Real-Time Location Tracking Mobile App

Andrei Toni Niculae
962569

Submitted to Swansea University in fulfilment
of the requirements for the Degree of Bachelor of Science

Swansea University
Prifysgol Abertawe

Department of Computer Science
Swansea University

April 28, 2021

Declaration

This work has not been previously accepted in substance for any degree and is not being con- currently submitted in candidature for any degree.

Signed  (candidate)

Date 27.04.2021

Statement 1

This thesis is the result of my own investigations, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed  (candidate)

Date 27.04.2021

Statement 2

I hereby give my consent for my thesis, if accepted, to be made available for photocopying and inter-library loan, and for the title and summary to be made available to outside organisations.

Signed  (candidate)

Date 27.04.2021

I would like to dedicate this work to my parents who encouraged me and my younger brother to study abroad and who I know are worried about our safety.

Family is most valuable to me!

Abstract

The COVID-19 worldwide pandemic has intensified our acknowledgment that family and our health and that of those around us is the most valuable thing we possess.

Safety of our loved ones and staying connected has become even more important. Being a victim of an accident, kidnapping, or a street crime could seriously affect our physical and mental health; therefore having an easy solution to stay connected and informed on the well-being of our loved ones, is essential to peace of mind.

There are many "fancy" smartphone applications that track users' location. Some are complicated to use, others infringe too much on users' privacy, or limit features based on subscription. Having an alternative easy-to-use app that users can easily enable/disable was important to users and the author of this application.

This dissertation paper discusses an Android mobile application that would provide piece of mind on the safety of those who use it by having the ability to keep track of the connected user's location in real-time. Furthermore, this application offers an integrated "Emergency / Panic" button to send notifications to selected people in one's network, hence providing an added level of security.

Since 72% of smartphone users globally are Android users, it was intuitive to create an Android application that keeps the most people connected by notifying each other in case of emergencies.

The easy to install and use Android app, was thoroughly tested and provides services that include real-time location tracking, detailed information about users, such as: their moving speed, battery level, a location history list, and alert functionality in case of emergency.

Acknowledgements

I would like to use this opportunity to acknowledge and thank my family who always supported me throughout my studies. Thank you Toni, Magda, Radu and Laura.

Table of Contents

Chapter 1

Introduction

In today's world, with all the technological advancements, it is more accessible to be interconnected with family and friends using the help of smartphones and smartwatches. More kids study abroad, away from parents and grandparents retire in warmer climates, away from their kids. Staying connected and having peace-of mind for the safety of loved ones was also intensified by the recent pandemic.

With this project, the author desired to create a tool which used all the available services to gather location data, which would then provide meaningful insights to the user or other connected/paired persons in the network.

The user will be able to get real-time information on the location and safety of an elder member of the family or kids and be alerted if they were in danger.

1.1 Motivations

From 2013, the number of smartphone users has increased by approximately 320%, currently being 3.8 billion users worldwide. This indicates the fact that smartphones have become modern tools possessed by most people in order to make their lives better. [1]

In the UK and Wales, just in 2019, there have been approximately 2 million crimes that involve physical contact (violence, homicide, knife or sharp instruments, and robbery). [2]

Being connected with someone the user cares for will give him peace of mind if he can see in real-time their trip to school, workplace, groceries, or another city, and that nothing unexpected happened on the road.

There are lots of “fancy” smartphone applications that track users’ location. Some are complicated to use, others infringe too much on users’ privacy, or limit features based on subscription. Having an alternative easy-to-use app, that users can easily enable/disable was important to users and the author of this application

1.2 Objective

After analysing the application market, user opinions, and case studies, a certain gap in the tracking applications niche was identified, as well as some serious problems a consistent number of users have raised.

Given the overwhelming number of smartphone users, it was concluded this can be achieved by creating a mobile application. This app's objective is to track and display in real time the following information: a marker on the map representing all the connected users, their moving speed, address, battery level, a location history list.

The user will also have the capability to press a button and send an emergency notification with their address to all connected users.

Hypothetical cases in which this app might be helpful:

1. Being followed by someone suspicious on the street might be a frightening experience. A "PANIC" button would become very handy in order to inform the relatives that the connected user is feeling unsafe and needs help.

2. In daily life, parents can use this app to track their kid's trips to school or simply have the peace of mind to see that their loved ones travelled safely.

3. Speeding on the highway is a dangerous thing that people sometimes do without even realising. The relatives that are connected to the app might notice the dangerous speed and later remind the user to be more careful.

4. In the case that a user has only 2 or 3% battery, the other users will know and can contact them as soon as possible to alert them to charge their phone.

5. Stay connected and share real-life location on the map to meet your "family/friends" that are nearby, using the pinpoint accuracy of the map location address as guidance.

1.3 Accessibility:

The developed app is an accessible tool that can be used by most people. It will give any Android user the same opportunities and methods to check on the health and safety of their close ones and act consequently. The only requirements are to have a 26 or newer version of Android, and a reliable internet connection.

Chapter 2

Background research

Because the concept of this application started from the need to satisfy people's desire to take care of their loved ones, a question that needs to be answered is: what is the origin of this feeling and why do people care about their relatives?

According to "Toward an Evolutionary Psychology of the Family" by Catherine A. Salmon and Todd K. Shackelford [3]: "*Our modern behaviour is the product of our evolutionary response to the pressures of living as a social species, just as the behaviour of other social animals is a product of such pressures*." As a result, this feeling cannot be eliminated.

