

Team Name: AAC-Tech

The mobile application is almost complete. Over the period of late February up to this current day, I continued to work on fixing the remaining bugs in the code and adding the remainder of the functionality.

As of this point, I have completed the following objectives:

- Determining the reason why the paramedic side of the paramedic side of the mobile application was glitching. The paramedic side of the application was glitching due to improper practices in writing code to listen for updates on firebase database. When instantiating a listener for a database, it is common practice to add code to terminate the listener after it has served its purpose. However, I didn't terminate these listeners. After a considerable period of research and attempts to fix the unstable behavior of the code, I finally realized that the paramedic-side was glitching due to the unterminated listener from the paramedic login code, which kept listening for updates in the database and re-executing code that only needed to be run once.
- The map modules have also been set up to enable GPS tracking, and both paramedic and patient is able to view and determine the whereabouts of both themselves and the other. This is done by constantly updating the database each time the location (latitude and longitude coordinates) of the paramedic or the patient changes, and fetching these coordinates to update their current location on the map.

The remaining objectives for the mobile application are as follows:

- Fetching sensor readings. This can only be simulated at this time since code wasn't added yet to send sensor data to the database
- The plan for dealing with the patient's personal information is to save it to a file on the firmware, and updating this file through the app using the Update Info activity. The rest of the information about the patient will be fetched through this file. This hasn't been set up yet.
- I plan to make some slight changes to the UI design for the 'home' screen of the paramedic account, and the 'recent' screen. On the 'recent' screen, I need to add a button to provide the paramedic with the capability to update the patient's status information when the patient has been dispatched at the hospital.

Link to Code:

https://github.com/N01263842/AACTech/tree/master/AACTech/aac_tech/src/main/java/aac_tech/automotiveui

Login Activity:

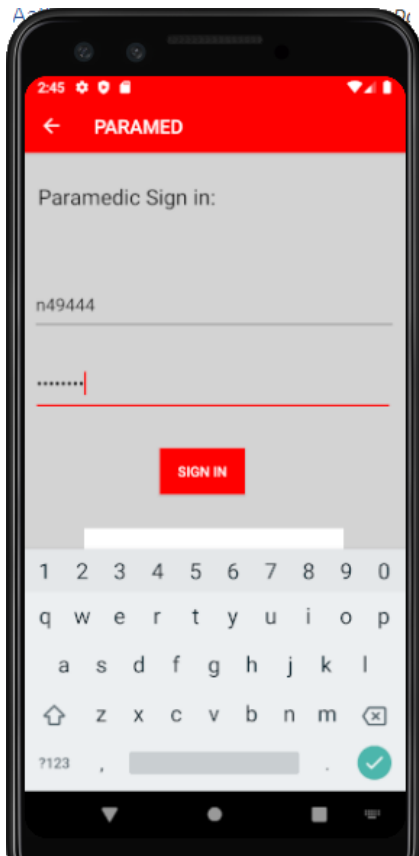


Figure 1- Login Activity: showing how UI appears to the paramedic as the login screen

When paramedic types in their credentials, this is then authenticated by checking the database for a paramedic with matching username and matching password. If it matches for both username and password, paramedic is then logged in to their account. If not, user will not be logged in and will go no further in the activity.

When a paramedic is logged in, to advertise themselves as available to receive calls, their status information in the database is updated from 'inactive' state to 'active' state. A patient, when requesting to chat with a paramedic, will check to see if paramedics are available using the status field.

Data Visualization Activity:

