

Florida International University
Knight Foundation School of Computing and Information Sciences

Software Engineering Focus

Final Deliverable

Pediatric Emergency Medicine App

Team Members: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Dianelis Insua, Felipe Arce

Product Owner(s): Jean Hannan, Bruce Quinn

Instructor: Masoud Sadjadi

The MIT License (MIT)
Copyright (c) 2016 Florida International University

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

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 fourth version of the Pediatric Emergency Medicine App.

This application will allow doctors, nurses, and other medical staff to have quick and easy access to vital information for common pediatric emergencies. It will also support account creation, a chatroom for the users, user account settings for handling user profiles, active status, creating calendar events, adding and reviewing certifications, and a full text search. 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

Commented [1]: still needs to be updated

INTRODUCTION 5

[CURRENT SYSTEM](#) 6

[PURPOSE OF NEW SYSTEM](#) 6

USER STORIES

[IMPLEMENTED USER STORIES](#) 7

[PENDING USER STORIES](#) 8

PROJECT PLAN

[HARDWARE AND SOFTWARE RESOURCES](#) 9

[SPRINTS PLAN](#) 10

[Sprint 1](#) 10

[Sprint 2](#) 9

[Sprint 3](#) 11

[Sprint 4](#) 12

[Sprint 5](#) 12

SYSTEM DESIGN

[ARCHITECTURAL PATTERNS](#) 14

[SYSTEM AND SUBSYSTEM DECOMPOSITION](#) 15

[DEPLOYMENT DIAGRAM](#) 17

[DESIGN PATTERNS](#) 18

SYSTEM VALIDATION 20

GLOSSARY 23

APPENDIX 23

[APPENDIX A - UML DIAGRAMS](#) 23

[APPENDIX B - USER INTERFACE DESIGN](#) 47

[APPENDIX C - SPRINT REVIEW REPORTS](#)

[APPENDIX D - USER MANUALS, INSTALLATION/MAINTENANCE DOCUMENT, SHORTCOMINGS/WISHLIST](#)

[DOCUMENT AND OTHER DOCUMENTS](#) 52

REFERENCES 68

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 with a forgot password, and a 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. In the profile, users will be able to access and edit their personal information, change their active status, review and add new documents, 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, 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:

- A new splash screen once the user opens the application.
- A friendly sign up screen with a 'Forgot password' option.
- A new and refined user interface and user experience throughout the application for iOS and Android devices.
- A profile implementation for users who create an account.
- A calendar screen where users are able to create events and save them to their device's calendar.
- A chat screen where users can create global chats around certain topics or create private chats with one or more online users.
- Removal of users from all private chats once the user logs out so that information sent in the chat is not available forever.
- Implementation of Administrator panel to Edit, Create and Delete Categories with Medical information.
- A screen that lists documents and certifications in the user profile where the user can also add and remove documents.
- A search bar that can be used to search across content on the application to quickly and easily find information.

Purpose of New System

The Pediatric Emergency Medicine App is in its fifth phase. The purpose of the new system is to improve the application by working on its performance, implementing new features such as, new UI/UX experience, profile integration with certification documents management, improvements in the chat and search features, fixing uploading issues and testing for usability and accessibility.

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 #1:** As a developer, I need to create GitHub workflow (Branches/Code Owner/Branch Protection) to improve the development process
- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application and grant users different permissions depending on their role
- **User Story #4:** As a user, I would like to find an alternative to Testflight for Android in order to test the app
- **User Story #5:** As a user I would like to remove the black gap in the categories screen
- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens
- **User Story #10:** As a developer, I need to create the ReadMe for the React app
- **User Story #12:** As a user, I want the changes made from the Admin Portal (web) to persist on the Admin Panel (mobile)
- **User Story #13:** As a user, I would like to search for answers to Questions already answered (Q&A Section)
- **User Story #14:** As a user I need to be able to have custom subheading for each category.
- **User Story #16:** As a user, I would like to remove the image icon from the treatments
- **User Story #18:** As a developer, I would like to fix the missing loading symbol on the login screen (Missing loading symbol on login bug)
- **User Story #19:** As a user I would like new treatments to be sorted alphabetically as I add them.
- **User Story #20:** As a user, I would like to know why I'm getting the expiration warning in order to avoid interruption to the test process.
- **User Story #21:** As a user, I would like the different topics to be sorted alphabetically to make the search easier
- **User Story #23:** As a user, I would like to remove the black gap from the categories Admin Screen
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content
- **User Story #27:** As user I would like to upload an image in the certifications screen and save it to the DB (currently not saving)
- **User Story #29:** As a user I would like to work on the app in horizontal orientation. (screen orientation vertical and horizontal)
- **User Story #31:** As a user, I would like for the edit and delete button text to appear properly on android devices

- **User Story #32:** As a user, I would like for the profile placeholder icon to not push buttons off screen on iOS devices.

Pending User Stories:

- **User Story #5:** As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)
- **User Story #7:** As a developer, I need to connect the front end of the Admin Portal (web) to Firebase
- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story #12:** As a user, I would like to be able to add more users to an already existing private chat with an "Add" button accessible from the Chat Details screen
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links
- **User Story #28:** As a user I need to upload images in subcategories without cropping them.

PROJECT PLAN

This section describes the planning that went into the realization of this project. This project incorporated agile development techniques and as such required the sprints to be planned. These sprint plannings are detailed in the section. 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:

Source Code Editor

Visual Studio Code, Atom.

Phones and Emulators

- Phones
- IOS iPhone 7, iPhone X , XR, 11 Pro, 12.
- Android Samsung Galaxy A11, S8, S20

Emulator

- Android Studio
- Pixel XL

Database/Storage

Firebase Realtime Database, Firebase Storage, Firestore, and Algolia.

User Authentication

Firebase Authentication.

Version Control

GitHub, Git

Sprints Plan

Sprint 1

Commented [2]: Added sprint stories for all members

Anabel Perez

- **User Story #1** : As a developer, I need to create a GitHub workflow (Branches/Code Owner/Branch Protection) to improve the development process.
- **User Story #6**: As a developer, I need to create the front end structure of the Admin Portal (web)
- **User Story #10**: As a developer, I need to create the ReadMe for the React app

Athenna Martinez

- **User Story #3**: As a developer, I want to grant users different permissions depending on their role.
- **User Story #5**: As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)

Ana Martinez

- **User Story #1**: As a developer, I need to create GitHub workflow (Branches/Code Owner/Branch Protection) to improve the development process

Peter Duarte

- **User Story #13**: As a user, I would like to search for answers to Questions already answered (Q&A Section)
- Kelvin Gonzalez
- **User Story #6**: As a developer, I need to create the front end structure of the Admin Portal (web)
- **User Story #4**: As a user, I would like to have access to the Admin Panel only if I am part of the admin group

Eduardo Dally

- **User Story #4**: As a user, I would like to have access to the Admin Panel only if I am part of the admin group
- **User Story #8**: As a developer, I need to create the repo with the proper permissions for the Admin Portal(web)
- **User Story #9**: As a user, I need to see the PEM logo on the Main and Home screens

Sprint 2

Anabel Perez

- **User Story #4**: As a user I would like something similar to Testflight for Android in order to test the app.
- **User Story #19**: As a user I would like new treatments to be sorted alphabetically as I add them.

Athenna Martinez

- **User Story #2**: As a developer, I need to create user roles and separate them into groups to improve application security and grant users different permissions depending on their role.

Ana Martinez

- **User Story #16**: As a user, I would like to remove the image icon from the treatments

- **User Story #21:** As a user, I would like the different topics to be sorted alphabetically to make the search easier.

Peter Duarte

- **User Story #13:** As a user, I would like to have a Q&A section where I can post questions and also search for previously answered questions.

Kelvin Gonzalez

- **User Story #7:** As a developer I need to log in to the admin portal

Eduardo Dally

- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens

Sprint 3

Anabel Perez

- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links.
- **User Story #21:** As a user I would like the different topics to be sorted alphabetically to make the search easier.
- **User Story #24:** As a developer, I would like to research and be able to deploy the app on Google Play for testing.
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content

Athenna Martinez

- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application security and grant users different permissions depending on their role.
- **User Story #26:** As a user I would like to have the treatments centered with the first letter of each word capitalized
- **User Story #22:** As the PO I would like to change the name of the app to PEM-NOW

Ana Martinez

- **User Story #21:** As a user, I would like the different topics to be sorted alphabetically to make the search easier.
- **User Story #23:** As a user I would like to remove the black gap from the categories Admin Screen

Peter Duarte

- **User Story #13:** As a user, I would like to have a Q&A section where I can post questions and also search for previously answered questions.

Kelvin Gonzalez

- **User Story #7:** As a developer I need to log in to the admin portal
- **User Story#6:** As a developer, I need to create the front end structure of the admin portal (web)

Eduardo Dally

- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens

Sprint 4

Anabel Perez

- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links.
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content

Athenna Martinez

- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application security and grant users different permissions depending on their role.
- **User Story #26:** As a user I would like to have the treatments centered with the first letter of each word capitalized
- **User Story #22:** As the PO I would like to change the name of the app to PEM-NOW

Ana Martinez

- **User Story #14:** As a user I need to be able to have custom subheading for each category.
- **User Story #12:** As a user, I would like to be able to add more users to an already existing private chat with an "Add" button accessible from the Chat Details screen.

Peter Duarte

- **User Story #13:** As a user, I would like to have a Q&A section where I can post questions and also search for previously answered questions.

Kelvin Gonzalez

- **User Story #7:** As a developer I need to log in to the admin portal
- **User Story#6:** As a developer, I need to create the front end structure of the admin portal (web)

Eduardo Dally

- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens
- **User Story #29:** As a user I would like to work on the app in horizontal orientation. (screen orientation vertical and horizontal)

Sprint 5

Anabel Perez

- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen

- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like font and links.

Athenna Martinez

- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application security and grant users different permissions depending on their role.
- **User Story # 18 :** As a developer, I would like to fix the missing loading symbol on the login screen (Missing loading symbol on login bug)
- **User Story #31:** As a user, I would like for the edit and delete button text to appear properly on android devices
- **User Story #32:** As a user, I would like for the profile placeholder icon to not push buttons off screen on iOS devices.

Ana Martinez

- **User Story #14:** As a user I need to be able to have custom subheading for each category.
- **User Story #12:** As a user, I would like to be able to add more users to an already existing private chat with an “Add” button accessible from the Chat Details screen.

Peter Duarte

- **User Story #13:** As a user, I would like to have a Q&A section where I can post questions and also search for previously answered questions.

Kelvin Gonzalez

- **User Story #7:** As a developer I need to log in to the admin portal
- **User Story#6:** As a developer, I need to create the front end structure of the admin portal (web)

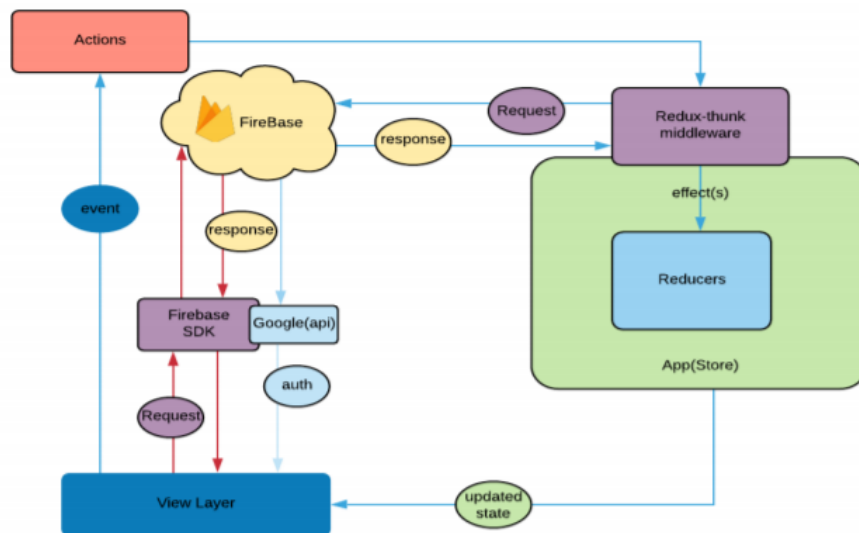
Eduardo Dally

- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens
- **User Story #29:** As a user I would like to work on the app in horizontal orientation. (screen orientation vertical and horizontal)

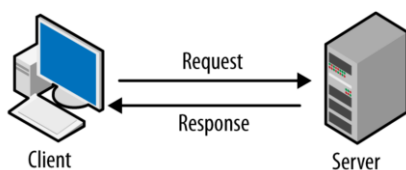
SYSTEM DESIGN

This section contains information on the design decisions that went into this project. The architecture patterns are outlined and explained. The entire system is shown in a package diagram and the subsystems are explained. Finally, the design patterns used in the project are discussed.

Commented [3]: added chart from last group's system design as a placeholder but I think it could stay since the system design has not really changed much. We can revise this in the upcoming days



Architectural Patterns



For this project we decided to use a Client-Server architectural pattern. On the left is a very simple diagram of what this would look like. The Client (the application running on the user's device) requests services from the server (application content, chatroom messages, user profile, etc.).

