EXPECTATIONS OF ORIGINALITY & STANDARDS OF ACADEMIC INTEGRITY

ALL SUBMISSIONS must meet the following requirements:

1.The decision on whether a submission is a group or individual submission is determined by the instructor.

Individual submissions are done alone and should not be identical to the submission made by any other student. In the case of group submissions, all individuals in the group must be listed on and must sign this form prior to its submission to the instructor.

2.All individual and group submissions constitute original work by the individual(s) signing this form.
3.Direct quotations make up a very small proportion of the text, i.e., not exceeding 5% of the word count.
4.Material paraphrased from a source (e.g., print sources, multimedia sources, web-based sources, course notes or personal interviews) has been identified by a numerical reference citation.
5. All of the sources consulted and/or included in the report have been listed in the Reference section of the document.
6.All drawings, diagrams, photos, maps or other visual items derived from other sources have been identified by numerical reference citations in the caption.
7.No part of the document has been submitted for any other course.
8.Any exception to these requirements are indicated on an attached page for the instructor's review.

REPORTS and ASSIGNMENTS must also meet the following additional requirements:

1. A report or assignment consists entirely of ideas, observations, information and conclusions composed by the student(s), except for statements contained within quotation marks and attributed to the best of the student's/students' knowledge to their proper source in footnotes or references.
2. An assignment may not use solutions to assignments of other past or present students/instructors of this
3. course or of any other course.
4. The documenthas not been revised or edited by another student who is not an author.
   For reports, the guidelines found in Form and Style, by Patrick MacDonagh and Jack Borden (Fourth Edition:May 2000, available at http://www.encs.concordia.ca/scs/Forms/Form&Style.pdf) have been used for this submission.

LAB REPORTS must also meet the following requirements:

1.The data in a lab report represents the results of the experimental work by the student(s), derived only from

the experiment itself. There are no additions or modifications derived from any outside source.
2.In preparing and completing the attached lab report, the labs of other past or present students of this course or any other course have not been consulted, used, copied, paraphrased or relied upon in any manner whatsoever.

SOFTWARE must also meet the following requirements:

1. The software represents independent work of the student(s).
2. No other past or present student work (in this course or any other course) has been used in writing this software, except as explicitly documented.
3. The software consists entirely of code written by the undersigned, except for the use of functions and libraries in the public domain, all of which have been documented on an attached page.
4.No part of the software has been used in previous submissions except as identified in the documentation.
5.The documentation of the software includes a reference to any component that the student(s) did not write.
6.All of the sources consulted while writing this code are listed in the documentation.

Important: Should you require clarification on any of the above items please contact your instructor.

List of Figures

1 Mission Statement

Opportunity statement: Develop a robust, climate-resistant personal tracker that measures critical environmental parameters to monitor health and surroundings of industry workers in extreme or dangerous conditions. The app will monitor multiple devices simultaneously and display tracking data of individuals or teams and compile real-time reports for supervisors or team leaders.

Product Description: The tracker is for use in extreme conditions and will monitor important personal metrics like environmental temperature and VOC concentrations. This information is made available on the accompanying app where it can be accessed by team members in charge of logistic decision making.

Benefit Proposition: Extreme work or play conditions can lead to safety risks and less efficient performance because of compromised physical capabilities. Unfamiliar and uncomfortable environments can lead to poor decision making and panic responses. These effects should be reduced to minimize accidents and life-threatening situations. Our product will allow professionals and athletes to track changes in their environment, implementing real-time monitoring and an alert system to help users traverse or escape difficult situations.

Key business goals:

- Create an all-in-one solution for at-risk workers and athletes that is modular, expandable, and user-friendly.
- Create a product platform that can later be extended to capitalize on feature development.
- Create opportunity for multiple income streams without sacrificing brand integrity or product quality.
- Position ourselves as the industry leaders in wearable hazard-monitoring products
- The team will require around 300 hours to create the app and device prototype at $50 per hour ($15000 labour costs). Product will require sensors and components totaling approximately $25 and will be sold for $300. A dynamic sales model (with units sold as the

independent parameter) gives a break-even point that occurs between 54 and 55 units sold, which asserts a per-unit labour cost of 277.78 that decreases as sales continue.

Target market: Our target market will be workers or athletes in harsh conditions that need to monitor their health, environment, and location data and/or share that information with others. Eventually, we would expand to wholesalers and distributors of wearable hazard-tracking technology.

Assumptions & Constraints:

Assumption #1: Data transmission infrastructure (i.e. on-site Wi-Fi or cellular data) is available.

Assumption #2: Wireless interference is not present or occurs to a negligible degree.

Assumption #3: There exists a facile way in software to identify physical devices as WhereSafe units and isolate them from other services.

Constraint #1: Maintain April 2023 deadline.

Constraint #2: Physical unit is lightweight and comfortable to wear.

Constraint #3: Physical unit includes a sensor that can collect and transmit data.

Constraint #4: Android app has functionality to store, access, and display the data.

Constraint #5: Due to its nature: system is robust, secure, dependable, and user-friendly.

Stakeholders:

Project manager.

Product developers.

Users who will purchase and use the product.

Suppliers for the sensors and other components.

Suppliers for the wearable unit fabrics and materials.

Google Play Store or other third-party software distributors.

Eventually, industry suppliers of health and safety products.

2 Introduction

The third release version of the WhereSafe System introduces a fully implemented sensor suite, utilizing the Adafruit BME680 sensor to collect and transmit a wide range of environmental data to the associated mobile application via an ESP32 microcontroller. We have made significant strides in database functionality, integrating local storage and Firebase services with authentication to bolster data security and storage/access features.

Our focus on user accessibility and experience has yielded important advancements, such as environmental threshold alert notifications and a dark mode display option. Feedback from multiple user experience test batteries has guided the refinement of our information display and app navigation, ensuring a seamless and intuitive user experience.

Additionally, we have successfully developed a prototype of the wearable device, now powered by a standard 9V battery, which allows for approximately four hours of portable operation. This achievement brings the WhereSafe System closer to its ultimate goal of providing an efficient and dependable solution for individuals and teams seeking accurate and up-to-date information about their surroundings.

The following sections of this report will detail the development process, challenges encountered, solutions implemented, and prospects for the WhereSafe Personal Safety System.

2.1 Definitions, Acronyms, and Abbreviations

The following table details the specific terms and abbreviations used throughout this report.

Table 1 Definition and Abbreviation

Abbreviation	Term	Meaning
BLE	Bluetooth Low Energy	Wireless personal area network technology that uses about half the peak current consumption of traditional Bluetooth Data
GATT	Generic Attribute Profile	transfer protocol utilized by Bluetooth Low Energy devices
UUID	Universally Unique Identifier	A unique 128-bit number that is used to identify objects (software/hardware) in a way that avoids naming conflicts with other objects

3 Ethical Considerations in the WhereSafe System

For the WhereSafe team, our desire was to place ethics at the core of our business strategy. Not only does this resonate with our own values, but a morally responsible approach to product design is especially important for a safety monitoring system. In examining the ethical dimensions of the WhereSafe application and wearable device, the team explored measures in our immediate control that could be taken to mitigate any potential ethical concerns, as well as measures that we could take in the future development of our product to further address these concerns. The final key areas of focus came to include *safety*, *integrity*, *security*, and *environmental stewardship*.

Safety and Reliability: WhereSafe is designed to provide real-time safety information to users in extreme conditions and environments. The target client base will be operating in dangerous, potentially lethal conditions. In some instances, their lives could depend on the proper functioning of this device. Consequently, ensuring the reliability of the product system is pivotal, and the team has therefore taken severable measures to confirm the dependability of this product. For instance, rigorous quality assurance processes have been incorporated in code development and management processes. Stress tests were used to ensure that the device can withstand various environmental challenges. The use of high-quality, durable materials in the manufacturing of the wearable device ensures its physical robustness in even the harshest environment. To address power availability, the circuit is designed to be energy-efficient and ensure the device can reliably operate for over 4 hours, allowing users to enjoy unit functionality without worrying about battery life. Moreover, further iterations will implement solar charging capabilities or alternative power sources to further enhance the device's performance in remote locations. All of these measures are designed to ensure correct operation of the device in any circumstances, especially for critical moments of extreme device dependence. Additional field testing to cover the various environmental extremes (such as temperature and VOC concentrations) would help further affirm the device's performance and reliability in real-world situations.

Data Integrity; Transmission Accuracy: Accurate data flows are crucial for the effectiveness of the WhereSafe system. Inaccurate or corrupted data could lead to users making delayed or poor decisions based on incorrect information, potentially resulting in dangerous outcomes. To address this concern, robust data transfer protocols have been implemented in a two-pronged approach: integrity of the data within each stage of the transmission stream, and accuracy of the protocols responsible for moving the data back and forth. Data handling between the wearable device and the application is designed to be reliable and accurate by using up-to-date industry standard protocols such as Bluetooth and WAP structures. A benefit of modern electronic devices is the ability to release regular updates to the application and the device firmware. This allows the WhereSafe team to address any precipitant issues affecting data integrity and transmission accuracy in nearly real time, resolving user security concerns within very short timeframes. Continued consultation with industry experts and professional data transmission software developers would only serve to further develop the effectiveness and efficiency of the product's methods.

Data Security: Given the sensitive nature of the data collected by WhereSafe, data security is a priority in order to protect sensitive information and maintain user privacy. This has resulted in the implementation of several security measures, including a decentralized storage solution for user data and multiple authentication protocols for accessing the app through a registered user account. By being proactive and transparent with respect to WhereSafe data security practices, the team aims to build trust among users and demonstrate the company's commitment to cybersecurity.

Ethically Sourced Materials: The ethical considerations of our product extend beyond its functionality and data security. We recognize the importance of responsibly sourcing materials for the wearable device to minimize our environmental impact. To this end, we have carefully chosen recycled and repurposed inputs and enacted a zero-waste policy for product manufacturing. By prioritizing ethically sourced materials (like recycled PLA and offcut fabrics) in the manufacturing process, WhereSafe is contributing to a more sustainable and socially responsible technology industry. The team is committed to continually seeking opportunities for

improvement, such as collaborating with organizations focused on reducing environmental impacts in the technology industry.

The WhereSafe system has been designed with ethical considerations at the forefront. By actively working to mitigate potential ethical concerns and committing to ongoing improvements in these areas, we aim to provide a product that not only helps keep users safe in extreme conditions, but also aligns with our mission statement and the values of our customers.

4 Requirements: Product Backlog

4.1 Completed Tasks

Story ID	Story Title	Card	Story Points	Sprint	Status	Conversation	Confirmation
COM-1	View Data	As a worker/athlete, I want to view metrics tracked by the device on my smartphone so that the data can be visualized.	5	Sprint 1	Completed	This feature allows the user to visualize the metrics tracked by the device in a user-friendly way.	1. Can the user view data that is measured by the device?
DA-1	Gas Sensor	As a user, I want to obtain real-time data of potential environmental gas-phase contaminants so that I can know if there are any dangerous gasses present	4	Sprint 1	Completed	This feature allows the user to determine if there are any dangerous gasses present in the environment which may require them to evacuate.	1. Can the user view metrics on the gasses present in the surrounding environment?
DA-2	Temperature Sensor	As a user, I want to obtain real-time data of ambient temperature conditions so that I can be aware of the temperature conditions and avoid any potential dangers.	4	Sprint 1	Completed	This feature allows the user to determine if the temperature is within a safe range to perform the desired activity.	1. Can the user view metrics on the current temperature?

DA-3	Pressure Sensor	As a user, I want to obtain real-time data of ambient pressure conditions for personal safety.	4	Sprint 1	Completed	This feature allows the user to be aware of the ambient pressure conditions to determine any potential concerns.	1. Can the user view data on the current ambient pressure?
UI-1	Display Metrics	As a user, I want to display safety metrics captured by the hardware device.	4	Sprint 1	Completed	Including safety metrics allows users to easily access data they need	1. Is the interface easily navigable? 2. Can we access all metrics as needed?
UI-2	Setup Account	As a user, I want to set up an account using my email so that I can save my data and preferences to my account.	7	Sprint 2	Completed	Gives users their own private and personalized environment	1. Can we access an account we created? 2. Is the corresponding data present?
UI-3	Firebase Log In	As a user, I want to securely log in to the system using my email so that I can view my saved data.	3	Sprint 2	Completed	Creates a safe and secure system that gives users privacy	1. Can we login with our email account?

UI-8	Data Visualization	As a user, I want to display these safety metrics in a clear and user-friendly interface so that I can easily navigate to the desired metrics.	11	Sprint 2	Completed	Including safety metrics allows users to easily access data they need	1. Is the interface easily navigable? 2. Can we access all metrics as needed?
UI-9	Create Team	As a user, I want to be able to create a team so that other users can join my team and I can view their data.	2	Sprint 2	Completed	This feature allows the user to create a team for other users to join.	1. Can the user create a team?
UI-10	Join Team	As a user, I want to be able to join a team, so that I can view my team member's data	2	Sprint 2	Completed	This feature allows users to join teams that are already created by another user	1. Can the user join a team?
DA-6	Altitude Sensor	As a user, I want to obtain real-time altitude data so that I can be aware of my general location relative to sea level.	4	Sprint 2	Completed	This feature can indicate to the user their altitude when performing climbing related activities. It can also indicate the depth at which a person is when working underground.	1. Can the user view metrics on their current altitude levels?

DA-7	Humidity	As a user, I want to obtain real-time humidity data so that I can be aware of the relative humidity in the surrounding environment.	4	Sprint 2	Completed	This feature allows the user to determine if the relative humidity is within a safe range to perform the desired activity.	1. Can the user view metrics on the current humidity levels?
DB-1	Analyze Metrics	As a user, I want to clearly observe past data metrics of each device-wearer, so that I can know if I was at risk of any dangers during past activities.	6	Sprint 2	Completed	This allows past data to be viewed by the user at any time. This is helpful if the user desires to view in- depth data following an excursion.	1. Can the user view past data that was tracked by the device?
DB-2	Firebase Database Setup	As a user, I want to observe past data metrics from anywhere.	7	Sprint 2	Completed	Allows the user to have their data accessible from anywhere using WIFI.	1. Can the user view data over WIFI from anywhere?
DB-3	Monitor Team	As a user, I want to monitor the real-time data of multiple device-wearers at once, so that I can make sure none of my co-workers/peers are at risk of any danger.	9	Sprint 2	Completed	This allows a user to view data of their team members to ensure that they are not at risk of any danger. This can indicate whether they need to assist anyone or call for help in an emergency.	1. Can the app user view the data of device users?

DS-0	Device Portability	As a user, I want my physical device to be battery powered and portable.	4	Sprint 3	Completed	Intermediate step between concept circuit and wearable prototype	1. Can we bring our device anywhere we want?
UI-5	Choose Team Member	As a user, I want to choose which user's data will be displayed, so that I can view the desired person's data.	8	Sprint 3	Completed	Allows the admin to control and manage all users' data and metrics	1. Can we access all devices connected?
DS-1	Design: Comfort	As a user, I want to have a comfortable wearable device to measure my real-time data, so that I can wear it for long periods of time.	6	Sprint 3	Completed	Allows users to wear the device for a very long time while keeping disturbance to a minimum	1. Can the user adjust the device?
DA-8	Modify Name	As a user, I would like to change my device's name if desired.	3	Sprint 3	Planned	This feature allows the user to give a unique name to their device.	1. Can the user change their device's name?

UI-12	Dark mode	As a user, I want dark mode functionality to alleviate eye strain when using the app.	7	Sprint 3	Planned	This feature allows the user to change the theme between light mode and dark mode	1. Can the user toggle the theme?
UI-13	App security: Frontend	As a user, my app should be robust and well-tested to ensure a seamless and pleasing user experience.	6	Sprint 3	Planned	Naive user operation will not cause improper or unexpected product behaviour	1. Can the user switch between views without crashing the app?
DB-4	App security: Backend	As a product owner, the app should be well-tested to ensure data security and integrity while the product is operational	4	Sprint 3	Planned	Random or irregular data streams and/or functioning of communication protocol will not cause unexpected product behaviour	1. Will the views fail if not given time to load?
UI-14	Leave Team	As a user, I want to be able to leave a team if I don't want to share my data with other users.	3	Sprint 3	Planned	This feature allows the user to leave a team if they would no longer like to share their data with those users.	1. Can the user leave their team?

UI-15	Language	As a user, I want to switch the app's language to French for a more comfortable experience in my preferred language.	3	Sprint 3	Completed	This feature allows the user to switch the app's language to French and also switch it back to English depending on their more comfortable language	1. Can the user change languages to French or English?
DA-9	Notifications	As a user, I would like to get notifications if the surrounding environment has bad air quality or extreme temperatures	4	Sprint 3	Completed	This feature allows the user to be alerted by any extreme conditions.	1. Is the user receiving notifications if the air quality is low? 2. Is the user receiving notifications if the temperature is extreme? 3. Is the user receiving notifications when the humidity levels are at extremes?
DA-10	Background Connection	As a user, I want to be able to track the surrounding environment data even while my phone is set to sleep so that I don't have to keep my phone on and save battery	5	Sprint 3	Completed	This feature allows the user to track surrounding environment data even while their smartphone is asleep	1. Can the user receive data while their phone is asleep?

DA-11	Easy Connection	As a user, I want to be able to easily connect to the device if I am disconnected to ensure that I can measure my surrounding environment	4	Sprint 3	Completed	This feature allows the user to connect to the device if the device is disconnected	1. Can the user easily connect to a device?
UI-16	Help	As a user, I would like to have some helpful information about the app so that I know what is possible within the app.	4	Sprint 3	Completed	The help section can help new users understand the app.	1. Can the user view a help section that provides information about the app
UI-17	Settings	As a user, I would like to easily change my preferences and settings so that I can alter the look and feel of the app, or manage my device.	4	Sprint 3	Completed	This feature allows the user to modify their preferences from a centralized location within the app	1. Can the user navigate to a settings page? 2. Can the user modify settings?
DA-13	Forget Device	As a user, I want to be able to forget my previously connected device so that I can connect to a new one.	3	Sprint 3	Completed	This feature allows the user to connect to a new device if they've previously connected to another device	1. Can the user forget the connection with a device?

DA-14	Pair Device	As a user, I want to be able to select the device I want to connect to within a list so that I can decide which device to select	3	Sprint 3	Completed	This feature allows the user to easily select a device to connect to from a list	1. Can the user view a list of nearby Wheresafe devices and connect to it?

4.2 Future Tasks

Story ID	Story Title	Card	Story Points	Sprint	Status	Conversation	Confirmation
UI-6	Create Profile	As an admin, I want to create new user profiles for various device-wearers, so that I can easily create multiple profiles for the people I manage.	3	Future	In Progress	Creates a modular system for users to implement their own systems and devices	1. Can we create a profile at any time?
DA-4	Light Sensor	As a user, I want to obtain real-time data of ambient light conditions so that I can make sure the visibility levels are safe for activities	4	Future	Planned	This feature can help the user to determine if the environment is too dark or too bright for the desired activity.	1. Can the user view data on the ambient light levels?

DA-5	Sound Sensor	As a user, I want to obtain real-time data of ambient sound conditions so that I can be alerted if the sound levels are at an unsafe level.	4	Future	Planned	This feature can help indicate to the user if the sound levels are within a safe range to not damage their hearing.	1. Can the user view metrics on the current ambient sound levels? 2. Can the user visually determine that the sound levels are within a safe range?
DS-3	Haptic Feedback	As a user, I want to receive haptic feedback based on my surroundings to improve situational awareness.	5	Future	On Hold	Updates the user on its surroundings through a set of specific haptic patterns.	1. Are the haptics recognizable? 2. How strong are the haptics?
DS-5	Battery Charge	As a user, I want the unit battery to charge efficiently and quickly.	1	Future	On Hold	Creates a sustainable and easily rechargeable device that can be reused quickly	1. How fast does the battery take to charge? 2. What kind of charger does it need?
COM-2	Device Proximity	As a user, I want to obtain real-time data of the device's proximity to other devices, so that I can know where my peers are.	4	Future	Planned	Having this ability will allow device users to locate other users.	1. Can the user view information about the distance of another user relative to their position.

