

# **THE 'RESCUE' APP**

## **GROUP MEMBERS:**

RAJARSHI MANDAL - **[B2130039]**

DIPANJAN MAITI - **[B2130026]**

AADESH DAS - **[B2130021]**

# OVERVIEW

This app is primarily focused on women to ensure their safety on roads. If someone is being chased by stalkers, she can pull out her mobile phone and start this Resue app. She only needs to tap one button and the following actions will be taken place:

1. The camera will open letting her to capture a video of the situation
2. Emergency calls will be placed which she already has saved earlier
3. SMS along with the location of her will be sent to the same contacts

A list will be created where there will be log of all accidents and anyone can tap anyone of them and will be redirected to a page where he/she will be able to see a map tracking the victim's location and the video made by her.

# APP WORKING

1. **Start the app**, user sees two buttons
2. Clicking the '**Emergency**' button opens the camera and user can start video recording
3. User now can **upload the video** by pressing the 'OK' button, the video is uploaded to firebase storage
4. After the video has been uploaded, the **location fetching** is started and continues to fetch current location of the user in fixed time intervals which are then stored in firebase real-time database
5. **SMS** are now sent to all the contacts which the user has already saved earlier regarding emergency situation along with the URL tapping on which google maps is opened indicating the last fetched location of the victim
6. **Calls** are also placed on the saved contacts one after another
7. On tapping the second button 'Accidents' a list of all the accidents registered so far are shown
8. Clicking on a particular entry directs the person to another page where he/she can **watch the video of the particular accident**, the video is not the original one but is processed with python OpenCV to extract faces of people captured in the video
9. A **map** is also there **marking all the locations of the victim**, the map is updated live

## Work done by Rajarshi Mandal:

1. Video recording and uploading it to firebase storage
  - Dexter: for permissions handling  
<https://www.geeksforgeeks.org/easy-runtime-permissions-in-android-with-dexter/>
  - VIDEO\_RECORD\_CODE: for video recording
2. Location fetching and storing in firebase real time database in fixed time intervals
  - FusedLocationProviderClient: for location fetching  
<https://developer.android.com/training/location/retrieve-current#:~:text=The%20fused%20location%20provider%20is,device's%20use%20of%20battery%20power.>
  - FirebaseDatabase, DatabaseReference, StorageReference: Firebase connection
  - Handler: for repeated execution of same block in intervals  
<https://stackoverflow.com/questions/13954611/android-when-should-i-use-a-handler-and-when-should-i-use-a-thread>
3. GUI where a person can see list of all accidents fetched from database and on click
  - Simple\_spinner\_list\_view: since only accidents are needed to be shown
4. In a new page one can see the video and the live locations fetched from firebase database uploaded by the victim
  - VideoView: <https://stackoverflow.com/questions/3263736/playing-a-video-in-videoview-in-android>
  - Fragment : google maps is showed here
  - SupportMapFragment: for displaying and updating map
  - Google map Api:  
<https://stackoverflow.com/questions/34582370/how-can-i-show-current-location-on-a-google-map-on-android-marshmallow>
  - MediaController: for playing video fetched from firebase  
<https://developer.android.com/reference/android/widget/MediaController>
5. When a victim uploads an accident in firebase, every person having the app installed gets notification
  - NotificationManager: class for managing notifications  
[https://www.youtube.com/watch?v=C\\_qdD6tFo28](https://www.youtube.com/watch?v=C_qdD6tFo28)

# Work done by Aadesh Das:

## ❖ Basic Overview of the work-

1. Ran a real-time edit-data recognizer which checks for the changes in firebase database continually throughout its lifetime.
2. When a new data is added (during emergency), it is fetched and the video that is embedded within the data is downloaded.
3. A face detection algorithm is running which detects the face(s) (if any) that are present in that video. I took help from [this portion of a YouTube video](#) to understand the face detection and face recognition procedure.
4. The face(s) are saved as different .jpg file(s) and detected faces in the video is marked (using rectangular boxes).
5. Lastly, upload the processed video back to the database. The video will be shown to the users for easy viewing of the face of the fraud.