This architecture was preferred since it was known that a server would have to constantly listen for when the client asks for necessary resources. However, the architecture used in this project slightly expands on the Client-Server architecture 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,

Commented [4]: working on architectural pattern

Commented [5]: ill be making some revisions/additions to this in the next 2 days

logout, profile generation and management, sending and receiving of messages, and user CME certificates. These events utilize the server to update the database in addition to serving the content. The whole application uses redux to manage application states, logic and maintain the entire application in a single immutable state tree (object), which can't be changed directly. These changes can only be made 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

In this section we will go into detail about the system and subsystem deposition. We will explain the structure of the application. The application is divided into five main folders: views, components, constants, context and function. The views folder contains the files that allow the user to communicate with the application. The components folder contains the models that permit the reuse of such components within the application. The constants folder contains configuration files that stay the same throughout the app. The context folder contains all of the files that handle the interactions between the views and the functions. The functions folder contains all of the Google Cloud functions that handle the user creation and authentication process.

The 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. Instead, the screens that the user interacts with, will send information for those files 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 throughout the application without having to go through all of them 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 folder contains the navigation structure of the applications. The screens folder contains all the different screens throughout the application. The store folder contains 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 permissions for users to upload photos.

Frontend

The frontend of this application was built using React Navigation and React Native. There are various important components that allow for the frontend of this application to be entirely

Commented [6]: working on system and subsystem decomposition.

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. The screens are being exported so that the navigator can then import them. We will talk about this next.

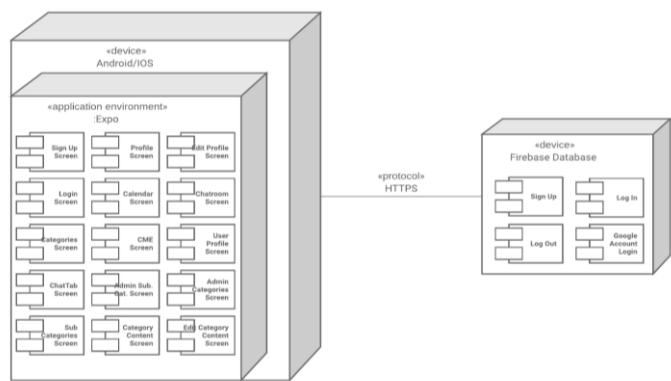
Navigator

The navigator is what is being used to allow the navigation between screens. One of the navigators used is called a "stack navigator". It is called this because all the screens the user visits are pushed onto a stack. Another navigator used is the "drawer navigator", which implements a side menu and allows the user to navigate to other screens in the app more easily. Next, there is a bottom tab navigator which lets you transition between different routes using a tab bar on the bottom of the screen. Finally, the main navigator is called 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 being to only ever show one screen at a time.

Backend

The backend of the application runs on Firebase with help from Algolia for the text search. Firebase manages all the user accounts, tracks the logged users, the chatroom, CME, profile, and is where all the static subcategory content is held. The majority of the screens are tied to Firebase in some way, whether it's to check if the user has logged into their account or if it's to pull the correct subcategory content from Firebase Realtime Database. The way it works is that screens will send a request for whatever they need to Firebase. Following that, the data is returned asynchronously to the screen. Due to it returning asynchronously, some of the screens use their state to manage whether the data has been returned or not. It is for this reason that the subcategory content screen is able to say "Loading..." until the data is returned. To implement the search feature there are two cloud functions that update the changes in Firebase into Algolia. Algolia creates an index of the content and then returns results for any of the search queries.

Deployment Diagram



Design Patterns

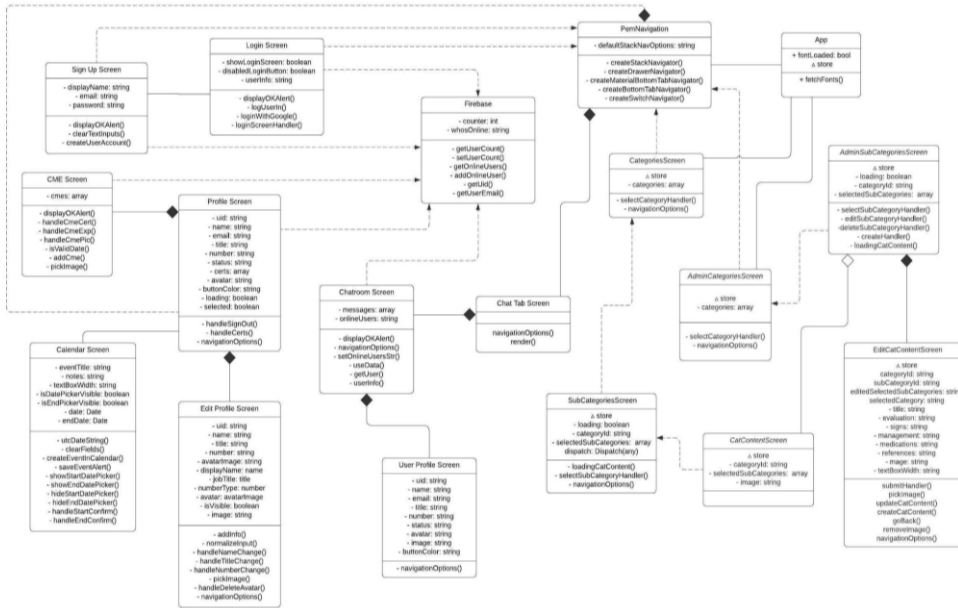


Figure 1: Class Diagram (Whole Project)

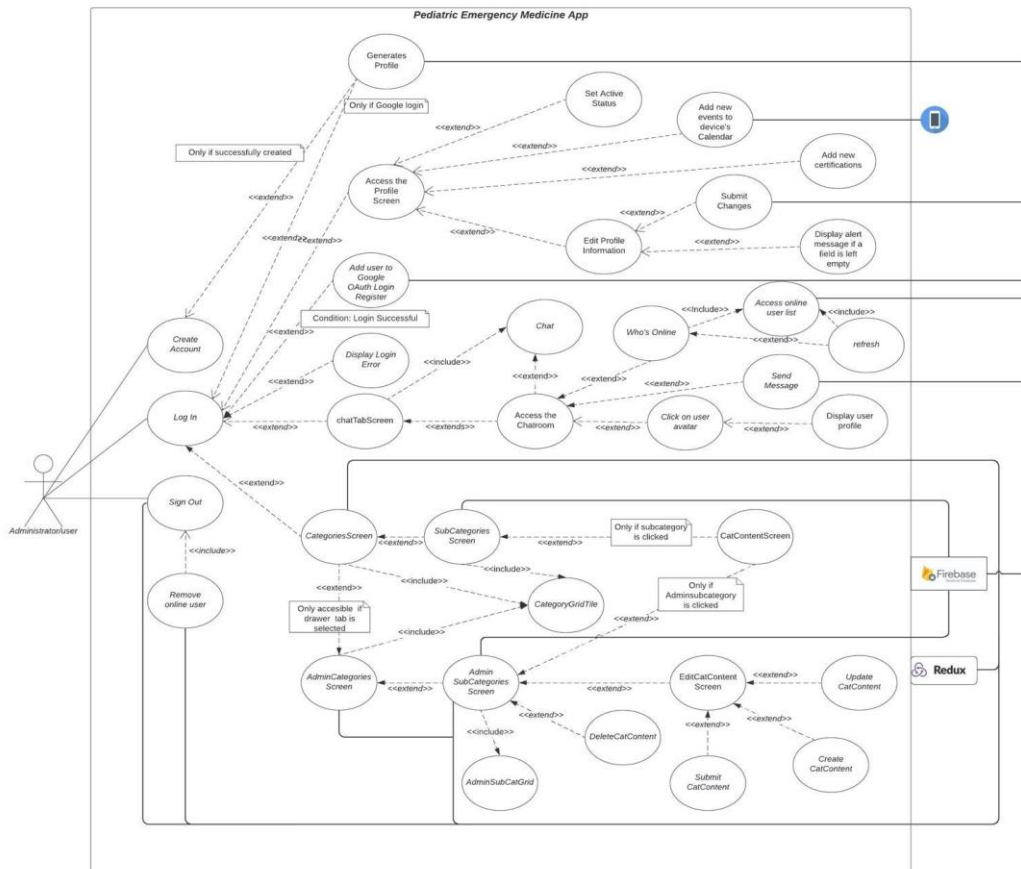


Figure 2: Use Case Diagram (Whole Project)

SYSTEM VALIDATION

Test Case ID	PEM-Chat Session
Purpose	To guarantee that the user is removed from all private chats when they logout.
Preconditions	Users are a part of one or more private chats.
Expected Results	When a user logs out, they are removed from all private chats in the firebase.
Actual Results	Users are removed from all private chats on logout.
Status	Pass

Test Case ID	PEM-Chat Management
Purpose	To test all functionality of chat management.
Preconditions	Users are logged in and there are one or more online users.
Expected Results	Users can create a global chat or private chat and send messages in real time. The user can also view chat details to remove users from chat or delete chat all together.
Actual Results	Chat is created successfully in firestore and messages are sent in real time. When a chat is deleted, all the messages are deleted as well in firestore.
Status	Pass

Test Case ID	PEM - Categories Subheadings
Purpose	To test all functionality of customized subheadings for each Category
Preconditions	Users are logged in: - From the Categories Screen select a category then select an item and notice the subheadings. Repeat the above to test the rest of the categories' subheadings. - From the Admin Categories Screen select a category then select an item and notice the subheadings. Repeat the above to test the rest of the categories' subheadings.
Expected Results	Each item listed within the Categories and Admin Categories screens should have the subheadings corresponding to its category.
Actual Results	Subheadings per Category: - Medical & Surgical: Evaluation, Differential Diagnoses, Management, Medications, References - Trauma & Toxicology: Evaluation, Signs/Symptoms, Management, References - Foreign Ingestion: Evaluation, Management, References
Status	Pass

Test Case ID	PEM - Screen Orientation
Purpose	Test that the application transitions from portrait to landscape mode depending on device orientation.
Preconditions	Device must have screen lock disabled.
Expected Results	The user interface recognizes that the device has been rotated and switches the user interface to landscape mode.
Actual Results	The user interface stays locked in portrait mode.
Status	Fail

Commented [7]: Done

Test Case ID	PEM - Non Admin User
Purpose	To avoid non admin users from editing, deleting, or adding content
Preconditions	Users press the admin button on side panel
Expected Results	User will be granted access to the admin mode and proceed to edit, delete, or add content to the categories
Actual Results	If they are not an admin, they will be prompted with a message "You do not have admin Rights" and to go back. They will navigate to the user panel
Status	Pass

Test Case ID	PEM - Q&A
Purpose	Allows users to ask questions that can be answered by other users.
Preconditions	Users need to be able to log in with their own credentials to their account into the app.
Expected Results	Users are able to post questions that other users can answer.
Actual Results	There is no way to access Q&A
Status	Fail

Test Case ID	PEM - Autolink
Purpose	Allow App Admins to save URLs in the content of the subcategories as plain text and the app will automatically convert them to hyperlinks.
Preconditions	Users need to be able to log in with their own credentials to their account into the app. Some test URLs need to be inter in the content
Expected Results	Users are able to navigate to external sites from within the app by clicking on the hyperlink.
Actual Results	User was able to access external content from within the app and come back to the same screen on the app once the user was done reviewing the external site.
Status	Pass

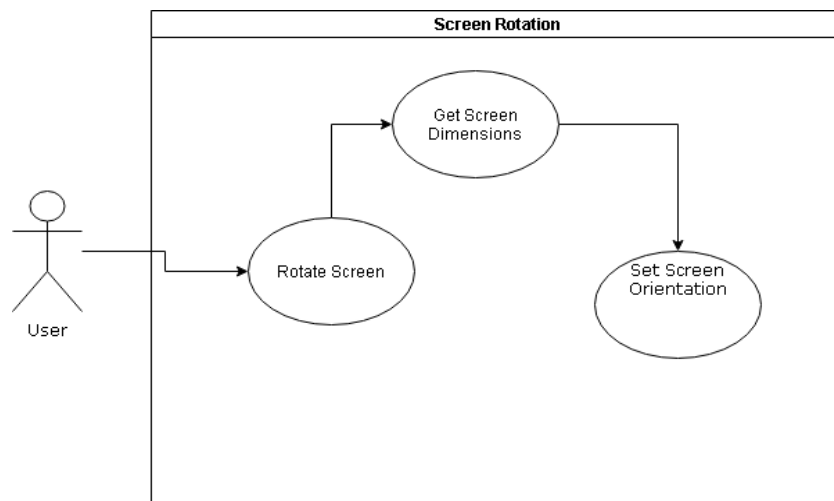
Test Case ID	(WEB) PEM- Login API Test
Purpose	Application should be able to get a token from the backend.
Preconditions	Users need to have internet access.
Expected Results	Users are expected to receive a token from Google API.
Actual Results	Users received a token from API.
Status	Pass