DS-2	Design: Robust	As a user, I want the device to be robust and secure, so that it can sustain harsh environments.	4	Future	In Progress	Raises confidence in the product as an environmental tracker and secured data system	1. Is the device secure against uncertain operating conditions?
DS-3	Design: Dependable	As a user, I want the device to operate reliably so I can depend on the data I am using.	4	Future	In Progress	Gives the device a longer and more reliable lifetime for a versatile environment compatibility	1. Can the device sustain shocks?
DS-4	Battery Life	As a user, I want the battery life of my unit to be long enough to last until the end of my shift or excursion.	3	Future	In Progress	Allows the device to be portable and reliable throughout a long journey	1. How long can the battery last with heavy usage? 2. Does the battery's duration change with the temperature?
UI-11	Secure Authentication	As a user, I would like to set up 2FA to increase my account's security	2	Future	In Progress	This feature increases the user's account security by adding a layer of verification before signing into an account.	1. Can the user set up 2FA on their account? 2. Is the verification being correctly sent to the user?

5 Design

5.1 Android Application Wireframes

5.1.1 Mockup Designs

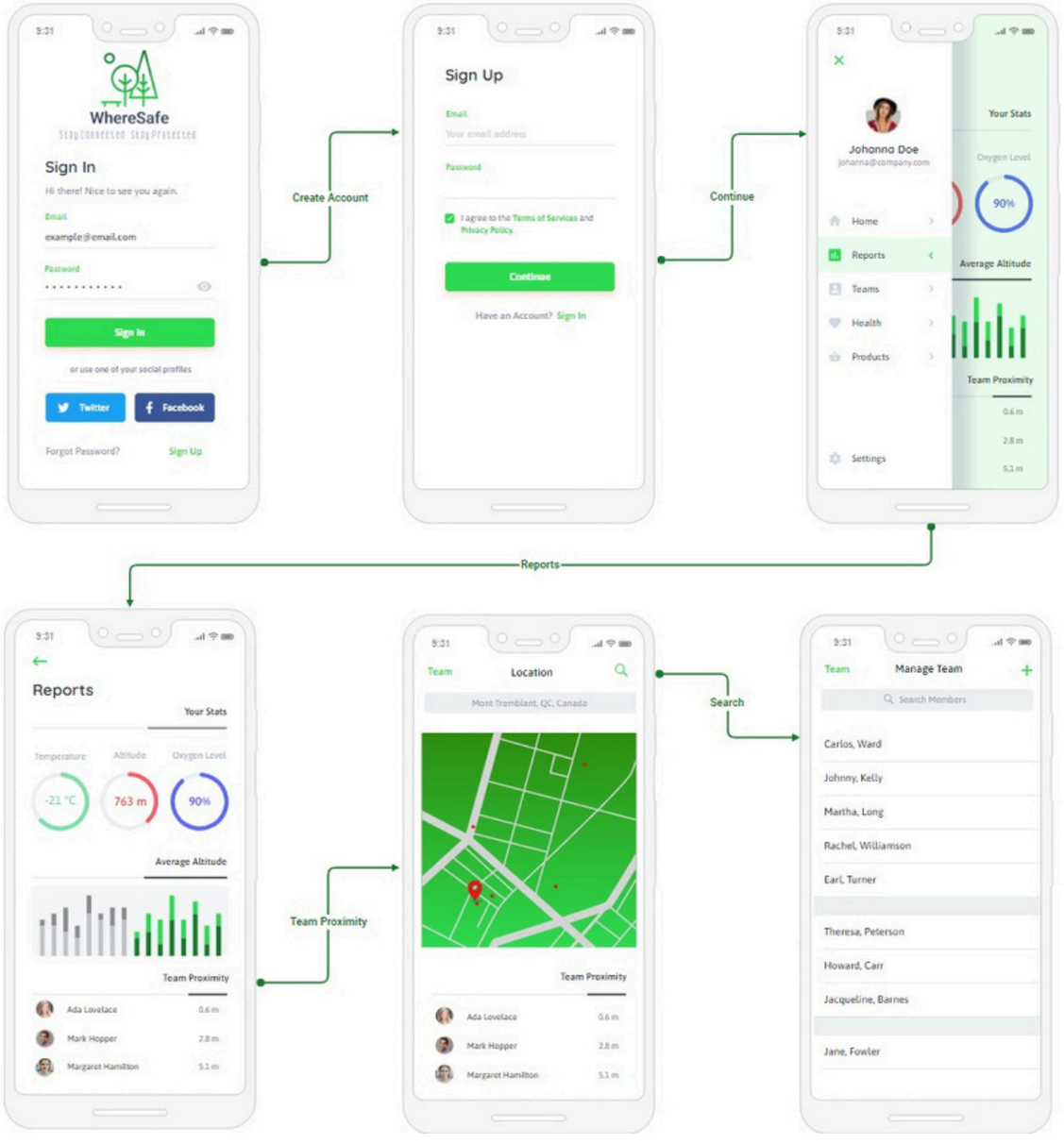
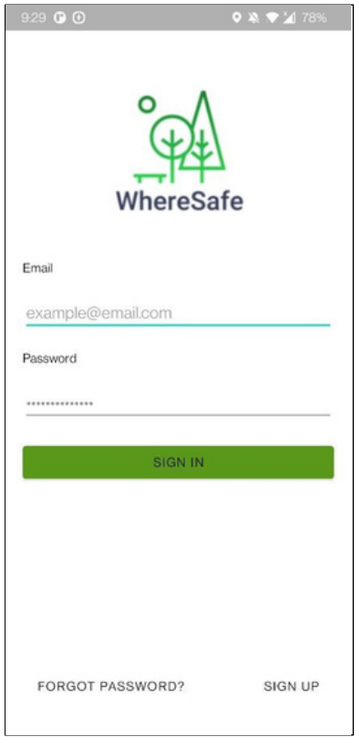


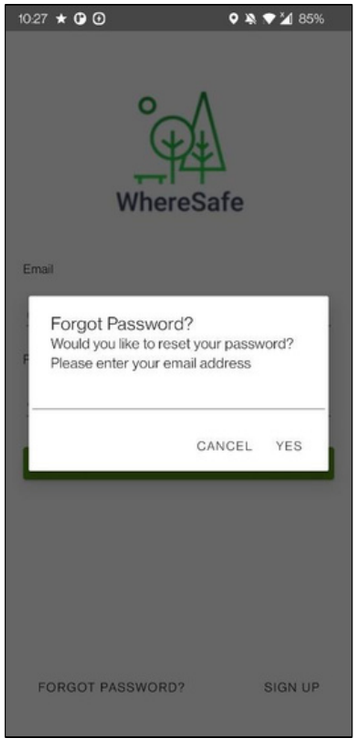
Figure 1 - Wireframe of WhereSafe App

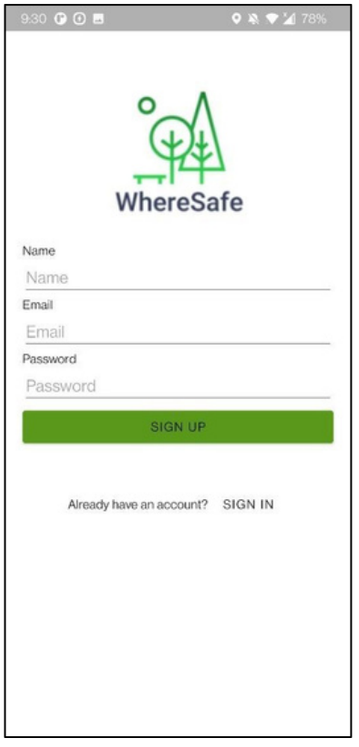
From the login screen, users will choose their login method or sign up for a new account. Once logged in, they will have access to features integrated into the hardware device such as location services or data displays. At this early stage, the team has focused their efforts on the data display feature since this can help confirm that the unit is functioning properly for high-level analysis purposes.

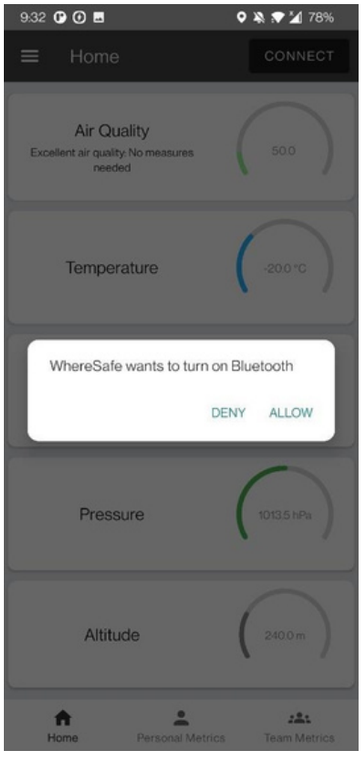
Having wrapped up Sprint 3, our login screen has a few key features. The user can now sign up with their email and a custom password to login into their specific account. The prompts ensure that any wrong input will be declined. In addition, the user can also reset their password by using the “forgot your password” feature. Finally, we have added some design changes to match our initial design.

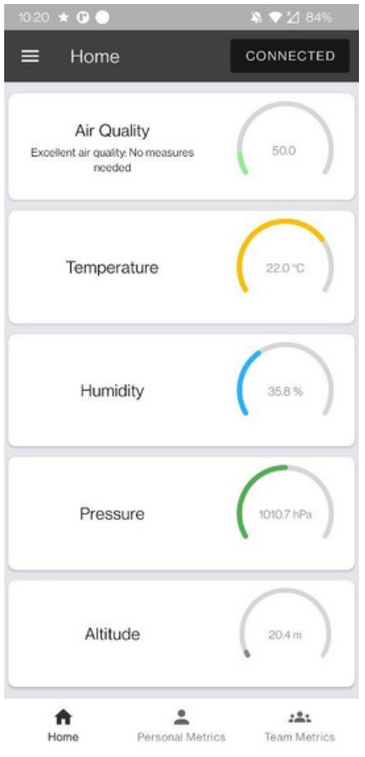
5.1.2 Screenshots

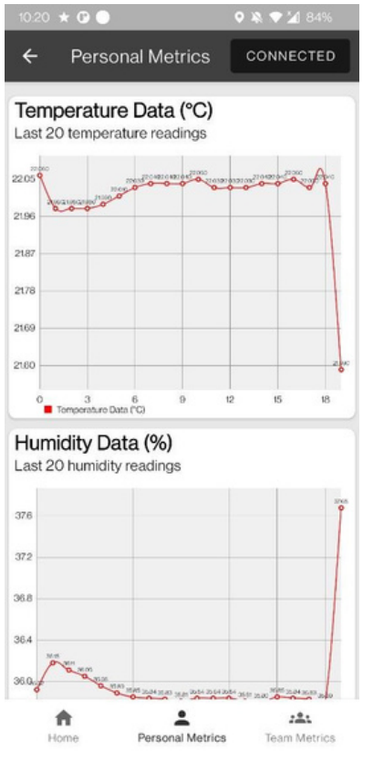


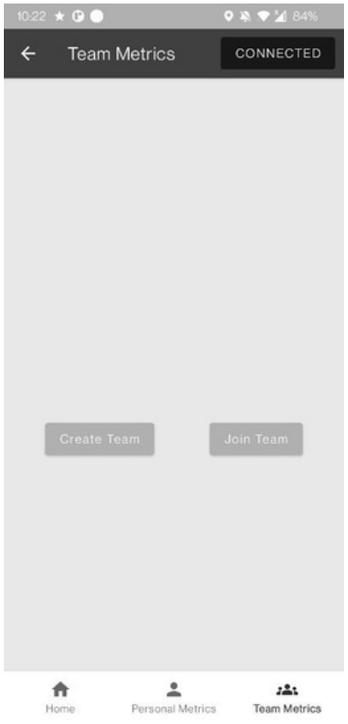


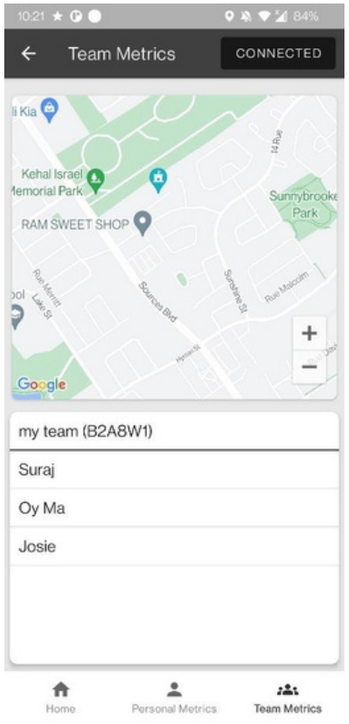


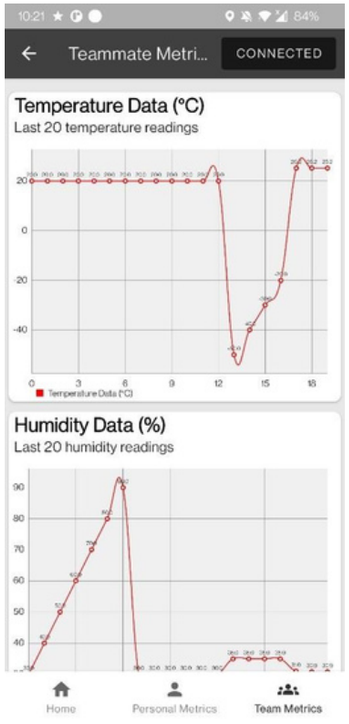

Home
CONNECT
Air Quality
Excellent air quality. No measures needed
50.0
Temperature
-20.0 °C
WhereSafe wants to turn on Bluetooth
DENY
ALLOW
Pressure
1013.5 hPa
Altitude
240.0 m
Home
Personal Metrics
Team Metrics

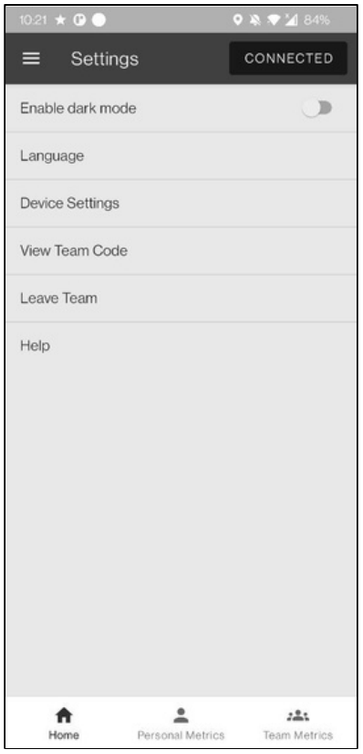

Home
CONNECTED
Air Quality
Excellent air quality. No measures needed
50.0
Temperature
22.0 °C
Humidity
35.8 %
Pressure
1010.7 hPa
Altitude
20.4 m
Home
Personal Metrics
Team Metrics

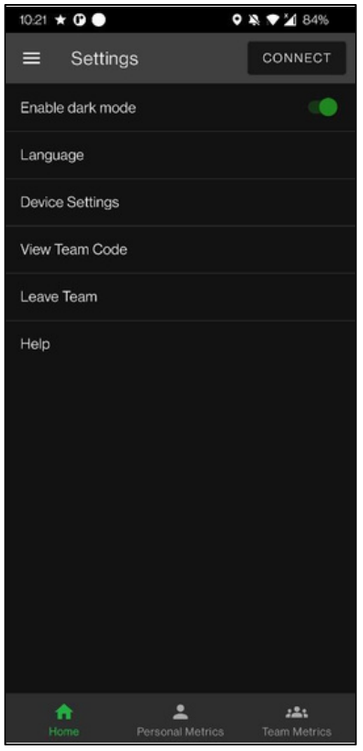

Personal Metrics
CONNECTED
Temperature Data (°C)
Last 20 temperature readings
Humidity Data (%)
Last 20 humidity readings
Home
Personal Metrics
Team Metrics

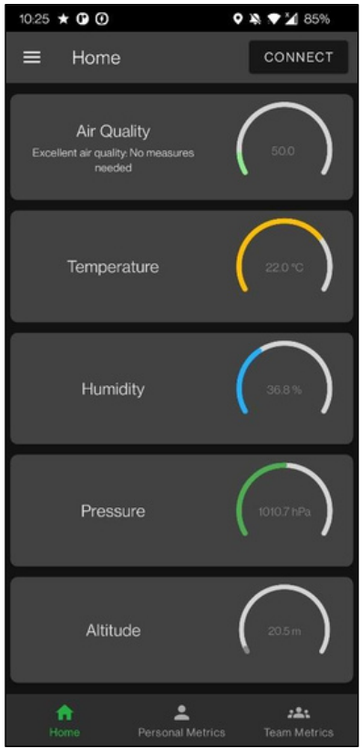

Team Metrics
CONNECTED
Create Team
Join Team
Home
Personal Metrics
Team Metrics


Team Metrics
CONNECTED
RAM SWEET SHOP
Sunnybrook Park
Google
my team (B2A8W1)
Suraj
Oy Ma
Josie
Home
Personal Metrics
Team Metrics

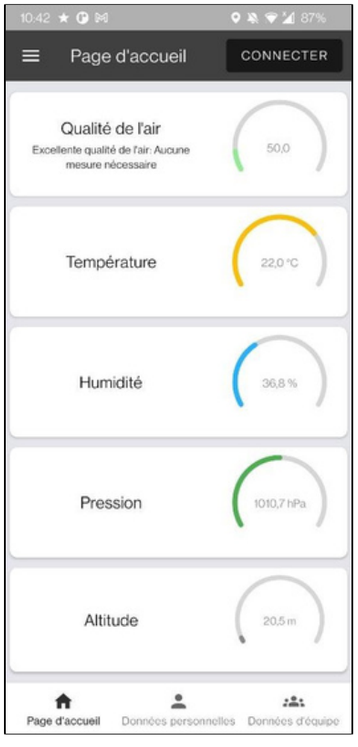

Teammate Metri...
CONNECTED
Temperature Data (°C)
Last 20 temperature readings
Humidity Data (%)
Last 20 humidity readings
Home
Personal Metrics
Team Metrics

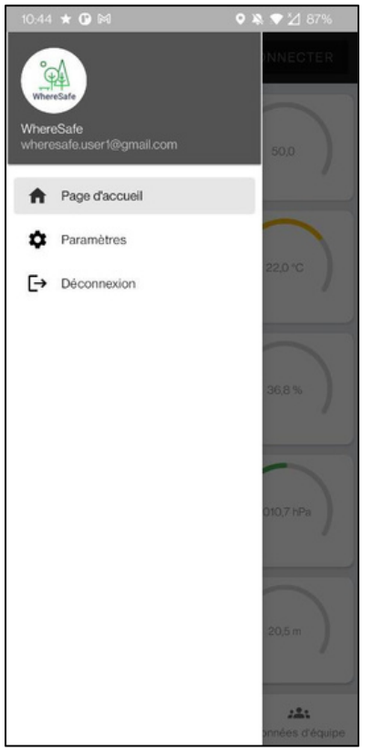

Settings
CONNECTED
Enable dark mode
Language
Device Settings
View Team Code
Leave Team
Help
Home
Personal Metrics
Team Metrics


Settings
CONNECT
Enable dark mode
Language
Device Settings
View Team Code
Leave Team
Help
Home
Personal Metrics
Team Metrics


Home
CONNECT
Air Quality
Excellent air quality. No measures needed
50.0
Temperature
22.0 °C
Humidity
36.8 %
Pressure
1010.7 hPa
Altitude
20.5 m
Home
Personal Metrics
Team Metrics


Paramètres
CONNECTER
Activer le mode sombre
Langue
Paramètres de l'appareil
Voir le code d'équipe
Langue
English
Français
ANNULER
Page d'accueil
Données personnelles
Données d'équipe


