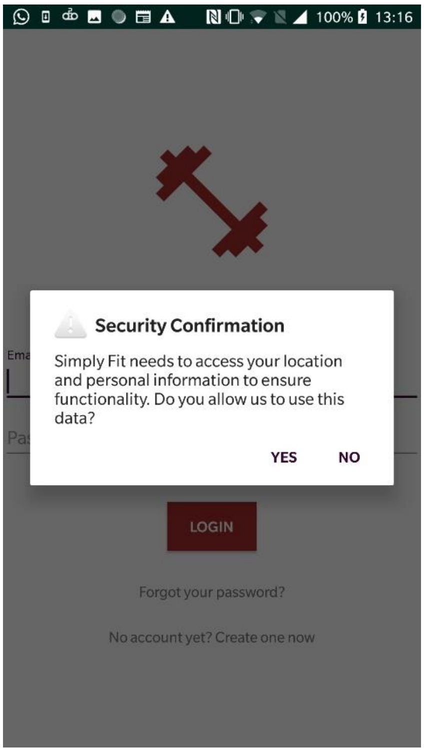
100% 12:34

167 steps taken
130 metres travelled
4 calories burned

Upon ending a run, the user is notified with a congratulatory message informing them of the number of calories they burnt, the distance they have ran and the number of steps they took during the run.

Privacy Notice

To ensure we have permission from the user to store sensitive data in accordance to GDPR laws, we have implemented a dialog that opens when the app is launched that requests permission from the user to store their data. If the user agrees, the dialog closes and the user can continue to login and use the app. If the user disagrees, then the app closes.



Security Confirmation

Simply Fit needs to access your location and personal information to ensure functionality. Do you allow us to use this data?

Evaluation

General Evaluation

We met of our functional requirements and project objectives to effectively combine fitness and social activity as outlined in our PDD.

We achieved implementation of all the essential requirements of the app to have a functioning and usable product that can be adopted by a user into their daily life. These include the ability to track users in real time, the ability to communicate with other users through a chat client to plan walks, a functional login system, the ability to track user statistics such as calories burned, and steps walked, and overall sleek design.

We were unable to implement some features such as the ability to save runs and the routes taken for specific runs due to a lack of time and expertise. We have learnt from this and in future we will implement a more thorough work plan that is better suited towards everyone’s work schedules.

Another weakness of our app is that Firebase is not very scalable and cannot perform complex queries. Our code is also very computationally complex and if we were to handle large amounts of users we would need to make our code more efficient. However, the use of Firebase suits our current needs which is a small community of users.

Testing

We have designed a thorough test plan that will test all the main functional requirements of our app. The following table describes this test plan to ensure our app is robust.

Test Plan

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
1	The app should allow users to login, provided they have a valid email address and password.	The user in on the Login activity. The user has registered their email address and password with firebase. The user has access to an internet connection and is connected to firebase.	The app will require, the user's email address and password. These details will be checked against the database to ensure validity.	On successful login, the user should be redirected to the map activity where they should see their current location, other user's location, steps taken, calories burnt, and meters travelled.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
2	The app should allow users to register, provided they aren't already registered.	The user is on the Sign-Up activity. The user is not registered with firebase. The app is connected to firebase.	The app will require the user's: Email address, first name, Last name, Weight in Lbs., Password, and Confirm Password. Optionally, choose to add profile picture.	On successful register, the user should be redirected to the map activity where they should see their current location, other user's location, steps taken, calories burnt, and meters travelled.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
3	The app should not allow users to register if password character length is < 6.	The user in on the Sign-Up activity. The app is connected to firebase.	The app will require the user to enter a password that is less than 6 characters.	User should not be redirected to the Map activity and should see the following toast displayed: 'Password too short, enter minimum 6 characters!'	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
4	The app should not allow users to	The user is on the register activity. The	The app will require the user to	User should not be redirected to Map activity and should	By Inspection and displaying the screen shot

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
	register if passwords do not match.	user has filled in all other fields. The password is greater than 6 characters. The app is connected to firebase.	enter passwords, that do not match.	see the following toast displayed: 'Password reentered does not match'.	which confirms the result. Stating pass or fail.
5	The app should not allow users to register if they leave any fields blank.	The user is on the Sign-Up activity. The app is connected to firebase.	None – The user must simply keep the fields blank.	User should not be redirected to the Map activity and should see a toast message for the field left blank.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
6	The app should not allow users to login if they enter incorrect details.	The user is on the Login activity. The app is connected to firebase.	The app will require the user to enter wrong email address and/or password.	User should not be redirected to the map activity and should see an authentication failed toast.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
7	The app should not allow users to login, if they leave the email address and/or password field blank.	The user is on the login activity. The app is connected to firebase.	The app will require the user to leave email address and/or password blank.	User should not be redirected to map activity and should see the following toast displayed: 'Enter email address!' or 'Enter password!'	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
8	The app should allow the user to reset their password if they have forgotten it.	The user is registered and has an existing email address. The app is connected to firebase. User is on the Reset password activity.	The app will require the user to press the 'Forgot your password?' text. User also needs to enter email address information.	User should be redirected to the 'Reset Password' activity, where they will need to enter and confirm their email address. User should press the reset button, and see a toast, notifying them of reset instructions.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
9	The user should be	The user has logged on, with location	User has logged in or	User should be redirected to the map activity,	By Inspection and displaying the screen shot