GLOSSARY

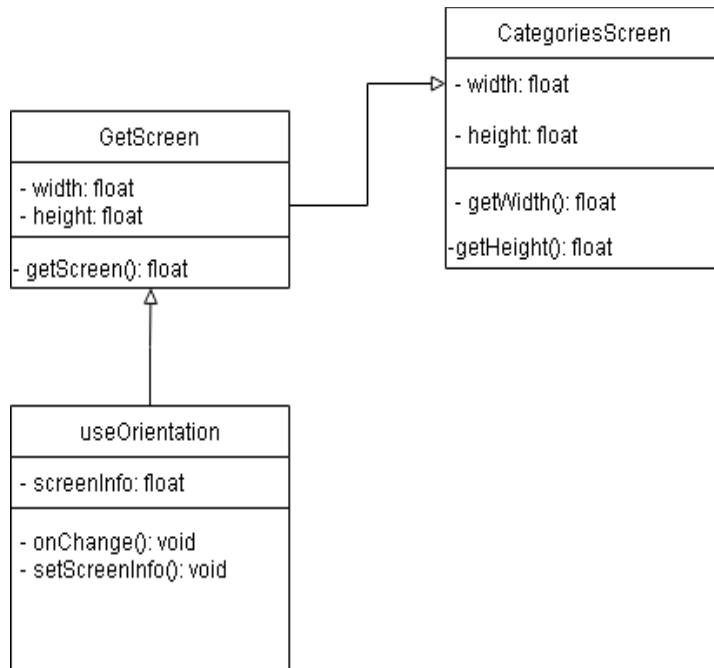
APPENDIX

Appendix A - UML Diagrams

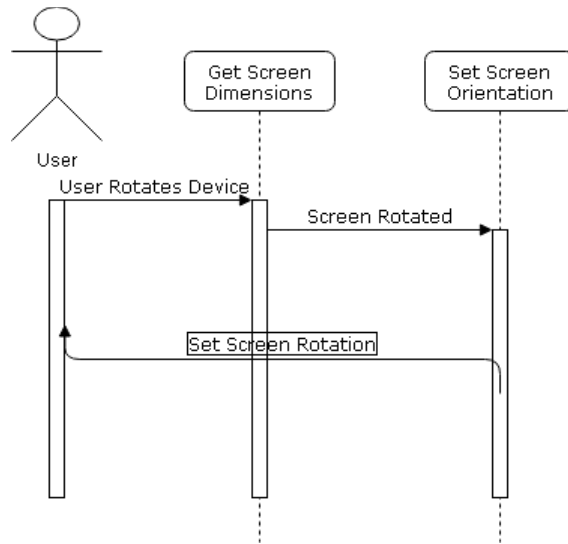
Use Case Diagram (Eduardo Dally)



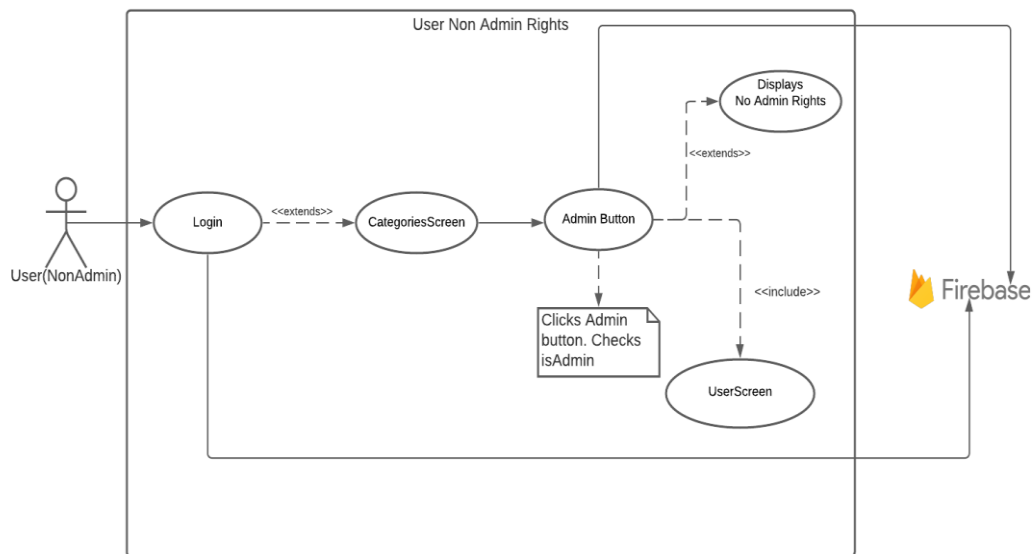
Class Diagram (Eduardo Dally)



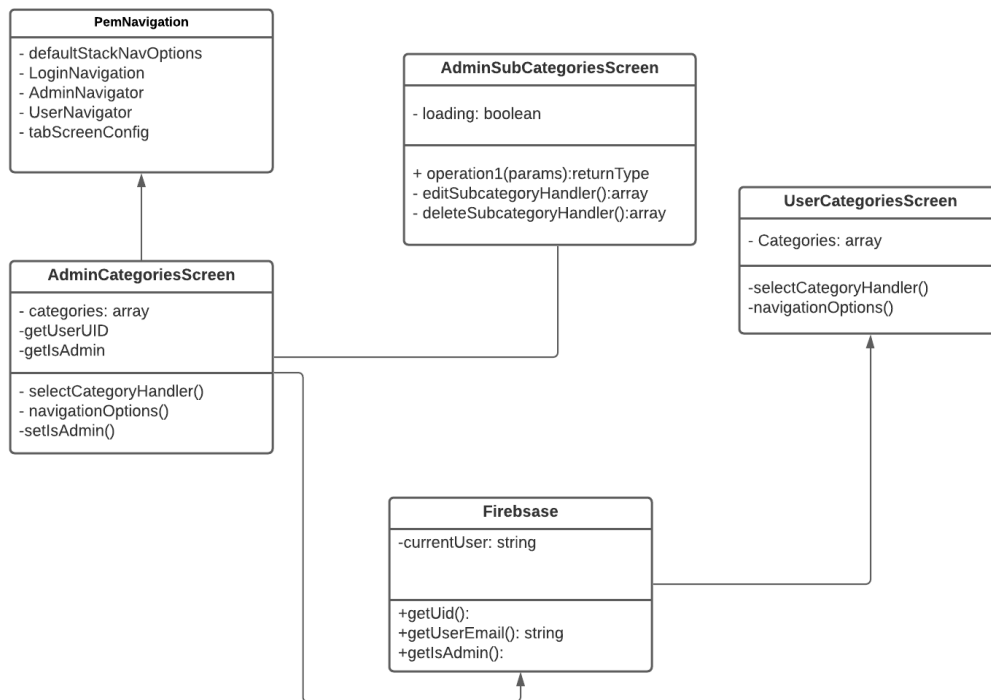
Sequence Diagram (Eduardo Dally)



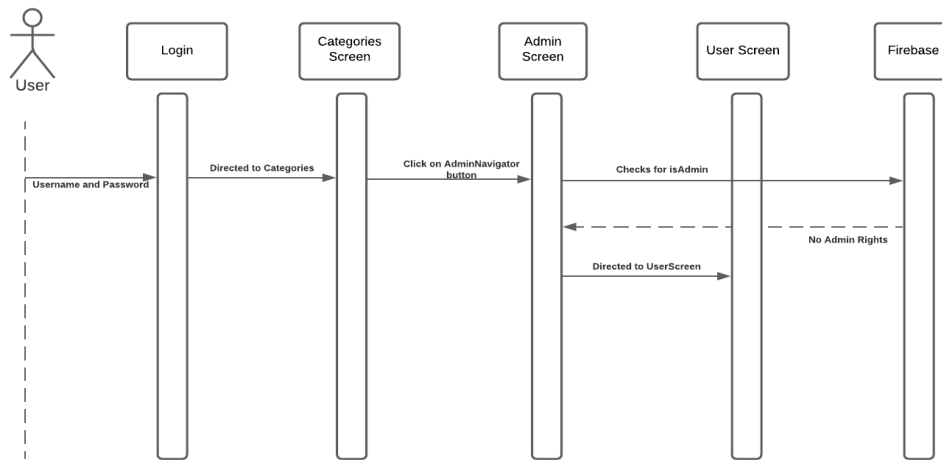
Use Case Diagram (Athena Martinez)



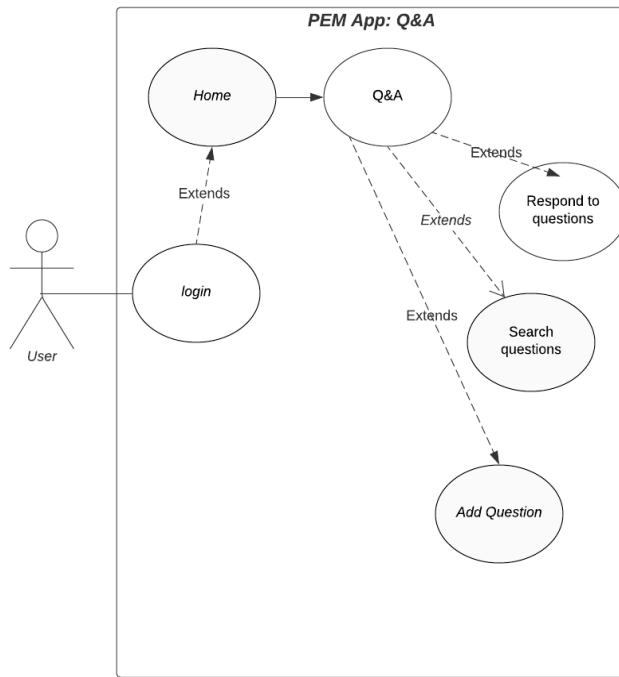
Class Diagram (Athena Martinez)



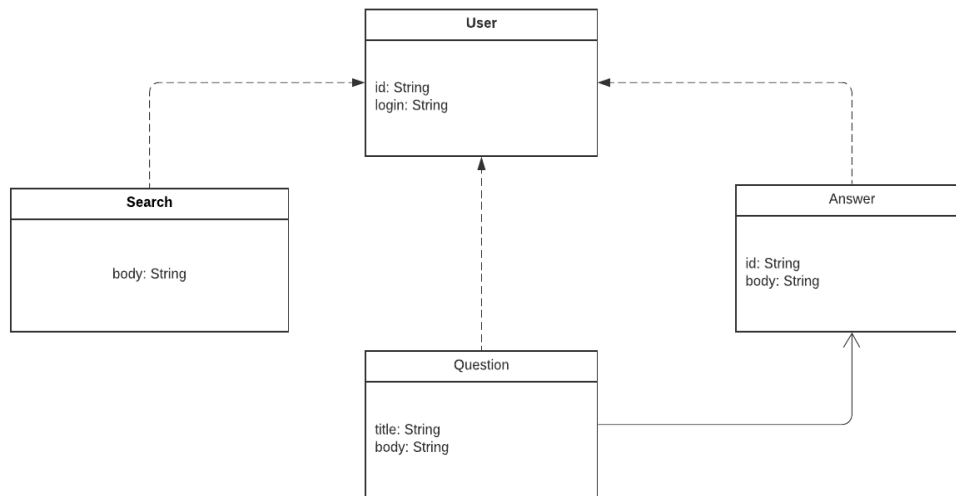
Sequence Diagram (Athena Martinez)



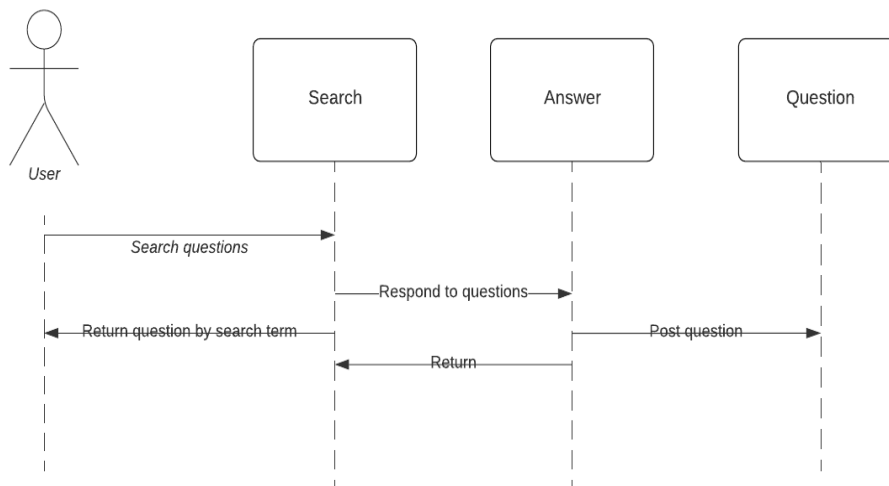
Use Case Diagram (Peter Duarte)



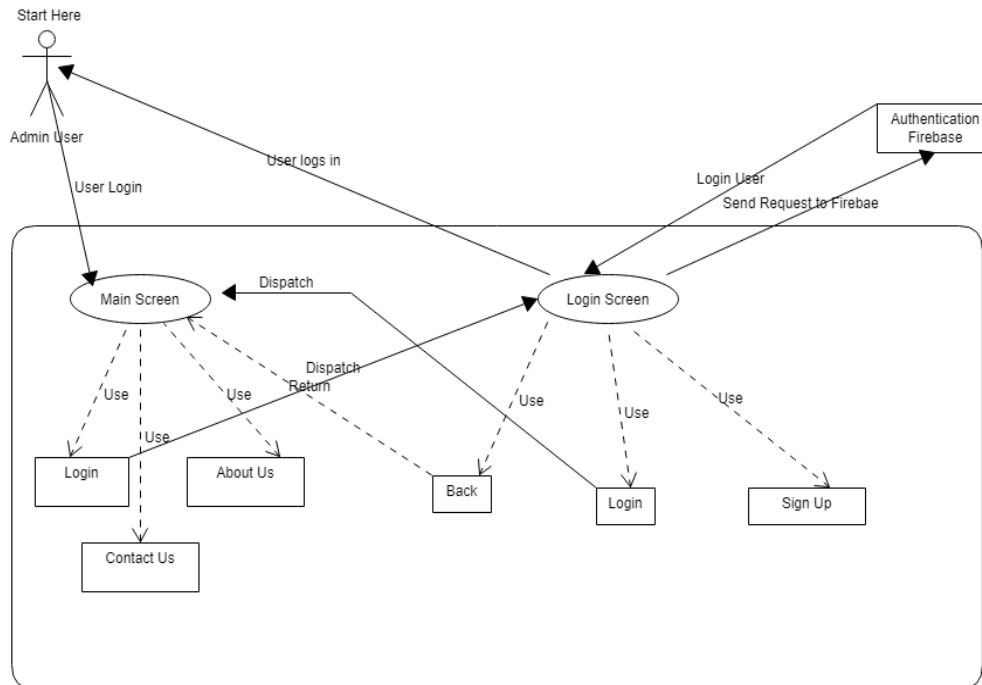
Class Diagram (Peter Duarte)



