COMP 8531 Project Report

Eunhak Lee, A01026056

Liang Li

# Non-functional Requirements Explanation

1. Input Validation / Integrity Constraints to prevent entry of erroneous data.
   1. Add maxLength property to TextInput component on a few screens, so that there's a maximum length of the entered content
   2. Add length checking of certain property in submit function, and it gives alert if the content's length is longer than a given value.
   3. Numbers and special characters are not allowed to be in client’s name.
2. Reliable transport i.e. retrying if transport error
   1. Add a function ‘performWithRetry’ in mockData.js to implement failure try mechanism
   2. Apply this function on firebase functions (addDoc, updateDoc, getDocs etc.)
   3. When there's connection issue of other errors from firebase service, it will retry a configurable times, and the delay time is also configurable.

# Security Requirements Explanation

* SSL connection is established by the React Native Firestore framework by default. They use HTTPS connection to establish connection to the Firebase Database.

# Project Requirement Overview

Dev Ops / Platform Requirements:

* **Done** Appropriate software engineering methodology and technologies should be followed.
* **Done** The app shall support popular smartphone and tablet platforms e.g. Android and iOS.  (multiplatform development kits such as React Native should be considered, however native app for a specific platform would be acceptable)

Functional Specifications:

* **Done** The app shall support management of Tasks/Daily roster of health care professionals such as community care nurses, physiotherapists, house keeping, food preparation etc
  + **Done** The tasks are retrieved from the backend exposed as a webAPI (WebAPI is preferred though off the shelf solutions e.g. firebase would be acceptable)
  + **Done** However, nurse could also create tasks locally and add to the list locally on-demand and thereafter synchronized to the backend.
  + **Done** List view of all tasks and more detailed view of each task and user profile.
  + **Done** Ability to sort tasks based on the client name, location (city/street), task type and time.
  + **Done** Ability to filter the list based on the client name and/or address.
  + **Done** Nurse shall be able to change the location and time of a task and also update the task status.  Updates shall be synchronized to the backend.
* **Done** The app shall support reminders
  + Automatic reminder for each task that is missed.  Ability to turn off the reminder.
  + **Done** Ability to set personal reminders (personal reminders are different from the task reminders and could be stored differently.  It is preferred that local calendar providers of iOS and Android as well as schedulers are leveraged)
  + **Personal Reminders may display text/images/animated-Giff  on a pop up window or play audio/video/animation**
  + **Ability to send reminders to apple/google watch**
  + **Done** Reminders could repetitive or episodic.

Non Functional Requirements:

* User Interaction Optimization
  + **Done** Filtering, sorting and navigation to task details and customer profile shall not take more than couple of swipes and/or taps.
  + **Done** Appropriate UI elements shall be chosen for data entry e.g. when entering vitals up and down arrow buttons could be utilized and long/short press could be employed to distinguish the finer or coarse changes from some anticipated/default values. (Use of OCRs to extract values from the pictures of the display panel could be considered).
  + **Done** Use of speech processing shall be utilized when entering text sentences.  It should be noted that you can receive the voice directly into your app from the microphone and use solutions such as Whisper to decipher voice as opposed to using the onboard speech processor.
* Performance
  + Nurse could take the photos of the client if it is missing which will then be synced with the backend.  The photo should be resized and resolution reduced for saving space and cutting down on the upload/download time and bandwidth.
  + **Done** Compression of JSON strings before transport should be explored.
  + **Done** Any interaction with the backend shall happen in the background threads or through callbacks i.e. the UI shall not freeze during the interaction with the backend (reliability/availability will be covered further in the next course)
* Scalability
  + **Done** It is preferable that local storage where the tasks, user profiles and task details will be cached after fetching these from the backend and where updates to the task/reminder status will be made utilizes an ORM.
  + **Done** Some tasks may require use of instructional videos some of which could be available as local resources of the application (and thus could be played using smartphone’s Media API) whereas others may involve streaming using Youtube or Exo Player.
  + **Done** The display shall scale to smartphone as well as tablets.

The following non functional requirements will be added on in the next course.