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
	able to access the map.	services enabled to get current location.	recently registered.	where they should see their current location, location of other users, steps taken, calories burnt, and meters travelled.	which confirms the result. Stating pass or fail.
10	The user should be able to access their profile.	The user has logged on. The user should be on the Map activity. The app is connected to firebase.	User presses, the floating action button, and selects the profile option.	User should be redirected to the Profile activity. The user should see their: Profile picture, first name, last name, steps taken, meters travelled, and calories burned.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
11	The user should be able to access the chat.	The user has logged on. The user should be on the Map activity. The app is connected to firebase.	User presses, the floating action button, and selects the chat option.	User should be redirected to the Chat activity. The user should see messages from every user.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
12	The user should be able to logout from their account.	The user has logged on. The user should be on the Map activity. The app is connected to firebase.	User presses, the floating action button, and selects the logout option. User then presses Yes on the confirm dialog box.	User should be redirected to the Login activity.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
13	The app should not allow new users to register with an existing email address.	The user is on the Register activity and is connected to firebase. User has entered valid password.	User registers with email: t@gmail.com.User tries to register again with same email address.	User should see a toast with the message: 'The email address is already in use by another account.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
14	The app must track, and update steps walked as the user is moving.	The user is on the map activity and is taking steps with the device.	None – The user simply must take steps with their device.	The 'Steps walked' counter should increment as the user is moving. The location marker should move. Value should be updated in firebase.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
15	The app must track, and update meters travelled as the user is moving.	The user is on the map activity and is taking steps with the device.	None – The user simply must take steps with their device.	The 'Meters travelled' counter should increment as the user is moving. The location marker should move. Value should be updated in firebase.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
16	The app must track, and update calories burnt as the user is moving.	The user is on the map activity and is taking steps with the device.	None – The user simply must take steps with their device.	The calories burnt value should increment as the user is moving. The location marker should move. Value should be updated in firebase.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
17	The user must be able to view the profile of other users.	The user has logged on. The user should be on the Map activity. The app is connected to firebase.	The user clicks on the info window of a location marker of another user.	The user should be able to view the other user's profile. User should see details such as: Profile picture, first name, last name, steps taken, meters travelled, and calories burned.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
18	User should be able to track another runner's movement.	The user has logged on. The user should be on the Map activity. The app is connected to firebase.	None – The user must simply obverse for movements for the location	The location marker for other users must change.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
			marker of other users.		
19	The user should be able to view the step count leaderboards.	The user has logged on. The user Is on the Map activity. The app is connected to firebase.	User presses, the floating action button, and selects the leaderboard s option.	The user should be redirected to the Leaderboards activity, where they should see user's and the number of steps taken, in ranked order (most to least).	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
20	The app should continue to track steps when offline and update on firebase once back online.	The user is logged on. The user is offline. The user is on the Map activity. The user continues to take steps.	None – The user simply must observe the step value on firebase with the value on Maps activity.	The steps value on firebase should update to the one displayed on Maps activity, once the user comes online.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
21	The user should be able to chat with other users nearby displayed on the map.	The user has logged on. The user is on the Map activity. The app is connected to firebase.	User presses, the floating action button, and selects the chat option. User then selects "Users" tab, and the user they want to chat with.	The User should see a chat interface, where they can send and receive messages from that specific user.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
22	The user should be able to view their most recent chats with users.	The user has logged on. The user is on the Chat activity. The app is connected to firebase.	User presses, the "Chats" tab, and selects on the user they want to view their chat with.	The User should see a chat interface, where they can view all their messages from that specific user.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
23	The user should be able to form group chats with other users.	The user has logged on. The user is on the Chat activity. The app is	User presses, the "Group Chats" tab, and selects on the	The User should see a chat interface, where they can view messages from the	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.

Test Number	Requirement	Preconditions/ Dependencies	Expected Inputs	Expected Results	Test Evaluation
		connected to firebase.	group chat they want to view messages for.	members of the group.	
24	The app should reset the step count every 24 hours at 00:00.	The user is registered and has taken more than 0 steps in 24hrs.	None – The simply must observe their step count reset to 0 at 00:00.	The step count should reset to 0, every 24hr at 00:00.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
25	The user should be able to upload a profile picture.	The user is on the Sign-Up activity. The app is connected to firebase.	The user needs to select an image as their profile picture.	The user should be able to view their profile picture on the Register activity or Profile activity. Other users should be able to view the user's profile picture via their location marker info window.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.
26	The user should not be able to see the location of other users more than 10 miles away.	The user has logged on. The user is on the Map activity. The app is connected to firebase.	None – The user simply must observe users around a 10-mile radius	The user should not be able to see users more than 10 miles away from their current location.	By Inspection and displaying the screen shot which confirms the result. Stating pass or fail.