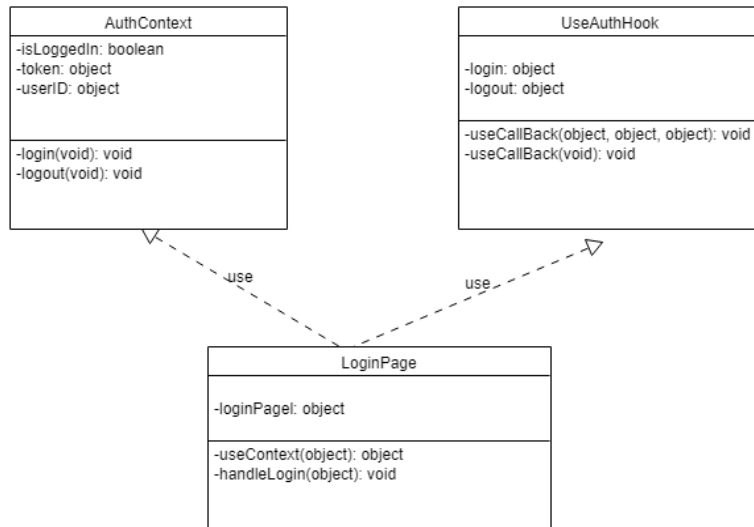
Sequence Diagram (Peter Duarte)



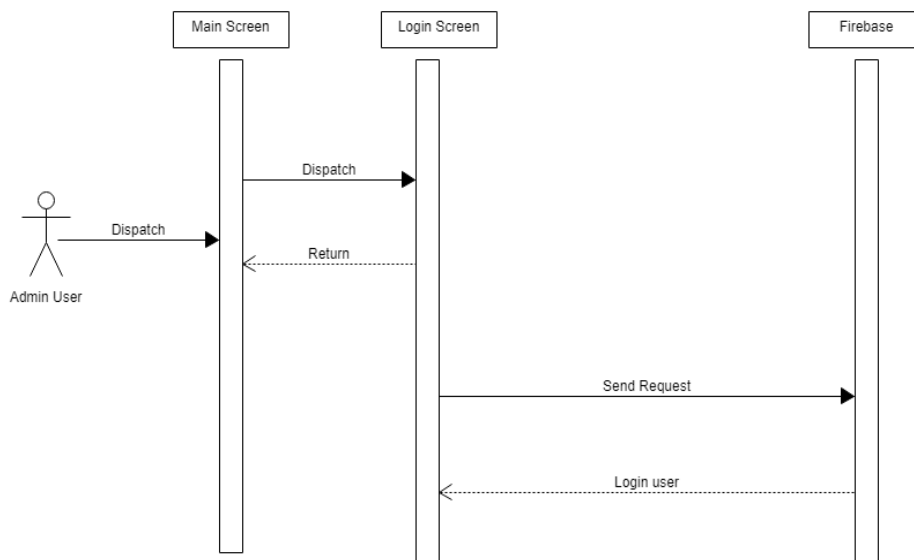
Use Case Diagram (Web) (Kelvin Gonzalez)



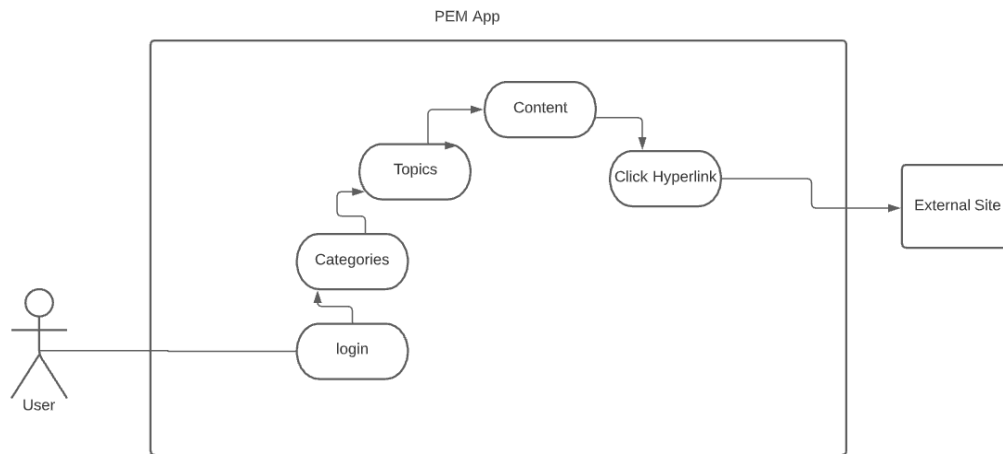
Class Diagram (Web) (Kelvin Gonzalez)



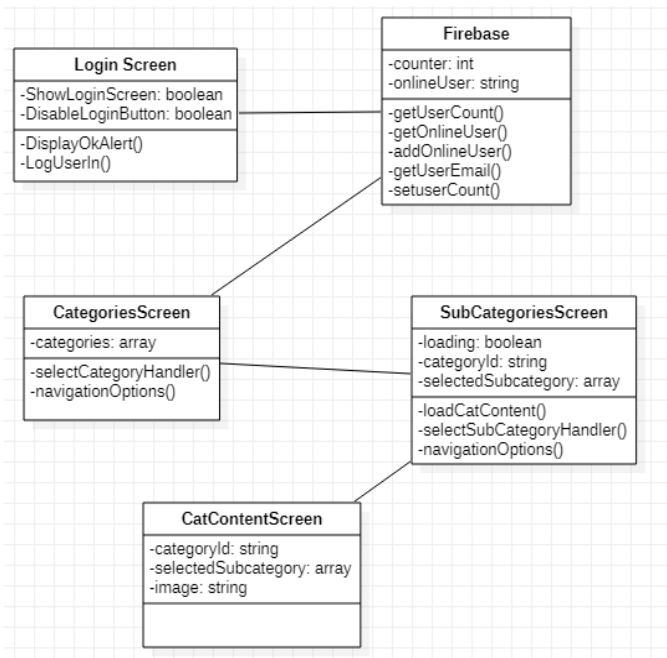
Sequence Diagram (Web) (Kelvin Gonzalez)



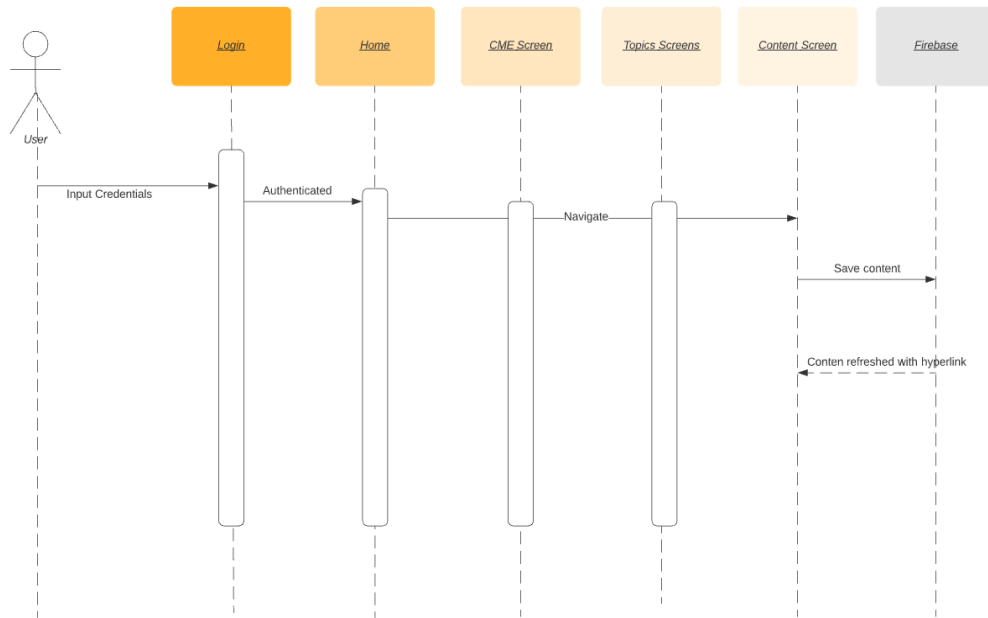
Use Case Diagram (Anabel Perez)



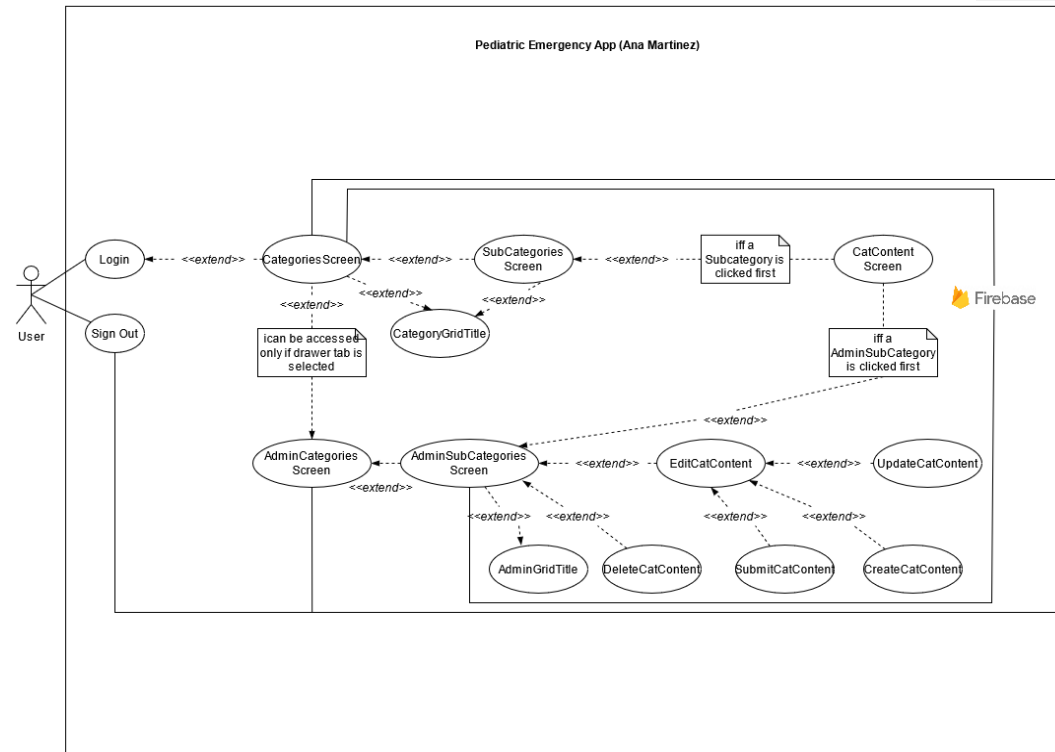
Class Diagram (Anabel Perez)



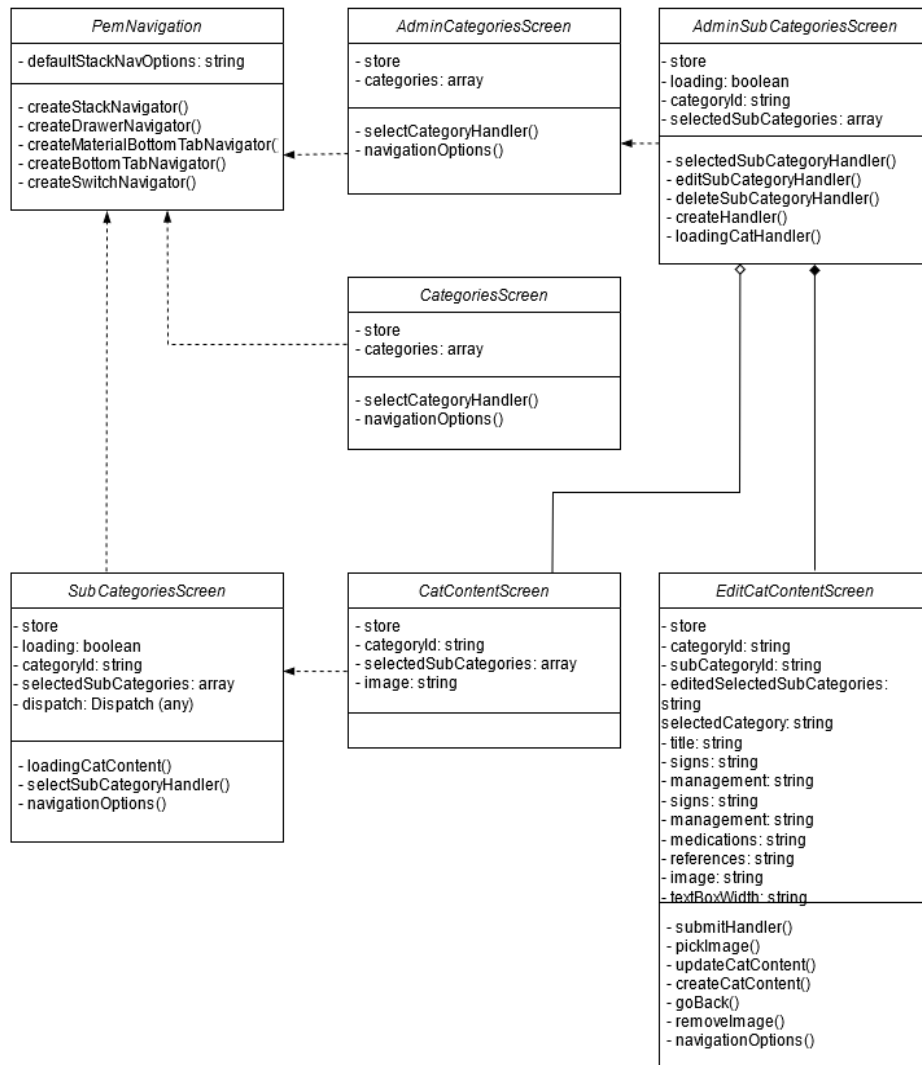
Sequence Diagram (Anabel Perez)