Also, this application will make users feel safer when they are alone. According to a study: "*91% of cell phone users agree that their phone makes them feel safer because they can always use it to get help*." [4]

Having the "Emergency / panic" button, directly in this application, will provide an added tool to notify selected contacts to get help, thus increasing the safety level of users.

To create a successful mobile health application, it was necessary to analyse the known facts about this area of work.

The research paper "Mobile Health Apps to Facilitate Self-Care: A Qualitative Study of User experiences" [5], provided valuable details, on the outline principles and features that should be developed to comply with the necessities and preferences of most users.

First, the functionality of this application needed to be without any errors. Glitches on the location or any other data that will be processed could cause some non-existent alerts. This could negatively affect the reliability of the app and the user's trust in the accuracy of the information.

Second, the frequency of the notifications that are being sent was an important factor identified in the user's overall satisfaction. This sensitive feature needed to be balanced to achieve its purpose of sending relevant information to the user without annoying them.

Therefore, the application's location tracking service should be discrete by restricting the number of notifications sent. This limitation will also prevent the user from becoming overly obsessed with checking location information of their connected parties in the network.

Third, the idea of “premium features” that are accessible only by users that buy an application was another key aspect that had a negative impact on the overall perception of these types of applications.

This new app is intended to be free of charge with all the options included because as stated in the project’s aims, this app will be accessible by everyone, independent of their financial status.

Additionally, another paper was research that described a possible solution regarding user safety is "Mobile DNUN: Danger Notification and User Navigation" [6]. The main feature of the app was being described as the "Danger Notification page" which "*allows users to send their current location with just a single click of a button to one or more added contacts*". This feature is a practical way of communicating relevant information that might become helpful for someone trying to help endangered people.

Reflecting on this idea of a notification for the new application, some potential issues were identified. Because the notifications would be sent through an email, the user will have to enter and save a valid email address for every contact that needs to be reached. There is also the issue, where some people might not receive email notifications on their phones for various reasons.
Another feature "Finding an object", that saves different locations related to certain objects, for example, the location of a parked car. Because remembering places might be challenging, this method was an efficient solution to keep track of different things.

2.1 Related work

In order to create a qualitative product, the first step was to research into the existing market, identify the strengths that other apps have, and come up with a better alternative aimed at solving their weaknesses.

Analysing a similar application will provide a better understanding of user preferences and how the app design can influence user behaviour. An application that implements similar functionality to this proposed project is called Life360, and described as “*Life360 is a location-based service designed primarily to enable friends and family members to share their location with each other*”. [7]

Anyone who downloads the Life360 app will appreciate its design and features. It has a beautiful and intuitive user interface that is relatively simple to use.

Nonetheless, users, especially teenagers and young adults, have complained about features that in their view encourage the abuse of tracking capabilities in regards to privacy and freedom of movement.

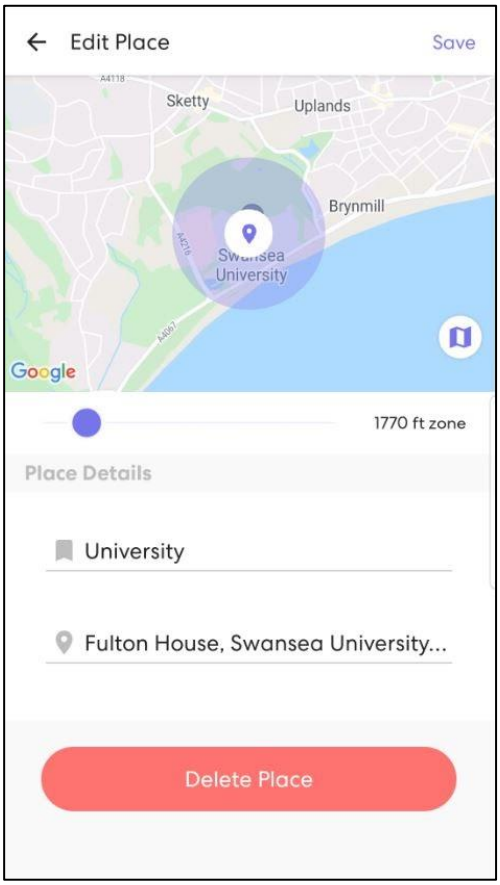
To get a deeper understanding of how successful this app is, reading user reviews on the app store was important. While it has a rating of 4.8 out of 5, the reviews were divided into two categories. The first category was the parents who track their kids, which were very happy with the capabilities this app offers them. The second category was formed of teenagers that were complaining and raising some important issues.

From these reviews, a number of problems that can be improved were identified. On the other hand, the strong points, such as the accuracy of location tracking and the consistent attractive UI, were also noted.

The most important issue in the way the Life360 app was designed, and which the new proposed application should not replicate, was that it encourages excessive surveillance and use of the application in a way that limits the freedom of other users.

The first thing when users start the app, they are asked about their "role". User can select "mom/dad" or kid. This option alone already created a separation between users, which was expressed by some people who find it very unpleasant.

Another example of an unpleasant user experience, was that parents can set up a circle on the map, and when the kid leaves that circle, the parents will be notified.



[A screenshot with a "Circle" from the app Life360]

The author of the new app, wanted to eliminate this “role” separation, hence the new app was created to be used responsibly, maturely, and will not give the authority to specific users to over-control others.

Another important characteristic that influences a user’s perception of any app is the notifications. In Life360, the person that tracks the other is constantly receiving notifications when the other has arrived at a destination or has entered or exited a specific area that was set on the map.

