Meet in The Middle

Conor Baker
C17320916

Submitted in partial fulfilment of the requirements
for the degree of

BSc in Business Computing

Technological University Dublin (City Campus)

May 2021

Supervisor : Neil O’Connor

Declaration

This is an original work. All references and assistance are acknowledged.

Signed: Conor Baker

Date: 29/04/2021

Acknowledgements

I would like to take this opportunity to thank a few people who have gone out of their way to make sure my 4 years of college have been the best they possibly could be.

Firstly, I would like to thank my parents for giving me this opportunity to attend third level education and study something that I had such a great interest in. Not only did they provide me with the opportunity to attend they also supported me through every step of the way. Between putting up with me being grumpy in the morning after I had spent all night being up coding, to making sure every day that I was on top of my deadlines and going to lectures, they helped motivate me throughout the thick and thin and I could not have gotten this far without them.

I would next like to give a big thank you to all my lectures throughout my time in DIT and TUD. Each one of them carried their own personality with different approaches to teaching and made each lecture and module a unique and enjoyable experience. I feel the structure that they set out from the start has enabled me to exceed any expectations I had in the start of first year. After studying Spanish for most of the secondary school and not knowing a word of it after I was done, I was worried coming into this course that I would struggle to pick up coding, but I feel the lecturers and the college approached it perfectly and without them I also would not be where I am today.

Finally, I would like to thank Neil my supervisor. From the start Neil has been nothing but helpful and encouraging, I can not express enough how much I appreciated his help throughout my final year project. It was clear to me that Neil was not only interested in my project but excited to be able to help me with it. On more than one occasion he sat down with me, and together we brain stormed ideas for my project, and he even would send me emails during the week with ideas to change or implement. From the start Neil encouraged me to push my boundaries and code my project in a language we had not covered in lectures, this was something I had never considered before but I am extremely grateful I did. Neil made sure I was on top of my project with frequent meetings and progress updates and I never felt like I could not reach out to him about anything. Without his help and encouragement, I do not think my project would be half of what it is now and for that I am very grateful.

Abstract

Meet in the middle is a compromise to all other parental tracking apps. The app allows parents to locate their children if needed, while still allowing the children to maintain a level of privacy, this is something that no other parental tracking apps offer.

The app will continually track all users in the background, storing their location privately every fifteen minutes on a database. If a parent is unable to contact their child, they can request that child's location in the app. This will send a SMS message to the child's phone and a notification will appear on their app. The child now has 15 minutes to respond to the parent or else their phone will send back their current location to the parent.

The app over time will also learn commonly visited places that the members of the family go to. These places will be suggested to the parent where they can categorize them as places they are happy their child is safe in, such as school or a friend's house, or places they do not want their child hanging around such as parks. The app will then display these places and if anyone in the family is at one of the places their name will be attached to that place. A notification will also be sent out to the parent if a child stays in one of the bad locations for too long.

The app also allows for a graph to be made up of any users four most visited places over the last 7 days.

The aim of Meet in the Middle is to allow children to maintain their privacy and not to be continually tracked by their parents while also allowing the parents to locate their child if they are unresponsive.

Table of Contents

Introduction

The aim of my project was to supply families with a better alternative to all current family tracking apps. I was inspired to make my app after a fight broke out in my family after my mum jokingly asked my eighteen-year-old sister to download Life360™ , The current leading family tracking app. My sister was offended and said she did not what her every move to be tracked and to throw away all her privacy. This is where I got my idea for Meet in the Middle. I asked my sister would she download an app that only shares her location sometimes, and she is given the chance to deny that request of her location. After she said yes, I then pitched the idea to my mum except from the parents' perspective. I asked her would she use an app that if her child is not answering their phone and she is worried about them she has the chance to request the child's location. If the child were just ignoring her before, they would have to deny the request and the app would notify her, she would then know then that they were ignoring her attempts to reach them. However, if something had happened to the child and they were not just ignoring her attempts to contact them, after fifteen minutes the app would notify her of her child's current location so she can check on them. With both my mum and sister agreeing they would use the app and checking online to make sure nothing similar had already been made I knew I had found a middle ground to what is a common fight between parents and teenagers in todays world of technology. This is where Meet in the Middle was born.

Objectives of Meet in the Middle

The main objective of my app is to facilitate parents with a platform to look after their children if needed too while also allowing children to maintain a level of privacy. This was a very thin line and while it was hard to find a balance, after much trial-and-error Meet in the Middle can now fulfill that need. The app will bring families closer together through trust and understanding without overstepping any boundaries unlike other more intrusive family tracking apps.

When using the app Children and Parents are be able to:

- Login or register using an email and password.
- Join a family using a code created by one of the Parents.
- Set up person details such as a name, phone number and a user icon.
- View all members of the family.
- View all assigned “Safe Places” and see if anyone is at them.
- View all assigned “No Go Zones” and see if anyone is at them.
- View any member of the family’s graph of time spent over the last week.
- Request a family members location.
- Accept or deny a location request.
- View location of a family member on a map if they accept or do not acknowledge the request in time.

When using the app Parents can but children cannot:

- Create a family and share a unique code.
- Approve suggested frequently visited places and categorize them as good or bad.
- Receive a notification if a child is in a “No Go Zone” for too long.

The App will provide the following:

- Store logged in user in apps cache, so they do not have to login each time.
- Store all users’ locations every 15 minutes.
- Calculate based off users’ locations, places commonly visited and suggest them.
- See if any users are in a saved place and display them under the place.
- Notify parents if a child is in a “No Go Zone”.

Business Case for Project

In today's world of technology privacy is hard to come by. While the importance of keeping your children safe and being able to know where they are is important, it should not be abused. Alternative family tracking apps start off with good intentions but soon often turn family's toxic against each other with fights breaking out over each other's every single move being judged. This is where Meet in the Middle comes in to offer a solution.

After coming up with this idea after a fight between my sister and mother, I conducted some market research and was surprised to find that there are no similar apps to my one. All other apps only offer constant tracking of users that are displayed on a map. This might work for families with younger children, but preteens and teenagers often would not consent to using such an app with their family as they are given little to no privacy.

Meet in the Middle it offers a perfect middle ground. Now, while locations are still constantly being tracked, they are not being displayed without approval or lack of acknowledgement of the request. This gives the child the privacy they are looking for while still allowing parents a chance to find out where they are.

Meet in the Middle also suggest commonly visited locations, this feature benefits both the children and the parents. Once a place is deemed "safe" there is no need for a parent to request the child's location as it is both displayed on the home screen and they are notified if they try request it. To the child helps build another level of privacy as they do not need to reply to their parents while in these zones which could consist of friends houses or coffee shops for example. To the parent they are able to receive instant gratification on where their child is without the need of waiting. This is similar to locations deemed "bad". The child is allowed to visit these places but if the app picks up on the child being there for too long it will notify the parent.

Finally Meet in the Middle offers a week summary of where the child has spent the most amount of time. This also builds on the level of privacy this app offers to children as while their locations are being stored, it is only being displayed later on and not instantly. This gives the parents an insight into where their child is spending the most amount of time, while still allowing that child to spend that time in those places without the parent knowing straight away.

To conclude, I feel like this app has potential to tap into, and revolutionize, the stale and un-innovative market of family tracking apps.

Technologies Used in Meet in the Middle

Dart is a programming language designed for client development, such as for the web and mobile apps. It is developed by Google and can also be used to build server and desktop applications. I chose to do my project in Dart enabled me to use flutter and it is very stable and it can be used to build production quality real-time applications.

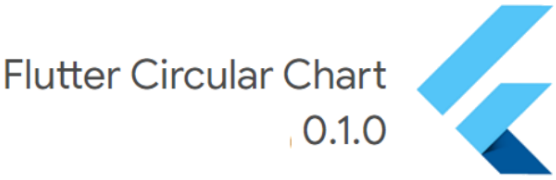
Flutter is an open-source UI software development kit created by Google. It is used to develop applications for Android and iOS from a single codebase. I wanted to use flutter for my project as it allows you to do so much stuff with your apps that are not available on other platforms because of its own powerful separate rendering engine and enormous UI customization potential.

Android Studio is the official integrated development environment for Google's Android operating system and designed specifically for Android development. I chose to use android studio as my IDE as it supports faster deployment of builds with its hot reload feature which applies code changes without restarting the app or building a new APK.

Firebase is a platform developed by Google for creating mobile applications and it is their flagship offering for app development. I chose firebase for my project as flutter was designed to work well with it and its very simple to setup and eliminates the need for a separate backend application.