Use Case Diagram (Ana Martinez)



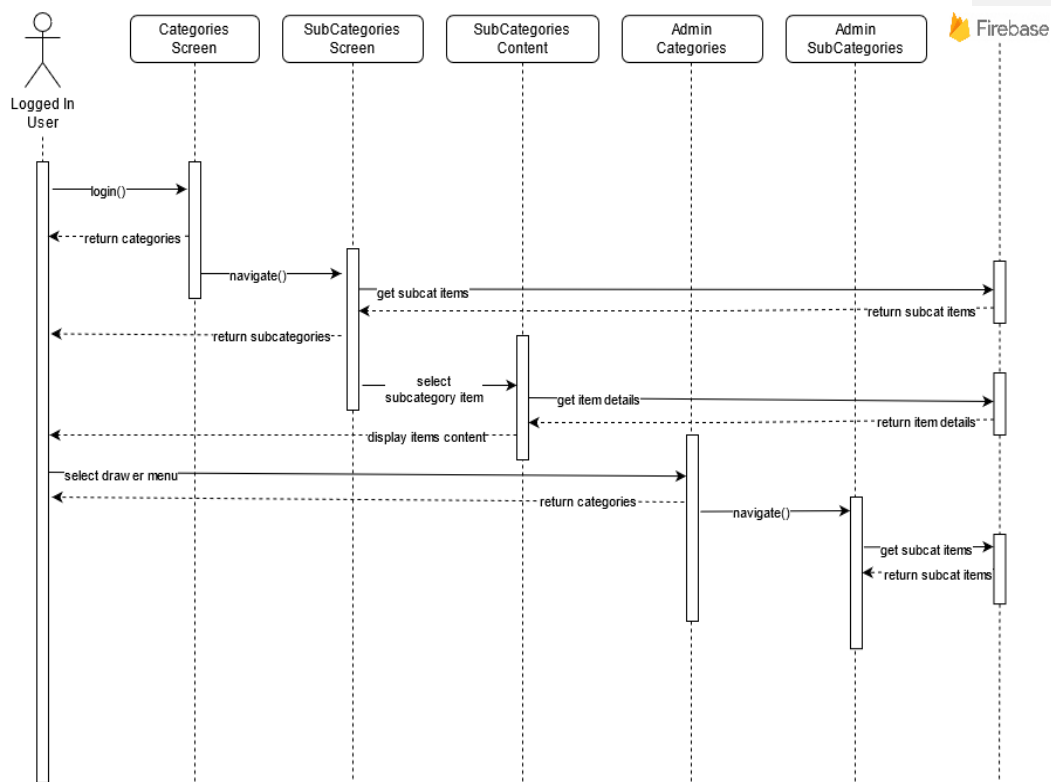
Class Diagram (Ana Martinez)



Final Deliverable

Pediatric Emergency Medicine App

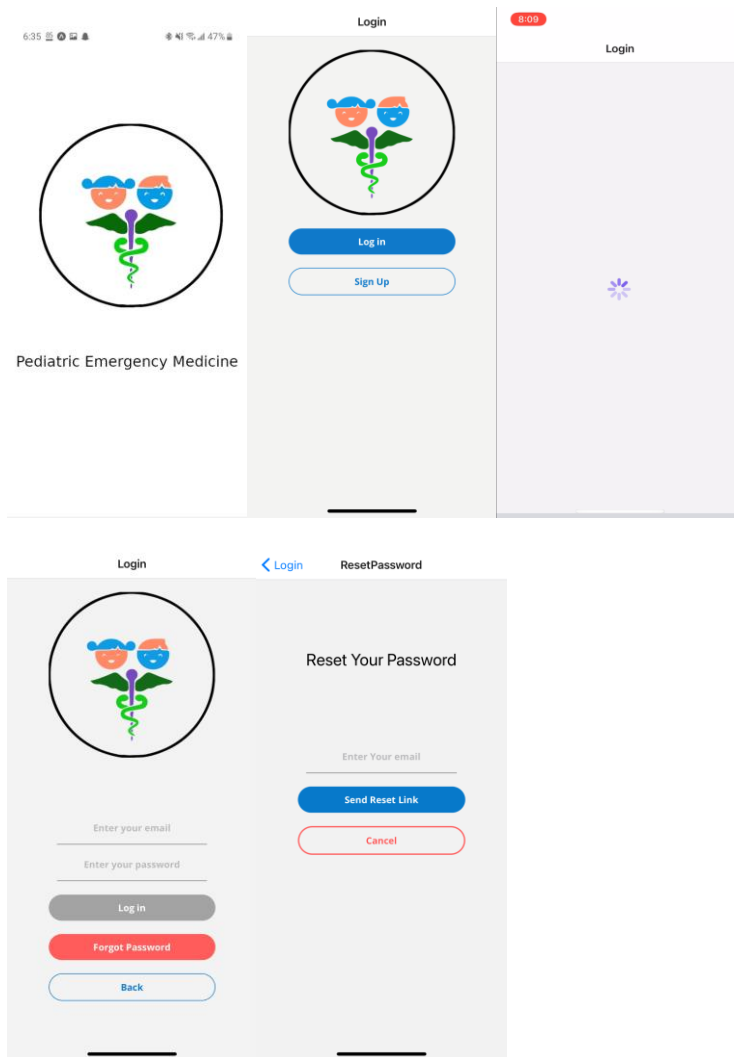
Sequence Diagram (Ana Martinez)



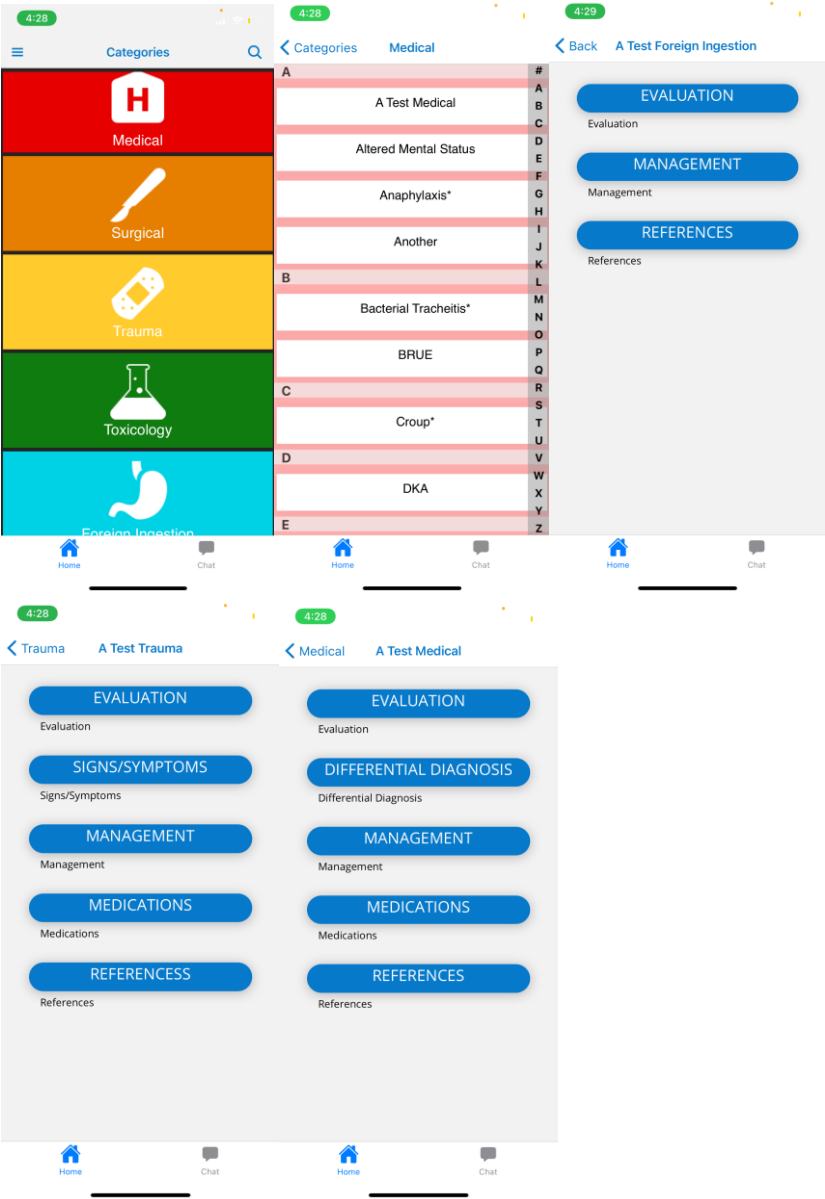
Appendix B - User Interface Design

Application UI

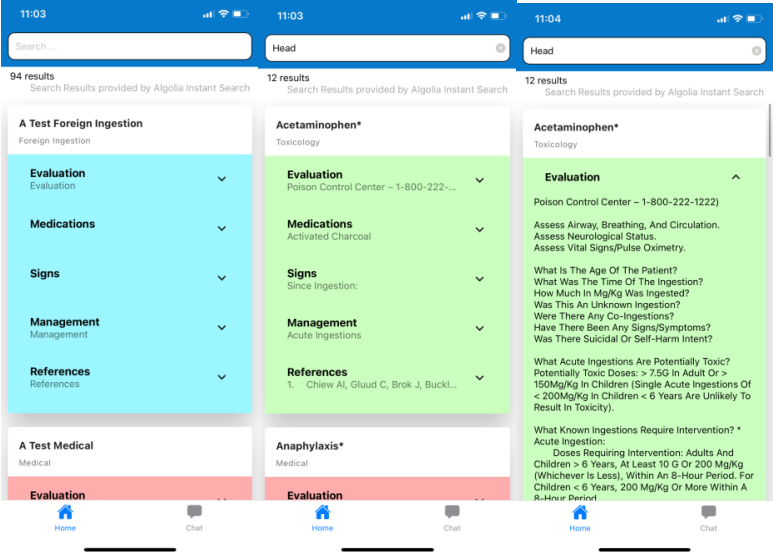
Loading Screen, Login, Sign up, Forgot Password.



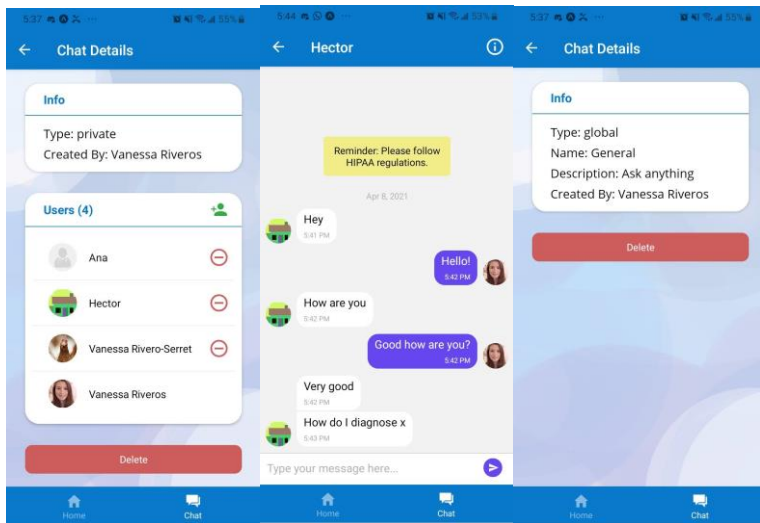
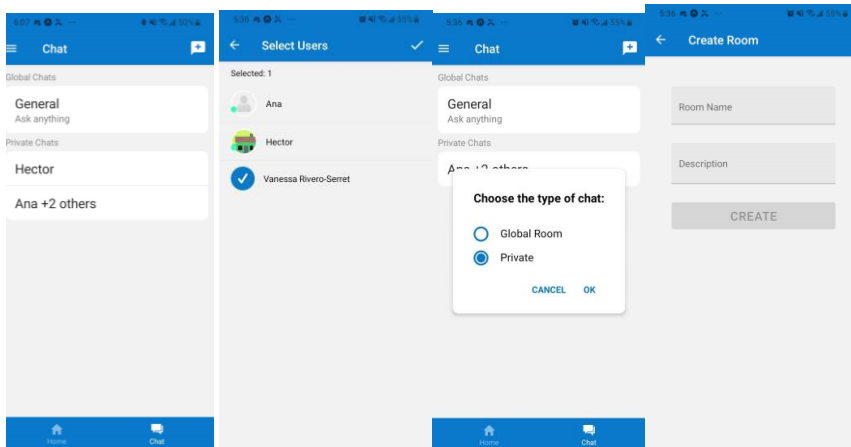
Categories screen, Subcategories with alphabetical scroll, customized condition details.