Again, the application discussed in this paper was planned to give the user peace of mind that their loved one were safe, not to control their actions.

Another aspect that bothers the users is the price they have to pay to access premium features such as car crash detection.

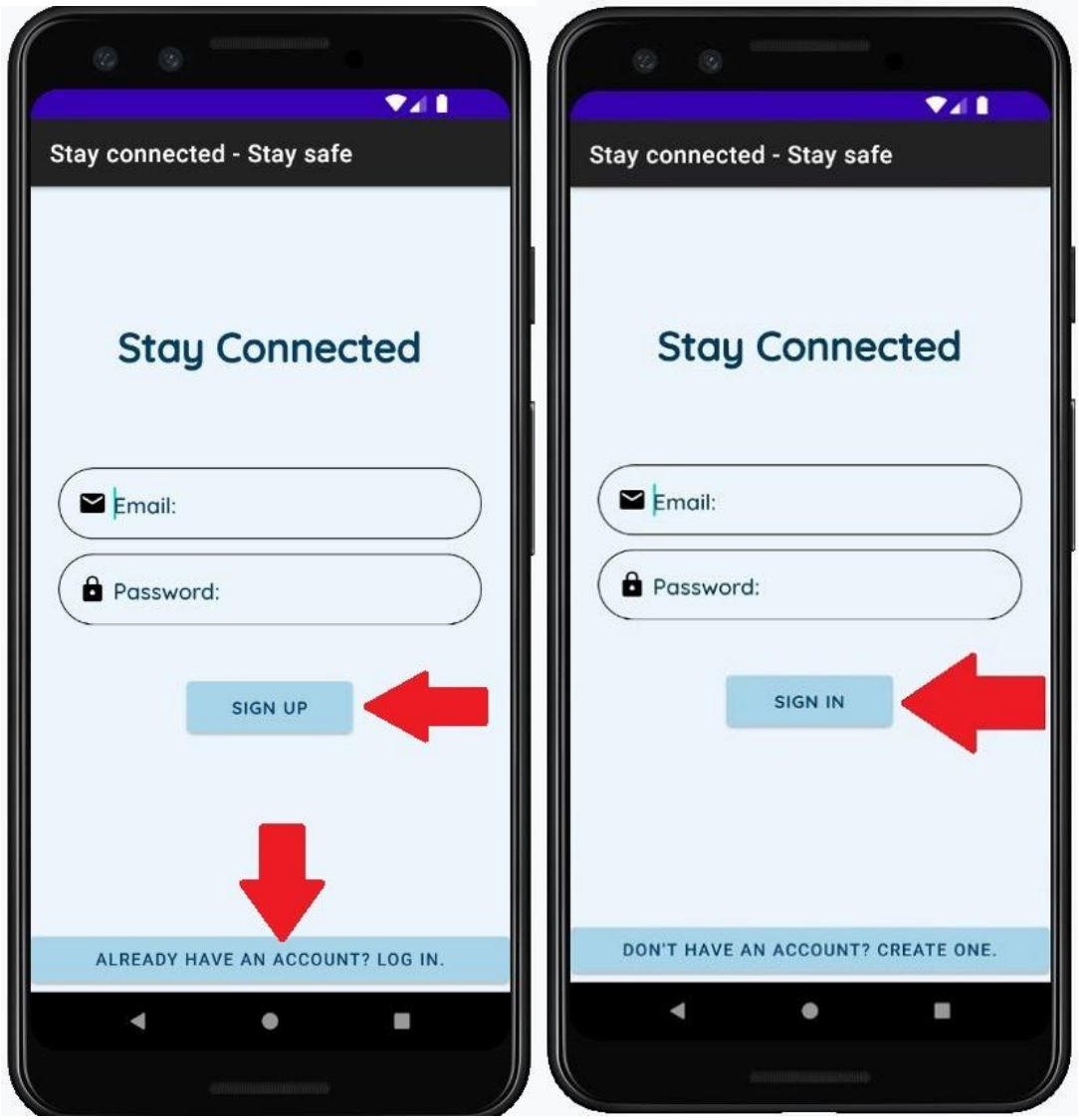
To address all these unpleasant aspects, this new app will not divide users in different “roles” or categories, but rather offer everyone the same capabilities with no premium features that cost money. And instead, will allow users the freedom to enable/disable notification at their discretion.

Chapter 3

Functionality

This chapter provides a quick overview on the ease of use of this new app.

When the user first opens the application, the sign-in screen appears (see the 2 screenshots below).



If an user already has an account created, they can click on the button at the bottom of the screen, and the button above will change from “Sign up” to “Sign in”, allowing them to authenticate with a pre-existing account.

If the user does not have an account yet, they will be asked to enter an email address and password, then press the SIGN-UP button.

Of course, precautions have been taken, and if a user tries to press “Sign up” or “Sign in”, without entering an email or password, a warning message will appear: "**Please enter an email and password**".

If the user is new and signs up, the next page that appears will be as follows:

Once again, if a user inadvertently clicks the NEXT button without entering any name, a Snackbar will appear on the screen that will prompt the user to "**Please enter your name**".

After the user has entered a name, they can click on the “ImageView” to add a profile picture. The app will open the phone's gallery from where the user can choose a photo that will be uploaded to the application.