Flutter WorkManager is a wrapper around Android's WorkManager, effectively enabling headless execution of Dart code in the background. I used WorkManager to run all background tasks in my app. It runs a periodic task that will collect the user's location every 15 minutes.

Google Maps for flutter is a Flutter plugin that provides a Google Maps widget. I used this in my app to display the location of the children after the parent's location request has timed out.



Flutter Circular Chart is a plugin used for creating animated circular chart widgets with Flutter, I used this in my project to create and design the countdown timer on the parents' app showing them how long the child has left to respond.

Paint.net is a graphics editor program for Microsoft Windows, developed on the .NET Framework. It is a free alternative to photoshop, and I used it to design and edit all graphics and icons in the project and this documentation.

GitHub is a provider of Internet hosting for software development and version control using Git. It offers the distributed version control and source code management functionality of Git. I decided to use GitHub for my version control as it is the most commonly used and is also built into Android Studio

Business Rules

When designing my application, I implemented the following business rules.

- User must be logged into the system.
- Users are only allowed to have one account per email.
- User must allow location to be tracked in the background.
- Users must allow app to send SMS messages.
- Users must allow app to send local notifications.
- Users are only allowed to join one family at a time.

Business Actors

In my application there are two types of business actors, Parents and Children.

Children Actor

When signing up a child must provide a unique email and a password. This will allow them to access the app and the system will remember their details and keep them logged in. From here they can join a family and then interact with all users in that family, such as requesting locations, accepting or denying requests and viewing a summary of a person's locations over the last 7 days.

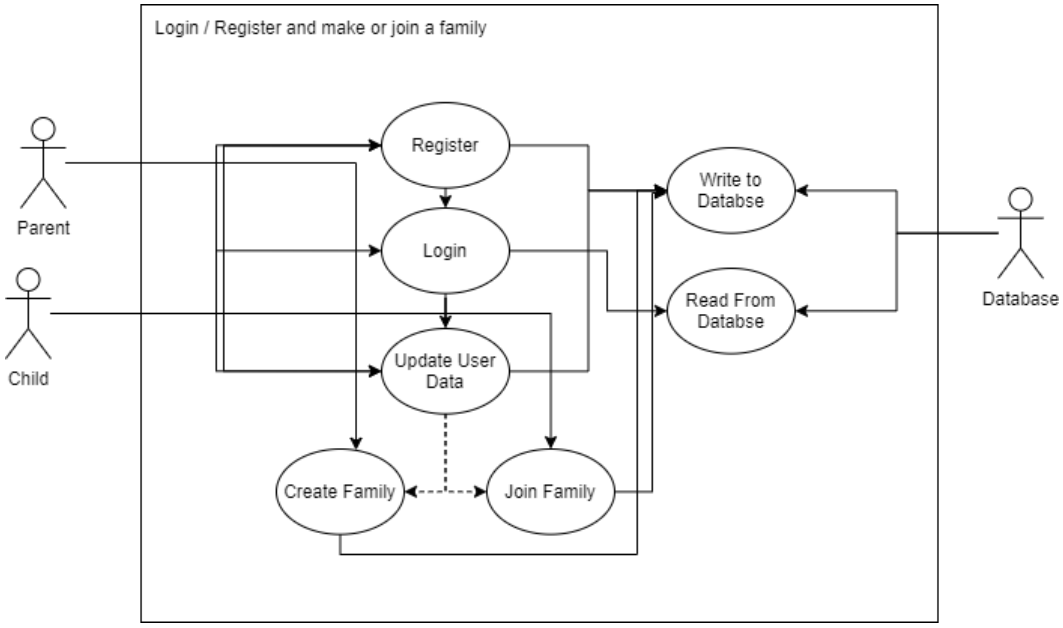
Parent Actor

While a parent can do everything a child can, they also have access to some features that the children do not. For example, after signing up a parent must create a family. This will generate a unique code for the parent that they can share with their children to join. Parents are also able to approve locations suggested by the application and categorize them. Finally, the parent will also receive notifications if any of the child are in a "No go zone" for too long.

Use Case Analysis

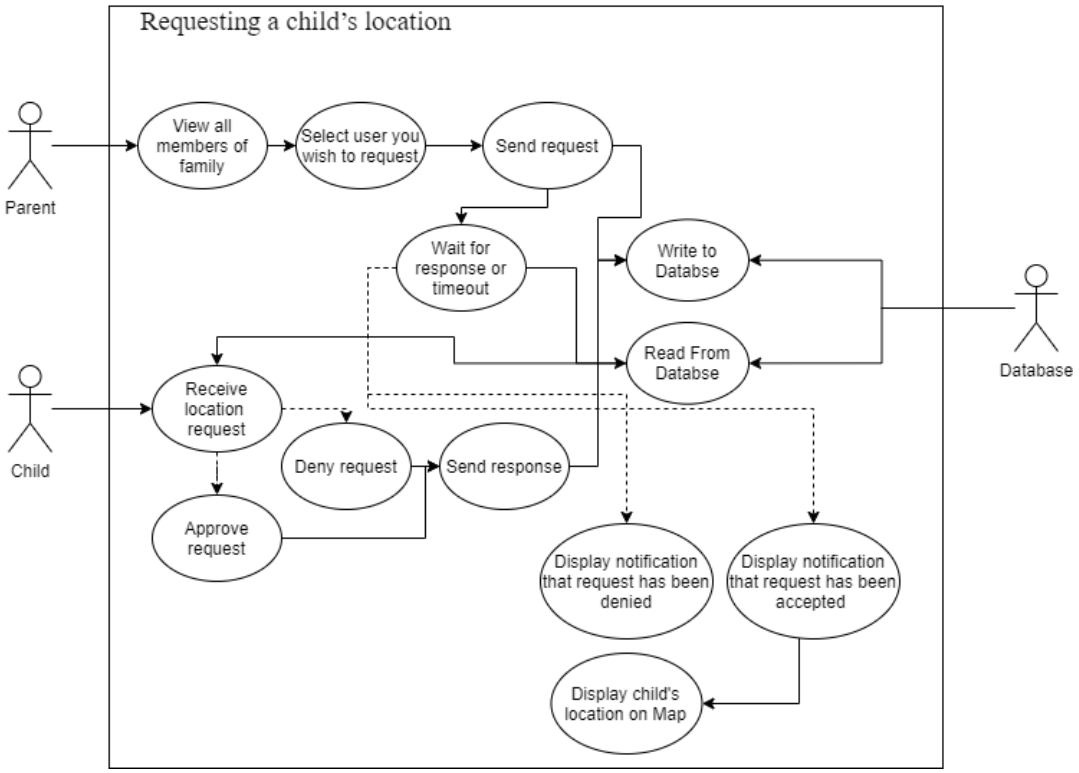
Register, login and create or join a family

This use case diagram displays the main functionality the Children and Parent access once they Register and Login.



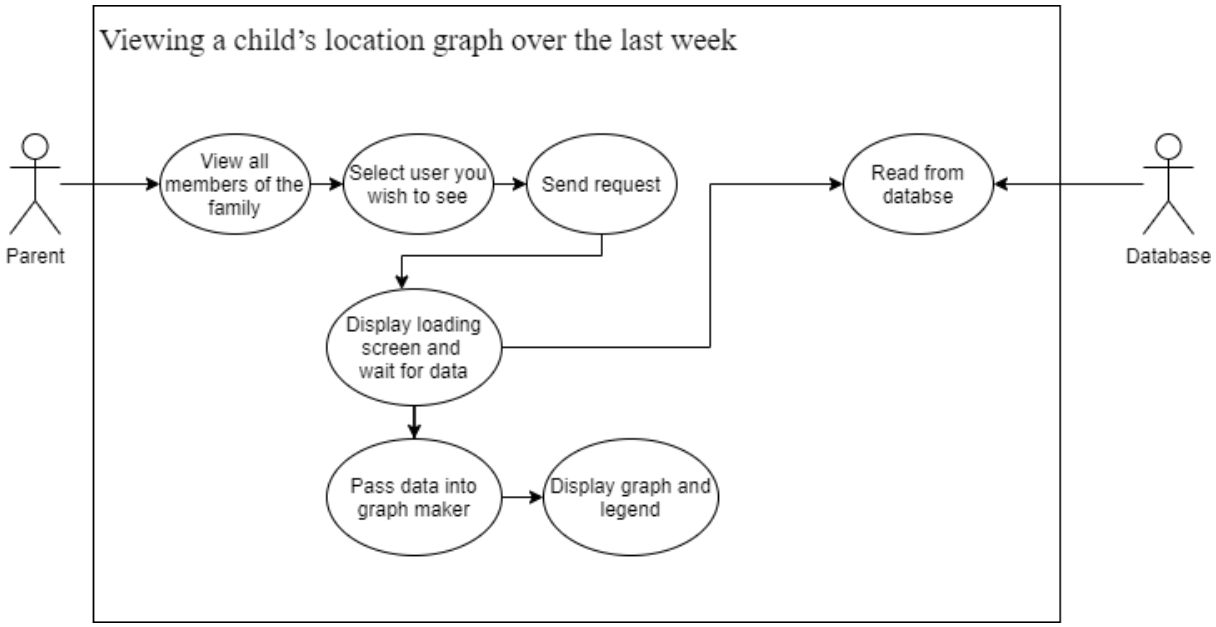
Requesting a child's location

This use case diagram displays the main functionality the Parent and the Child have access to when a location has been requested.



