Florida International University

School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Pediatric Emergency Medicine App

Team Members: Yeilys Fundora, Abel Gonzalez, Dianet Cruz, Santiago De Irala Mut
Product Owner(s): Jean Hannan, Bruce Quinn

Instructor: Masoud Sadjadi

Abstract

This document presents the information necessary to gain a good understanding of the background, requirements, progress, technologies, and effort put into the development of the second version of the Pediatric Emergency Medicine App.

This application will allow doctors, nurses, and other medical staff to have a quick and easy access to vital information for common pediatric emergencies. It will also support account creation, google login, a chatroom for the users, user account settings for handling user profiles, active status, creating calendar events, adding and reviewing certifications. More importantly, the application implements an admin panel for users with administrative privileges to create, modify and/or delete medical information in the app.

Table of Contents

Introduction

The Pediatric Emergency Medicine App, or PEM App, is a smartphone application designed to assist doctors, nurses, and other medical staff in the midst of pediatric emergencies.

This application will offer important content like a vast amount of information on some of the most common pediatric emergencies, other features include a splash screen once the user opens the application, a redesigned login/signup landing page, Google sign in implementation, new and refined user interface and user experience throughout the application for both iOS and Android devices. The app will also support profile implementation for users when they create an account or login using a Gmail account. In the profile, users will be able to access and edit their personal information, change their active status, review and add new medical certifications, and be able to create new events and save to their device's calendar. There will also be a chatroom for healthcare providers to communicate with each other, in which they can identify each other by clicking on a specific user's avatar/profile photo and see their profile. In addition, the application will have an Administration panel, where admin-users could Edit, Create, and Delete Categories content with Medical Information such as Title, Evaluation, Management, Medication, Symptoms and References, it will also allow the upload and removal of images. The Categories' changes will be displayed throughout the application. Finally, there will be a search area easily accessible from the Home Page (Categories Screen) using the search icon, for the user to look up a specific medical condition.

The PEM App was made possible with React Native, Facebook's mobile application framework. This allowed for the application to be compatible with both Android and iOS devices without having to develop for both separately. React Native is also very robust when it comes to libraries, and some were used to enable the features of this application to work at their best. Firebase was also used to manage the data that will be stored for profile info, medical certifications, medical information under the categories content, and to handle the user accounts and all their operations (account creation, signing in and out, etc.)

Current System

Currently, users can navigate around the app through the categories and find conditions they'd like to know more about, they can search for conditions, they can create and sign into accounts, they can use the chatroom to send messages and share information with their colleagues, and they can keep track of their continuing medical education certifications.

The application works for both Android and iOS devices while satisfying the requirements the product owner asked to be in this application.

Those requirements include:

- Secure account creation
- A chatroom that automatically deletes its messages once all users have signed out.
- A method to see which users are currently logged in
- A place where users can store some information about their continuing medical education so that they are able to keep track of it
- A search area where users can search across the app for whatever they may need
- Categories for types of conditions and subcategories for specific conditions

Purpose of New System

The Pediatric Emergency Medicine App is in its second phase. The purpose of the new system is to improve the application by working on its performance, implementing new features such as Google sign in, new UI/UX experience, admin mode for adding, updating or deleting medical information content, profile integration, improvements in the chat, and testing for usability and accessibility.

- A new implementation of Google Sign-in as a Login Option
- A new splash screen once the user opens the application
- A friendly sign up screen
- A new and refined user interface and user experience throughout the application for iOS and Android devices
- A profile implementation for users who log in using Gmail account or create an account
- A calendar screen where users are able to create events and save them to their device's calendar.
- Users using the chatroom can now see who they are interacting with by tapping on a specific user's profile photo and reviewing their profile info.
- Implementation of Administrator panel to Edit, Create and Delete Categories with Medical information
- structured the previous application's code using react-redux and redux-thunk to successfully manage the application states, logic and allowing an efficient handle of data with server requests and responses.

User Stories

The following details the user stories that have been successfully implemented, as well as some stories that should be implemented in the future.

Implemented User Stories

User Story Number	User Story Task
1	As a user, I would like to access my profile, be able to edit my personal information and save my changes.
2	As a user, I would like to have navigation tabs and menus to access the app's contents easily.
3	As a developer I want to work in restructuring and modifying the application's code using react-redux to manage the application states, logic and be able to handle data efficiently.
4	As an admin-user, I would like to have an interface panel to be able to edit, add and remove data within the application.
5	As a user, I would like to sign into my account using other types of applications such as Gmail.
6	As an Admin-user I want to be able to copy and paste information from different sources to the administrator panel EditCatContent component which handles the adding and editing of data within the application.
7	As a developer I want to use a real-time database as storage to upload, update, and fetch all the medical information related to categories of the application using redux-thunk middleware asynchronous calls.
8	As a developer, I want to restructure the application, moving the categories Search, Chat and CME to their own navigators' components for easier access.
9	As a developer, I want to be able to save all the user information and future updates to a real-time database.
10	As a developer, I want to make sure users have a friendly sign up where it's easy and manageable to fill out.
11	As a developer, I would like to create a way for users to log in and sign up with google when opening the application.
12	As a user, I would like to see a splash screen when first entering the application before going to the login/signup page.

13	As a user of the app, I would like to be able to chat with other users and see previous messages sent in the chat.
14	As a developer I want to implement the remove image functionality for Android and IOS
15	As a developer, I want to show a responsive user profile for when the user's avatar is pressed in the chatroom.
16	As a developer, I want users to create events and be able to add them to their own device calendar.

Pending User Stories

These are user stories that we would like implemented in the future.

User Story Number	User Story Task
17	As a user, I would like to have the advantage of adding content to my Favorites.
18	As a user, I would like to upload a certification.
19	As a user, I would like to be able to remove a certification after it is not valid anymore.
20	As a user, I want to be able to make my user profile info public or private to other users in the application.
21	As a user, I want to have private conversations with specific users in the app.

Project Plan

This section describes the planning that went into the completion of this project. This project incorporated the Agile development methodology and as such required the Sprints to be planned. These sprint planning meetings are detailed below. This section also describes the components, both software and hardware, chosen for this project.

Hardware and Software Resources

The following is a list of all hardware and software resources that were used in this project:

Computer 1
OS Name: macOS Catalina
Version: 10.15.6
Processor: 3.5 GHz Dual-Core Intel Core i7
Memory: 16 GB 2133 MHz LPDDR3

Computer 2
OS Name: Microsoft Windows 10 Pro
Version: 10.0.18363 Build 18363
Processor: Ryzen 5 1600 Six-core Processor
Memory: 16GB 3000MHz DDR4

Computer 3
OS Name: macOS Catalina
Version: 10.15.5
Processor: 1.4 GHz Dual-Core Intel Core i5
Memory: 4 GB 1600 MHz DDR3

Computer 4
OS Name: Microsoft Windows 10 Pro
Version: 1903
Processor: Intel(R) Core(™) i3-2120 CPU @ 3.30GHz
Memory: 16GB 3000MHz DDR3

Source Code Editor
Visual Studio Code

Phones and Emulators
Phones

- IOS
    - iPhone XR, iPhone X
- Android
    - Samsung Galaxy Note 8

Emulators

- Genymotion
    - Samsung Galaxy S8
- Android Studio:
    - Pixel 3
- Xcode Simulator:
    - iPhone 11, iPhone 11 Pro, iPhone SE

Database/Storage
Firebase Realtime Database

User Authentication
Firebase Authentication

Version Control
GitHub

Sprints Plan

Sprint 1

05/18/20 - 05/29/20
Summer 2020 Sprint Planning 05/18/2020 7:00pm-8:00pm

User Story #1: As a developer, I would like to have a working environment where I can start working on the application with the minimum amount of hurdles.
Assigned to: Santiago de Irala Mut

User Story #2: As a developer, I am responsible for learning how the technologies work, and be able to integrate Firebase with React Native, this way I can start contributing to the project right away.
Assigned to: Yeilys Fundora

User Story #3: As a developer, I want to gather a good understanding of the technologies being used in the project by watching several tutorials and following up with the documented information provided by the class instructor. In addition, install and connect all the required environments to successfully work on the application such as Visual Studio Code, Firebase and React-Nactive.
Assigned to: Abel Gonzalez

User Story #4: As a developer, I need to be familiar with the technology needed to work on this project, have everything set up, the app running to understand its functionality and all the required programs.
Assigned to: Dianet Cruz

Sprint 2

06/01/20 - 06/12/20
Summer 2020 Sprint Planning 06/01/20 8:00pm-8:38pm

User Story #1: As a user of the app, I would like to have a way to chat with multiple people using the app at the same time.
Assigned to: Santiago de Irala

User Story #3: As a user, I would like to access my profile, be able to edit my personal information and save my changes.
Assigned to: Yeilys Fundora

User Story #5: As a developer I want to work in restructuring and modifying the application's code using react-redux to manage the application states, logic and be able to handle data efficiently.
Assigned to: Abel Gonzalez

User Story #6: As an admin-user, I would like to have an interface panel to be able to edit, add and remove data within the application.
Assigned to: Abel Gonzalez

User Story #7: As a user I would like to have a drawer navigation to access the admin panel, categories and categories content from the home page of the application
Assigned to: Abel Gonzalez

User Story #10: As a user, I would like to sign into my account using other types of applications such as gmail.
Assigned to: Dianet Cruz.

Sprint 3

06/15/20 - 06/26/20
Summer 2020 Sprint Planning 06/15/20 1:30pm-1:57pm

User Story #1: As a Admin-user I want to be able to copy and paste information from different sources to the administrator panel EditCatContent component which handles the adding and editing of data within the application
Assigned to: Abel Gonzalez

User Story #3: As a developer I want to use a real-time database as storage to upload, update, and fetch all the medical information related to categories of the application using redux-thunk middleware asynchronous calls.
Assigned to: Abel Gonzalez

User Story #4: As a developer I want to redesign the categories content screen to be more user friendly and display data when a tab is pressed
Assigned to: Abel Gonzalez

User Story #5: As a developer, I want to restructure the application, moving the categories Search, Chat and CME to their own navigators components for easier access.
Assigned to: Yeilys Fundora

User Story #7: As a developer, I want to be able to save all the user information and future updates to a real-time database.
Assigned to: Yeilys Fundora

User Story #9: As a developer, I want to make sure User's have a friendly sign up where it's easy and manageable to fill out.
Assigned to: Dianet Cruz

User Story #15: As a user of the app, I want to be able to see a picture of the person/people I am chatting with when using the chat feature of the application.
Assigned to: Santiago de Irala Mut

Sprint 4

06/29/20 - 07/10/20
Summer 2020 Sprint Planning 06/29/20 12:00pm-12:30pm

User Story #3: As a developer, I want to create a calendar screen for users to mark important dates and add events.

Assigned to: Yeilys Fundora

User Story #5: As a developer, I would like to create a way for users to log in and sign up with google when opening the application.
Assigned to: Dianet Cruz

User Story #8: As a user, I would like to be able to see the last message sent in a chat and by who.
Assigned to: Santiago de Irala Mut

User Story #10: As a developer I want to implement the remove image functionality for Android and IOS
Assigned to: Abel Gonzalez

Sprint 5

07/13/20 - 07/24/20
Summer 2020 Sprint Planning 07/13/20 6:37pm-7:19pm

User Story #2: As a developer, I want to show a responsive user profile for when a user avatar is pressed in the chatroom.
Assigned to: Yeilys Fundora

User Story #3: As a developer, I want users to create events and be able to add them to their own device calendar.
Assigned to: Yeilys Fundora

User Story #4: As a developer I want to work in fixing minor bugs in the admin panel system and adding user input validation.
Assigned to: Abel Gonzalez.