Page d'accueil
CONNECTER
Qualité de l'air
Excellente qualité de l'air. Aucune mesure nécessaire
50,0
Température
22,0 °C
Humidité
36,8 %
Pression
1010,7 hPa
Altitude
20,5 m
Page d'accueil
Données personnelles
Données d'équipe


WhereSafe
WhereSafe
wheresafe.user1@gmail.com
Page d'accueil
Paramètres
Déconnexion

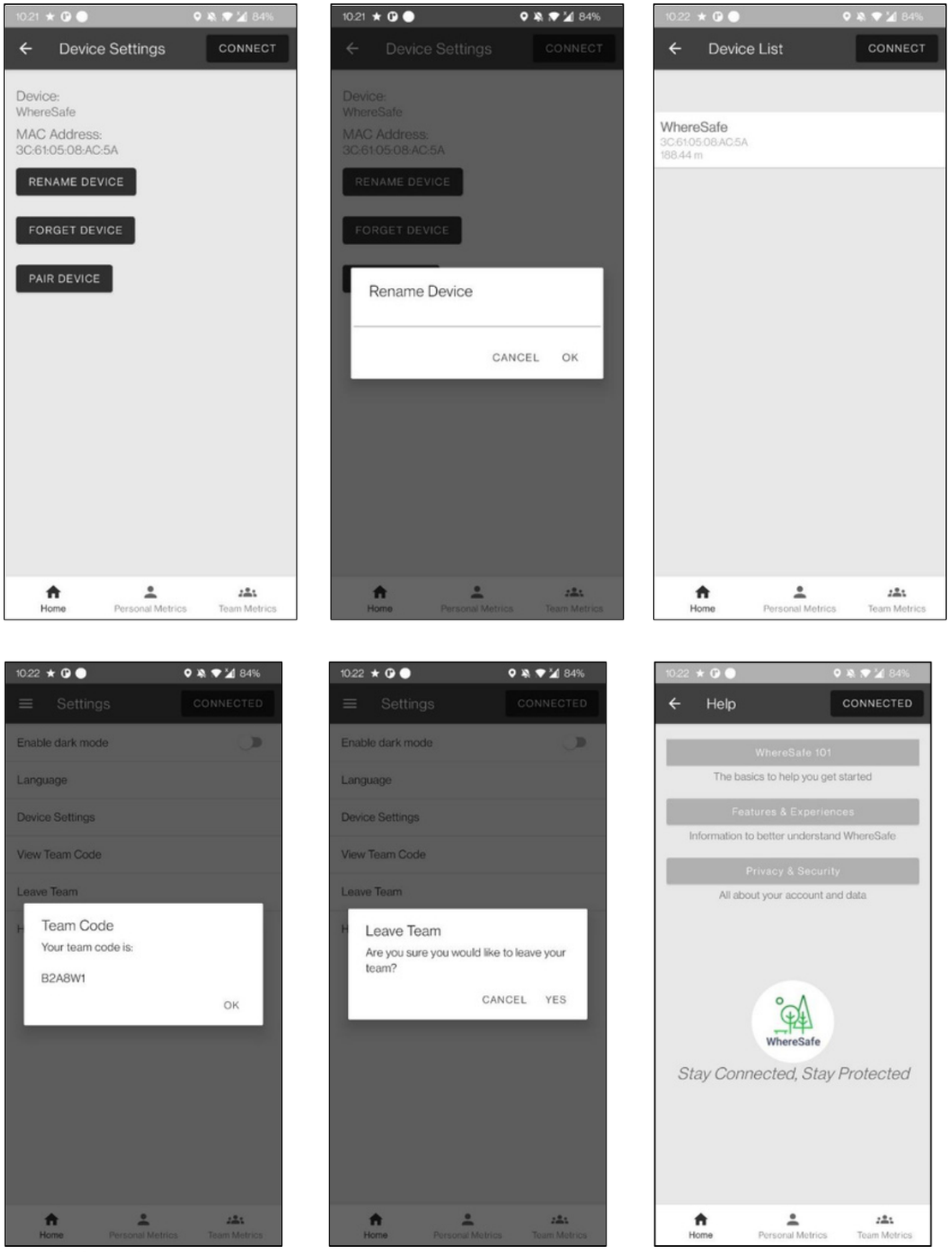


Figure 2 - Screenshots

5.2 System Architecture

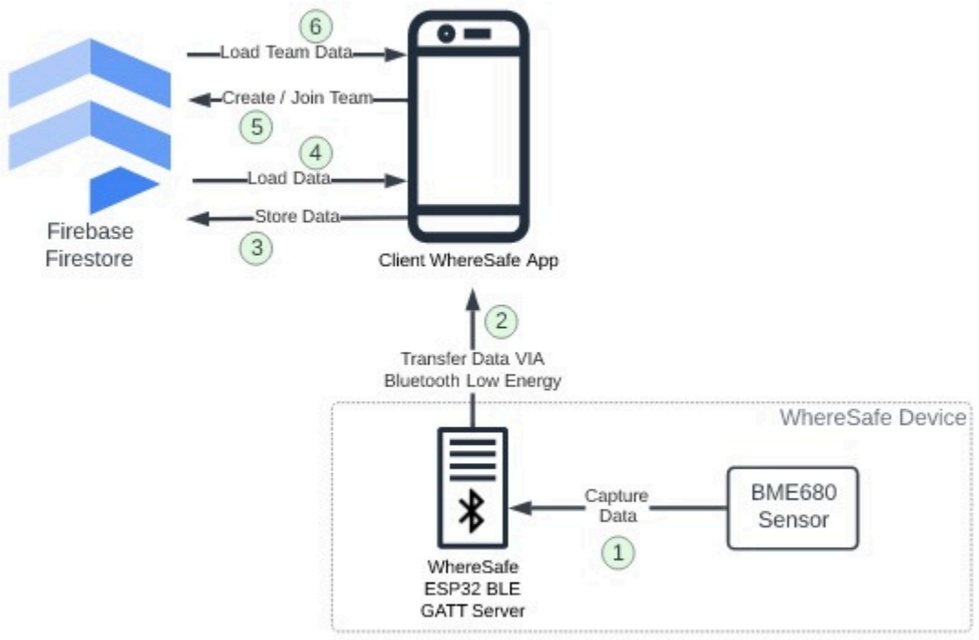


Figure 3 - Simplified System Architecture

1. The BME680 sensor captures temperature, pressure, humidity, and VOC gas. All the environmental data is read by the ESP32 device.
2. The ESP32 device is set up as a GATT server and transfers the captured data over Bluetooth Low Energy to the android device, which serves as the GATT client.
3. The WhereSafe android app parses the received data and encapsulates it as an object before storing it into the Firebase Firestore database.
4. The android app queries the database to retrieve the stored BME680 data before displaying it on the app.
5. The user can create or join a team, this information is stored within the firebase database.
6. The app queries the firebase database to get team information and team members' data.

Previously, the user's data received from the WhereSafe device was stored within an SQLite database locally on the smartphone. However, to allow users to view other team members' data, a cloud storage solution was needed. Firebase Firestore was the solution we decided to implement. It can store data on the cloud so that the user can access their data from anywhere as well as view team members' data. Firestore also enables the user to store data locally when an internet connection is unavailable, and the data is later pushed to the cloud whenever an internet connection is available.

The creation of the team is done through the app by providing a team name. The team creation process will generate a 6-digit alphanumeric code that will be used by others to join the team.

5.3 Hardware Architecture

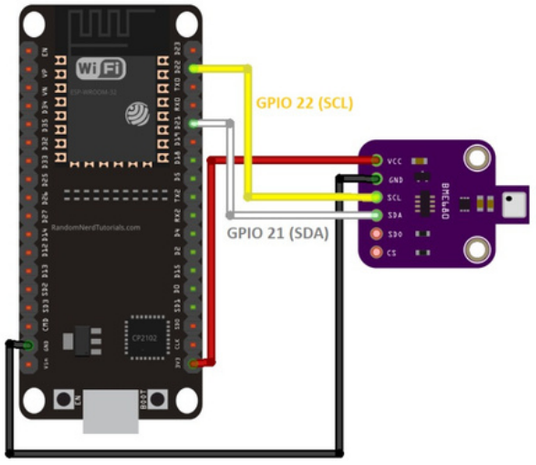


Figure 4 – Sample SparkFun ESP32 Thing / Adafruit BME680 wiring configuration (https://randomnerdtutorials.com/micropython-bme680-esp32-esp8266/)

At this stage the BME 680 is the sole sensor sending data to the development board, a SparkFun ESP32 Thing. The sensor captures ambient temperature, barometric pressure, relative humidity, and environmental VOC concentration. The captured data is read by the ESP32 over I2C. This data is then encapsulated and transmitted to the android device over Bluetooth Low Energy.

5.4 Software Architecture

5.4.1 Software Component

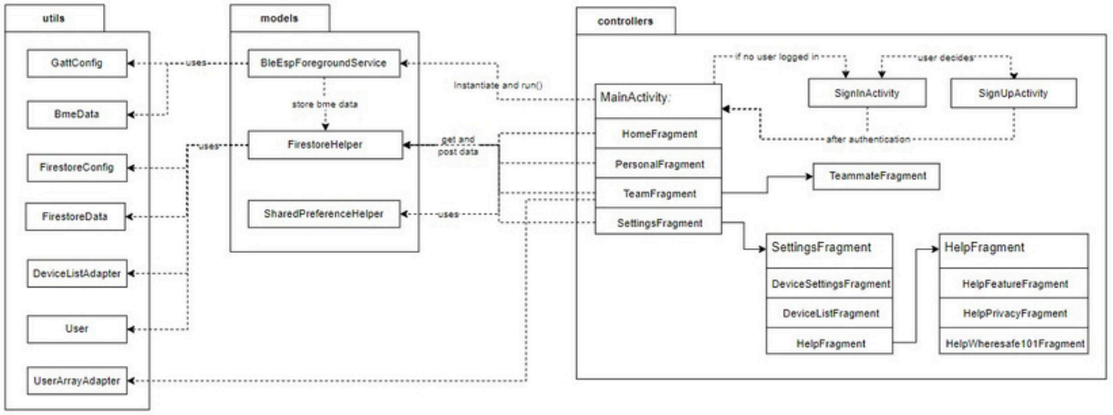


Figure 5 – CMU View Diagram

When the app is launched, the MainActivity is started and checks if a user is currently logged in. If a user is not logged in, they get redirected to the SignInActivity, which allows the user to sign in. If the user does not have an account, they can use the “Sign Up” button to navigate to the SignUpActivity. When the user successfully signs up or signs in, the user is redirected to the MainActivity and the app starts the main part of the application.

To connect the smartphone with the ESP32 device over Bluetooth Low Energy, the MainActivity has a connect button that instantiates the BleEspForegroundService class, which is responsible for establishing and maintaining the connection with the GATT server on the ESP32. This service runs as a foreground service so it can keep running even while the phone is asleep and shows a persistent notification for it. Once the connection is established, the app listens for changes in the GATT server characteristics. When changes occur, the characteristic values are parsed, encapsulated, and stored into the Firestore database using the helper functions implemented in the FirestoreHelper.

To ensure that the Bluetooth connection remains stable, the app is designed to use a single activity throughout the navigation of the app, the MainActivity, which maintains the Bluetooth connection throughout the app's lifecycle. This approach enables the app to continuously

monitor the device's characteristics without interruption. Additionally, the MainActivity is divided into various fragments, each implementing a different part of the application. This design allows for seamless navigation between different features and user interfaces while maintaining the Bluetooth connection established by the BleEspForegroundService.

When the Home page (HomeFragment) is displayed, the widgets are updated with the last stored document in the user's sensor_data collection. A listener is attached to the user's sensor_data collection which triggers an event whenever a new document is added to the collection, which allows the widgets to be updated accordingly in real time.

The Personal Metrics (PersonalFragment) displays charts with the last 20 recorded data points stored in the user's sensor_data collection. When the fragment is launched, the FirestoreHelper is used to query the firebase database and retrieve the most recent data.

The Team Metrics (TeamFragment) displays a map to show team members location and also a list of team members. The list of team members is clickable, and it takes the user to see their teammates personal data. If the user does not have a team, the user is shown the options to create a team or join an existing team using a team code.

Contollers is a package containing the activities and fragments of the application. The MainActivity handles all services and events throughout the app's lifecycle. The fragments are used to display user interfaces that can be dynamically changed and replaced during the runtime of the MainActivity. The SignUpActivity and SignInActivity are shown at startup to authenticate the user.

Models is a package containing the logic of the application. The BleEspForegroundService uses androids BluetoothLeScanner to establish a connection with the WhereSafe ESP32 device. The BleEspForegroundService class also receives data from the connection which is then stored into the Firestore database using the FirestoreHelper.

Utils is a package containing files to help instantiate the models and encapsulate data. The GattConfig file contains various predefined UUIDs that are necessary for the BleEspForegroundService class to read data from the GATT server running on the physical device.

When the data is read, it is encapsulated in a BmeData object before storing it into the database. The FirestoreConfig file stores strings for the setup of the database within the FirestoreHelper class.

5.4.2 Firestore Database Design

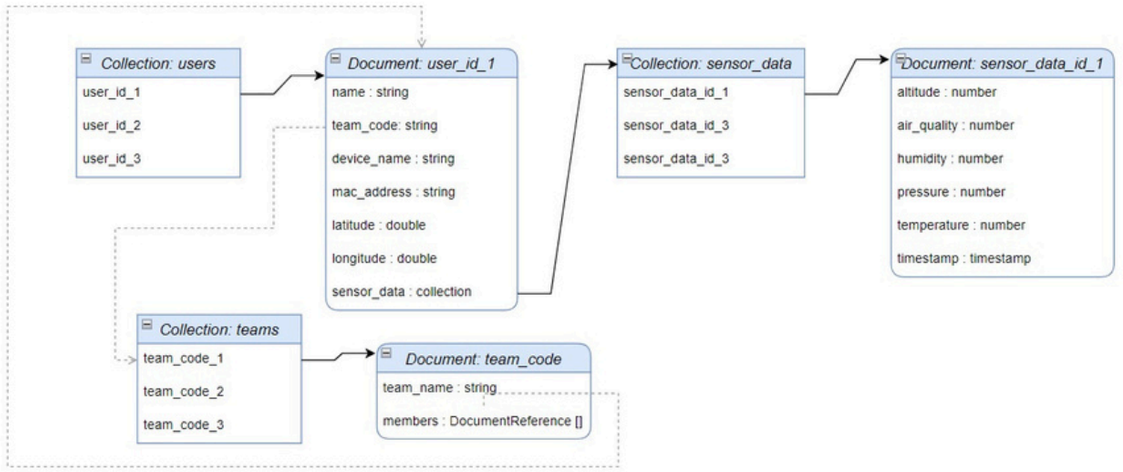


Figure 6 - Database Design

Firebase's Firestore is a NoSQL cloud database that allows our users to access their data from anywhere and they can view information in real-time as data is recorded by the hardware device.

The data is organized within two main collections, the "users" collection and the "teams" collection.

The "users" collection stores the documents associated with each user, which are identified by an auto-generated user ID. Each user document stores the user's name, a team code used to reference their team (generated when creating/joining a team), the device name and MAC address, the user's coordinates, and the "sensor_data" collection to store data received from the physical device.

The "sensor_data" collections contain all documents created to store data received from the WhereSafe device's environmental sensor. The sensor data document stores the altitude, the air quality value, the humidity, the pressure, the temperature, and the timestamp of the data entry.

The “teams” collection is made up of the documents for each team, which are identified by the generated team code. Each team document stores the team’s name, and a list of the document references to each team member.

There is a two-way relationship between each user and their team. Each user has a team code, which is associated with a team document, and each team document has references to users. This design allows us to easily query each user in a team and retrieve their data.

5.5 Communication Interfaces

5.5.1 Bluetooth Protocol

The WhereSafe app communicates with the ESP32 hardware device using Bluetooth Low Energy. BLE devices transfer data using a hierarchical data structure known as GATT (Generic Attribute). The ESP32 is set up as a GATT server and the android app acts as the GATT client. The server has its profile setup in the GATT structure shown in the figure below.

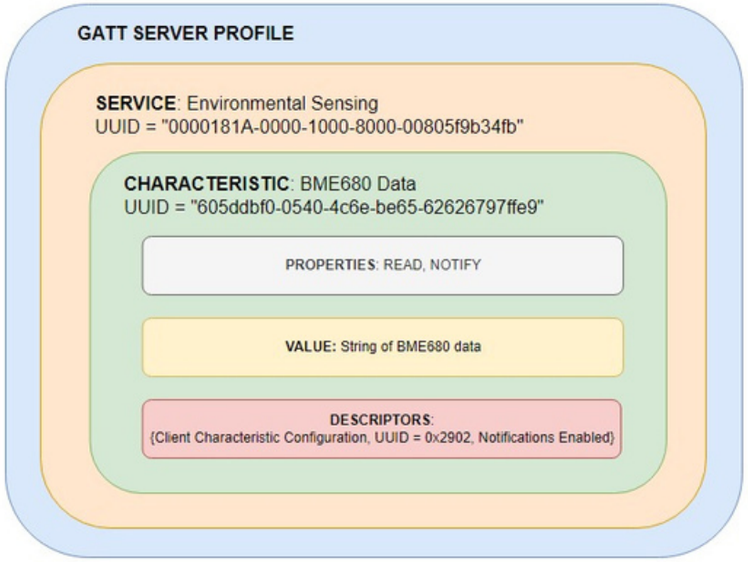


Figure 7 - GATT Layer Analysis

The official service for environmental sensing is used since the BME680 sensor provides data about the environmental surroundings. A custom characteristic is used to transmit the BME680

data within a single characteristic. The characteristic is set up so that it notifies the client device whenever the value of the characteristic is changed. Therefore, whenever a new reading is performed using the BME680 sensor, the BME680 Data characteristic value will be changed, and the android app will be notified of the change.

5.5.2 ESP32 Bluetooth Methods

Command: setValue

Structure: "{temperature}|{humidity}|{pressure}|{air quality}|{altitude}"

Field	Type	Description	Example
{temperature}	String value of float	Ambient temperature	25.38
{humidity}	String value of float	Relative humidity	31.76
{pressure}	String value of float	Barometric pressure	1013.1
{air quality}	String value of float	Air quality index value	50.0
{altitude}	String value of float	Altitude	49.00

5.5.3 Client-side Bluetooth Methods

On the client side of the BLE data transfer, the app needs to initiate the connection with the device before listening to changes in the provided services' characteristics.

Command: setCharacteristics

Structure: {BluetoothGatt}

Notes: Responsible for permissions handling and loading BME data (in the form of a Bluetooth characteristic) onto the descriptor

Command: readCharacteristics

Structure: {BluetoothCharacteristic}

Notes: Responsible for accessing incoming characteristics and helping pass data to database

Command: scanConnect

Structure: {BluetoothLeScanner, ScanFilter, ScanSettings}

Notes: Low-level interface for detection, connection, and transmission over Bluetooth.

Field	Type	Description	Example

“scanner”	BluetoothLeScanner	scanner object for Bluetooth LE applications	N/A (any such declared object works)
“filter”	ScanFilter	Back-end descriptor for locating desired devices with appropriate identifiers	DEVICE_NAME = “WhereSafe”
“settings”	ScanSettings	Back-end descriptor for selecting scan settings	scanSettings .setScanMode(LOW_POWER)

5.5.4 Low-Level Protocols and Methods

Interface	Implementation	Goal of Interface	Relevant Commands
Data capture	Physical	Capture environmental data and make available to ESP32	getTemperature() getPressure() getGas() getHumidity() getAltitude()
BLE Data Transfer	Bluetooth Low Energy	Pass data from electronics (hardware) to app (software)	setCharacteristics() readCharacteristics() scanConnect()
Local Data Handling	SQLite Database	Store and access data locally	storeData() insertBmeData() getBmeData()

Command: getTemperature
Structure: N/A
Notes: library method

Command: getPressure
Structure: N/A
Notes: library method

Command: getGas
Structure: N/A
Notes: library method

Command: getHumidity
Structure: N/A
Notes: library method

Command: getAltitude
Structure: N/A
Notes: library method

Command: storeData
Structure: N/A
Notes: interfaces with (DatabaseHelper).insertBmeData (below)

Command: insertBmeData
Structure: {BmeData}
Notes: Unpacks BMEData object and inserts data into SQLite database.