Viewing a child's location graph over the last week

This use case diagram displays the main functionality the Children or Parent access when viewing the user's location graph over the last week.



Functional Requirements

Category	Title	Requirement	Justification
User	Register	User must provide a unique email and password.	To ensure there is only one user per email and all users are unique.
User	Login	App must validate users' details and compare to the firebase authentication.	To access the app the user must provide the correct login details.
User	Logout	The user is able to log out at any stage.	To logout of the app and let a different user login on the same device.
User	Update Details	The user can update their person details	Allows a user to update their name phone number and profile picture
User	Create Family	The user can create a family which assigns a unique code to them to share.	To allows parents to connect with the children on the app.
User	Join Family	The user can enter the families code to be assigned to that family.	Allow the children to join the parents' family on the app.
User	View all Family Members	A user can view everyone in their family.	To allow users to interact with each other.
User	View all Safe Places	A user can view all safe places associated with their family.	To allow users to view all safe places and see if anyone is at them.

Category	Title	Requirement	Justification
User	View all Bad Places	A user can view all bad places associated with their family.	To allow users to view all bad places and see if anyone is at them.
User	Approve Locations	A user can approve suggested commonly visited places.	A parent is able to approve and categorize the place suggested.
User	Request Location	A user can request another user's location.	Allows any user to request the location of someone in the family.
User	View Location Graph	A user can view another users' location graph.	Allows any user to view another users' common locations over the last week in the form of a graph
System	Store User's Location in the Background	The system must continue to run periodically in the background and store the users' location.	This allows the system to collect the needed data for the app the function.
System	Calculate Commonly Visited Places	The system must store the users' locations frequently and calculate places they visit often and spend a lot of time in.	This allows the app to suggest the places the members of the family visit often to the parent.
System	Check if User is at a Saved Location	The system must compare all users' current locations to saved locations.	This allows the app to display on the home screen if any of the users are in a saved location.
System	Check if a User is at a Bad Location	The system must compare all users' current locations to saved bad locations.	This allows the app to notify the parent if someone is at a bad location.

Non-Functional Requirements

Simplicity

The whole user interface must be made to ensure it is as easy as possible to understand and navigate, as this app is aimed at both parents and children who sometimes might not be comfortable with technology and modern-day applications.

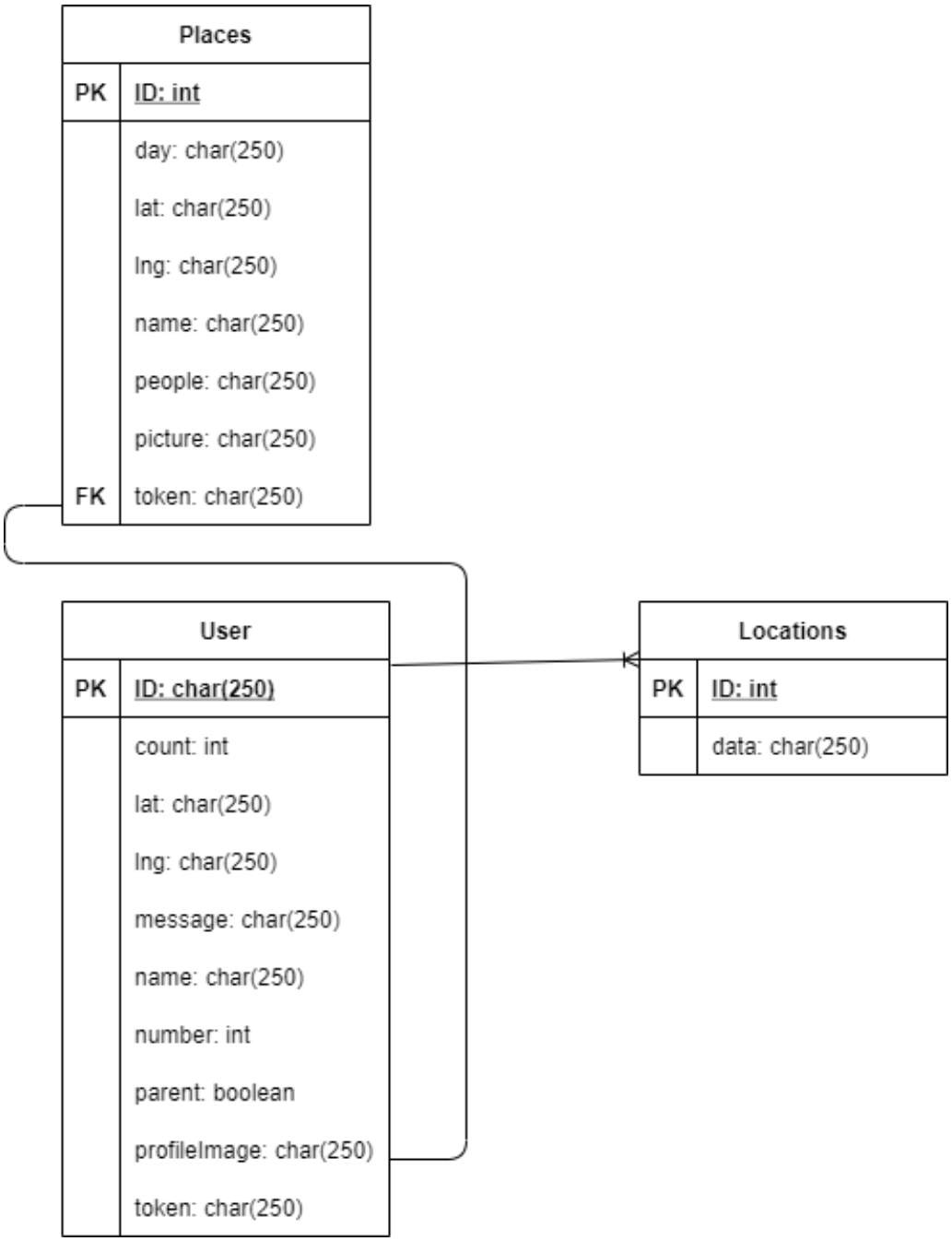
Theme

The user must be presented with a common theme throughout all the navigational pages. This is essential to make the user experience as best as possible with all aspects of the app being the same theme and colors.

Availability

The application must be ready for use off the bat on both android and IOS. As expected with all modern-day applications permissions will be requested from the user on the first-time use.

Data Model Design



Database Tables

Places

The Places table has a unique ID that is generated when the system automatically creates a Place. The current date gets stored along with the latitude and longitude of the place, the geolocated name and the family token who it belongs to.
The user can then interact with the place and assign a picture to represent this place. The system will automatically assign a user's name to people field if they are in that location.

Places	
PK	ID: int
	day: char(250)
	lat: char(250)
	lng: char(250)
	name: char(250)
	people: char(250)
	picture: char(250)
FK	token: char(250)

Users

The users table has a unique ID that is generated by Firebase when the user is created. When a user updates their information the name, number and profileImages' are assigned. Token is the user's family code and is either
inputted to join a family or generated when a family is created. Creating a family would also set the parent field to be true. Lat and Lng are the users Latitude and Longitude and are updated every fifteen minutes. Message is assigned by the app when a user requests a location or their location is requested.

User	
PK	ID: char(250)
	count: int
	lat: char(250)
	lng: char(250)
	message: char(250)
	name: char(250)
	number: int
	parent: boolean
	profileImage: char(250)
	token: char(250)

Location

The Locations table has a unique ID that is generated when the system automatically creates a Location every fifteen minutes. This stores in one string the date and time, the Latitude and Longitude, and the geolocated name of where the user is.