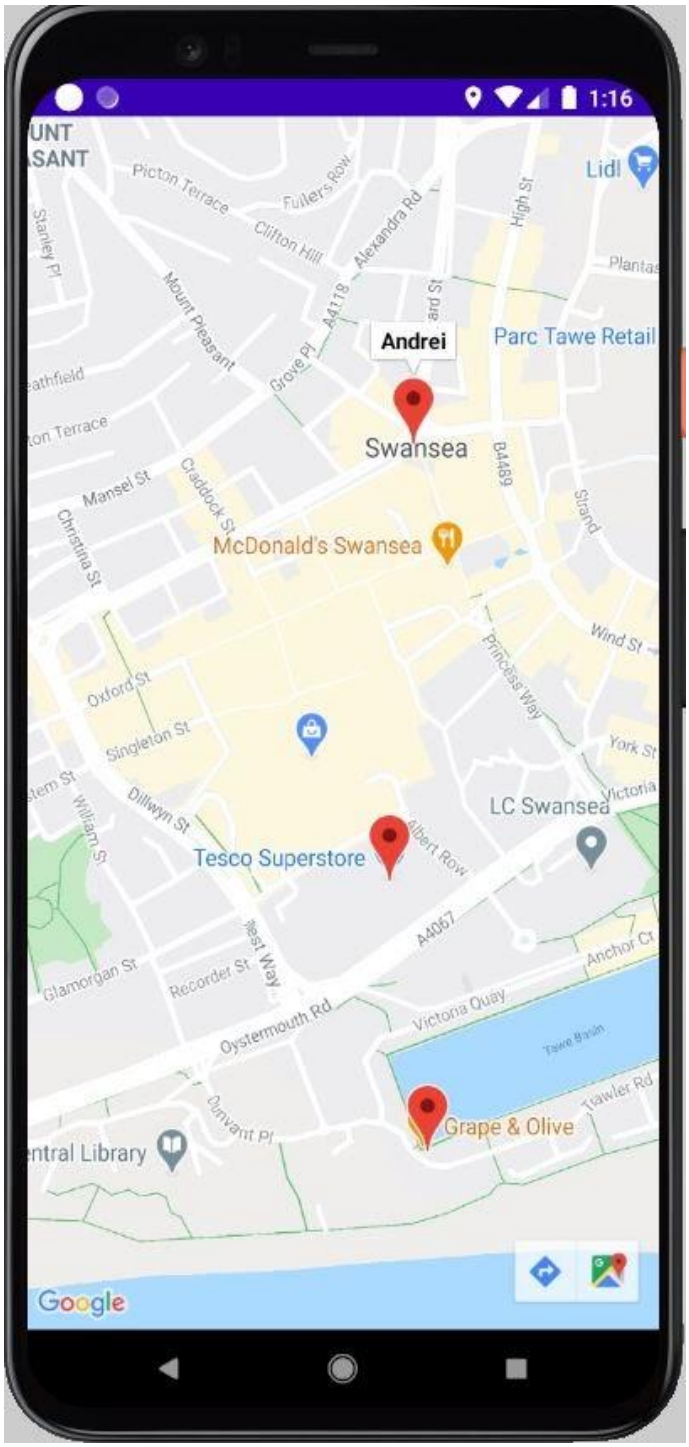
When the NEXT button is pressed, the “Dashboard” menu appears, as follows.

In the screenshot above is the Dashboard menu, that allows users to access several functions.

The “Share Location” green button at the top of the screen will start a service that will send the user's current location and make it visible to the other people connected to the application. A confirmation will be required, and the user will be asked if they agree to share their location, and a notification will remain constant as a reminder that their location is actively used.

If the “Stop Sharing” red button is pressed, the location service will be interrupted. The other users will not receive updates about the location of his phone anymore. Having the ability to easily enable/disable location tracking was an important objective for this app, aimed at returning control to the user themselves.

When the “Show Map” yellow button is pressed, a new activity will start where a map containing markers will be displayed, representing each person using the application (as shown in the screenshot below).



