**Technical Documentation for *SafeHer***

**CONTENTS**


- Quick Start Guide
  - How to use/access SafeHer
- Prototype Features
  - AI Triggers
    - Scream Detection
    - Crash Detection
  - Security Protocols
    - SOS Alert
    - User Verification
    - Invitation Only Access
  - Geolocation
    - Dashboard Map
    - Sharing Live Location
    - Nearby Women Commuters
  - Peer to Peer
    - Filing a Report/Incident
    - Number of Reported Incidents
    - Reported Incidents
    - Resolved Issues
- Pilot Features
  - SafeWord
  - Hotline Database
  - Commuting Diary
  - Admin Tool
  - Biometric Login
- In-App Implementation
  - Front-End
  - Database & Data Security
    - Database
    - Data Security
- Potential Risks in Development
- Developer Tools
- Implementation Guide for Other Countries


**Quick Start**

This manual can be used by researchers and developers who wish to use SafeHer as a reference for further developments in women empowerment and equality.

**How to use/access SafeHer**

*Back to contents*

1. Download the app by sending a request to the admin for access via safeher.ph.
2. Open the SafeHer App
3. Sign up to create an account
4. Verify your identity
5. Add your emergency contacts
6. Click “Emergency SOS” to trigger the app’s security protocols
7. Click “File a Report” to file incidents in your area
8. Click “Share Location” to share where you are to your emergency contacts
9. Security protocols can automatically trigger when the app detects your scream

## PROTOTYPE FEATURES

### AI Trigger

### AI Scream Detection

This feature allows SafeHer to detect audio and categorize it as a scream to trigger security protocols. We implemented this feature by utilizing the following:

1. Creating the machine learning training algorithm. Utilizing Python as the language for creating the model.
2. Utilized Support Vector Machine (SVM) to classify if an audio contains a scream or not.
3. To feed the SVM with data, we utilized the Librosa library and Mel-frequency cepstral coefficients (MFCC) to extract 200 scream audio and 200 non-scream audio.
4. The python script will then predict if the audio is a scream or not using the aforementioned training model
5. To increase accuracy in detecting scream, we also used the Scikit-Learn library in order to achieve 80%-85% accuracy rate.

### AI Crash Detection

This feature allows SafeHer to detect crash accidents through the use of the user’s phone accelerometer. We designed this feature through the following:

1. We needed to identify the property of a “crash” and this can vary per vehicle type. With the limitation in open data on “crash” properties, we identified that it can only be limited to one calculation.
2. SafeHer “crash” calculation is defined as

   *crash = ave kmph drops by 40 kmph within a 5 second period*

3. As of development, we identified that testing this is dangerous and risky. Therefore, it was recommended that we develop a word that will trigger the crash detection such as saying “Crash!”.

*Back to contents*

## SECURITY PROTOCOLS

### SOS Alert

This feature triggers a series of alerts either from AI triggers or done manually by the user.

- **Sound Alarm:** we used a text to speech code for the alarm to dictate the following sentences on loop
    - Hey! Stop harassing me!
    - Stop it. Right now!
    - I already contacted the authorities!
- **Flashing Lights:** we coded into the app the ability for SafeHer to turn on the user's phone flashlight that flashes on a 1 second interval.
- **Capture Photos (Front & Rear):** we enabled capturing photos of the incident when the SOS alert is triggered. This can be turned on as long as the user enables app permission to use the camera.
- **Emergency Texts:** utilizing a third-party SMS gateway app (Semaphore) SafeHer is able to automatically send a text message to the user's emergency contacts about the possible danger and location of the user.

### User Verification

We added a layer of ID and Selfie verification into the signup process. The app requires the user to submit a valid ID and take a selfie. Assessment of the documents are done by the data processor of the app.

### Invitation Only access

Access to the app is strictly done via invitations only. This further safeguards our users. This means that the app cannot be accessed publicly on the app store and can only be used once given access. Submit a request via https://safeher.ph.

## GEOLOCATION FEATURES

This main function integrates live location into SafeHer. We utilized google maps and geocoding into the app.

### Dashboard Map

This feature enables the user to see her location in real time. For identifying the location/address of the user's current location, we used a technique called reverse geocoding which relies on the Geocoding API from Google.

### Sharing Live Location to contacts

The user can then share her location to her contacts. A link is generated and sent to her emergency contacts via SMS. The SMS contains a Google Map link which enables the emergency contacts to view where the user is.

*Back to contents*

**Nearby Women Commuters**
We enabled the user to be able to see other SafeHer commuters in their dashboard map within a 2km radius. The user is only able to see numbers and pins within her dashboard map.

## PEER TO PEER FEATURES

This main function contains a series of peer to peer related features wherein the SafeHer database calls on a third party database for these to work.