Command: getBmeData
Structure: N/A
Notes: Packetizes database table information and returns a BMEData object

5.6 UML Diagrams

5.6.1 Use Case Diagram

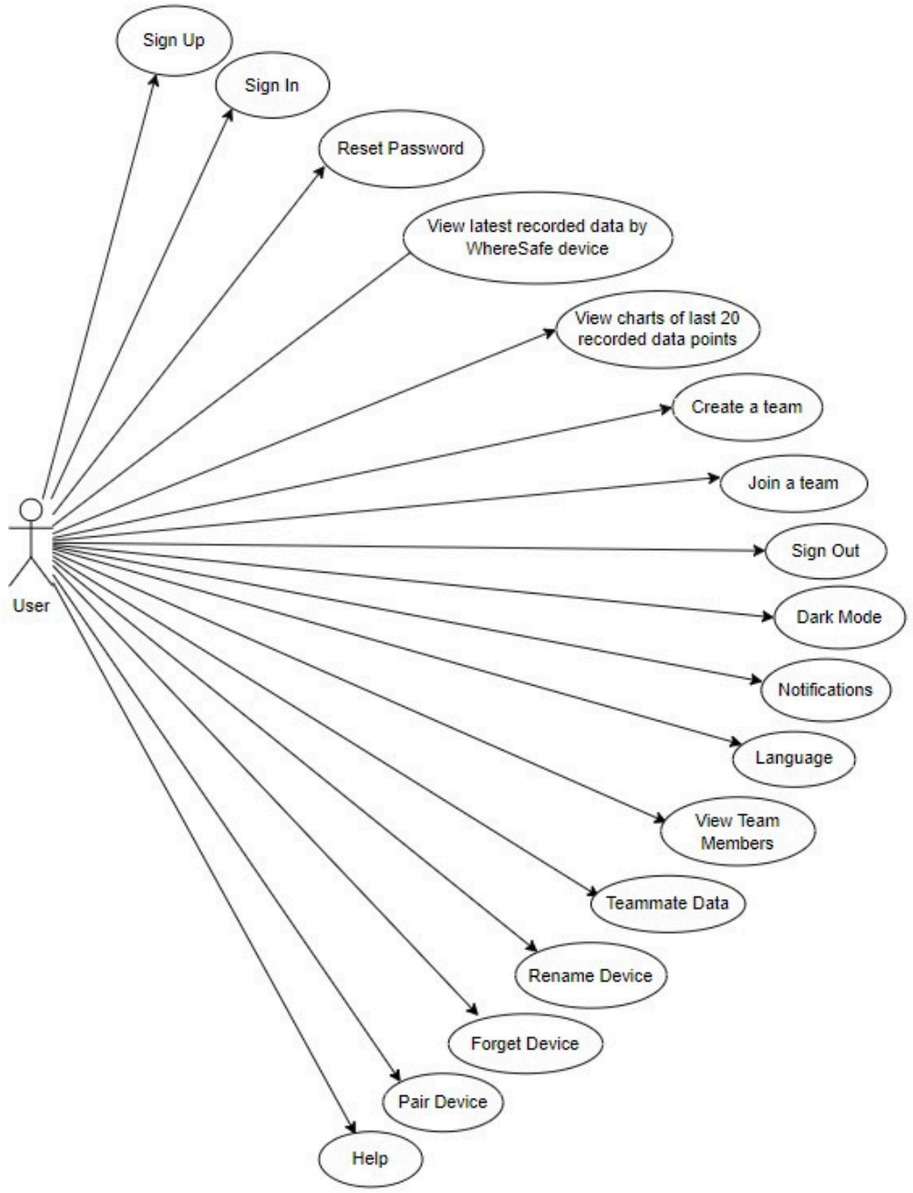


Figure 8 - Use Case Diagram

5.6.2 Sequence Diagrams

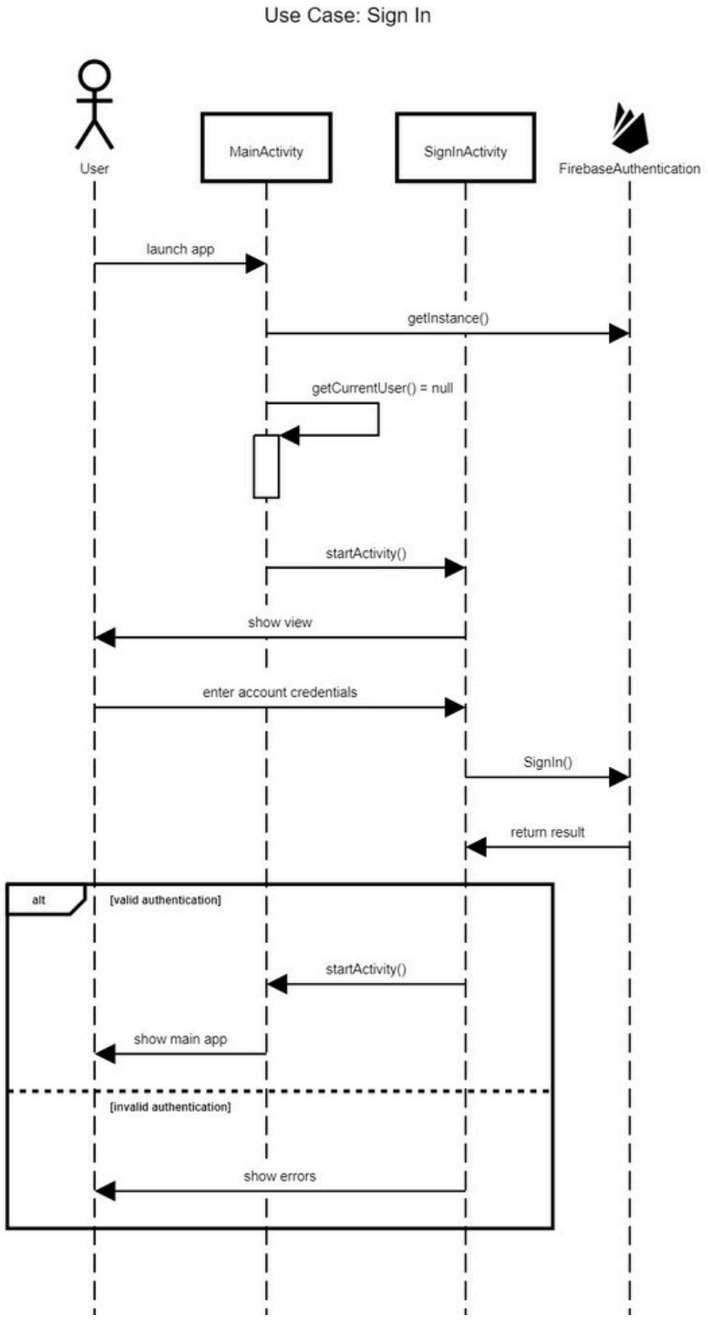


Figure 9 - Sequence Diagram: Sign In

The following diagram skips the beginning part of the app startup which is already shown in the sign in sequence diagram.