User Story #5: As a developer, I want to work on some small fixes and QoL improvements before the sprint ends.
Assigned to: Santiago de Irala

User Story #7: As a developer, I want to begin working on all the needed documentation for my deliverables.
Assigned to: Dianet Cruz

Sprint 6

07/27/20 - 07/31/20
Summer 2020 Sprint Planning 07/27/20 7:05pm-7:30pm

User Story #1: As a developer I want to finish all the necessary documentation and submit the final deliverable before the deadline.
Assigned to: Yeilys Fundora

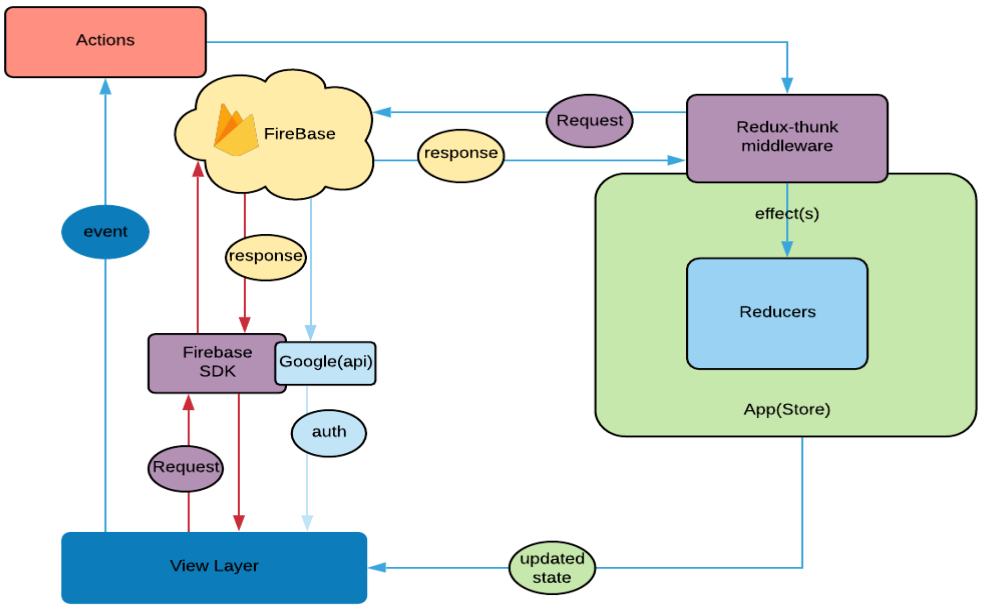
User Story #2: As a developer, I want to finish all the documentation for the project in order for the next team to have a good starting foundation.
Assigned to: Santiago de Irala Mut

User Story #3: As a developer, I want to complete the needed documentation for the entire project and turn in final deliverables before the deadline closes.
Assigned to: Dianet Cruz

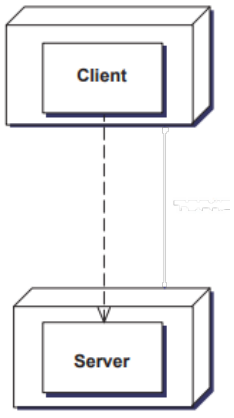
User Story #4: As a developer. I want to complete all the necessary documentation and submit the final deliverable before the deadline closes.
Assigned to: Abel Gonzalez

System Design

This section contains information on the design decisions that went into this project. It uses a simple and easy to use application design to help the user understand the application, as well as a simple architecture design to make it better for developers to understand the overall flow of the app. The architecture patterns are outlined and explained below. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.



Architectural Patterns



The pattern that was used when designing this application was the Client-Server pattern. On the left is a very simple diagram of what this would look like. In it, the Client (in this case, the application running on the user's device) would request services from the server (the application content, chatroom messages, user profile info, etc.). This was used since it was known that a server would have to be constantly listening for when the client either asks for resources it will need. However, the architecture used in this project slightly expands on the Client-Server model. This is due to the fact that the server not only provides resources, but it also handles events like user account creation, secure log in implement with OAuth 2.0 best practices through Google Sign in feature, logout, profile generation and management, messages being sent and received in the chatroom, and user CME certification additions. These events will cause the server to make slight changes to the database in response, as well as serve the content. In addition, the whole application uses redux to manage the application states, logic and maintain the entire application in a single immutable state tree (object), which can't be changed directly. Only through the use of reducers and actions. Furthermore, it integrates redux-thunk to create http requests by using the api middleware to make asynchronous calls to the server and redux-store

System and Subsystem Decomposition

Here, we will go into detail about the system and subsystem deposition. We will explain how the application is structured. The application is divided in 5 main folders: views, components, constants, context and function. The views are what the user interacts with also known as the User Interface, the components are the models that permit the reuse of such components within the application, the constants are the configuration files that stay the same through their use within the app, the context is where all the handling between the views and the functions happen and lastly the functions are the Google Cloud functions that handle the user creation and auth process.

This application is then broken down into the following folders: assets, backend, components, constant, data, models, navigation, screens, store and utilities. The assets folder contains the fonts and the splash screen, the backend contains only one file where most of the details regarding Firebase are handled. The user never interacts with what's in this folder directly. Rather, certain screens the user can interact with will send information to this file for it to handle. Components are the actual category and subcategory tiles the user can press. The Constant folder is where all the colors are stored. If you want to change a color throughout the entire application, just change it there and it will change automatically without having to go one by one. Data is where the specific information mentioned for the pressable tiles lies. The model's folder is where you can change the Category Content throughout the entire application. The navigation is the navigation of the application. The screens are all the different throughout the application. The store folder is the structure of the admin portal and how the information is handled. Lastly, the utilities folder contains a file with the configuration for getting camera permission for users to upload a photo.

Frontend
The frontend of this application was built using React Native and React Navigation. There are numerous important components that allow the frontend of this application to be fully functional. The main two are the screen files, and the React Navigation navigator. Let's explain both in detail.

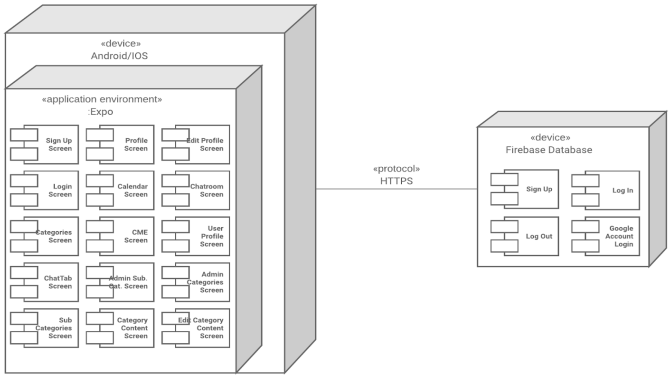
Screens
The screens in this application are mostly represented by React Native classes, but some are implemented using react hooks. These screens are what the user will be seeing whenever they navigate across the application. These screens are being exported so that the navigator can import them. We will talk about this next.

Navigator
The navigator is what's being used to allow the navigation between screens. One of the navigators used is called a "stack navigator". It is given this name because all the screens the user visits are pushed onto a stack. Another navigator used is the "drawer navigator", which implements a side menu, allowing the user to navigate to other screens in the app more easily. Next, there is the bottom tab navigator which includes a tab bar on the bottom of the screen that lets you transition between different routes. Finally, the main navigator is the "switch navigator", which helps switch from one navigator to the other. This is very useful when a user chooses to logout of the app and needs to be redirected to the Login screen from any other screen and no back button should be visible, the purpose is to only ever show one screen at a time.

Backend
The backend of this application runs entirely on Firebase. Firebase is what's managing all the user accounts, tracking the logged users, the chatroom, CME, profile, and it's where all the static subcategory content is held. Most of the screens are involved with Firebase in some way, whether it's to check if the user has logged in to their

account or if it's to pull the correct subcategory content from Firebase Realtime Database. In this case, the way it works is that screens will send a request for whatever they need to Firebase. From there, the data is returned asynchronously to the screen. Because it's returned asynchronously, some screens use their state to manage whether the data has been returned or not. This is how the subcategory content screen is able to say "Loading…" until the data is returned.

Deployment Diagram



Design Patterns

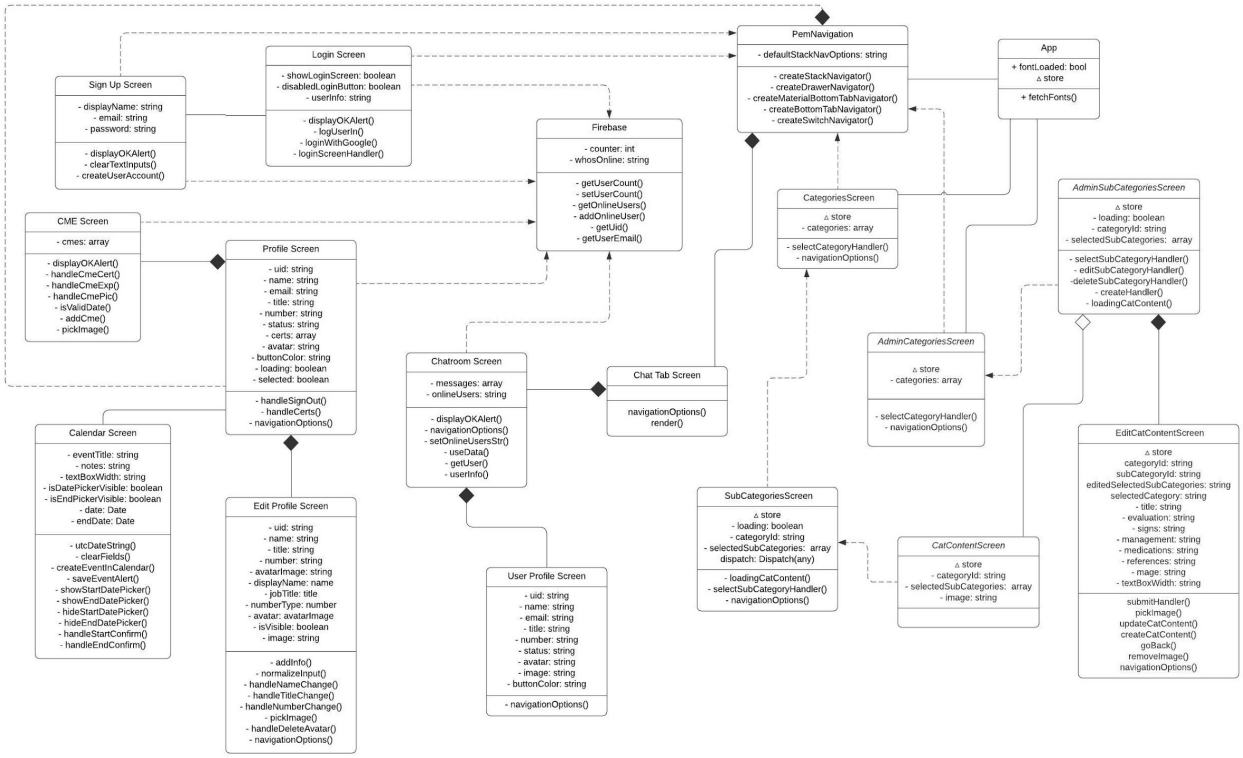


Figure 1: Class Diagram (Whole Project)

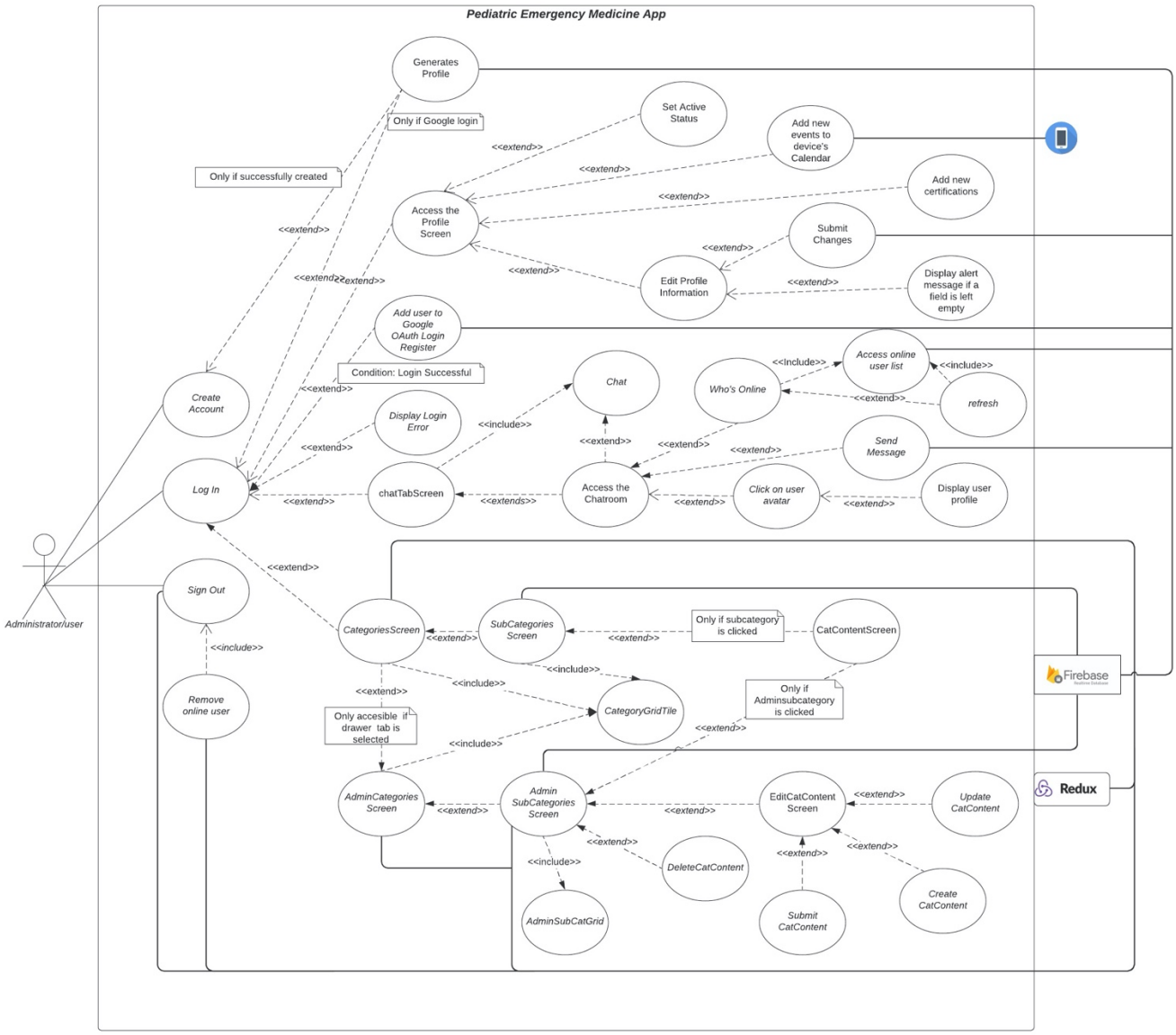


Figure 2: Use Case Diagram (Whole Project)