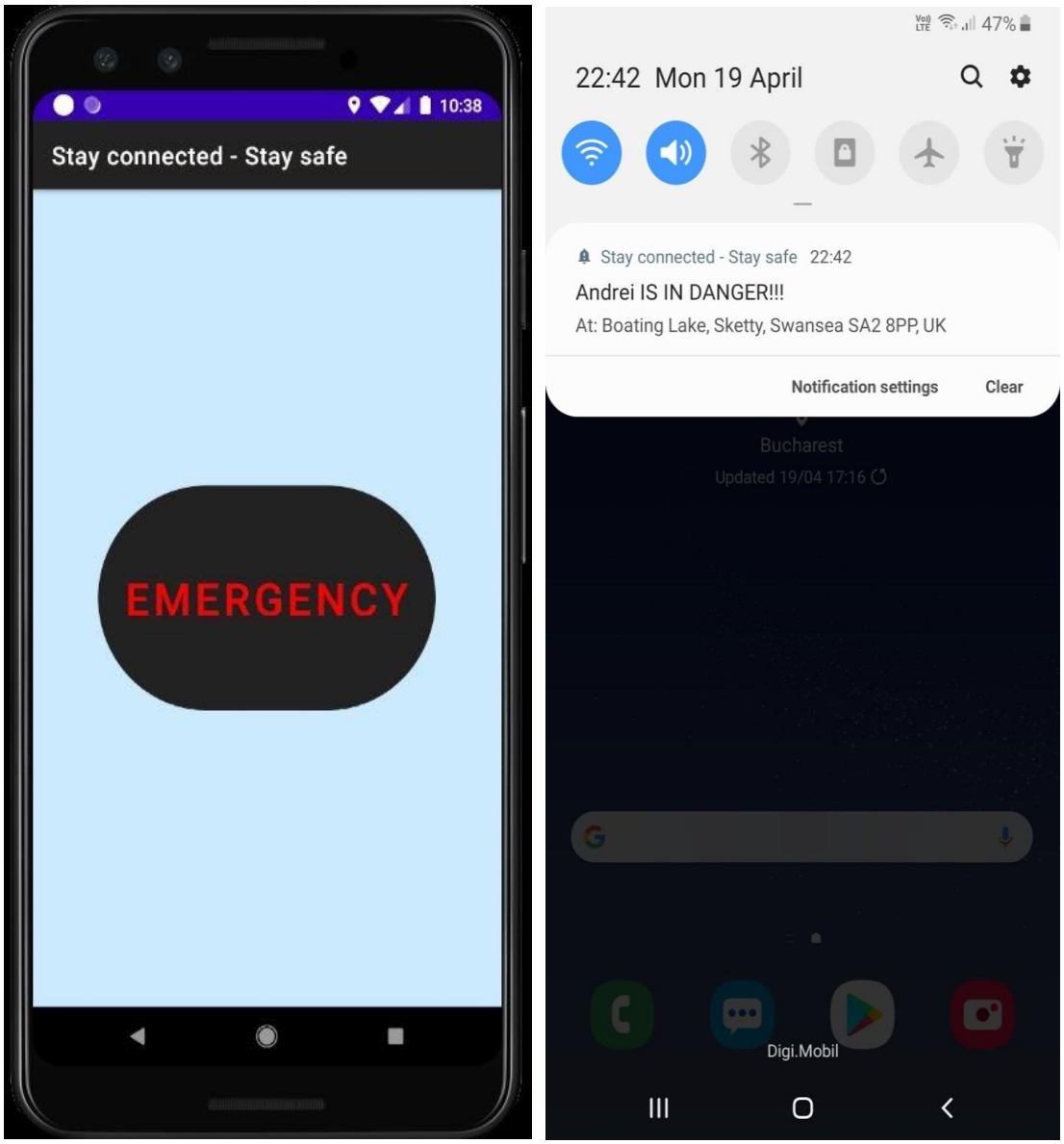
If a marker is clicked, a new activity will open with detailed information about the respective user.

The detailed information about the user selected, includes a picture of the user if they uploaded one when they created their account, their name, current travel speed, the location history, and battery level.

It is important to note that the location of the markers will be updated every 2 (two) seconds depending on the user's new location.

In the screenshot below, the locations were saved at a much shorter interval than in a real situation because the screenshot was taken from an emulator during a test. (22:21:28-22:22:37-22:25:20). In reality, the application will save a new address to the location history list after 10 minutes of staying at the same address.

Back on the “Dashboard” menu, if the user clicks the panic button, a new activity will pop up with only one emergency button in the centre of the screen.



The above screenshot from the right side was taken from a physical device where the notification popped up immediately after the user from the emulator pressed the emergency button.

It is important to mention that the notification will appear on all devices that have this app installed and will tell which user is in danger, regardless of the app running in the background or not running at all.

And finally, back on the “Dashboard” menu, when the “Log out” button is pressed, the user is notified that the location service has stopped (if it was running) and it will be sent to the first login activity.

Chapter 4

Design Concept and Implementation

Taking into consideration that the application was implemented in Android Studio, the app will be available for use only on devices that run the Android operating system.

The overwhelming number of Android users worldwide influenced this decision, taking into account that this operating system is currently holding 72% of the global market share.

Another important aspect was that in order to develop Android applications, knowledge of Java and Kotlin was necessary.

During the development of this app, it was taken into account that users may run the app on devices that have different screen sizes or versions of the operating system.

With this in mind, the layouts and visual components were designed to adapt and have a consistent look regardless of the user's phone.

The final goal was to create a user interface that is clean, intuitive, and easy to use for everyone.

This application was implemented in Android Studio, using the Java programming language, also with the support of other services offered by Google Firebase. Firebase, a Backend-as-a-Service (BaaS), was used for managing the information about each user as well as their notification system.

For example, when a user creates an account by entering an email and a password in the sign-up / sign-in activity, and the login button is pressed, the data entered is validated. Then the Firebase Authentication system will create a new record in the cloud database, representing a new user. When the user is created, a unique user ID (UID) will be automatically generated by Firebase.

Based on this user ID, the app creates a document in the Firebase database and adds two new fields: "email" and "userID". These fields will contain the email address provided by the user and the auto-generated userID.

Structuring the information this way was best because a link is created between the user account itself, and the stored data in the document. Each user has its own document with its own fields, and the app can keep adding new fields in a well-organised way. Every time a new account is created, a new document corresponding to it is also created with the email and userID fields.

In the next activity, when a user enters their name, this will be uploaded in the Firebase cloud document as a new field.

To store the user's profile picture, the most powerful and simple way was to use Cloud Storage for Firebase. This storage service allows secure, fast, and reliable uploads and downloads of images.

The problem with Firebase Storage was that the files were not linked to a specific user account or document in Firestore, therefore a solution was needed to make this connection between the user to whom the picture belongs, and the object in Firebase Storage.

For this problem, the name entered by the user will be used as part of the path to the picture in the cloud storage.

This is how the reference is set:

```
String name = edt_name.getText().toString();
StorageReference myReference = storageReference.child("images/" + name)
```

And this is the result: gs://stayconnectedv3.appspot.com/images/Andrei Toni Niculae

If the user tries to upload an image before entering their name, a StorageReference will be created that will not contain a reference to the user, meaning the image will be uploaded with the wrong path.