Evaluation of our Ethical/Social Issues

We can see how the current state of the app can raise some concerns regarding the ethical usage of data. This is because the current app draws markers for every user without an option to opt out of location updates. We also do not have extensive security measures to ensure the integrity of the user's data, however Firebase does provide basic levels of security such as password hashing and encrypted users. In future, we will add the ability to allow users to opt out of publicly sharing their location and provide more offline functionality.

Another potential ethical and social concern is that we are storing the chats in our database, therefore chats between users are not completely secure. This is true, however, with this choice we can also ensure that our chat is not being abused in any illegal or inappropriate way and can moderate the data on our application if a user raises a concern with us. To ensure that our own members do not abuse the ability to view the chat history, access to the database is provided to a select few members, who also do not possess any major criminal background.

The nature of a fitness app is not the most accessible as our target audience is mainly people who are able to be active. This excludes a subset of people who are physically disabled and thus are unable to utilise the benefits of our app. This could negatively impact the reach of our app, however in the future we could add features for disabled people too.

Reflection

General Group Reflection

Throughout this project, we believe we have been a productive team, like a real-life software development group. Each individual has been responsible for the outcome of the project which we believe to be a success. Throughout the process, each member has followed our team charter, creating a positive work environment for us to succeed in. We have also all had a chance to show our ability in our area of strength and at the same time learnt something new. Having all set personal development targets at the start of the project, we can now reflect on these to determine how the project has affected us.

We have reflected on each stage of our project and have evaluated our efforts to reach our goals and whether it was successful or not. These are detailed below.

Definition and Planning Reflection

Our Goal	Evaluation and Evidence
The planning and Research stage of our development was aimed at understanding the current market and deciding what to create for the project. We also aimed to establish a good work ethic amongst group members and sort the group administration out.	We started out by researching ideas as discussed in the planning section of this document. Throughout the project all group members were productive and were generally enthusiastic about achieving our goals. This can be seen from our meeting logs and minutes and the initial planning documents, which are all provided in the appendix. To improve as a group, we would spend more time planning the specific parts of the application. In this project we simply formed a vague idea to move forward with into design. It was for this reason we struggled to visualise how to progress with GUI mock ups and other core design features.
We wanted to form a good team charter encompassing everybody's needs and suggestions. We wanted to focus people on areas of strength whilst also giving them room to expand their knowledge and meet their personal goals.	Again, as seen in our documented meeting minutes, we spent a considerable amount of time discussing each individuals' strengths and personal targets. We came up with the best roles to maximise efficiency, whilst keeping the workload varied amongst everyone. We were happy with these choices and believe they were the correct ones. In future, we would stick to the same team charter as we believed it was successful in this project.

Design Reflection

Our Goal	Evaluation and Evidence
We tried to simplify the UI of program as much as we could. Instead of showing the main menu page and displaying the possible pathways a user could take, we implemented main menu on the map. This and other choices minimized the time user spends to navigate throughout our application	The evidence of our design choice can be seen in the final deployment of our app. We believe this choice we made was effective as it significantly reduces the number of redundant steps a user needs to take to reach their desired page. This provides a simple and effective UI for the user. In future, we would spend more time on more detailed aspects of the design. These include things like custom map markers and a nicer looking step display. This would provide the user with a more enjoyable in app experience.
MVC architecture design pattern helped us to distinguish and divide each part of an app to model view and controller.	As mentioned in design documentation we used firebase as our database to store all user data, it is extremely secure because it provides data encryption. However, whilst implementing our app, we experienced some serious issues with our firebase database which caused our whole project to become corrupt. We fixed this issue by creating a new database and rebuilding the project.
We aimed to create a secure, user friendly application that complied to all the necessary requirements including GDPR. As our application handles real time location data we wanted to spend time focusing on getting this area of the application correct.	We didn't have any issues with implementing design security. This outcome was likely as our application was constructed to be user friendly, for example it tries to protect users from typos by forcing them to confirm chosen password and email. We also give the user a chance to reset password which can only be done by receiving the confirmation email. Our database is secure due to encryption and use of firebase which is GDPR ready and is also being managed by Google.
The main goal of choosing colours was to attract user's attention when they start using their phone.	This combination of characteristics made it easy for us to use the colour in our logo and in applications UI. The feedback we got from people that we had given access to the app fulfilled our expectations, a lot of them emphasized how the colour helped app to stand out from the others.

