**Course Project:** Family Tracker Application (Android)

by

**Group – 16**

**BSCS14035** – Fizza Tauqeer

**BSCS14059** – Mamoona Riaz

**BSCS14072** – Siddiqua Nayyer

**BSCS14004** – Awab Aqib

* **Requirements:-**

1. Reliable location tracking of registered family members.
2. Authentication before tracking.
3. Two-way verification/authentication approach (Family members must verify other ‘watcher’ members for tracking).
4. Exact and a pinpointed location of the member to be tracked.
5. Entire location track of the member to be tracked, given time constraints (for example, the entire location track of the member since the past hour or two).
6. Emergency buttons for alerting family members in matters of danger (for example, if you are kidnapped).
7. SMS Alerts - that are sent when the emergency button is clicked/pressed - to family members that are accompanied by embedded links of your current location (or last known location).
8. Concept of alarms upon entering restricted locations, violating set perimeters or upon receiving alerts.
9. Multiple family members tracking.
10. Multiple watchers – those who can track you.
11. Thresholds for distances – location tracking of even a minute difference in location and location tracking of generalized or expansive differences in location.
12. Push notifications of the application – for prompting user to turn on GPS, WiFi, Mobile Data, any alerts, etc.

* **Do you think you need mathematical verification of correctness of your system or a part of your system? Why?**

Significant parts of our system, that heavily influence the entire system and are a pivotal part of binding it entirely together, will require mathematical verification of correctness when it fails in certain scenarios, like transiting from one activity to another activity (through the help of finite state machines), unexpected errors - that severely defect the system - and their prediction (through the help of probabilistic computations), and similar issues that are highly probable to occur in a developing system.

Specifically for our system, this method of verification can be used for comparing and validating the defined period of time for location tracking in the application and the actual time taken for the process, and can also be used for verifying the correctness (or absence of it thereof) of detection of location – in terms of perimeters, latitude, longitude and distance – being done by the application. For example, if a family member’s demand is regarding a certain distance threshold then our application must have to calculate it right in order to continuously cater to this requirement. Such requirements can be ensured by methods that come under mathematical verification of correctness.

* **Can you separate various concerns of your project from functional and quality perspectives? Highlight the concerns and describe how can you handle concerns separately?**

Yes, we can. A functional perspective would highlight whether, say, the application is providing location at all, irrespective of correct or incorrect. A quality perspective would, on the other hand, highlight whether the application is providing the correct location or not. From a functional perspective, our concern for the project would be if the transfer from one module to another dependent module is occurring or not. From a quality perspective, our concern for the project would be if the transfer from one module to another dependent module is giving us the intended and desired result or not. Following such a methodology, all concerns can be separated into either of the perspective and handled accordingly.

Regarding our application, our concerns – including the above mentioned concern of location – would encompass whether all our requirements are indeed being fulfilled, like alerts being delivered, location track being constantly updated and stored, track being showed, tracking being done by ‘watchers’, accurate distance being maintained, etc. These concerns can be handled separately by ensuring that they are each functionally and qualitatively doing their envisioned job as dictated in the former paragraph.

* **Identify some functional modules in your system. Discuss coupling and cohesion aspects.**

Some functional modules could be:-

1. User – each family member has to be a user to use the application’s service. For that, a singular module must exist to cater specifically to them and directing them to Login module. Highly cohesive, low coupling.
2. Login Activity – each family member must be catered in a module that verifies their signed-in/login status to other modules, otherwise un-authenticated people could also get a hold of the application’s information regarding location. Highly cohesive, low coupling.
3. Maps Activity – each family member’s track or exact location will have to be hosted on some mapping grid. This module would host and visualize these tracks and locations on such an embedded mapping grid, while correlating with the Location module. Highly cohesive, yet moderate coupling.
4. Location Activity – actual location tracking of each family member would have to be carried out in a separate module, namely this one. This would work in cohesion with the Maps module. Highly cohesive, yet moderate coupling.

* **Identify the potential future changes in your system. Pick one potential change and discuss how would you address it in your system?**

Some potential changes or features that we may adhere to in near future would be:-

1. Perimeter Scaling – keep track of family member under some specified distance meter (for example, alerting family members if another member goes outside a set perimeter of 10 kilometers).
2. Enabling multiple sub-families (for example, a service where a grandfather could track his children and their extended families).
3. Family Chat inside application between users/members.
4. Save all location tracks on external cloud services (like Firebase) for future prediction and analysis (for example, where a family member usually located on an average basis).
5. Extending the ‘Alerts’ service to Emails as well.

Picking 4) as a potential change, hosting the location tracks on an external cloud service would lessen the computational burden of the application (and any complimentary applications it may require, like Google Maps) and the device itself. It will also be helpful in surveying and analyzing the application along with its activities. In our system, it would be addressed in the source code, since our application will launch on Android and our platform will be Android Studio (both of which support Java, in which there are built-in modules and libraries for enabling and accessing Firebase services). Firebase will be used in the user verification module of the system; hence the track hosting will be an extension on that source code.

* **Which increments would you suggest if you are asked to build your system incrementally?**

If we were embarking incrementally in building our system, increments we would suggest would start with creating empty classes with their preset functions and methods (like that used in classes that extend ‘Broadcast Receiver’**[[1]](#footnote-1)** and classes that implement ‘Location Listener’**[[2]](#footnote-2)**) to visualize an overall flow of the application along with its design. Each stage would house increments that would test screen flow from one to another, with functionality of each screen added in later incremental stages after this flow has been achieved. In incremental versions of the system, those functions and methods of the classes would be over-ridden to do jobs that they inherently could not do.

1. https://www.tutorialspoint.com/android/android\_broadcast\_receivers.htm [↑](#footnote-ref-1)
2. https://developer.android.com/reference/android/location/LocationManager.html [↑](#footnote-ref-2)