* Reliability
  + **Done** [3 marks] Interaction between the GUI thread and background thread is thread safe.
    - Describe how your design account for situations such as the app crashing and restarting (and thus getting repopulated with data) and periods of loss of connectivity when your app cant reach the backend to get or post data.  Hint:  perhaps consider local storage with a background thread(s) or timers or callbacks conducting periodic autosave and synchronizing data with the backend.
  + **Done** [2 marks]  No memory leaks.
    - **Done** Verify that your mobile app does not leak memory by either periodically and programmatically measuring memory or using tools such as memory analyzer.  Upload screenshots verifying no memory leaks.
  + **Done** [2 marks] ACID of local storage.
    - **Done** Point out if situations lost update can happen due to GUI thread interacting with any background thread in your app or any violations of atomicity could happen if the app crashes while local storage is being updated or connectivity breaks while data being posted to the backend.
  + **Done** [4 marks] Input Validation / Integrity Constraints to prevent entry of erroneous data.
    - Implement at least 2 input validation rules in your app and/or integrity constraints in your persistent local storage.  Present the code snippets as well as verification via screen shots or project demo recording.
  + **Done** [2 marks]  Reliable transport i.e. retrying if transport error.
    - Using code snippets describe implementation of your reliable transport strategy (e.g. resumable upload).  Using wireshark captures or other means, demonstrate that it is working i.e. retransmission after the TCP connection is reset.
* Availability and Fault Tolerant:
  + **Done** [3 marks]  The app shall support location sensing (possible sensor fusion extension) and list coworkers in the vicinity.
    - Present code snippets that describe how GPS, Assisted GPS, sensing of WiFi Access Points, Sensing of Bluetooth sources, sensing of NFC and perhaps other location sensing apparatus is achieved in your app.  If you are developing the mobile app using Flutter or React Native, highlight any differences/parametrization for iOS vs Android, and specifications of underlying permissions.

* + **Done** [6 marks]  The app shall allow messaging (web sockets) and phone callto nearby co-workers
    - Present code snippets showing sending and receiving of web socket messages in your mobile app along with code of your web socket hub.
    - Upload screenshot of wireshark capture confirming use of web socket protocol for messaging.
    - Present code snippets on how a phone call could be made programmatically via  your app.
  + The app shall allow transfer of tasks to (nearby) coworkers
    - This will simply be repurposing of the above implementation.
  + [3 marks]  The app shall facilitate effectual switching between WiFi/MDN (Possible vertical handoff extension) or data transfer shall happen only when network with necessary signal strength is available (to reduce transmission errors and consequently retransmissions).
    - Present code snippet detailing automatically detecting WiFi Access Points in the vicinity, checking signal strength and connecting to a select WiFi access point.
* Security
  + **Done** [3 marks] The app shall allow only authenticated users but solutions such as single sign on and /or multi factor (e.g. Finger Printing if available) authentication shall be utilized.  It should be assumed that smartphone/tablet may be used by different staff members at different times.
    - Comment or whether you think the app is candidate for support for SSO.
    - Describe how your app could contribute towards MFA.  Feel free to use sequence diagram to show the control/message flow to achieve MFA.
  + **Done** [2 marks] Backend shall be SSL enabled.
    - List all steps involved in configuring your backend for SSL (including steps to generate self signed certificate) and any configuration that you needed to do on the mobile to support SSL.