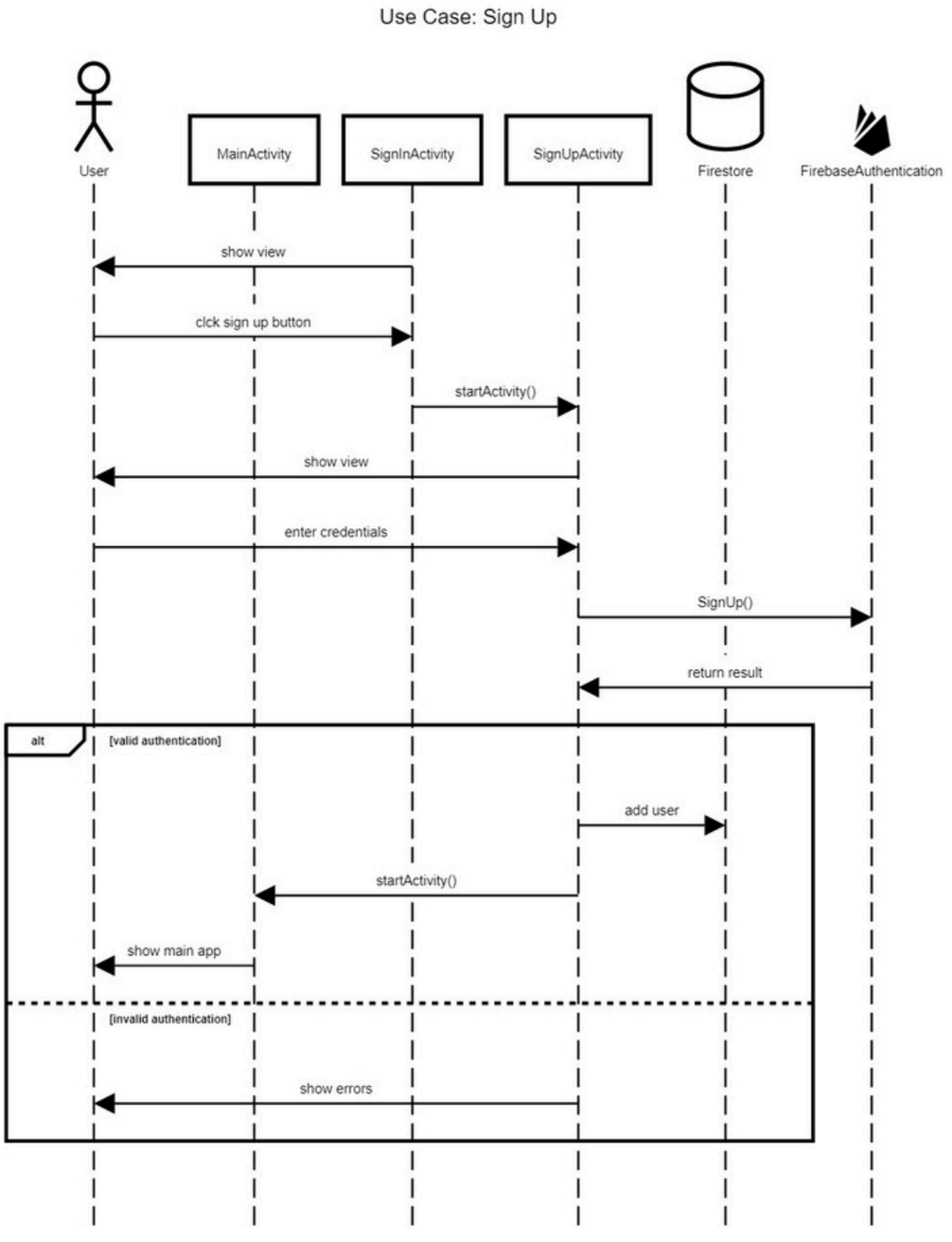


Figure 10 - Sequence Diagram: Sign Up

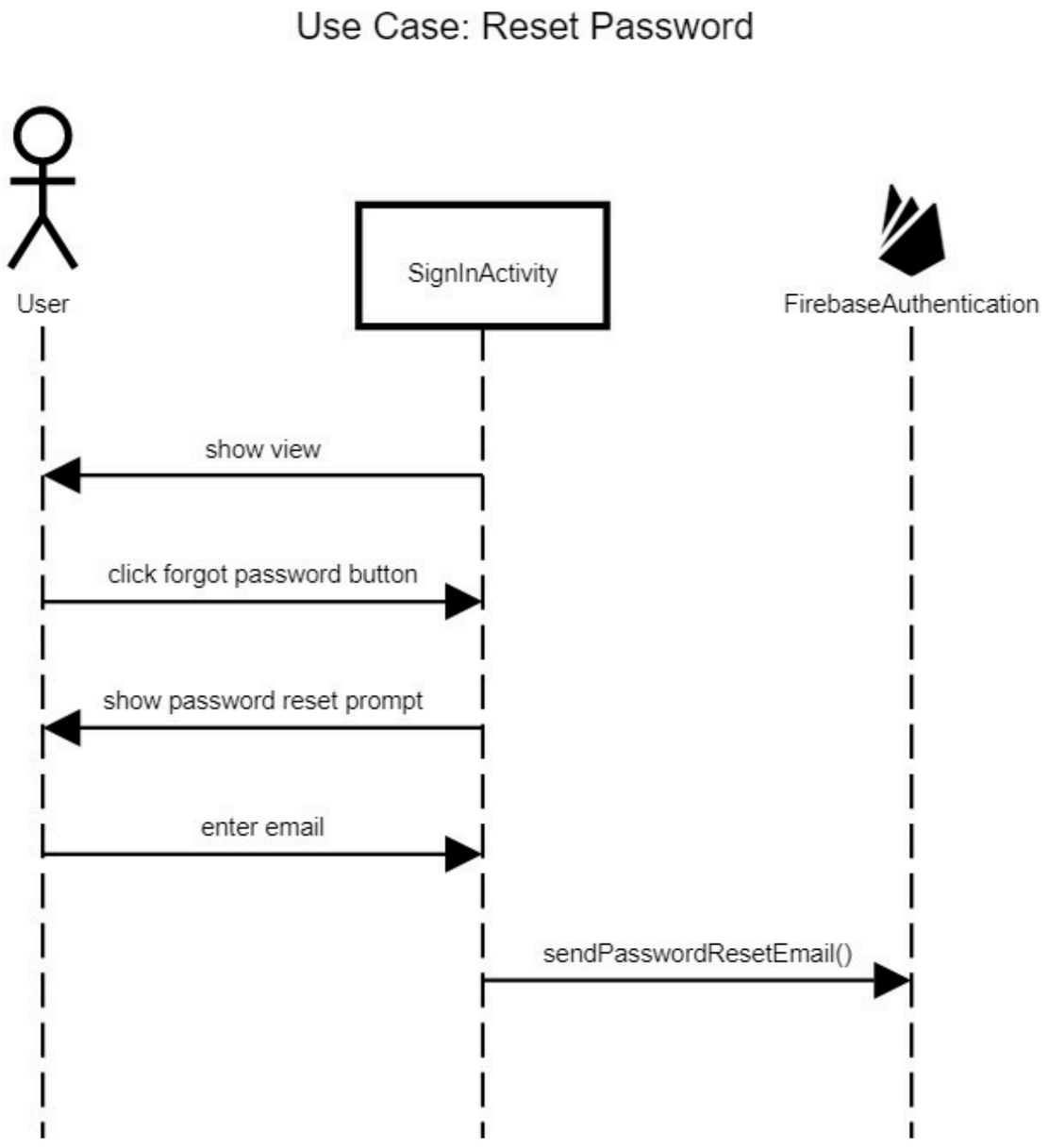


Figure 11 - Sequence Diagram: Reset Password

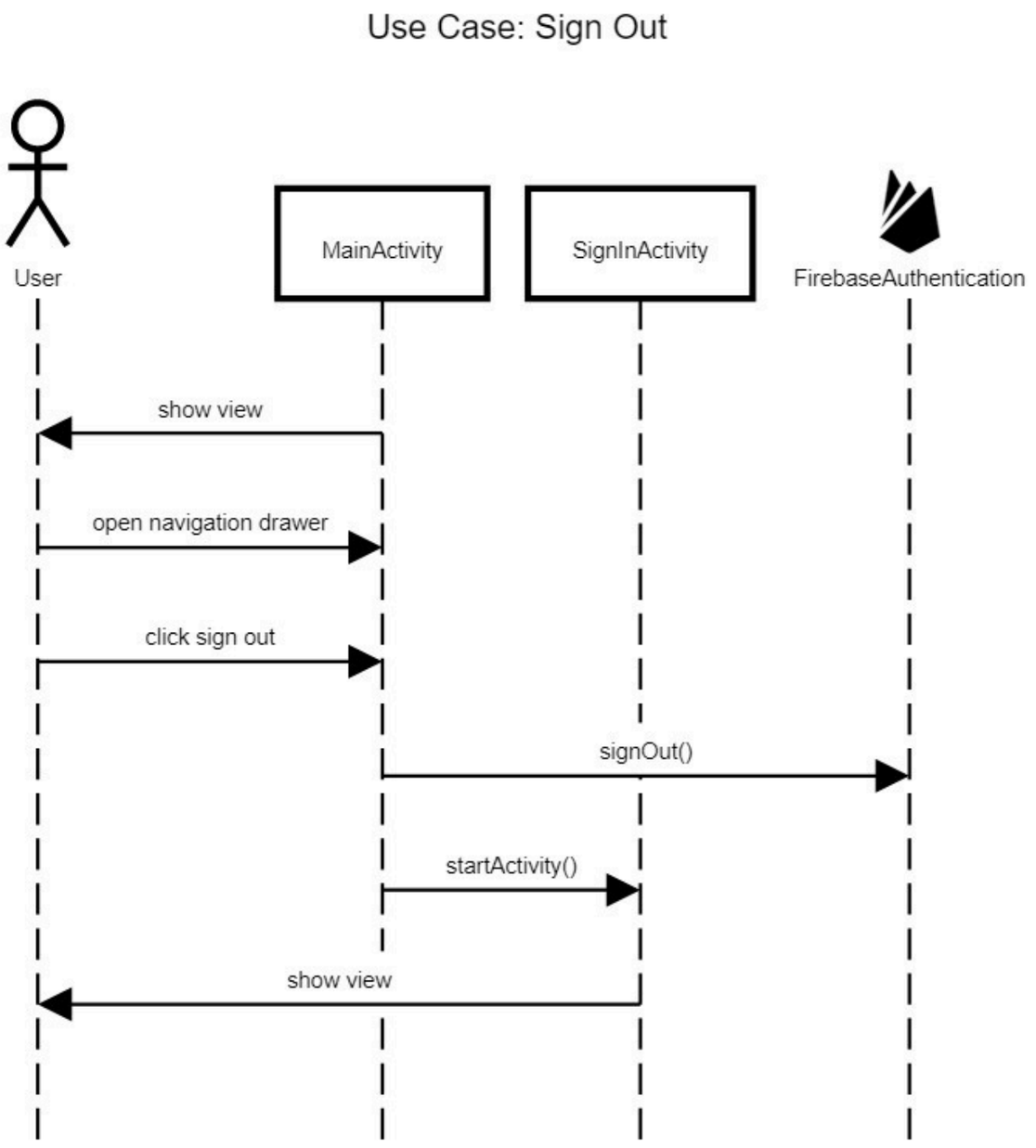


Figure 12 - Sequence Diagram: Sign Out

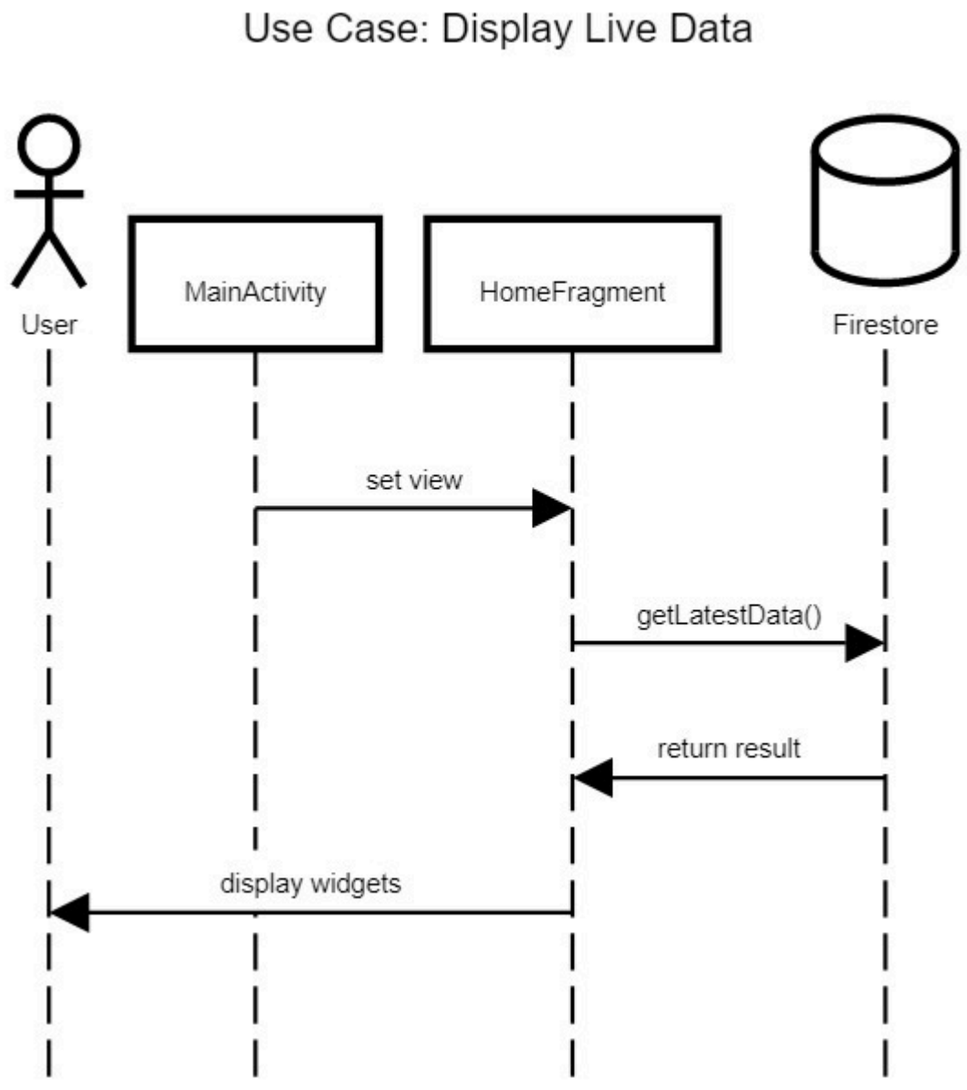


Figure 13 - Sequence Diagram: Display Live Data

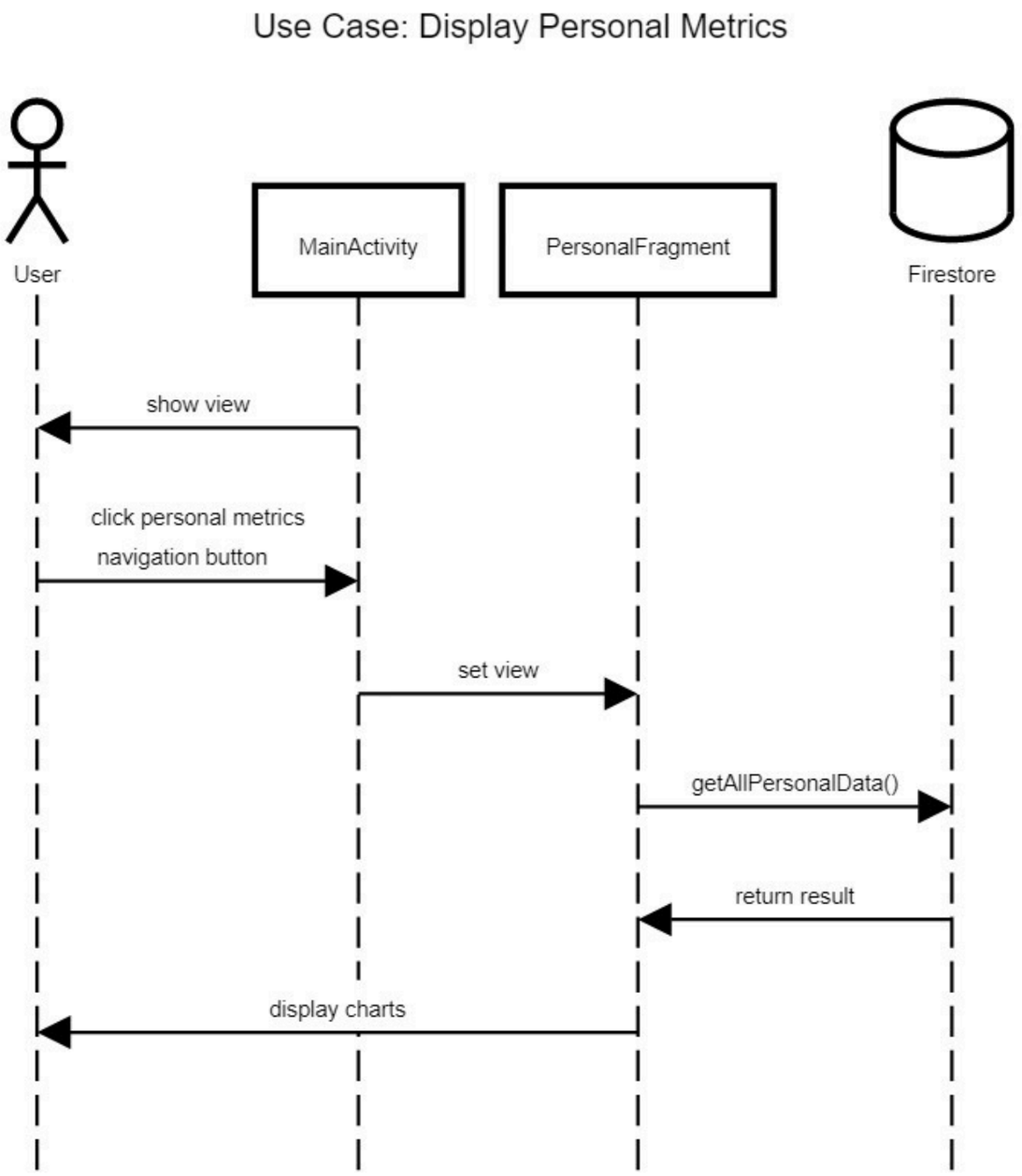


Figure 14 - Sequence Diagram: Display Personal Metrics

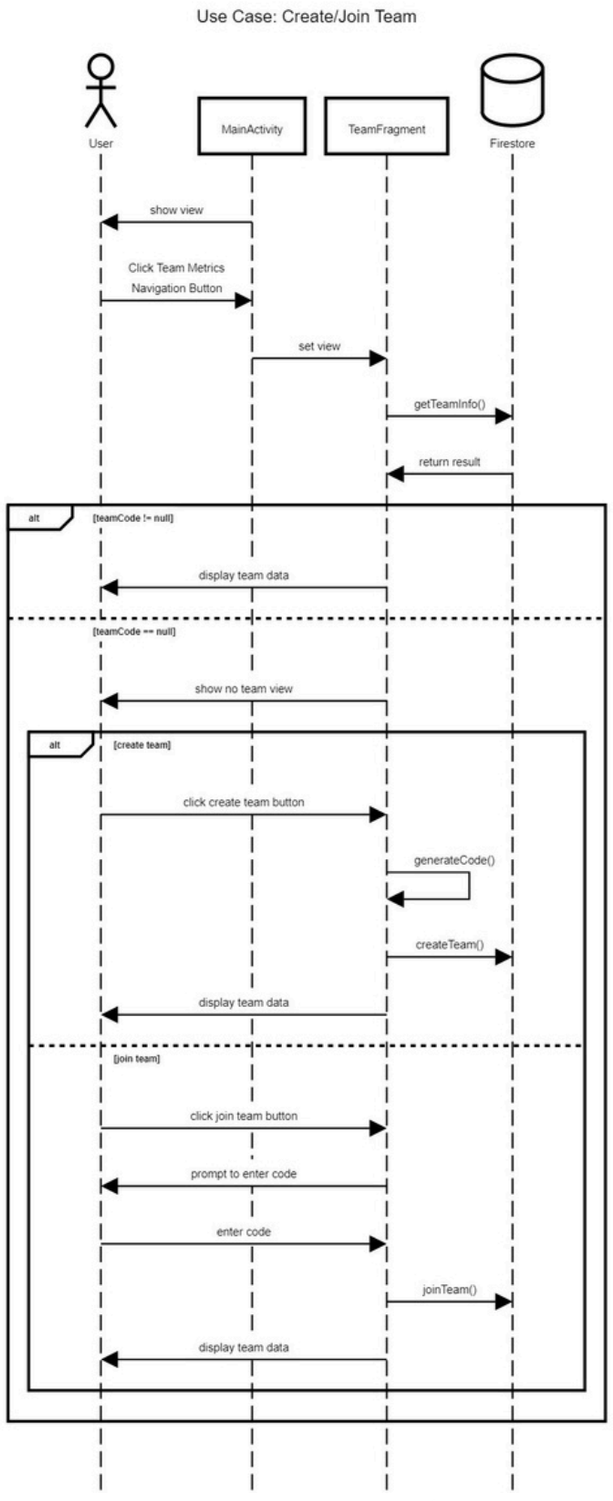


Figure 15 - Sequence Diagram: Create/Join Team

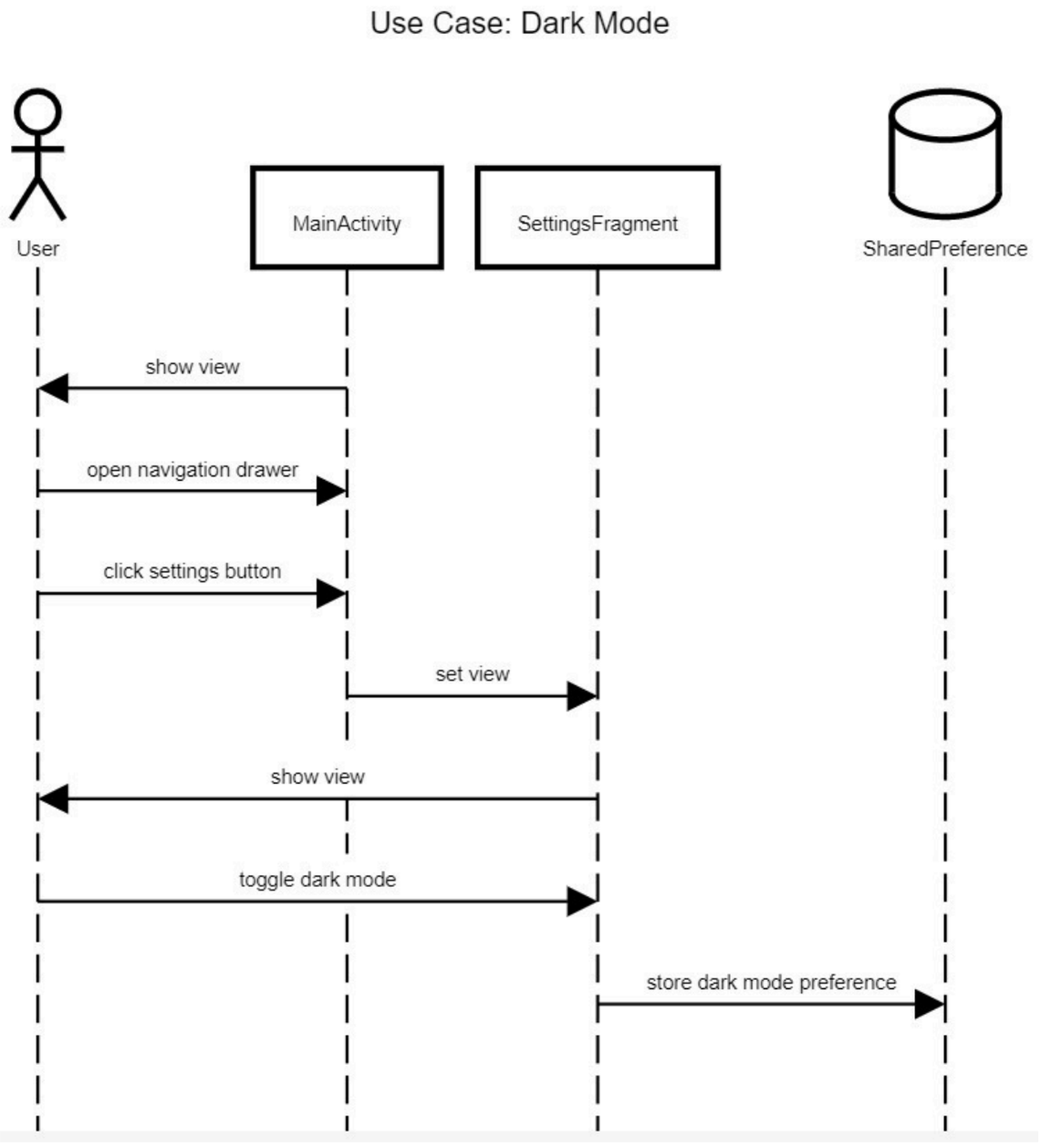


Figure 16 - Sequence Diagram: Dark Mode

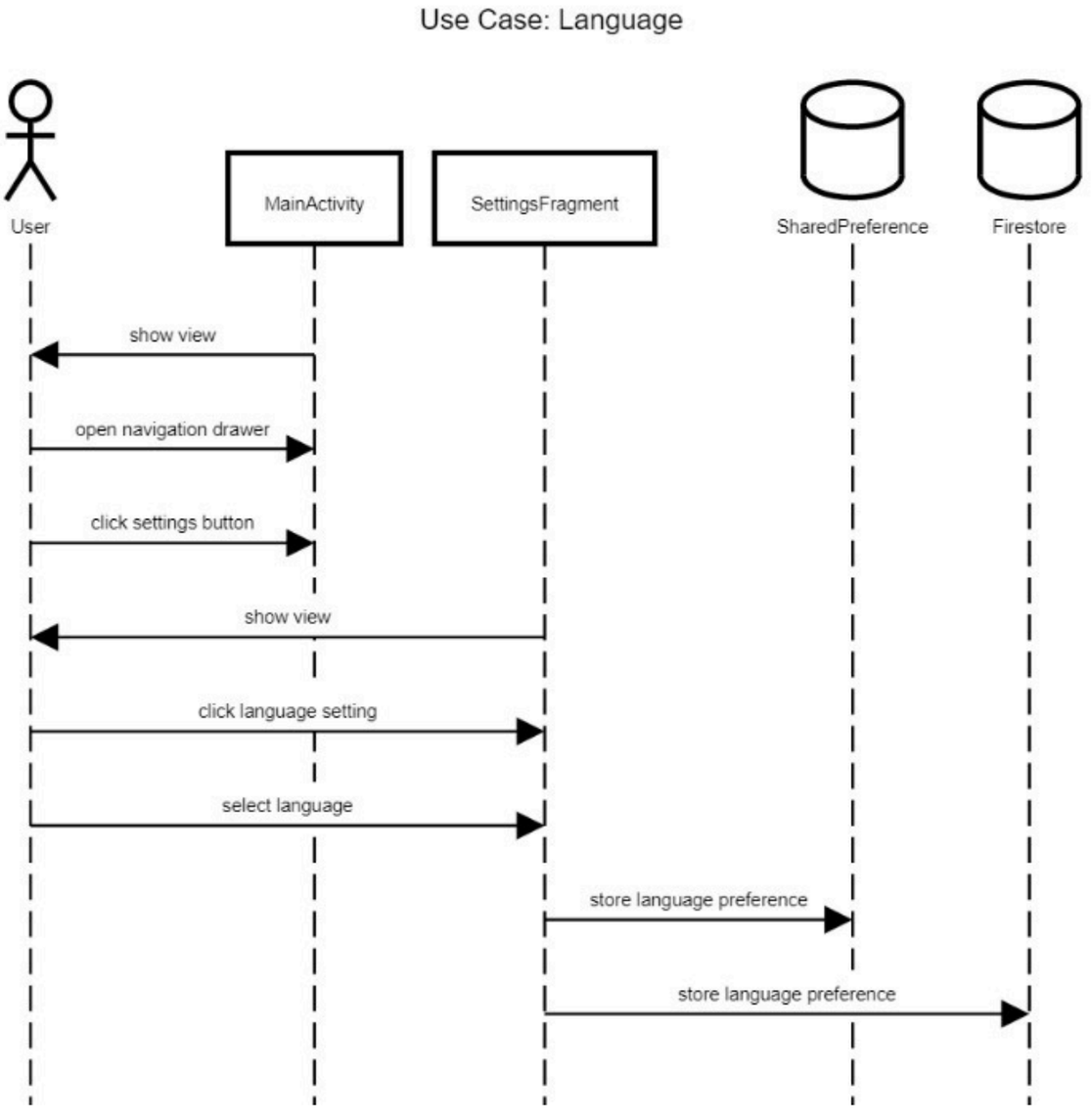


Figure 17 - Sequence Diagram: Language

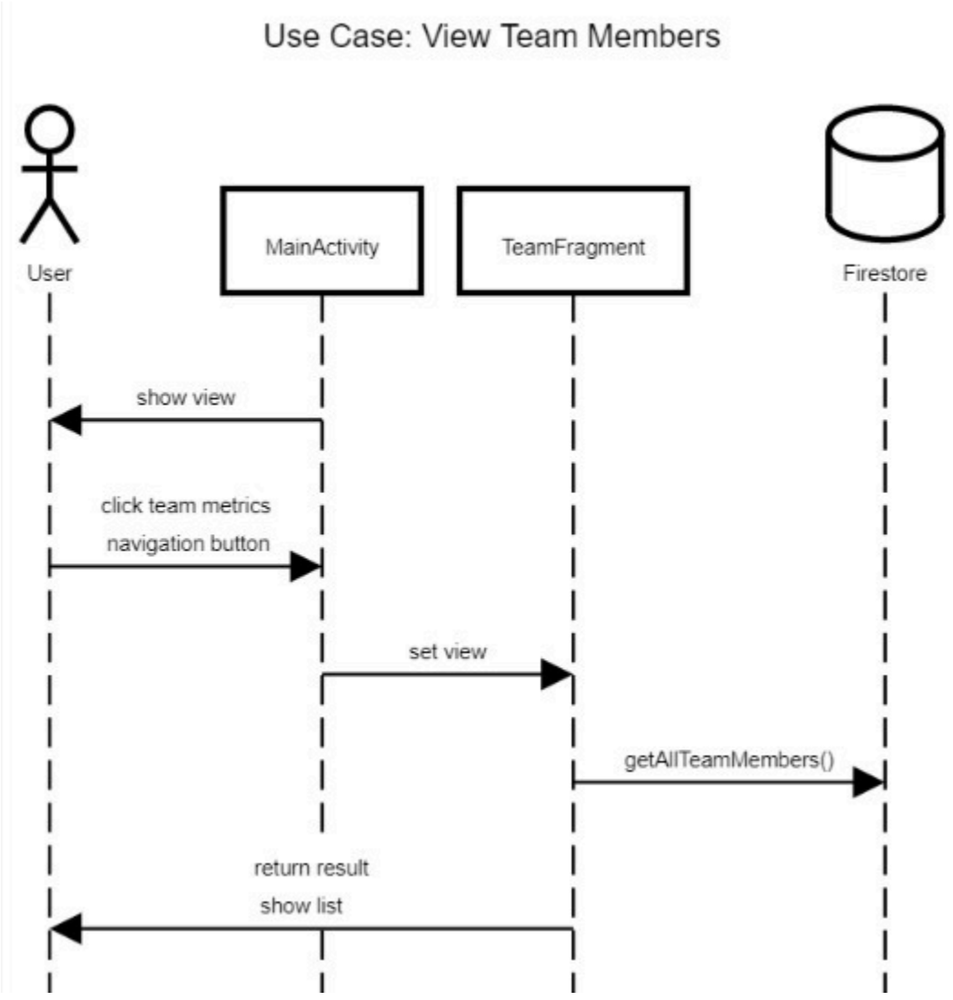


Figure 18 - Sequence Diagram: View Team Members

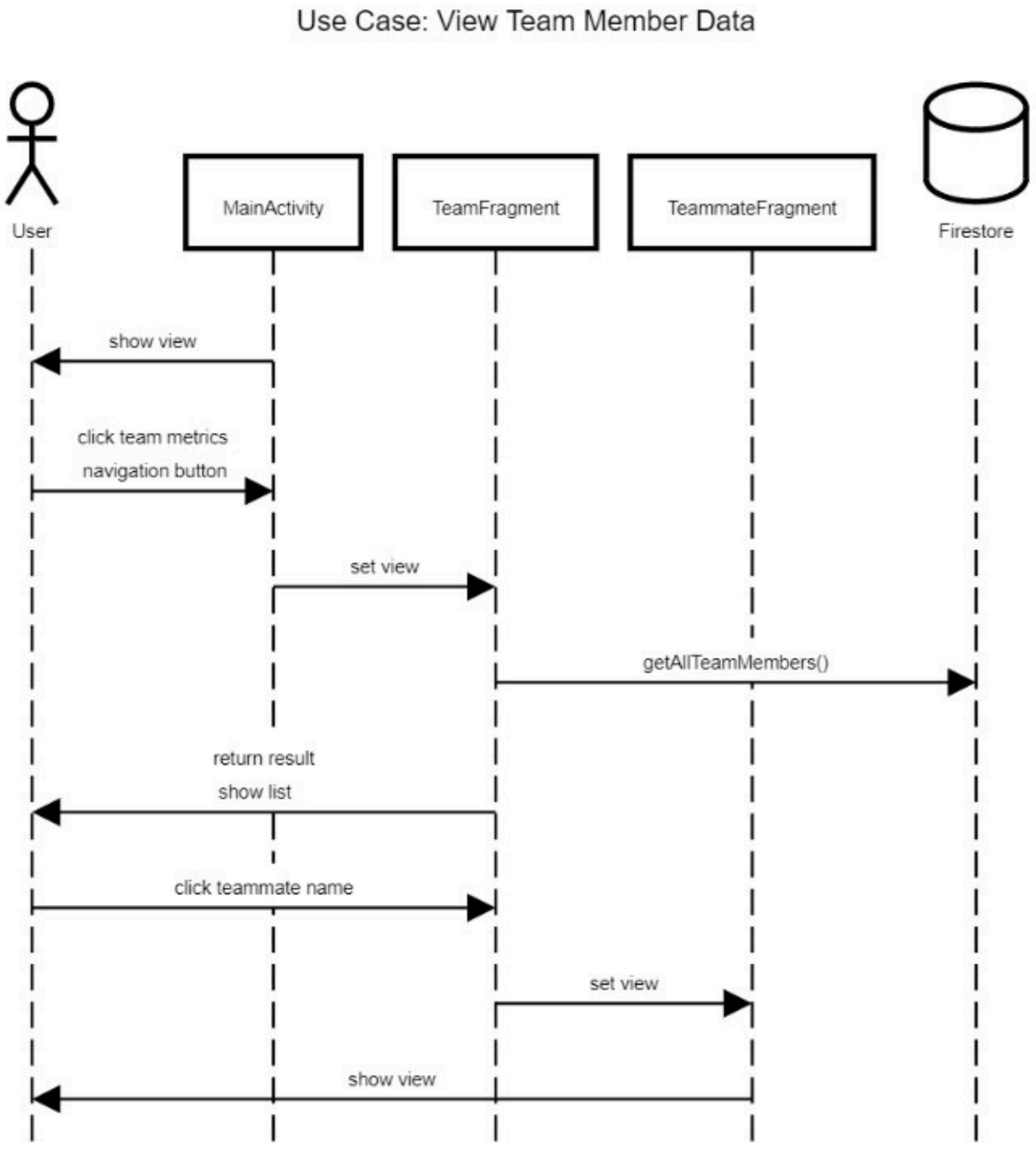


Figure 19 - Sequence Diagram: View Team Member Data

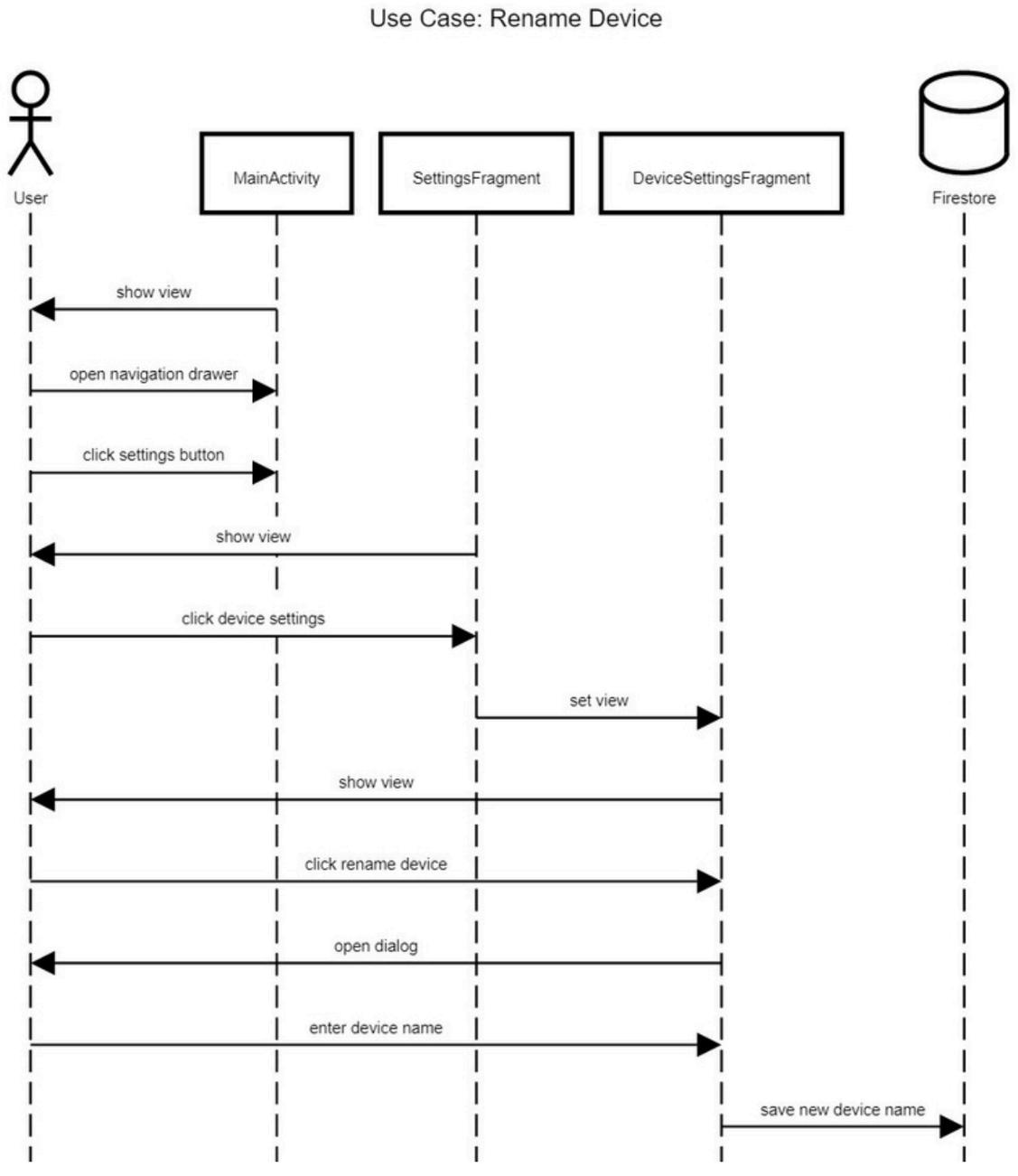