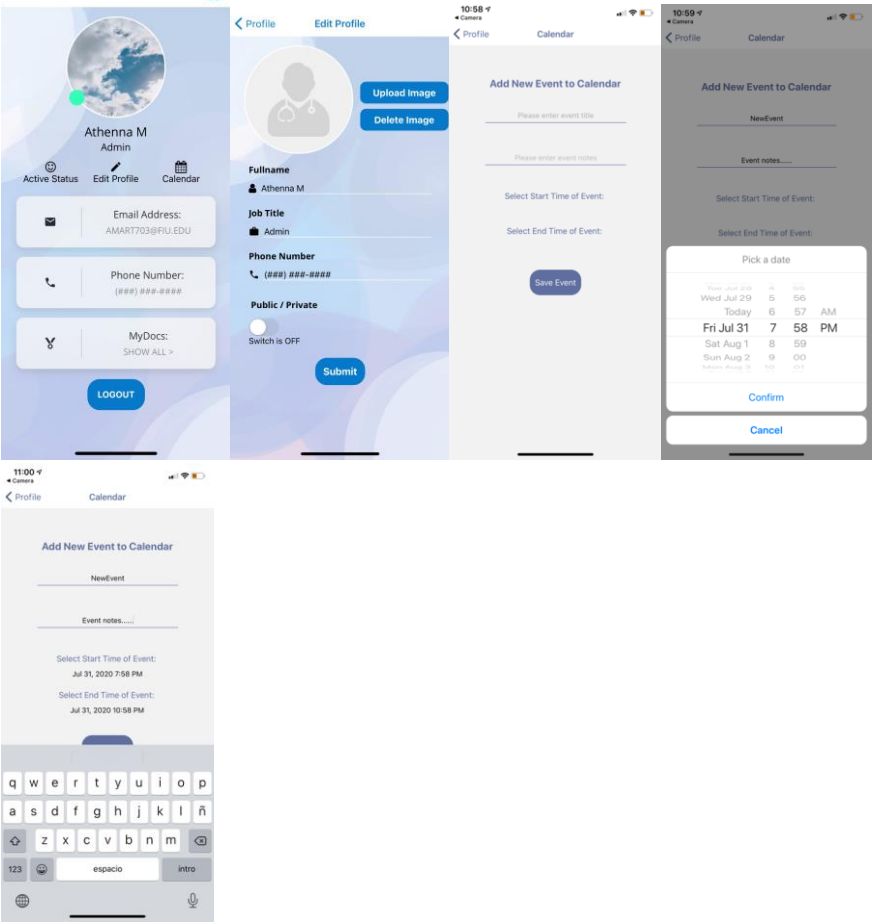
Search, Expanded View, Highlighted match



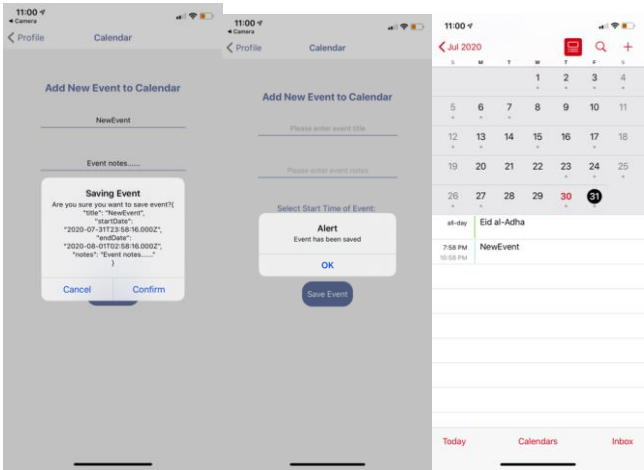
Chat List, Create Global Chat, Create Private Chat, Online Users, Messaging, Chat Details.



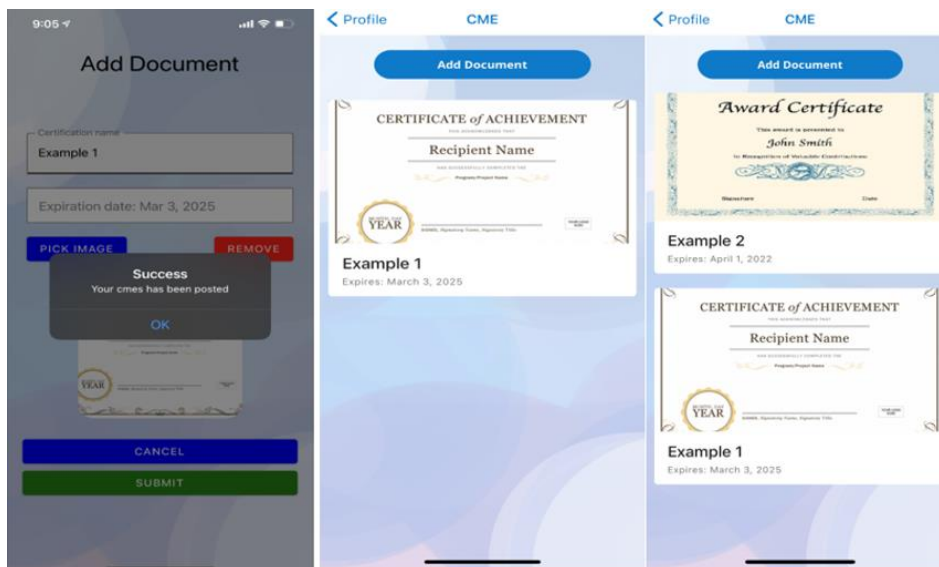
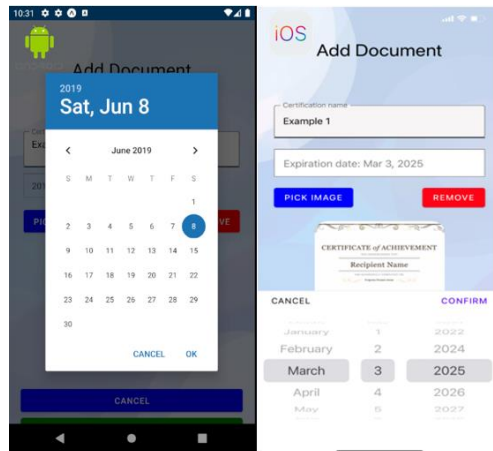
Profile, Edit Profile, Add Calendar, Event



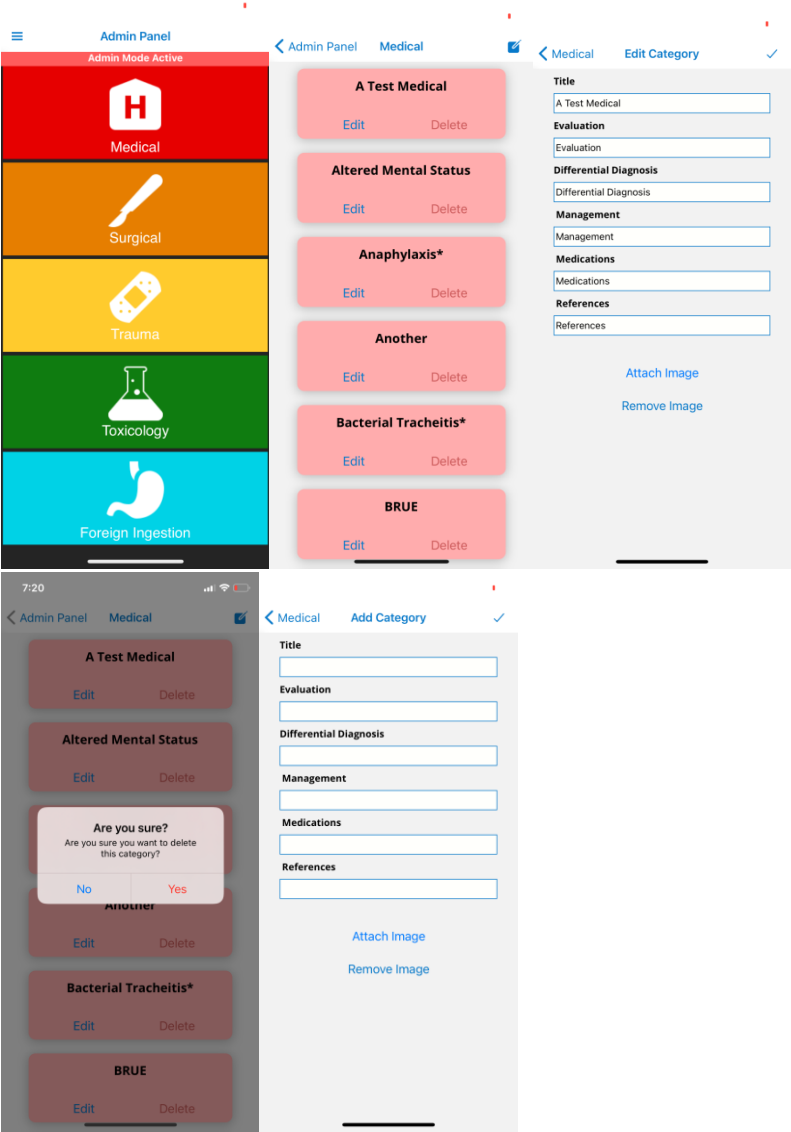
Save Event, Calendar



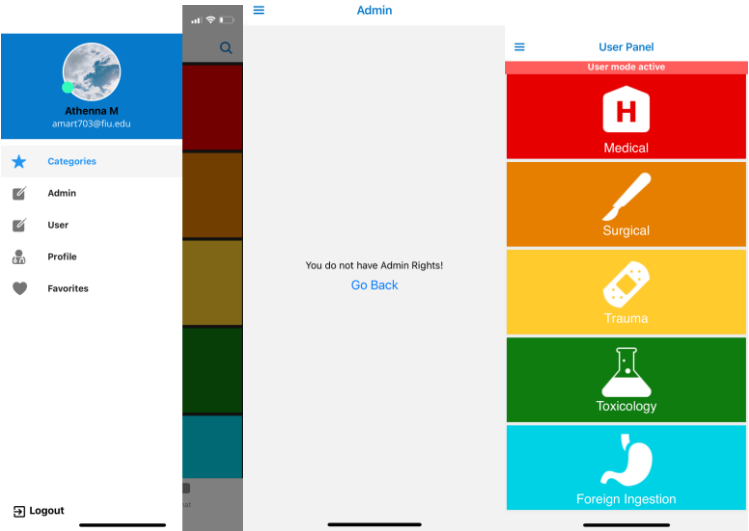
Android DatePicker, iOS DatePicker, Add document, List sorted certifications.



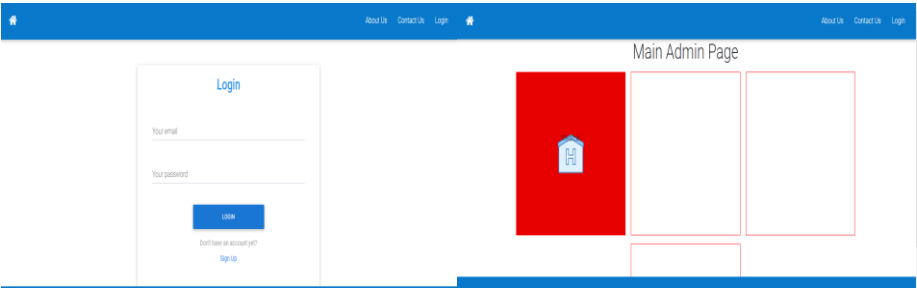
Admin Panel, List Subcategories, Edit, Delete, Add



User Panel, No Admin rights



Web Version



Appendix C - Sprint Review Reports

Sprint 1

Sprint Review Meeting Minutes

Attendees: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Danelis Insua, Felipe Arce.

Start time: 12:00 pm

End time: 12:30 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application security.
- **User Story #3:** As a developer, I want to grant users different permissions depending on their role.
- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens
- **User Story #13:** As a user, I would like to search for answers to Questions already answered (Q&A Section)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- **User Story #5:** As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)
- **User Story #7:** As a developer, I need to connect the front end of the Admin Portal (web) to Firebase
- **User Story #10:** As a developer, I need to create the ReadMe for the React app
- **User Story #12:** As a user, I want the changes made from the Admin Portal (web) to persist on the Admin Panel (mobile)

Sprint 2

Sprint Review Meeting Minutes

Attendees: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Danelis Insua, Felipe Arce.

Start time: 12:00 pm

End time: 12:30 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners:

- **User Story #5:** As a user I would like to remove the black gap in the categories screen.
- **User Story #16:** As a user I would like to remove the image icon from the treatments

- **User Story #19:** As a user I would like new treatments to be sorted alphabetically as I add them.
- **User Story #20:** As a user I would like to know why I'm getting the expiration warning in order to avoid interruption to the test process.