Locations	
PK	ID: int
	data: char(250)

Software Design

While Meet in the Middle might seem basic from the outside it is carried by its complex internal high-level algorithms. One of the main things I focused on when creating my application was how I was going to determine what are common locations that the users are spending time in. On my initial tests of algorithms designs, the system would suggest incorrect locations. For example, if you walked to a bus stop to catch a 9 o'clock bus every day and are always halfway to the stop from your house at 8:45, the algorithm used to think that that random spot was an oftenly visited location, even though you never stopped there. To combat this, the algorithm had to make sure you were spending a certain amount of time in the place as well as visiting it often. Another issue was that the Latitude and Longitude were too precise, to combat this, the algorithm had to be altered to compare if you were inside a certain radius of a place you had visited before. After a lot of tweaking with timings and radius sizes, the final algorithm will firstly, check if your current position is one of the saved locations for your family. If so, it will then check how long you have spent at that location without leaving the radius of it. After 40 minutes if you are still there, it will then check if you had visited that place in the last 7 days. If you have, the place is then suggested to the parent as somewhere the app thinks that family member spends a lot of time. This was the perfect middle ground I found that gave the best results.

Next, I had to make the app run in the background of the phone. This needed to be done as otherwise the user would need to have the app open twenty-four hours of the day for the app to be functional. This turned out harder that I thought it would be but after a lot of testing I found a solution.

Using a library called WorkManager I was able to have the app run periodic tasks every twenty minutes, this called my main position checking algorithm and stored the users' locations on the database. The minimum android apps let you run a task in background is every 15 minutes. However, my tasks runs every 20 minutes as I felt it suited my needs better.

```
WidgetsFlutterBinding.ensureInitialized();
Geolocator().getCurrentPosition(desiredAccuracy: LocationAccuracy.high);
Geolocator().checkGeolocationPermissionStatus();
Workmanager.initialize(callbackDispatcher, isInDebugMode: false);
Workmanager.registerPeriodicTask("1", fetchBackground,
    inputData: {},
    initialDelay: Duration(seconds: 30),
    frequency: Duration(m
```

```
void callbackDispatcher() async {
  WidgetsFlutterBinding.ensureInitialized();
  Workmanager.executeTask((task, inputData) async {
    switch (task) {
      case fetchBackground:
        await execute(inputData);
        break;
    }
    return Future.value(true);
  });
}
```

In order to have two users be able to communicate with each other. I essentially built a chat app that sent and read specific messages that would trigger actions in the app. For example, when a User requests another users location the “message” field of both Users is updated to reflect this in the database. The app will continually look for changes in the database using a Provider, and if it notices the “message” field being changed to an appropriate string, it will know that the users’ location is being requested and who requested it.

```
final users = Provider.of<List<UserData>>(buildContext) ?? [];
```

```
      if (spl[0] == "request" && userid == user.uid) {
```

User Interface Design

I chose to develop my application using Flutter and Dart as after a bit of research, I was blown away by the capabilities these two hold while working together. While the learning curve was definitely a challenge to begin with and at first felt like it was hindering my progress, the payoff in the end in terms of capabilities to UI design was worth it. I needed to teach and then familiarize myself with both of these to begin with as I had never dealt with either, but after a while it all became second nature, and the possibilities were endless. Flutter works by placing many widgets on top of each other in order to build a UI, this makes designing UI through flutter really easy and enjoyable.

Issues and Resolutions

The main issue I faced when designing my application was Flutter's lack of a built-in background service. While there were library's that I could use to solve this issue, it was hard to get all these differently libraries to work together and communicate to each other. However, the other benefits of Flutter greatly compensated for this issue.

Implementation

Login and Register

My application has a very simple and clean login and register screen. I had originally planned to expand on this but focused my time on other aspects of the app instead. When the user opens the app for the very first time, they are met with a register screen (Figure 1). To register the app needs to take in a unique email and a password to allow the sign up. If the user already has an account, they can navigate to the login screen by pressing the button placed on the right of the app bar. Validation has been applied to both text boxes to ensure they are not empty, and the user will be greeted with a small warning below the text box if its left empty. (Figure 2)

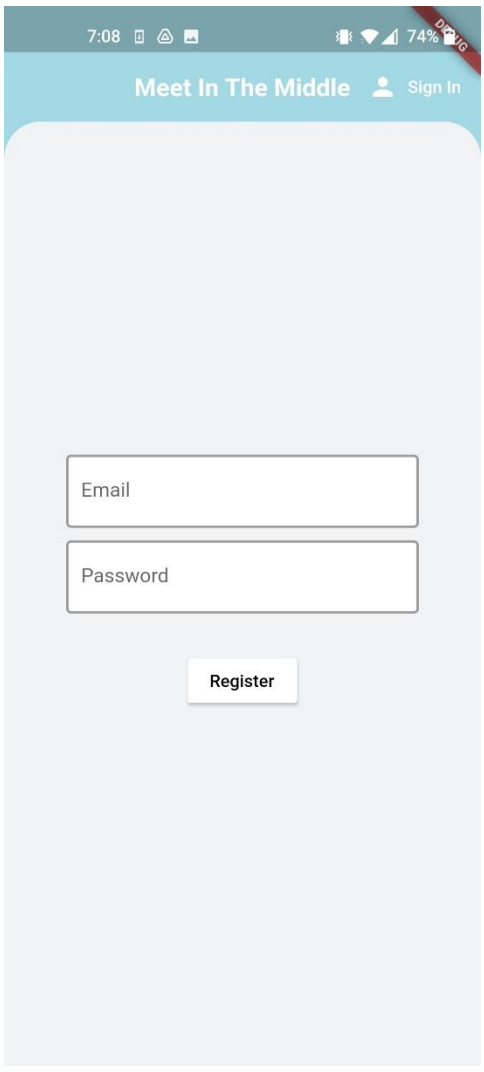


Figure 1

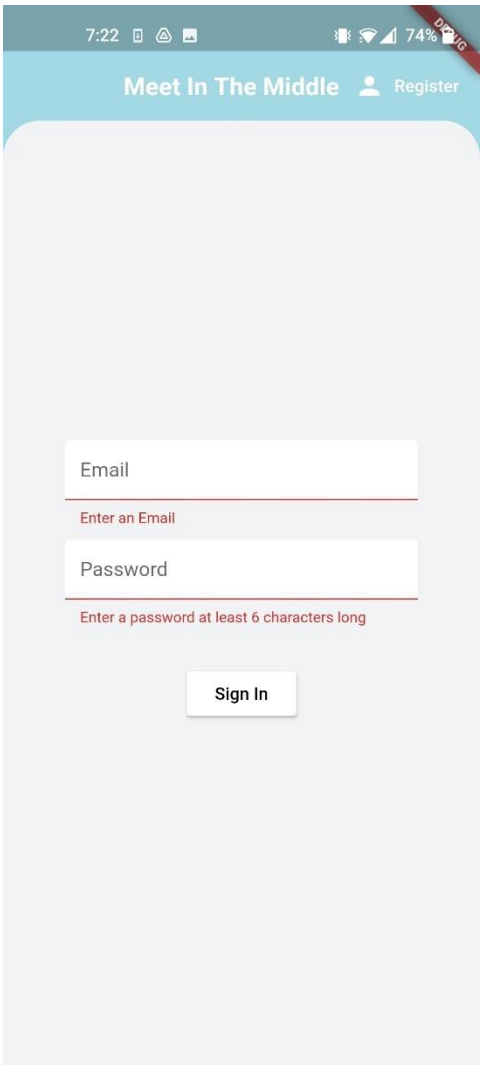


Figure 2

Update User Details and Create or Join a Family

After logging in the user can open the app drawer (Figure 3) by pressing the three-bar icon on the top left. Here they will be greeted with four options. Update profile will allow them to set their user details and pick a profile picture. (Figure 4) Create Family will create a unique family code and assign it to the user. (Figure 5) Join family allows a user to join an already made family by entering the family code. (Figure 6) Approve Safe Locations I will cover later.