System Validation

Test Case ID	PEM-11
Purpose	Guarantee that accounts are created properly with Google Login
Preconditions	Firebase & Google API services are configured correctly and running well
Expected Results	Users successfully Login with Google Account and are welcomed
Actual Results	Users successfully Login with Google Account and are welcomed
Status	Pass

Test Case ID	PEM-13
Purpose	To guarantee that messages are properly sent and stay in the chat after all users disconnect.
Preconditions	Two or more users are connected to the chatroom.
Expected Results	The users are able to contact each other and anyone else in the chatroom, and the messages remain even after
Actual Results	The users are able to contact each other and anyone else in the chatroom, and the messages remain even after
Status	Pass

Test Case ID	PEM-15
Purpose	To make sure that the profile of the user is displayed correctly when someone accesses it from the chat.
Preconditions	User being clicked on has set up their profile and it’s connected to the chatroom
Expected Results	User is able to see the profile of the person, including name, CME and similar information, as made available by the other user. The user clicking is NOT able to edit the information.
Actual Results	User is able to see the profile of the person, including name, CME and similar information, as made available by the other user. The user clicking is NOT able to edit the information.
Status	Pass

Test Case ID	PEM-01
Purpose	Check that a user can successfully set up and edit their profile and save the changes.
Preconditions	Firebase Real-time Database is connected and the user is correctly logged in.
Expected Results	The user is able to change and edit all their information on the app, such as profile picture, full name, email and CMEs.
Actual Results	The user is able to change and edit all their information on the app, such as profile picture, full name, email and CMEs.
Status	Pass

Test Case ID	PEM-04
Purpose	Make sure the admin is able to access the app to add information to the relevant categories and subcategories.
Preconditions	The admin-user's account has been marked as an admin account.
Expected Results	The admin-user is able to access and edit all the information for each category and sub-category.
Actual Results	The admin-user is able to access and edit all the information for each category and sub-category.
Status	Pass

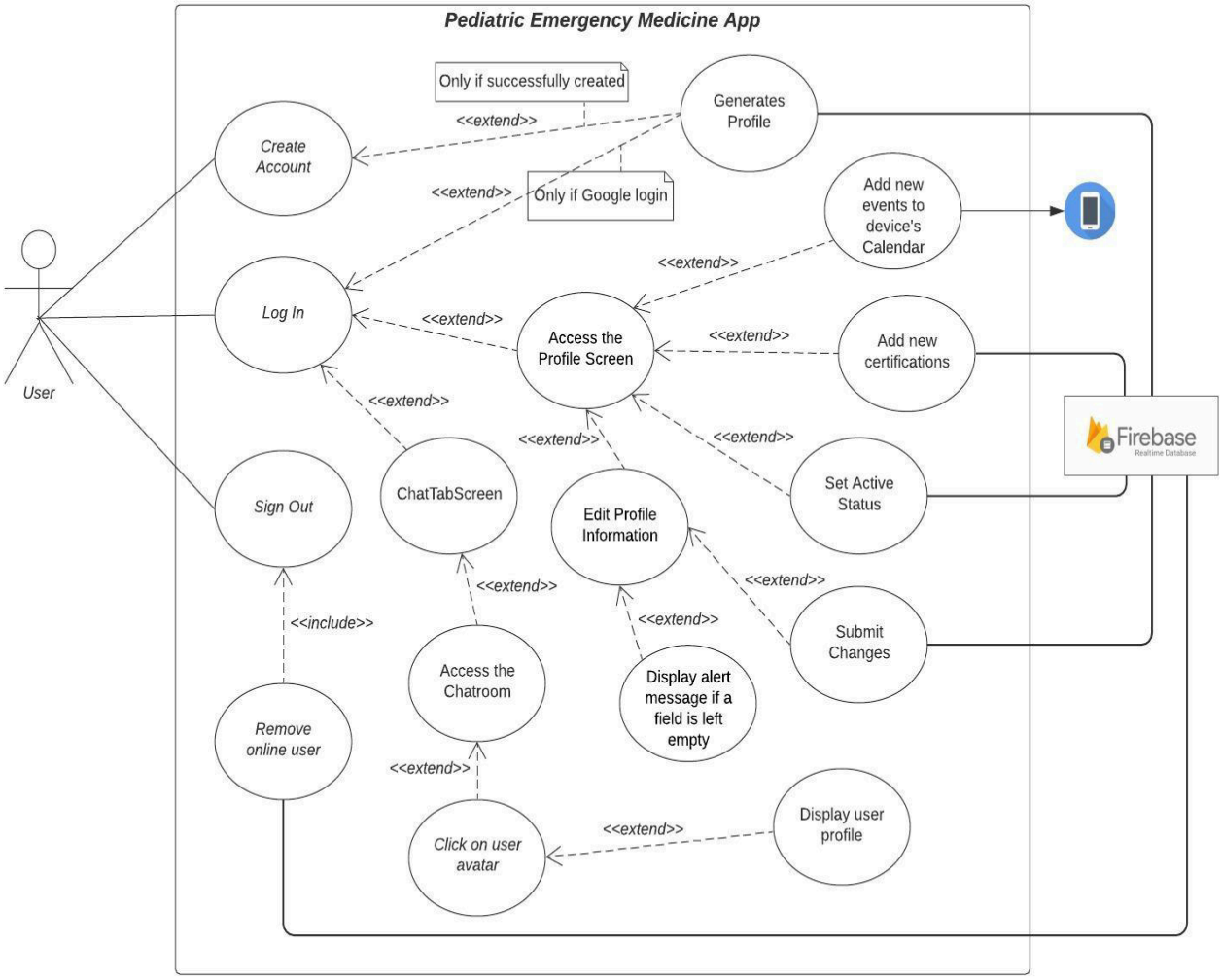
Glossary

- CME: Continuing Medical Education
- PEM: Pediatric Emergency Medicine

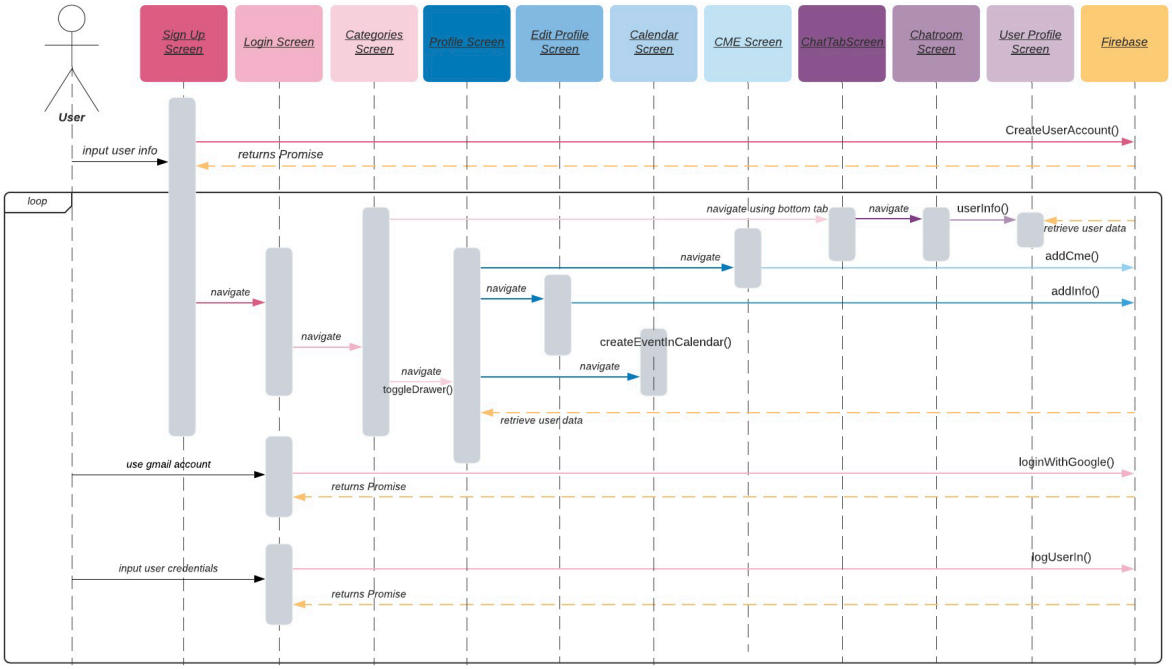
Appendix

Appendix A - UML Diagrams

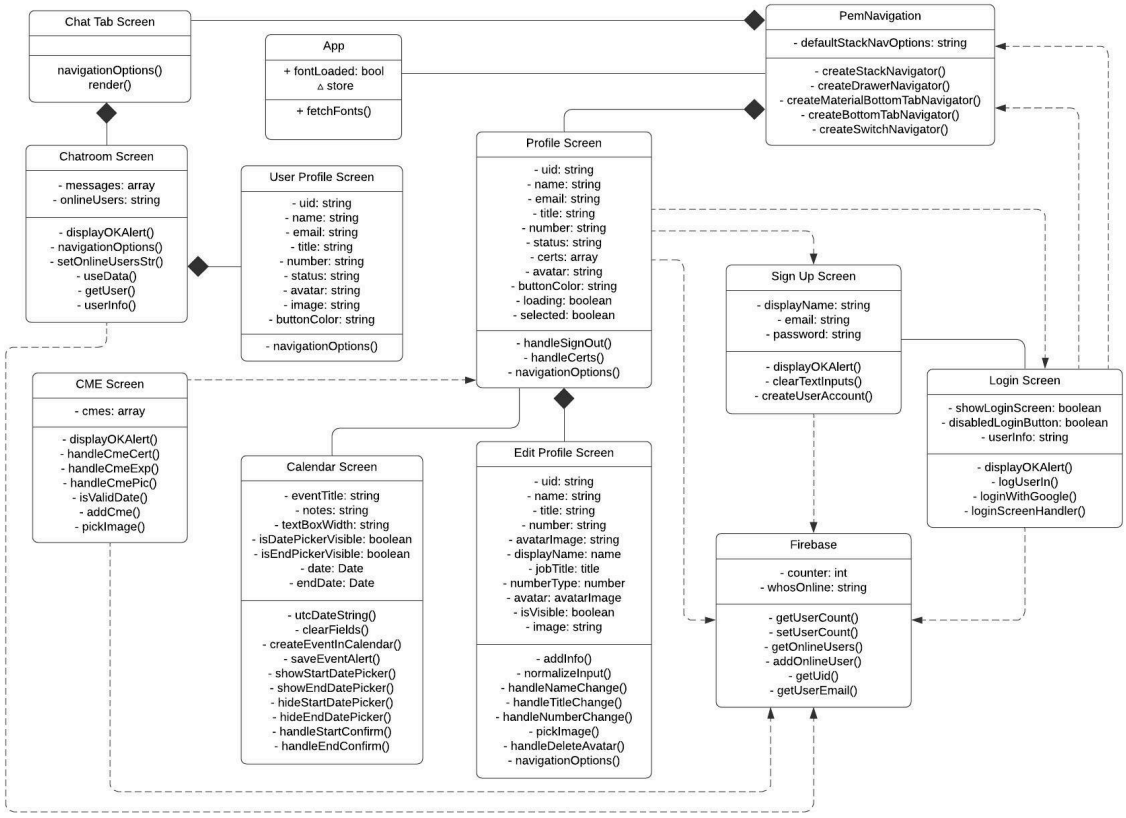
Use Case Diagram (Yeilys Fundora's Contribution)