Implementation Reflection

Our Goal	Evaluation and Evidence
We set out to use our design methods along with our functional requirements to develop a user friendly and useful application. (See Functional Requirements)	All our group members believe that the application that we created is what we set out to achieve. Although there are some areas that we wish to improve, we understand that very few projects are perfect and without areas of improvement. We have used all our skillsets to produce a fully working map activity that tracks steps, distance and calories. It also only displays users within a 10-mile radius. We have a fully implemented chat and the application has server and offline functionality. The application more than meets the constraints we were given and believe we have created something with a purpose and that will benefit society. We do have some ideas for improvement however. Our first area of improvement would be to create a group chat feature allowing users to create walking groups in their communities. We would also like to implement a challenge system that issues rewards for completing certain tasks. However, as this would require third party sponsorship, this was outside the limits of this project.

Bibliography

Aicardi, S. (2016, June 13). *4 Reasons Why Google's Firebase is Essential if You Have a Business App*. Retrieved from Semrush: https://www.semrush.com/blog/4-reasons-why-google-s-firebase-is-essential-if-you-have-a-business-app-1-1-1/

Alurkar, A. (2018, March 13). *What are the benefits of creating an app in Android Studio?* Retrieved from Quora: https://www.quora.com/What-are-the-benefits-of-creating-an-app-in-Android-Studio

Android SDK Documentation. (n.d.). Retrieved from Google: https://developers.google.com/maps/documentation/android-sdk/intro

Bradford, L. (2019, January 6). *What is GitHub and why you should use it*. Retrieved from The Balance Careers: https://www.thebalancecareers.com/what-is-github-and-why-should-i-use-it-2071946

Brown, K. (2017, September 6). *What is GutHub and what is it used for*. Retrieved from How-To Geek: https://www.howtogeek.com/180167/htg-explains-what-is-github-and-what-do-geeks-use-it-for/

Overview of the GDPR Retrieved from https://ico.org.uk/media/for-organisations/data-protection-reform/overview-of-the-gdpr-1-13.pdf

Data Protection. (2018). Retrieved from Gov UK: https://www.gov.uk/data-protection

Dupressoir, F. (2019, February 14). *Software Engineering - Student Handbook.* Retrieved from Surrey Learn: https://surreylearn.surrey.ac.uk/d2l/le/content/172325/viewContent/1542393/View

Falbe, T. (2018, March 16). *Ethical Design: The Practical Getting-Started Guide*. Retrieved from Smashing Magazine: https://www.smashingmagazine.com/2018/03/ethical-design-practical-getting-started-guide/

Fernandes, R. P. (2017, February 4). *Firebase keepSynced(true)*. Retrieved from Stack Overflow: https://stackoverflow.com/questions/41478849/firebase-keepsyncedtrue?rq=1

Firebase - Privacy. (2019, February 7). Retrieved from Google: https://firebase.google.com/support/privacy

Kotorov, R. (2015, June 12). *Why Ethics Matter when building an app*. Retrieved from Business2Community: https://www.business2community.com/mobile-apps/ethics-matters-building-app-01248104

Metz, C. (2015). *How GitHub conquered Google, Microsoft and everyone else*. Retrieved from Wired: https://www.wired.com/2015/03/github-conquered-google-microsoft-everyone-else/

Obesity and overweight. (2018, February 16). Retrieved from WHO: https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight

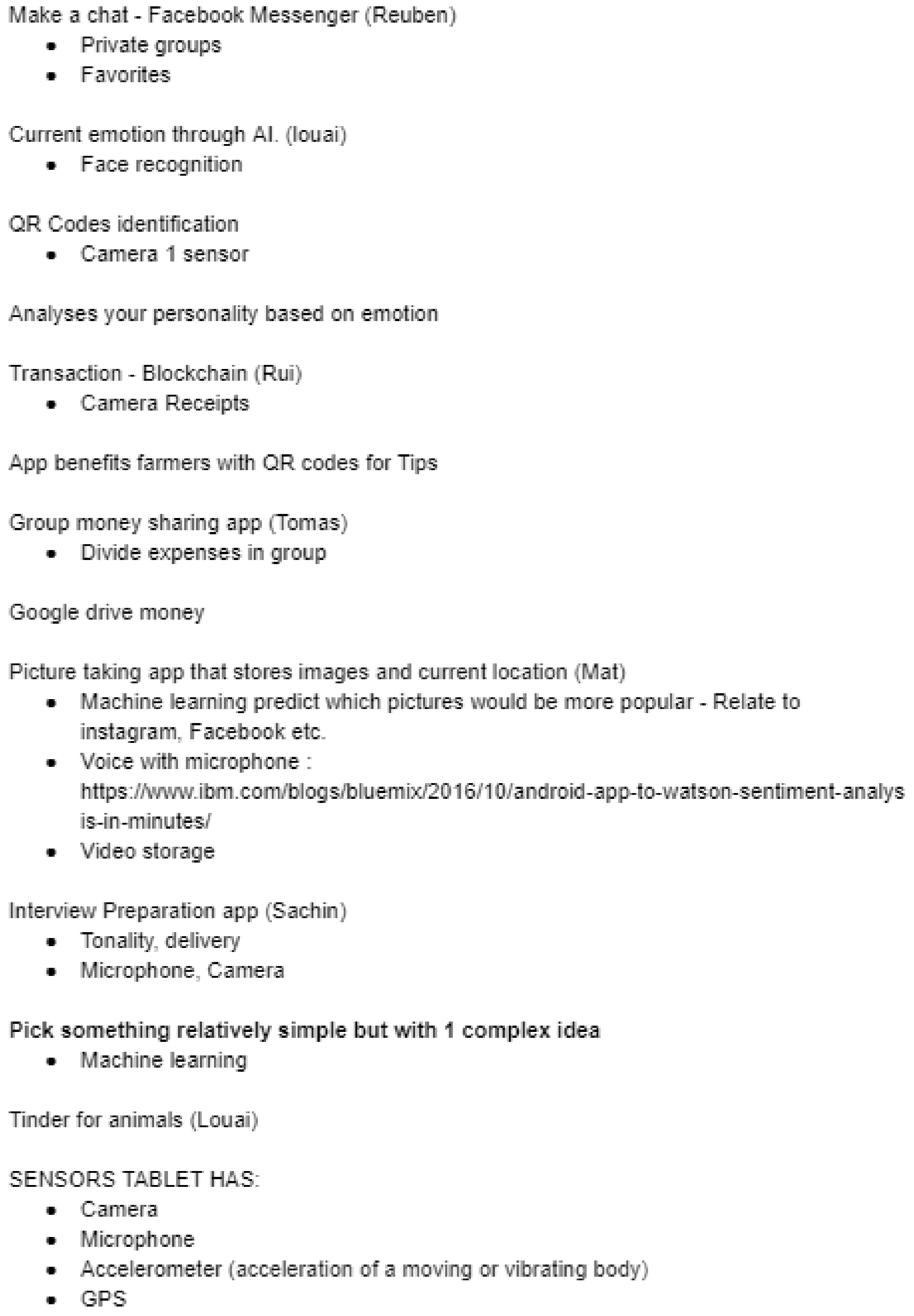
Rajput, M. (2015, May 20). *Why Android Studio Is Better For Android Developers Instead Of Eclipse*. Retrieved from Dzone: https://dzone.com/articles/why-android-studio-better

Schulze, H. (2018, April 16). *Loneliness: An Epidemic?* Retrieved from Harvard: http://sitn.hms.harvard.edu/flash/2018/loneliness-an-epidemic/

Sensors Position. (n.d.). Retrieved from Android Developer Guide: https://developer.android.com/guide/topics/sensors/sensors_position.html

Appendix

Appendix A – Ideas document



Make a chat - Facebook Messenger (Reuben)

- Private groups
- Favorites

Current emotion through AI. (louai)

- Face recognition

QR Codes identification

- Camera 1 sensor

Analyses your personality based on emotion

Transaction - Blockchain (Rui)

- Camera Receipts

App benefits farmers with QR codes for Tips

Group money sharing app (Tomas)

- Divide expenses in group

Google drive money

Picture taking app that stores images and current location (Mat)

- Machine learning predict which pictures would be more popular - Relate to instagram, Facebook etc.
- Voice with microphone : https://www.ibm.com/blogs/bluemix/2016/10/android-app-to-watson-sentiment-analysis-in-minutes/
- Video storage

Interview Preparation app (Sachin)