**Filing a report/incident**
This feature enables the user to report incidents inside SafeHer. It's essentially an easy to fill out form with the following fields:

- Location
- Date & time
- Category of Incident
    - Touching
    - Ogling
    - Facial Expressions
    - Sexual Assault
    - Cat Call
    - Indecent Exposure
- Further Description
- Submit Photo Attachment

Once a user submits a report, the data is saved into the SafeHer Database and can be sent to partner cities via email.

**Number of reported incidents within vicinity**
We coded into the app a location incident report counter wherein the user is able to see a series of incident numbers in a vicinity within their Dashboard Map. These reports are sent to city partners either via email or sms. So far, we've coded the incidents to be reported to the authorities via email.

**Reported Incidents**
This is an in-app notification function when a report is submitted to the SafeHer Database and is sent to the partner database.

**Resolved Issues**
This is another in-app notification function when a report has been resolved by the partner authority.

*Back to contents*

## PILOT FEATURES

### SafeWord
This feature enables the user to use set words such as “Help!” to trigger emergency protocols. SafeWord uses voice recognition and recording to recognize the user's voice and read the words set by the user.

### Hotline Database
This database enables the user to access national level, city and barangay level emergency hotlines. This is only limited to a record of numbers set by the developers.

### Commuting Diary
This feature is used for gathering response data of pilot testers. Feedback gathered from this diary is used by the developers to improve the user experience.

### Admin Tool
This is a security feature where app administrators counter check sign up submissions to diminish fraudulent and questionable app access. This administrative feature can only be accessed by app admins.

### Biometric Login
This additional security feature increases user login protection using their unique physical features to verify themselves. The developers have implemented fingerprint and facial recognition technology to enable this security function.

## IN-APP IMPLEMENTATION

### Front-End
We used a JavaScript-based mobile framework called react-native paired with a UI framework called Tamagui.

- Android and iOS share a common codebase for the most part with some exceptions like for alerts. Android uses Toast while iOS uses an alert box.
- For the scream detection and running the python script, heavily relies on libraries for the script to run. For android, we used Chaquopy. As for ios, we are still looking for a suitable library for this.

## DATABASE AND DATA SECURITY

### Database
We used Amazon dynamodb for our database, Amazon Cognito for user sign up and authentication, and Amazon S3 for cloud storage.

### Data Security

*Back to contents*

Using Amazon's services all data that comes inside the SafeHer database utilizes AES-256 Encryption. This means that only designated personnel can view unencrypted user data if given permission by the user (such as troubleshooting an account issue).

## POTENTIAL RISKS IN DEVELOPMENT

We've listed down possible risks that other developers might encounter in creating a similar app. Our development team has already addressed these potential risks in our internal protocols.

1. Susceptible to inaccuracies especially in scream detection of natural voice versus pre-recorded voice
2. Crash Detection feature can be bypassed in some other ways such as throwing the device to another person. This means that as long as all set rules are true for the accelerometer sensors, even if it's not a crash, the SOS alert message may trigger
3. Unverified users can submit fraudulent IDs to fake submissions to get access to community-based features (such as Find nearby commuters),
4. Fake users can utilize geolocation data to perpetuate malicious deeds.
5. Users may upload malicious files unknowingly that may contain malware that can possibly corrupt some database files.
6. Stolen phones may result in stolen personal data of the user inside the app
7. Since the prototype will be an open code application, others can easily reverse engineer the code to create an identical app but for the purpose of malicious intent
8. Access to the database (via AWS) is only limited to the back-end developer and data controller; if either of these two assigned personnel had their network hacked, a breach in the database may occur.

## DEVELOPER TOOLS

These are the tools that were used in deploying the prototype version of SafeHer.

- Phyton Programming Language
- SVM Algorithm
- Librosa Library
- MFCC
- Scikit-learn Library
- JavaScript
- React-Native
- Tamagui
- Toast for Android
- Alert Box for iOS
- Chaquopy for Android
- Semaphore

*Back to contents*

- Reverse Geocoding
- Geocoding API by google
- Amazon Dynamodb
- Amazon Cognito
- Amazon S3 for Cloud Storage
- GoDaddy
- Figma
- Confluence
- Jira
- Google Play Store
- Apple App Store
- AWS Amplify
- Github

**IMPLEMENTATION GUIDE FOR OTHER COUNTRIES**

This quick guide can help developers develop similar apps to SafeHer.

1. Create an account on Amazon and setup your own Dynamodb, Cognito, Cloud Storage, and Amplify
2. Copy the app's code base from github.
3. In order to localize labels/texts inside the app, change the text fields accordingly. Such as changing "Login" into "Accesso" in spanish.
4. Setup your own SMS solution provider for the emergency texts. We used Semaphore. The provider should be local in your country (such as BulkSMS.com in Spain).
5. Make sure to deploy in a test environment first.

Note: We advise the developers to develop first on ANDROID before trying to implement it for iOS.

*Back to contents*