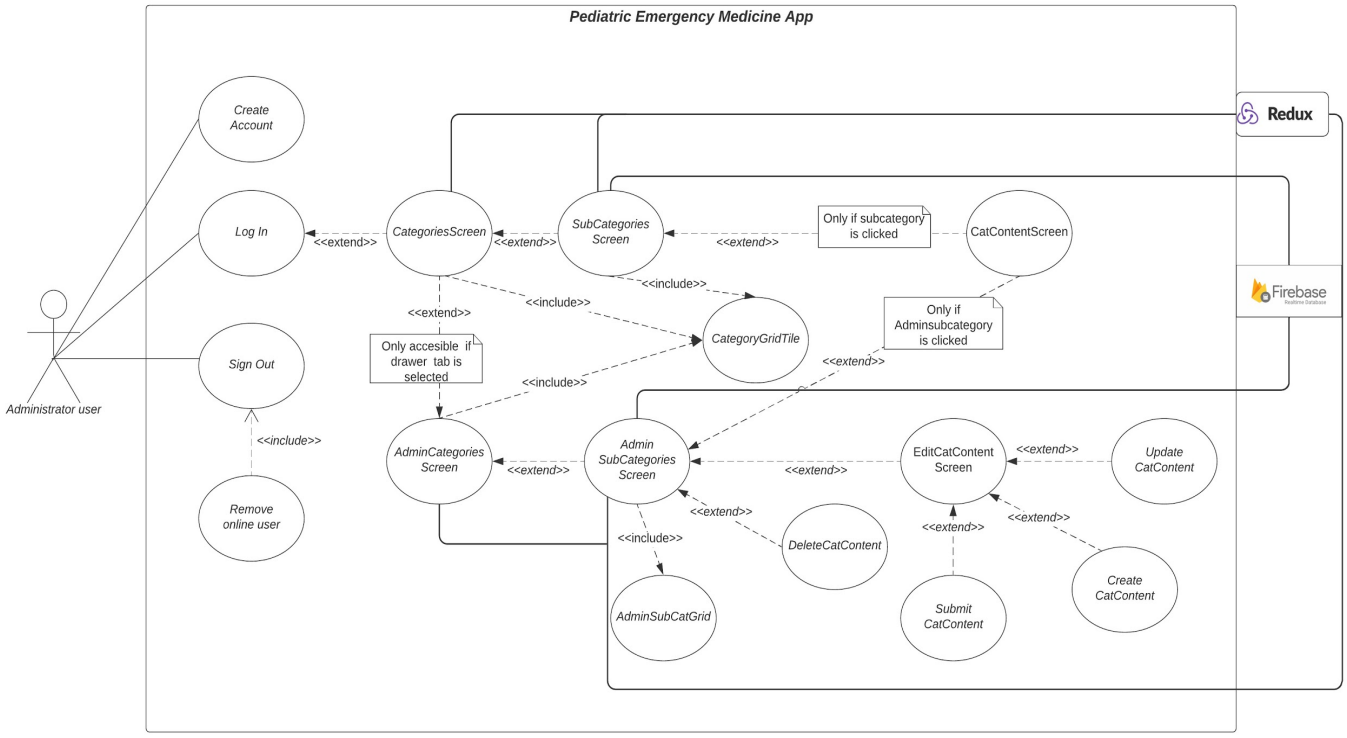
Sequence Diagram (Yeilys Fundora's Contribution)



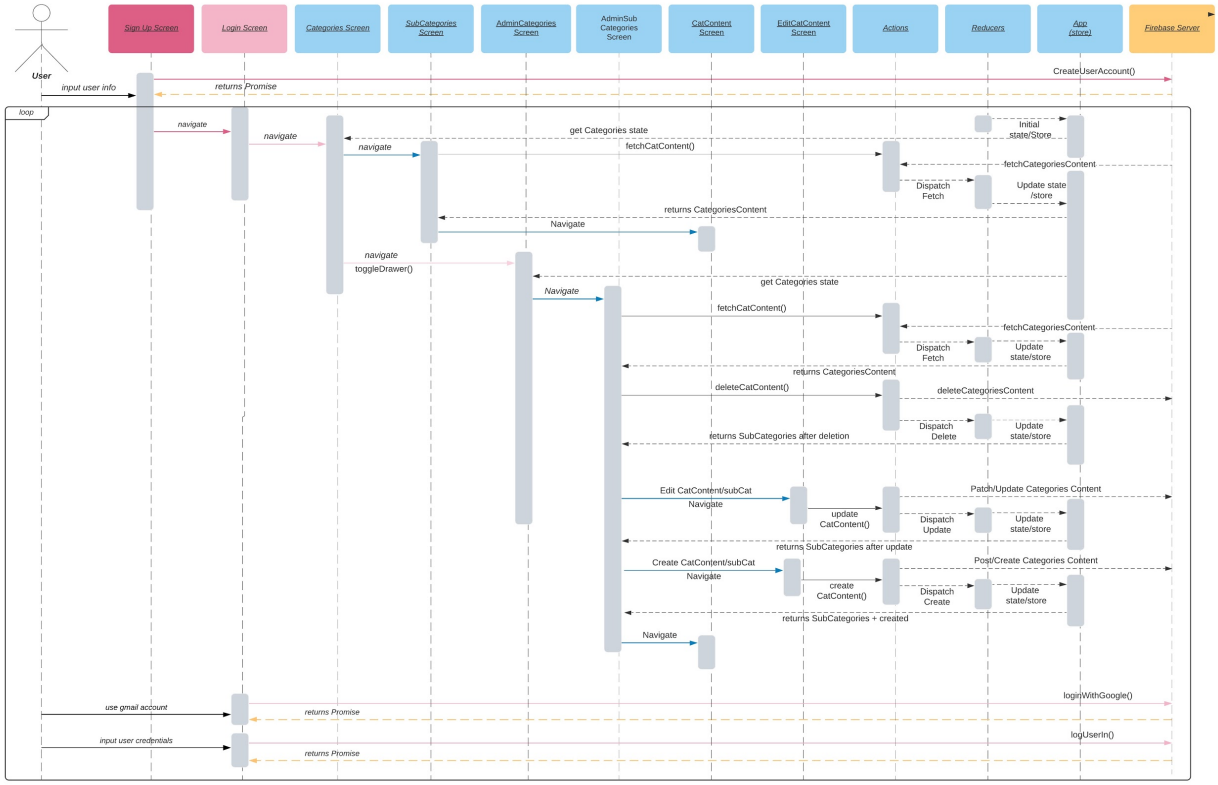
Class Diagram (Yeilys Fundora's Contribution)



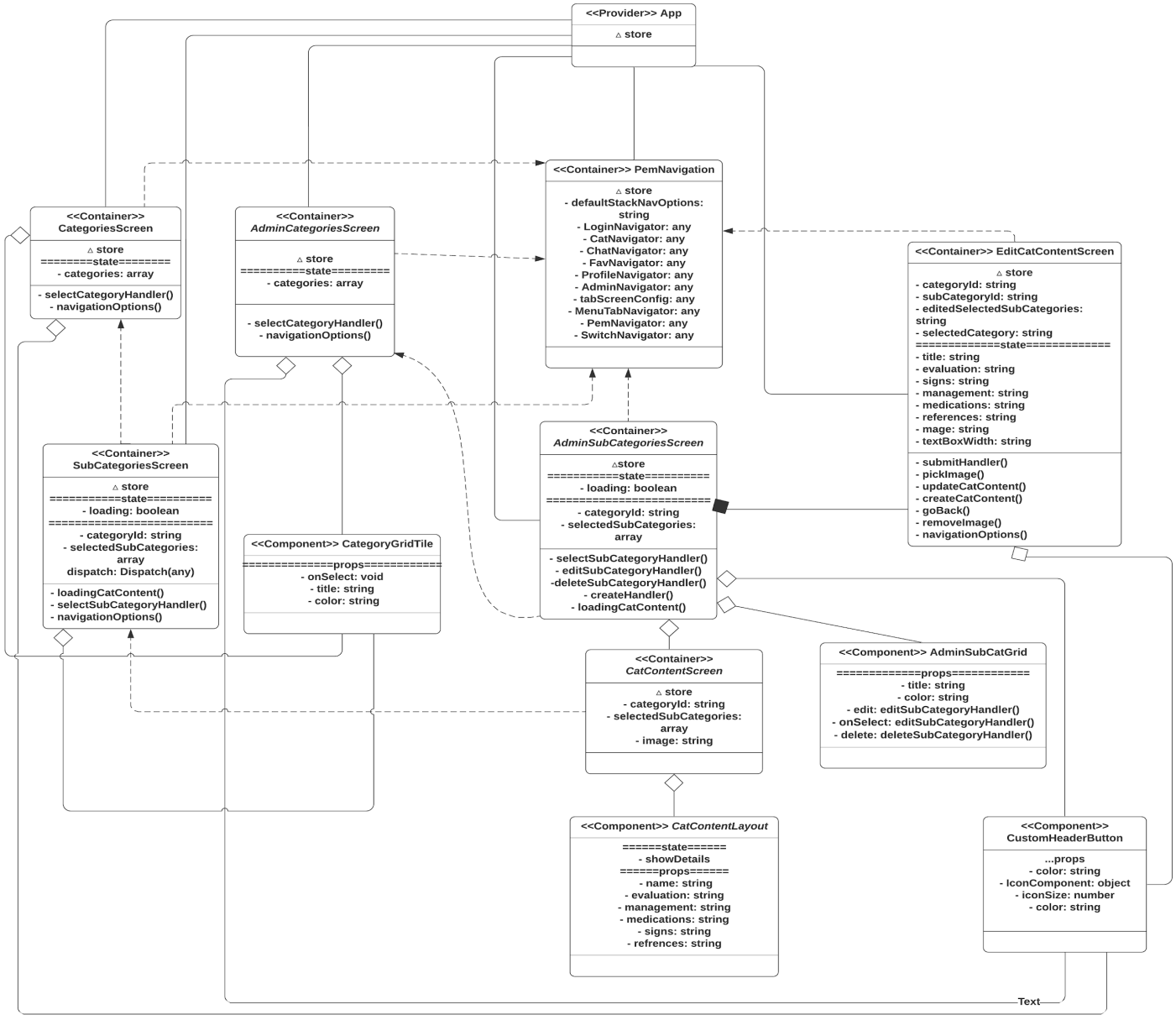
Use Case Diagram (Abel Gonzalez's Contribution)



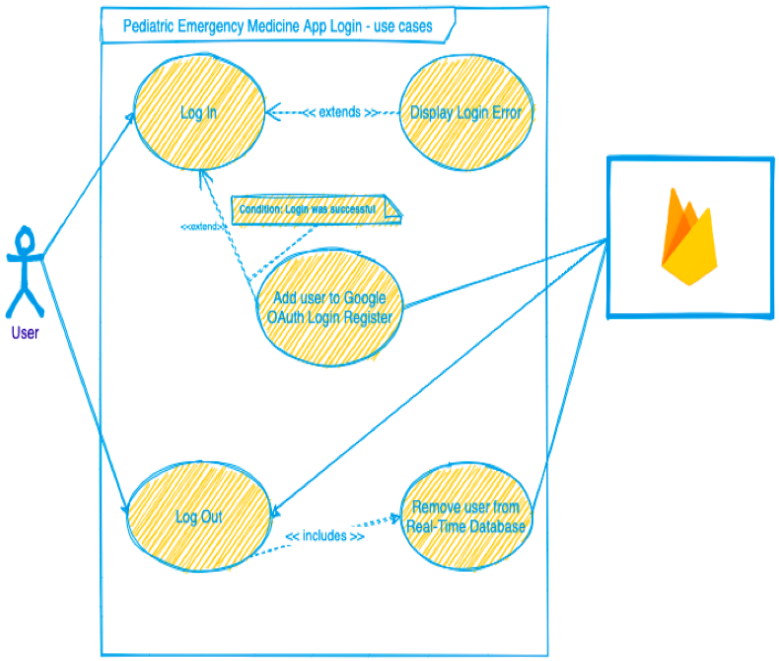
Sequence Diagram (Abel Gonzalez's Contribution)