- Tonality, delivery
- Microphone, Camera

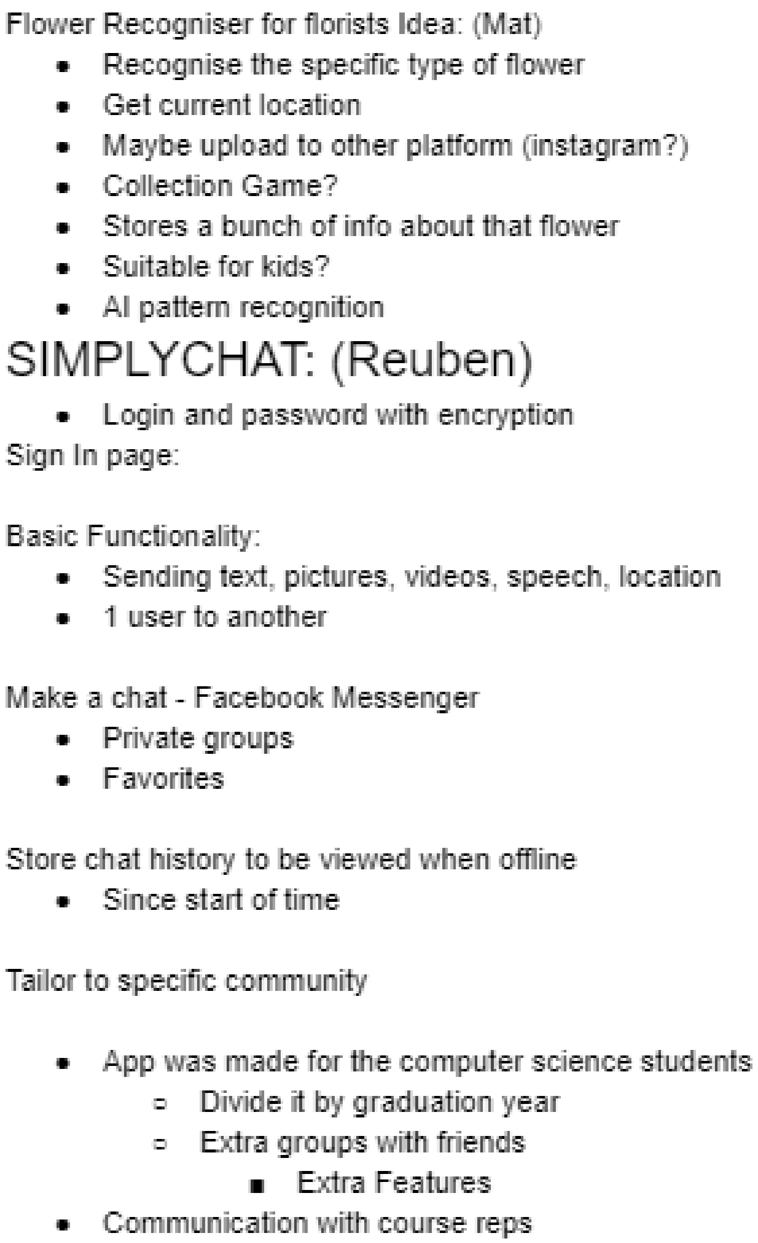
Pick something relatively simple but with 1 complex idea

- Machine learning

Tinder for animals (Louai)

SENSORS TABLET HAS:

- Camera
- Microphone
- Accelerometer (acceleration of a moving or vibrating body)
- GPS



Flower Recogniser for florists Idea: (Mat)

- Recognise the specific type of flower
- Get current location
- Maybe upload to other platform (instagram?)
- Collection Game?
- Stores a bunch of info about that flower
- Suitable for kids?
- AI pattern recognition

SIMPLYCHAT: (Reuben)

- Login and password with encryption

Sign In page:

Basic Functionality:

- Sending text, pictures, videos, speech, location
- 1 user to another

Make a chat - Facebook Messenger

- Private groups
- Favorites

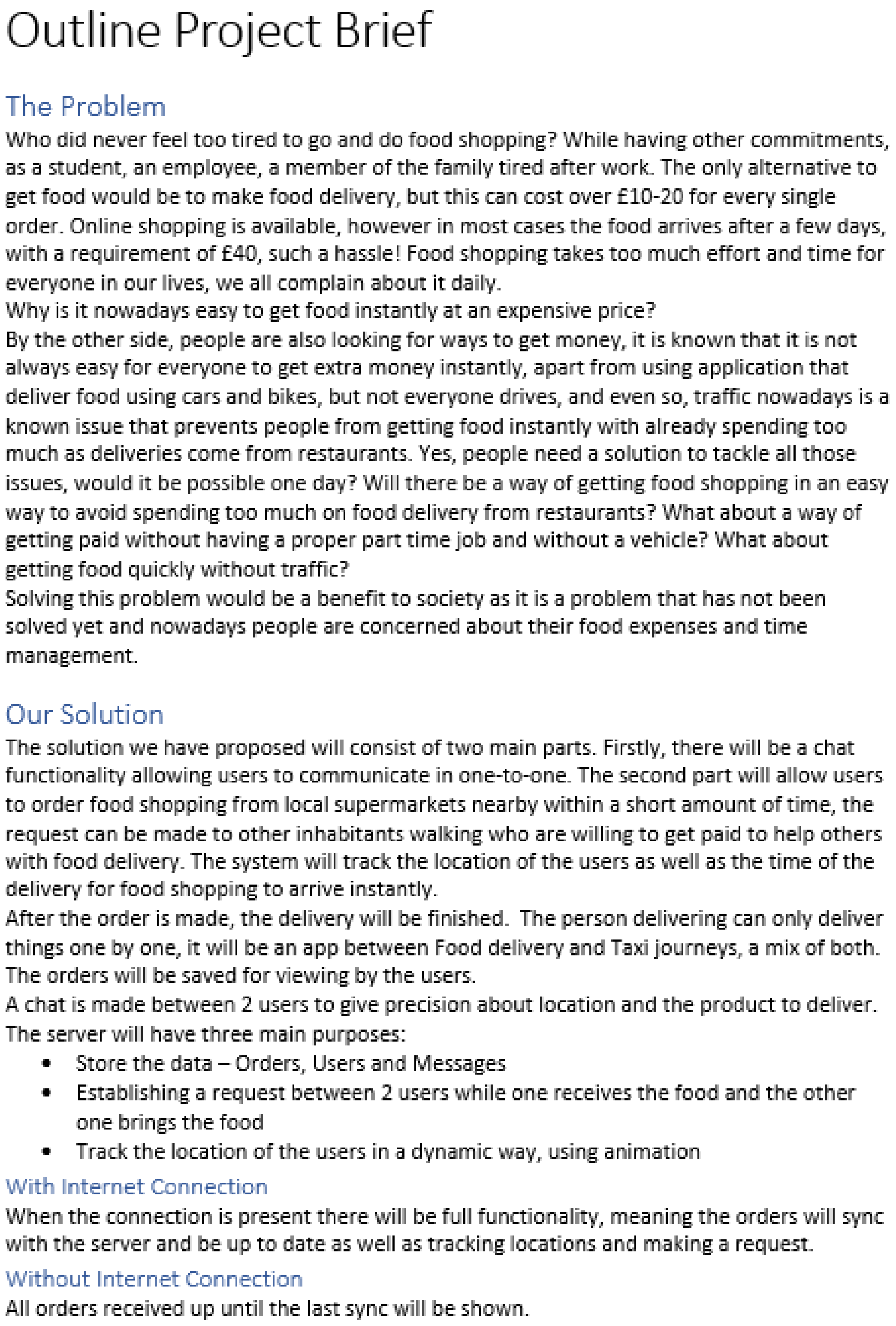
Store chat history to be viewed when offline