Figure 20 - Sequence Diagram: Rename Device

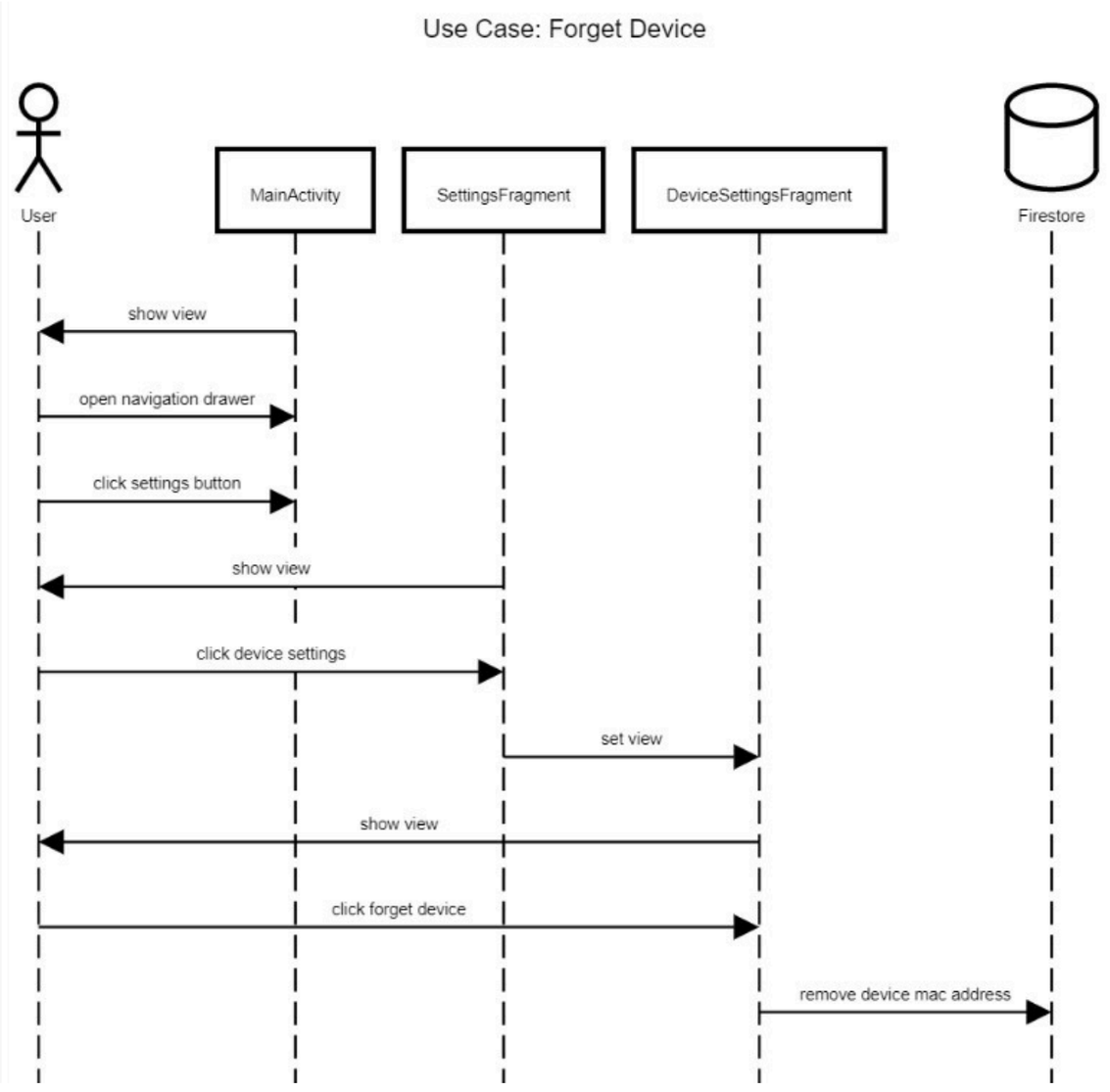


Figure 21 - Sequence Diagram: Forget Device

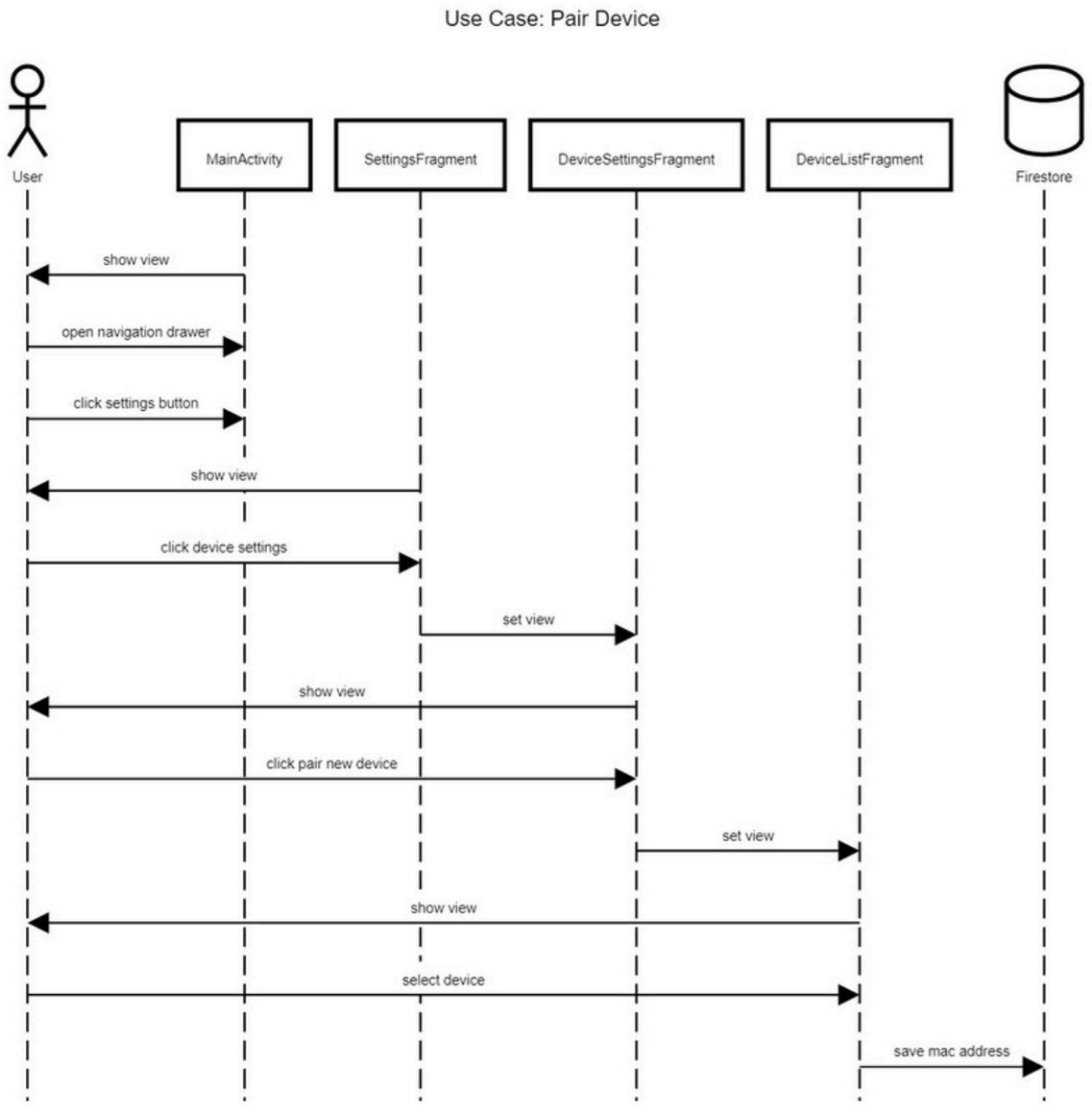


Figure 22 - Sequence Diagram: Pair Device

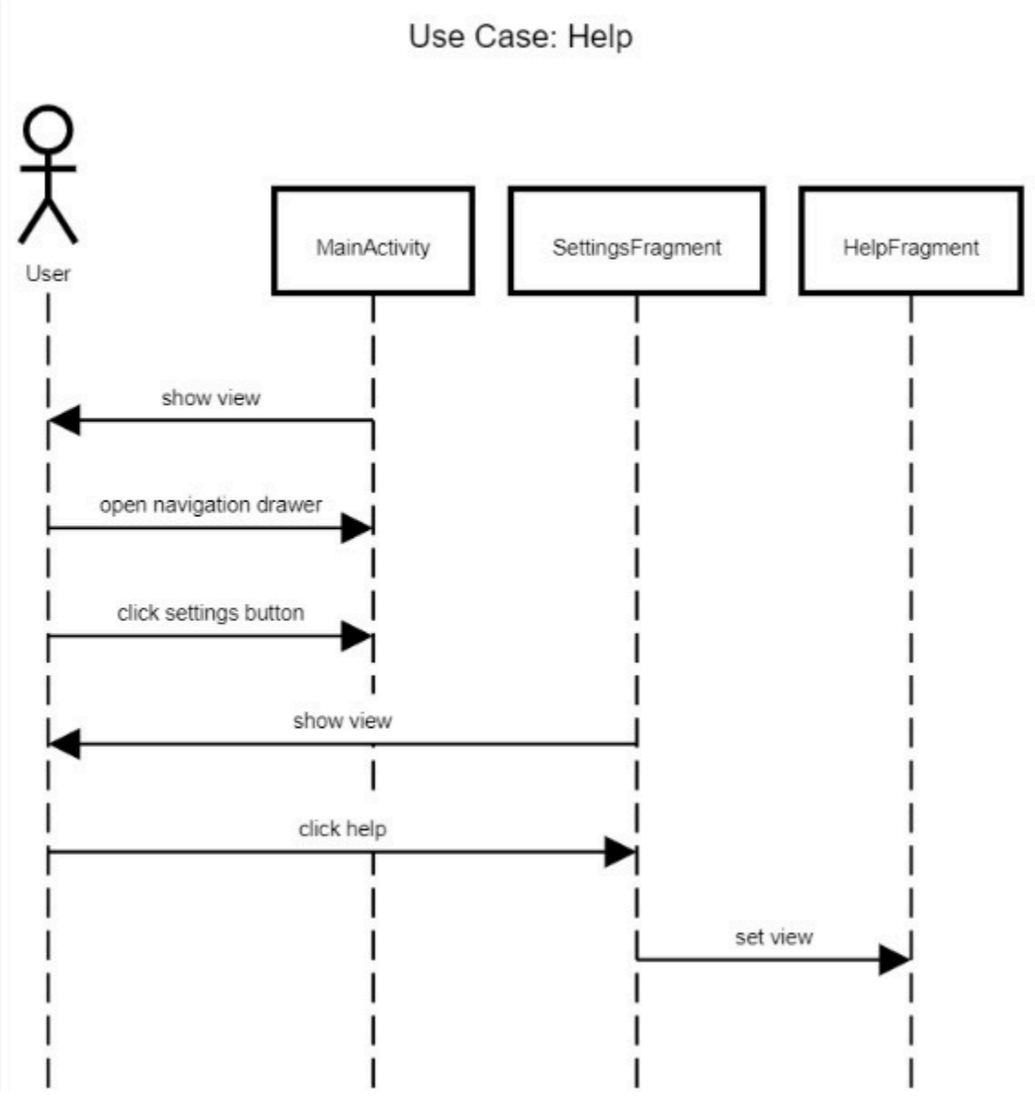


Figure 23 - Sequence Diagram: Help

6 Testing

The WhereSafe handles data from several sources and sinks, simultaneously creating and feeding packets for use in various parts of the application. Therefore, testing must ultimately account for each of four steps along the way: collection of the raw environmental data, processing that data at the device level, transmission of the data to the application, and data handling / display procedures.

6.1 Test Plan 1: Data Collection and Transmission

6.1.1 Summary

Story ID: DA-1, DA-2, DA-3, DA-6, DA-7

The user should be able to activate the sensors and receive environmental data on the application through the device.