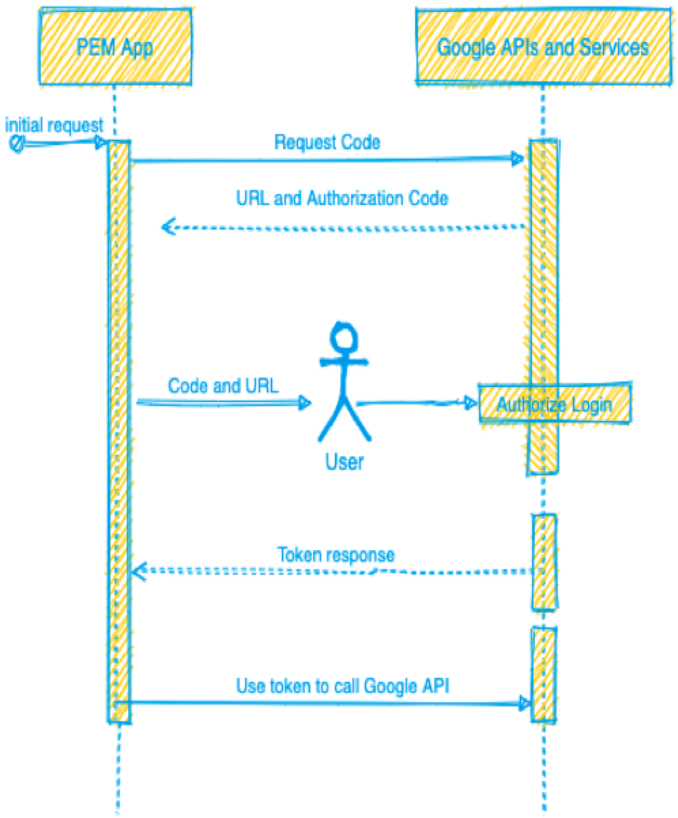
Class Diagram (Abel Gonzalez's Contribution)



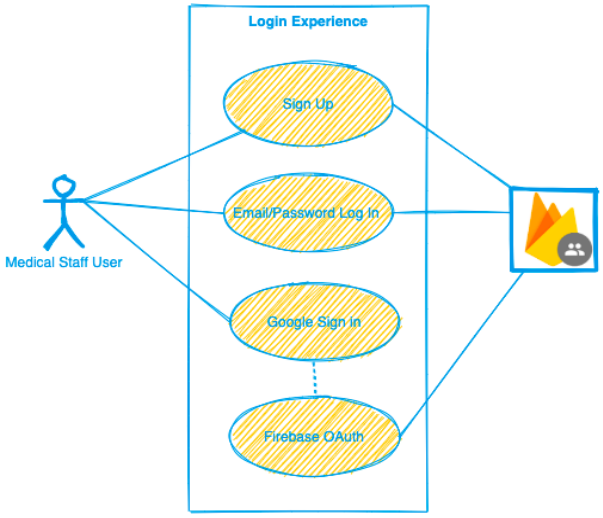
Use Case Diagram (Dianet Cruz's Contribution)



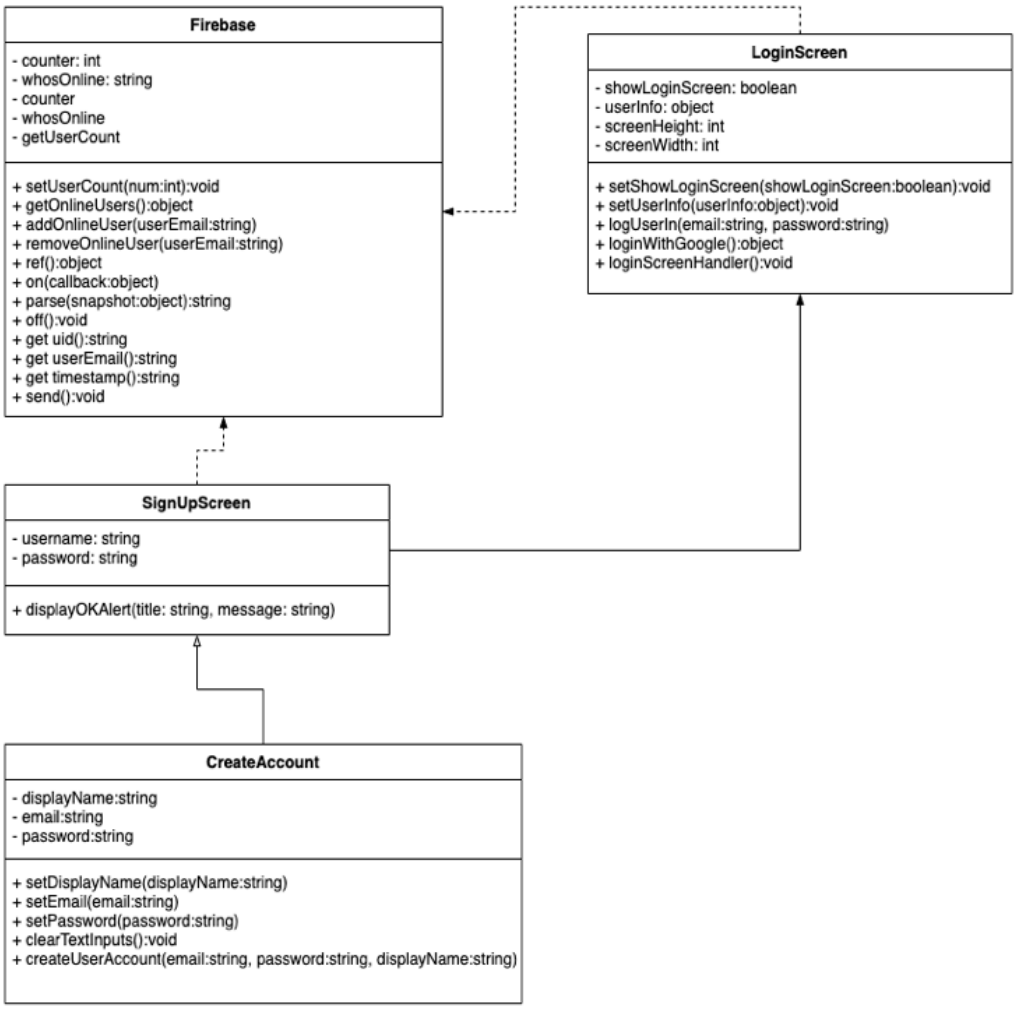
Sequence Diagram (Dianet Cruz's Contribution)



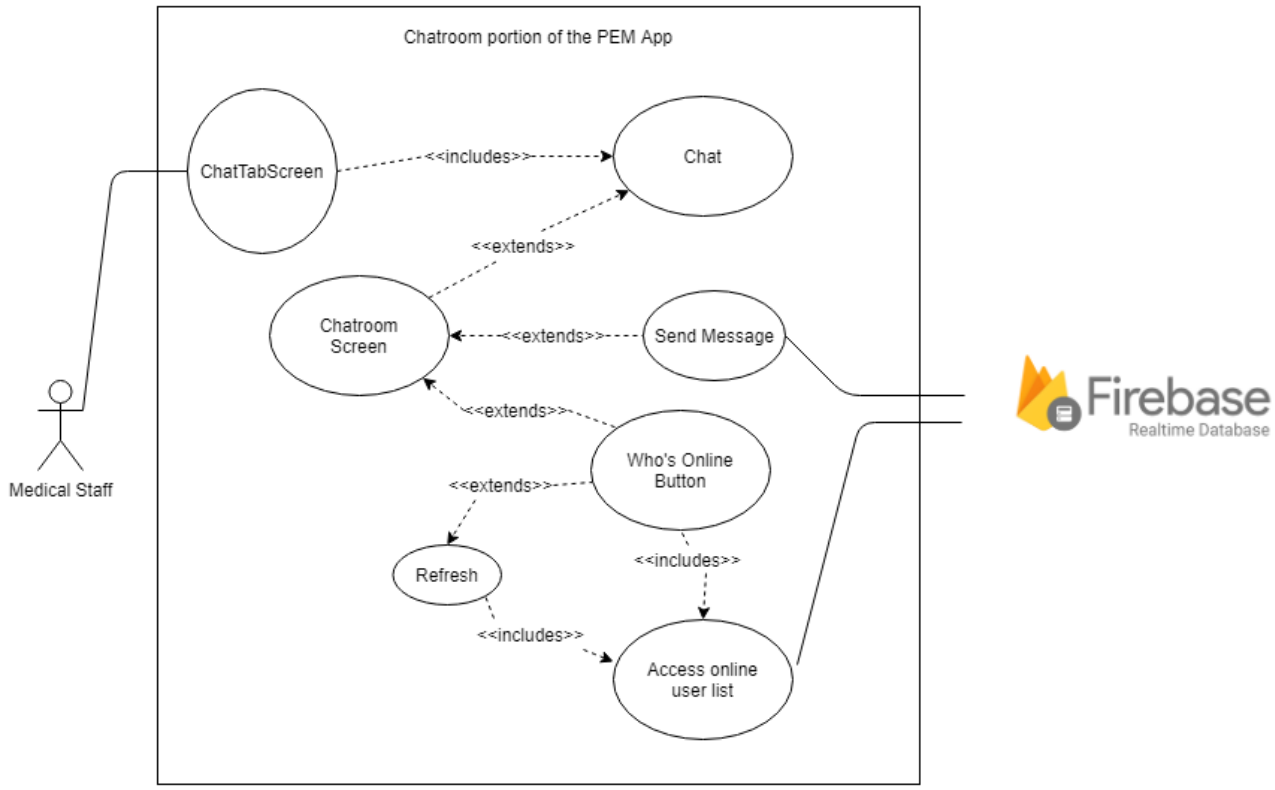
User Story Diagram (Dianet Cruz's Contribution)



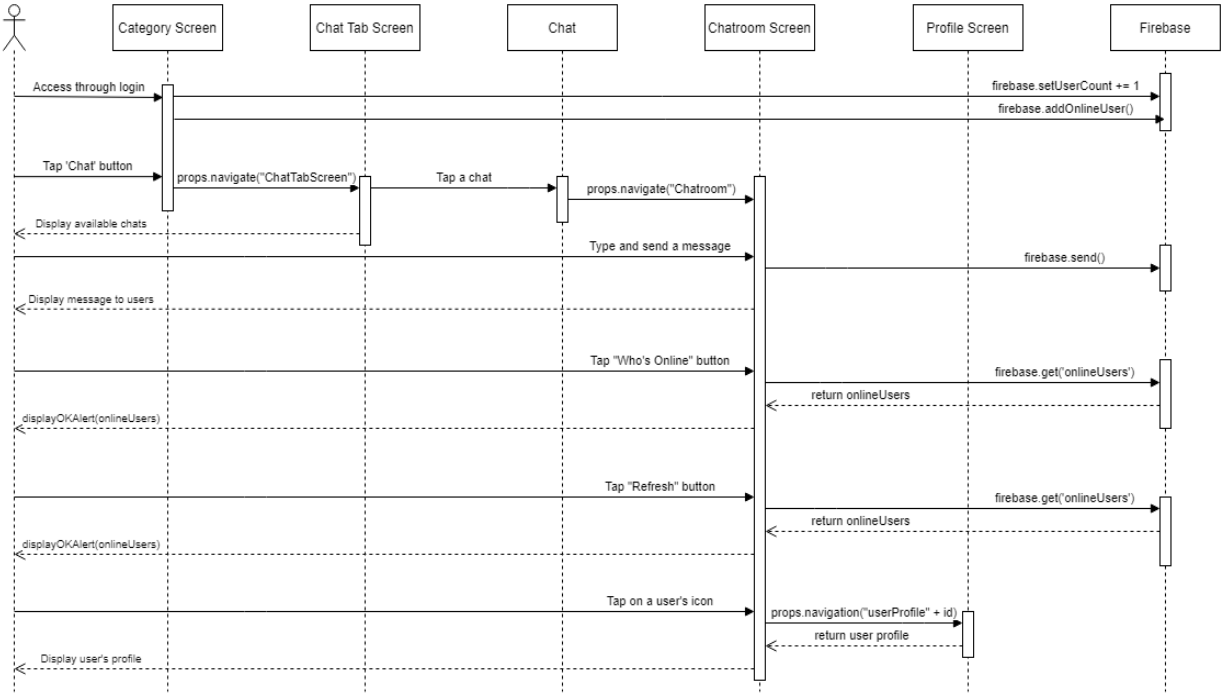
Class Diagram (Dianet Cruz's Contribution)



