

Florida International University Knight Foundation

School of Computing and Information Sciences

Computer Science

Final Deliverable

Project Title: Web Based Neurological Assessment

Team Members: Nicolas Bello, Jordan Bravo, Jonathan Oviedo

Mentor: Dr. Christian Poellabauer

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 Web Based Neurological Assessment project (NeuroTask as placeholder name) and the work accomplished by the development team this term.

NeuroTask is a web application that provides a means by which patients and medical providers can connect, communicate and be examined/give examinations through virtual means.

The web application will allow for one-on-one video calls, ability to schedule appointments, view medical history and records, perform examinations given by providers (live on video call or on their own time) and view the history of these results on their portal.

This is brand new software realized during the semester of Fall 2023 and therefore is in a very novice and incomplete state. It will require great work and cooperation from future teams to realize the visions that this web application has and the continued support of the hard work that teams will put forth.

In its current state, the web application allows for the creation of both patient and provider accounts, requesting password resets and resetting the password of a user, a minimal log in experience with a WIP dashboard, contacts page with ability to request a provider as contact (only works for patient end users currently), a skeleton of the calendar, side navigation and user drawer that allows logging out.

Introduction	7
Current System	7
Purpose of New System	7
User Stories	8
Implemented User Stories	8
Project Plan	10
Hardware and Software Resources	10
Sprints (Planning Sessions)	11
Sprint 1	11
Sprint 2	12
Sprint 3	13
Sprint 4	15
Sprint 5	16
Sprint 6	17
Sprint 7	18
System Design	19
Deployment Diagram	19
Architectural Patterns	19
System Validation	20
Appendix A – UML Diagrams	21
Use Case Diagrams	21
Sequence Diagrams	22
Appendix B – User Interface Design	27
Appendix C – Sprint Review Reports	30
Sprint 1 Review	30
Sprint 2 Review	31
Sprint 3 Review	32
Sprint 4 Review	33
Sprint 5 Review	34
Sprint 6 Review	35
Sprint 7 Review	36

Appendix D – User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist	
Document and other documents.....	37
Installation Manuel.....	37
Accessing the Repo	37
Accessing Firebase Production.....	37
Installing Dependencies and Packages	38
Project Dependencies	38
Running Project for First Time.....	38
Shortcomings/Wishlist	38

Introduction

Patients with neurological disorders require a way to communicate with their medical providers through virtual means, with the same set of examinations and feedback that an in-person consultation can provide. This includes tests that a provider can assign to a patient, one-on-one video calls with live feedback on those tests, and a robust history concerning previous calls and examinations between a patient and provider. This is the goal of this web application, to provide that bridge between a patient and a provider through a virtual network. The system is brand new and built from the ground up this semester. The primary focus was to instill a solid, concrete architectural basis for teams to build upon as this project continues to mature through various semesters. This included setting up the guidelines for testing, component creation, styling, back-end and front-end logic, and directory structure. Since the development was initiated in the semester of Fall 2023, the project is still in a very nubile stage and requires way more work to be able to arrive at the point where some of the visions are being met.

Current System

When we started working on the system this semester, the system was being developed with Flutter SDK as a mobile application. The project was being programmed in Swift language and used Firestore database to store user data. We wanted to make a switch from mobile to a web application since much of the project was still in its infancy. Doing so would allow us to reach a wider audience and would also let us and future iterations scale the project, if need be, since Flutter has many limitations therefore limiting our creativity and productivity.

Purpose of New System

We decided to start the project anew using Angular Framework to develop our front-end and Firebase SDK for our back-end as well as using Angular services that communicate with our back-end. Our focus was to set up the infrastructure of the web application to allow future semesters and teams to focus on more specific and intensive features. We set up the basis of the production environment and Firebase emulator for mock data (fake documents within the Firebase database) when testing out features and components. This will allow future teams to test components and features such as chat systems between user providers and patients. A simple

user login and register page has been developed and is stored within the Firestore database. A homepage and sidebar navigation have been implemented to allow users to easily navigate through the website. So far, the only pages that have been built are the calendar, homepage/dashboard, user login/registration (with respect to the different user types i.e. medical provider and patient), and contact page. These pages are still in their experimental phase and are subject to change to how future iterations and teams see fit. We currently have a rough basis for teams to build upon and expand from existing features.