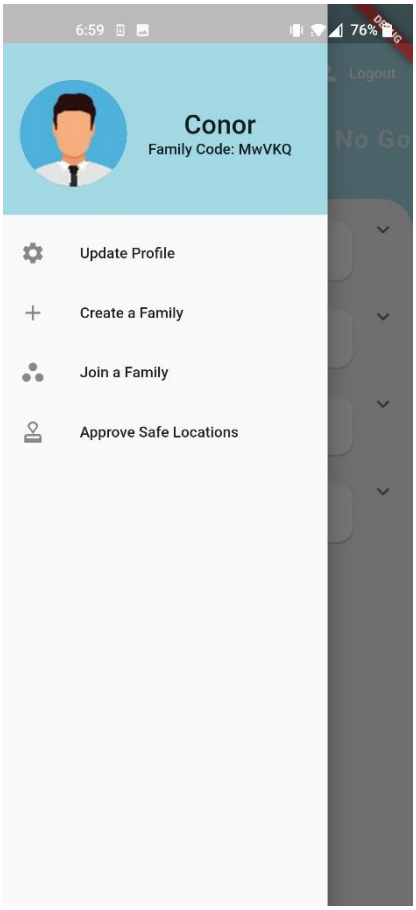


Figure 3

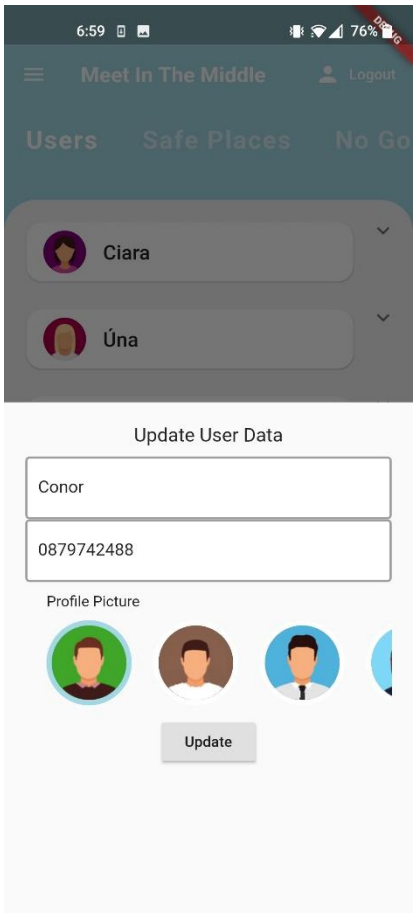


Figure 4

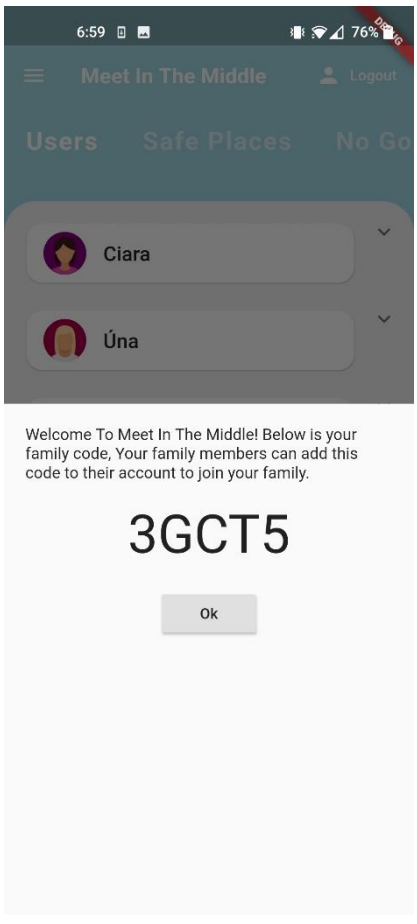


Figure 5

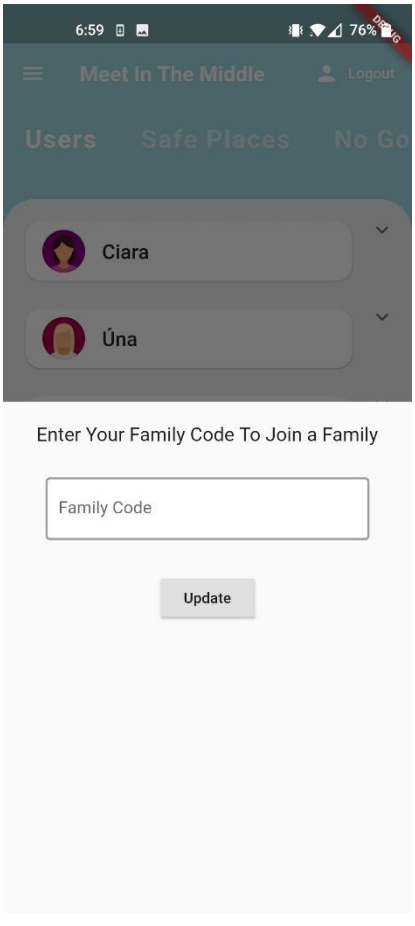


Figure 6

View Users and Saved Locations.

Once the user creates or joins a family and enters their user details, they are ready to use the app. The app is built around one homepage with three internal subpages. These pages allow the user to view and interact with all Users, (Figure 7) view all the “Safe Places” and who is at them (Figure 8) and view all the “No Go Zones” and who is at them. (Figure 9)