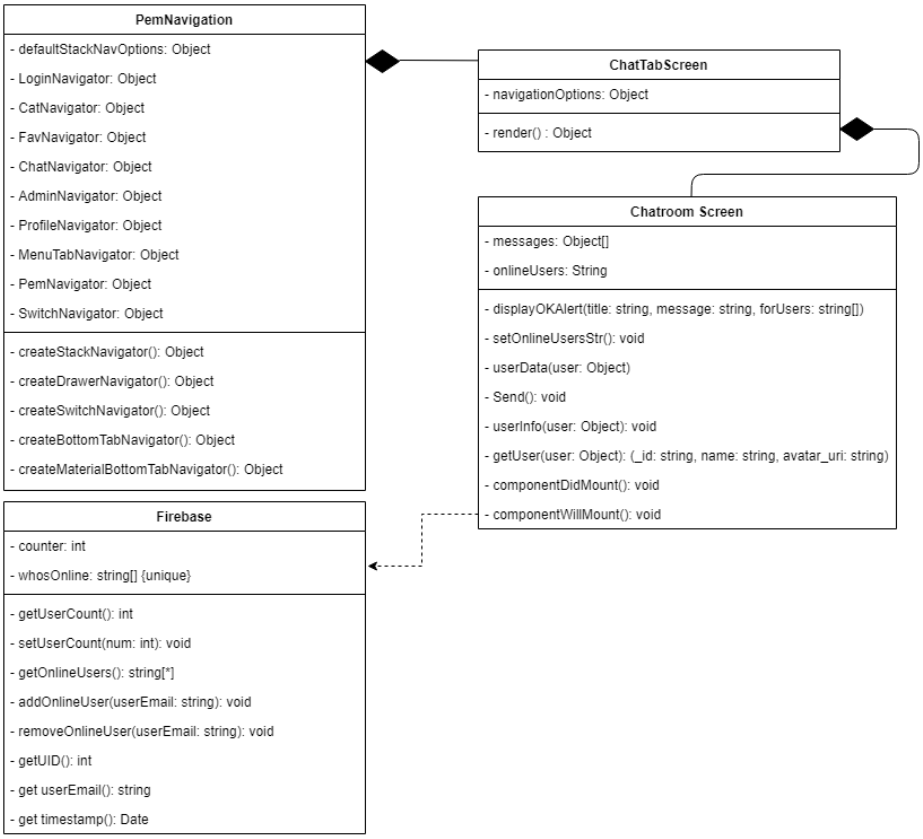
Use Case Diagram (Santiago de Irala's Contribution)



Sequence Diagram (Santiago de Irala's Contribution)



Class Diagram (Santiago de Irala's Contribution)



Appendix B - User Interface Design

Application UI on iPhone XR (iOS device)