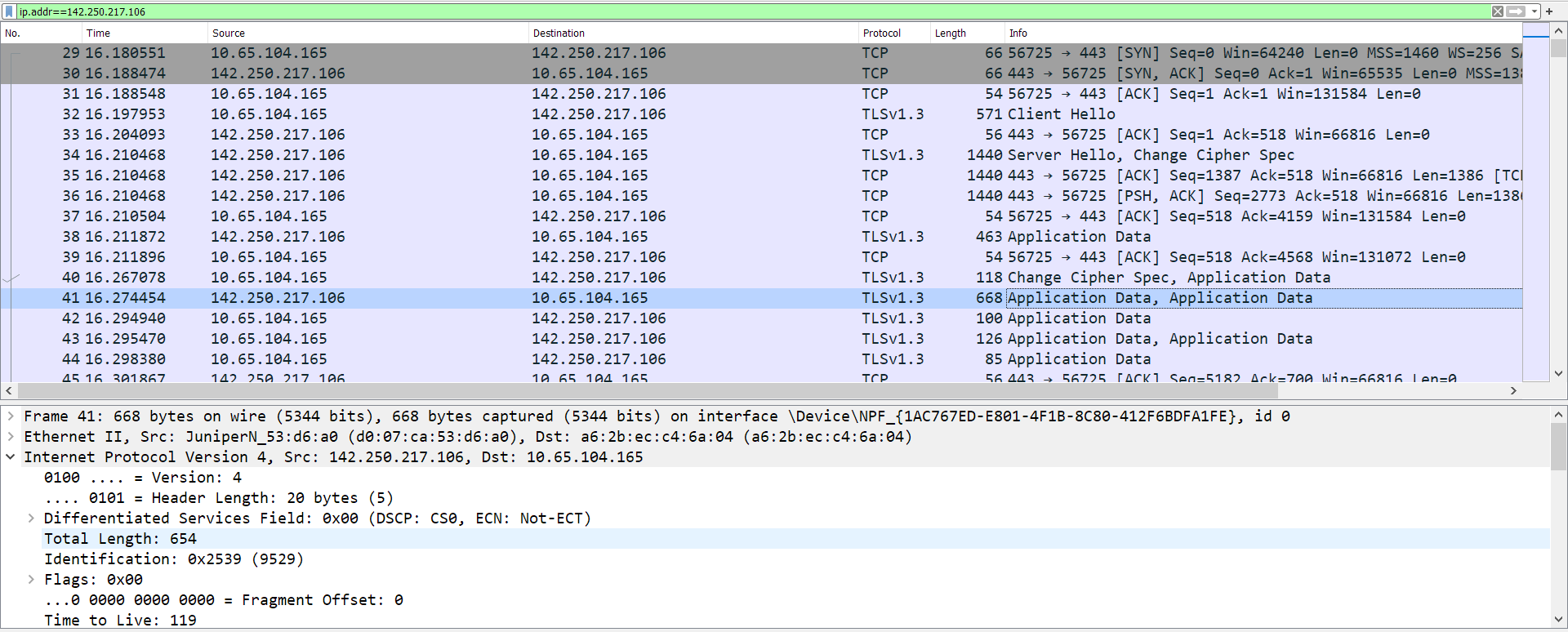
Firestore framework provides HTTPS connection and SSL certificate encryption by default. SSL certificate sent from the Firebase gets verified by the client to ensure they are the legitimate domain. Firebase on the other hand, verifies the API key from the client machine to ensure this client has access to the DB.

* + - Upload screenshot of wireshark capture depicting SSL handshake.
  + **Done** [4 marks]  Use JWT tokens or other recommended practices for access to the backend web API.
    - List all steps needed to configure backend so that it issues JWT token for subsequent authorizations following user authentication.

A screen shot of a computer

Description automatically generated

* + - **Done** Upload screenshots of wireshark capture depicting issuing of JWT token and its inclusion in the subsequent requests to your web api (you may need to disable SSL to see the JWT token in the HTTP Headers).



* + - List possible vulnerabilities of JWT tokens and countermeasures.
  + [3 marks]  A cryptographic framework (possible HIPAA compliant) shall be incorporated to allow sharing of private information when needed.  In the least, hashing of password, and digital signing of the status update of the task by the nurse shall be achieved.
    - Provide code snippets showing use of cryptographic libraries to hash the password.  Upload screen shot of the wireshark capture indicating that while username is in plain text, the password is hashed.   Also upload a screen shot of hashed password stored in the database.
    - Describe how the app will digitally sign the status update of a task by the nurse.  Feel free to use sequence diagram to describe the control/message flow.
  + [1 mark]  Secure App signing and making sure that app is not vulnerable to collusion attacks.
    - List steps to sign your app for release on Android and/or iOS platforms (and/or app stores).
  + **Done** [2 marks] Appropriate permissions shall be requested statically or dynamically.
    - Describe how permissions are specified in the platform that you have chosen for the web app development?  For flutter and React Native, describe for both i.e. for Android as well as iOS variants of the app.

# End Of Report.