

[TMBR Project Final Write Up]

Concept and Motivation:

The main concept that was created by the team was to implement an app that promotes user safety in a contemporary setting. To specify, in today's society, there are still many risks that arise from a variety of sources including encounters with people from dating apps, walks through dark areas, and even a trip away from home. All these things have always been topics of concern for the populous. However, we now have the technology to reduce that risk. We created an app to accompany a user while they are in an uncomfortable environment or situation. The app is designed to be non-intrusive and run in the background of the phone. The basic task of the app is to provide a means of communication without having to take out your phone and physically dial or type anything. If you are out past what the user set as their curfew for the night, designated contacts specified by the user then become notified through a simple text message generated by the app stating something simple like "Hello, this is ____'s security buddy. I have stayed past my designated time out. Check up on me!" This would then encourage the contact to message or call the user to ensure their safety.

Main Activity:

The Main Activity is composed of a tab fragment at the top and a fragment container in the lower body which contains the contents of each tab as you cycle through them. The fragment container is a frame which is filled with fragments as the view when selected and adds the previous fragment to a stack upon swapping.

Main Fragment:

The Main Fragment class contains a TimePicker that is used to set the duration timer. It also features a Start Button at the bottom of the screen that is used to accept the time selection, and send an initial SMS message to the TMBR Buddy. This is accomplished by using the contents of the TimePicker to set a time in the future for our Service Client which initiates the background service which keeps track of the time and sets an alarm for the specified future time. Upon hitting the start button, two services run in parallel with one another. The first is the durational service which only checks in on the user upon the completion of the TimePicker's contents. The second is an interval service identical to the first except that it runs up until a time specified in the preferences of the app and occurs in intervals of that specified time. Essentially, at any given moment, there are two threads running sequentially throughout the duration of the services

(unless the user chooses to untoggle check-ins which disables the interval feature). Upon the arrival of the set time or interval, a signal to push a notification is broadcast to the device.

Tab Fragment:

The tabs at the top are represented as a Fragment which exists throughout the app. An ActionBar would accomplish the task as well, but we chose the Fragment method because it was easier to design and work with. There are four tabs (HOME, SERVICES, BUDDY, PREFERENCES) that take the user to different corresponding activities.

Services Tab:

The services tab provides toggle options that allow for Enabling Location and Enabling Check-ins. This information is kept persistent in the easiest way which is to make this a preferencefragment which then stores all the information without having to reset upon relaunching the app.

Buddy Tab:

The Buddy tab is responsible for selecting a contact to set as your designated TMBR Buddy. The “SELECT A CONTACT” button at the bottom of the screen takes the user to the contact activity which allows the user to select a contact onclick. Upon selecting a contact as a “buddy,” this contact is added to a database managed by the app in internal storage. This database holds the contact name and number in different columns. The information is then relayed back and displayed in the view of the buddy tab fragment. The information is obtained by means of a cursor that goes through the database. At the time of submission, we only have support for one buddy as this was the central idea; however, it was thought that in the future, more buddies could be added to this database and used to make a much more involved and community defining experience for the user. It is important to note that the number associated with the buddy is what is used to push SMS messages to that buddy which is the main functionality of the app. It is not possible to start the app without having selected a buddy beforehand.

Preferences Tab:

The preferences tab allows the user to change some basic settings of the app such as the Check-In Time intervals, and the ringtone notification. This, like the services tab, is managed by a preferencefragment which keeps the information persistent throughout the app.

Notification:

The way that notifications work in the app is simple. Notifications are generated after the Alarm Manager of the app has decided that the correct amount of time has elapsed for there to be a new notification broadcast. Since there are two services running sequentially, one is tied to the durational manager while the other is tied to the interval timer. Upon broadcasting for the

notification, one of these customized notifications is generated which either prompts the user if they are feeling ok or asks them how their time was if the duration has ended. It was planned to have the durational service also track the user and see if they have arrived home by the end of the duration but this service was never coded. Upon clicking the notification, the user is taken to a separate activity which prompts the user to see if they are feeling well or if they are uneasy. This was done by adding an intent to the notification with the direction of the activity which provided the options described. Depending on the response, the app generates and sends an SMS message for the assigned buddy to receive.

SMS Messaging:

Through a simple SMS based system, the app would be able to generate messages to send to the TMBR Buddy alerting them of staying out past a predetermined time, not arriving home according to the system's GPS, or even a prompt from the user that they feel uncomfortable and would like company. This was done by creating a sendSMS method which created an SMSManager object which was used to push the message to the number stored in the database. Before sending the message, if the user had selected that they are uneasy and they have also enabled location services, then the generated message included the gps location of the user which is then pushed as latitude and longitude as a String to be included in the message content. Regardless of the user response, a toast is generated which displays when the next prompt will be arriving (specified by the user but also acts as a reminder).

GPS Location:

The GPS location was used upon sending a message to the buddy. The latitude and longitude were converted to text and pushed as a part of a larger message. This was implemented by implementing the LocationListener interface. In order to generate the appropriate latitude and longitude, the status of the phone's GPS services is verified and if on, the service gets the last known location from the network provider. Through this, the location is then converted into latitude and longitude which is then stored as fields in the GPSTracker class that was created and the location is returned to where it was required.

Conclusion and Future:

This app was developed with the intention of creating a service which connects two people to provide a sense of security. It was implemented by using notifications, sms, background services, gps location, and databases as the main sources of the content. That being said, there were more features that were thought of but not implemented. These include the incorporation of additional buddies, the optimization of the GPS tracker, integration with Google Maps, customizable message options, and additional notification options.