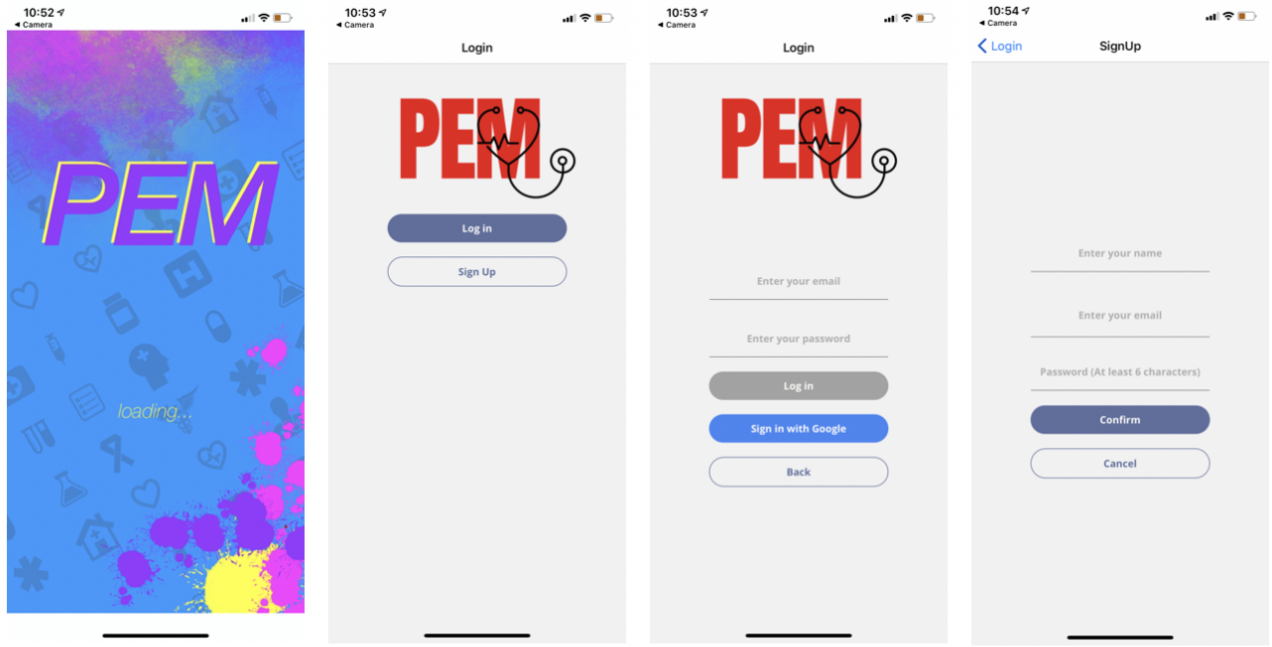


Figure 1: Splash, Sign up & Login Screen

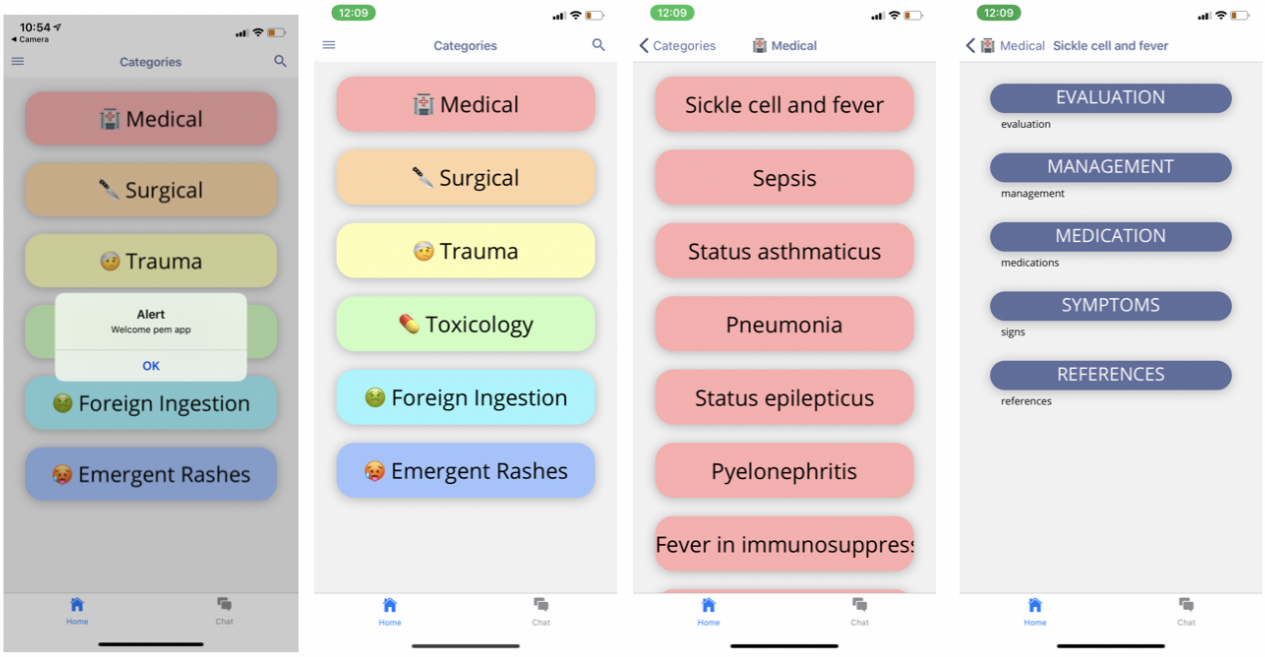


Figure 2: Categories, Subcategories & Content Screen

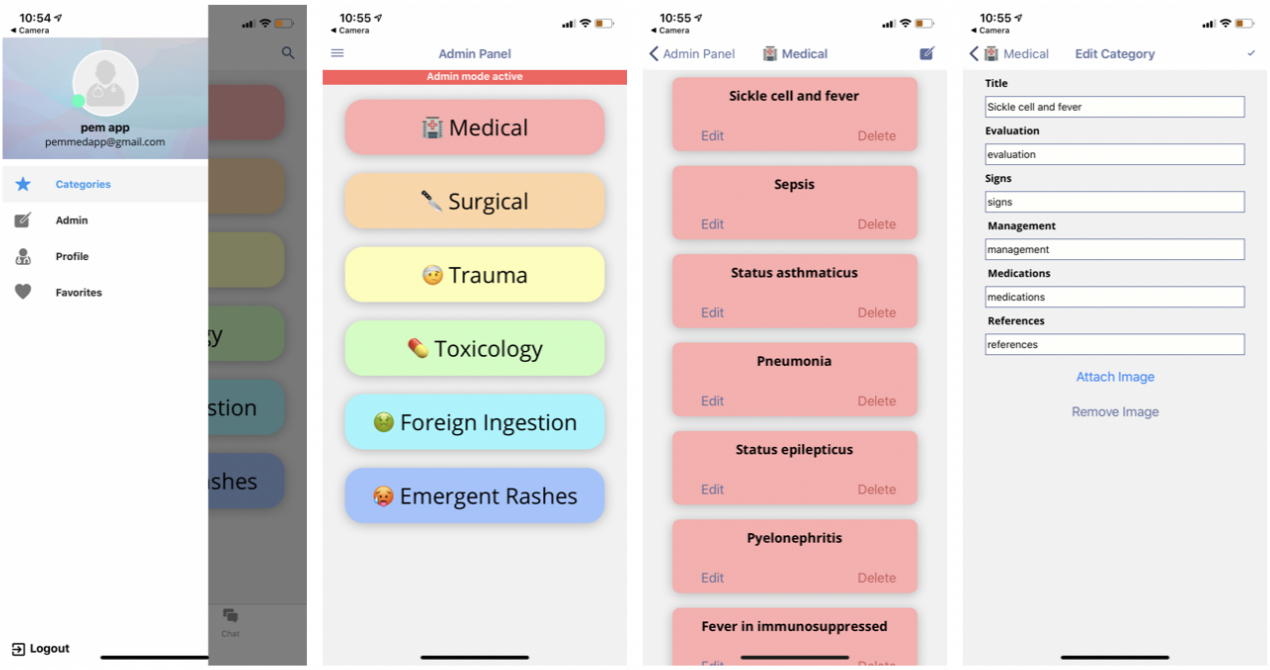


Figure 3: Drawer, Admin Panel, Edit Subcategories & Edit Content Screen

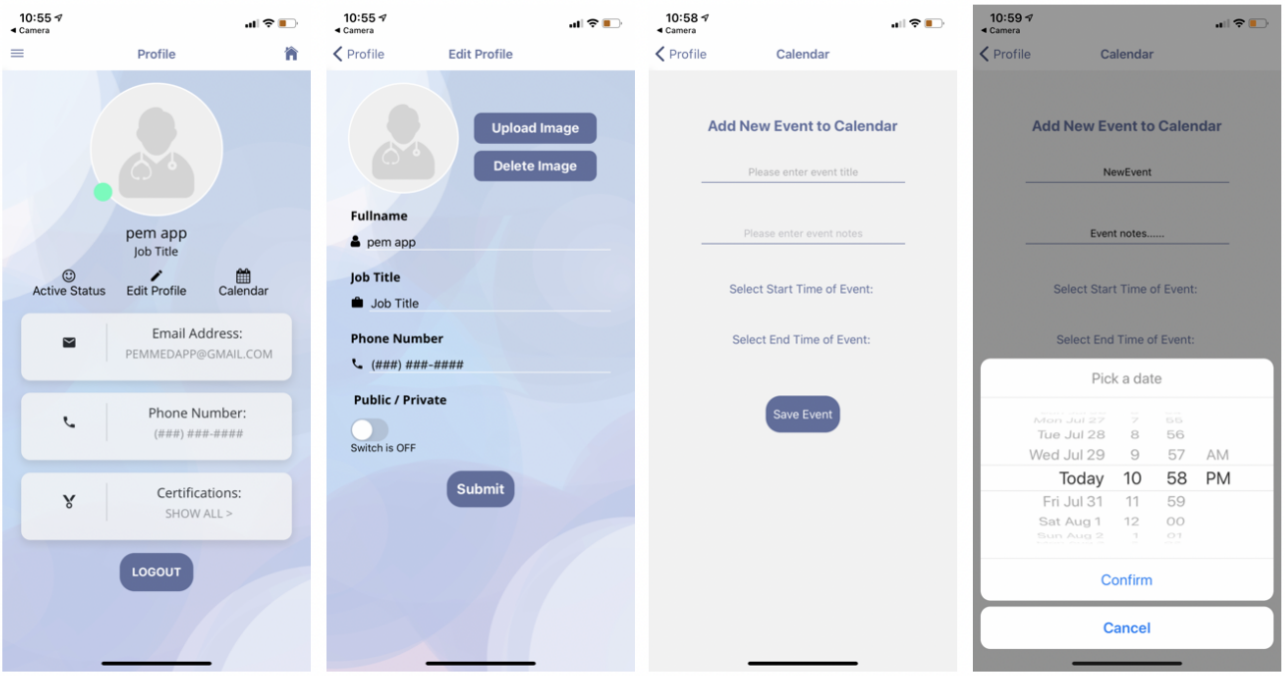


Figure 4: Profile, Edit Profile & Calendar Screen

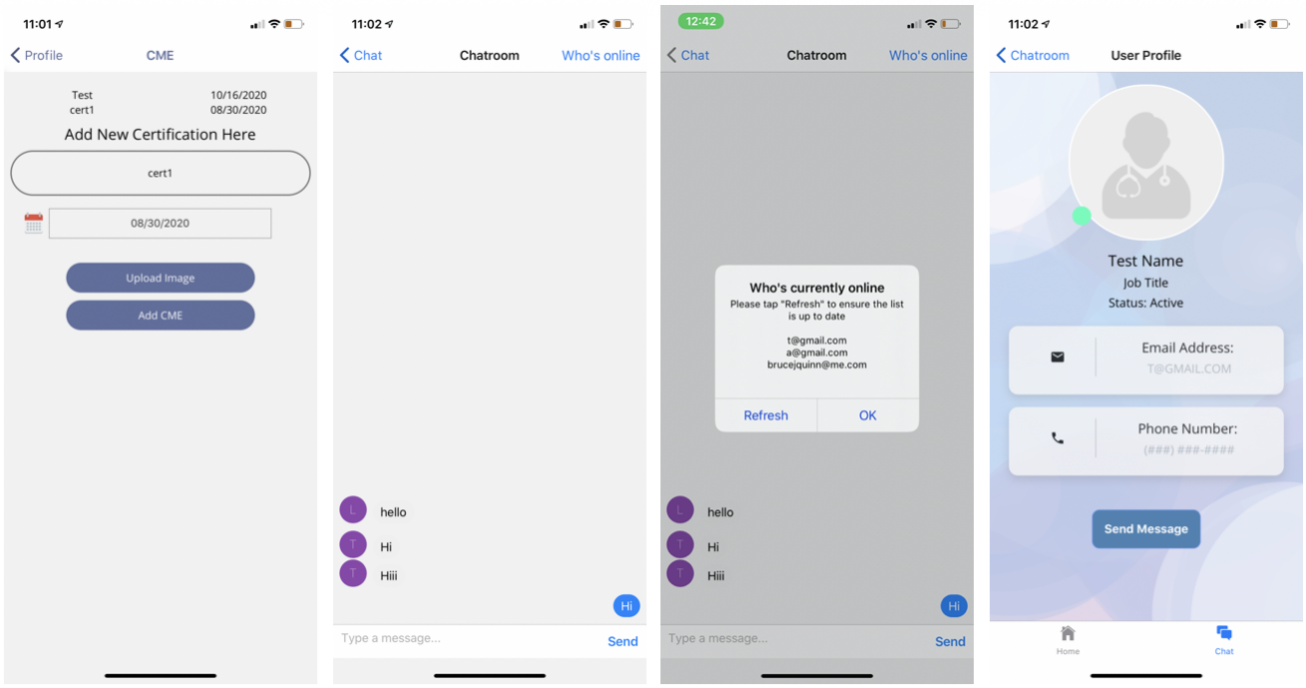


Figure 5: Certifications, Chatroom, & User Profile (when user's avatar is pressed) Screen

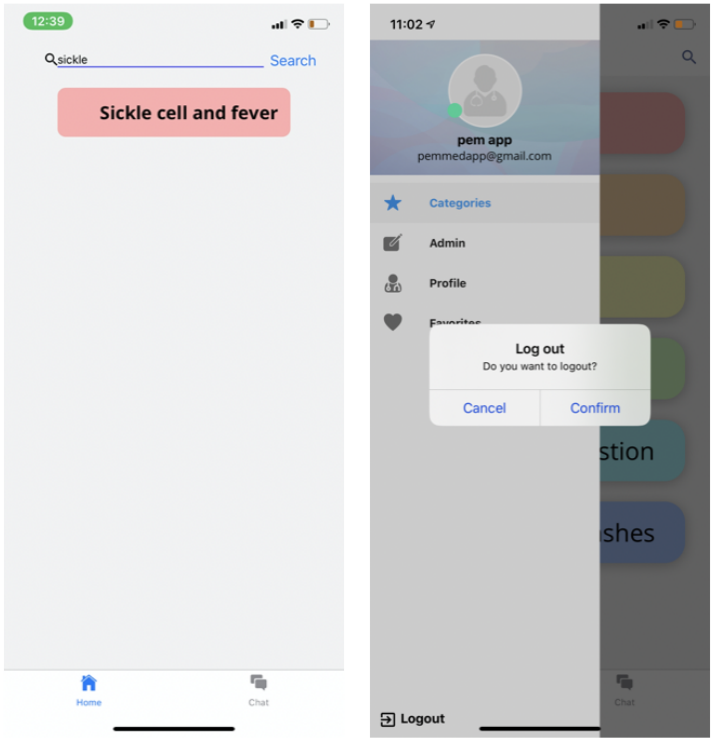
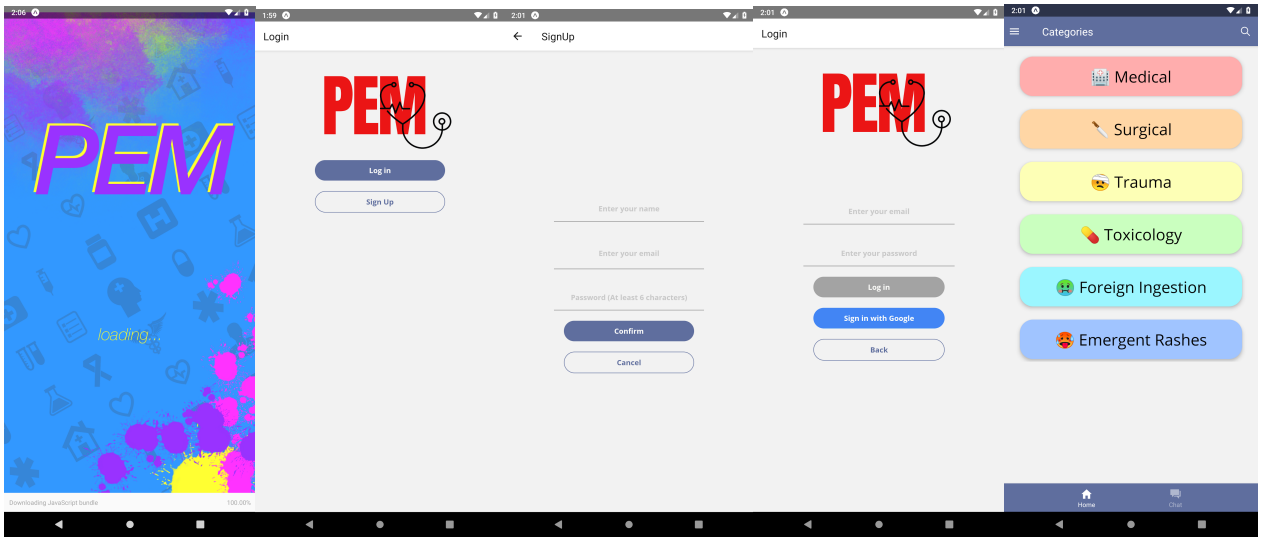
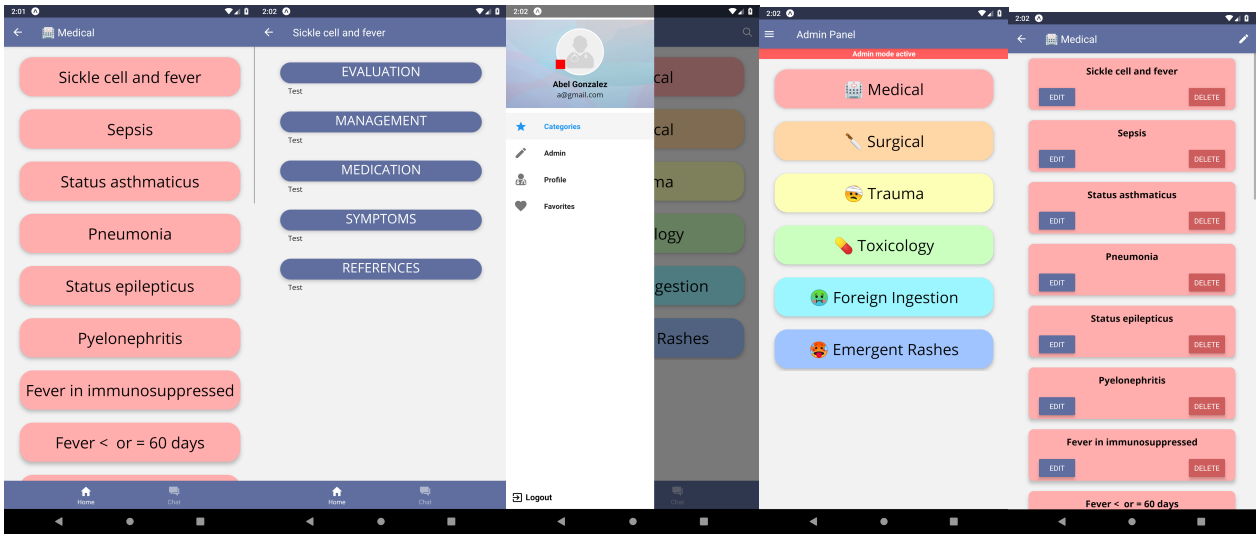
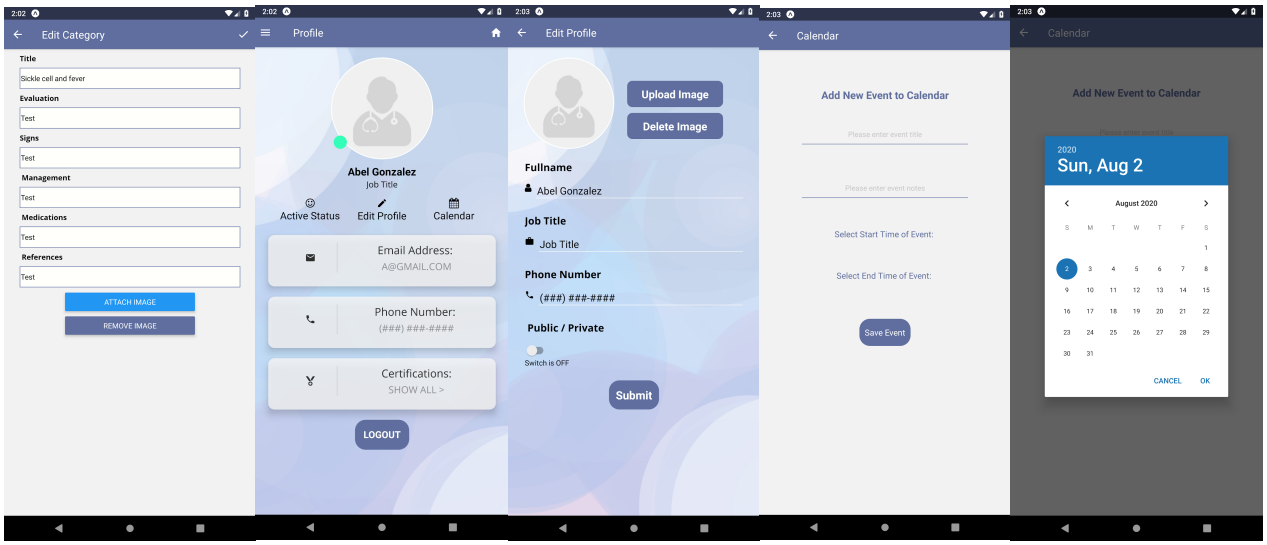