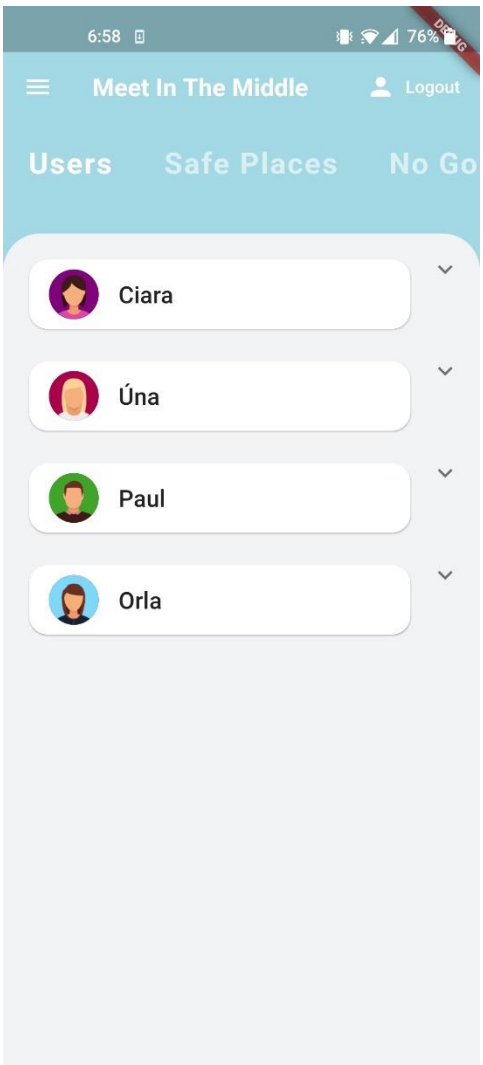


Figure 7

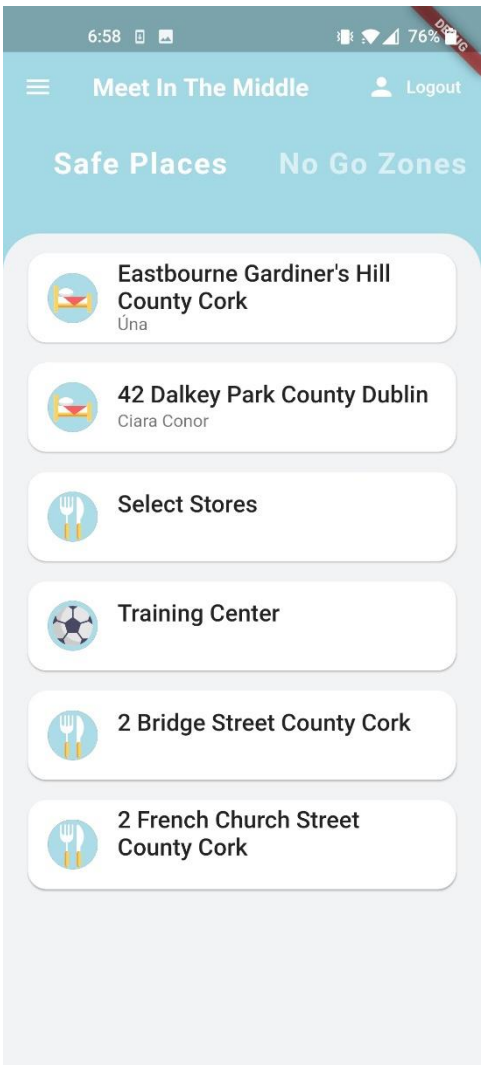


Figure 8

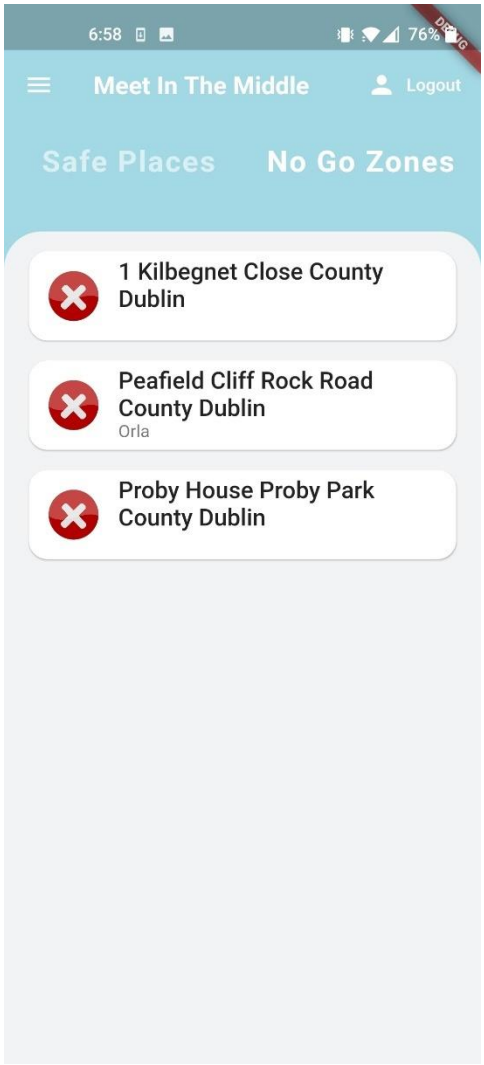


Figure 9

Interacting with a User.

If a user wants to interact with a member of the family they can simply click on their name. This pops down and animated drop-down menu. From here they are given two options, request location and View Location Graph. (Figure 10) Requesting location allows them to request the current location of the user and view location graph displays a week's summary of the four most common places they spent their time.

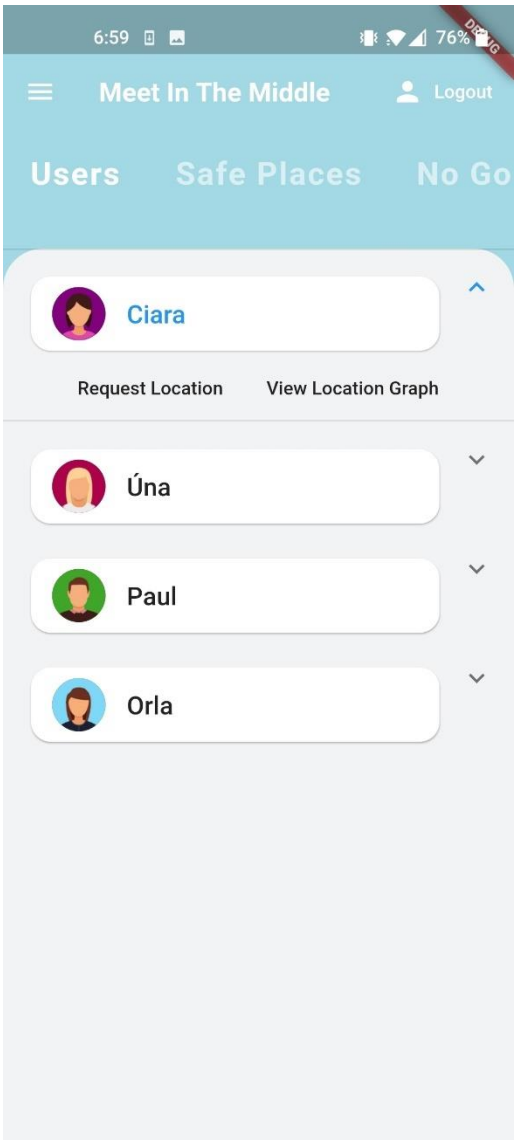


Figure 10

Requesting a User's Location

If the user decides they want to request the location of someone, they are greeted by a popup menu. (Figure 11) After pressing yes, the system will check if the selected user is in an already saved location. If they are, an appropriate message is displayed on screen (Figure 12). If they are not, the user is brought to their default SMS messaging app and a prewritten message is inserted ready for them to send. (Figure 13) After this they are brought back to the app and a countdown timer is displayed showing the remaining time left that the selected user has to respond before their location is shared. (Figure 14)