User Stories

In this inaugural iteration of NeuroTask (placeholder namer), our primary focus was establishing the bedrock for subsequent phases. Our implemented user stories revolved around pivotal objectives: configuring backend services facilitating seamless communication between the web app and Firebase for robust user data storage, crafting an intuitive UI/UX through frontend design, and other quintessential components defining a comprehensive web application experience.

Implemented User Stories

User Story 1: Research Google Flutter for Development and Scalability and Cross-Developing to Web Based Platforms from Mobile

User Story 2: Research Architectural Design and Infrastructure

User Story 3: Research Integration Possibilities with Python and other Technologies

User Story 4: UI/UX of Application

User Story 5: Implement Landing Page

User Story 6: Set up NgRx, Documentation, Guidelines and Initial Commit of Project

User Story 7: Setting up Firebase + Introducing Fake Data

User Story 8: Implement Login Page

User Story 9: Implement Routes Depending on User being Patient or Provider

User Story 10: Implement Dashboard

User Story 11: Implement User Sign Up Option Page

User Story 12: Implement Registration Form

User Story 13: Implement User Drawer

User Story 14: Redesign Firestore Database to Accommodate for Facilities, Admins, Providers and Patients

User Story 15: Implement Email Verification for New Accounts

User Story 16: Implement Backend Logic for New Provider Creation and Handling.

User Story 17: Create Tests for Existing Functionality.

User Story 18: Dashboard Responsiveness and Route Guarding.

User Story 19: MVP of Contacts Page for a User

User Story 20: Password Forgot Feature

User Story 21: Login Via Gmail

User Story 22: Route Guards for Route Protection

User Story 23: Finalize the Side Nav for MVP

User Story 24: Implement Landing Page for Contacts Route

User Story 25: Implement Page for Pending Contact Requests

User Story 26: Implement Page for Contact List

User Story 27: Implement Backend Logic for Contacts Route

User Story 28: Document Previous Features

User Story 29: Prepare for Showcase.

User Story 30: Document Remaining Code

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:

Visual Studio Code

Firebase v.9.12.1

Firestore Database

Google Cloud Functions

Angular Framework v.16.2.1

Bootstrap v.5.3.1

Typescript v.5.1.3

Java JDK 17

Node JS v.16.14.0 - v.18.18.0

Git / GitHub

Sprints (Planning Sessions)

Sprint 1

Sprint Planning Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Nicolas Bello, Jonathan Oviedo, Christian Poellabauer (PO)

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 15.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 1: Research Google Flutter for Development and Scalability and Cross-Developing to Web Based Platforms from Mobile
- User Story 2: Research Architectural Design and Infrastructure
- User Story 3: Research Integration Possibilities with Python and other Technologies
- User Story 4: UI/UX of Application

The team members indicated their willingness to work on the following user stories:

- Jonathan Oviedo
 - User Story 3: Research Integration Possibilities with Python and other Technologies
- Jordan Bravo
 - User Story 2: Research Architectural Design and Infrastructure
 - User Story 4: UI/UX of Application
- Nicolas Bello
 - User Story 1: Research Google Flutter for Development and Scalability and Cross-Developing to Web Based Platforms from Mobile

*Sprint 2***Sprint Planning Meeting Minutes: 30 Minutes**

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo, Chris Poellabauer

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 15.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 5: Implement Landing Page
- User Story 6: Implement Login Page
- User Story 7: Set up NgRx, Documentation, Guidelines and Initial Commit of Project
- User Story 8: Setting up Firebase + Introducing Fake Data
- User Story 9: Implement Routes Depending on User being Patient or Provider

The team members indicated their willingness to work on the following user stories.