## ❖ Pre-requisites-

1. A text editor or an IDE (I have used PyCharm IDE) [\[official-link\]](#)
2. Any latest version of python (I have used Python 3.8.10) [\[official-link\]](#)
3. The following python packages need to be installed-

Package name	Command to install it	Official link
a. Pyrebase	[pip install Pyrebase4]	<a href="#">[official-link]</a>
b. cmake	[pip install cmake]	<a href="#">[official-link]</a>
c. dlib*	[pip install dlib]	<a href="#">[official-link]</a>
d. face-recognition	[pip install face-recognition]	<a href="#">[official-link]</a>
e. numpy	[pip install numpy]	<a href="#">[official-link]</a>
f. opencv-python	[pip install opencv-python]	<a href="#">[official-link]</a>

\* - might face some error while installing it (troubleshooting given later)

## ❖ Api used-

1. Connection to the database is done by the following code which works with a configuration key which contains all the necessary components required to connect to the database:

```
34. firebase = pyrebase.initialize_app(config)
35. database = firebase.database() # connecting to db
36. storage = firebase.storage() # connecting to storage
```

2. The configuration key (here, config) contains the following data:

```
20. config = {
21.     "apiKey": "AIzaSyAYtUkE9wcGSrU9dSI1FU0AU2-czuRgFxs",
22.     "authDomain": "pythondbtest-de47f.firebaseio.com",
23.     "databaseURL": "https://pythondbtest-de47f-default-rtdb.firebaseio.com",
24.     "projectId": "pythondbtest-de47f",
```

```
25.  "storageBucket": "pythondbtest-de47f.appspot.com",
26.  "messagingSenderId": "91516807715",
27.  "appId": "1:91516807715:web:925f598616b3614f071f96",
28.  "measurementId": "G-M3LSNQHDNC",
29.  "serviceAccount": "serviceAccountKey.json"
30. }
```

### ❖ Problems encountered in the process-

In the whole process, I have mainly faced one problem:

- Repeated failing of installation of the **dlib** package.
  - The problem can be easily avoided by installing 'dlib' from the following link instead of installing it using the command.
  - The link is provided [here](#) (just make sure to install an older version instead of the latest version which usually causes the problem).
  - I have used version 19.22.1 and it worked just fine.
- Another way to avoid the problem is to install all the packages using PyCharm's own package installer (detailed process can be found [here](#))

# Work done by Dipanjan Maiti:

## Saving emergency contacts with SQLite:

1. SQLite is implemented extending SQLiteOpenHelper.
2. Using SQLiteOpenHelper class, a local database is created and a table.
3. Data is added to the table using insert method of the SQLiteOpenHelper class.
4. Data display- For this we need sql command- "select \* from contacts\_table" for selecting all data from the table and a CURSOR class instance that helps to display the data on each and every row.
5. For deleting data, we have used delete method in SQLiteOpenHelper.

## SMS:

1. All phone numbers saved in emergency contact will be send a SMS asking for help and the current location of the victim will be given.
2. SMS is send using the Api SmsManager:  
[https://www.tutorialspoint.com/android/android\\_sending\\_sms.htm](https://www.tutorialspoint.com/android/android_sending_sms.htm)

## PHONE\_CALL:

- 1) Calling will be made to all emergency contacts sequentially.
- 2) We need to take permission (only for first time) to make a phone call.
- 3) We need to add  

```
<uses-permission android:name = "android.permission.CALL_PHONE" />
```

  
in Android Manifest.
- 4) Calling is done in a separate thread because if done in main thread it leads to ANR as a call takes more than 5 secs.
- 5) Each call is implemented using Intent.ACTION\_CALL:

<https://www.javatpoint.com/how-to-make-a-phone-call-in-android>

## ADDITIONAL WORKS THAT CAN BE DONE:

The call can be made simultaneously to all emergency contacts rather can sequentially. It can be implemented inside the app itself like what we see in WhatsApp group call.