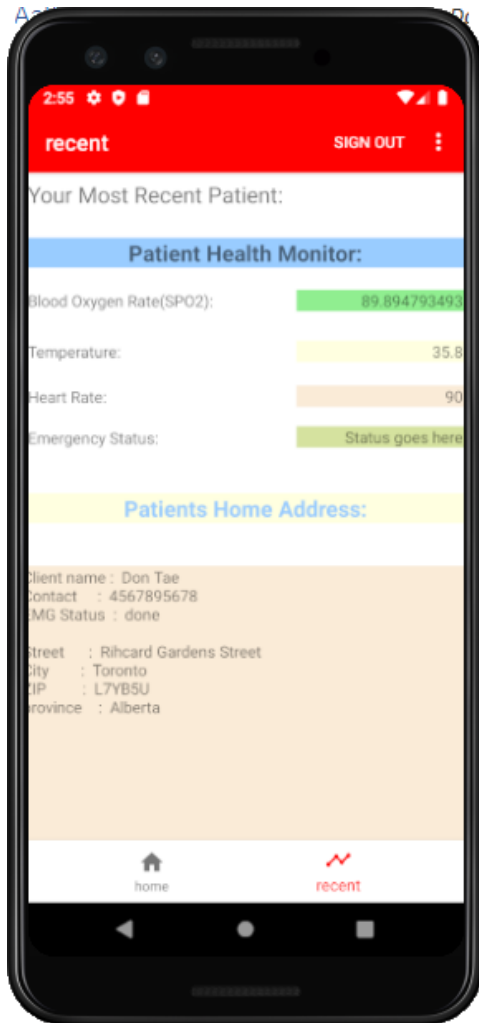


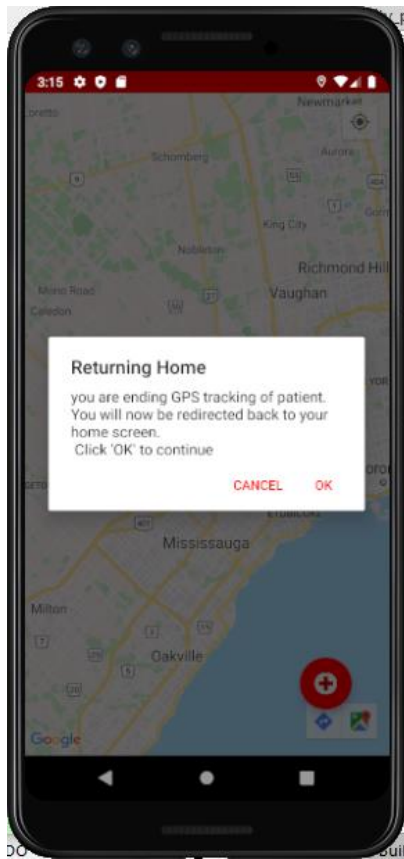
Figure 2-Recent Screen: shows paramedic, when logged in, is able to view the latest emergency scenario that has been done

The recent tab fetches the latest client information from the most recent emergency scenario that was completed or still in effect by the paramedic. This is done by assigning a time stamp as a field in the client's information.

When a client requests for a call with a paramedic, the time that they did this will automatically be saved to the database in epoch format. When fetching the most recent emergency scenario for display to the paramedic, this time stamp for clients assigned to the paramedic, will be compared, and the client information with the largest epoch time, i.e., the latest time, is then displayed to the paramedic

As shown by Figure 2, paramedic is able to view the blood oxygen rate, temperature, heart rate, emergency status, and home address of the most recent patient. This information was separated and presented using various colours to improve its readability, in this way, making it easier for the paramedic to fully grasp what is being presented.

Action Control Activity:



Map module from paramedic side. As shown in the Figure 3, when the red button at the bottom-right side of the screen is clicked, this starts up a dialog box that informs the paramedic that they will be ending GPS tracking of the patient and returning to their home screen, thus, ending the interaction between patient at this point. The objective of this button is so that paramedic can confirm that they have arrived at the pick-up location of the patient, and will now proceed to treat and dispatch the patient to the nearest hospital.

When paramedic clicks 'OK', a series of actions will be performed:

- Updates the status of the client information to 'dispatched'. This will be checked on the client side of the map activity, which will run a dialog box to inform the patient that the paramedic has arrived. After clicking 'OK' on the dialog box, the application will terminate.
- On the paramedic side, application will redirect back to paramedic's home screen, where they will be able to view client's information from there.\

Demo

/1 Hardware present?

/1 Memo by student A

/1 Login activity

/1 Data visualization activity

/1 Action control activity

Report

/1 Login activity

/1 Data visualization activity

/1 Action control activity

/1 Modified Code Files in Appendix

/1 Link to Complete Code in Repository