To solve the above potential problem, in the function setOnClickListener of the ImageView, an "if statement" was added that verifies if the name string is empty or not. If it is, a Toast will be displayed with the text: "Please enter your name first", if not, it will open the gallery for the user to upload a picture.

After the user selects a picture, it will automatically be saved in the cloud storage and a Toast will notify the user if his image was uploaded successfully or if it failed. After many tests, it was concluded that the user will be notified of the success in approximately 1 second or less, therefore a ProgressBar would not be needed to show the user the upload progress.

For the implementation of the main functionality of this application, the one that provides real-time location updates for each user, several classes were created.

One of them is the User class, which defines many attributes that a user has, such as:

- userID,
- email,
- firstName,
- latitude,
- longitude,
- speed,
- a Marker,
- an ArrayList for the location history, as well as getters and setters for them.

This class contains the functions setupMarker() and updateMarker(), which will set and update the marker at new coordinates retrieved with the getter functions for the latitude and longitude attributes.

The User class also overrides the equals method. This method returns true if two User objects have the same user ID.

Another key function that will be used in another class is the 'update(User other)' method. This function overwrites the values of the instance variables of the current user object to the values retrieved from the 'other' user.

One of the most important functions in this class is the userFromDoc(DocumentSnapshot doc) method.

This function creates a new User instance and sets all the attributes of a user to the values retrieved from a document in the Firestore database.

By using this method, the app gains local access to the information in the cloud.

Another class that accompanies the User, is the UserList class. When an instance of this class is created, the constructor creates an ArrayList of type User called 'users'. This class has a method, `public boolean contains(User u)`, that checks if the user passed as parameter exists in the 'users' ArrayList, by comparing the userID. Another function worth mentioning in this class is the `public void update(User u).`

This method iterates through all the users in the ArrayList. If one of the users in the list has the same ID as the user passed as a parameter, that user will call the update function from the class User described above. In other words, the user in the list will update its latitude, longitude, speed, address, first name, location history, and battery level.

The main class that will supply the app with user information is the LocationService class. This class implements a foreground location service that will provide continuous location updates using the fused location provider API.

This API will send latitude and longitude coordinates having the fastest update interval set at 2000 milliseconds, and the LocationRequest has the priority set to HIGH-ACCURACY.
The latitude and longitude coordinates are then used in the getAddress function, which uses a Geocoder to transform them into an address.

One of the challenges that came up during the development of this project was to create the location history list. So far, the app receives new coordinates every 2 seconds, so how is decided which address to save in the list and which to omit?

Which locations can be considered suitable and which can be left apart? The goal was to only save the relevant locations where the user has been during the day and discard the others.

After carefully taking into consideration the possible factors, it appeared that time was the most relevant one. Therefore, the location history list will save an address only if it did not change for the last 10 minutes.

To implement this, the solution was to introduce a 'timer' variable and an if statement (the 'loop' already exists because the app executes the service every 2 seconds) that checks if the timer is greater than the period of time set in seconds, times two - because the service executes every 2 seconds, the increment will be +2 (10 minutes = 600 seconds x 2 = 1200 seconds). When the timer is greater than 1200, it is reset at 0 and the address is added to the location history list.

After implementing this solution, a problem still remained. If the user spends a couple of hours at the same address, the service will add that location every 10 minutes. It is preferable not to have a location history list full of duplicated addresses. Therefore, the solution that was implemented to solve the above issue, was to add one more if statement. It will check if the last element in the list is different from the current address, if it is, add it to the list.

A factor that a user will want to know is when other users were at those addresses. A simple solution to this was to append the current date and time at the end of the address string.

A problem that appeared by adding the time was that the if statement that tests if the last address is equal to the current one always tested false because they were ending with the time string which was different every time. Because of this, if the user stays all day at the same address, it will now be duplicated every 10 minutes, which is not ideal.
Adding the following statement solved the problem above:

```
// if the first 25 characters in the previous address are NOT the same as
// the first 25 characters in the current address, add the current address to the favourite
addresses list.
if (!favouriteAddressesList.get(favouriteAddressesList.size() - 1).regionMatches(false, 0,
currentAddress, 0, 25))
{
favouriteAddressesList.add(currentAddress + " at " + date);
}
```

Testing showed that comparing the first 25 characters for equality was a viable solution because the address contains not just the street name and number, but also the city, postcode, and country, resulting in a long enough string. Errors could happen if all the above are extremely short, but it is unlikely.

Here are some addresses in the location history list (Note that the timer was set to less than 10 minutes when the app was tested on the emulator):

"29 Albert Row, Swansea SA1 3RA, UK at Thu 8 Apr 2021, 13:34:07"
"2 B4290, Swansea SA1 5AE, UK at Thu 8 Apr 2021, 13:35:00"
"5 Cadwaladr Cir, Mayhill, Swansea SA1 6UB, UK at Thu 8 Apr 2021, 13:35:28"
"Swansea University, Sketty, Swansea SA2 8PP, UK at Thu 8 Apr 2021, 13:36:37"

After all the necessary fields were retrieved, the information was added to Firestore.

This information includes latitude, longitude, address, the location history list, and the moving speed of the user. This service is also responsible for updating the user's markers on the map.

The location service also executes the method 'getAllUsersFromDB' from the DashboardActivity.

At the beginning of this class, a public static UserList allUsersList is declared and initialized which will locally contain all the users in the database.