- Since start of time

Tailor to specific community

- App was made for the computer science students
  - Divide it by graduation year
  - Extra groups with friends
    - Extra Features
- Communication with course reps

Appendix B - Louai's final idea project brief



Outline Project Brief

The Problem

Who did never feel too tired to go and do food shopping? While having other commitments, as a student, an employee, a member of the family tired after work. The only alternative to get food would be to make food delivery, but this can cost over £10-20 for every single order. Online shopping is available, however in most cases the food arrives after a few days, with a requirement of £40, such a hassle! Food shopping takes too much effort and time for everyone in our lives, we all complain about it daily.

Why is it nowadays easy to get food instantly at an expensive price?

By the other side, people are also looking for ways to get money, it is known that it is not always easy for everyone to get extra money instantly, apart from using application that deliver food using cars and bikes, but not everyone drives, and even so, traffic nowadays is a known issue that prevents people from getting food instantly with already spending too much as deliveries come from restaurants. Yes, people need a solution to tackle all those issues, would it be possible one day? Will there be a way of getting food shopping in an easy way to avoid spending too much on food delivery from restaurants? What about a way of getting paid without having a proper part time job and without a vehicle? What about getting food quickly without traffic?

Solving this problem would be a benefit to society as it is a problem that has not been solved yet and nowadays people are concerned about their food expenses and time management.

Our Solution

The solution we have proposed will consist of two main parts. Firstly, there will be a chat functionality allowing users to communicate in one-to-one. The second part will allow users to order food shopping from local supermarkets nearby within a short amount of time, the request can be made to other inhabitants walking who are willing to get paid to help others with food delivery. The system will track the location of the users as well as the time of the delivery for food shopping to arrive instantly.

After the order is made, the delivery will be finished. The person delivering can only deliver things one by one, it will be an app between Food delivery and Taxi journeys, a mix of both. The orders will be saved for viewing by the users.

A chat is made between 2 users to give precision about location and the product to deliver.

The server will have three main purposes:

- Store the data – Orders, Users and Messages
- Establishing a request between 2 users while one receives the food and the other one brings the food
- Track the location of the users in a dynamic way, using animation