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- **User Story #1:** As a developer, I need to create GitHub workflow (Branches/Code Owner/Branch Protection) to improve the development process
- **User Story #2:** As a developer, I need to create user roles and separate them into groups to improve application security and grant users different permissions depending on their role.
- **User Story #3:** As a user I need to be able to have custom subheading for each category.
- **User Story #4:** As a user I would like something similar to Testflight for Android in order to test the app.
- **User Story #7:** As a developer I need to log in to the admin portal
- **User Story #9:** As a user, I need to see the PEM logo on the Main and Home screens
- **User Story #13:** As a user, I would like to have a Q&A section where I can post questions and also search for previously answered questions.

Sprint 3

Sprint Review Meeting Minutes

Attendees: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Danelis Insua, Felipe Arce.

Start time: 12:00 pm

End time: 12:30 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- **User Story #4:** As a user I would like to find an alternative to Testflight for Android in order to test the app.
- **User Story #16:** As a user I would like to remove the image icon from the treatments
- **User Story #19:** As a user I would like new treatments to be sorted alphabetically as I add them.
- **User Story #21:** As a user I would like the different topics to be sorted alphabetically to make the search easier.
- **User Story #20:** As a user I would like to know why I'm getting the expiration warning in order to avoid interruption to the test process.
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- **User Story #5:** As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)
- **User Story #7:** As a developer, I need to connect the front end of the Admin Portal (web) to Firebase
- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links.
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #24:** As a developer, I would like to research and be able to deploy the app on Google Play for testing.

Sprint 4

Sprint Review Meeting Minutes

Attendees: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Danelis Insua, Felipe Arce.

Start time: 12:00 pm

End time: 12:30 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- **User Story #4:** As a user I would like to find an alternative to Testflight for Android in order to test the app.
- **User Story #16:** As a user I would like to remove the image icon from the treatments
- **User Story #19:** As a user I would like new treatments to be sorted alphabetically as I add them.
- **User Story #21:** As a user I would like the different topics to be sorted alphabetically to make the search easier.
- **User Story #20:** As a user I would like to know why I'm getting the expiration warning in order to avoid interruption to the test process.
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content
- **User Story #26:** As a user I would like to have the treatments with the first letter of each word capitalized
- **User Story #27:** As user I would like to upload an image in the certifications screen and save it to the DB (currently not saving)
- **User Story #28:** As a user I need to upload images in subcategories without cropping them.
- **User Story #29:** As a user I would like to work on the app in horizontal orientation. (screen orientation vertical and horizontal)

The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- **User Story #5:** As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)
- **User Story #7:** As a developer, I need to connect the front end of the Admin Portal (web) to Firebase
- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links.
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #24:** As a developer, I would like to research and be able to deploy the app on Google Play for testing.

Sprint 5

Sprint Review Meeting Minutes

Attendees: Ana Martinez, Athenna Martinez, Peter Duarte, Anabel Perez, Kelvin Gonzalez, Eduardo Dally, Bryant Rojas, Danelis Insua, Felipe Arce.

Start time: 12:00 pm

End time: 12:30 pm

After a show and tell presentation, the implementation of the following user stories were accepted by the product owners: All.

- **User Story #4:** As a user I would like to find an alternative to Testflight for Android in order to test the app.
- **User Story #16:** As a user I would like to remove the image icon from the treatments
- **User Story #19:** As a user I would like new treatments to be sorted alphabetically as I add them.
- **User Story #21:** As a user I would like the different topics to be sorted alphabetically to make the search easier.
- **User Story #20:** As a user I would like to know why I'm getting the expiration warning in order to avoid interruption to the test process.
- **User Story #25:** As a user, I would like to be able to add links to topics and be redirected to the content
- **User Story #26:** As a user I would like to have the treatments with the first letter of each word capitalized
- **User Story #27:** As user I would like to upload an image in the certifications screen and save it to the DB (currently not saving)
- **User Story #28:** As a user I need to upload images in subcategories without cropping them.
- **User Story #29:** As a user I would like to work on the app in horizontal orientation. (screen orientation vertical and horizontal)

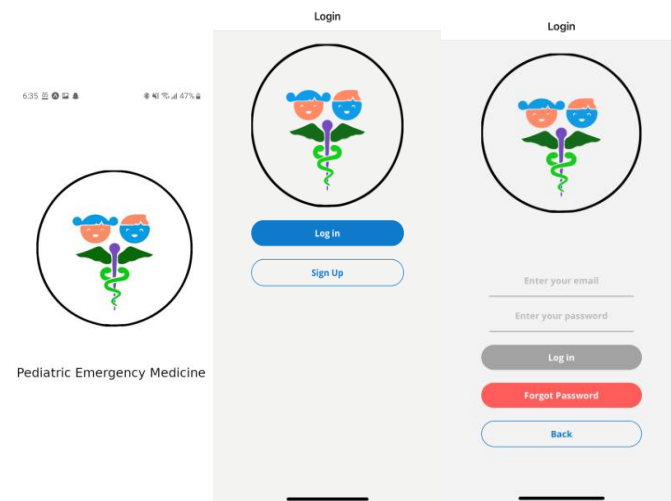
The following ones were rejected and moved back to the product backlog to be assigned to a future sprint at a future Sprint Planning meeting.

- **User Story #5:** As a user, I would like to access an Admin Portal (web) from my desktop that has the same capabilities that I have in the Admin Panel (mobile)
- **User Story #7:** As a developer, I need to connect the front end of the Admin Portal (web) to Firebase
- **User Story #11:** As a user, I would like to have an "Edit" button to edit certifications cards on the CME screen
- **User Story #17:** As a user I would like to be able to paste formatted content inside the subcategories without losing formatting like fonts and links.
- **User Story # 15:** As a user, I would like to be able to upload PDF files on the CME screen
- **User Story #24:** As a developer, I would like to research and be able to deploy the app on Google Play for testing.

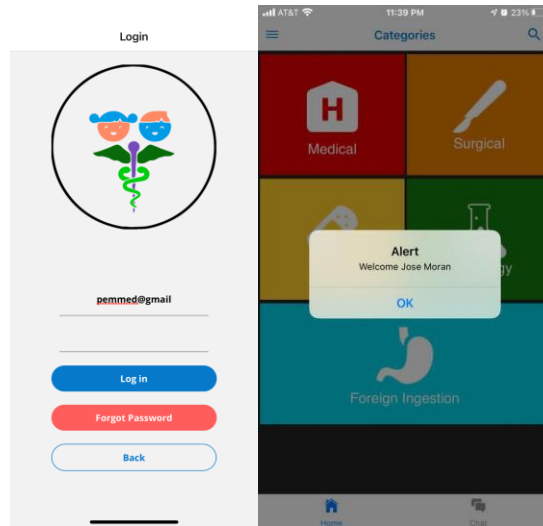
Appendix D - User Manuals, Installation/Maintenance, Shortcomings/Wishlist

User Manuals

Sign Up: 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 “Confirm” 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 reset password by clicking on the “Forgot Password” button. Once the user logs into their account, they will be redirected to the category screen.



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 belong 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 them to the CatContentScreen.

< Categories	Medical	< Categories	Surgical	< Categories	Trauma
A	#	A	#	A	#
A Test Medical	A	A Test Surgical	A	A Test Trauma	A
Altered Mental Status	B	Acute Cholecystitis	B	Blunt Abdominal	B
Anaphylaxis*	C	Appendicitis*	C	BLUNT Thoracic	C
Another	D	Globe Rupture	D	Cardiac Tamponade	D
B	E	Intracranial Bleed	E	GSW Extremity	E
Bacterial Tracheitis*	F	Intussusception	F	Head	F
BRUE	G	Midgut Volvulus	G	MVC	G
C	H	Nasal Septal Hematoma	H	Pedestrian HBC	H
Croup*	I	Necrotizing Enterocolitis	I		I
DKA	J		J		J
Ectopic Pregnancy	K		K		K
	L		L		L
	M		M		M
	N		N		N
	O		O		O
	P		P		P
	Q		Q		Q
	R		R		R
	S		S		S
	T		T		T
	U		U		U
	V		V		V
	W		W		W
	X		X		X
	Y		Y		Y
	Z		Z		Z

Home

Chat

Home

Chat

Home

Chat

< Categories

Toxicology

A	#
A Test Toxicology	A
Acetaminophen*	B
Alcohols	C
B	D
Bath Salts	E
Benzodiazepines	F
Beta Blockers*	G
C	H
Carbon Monoxide*	I
Clonidine*	J
Cocaine	K
Cyanide	L
	M
	N
	O
	P
	Q
	R
	S
	T
	U
	V
	W
	X
	Y
	Z

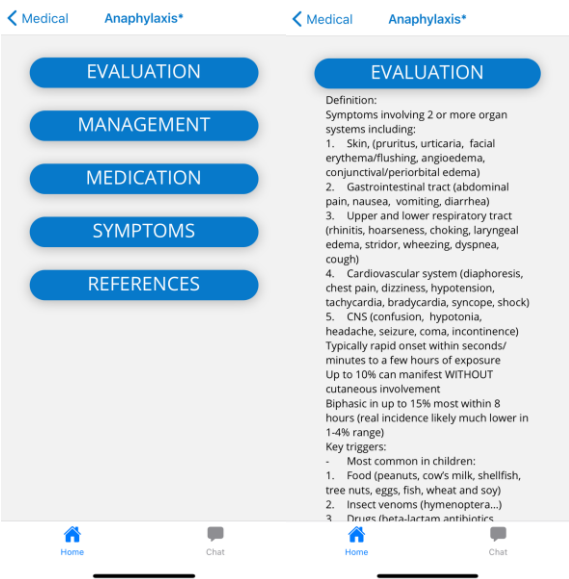
< Categories Foreign Ingestion

A	#
A Test Foreign Ingestion	A
Another Test	B
B	C
Batteries	D
C	E
Coins	F
M	G
Magnets	H
N	I
Needles	J
T	K
Test	L
	M
	N
	O
	P
	Q
	R
	S
	T
	U
	V
	W
	X
	Y
	Z

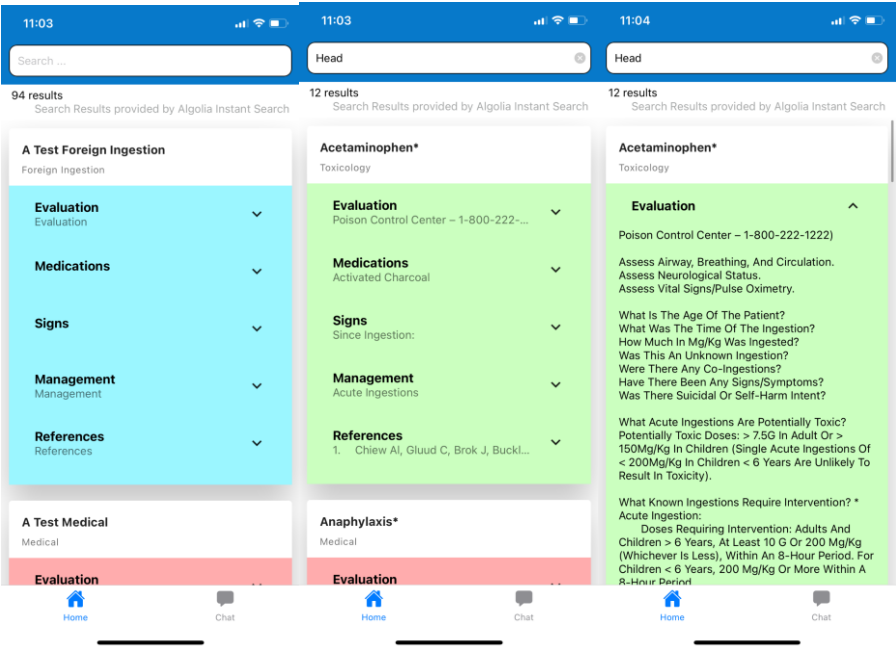
Home Chat

Home Chat

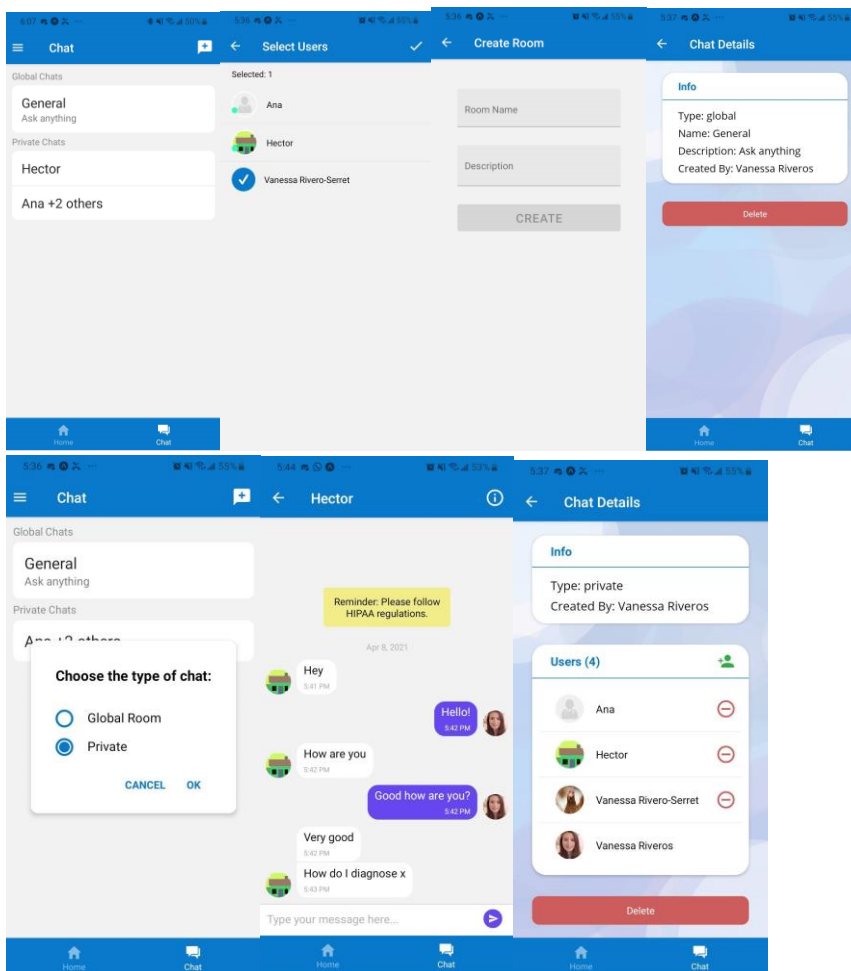
CatContentScreen: This screen displays the subcategories’ information including title, evaluation, medication, management, symptoms, references and images. Every section can be clicked to expand on information.