The 'getAllUsersFromDB' method connects to Firestore and as it goes over each document(user) in the database it creates a new User object. It then calls the function userFromDoc, which populates the attributes of the user(latitude, longitude, etc...). After that, it checks if the 'allUsersList' contains the newly created user from the document, and if it does, it updates its values using the method 'update' declared in the UserList class. If the user created from the document is not in the list, it adds it.

```
protected static void getAllUsersFromDB() {

    FirebaseFirestore db = FirebaseFirestore.getInstance();

    db.collection( collectionPath: "users")
            .get()
            .addOnCompleteListener(task -> {
                if (task.isSuccessful()) {
                    for (QueryDocumentSnapshot document : task.getResult()) {

                        User user = User.userFromDoc(document);
                        if (allUsersList.contains(user)) {
                            allUsersList.update(user);
                        } else {
                            allUsersList.add(user);
                        }
                    }
                } else {
                    Log.w( tag: "GET_OTHER_USERS",  msg: "Error getting other users.", task.getException());
                }
            });
}
```

[Fig. The implementation of getAllUsersFromDB method]

The Dashboard class is responsible for asking for the user's permission to access its location and starting or stopping the location service.

4.1 GDPR compliance and app security

The General Data Protection Regulation (GDPR) is a significant and globally influential data and privacy law from the European Union. The GDPR applies to mobile applications that collect and process the personal data of EU citizens. It does not matter if the app is operated from outside of the EU. The GDPR will still apply.

Because the application uses personal data such as name, email, and real-time location, it is extremely important to account for and respect the privacy of the users.

To this degree, the application only collects and processes relevant information about the user. It also addresses the "purpose and storage limitation" principles. To comply with them, the application collects data to achieve a specific purpose, and as long as it is necessary.

For example, the location history list data is deleted and replaced with new data when the user restarts the location tracking updates. The old information is deleted and cannot be used anywhere else again.

The application allows for the creation of a new user only if they enter a password longer than 5 characters. This will secure the user accounts from being accessed by unauthorized persons. Since this application was designed for family use, presumably nobody will have a reason to hack into a family member's account. If in the future this application will have a scope larger than a family, this security feature will be very useful.

All the information is stored in the cloud database that is offered and secured by Google. No one can access the user's information without being the administrator of the database.

Chapter 5

Testing

After the implementation phase of a feature ended, the whole application was tested to verify it works as expected.

To test how the markers were updated on the map when a user changed their location, a physical device and an emulator were used.

The test started by changing the location of the emulator to different countries, cities or streets, to verify how the marker for that user will update on the physical phone.

This testing method has proven to be successful, but applying only this approach does not offer complete coverage of all the possible situations.

For example, the test could not show how the app keeps updating when the user uses it in the background, or if the marker updates smoothly and precisely on the map.

Therefore, testing the new app in real-life situations was necessary. The new app was installed on a test phone and a TestUser started walking around the city. This was done to confirm that on the emulator and in the Firebase console the data changed in real-time. Through constant communication with the TestUser, the accuracy of the marker was established. The application displayed the marker on the map at the exact location where the TestUser was with the test phone.

Another method to test the functionality of the app was to take the test phone with the new app installed on it and drive around town. The TestUser would intentionally stay for longer periods at the same address to check if that location was stored in the location history list correctly. The TestUser communicated that the speed displayed by the car's speedometer was the same as the one displayed in the app, confirming the speed parameter worked correctly.

Testing different scenarios was important, and it was noted that after the initial presentation of the demo app in December, a major issue with the location tracking was found. If the application was used in the background, the location updates will continue to work for an amount of time and then stop. One time the location updates stopped when the user stood still for more than 30 seconds, at a stoplight.

The first thought for this issue, was that the LocationService class was not implemented correctly, but after many tries of debugging, it resulted in the problem coming from other classes that were linked with the service.

In conclusion, the testing part of developing this new application has been very interesting and instructive, especially going outside the classroom and checking in different real-life scenarios how different parameters were changing in the app while moving.
A lesson learned was that in order to have a stable and reliable application of this kind, extensive testing is required, both at the computer and in real-life scenarios.

Chapter 6

Evaluation and Analysis of The Work and Results Produced

Reflecting on the timeline of this project, most of the planned goals were achieved and the results are greatly satisfying.

The new app is a very efficient and accurate real-time location tracking app, that based on the research done, aligns with the needs of the majority of users.

During the development period, when implementing a new feature, all the possible solutions were analysed and the best one was chosen in terms of efficiency and optimization.

After thorough testing in real-life conditions, the application is stable and all the bugs that were found have been taken care of.

This application achieved its main functionality purposes, and with future work on the design and other functionality aspects it can further enhance the user experience. Taking into consideration the time constraint, the app is of high quality from an implementation point of view and is accomplishing the user needs and the authors objective.

Looking back at the planning and the milestones that were set in the Initial Document, the original plan was followed. As proposed, the Spiral Model was used because this technique allowed to divide the project into smaller parts and to develop progressively more complete versions of the software.

Being a large project that involved many risks and changes, the flexibility to work in repeated phases where problems are analysed and handled was very advantageous.

At the project presentation in December, the app was in an initial version that went through many changes until its final form. In that initial version, users were created manually from Android Studio, and the authentication with Firebase did not exist. The user management was not implemented yet, and neither the notification system. The user interface was also at an early stage.

Looking back at the project plan and the work schedule, the planned duration of some features ended up differently. For example, one week was set for implementing the login with Firebase and three weeks for the map activity.