6.1.2 Test Cases

Test Case: 1.1		
Pre-Condition: Code written to link sensor, establish PC serial connection, and display incoming analog data. Code is uploaded to board. Gas sensor is connected to board.		
Steps:	Expected Results:	Actual Results:
1. Connect board to PC via USB and open serial monitor	"ready" pattern on LED shows code is uploaded and board is operational	As expected
2. Activate gas sensor by providing power to circuit	normal sensor functionality	
3. Monitor incoming analog data	environmental data visible on serial monitor	
Result	PASS	

Test Case: 1.2		
Pre-Condition: Code written to link sensor, establish PC serial connection, and display incoming analog data. Code is uploaded to board. Temperature sensor is connected to board.		
Steps:	Expected Results:	Actual Results:
1. Connect board to PC via USB and open serial monitor	"ready" pattern on LED shows code is uploaded and board is operational	As expected
2. Activate temperature sensor by providing power to circuit	normal sensor functionality	
3. Monitor incoming analog data	environmental data visible on serial monitor	
Result	PASS	

Test Case: 1.3		
Pre-Condition: Code written to link sensor, establish PC serial connection, and display incoming analog data. Code is uploaded to board. Pressure sensor is connected to board.		
Steps:	Expected Results:	Actual Results:
1. Connect board to PC via USB and open serial monitor	"ready" pattern on LED shows code is uploaded and board is operational	As expected
2. Activate pressure sensor by providing power to circuit	normal sensor functionality	
3. Monitor incoming analog data	environmental data visible on serial monitor	
Result	PASS	

Test Case: 1.4		
Pre-Condition: Set up ESP32 device as a BLE GATT server. Code written to encapsulate captured data from the BME680 sensor and set the BLE characteristic value. Start the ESP32 device and launch application.		
Steps:	Expected Results:	Actual Results:
1. Click on the “Personal Metrics” tab in the bottom navigation.	Populated text views to display temperature, humidity, pressure, gas, altitude, and the timestamp of the reading.	As expected
Result	PASS	

Test Case: 1.55		
Pre-Condition: Code written to link sensor, establish PC serial connection, and display incoming analog data. Code is uploaded to board. Altitude sensor is connected to board.		
Steps:	Expected Results:	Actual Results:
1. Connect board to PC via USB and open serial monitor	"ready" pattern on LED shows code is uploaded and board is operational	As expected
2. Activate altitude sensor by providing power to circuit	Normal sensor functionality	
3. Monitor incoming analog data	Environmental data visible on serial monitor	
Result	PASS	

Test Case: 1.6		
Pre-Condition: Code written to link sensor, establish PC serial connection, and display incoming analog data. Code is uploaded to board. Humidity sensor is connected to board.		
Steps:	Expected Results:	Actual Results:
1. Connect board to PC via USB and open serial monitor	"ready" pattern on LED shows code is uploaded and board is operational	As expected
2. Activate humidity sensor by providing power to circuit	Normal sensor functionality	
3. Monitor incoming analog data	Environmental data visible on serial monitor	
Result	PASS	

6.2 Test Plan 2: Databasing

6.2.1 Summary

Story ID: DB-1

The user should be able to store and retrieve data collected by the environmental sensors.

6.2.2 Test Cases

Test Case: 2.1		
Pre-Condition: Completed sensor-board system with appropriate code is operational and in data collection mode. Database functionality has been implemented.		
Steps:	Expected Results:	Actual Results:
1. Verify latest data from BME680 sensor. 2. Pass data to SQLite database and store it. 3. Retrieve data from database and display on UI.	Active SQLite data packet matches incoming sensor data Data integrity remains and reflects proper database entry	As expected
Result	PASS	

6.3 Test Plan 3: Display of Data and/or Results

6.3.1 Summary

Story ID: COM-1, UI-1, UI-12, UI-15, UI-16, UI-17

The user should be able to view all application data in a simple, user-friendly format.

6.3.2 Test Cases

Test Case: 3.1.1		
Pre-Condition: Code to display and render a simple interface to click and access basic features of the application. Needs the phone to be launched with app installed		
Steps:	Expected Results:	Actual Results:
1. Open the app from the homepage of your phone	App opens and lands on the homepage	As expected
2. Press the personal metrics section	App switches tabs and displays the live-data section	As expected
3. Read the displayed data that is live and updating in a small interval	App updates the data in live time (few seconds interval) accurately	
Result	PASS	

Test Case: 3.1.2		
Pre-Condition: Code to display the tabs to access different parts of the app.		
Steps:	Expected Results:	Actual Results:
1. Open the app through the phone's homepage	App opens to the homepage and displays the bottom navigation bar	As expected
2. On the homepage, view the bottom navigation bar and click any of the buttons	The button would open the specific tab	As expected
Result	PASS	

Test Case: 3.1.3		
Pre-Condition: Code to toggle string resources to English/French in all fragments of the application.		
Steps:	Expected Results:	Actual Results:
1. Open DialogFragment containing the option to select 'English' or 'Français'	DialogFragment opens.	As expected
2a. Select 'English' if the app is currently in French.	String resources converted upon toggling.	
2b. Select 'Français' if the app is currently in English.	String resources converted upon toggling.	
Result	PASS	

Test Case: 3.1.4		
Pre-Condition: Ample functionality available to justify existence of the settings menu. Menu and inflater implemented.		
Steps:	Expected Results:	Actual Results:
1. Open the sidebar	Sidebar opens.	As expected
2. Select the Settings menu	Settings menu opens	
Result	PASS	

Test Case: 3.1.4		
Pre-Condition: Dark Mode functionality implemented		
Steps:	Expected Results:	Actual Results:
1. Open the sidebar	Sidebar opens.	As expected
2. Select the Settings menu	Settings menu opens	
3. Select the Dark Mode option	Dark Mode activates	
Result	PASS	

Test Case: 3.1.5		
Pre-Condition: Help menu functionality implemented		
Steps:	Expected Results:	Actual Results:
1. Open the sidebar	Sidebar opens.	As expected
2. Select the Settings menu	Settings menu opens	
3. Select the Help option	Fragment opens to display helpful information	
Result	PASS	

6.4 Test Plan 4: Account Services

6.4.1 Summary

Story ID: UI-2, UI-3, UI-5, UI-9, UI-10, UI-14

The user should be able to create a new account to use within the application, log into the application using an existing account, and join a team.

6.4.2 Test Cases

Test Case: 4.1		
Pre-Condition: Firebase functionality implemented within app backend. UI frontend has fields for capturing user input.		
Steps:	Expected Results:	Actual Results:
1. Set up a new account	App passes control to Firebase services for account creation	As expected
2. Log in to existing account	App identifies valid credentials and logs user in	
3. Click sign out button	User gets logged out	
Result	PASS	

Test Case: 4.2		
Pre-Condition: Firebase functionality implemented within app backend. UI frontend has fields for capturing user input.		
Steps:	Expected Results:	Actual Results:
1. Attempt to log in with bad credentials	Error message(s) appear that inform the user of any discrepancies	As expected
Result	PASS	

Test Case: 4.3		
Pre-Condition: Firebase functionality implemented within app backend. Team functionality implemented within UI frontend including fields and buttons.		
Steps:	Expected Results:	Actual Results:
1. Tap “Create a new team”	App prompts user to create a team	As expected
2. Name and create a team	Team is created and user is added to team	
3. Join an existing team	App recognizes a valid team code and adds user to existing team	
4. Leave a team	App removes user from team database and returns view to a “no team” display.	
5. Attempt to join a non-existent team	Error message(s) appear that inform the user of any discrepancies	
Result	PASS	

6.5 Test Plan 5: Data Handling

6.5.1 Summary

Story ID: UI-5, UI-8, DB-1, DB-2, DB-3

The user should be able to store team and environmental data using integrated database functionality.

Data should be accessible and viewable in a user-friendly format.

6.5.2 Test Cases

Test Case: 5.1		
Pre-Condition: Firebase functionality implemented within app backend. Team functionality implemented and at least one team created with at least one member.		
Steps:	Expected Results:	Actual Results:
1. Open, close, pause, freeze, reset etc. the app but do not kill process 2. Close the app and kill the process	Real time data collection continues Data collection stops.	As expected
Result	PASS	

6.6 Test Plan 6: Device Services

6.6.1 Summary

Story ID: DA-8, DA-9, DA-10, DA-11, DA-13, DA-14

The user should be able to easily interact with the device through the application.

6.6.2 Test Cases

Test Case: 6.1		
Pre-Condition: Device ID family recognizable to app as a WhereSafe product. Device Settings menu implemented.		
Steps:	Expected Results:	Actual Results:
1. Open Settings > Device Settings	Device Settings fragment is displayed	As expected
2. Tap ‘Pair Device’	A list of available devices is displayed	
3. Select target device	Device is paired to app and the selection is confirmed	
4. Tap ‘Rename Device’, enter a name, and tap ‘Confirm’.	Dialog is displayed with a textbox to enter the new name. Once confirmed, the device name is changed in the device settings menu.	
Result	PASS	

Test Case: 6.2		
Pre-Condition: Device ID family recognizable to app as a WhereSafe product. Device Settings menu implemented. A device is already paired		
Steps:	Expected Results:	Actual Results:
1. Open Settings > Device Settings	Device Settings fragment is displayed, including paired device info.	As expected
2. Tap ‘Forget Device’	Device is decoupled from app and device info is removed from display.	
Result	PASS	

Test Case: 6.3		
Pre-Condition: A device is paired to the app. Bluetooth functionality is active in the app but Bluetooth services are turned off on the mobile device.		
Steps:	Expected Results:	Actual Results:
1. Open app without Bluetooth services active	Dialog displays an alert that WhereSafe wants to enable Bluetooth	As expected
2a. Tap 'Cancel'	Bluetooth remains off and app remains at login screen	
2b. Tap 'Confirm'	Bluetooth is turned on and the system returns to starting up as normal	
Result	PASS	

Test Case: 6.4		
Pre-Condition: A device is paired to the app. Bluetooth functionality is active in the app and Bluetooth services are turned on for the mobile device.		
Steps:	Expected Results:	Actual Results:
1. Open app and let Bluetooth services activate	N/A	As expected
2. Minimize app and check mobile device notifications menu	WhereSafe app runs as a background service	
Result	PASS	

Test Case: 6.5		
Pre-Condition: Notification services are active on the mobile device. Notification functionality is present in the code.		
Steps:	Expected Results:	Actual Results:
1. Open app and let data services activate	N/A	As expected
2. Allow or force target environmental parameters outside of recommended levels	User receives push notifications that depend on active sensor and level of danger	
Result	PASS	

6.7 Test Plan 7: App Security and Reliability

6.7.1 Summary

Story ID: DB-4, UI-13

The user should be able to interact with the app without concern of the app crashing or data being corrupted.

Test Case: 7.1		
Pre-Condition: User is logged in and connected with a device.		
Steps:	Expected Results:	Actual Results:
1. Open app and let Bluetooth services activate	N/A	As expected
2. Rapidly switch between tabs and views in the bottom and side navigation panels	The app should not crash, and unloaded data should be handled.	
3. Within each view and fragment, rapidly tap the screen in random or unexpected patterns	The app should not crash, and unexpected behaviour should not occur.	
Result	PASS	

7 Definition of Done Checklist

7.1 Sprint 1

Story ID	COM-1
User Story	
As a user, I want to view metrics tracked by the device on my smartphone so that the data can be visualized.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-1
User Story	
As a user, I want to obtain real-time data of potential environmental gas-phase contaminants so that I can know if there are any dangerous gasses present	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-2
User Story	
As a user, I want to obtain real-time data of ambient temperature conditions so that I can be aware of the temperature conditions and avoid any potential dangers.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-3
User Story	
As a user, I want to obtain real-time data of ambient pressure conditions for personal safety.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-1
User Story	
As a user, I want to display safety metrics captured by the hardware device.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

7.2 Sprint 2

Story ID	DA-6
User Story	
As a user, I want to obtain real-time altitude data so that I can be aware of my general location relative to sea level.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-7
User Story	
As a user, I want to obtain real-time humidity data so that I can be aware of the relative humidity in the surrounding environment.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DB-1
User Story	
As a user, I want to clearly observe past data metrics of each device-wearer, so that I can know if I was at risk of any dangers during past activities.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DB-2
User Story	
As a user, I want to monitor the real-time data of multiple device-wearers at once, so that I can make sure none of my co-workers/peers are at risk of any danger.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DB-3
User Story	
As a user, I want to monitor the real-time data of multiple device-wearers at once, so that I can make sure none of my co-workers/peers are at risk of any danger.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-2
User Story	
As a user, I want to set up an account using my Google account so that I can save my data and preferences to my account.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-3
User Story	
As a user, I want to securely log in to the system using my Google account so that I can view my saved data.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-9
User Story	
As a user, I want to be able to create a team so that other users can join my team and I can view their data.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-10
User Story	
As a user, I want to be able to join a team, so that I can view my team member's data.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

7.3 Sprint 3

Story ID	DA-8
User Story	
As a user, I would like to change my device's name if desired.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-9
User Story	
As a user, I would like to get notifications if the surrounding environment has bad air quality or extreme temperatures.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-10
User Story	
As a user, I want to be able to track the surrounding environment data even while my phone is set to sleep so that I don't have to keep my phone on and save battery.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-11
User Story	
As a user, I want to be able to easily connect to the device if I am disconnected to ensure that I can measure my surrounding environment.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-13
User Story	
As a user, I want to be able to forget my previously connected device so that I can connect to a new one.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DA-14
User Story	
As a user, I want to be able to select the device I want to connect to within a list so that I can decide which device to select.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DB-4
User Story	
As a product owner, the app should be well-tested to ensure data security and integrity while the product is operational.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	DS-0
User Story	
As a user, I want my physical device to be battery powered and portable.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Integration tests performed and passed (with other device components)	Done Done Done
3. Requirements mentioned in PBI are satisfied	Done
4. Acceptance criteria for PBI are satisfied	Done Done
5. Unit tests performed and passed (battery life, power consumption, etc.)	Done Done
6. End-User Documentation Updated	Done
7. Tested	
8. Zero Known Defects	
9. Acceptance Tested	

Story ID	DS-1
User Story	
As a user, I want to have a comfortable wearable device to measure my real-time data, so that I can wear it for long periods of time.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Integration tests performed and passed (with other device components)	Done Done
3. Requirements mentioned in PBI are satisfied	Done
4. Acceptance criteria for PBI are satisfied	Done
5. Unit tests performed and passed (comfort, wearability, measurement accuracy)	Done Done
6. End-User Documentation Updated	Done
7. Tested	
8. Zero Known Defects	
9. Acceptance Tested	

Story ID	UI-5
User Story	
As a user, I want to choose which user’s data will be displayed, so that I can view the desired person's data.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-8
User Story	
As a user, I want to display these safety metrics in a clear and user-friendly interface so that I can easily navigate to the desired metrics.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-12
User Story	
As a user, I want dark mode functionality to alleviate eye strain when using the app.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-13
User Story	
As a user, my app should be robust and well-tested to ensure a seamless and pleasing user experience.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-14
User Story	
As a user, I want to be able to leave a team if I don't want to share my data with other users.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-15
User Story	
As a user, I want to switch the app's language to French for a more comfortable experience in my preferred language.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-16
User Story	
As a user, I would like to have some helpful information about the app so that I know what is possible within the app.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

Story ID	UI-17
User Story	
As a user, I would like to easily change my preferences and settings so that I can alter the look and feel of the app, or manage my device.	
DoD checklist for this PBI	Status
1. Design Reviewed	Done
2. Code Completed	Done
2.1 Code refactored	Done
2.2 Code is commented	Done
2.3 Code reviewed	Done
3. End-User Documentation Updated	Done
4. Tested	Done
5. Zero Known Defects	Done
6. Acceptance Tested	Done
7. Live on production servers	Done

8 Computer Simulations

In the interest of saving time, effort, and resources, the development of the WhereSafe system benefited from extensive use of computer simulation for predicting and optimizing parameters.

Developer-side UI live testing was employed to estimate the user experience. Suites were built using Android's automated test capabilities for stepping through views and fragments and ensuring that app navigation lined up with the planned layout. Very active fragments (i.e., those with many live parameters or that possess advanced data display functionality) could be automated and tested in the same way. This branch of testing was most useful for small code pushes that affected global parameters or the navigation structure. The ability to instantly test the integrity of the app without running the entire program from within Android Studio's infamously slow builder was exceptionally useful; it is akin to command-line testing in procedural languages or HDL in that the outcome of one's code changes can be observed much faster (albeit somewhat indirectly).

Hardware simulation was carried out using OrCAD and PSPICE to facilitate the design of the physical unit. Product bugs are cheaper and easier to catch in earlier stages of production, and the time constraints of the product necessitated a proper development pipeline to ensure timely delivery. PSPICE enabled us to map circuit nets, ensure consistent power delivery, and finetune circuit elements towards optimizing device efficiency. Further CAD testing helped us envision what the device would look like as we strove to miniaturize the design and reduce its physical footprint.

Physical CAD programs like TinkerCAD and PrusaSlicer were key tools in developing the 3D printed case that houses the electronic components of the device unit. Computer-aided design streamlines many tedious aspects of the design process, which allowed our constraints and specifications to be defined before the first edge was ever drawn. An .stl file is also easy to share and analyze, which allowed the design to be collaboratively fine-tuned on the fly. The final design is compact, durable, and can be printed in its entirety (PLA; 0.3 mm layer height) in under four hours. In a similar vein, the VStitcher textile CAD suite provided essential libraries for

measurements and patterns used in creating the wearable pouch, which led to minimized fabric loss during the manufacturing process.

Data spoofing from a virtual BME represented a cornerstone of our app integration process. While the ESP chip is usually fed data from the BME sensor, a “soft” BME was coded to send matching streams to the development board. This has many applications, chief among which was the ability to set up the development board by testing it in isolation. This allowed us an early opportunity to arrange our circuit to our specifications and abstract the design away, pushing development deeper into the software phase ahead of schedule. This setup also made debugging the board and app environment at later stages of development much easier. Data was spoofed using both directed and constrained random methodologies. The former allowed us to quickly simulate large amounts of “up” time, facilitating the integration of data storage and display protocols. The latter allowed us to build realistic ranges of parameter values, which helped us create boundaries for usage with respect to data display. This included corner case testing and the development of event triggers for user notifications (e.g., determining at which concentrations of VOC we should push alerts to the team leader).