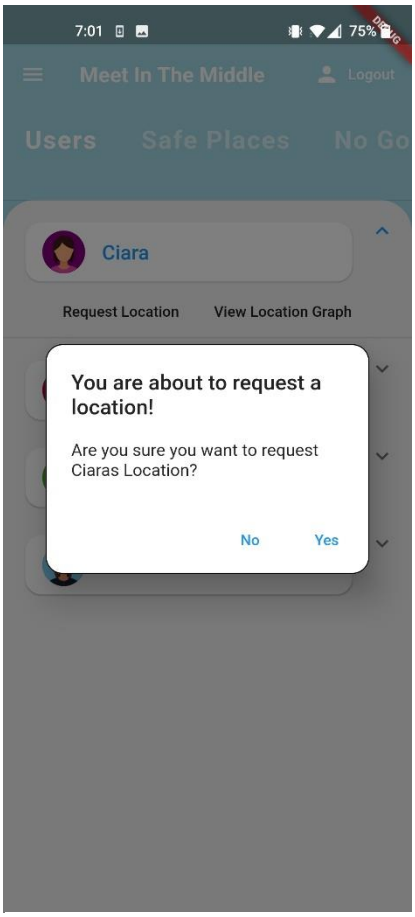


Figure 11

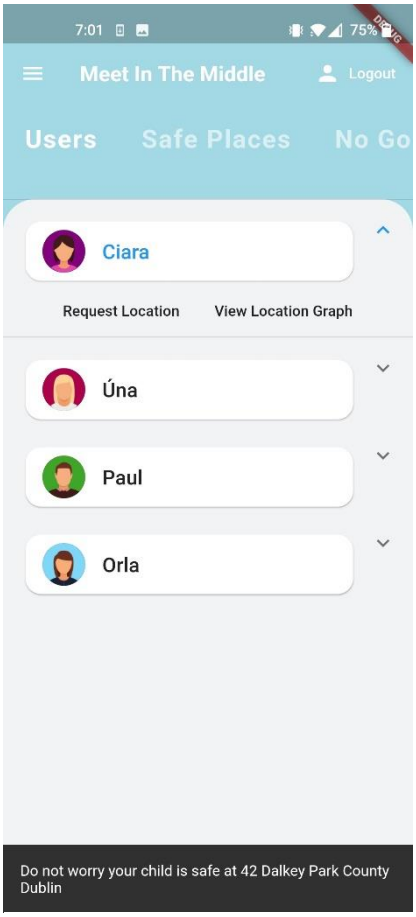


Figure 12

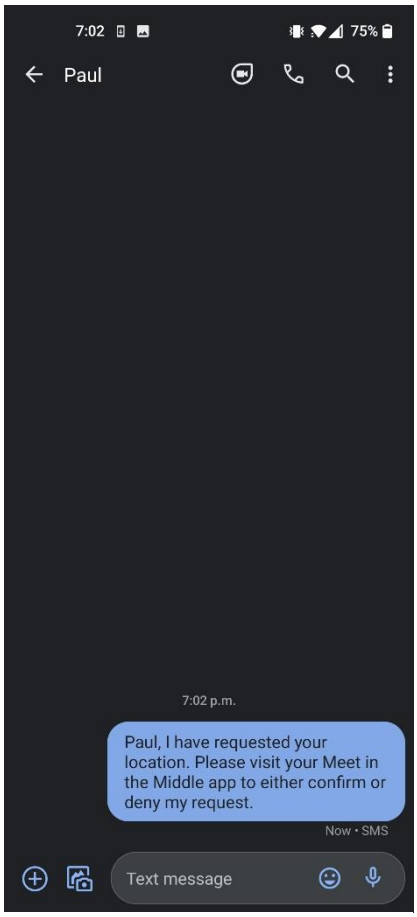


Figure 13

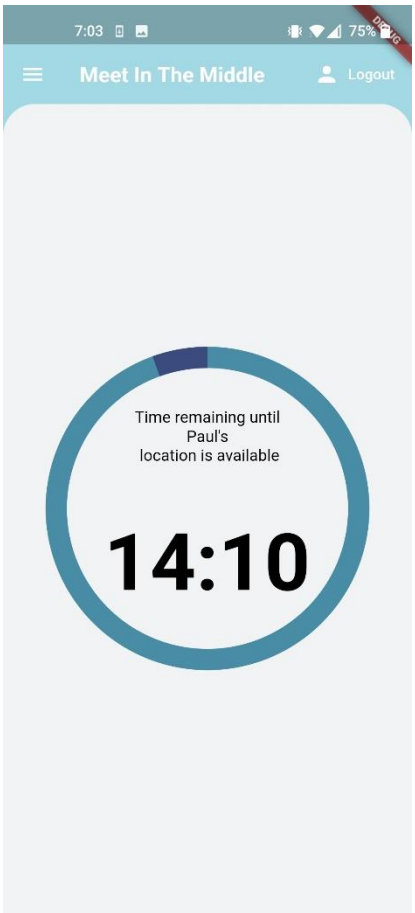


Figure 14

Notifications

Now that the user has requested a location they need to wait for a response or let the timer run out. Either way they will be greeted by one of two local notifications. If the selected user accepts the location request or the timer runs out without the selected user responding, the current user will receive a notification informing them that the location is available to view. (Figure 15) They will then be able to see the selected user's location on a map. However, if the selected user denies the location request the current user will receive a notification saying that the request has been denied and they will not be able to see the map. (Figure 16)

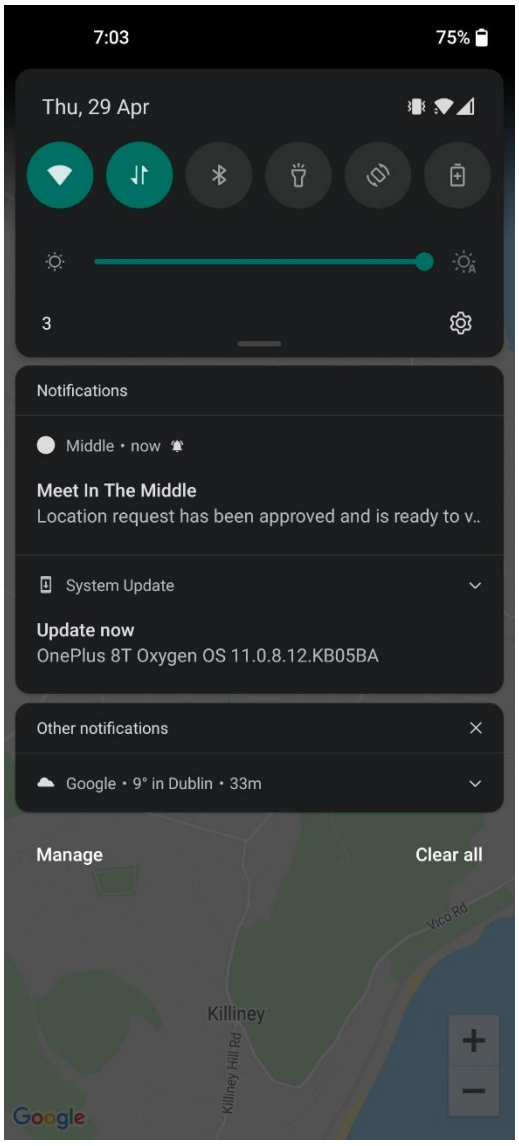


Figure 15

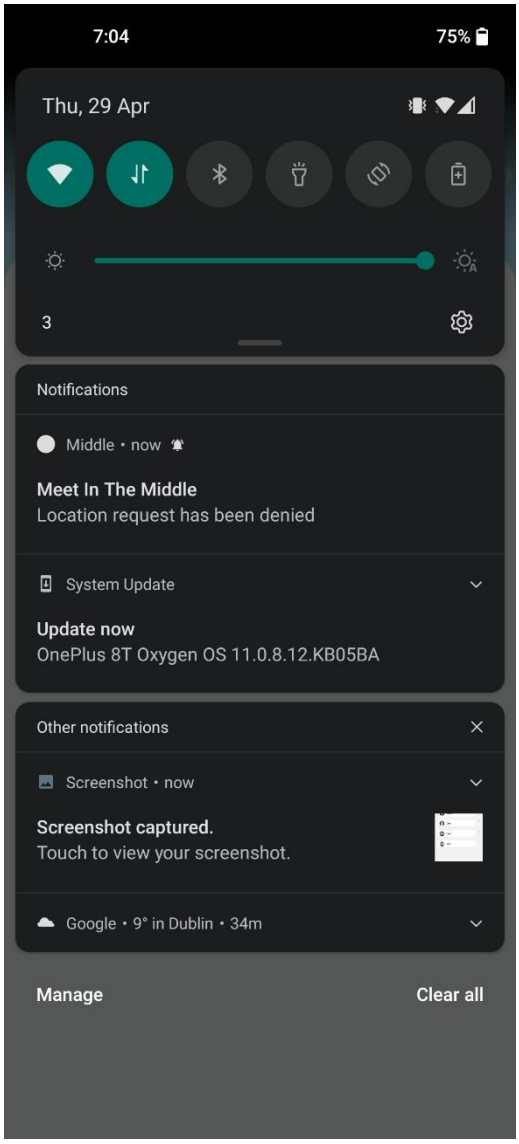


Figure 16

Viewing the Location on a Map

After the location request has been approved or the timer has run out, the user will see a map with a pin dropped on the selected users last known location. (Figure 17)

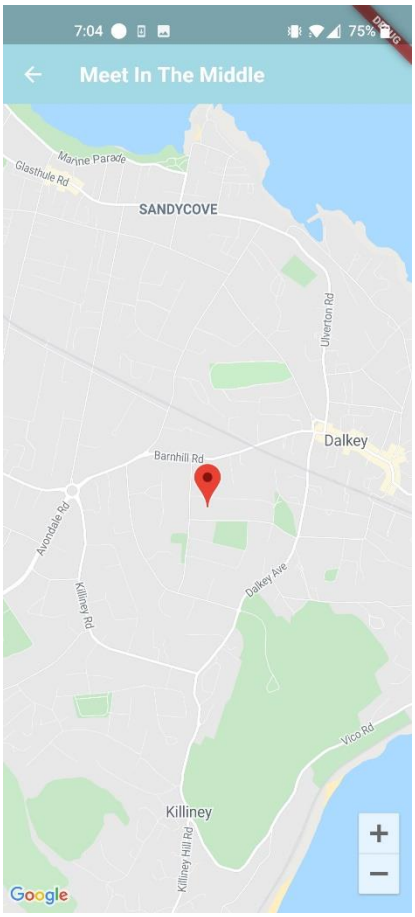


Figure 17

Viewing Week Summary of Places Visited

Going back to the animated drop-down menu in Figure 10, if the user were to select "View Location Graph" it would bring the user to an animated graph that will display the selected users four most commonly visited places over the last week. (Figure 18)

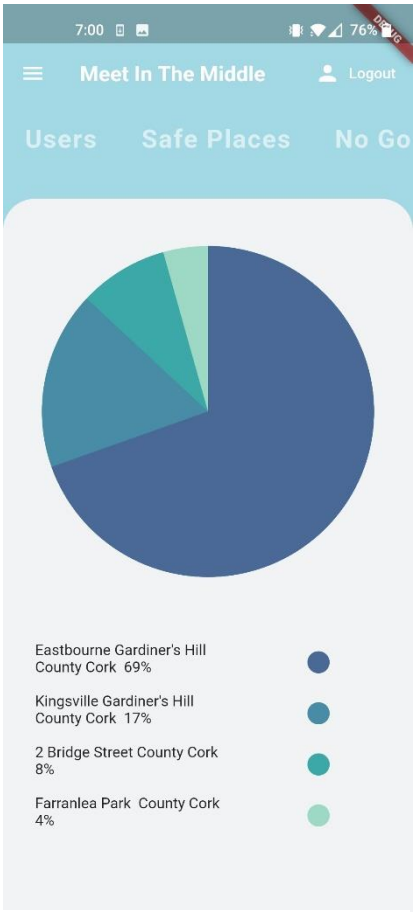


Figure 18

Approving Safe Locations

Returning back to Figure 3, If the user were to click on "Approve Safe Locations", the app will check if they are a parent. If not, an appropriate message will pop up letting them know they are not allowed access that feature (Figure 19). If the user is a parent however, the app will check if it has detected any suggested locations. If there is none an appropriate message appears letting the user know there is none to add (Figure 20). If there is a suggested place, the app will display a popup form with an editable name of the location, and a choice of icons to describe the location. As well as this, the user is given three choices. They think it is a safe space, they think its bad place, or they are not familiar with it. (Figure 21).