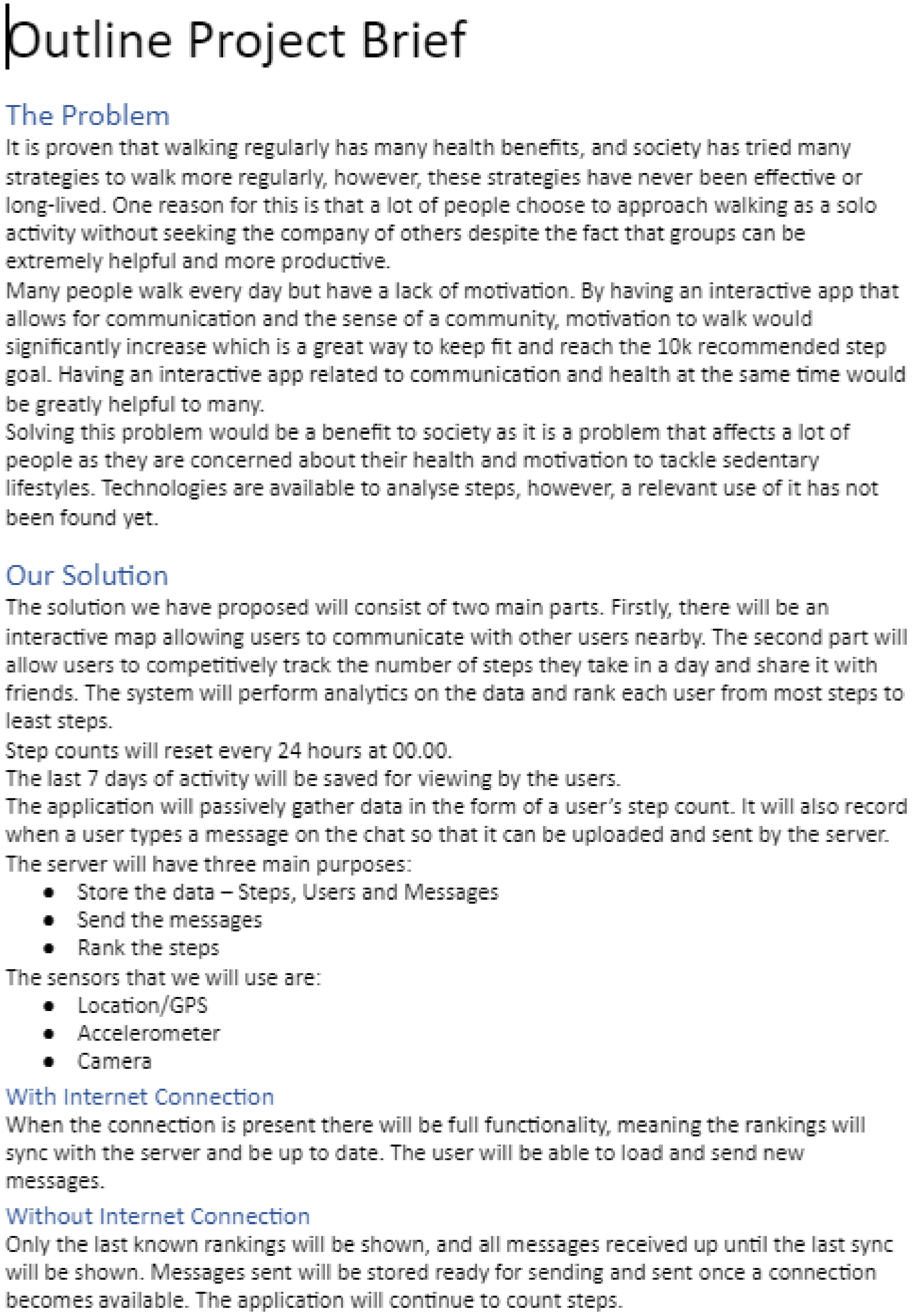
With Internet Connection

When the connection is present there will be full functionality, meaning the orders will sync with the server and be up to date as well as tracking locations and making a request.

Without Internet Connection

All orders received up until the last sync will be shown.

Appendix C - Mat's final idea project brief



Outline Project Brief

The Problem

It is proven that walking regularly has many health benefits, and society has tried many strategies to walk more regularly, however, these strategies have never been effective or long-lived. One reason for this is that a lot of people choose to approach walking as a solo activity without seeking the company of others despite the fact that groups can be extremely helpful and more productive.

Many people walk every day but have a lack of motivation. By having an interactive app that allows for communication and the sense of a community, motivation to walk would significantly increase which is a great way to keep fit and reach the 10k recommended step goal. Having an interactive app related to communication and health at the same time would be greatly helpful to many.

Solving this problem would be a benefit to society as it is a problem that affects a lot of people as they are concerned about their health and motivation to tackle sedentary lifestyles. Technologies are available to analyse steps, however, a relevant use of it has not been found yet.

Our Solution

The solution we have proposed will consist of two main parts. Firstly, there will be an interactive map allowing users to communicate with other users nearby. The second part will allow users to competitively track the number of steps they take in a day and share it with friends. The system will perform analytics on the data and rank each user from most steps to least steps.

Step counts will reset every 24 hours at 00.00.

The last 7 days of activity will be saved for viewing by the users.

The application will passively gather data in the form of a user's step count. It will also record when a user types a message on the chat so that it can be uploaded and sent by the server.

The server will have three main purposes:

- Store the data – Steps, Users and Messages
- Send the messages
- Rank the steps

The sensors that we will use are:

- Location/GPS
- Accelerometer
- Camera

With Internet Connection

When the connection is present there will be full functionality, meaning the rankings will sync with the server and be up to date. The user will be able to load and send new messages.

Without Internet Connection

Only the last known rankings will be shown, and all messages received up until the last sync will be shown. Messages sent will be stored ready for sending and sent once a connection becomes available. The application will continue to count steps.

Appendix D – Other Supporting Documents

Please see a zip folder uploaded with this document for all supporting documents mentioned within this document.

List of documents included:

- Meeting Minutes
- Diagram Files
- Gantt