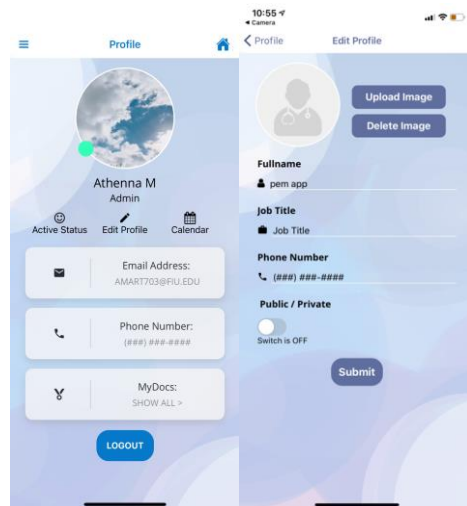
Search: When the user enters the search category, it will open up a screen with a search bar in it. After entering a search term, the application will display the subcategories which contain that specific term. These will be direct links to those specific subcategories. If the term the user enters is not part of any subcategory 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.



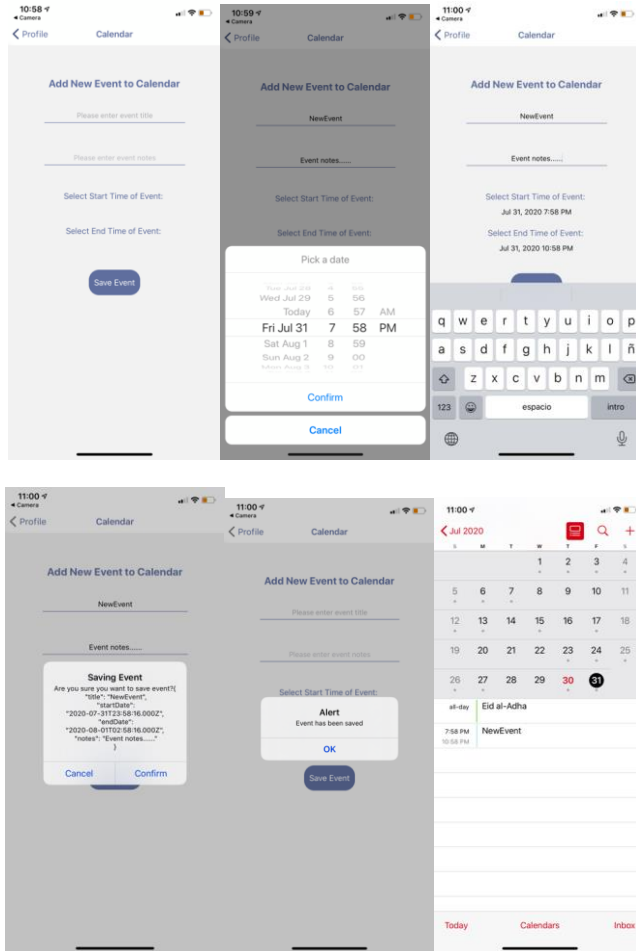
Chatroom: Whenever a user clicks on the chat button at the bottom of the navigation bar, a list of global and private chats will be displayed. In order to create a new chat, the user can click on the ‘Add’ icon at the top right of the screen. This will then prompt the user to choose if they want to create a global or private chat. A global chat is centered around a topic and can be accessed by all users. A private chat is a chat between two or more online users. Once the chat has been created, the user will be redirected to the chat screen. At the top right of each chat window, there is an “Info” icon. If the user clicks this icon, they can see chat details such as type, who it was created by, users currently in the chat, the ability to add and remove users, and a delete button. The ability to add or remove users and delete the chat is only available to the person who created it.



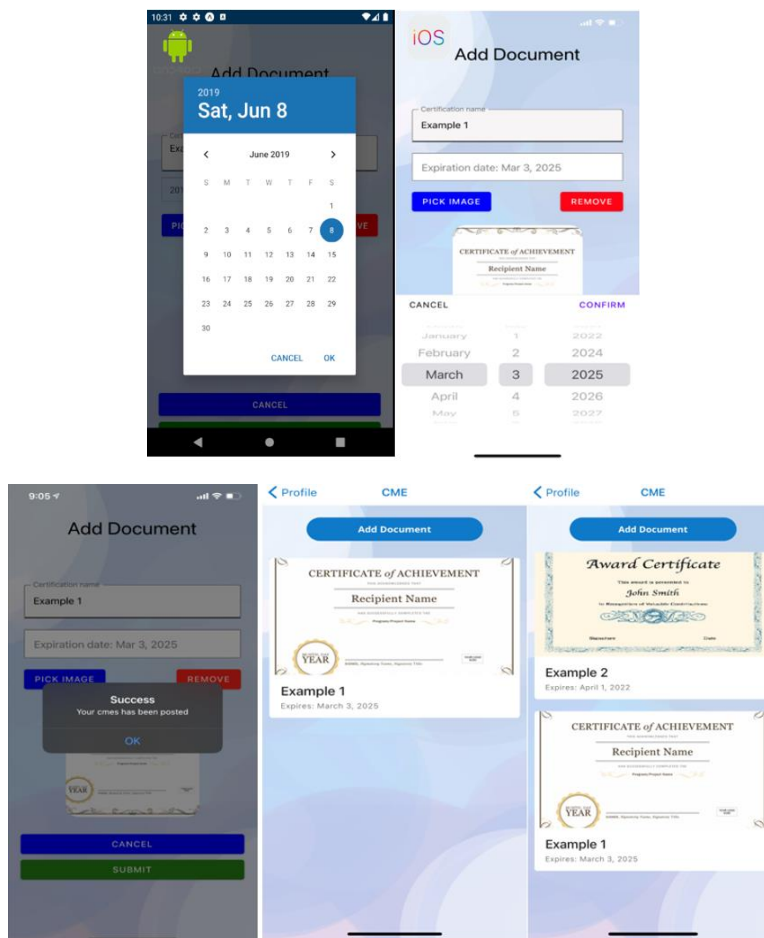
Profile: The profile screen allows users to review personal information like name, job, title, email phone numbers, profile photo, and set the user's active status. From here, you have several options, you can proceed to the Edit Profile Screen, in which users can edit the data being displayed in the profile, you can go to the Calendar Screen, or you can access the CME screen.



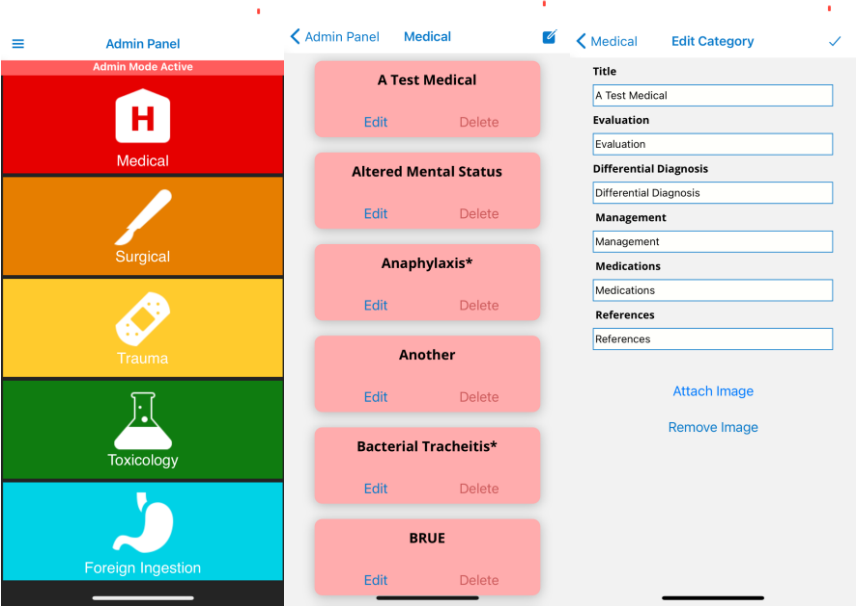
Calendar Screen: In this screen, users can create events with details such as event title, event notes, choose a start date and end date, an alert message will appear displaying event summary when the “save changes” button is pressed, the event will be saved to the user device’s calendar, and an alert message for confirmation that the event was successfully saved will show up.



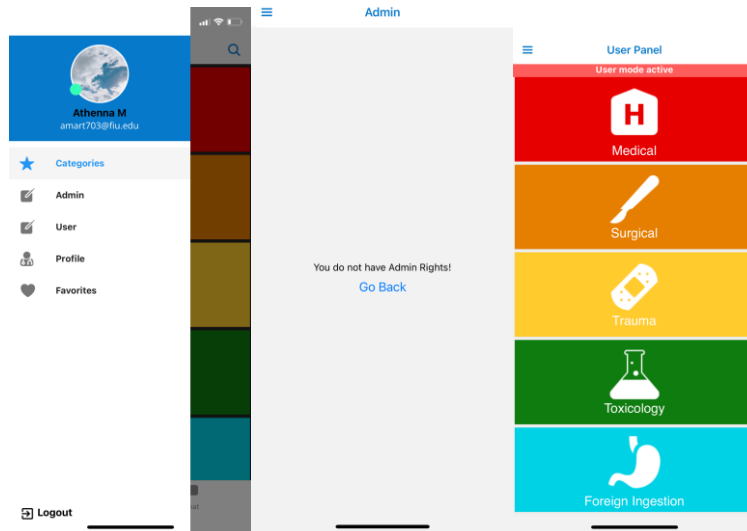
CME: In the CME Tracker, the user can enter the name of a certificate, select expiration date, and choose the image. If the certificate is closer to expiration, it will be displayed at the top of the list. The UI for date selection is different on both platforms but they both have the same functionality.



Admin Panel: The Admin panel is composed 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 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 CatContentScreen to see how the data was updated or created.



User Panel: The User panel appears on the side panel options. If a user does not have admin rights to edit, delete, add content they will be taken to the User Panel where they can see their own added content. A message appears if the user attempts to go to the admin panel without the correct privileges it will ask you to go back.



Installation/Maintenance**Commented [8]:** Just need to add section for website**STEP 1:****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. Alternatively, instead of using an emulator, you can also download the Expo App to run the app directly on your own device, Android or IOS.
7. 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`.

STEP 2:**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/Pem-Med/pem-app.git` in the same directory where you ran Step 2.

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: penmedapp@gmail.com
Password: (found in credentials folder, ask your teammates where to find it)
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. You will need to create a `firebaseConfig.js` and `algoliaConfig.js` files and put the following firebase and algolia config keys as shown below: (this is also found in credentials folder of google drive)

`firebaseConfig.js`

```
const config = {
  apiKey: "AIzaSyB9-PR4nCh8T6nNtqvnMhYFLyxr7ZLXJV8",
  authDomain: "med-app-519aa.firebaseio.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
```

`algoliaConfig.js`

```
const algoliaConfig = {
  appId: "WK4HK1IJPD",
  adminKey: "f10359538512b8ad00ae25e8eea6aa47",
  searchKey: "fb18e71dc7f79b9a9d7277ddadb522f2",
  indexName: "med_Categories",
}

export default algoliaConfig
```

There is no need to delete these files 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.

5. Place this file in the root directory of the project. Ex: `pem-app/firebaseConfig.js`
6. In the credentials folder of google drive, you will also find `algoliaConfig.js`. Place this file in the root directory of the project folder. Ex: `pem-app/algoliaConfig.js`
1. Optionally, press `Alt+Shift+F` to format.
2. Save your changes and in the Firebase window from Step 8, click “Continue to console”.

STEP 4:

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.

STEP 5:**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.
5. If using a physical Android phone, open the expo app and scan the barcode on the terminal.
6. If using an Iphone, open your camera and scan the barcode, this should open the expo app and start up the app.

Web Version Installation/Maintenance**Commented [9]:** Just need to add section for website**STEP 1:****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/>

STEP 2:**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.
3. Run git clone <https://github.com/PemMed-Web/pem-admin> in the same directory where you ran
Step 3

STEP 3: Please, see STEP 3 above (Configuring Firebase Part 1).

STEP 4:

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.

Shortcomings/Wishlist

Have the capability to upload PDF files in the CME screen feature.

Full implementation of the user roles

Web version to connect to Firebase and Agolia

The favorite button on panel will allow users to add their favorite content

Add PEM logo on the main home screens.

Have a fully functional Q&A section where questions and answers are saved and could be searched on

Change the name of the app to PEM-NOW

Add functionality to add a user button in the chat details screen.

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/>

Commented [10]: if you guys have any more please add them here