The Spiral Model soon proved a disadvantage, that it was hard to define objective, verifiable milestones. Setting one week for the login and three weeks for the map was a bit naive because this project has so many interconnected components, making it hard to have something working without the other. All these have been taken as learning opportunities as part of growing.

Compliance with the proposed specifications.

The functionality of the final version of the application followed the one proposed in the initial document.

- The user is able to log in to the application and see the connected users on the map.

- The app tracks the user's location on a map as well as the places they visit and stores them for later access.

- When two or more users connect through the app, they will be able to track all the information the other provides.

- Alert functionality: when facing an emergency, the person in danger will only have to press a panic button and all the paired users will be instantly alerted to connect and provide immediate help.

- The paired user will be able to see the battery level of the other user. This feature is beneficial because in the case that a user has only 2 or 3% battery, the other user will know and can contact them as soon as possible to alert them to charge their phone.

There was one proposed initial feature that unfortunately was not implemented in this version of the app. This was considered a risk in the Initial Document, where it was identified as a potential too ambitious goal, given the timelines.

The initial proposed feature that had not yet been implemented was to connect the application with a smartwatch to get the heartbeat of the user. The main factor that led to the decision to stop implementing this feature was the lack of documentation and resources that were found online, and of course, time constraints.

The documentation available about the Google Fit API was very general and did not offer concrete solutions. This resulted in an unsuccessful link that was needed between a phone and a smartwatch. Also finding code examples or projects that implemented this API proved to be extremely hard.

While this ambitious additional feature was not implemented, it has no negative impact on the current launch of the new app, since the main goal for a real-time accurate location tracking has been successfully implemented.

Chapter 7

Future Work

This application opens many future opportunities to build upon the existing structure.

This software was intended for family use, but because it was designed optimally and efficiently, it can be also used for large groups of people.

With minor changes to the interface of the app, it can be adapted and easily used for big companies that want to track their cars, trucks, or employees.

The reason why it can be adapted easily is that the internal logic and data flow of the app has been carefully thought and can handle dozens, maybe hundreds of users simultaneously.

Because the location tracking is so precise, it can also be adapted to be a COVID tracker app. To achieve this, we can simply add another attribute to the user account called “infection status”. If a user is infected, all other users that are at the same address at the same time as the infected person will receive a warning notification.

A future perspective of the application still remains its compatibility with smartwatches. Having this app installed on a smartwatch would make the goals of feeling safer and sharing relevant information about the user to certain people easier to achieve since a watch is more portable and convenient than a smartphone. In addition, this feature will be able to provide a lot more relevant information about the user’s health, such as heart rate, blood pressure, or oxygen level.

Chapter 8

Conclusion

This project was undertaken to plan, develop and evaluate an Android application that more easily connects several users to a common platform that offers them a range of services.

Having an alternative easy-to-use app, that users can easily enable/disable was important to users and the author of this application. Furthermore, this application offers an integrated “Emergency / Panic” button to send notifications to selected people in one’s network, hence providing an added level of security.

The easy to install and use Android app, was thoroughly tested and provides services that include real-time location tracking, detailed information about users, such as: their moving speed, battery level, a location history list, and alert functionality in case of emergency.

After clearly establishing the context and necessity of this application, the motivation of the project has been presented. This motivation comes from the will to create a safer environment for everyone and to be present in the life of our loved ones.

After examining the tools people currently have to accomplish this goal, as well as feedback from them and user studies, it was clear what features needed to be implemented. These features are exclusively focused on addressing the needs and respecting the privacy of the users.

The resulting app described in this paper provides resources that increase the safety of the users by offering them essential information, as well as alert functionality.

Regarding the future of this application, it opens a road for many possible new features. As stated earlier, a major improvement could be to integrate a smartwatch and read the heart rate from the users. With this supplementary information, the app can send alerts if a person’s heart rate increased drastically. After testing the final version of the application, it has proven to be a successful and stable application that meets the requirements of most users. It demonstrated to achieve all the presented goals in an optimal, fast and reliable way.

Bibliography

[1] Number of smartphone users worldwide from 2016 to 2023. [Online]. Available: https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[2] 2019/20 Crime Statistics Revealed. [Online]. Available: https://www.farsight.co.uk/blog/2019-20-crime-statistics-revealed/

[3] Toward an Evolutionary Psychology of the Family. [Online]. Available: https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780195396690.001.0001/oxfordhb-9780195396690-e-001

[4] Part Three: Adult attitudes towards the cell phone. [Online]. Available: https://www.pewresearch.org/internet/2010/09/02/part-three-adult-attitudes-towards-the-cell-phone/

[5] Mobile Health Apps to Facilitate Self-Care: A Qualitative Study of User experiences Anderson K, Burford O, Emmerton L (2016) Mobile Health Apps to Facilitate Self-Care: A Qualitative Study of User Experiences. PLOS ONE 11(5): e0156164. [Online]. Available: https://doi.org/10.1371/journal.pone.0156164 Date accessed 27/10/2020

[6] K. R. Nalla and H. El-Ocla, "Mobile DNUN: Danger Notification and User Navigation," in IT Professional, vol. 19, no. 1, pp. 21-26, Jan.-Feb. 2017, doi: 10.1109/MITP.2017.12. [Online]. Available: https://ieeexplore.ieee.org/document/7839862

[7] Life360 [Online]. Available: https://en.wikipedia.org/wiki/Life360) Date accessed 15/11/2020