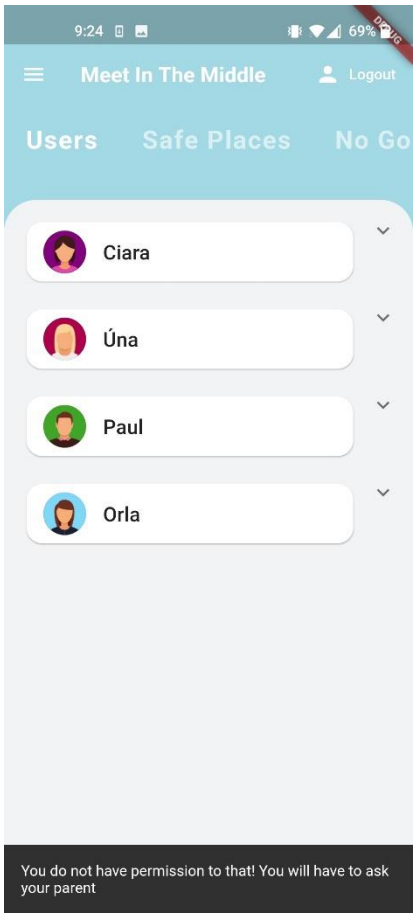


Figure 19

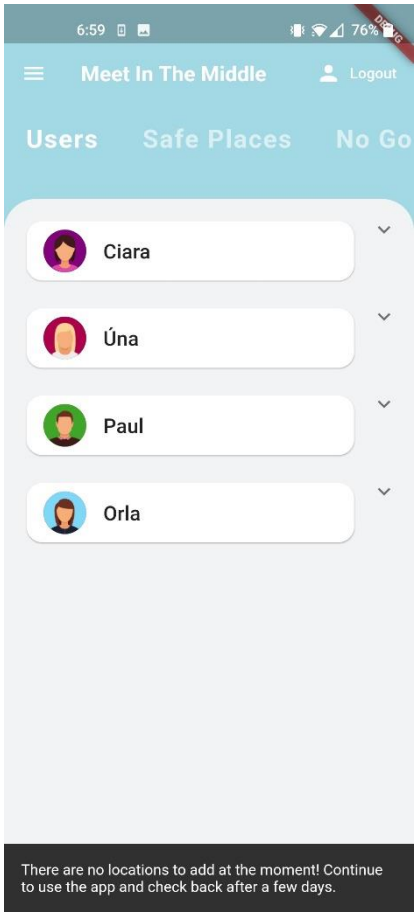


Figure 20

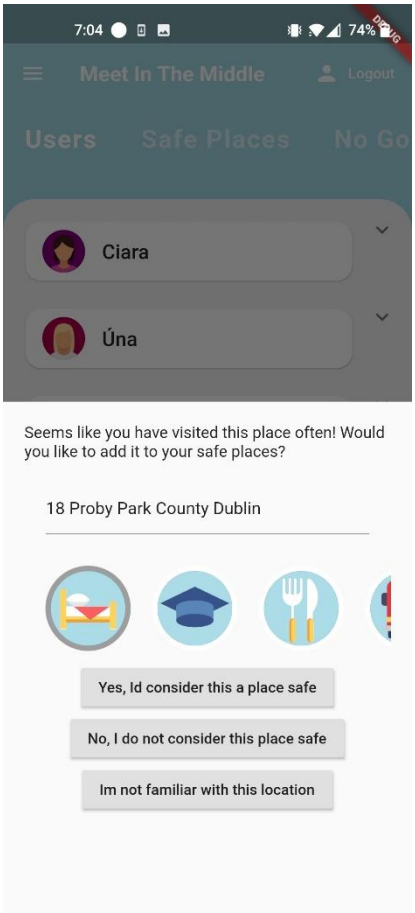


Figure 21

Test Cases

Test Case	Test Scenario	Expected Result	Actual Results	Pass/Fail
1.	Validate Text Fields	App should let the user know that fields cannot be blank.	As Expected	Pass
2.	Check Email	App will not let the user sign up with an existing email address.	As Expected	Pass
3.	Register User	App should add the user to the database and log them in.	As Expected	Pass
4.	Check Correct Details	App should let the user know if the username or password is wrong when logging in.	As Expected	Pass
5.	Login User	App should log in the user.	As Expected	Pass
6.	Remember User	App should not ask the user to sign back in after the app has been closed and the user has signed in before.	As Expected	Pass
7.	Log out	App should log out the user.	As Expected	Pass
8.	Update Details	App should allow the user to update their person details.	As Expected	Pass
9.	Create Family	App should allow the user to create a family and assigns a unique code to them to share.	As Expected	Pass

10.	Join Family	App should allow the user to enter a family's code to be assigned to that family.	As Expected	Pass
11.	View Family Members	App should allow the user to view everyone in their family.	As Expected	Pass
12.	Only View Family Members	App should only display users in their family to the user.	As Expected	Pass
13.	View Safe Places	App should allow the user to view all safe places associated with their family.	As Expected	Pass
14.	Only View Family's Safe Places	App should only display safe places assigned to their family to the user.	As Expected	Pass
15.	View Bad Places	App should allow the user to view all bad places associated with their family.	As Expected	Pass
16.	Only View Family's Bad Places	App should only display bad places assigned to their family to the user.	As Expected	Pass
17.	Approve Safe Location	App should allow a parent to approve a safe location.	As Expected	Pass

18.	Approve Bad Location	App should allow a parent to approve a bad location.	As Expected	Pass
19.	Ensure User is a Parent	App should ensure the user is a parent before allowing them to approve locations.	As Expected	Pass
20.	Request Location	App should allow the user to request another user's location.	As Expected	Pass
21.	See if User is at a Saved Location	App should check if the user is at a saved location after their location has been requested and notify the person who requested instantly without any wait.	As Expected	Pass
22.	Display Conformation Popup	App should display a conformation message asking if they are sure they want to request the location.	As Expected	Pass
23.	Send SMS	App should bring the user to their default SMS service and insert a readymade message.	As Expected	Pass

24.	Display Timer	App should display the animated countdown timer.	As Expected	Pass
25.	View Location Graph	App should allow the user to view another user's graph of popular places visited that week	As Expected	Pass
26.	Run Periodic Task in the Background	App should run every 20 minutes in the background.	As Expected	Pass
27.	Store Users Location in the Background.	App should write user's location to the database every 20 minutes.	As Expected	Pass
28.	Check for Common Locations in during Background Task	App should keep track of the user's location and run it through the algorithm in the background.	As Expected	Pass
29.	Add Common Location to Database if found	App should write the new place to the database in the background.	As Expected	Pass

30.	Check to See if a User is at a Safe Location in the Background.	App should check if the user is at a safe location in the background and if so displays their name under that location	As Expected	Pass
31.	Check to See if a User is at a Safe Location in the Background.	App should check if the user is at a bad location in the background and if so displays their name under that location	As Expected	Pass

Conclusion

To conclude, I am really happy with how my project turned out. While there were some features I was not able to add to it, I am more than happy with how the features I did add turned out. The whole experience of developing a full stack functional product was immensely gratifying and was a perfect conclusion to my four years in DIT/TUD. The experience itself was, while challenging and frustrating at times, extremity rewarding and fulfilling. It felt amazing to apply everything I had spent my last four years learning into one big project. I really apricated the level of creativity the college offered us when it came to this project, as we were allowed to express ourselves whatever way we deemed fit. Working on this project really opened my eyes to a lot of things that I feel will carry on with me in life. It has shown me the importance of time management, something I have greatly struggled with before. The fact that when I started this project the deadline was so far away lead me to not get working on it straight away. However, I soon realized that this was not an option, and through the help of the college with checkpoints and my own motivation I soon found myself enjoying working on the project and applying this level of time management to other aspects of my final year. Another thing working on this project has shown me is appropriate goal setting. My supervisor Neil was very good and broke down my project with me into induvial goals to reach. This was something I had not done before, and it was immensely useful. I will definitely approach more problems this way moving forward. If I were to do a project of this scale again, I would apply both these practices from the start. As well as making sure not to spend too long on one problem and to understand that sometimes its better to work on something else and then come back to it with a fresh mind. All in all, I am very satisfied with the work I got done on my project. I feel like it has a lot of potential and I am planning to continue to develop it into the future.