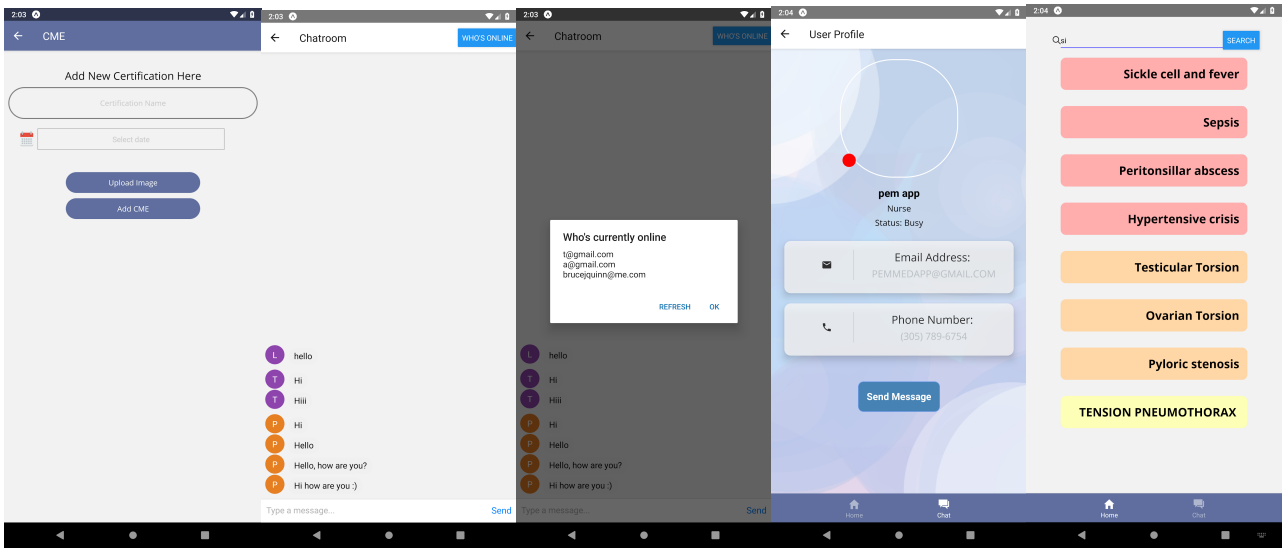
Figure 6: Search Screen and Log out

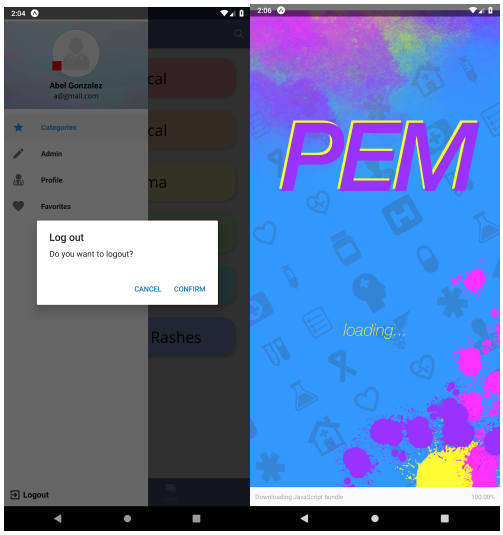
Application UI on Samsung Galaxy Note 10 (Android device)






Edit Category
Title
Sickle cell and fever
Evaluation
Test
Signs
Test
Management
Test
Medications
Test
References
Test
ATTACH IMAGE
REMOVE IMAGE
Profile
Abel Gonzalez
Job Title
Active Status
Edit Profile
Calendar
Email Address:
A@GMAIL.COM
Phone Number:
(###) ###-####
Certifications:
SHOW ALL >
LOGOUT
Edit Profile
Upload Image
Delete Image
Fullname
Abel Gonzalez
Job Title
Job Title
Phone Number
(###) ###-####
Public / Private
Switch is OFF
Submit
Calendar
Add New Event to Calendar
Please enter event title
Please enter event notes
Select Start Time of Event:
Select End Time of Event:
Save Event
Add New Event to Calendar
2020
Sun, Aug 2
August 2020
CANCEL
OK


CME
Add New Certification Here
Certification Name
Select date
Upload Image
Add CME
Chatroom
WHO'S ONLINE
hello
Hi
Hiii
Hi
Hello
Hello, how are you?
Hi how are you :)
Type a message...
Send
Who's currently online
t@gmail.com
a@gmail.com
brucejquinn@me.com
REFRESH
OK
User Profile
pem app
Nurse
Status: Busy
Email Address:
PEMMEDAPP@GMAIL.COM
Phone Number:
(305) 789-6754
Send Message
SEARCH
Sickle cell and fever
Sepsis
Peritonsillar abscess
Hypertensive crisis
Testicular Torsion
Ovarian Torsion
Pyloric stenosis
TENSION PNEUMOTHORAX


Abel Gonzalez
Categories
Admin
Profile
Favorites
Log out
Do you want to logout?
CANCEL
CONFIRM
Logout
PEM
loading...

Appendix C - Sprint Review Reports

Sprint 1

05/18/20 - 05/29/20
Summer 2020 Sprint Review 05/29/2020 8:30pm-9:02pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Sprint 2

06/01/20 - 06/12/20
Summer 2020 Sprint Review 06/12/2020 7:00pm-7:29pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Sprint 3

06/15/20 - 06/26/20
Summer 2020 Sprint Review 06/26/2020 9:04pm-9:23pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Sprint 4

06/29/20 - 07/10/20
Summer 2020 Sprint Review 07/09/2020 8:01pm-8:35pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Sprint 5

07/13/20 - 07/24/20
Summer 2020 Sprint Review 07/23/2020 7:40pm-8:00pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Sprint 6

07/27/20 - 07/31/20
Summer 2020 Sprint Review 07/31/2020 7:40pm-8:10pm

List of approved user stories:

- All user stories were approved for this sprint.

List of rejected user stories:

- None.

Appendix D – User Manual, Installation/Maintenance Document, Shortcomings/Wishlist

User Manual

SignUp: When the user opens the application, they will see two buttons: sign up or Login. When the user clicks the Sign up page, they will see a screen where the user can enter their credentials such as name,email, and password. The password must be at least 6 characters for the creation of the account. If not, it will not allow the user to create their account. Once the user clicks on the "Create account" button, they will be able to login with their new credentials.

Login: In the Login screen, the user will be able to put their credentials to login into the application such as their email and password. The Login button will turn bold once the credentials are successfully inputted. The user will also have the ability to Login using Google Sign-in. Once the user clicks on "Sign in with Google" they will be redirected to their Gmail accounts. Once the user successfully logs in, they are welcomed with a message corresponding to your Gmail name.

Categories: This screen includes six links to the subcategories. These are: Medical, Surgical, Trauma, Toxicology, Foreign Ingestion, and Emergent Rashes. When clicking on any of these categories, the user is going to find a list of the subcategories which belongs to that specific category.

Subcategories: Inside the subcategory screen the user is going to find a list of all the subcategories for the specific category they pressed. This list is ordered alphabetically and has the same color as the category. Once the user clicks a specific subcategory, it will take him/her to the CatContentScreen.

CatContentScreen: This screen displays the categories' information including title, evaluation, medication, management. symptoms, references and images, every section can be clicked to expand on information.

Search: When the user enters the search category, it will open up a screen with a search bar in it. After entering a search term, the application will display the subcategories which contain that specific term. These will be direct links to those specific subcategories. If the term the user enters is not part of any subcategory's title, the screen will remain blank. If the user doesn't enter a term at all, and simply clicks the search button, all subcategories will be displayed.

Chatroom: In the chatroom, the user can type anything and send messages to other users. If the user would like to know who is online and able to read their messages, they may click on the "Who's online" button to see who currently logged in.

Profile: The profile screen allows users to review personal information like name, job title, email, phone numbers, and set the user's active status. From here, you have several options, you can proceed to the Edit Profile Screen, in which users can edit the data being displayed in the profile, you can go to the Calendar Screen, where users can create events and save them to their device's calendar, or you can access the CME screen for adding and keeping track of medical certifications.

User Profile: The user profile is accessible from the chatroom and can be seeing once a user's avatar is pressed. This screen is only for online users to identify each other, getting access to a specific user's profile description, but no means to change what is shown.

CME: In the CME tracker, the user can enter the name of the certificate and choose a renewal date. If the renewal date is before the date it was being created, it will display an error message. After adding the correct date, the system will show the name and date of the certificate in a list above the two text input boxes.

Admin Panel: The Admin panel is composed out of 3 different screens. The first one is the AdminCategoriesScreen which is the same as the Categories Screen, except for the Admin mode active label. This screen is followed by the AdminSubCategoriesScreen which navigates to the EditCatContentScreen depending on the functionality and Category selected. This includes Edit, Delete and Add of Medical information, The last screen is the EditCatContentScreen, which depending on the selected functionality it will render the right feature. This screen integrates text input boxes for title, evaluation, medication, management, symptoms, references and removal and upload of images. Furthermore, through the AdminSubCategoriesScreen you can access the CatContenetScreen to see how the data was updated or created.

Installation/Maintenance Document

Prior to installing

1. Make sure you have an IDE available. **Visual Studio Code is highly recommended, as it is free and has a built-in terminal, which we will be using.**
2. Make sure you have Git installed, so you can run git clone.
3. Make sure you have Node.js installed. If you don't have it, you can find it here: https://nodejs.org/en/
4. Make sure you have an Android emulator available. If you don't, the Android Studio is recommended. Visit https://developer.android.com/studio to download the free version.
5. If you are a Mac user, you can use the Xcode simulator for running IOS devices. The only requirement is to have Xcode installed in your computer, which you can download from the Mac App Store.
6. Make sure you have the Expo CLI installed. To do this, simply open your terminal and run npm install expo-cli --global.
   Note: If using a Mac, run sudo npm install expo-cli --global.

Getting set up and cloning the project

1. Open Visual Studio and press Ctrl+` (that's the backtick key. It should be above your Tab key and to the left of your 1 key). This will open up Visual Studio Code's built-in terminal.
2. Use cd <directory> to navigate to the directory where you plan to clone the project and run npm install expo-cli --global to install the expo CLI.
   Note: If using a Mac, run sudo npm install expo-cli --global.
3. Run git clone https://github.com/diacruz/med-app.git in the same directory where you ran Step 3.

CONFIGURING FIREBASE (PART 1)

Setting up Firebase

1. Go to www.firebase.google.com and utilize the following credentials below to login into the pre-existing firebase account that was created.

   Email: pemmedapp@gmail.com
   Password: panthers2020

2. Once that's done, click "Go to console" in the upper right corner.
3. Open Visual Studio Code and in it, click "File" → "Open Folder", and then open the directory created when you completed Step 4 in the "Getting set up" section (it should be called PediatricEmergencyApp).
4. On the left hand side, you should see a list of directories and files. Open the one named "backend" and open "firebase.js".

5. You will need to create a firebaseConfig.js file and put the following firebase config keys as shown below:

```
const config = {
  apiKey: "AIzaSyB9-PR4nCk8T6nNtqvnMhYFLyxr7ZLXJV8",
  authDomain: "med-app-519aa.firebaseapp.com",
  databaseURL: "https://med-app-519aa.firebaseio.com",
  projectId: "med-app-519aa",
  storageBucket: "med-app-519aa.appspot.com",
  messagingSenderId: "692901117220",
  appId: "1:692901117220:web:b507bd72c70400551587b1"
}
export default config
```

   There is no need to delete this file or erase the keys when uploading to master because in ".gitignore" it will ignore this file and not upload it. Please do not share these keys with anyone.
6. Optionally, press Alt+Shift+F to format.
7. Save your changes and in the Firebase window from Step 8, click "Continue to console".

If you downloaded Android Studio (if you didn't, open your available Android emulator and go to "Running the project" section)

1. Open Android Studio.
2. Select the "Configure" option on the right-bottom corner and select "AVD Manager" from the list of options.
3. Click on the "+ Create Virtual Device" button at the bottom of the screen, and then choose the device you would like to install.
4. Click "Next".
5. Choose a system image or leave the default and click "Next".
6. Leave current settings and click "Finish".
7. Once installation is complete, double click the device to start it up.

Running the project

1. Go to Visual Studio Code, and in the terminal, run npm install. This will ensure you have all the packages needed for the application to work.
2. Once that finishes, make sure you have an Android emulator running, whether it be Android Studio or otherwise.
3. Run expo start --android. This will install the Expo client on your running emulator and start the project.
4. If using the Xcode simulator for Mac, run expo start –-ios.

Congratulations!

The Pediatric Emergency Medicine App is all set and ready to go!

Shortcomings/Wishlist

Shortcomings

- There is a bug when users in the chatroom click on another user's avatar and try to see their profile if that specific user does NOT have an avatar image uploaded.

Wishlist

- Implement a "Forgot Password" option in the Login Screen
- Allow users to chat with individual users as well as in a public chat room.
- Adding content to the Favorites Screen
- Set admin privileges to specific users
- The certification page needs to be fixed to save the uploaded cme image or file.
- Implement a remove/delete functionality for the certifications.

References

Expo:
https://docs.expo.io/

React Native:
https://reactnative.dev/docs/getting-started

Android Studio:
https://developer.android.com/studio

Firebase:
https://console.firebase.google.com

Redux:
https://redux.js.org/