9.2 Milestone 2

Date	Who					Type of Activity	Num-ber of hours spent	Purpose	Output	Hours spent					
	SURAJ	CRIS	SHENG	ALEX	JAMIE					Suraj	Cris	Sheng	Alex	Jamie	
31-Jan-23					1	work at home	1.5	Create draft interview script	Draft interview script Stubs for all	0	0	0	0	1.5	
4-Feb-23	1	1	1	1	1	meeting	1.5	Plan Milestone 2	sections. Delegates selected for most sections. Interviews planned. Script finalized. Interview 1	1.5	1.5	1.5	1.5	1.5	
5-Feb-23					1	work at home	2	Collect interview 1	Interview 2	0 0	0 0	0 2	0 0	2 0	
5-Feb-23			1			work at home	2	Collect interview 2	Followed up on interview status.	0.5	0.5	0.5	0.5	0.5	
5-Feb-23	1	1	1	1	1	meeting	0.5	Plan Milestone 2	Finalized what we could without interviews. two sections of milestone doc plus						
10-Feb-23			1		1	work at home	1	Integrate interview data into Milestone doc	small extras Product Backlog. Sprint 1 Backlog.	0	0	1	0	1	
11-Feb-23	1	1	1	1	1	meeting	6	Crunch session for Milestone 2, Sprint meeting 1, and Scrum Assignment 1	Sprint 1 Plan. Presentation for Efran. Scrum	6	6	6	6	6	

									Assignment 1 document.						
12-Feb-23	1	1	1	1	1	meeting x2	5	Crunch session continued	Finished milestone 2 document. Finished Presentation. Finished Scrum assignment.	5	5	5	5	5	
									Total hours	13	13	16	13	17.5	
									Total team hours						72.5

9.3 Sprint 1

Date	Who					Type of Activity	Num-ber of hours spent	Purpose	Output	Hours spent					
	SURAJ	CRIS	SHENG	ALEX	JAMIE					Suraj	Cris	Sheng	Alex	Jamie	
13-Feb-23	1	1	1	1	1	Dev team meeting	1.5	Consolidate product design based on scrum meeting 1 feedback. Assign tasks. Set up ESP32 and BME680 sensor	Individual / pair task lists.	1.5	1.5	1.5	1.5	1.5	
17-Feb-23	1					work at home	1	Create basic data collection/display from BME680 thru Sparkfun Thing	As stated	1	0	0	0	0	
18-Feb-23					1	Basic Sensor functionality	2		As stated	0	0	0	0	2	

18-Feb-23					1	ESP Wifi connectivity	3	Connect ESP chip to personal wifi network	Connection protocol and simple debug for serial monitor. Actual connection still not established	0	0	0	0	3	
18-Feb-23					1	Documentation	1	Create documentation of circuit, code, and implementation notes for Sparkfun Thing / BME680	Collected OM chip diagrams and pinout specs. Collected ESP datasheet. Formatted some graphics for ease of viewing.	0	0	0	0	1	
18-Feb-23				1		work at home	4	Implement functional bottom navigation bar	Implemented functional bottom navigation bar	0	0	0	4	0	
18-Feb-23	1					work at home	2	Set up ESP32 as webserver	ESP32 was setup as a webserver and the sensor data is viewable through a web browser	2	0	0	0	0	
19-Feb-23	1	1				work at home	2	Set up ESP32 as a Bluetooth Low Energy Server	ESP32 was setup as a BLE server and the sensor data is viewable through a bluetooth scanner app	2	2	0	0	0	
19-Feb-23			1			work at home	2	Integrate android application	Home page setup according to the	0	0	2	0	0	

								homepage/main activity	first design sketches					
19-Feb-23	1	1				work at home	3	Connect android device to ESP32 over bluetooth to read data	Connection between devices is established but cannot read data	3	3	0	0	0	
21-Feb-23				1		work at home	4	Implement functional side navigation drawer	Implemented functional side navigation drawer	0	0	0	4	0	
21-Feb-23					1	Documentation	1.5	Develop Testing Plan	Template, alignment with user stories	0	0	0	0	1.5	
23-Feb-23					1	Documentation	1	Fix Testing Plan	Fixing erroneous template formatting	0	0	0	0	1	
23-Feb-23	1					work at home	4	Read data from ESP32 over bluetooth	The android app can read the data sent by the ESP32 device over bluetooth	4	0	0	0	0	
24-Feb-23	1		1			work at home	1	Store incoming data in a database	Implemented an SQLite database to store BME680 sensor data	1	0	1	0	0	
26-Feb-23				1		work at home	2	Implement WhereSafe logo on startup	Implemented WhereSafe logo on startup	0	0	0	2	0	
27-Feb-23					1	Documentation	2	Solidify Testing Plan	Finalize format; template created for stories conversion; mass population of document begins	0	0	0	0	2	

1-Mar-23	1	1	1	1	1	meeting	4.5	Crunch session for Sprint 1 product, documentation	Produced Sprint Backlog 2 and revised product backlog. Added testing, UI wireframe, and environment data to asst document	4.5	4.5	4.5	4.5	4.5	
2-Mar-23	1	1	1	1	1	meeting	4.5	crunch session continued	Finished product iteration and Definition of Done. Completed deliverables for upcoming meeting with scrum master	4.5	4.5	4.5	4.5	4.5	
									Total hours	23.5	15.5	13.5	20.5	21	
									Total team hours						94

9.4 Sprint 2

Date	Who					Type of Activity	Number of hours spent	Purpose	Output	Hours spent					
	SURAJ	CRIS	SHENG	ALEX	JAMIE					Suraj	Cris	Sheng	Alex	Jamie	
9-Mar-23						work at home	3	Research and set up Firebase database in android app Design database	Created a Firebase Project and set up Firebase Firestore database	3	0	3	0	0	
10-Mar-23	1					Documentation	1		Created diagrams to visualize database structure	1	0	0	0	0	
10-Mar-23		1				work at home	2	Implement join/create Team frontend feature with random generated team code	Implemented the buttons and code generator Implemented	0	2	0	0	0	
10-Mar-23	1					work at home	3	Implement Firestore database helper functions in android app to easily store and read BME680 sensor data	Firestore database helper functions	3	0	0	0	0	
11-Mar-23		1				work at home	5	Implement Google Maps API for real-time user location and begin the skeleton for the future Teams view integration	Implemented Google Maps view of user's current location	0	5	0	0	0	

11-Mar-23				1		work at home	4	Implement functional graphs in Personal Metrics to display data	Graph libraries were added, debugging necessary for graph functionality.	0	0	0	4	0	
11-Mar-23	1					work at home	2	Merge database feature, charts, and team metrics features onto main project repository branch	Merged different features onto main branch and resolved conflicts	2	0	0	0	0	
15-Mar-23			1			work at home	4	Research + Implementation of Login page (design and setup)	Created a similar design to initial prototypes and matching it with Firebase's login requirements	0	0	4	0	0	
15-Mar-23				1		work at home	3	Implement functional graphs in Personal Metrics to display data	Implemented functional graphs in Personal Metrics to display data	0	0	0	3	0	
15-Mar-23	1					work at home	3	Implement Firestore database functions to handle team creation and joining	Implemented Firestore helper functions and integrated with team metric fragment in android app	3	0	0	0	0	
16-Mar-23					1	Physical Circuit	3	Simulate and design ideas to make physical unit portable	Several design ideas; shopping/begging list of components; documentation	0	0	0	0	3	
16-Mar-23	1		1			work at home	2	Add sign up and integrate app authentication with firebase	Added sign up activity and integrated app authentication with firebase	2	0	2	0	0	

17-Mar-23		1				work at home	3	Implement features in the Settings tab to change hardware device name or to disconnect/reconnect hardware device	Implemented features in the Settings tab to change hardware device name or to disconnect/reconnect hardware device	0	3	0	0	0	
17-Mar-23		1				work at home	2	Implement feature to force user to enable Bluetooth whenever disabled	Implemented feature to force user to enable Bluetooth whenever disabled	0	2	0	0	0	
17-Mar-23				1		work at home	4	Implement widgets on home page to display data	Implemented widgets on home page to display data	0	0	0	4	0	
17-Mar-23	1					work at home	2	integrate with team features	Implemented Firestore functions to integrate authentication throughout the app	2	0	0	0	0	
17-Mar-23					1	Customer Feedback	1	Reinterview of C4	Answers to queries	0	0	0	0	1	
17-Mar-23					1	Physical Circuit	4	Component retrieval; build and test circuit ideas	First rough working prototype; documentation	0	0	0	0	4	
18-Mar-23			1			Customer Feedback	2	Reinterview of V.M	Feedback on current implementations and further details on what's to come/plans	0	0	2	0	0	
18-Mar-23	1					work at home	0.5	Connect home page data with firebase so that they are updated	Connected home page widgets with firebase	0.5	0	0	0	0	

								when new data is recorded							
17-Mar-23		1				work at home	2	Re-implement feature to force user to enable Bluetooth whenever disabled	Re-implemented feature to force user to enable Bluetooth whenever disabled	0	2	0	0	0	
18-Mar-23	1	1	1	1	1	work at home	4	Crunch session for completion/refinement of Sprint 2 implementations	Finished product iteration and Definition of Done. Completed deliverables for upcoming meeting with scrum master	4	4	4	4	4	
19-Mar-23	1	1	1	1	1	work at home	5	Crunch session for completion/refinement of Sprint 2 implementations		5	5	5	5	5	
19-Mar-23					1	Physical Circuit	3	Finalize and miniaturize circuit	Second, smaller circuit prototype; new circuit diagrams and documentation	0	0	0	0	3	
									Total hours	25.5	23	20	20	20	
									Total team hours						108.5

9.5 Sprint 3

Date	Who					Type of Activity	Number of hours spent	Purpose	Output	Hours spent					
	SURAJ	CRIS	SHENG	ALEX	JAMIE					Suraj	Cris	Sheng	Alex	Jamie	
24-Mar-23	1					work at home	4	Integrating language support within the backend	Added language preferences to the firebase database	4	0	0	0	0	
24-Mar-23					1	work at home	3	Produce final circuit design with physical measurements	Physical unit design template	0	0	0	0	3	
26-Mar-23	1					facilitated workspace	3	Design, print, and aestheticize physical unit protective case	Physical unit protective case	3	0	0	0	0	
26-Mar-23		1				work at home	4	Implement "Device Settings" in "Settings"	Implemented "Pair device", "Forget Device", and "Rename Device" features	0	4	0	0	0	
26-Mar-23			1			work at home	6	Dark Mode implementation	Implementation of Dark Mode to all parts of the app + Research on Colour science for designs and contrasts	0	0	6	0	0	
27-Mar-23					1	work at home	4	Adding notification support	Started implementing notifications, not functioning	0	0	0	0	4	
28-Mar-23	1				1	facilitated workspace	2	collect components and solder onto board	Physical unit final circuit	2	0	0	0	2	

28-Mar-23	1				1	facilitated workspace	3	design and produce wearable pouch	Physical unit wearable pouch	3	0	0	0	3	
28-Mar-23			1			work at home	5	Updated UI for Login/Sign Up Screens	Update of Sign In and Sign Up + Forgot Password to match our updated version of the UI (making it more user friendly)	0	0	5	0	0	
28-Mar-23		1				work at home	4	Implement backend code that will send end user's current location to Firestore database for further integration in Team Metrics	Completed backend code implementation that will send end user's current location to Firestore database for further integration in Team Metrics	0	4	0	0	0	
29-Mar-23	1					work at home	4	Implementing Bluetooth connection as a foreground service	Started implementing the Bluetooth service as a foreground service, but it is still not functional	4	0	0	0	0	
29-Mar-23				1		work at home	2	Create new string resources in English/French for all previously hardcoded strings	String resources created in English/French	0	0	0	2	0	
29-Mar-23		1				work at home	4	Integrate a "device proximity" feature	Researched ways to display device proximity using RSSI as a "Find my Device" feature and began implementing code Final assembled physical device	0	4	0	0	0	
30-Mar-23	1				1	facilitated workspace	2	Integrate physical device		2	0	0	0	2	

30-Mar-23				1		work at home	4	Implement language toggle feature (English/French) in Settings fragment	Started implementing language toggle feature. Feature is not fully functional, application crashes upon toggling.	0	0	0	4	0	
30-Mar-23		1				work at home	4	Integrate a "device proximity" feature	Continued implementing code and troubleshooted/debugged	0	4	0	0	0	
31-Mar-23	1					work at home	3	Add manual connection button for device	Added button to manually connect to device but the connection is not properly implemented	3	0	0	0	0	
31-Mar-23				1		work at home	4	Fully implement language toggle feature (English/French), fix existing bugs.	Implemented language toggle feature. All strings within app are converted upon toggling.	0	0	0	4	0	
31-Mar-23		1				work at home	4	Integrate a "device proximity" feature	Concluded there is no efficient way to display the RSSI proximity of the device after device pairing. Decided to display the device proximity during BT pairing instead	0	4	0	0	0	
01-Apr-23	1					work at home	2	Fix connection and reconnection with device	Implemented a way for the connection button to connect or reconnect to the device	2	0	0	0	0	
01-Apr-23					1	work at home	3	Fixing notifications	Fully implemented notifications for air quality, extreme temperature, and extreme humidity	0	0	0	0	3	

01-Apr-23	1					work at home	4	Fully implement Bluetooth as a foreground service	Finish implementing Bluetooth connection as a foreground service	4	0	0	0	0	
01-Apr-23	1		1			work at home	2	Merge UI	Merged UI related code with the main project	2	0	2	0	0	
01-Apr-23		1				work at home	4	Integrated "device proximity" feature during BT device pairing Updated UI	Implemented device proximity feature for WhereSafe devices in the vicinity during BT pairing	0	4	0	0	0	
01-Apr-23			1			work at home	2	for Data on home screen	Updated version of homepage UI for quick access data to match the schematics and for additional user readability	0	0	2	0	0	
02-Apr-23	1					work at home	3	Setup settings screen	Implemented preference screen	3	0	0	0	0	
02-Apr-23	1					work at home	2	Add toggle for dark mode and save preferences	Added toggle for dark mode and saved information within users' preferences. Ensured correct mode is triggered based on previous saved mode	2	0	0	0	0	
02-Apr-23	1					work at home	1	Merge dark mode code	Merged code for dark mode	1	0	0	0	0	
02-Apr-23				1		work at home	2	Write descriptive information surrounding app functionality & features (to be	Word document created.	0	0	0	2	0	

								inputted into application).						
02-Apr-23		1				work at home	4	Implement backend code that will send end user's current location to Firestore database for further integration in Team Metrics	Started backend code implementation that will send end user's current location to Firestore database for further integration in Team Metrics	0	4	0	0	0	
03-Apr-23	1				1	work at home	2	Add air quality measurement to hardware code	Implemented bsec library to add air quality metrics	2	0	0	0	2	
03-Apr-23	1				1	work at home	1	Adapt android code with new hardware code	Added support for air quality measurements	1	0	0	0	1	
03-Apr-23				1		work at home	4	Implement 'help' section in Settings, consisting of multiple fragments to display information surrounding app functionality & features.	Created fragments 'WhereSafe 101', 'Features & Experiences', 'Privacy & Security'. Debugging necessary, unexpected behavior when navigating between fragments.	0	0	0	4	0	
03-Apr-23		1				work at home	4	Implement Team GPS location of all team members	Begin implementation of backend code to populate Google Maps view with team members' coordinates	0	4	0	0	0	
04-Apr-23				1		work at home	3	Fully implement 'help' section in Settings.	Fixed unexpected behavior when navigating between fragments.	0	0	0	3	0	

								'Help' section fully implemented.							
04-Apr-23	1					work at home	2	Add a list of teammates within the team fragment	Added a list view within the team fragment to show list of user's team members	2	0	0	0	0	
04-Apr-23	1					work at home	1	Update home fragment UI	Added rounded cards to show all home page metrics	1	0	0	0	0	
04-Apr-23	1					work at home	1	Update personal metrics UI	Wrapped charts within rounded cards views	1	0	0	0	0	
04-Apr-23	1					work at home	2	Add teammate fragment to show teammate's information	Made the teammate list clickable to show teammate's information within a fragment	2	0	0	0	0	
04-Apr-23	1	1	1	1	1	work at home	2	Prepare Final Demo	Slides and presentation plan for final demo	2	2	2	2	2	
05-Apr-23	1	1	1	1	1	work at home	1	Prepare and deliver final demo	Final product demonstration	1	1	1	1	1	
05-Apr-23			1	1		work at home	2	Update UI in all fragments.	Modified colors of buttons and app bar, modified fonts.	0	0	2	2	0	
									Total hours	47	35	20	24	23	
									Total team hours						149

9.6 Final Submission

Date	Who					Type of Activity	Number of hours spent	Purpose	Output	Hours spent					
	SURAJ	CRIS	SHENG	ALEX	JAMIE					Suraj	Cris	Sheng	Alex	Jamie	
13-Apr-23	1					work at home	2	Update design document and definition of done	Created new diagrams, and updated descriptions. Added new definitions of done	2	0	0	0	0	
13-Apr-23					1	work at home	2	Create framing elements of final report submission	Revised mission statement and abstract; structure edits and formatting	0	0	0	0	2	
13-Apr-23				1		work at home	3	Create report on ethical dimensions of our product	Ethical dimensions report created, review necessary	0	0	0	3	0	
14-Apr-23				1		work at home	1	Added test cases/formatted document.	Test cases added, document formatted. Revised ethical dimensions report	0	0	0	1	0	
14-Apr-23					1	work at home	1	Review/finalize ethical dimensions report.	Introduction written	0	0	0	0	1	
15-Apr-23		1				work at home	1	Write the Introduction		0	1	0	0	0	
15-Apr-23		1				work at home	2	Review code and add comments	Code reviewed and commented	0	2	0	0	0	
15-Apr-23					1		2	Formalize test cases	Group member statements (ephemeral); consolidated test suite library	0	0	0	0	2	
									Total hours	2	3	0	4	5	
									Total team hours						14

10 References

[1] Adafruit Industries, "Adafruit_BME680," GitHub, 2021. [Online]. Available: https://github.com/adafruit/Adafruit_BME680. [Accessed: 11-Feb-2023].

[2] A. Industries, “Adafruit APDS9960 proximity, light, RGB, and gesture sensor,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/3595. [Accessed: 11-Feb-2023].

[3] A. Industries, “Adafruit BME680 - temperature, humidity, pressure and gas sensor,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/3660. [Accessed: 11-Feb-2023].

[4] A. Industries, “Adafruit BMP388 - precision barometric pressure and Altimeter,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/3966. [Accessed: 11-Feb-2023].

[5] A. Industries, “Adafruit DRV2605L haptic motor controller - Stemma QT / Qwiic,” *adafruit industries blog RSS*. [Online]. Available: https://www.adafruit.com/product/2305. [Accessed: 11-Feb-2023].

[6] Android Developers, "Bluetooth Low Energy overview," Android Developers, 2021. [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth/ble-overview. [Accessed: 19-Feb-2023].

[7] Bosch Sensortec GmbH, “BSEC-Arduino-Library,” *GitHub*, 2020. [Online]. Available: https://github.com/BoschSensortec/BSEC-Arduino-library. [Accessed: 29- Mar-2023].

[8] C. S. Staff: “Interfacing BME680 with Arduino also Measure Indoor Air Quality index,” *Circuit Schools*, 03-Mar-2022. [Online]. Available: https://www.circuitschools.com/interfacing-bme680-with-arduino-also-measure-indoor-air-quality-index/#:~:text=BME680%20can%20detect%20wide%20range,old%20models%20BMP180%20and%20BME280. [Accessed: 20-Mar-2023].

[9] Espruino, “BME680 environment sensor,” *Espruino*. [Online]. Available: https://www.espruino.com/BME680. [Accessed: 11-Feb-2023].

[10] Google, “Using API Keys,” *Google*. [Online]. Available: https://developers.google.com/maps/documentation/android-sdk/get-api-key. [Accessed: 27-Mar-2023].

[11] Google, “Foreground services : Android developers,” *Android Developers*. [Online]. Available: https://developer.android.com/guide/components/foreground-services. [Accessed: 29-Mar-2023].

[12] M. Moqups, “Online mockup, Wireframe & Ui Prototyping Tool · Moqups,” *Online Mockup, Wireframe & UI Prototyping Tool · Moqups*, 2023. [Online]. Available: https://moqups.com/. [Accessed: 11-Feb-2023].

[13] Philipp Jahoda, “MPAndroidChart,” *GitHub*, 2020. [Online]. Available: https://github.com/PhilJay/MPAndroidChart. [Accessed: 29-Mar-2023].

[14] R. Forest, “Online logo maker,” *Renderforest*, 2023. [Online]. Available: https://www.renderforest.com/logo-maker. [Accessed: 11-Feb-2023].

[15] R. Santos, "BME680 Sensor with Arduino – Gas, Temperature, Humidity, and Pressure," Random Nerd Tutorials, 2021. [Online]. Available: https://randomnerdtutorials.com/bme680-sensor-arduino-gas-temperature-humidity-pressure/. [Accessed: 19-Feb-2023].

[16] S. Avigezer, “How to communicate with Bluetooth Low Energy Devices on Android,” *Medium*, 26-Feb-2019. [Online]. Available: https://medium.com/@shahar_avigezer/bluetooth-low-energy-on-android-22bc7310387a. [Accessed: 19-Feb-2023].

[17] Stack Overflow, "How to connect Android App to ESP32 via Bluetooth," Stack Overflow, 2019. [Online]. Available: https://stackoverflow.com/questions/57105709/how-to-connect-android-app-to-esp32-via-bluetooth. [Accessed: 16-April-2023].

[18] SparkFun Electronics, "ESP32 Thing Plus Hookup Guide," SparkFun, 2021. [Online]. Available: https://learn.sparkfun.com/tutorials/esp32-thing-plus-hookup-guide/all#arduino-example-esp32-ble. Accessed: 19-Feb-2023]

[19] "UUID Generator," UUIDGenerator.net, 2021. [Online]. Available: https://www.uuidgenerator.net/. [Accessed: 19-Feb-2023].