- Jordan Bravo
 - User Story 7: Set up NgRx, Documentation, Guidelines and Initial Commit of Project
 - User Story 8: Setting up Firebase + Introducing Fake Data
- Jonathan Oviedo
 - User Story 5: Implement Landing Page
- Nicolas Bello
 - User Story 6: Implement Login Page
 - User Story 9: Implement Routes Depending on User being Patient or Provider

*Sprint 3***Sprint Planning Meeting Minutes: 30 Minutes**

Project: Web Based Neurological Assessment

Attendees: Nicolas Bello, Jordan Bravo, Jonathan Oviedo, PO

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 18.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 10: Implement Dashboard
- User Story 11: Implement User Sign Up Option Page
- User Story 12: Implement Registration Form
- User Story 13: Implement User Drawer
- User Story 14: Redesign Firestore Database to Accommodate for Facilities, Admins, Providers and Patients
- User Story 15: Implement Email Verification for New Accounts

The team members indicated their willingness to work on the following user stories.

- Jordan Bravo
 - User Story 12: Implement Registration Form
 - User Story 11: Implement User Sign Up Option Page
 - User Story 14: Redesign Firestore Database to Accommodate for Facilities, Admins, Providers and Patients
- Nicolas Bello
 - User Story 10: Implement Dashboard
 - User Story 15: Implement Email Verification for New Accounts
- Jonathan Oviedo
 - User Story 10: Implement Dashboard

- User Story 15: Implement Email Verification for New Accounts

*Sprint 4***Sprint Planning Meeting Minutes: 30 Minutes**

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo, Chris Poellabauer

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 15.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 16: Implement Backend Logic for New Provider Creation and Handling.
- User Story 17: Create Tests for Existing Functionality.
- User Story 18: Dashboard Responsiveness and Route Guarding.

The team members indicated their willingness to work on the following user stories.

- Nicolas Bello
 - User Story 17: Create Tests for Existing Functionality.
 - User Story 18: Dashboard Responsiveness and Route Guarding.
- Jonathan Oviedo
 - User Story 17: Create Tests for Existing Functionality.
 - User Story 18: Dashboard Responsiveness and Route Guarding.
- Jordan Bravo
 - User Story 16: Implement Backend Logic for New Provider Creation and Handling.
 - User Story 17: Create Tests for Existing Functionality.

*Sprint 5***Sprint Planning Meeting Minutes: 30 Minutes**

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo, Chris Poellabauer

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 21 points.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 19: MVP of Contacts Page for a User
- User Story 20: Password Forgot Feature
- User Story 21: Login Via Gmail
- User Story 22: Route Guards for Route Protection
- User Story 23: Finalize the Side Nav for MVP

The team members indicated their willingness to work on the following user stories.

- Jordan Bravo
 - User Story 20: Password Forgot Feature
 - User Story 21: Login Via Gmail
 - User Story 22: Route Guards for Route Protection
- Nicolas Bello
 - User Story 22: Route Guards for Route Protection
 - User Story 23: Finalize the Side Nav for MVP
- Jonathan Oviedo
 - User Story 19: MVP of Contacts Page for a User
 - User Story 20: Password Forgot Feature

*Sprint 6***Sprint Planning Meeting Minutes: 30 minutes**

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Jonathan Oviedo, Nicolas Bello, Chris Poellabauer

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 24.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 24: Implement Landing Page for Contacts Route
- User Story 25: Implement Page for Pending Contact Requests
- User Story 26: Implement Page for Contact List
- User Story 27: Implement Backend Logic for Contacts Route
- User Story 28: Document Previous Features

The team members indicated their willingness to work on the following user stories.

- Jordan Bravo
 - User Story 27: Implement Backend Logic for Contacts Route
 - User Story 28: Document Previous Features
- Jonathan Oviedo
 - User Story 26: Implement Page for Contact List
 - User Story 25: Implement Page for Pending Contact Requests
 - User Story 28: Document Previous Features
- Nicolas Bello
 - User Story 24: Implement Landing Page for Contacts Route
 - User Story 28: Document Previous Features
 - User Story 25: Implement Page for Pending Contact Requests

*Sprint 7***Sprint Planning Meeting Minutes: 30 Minutes**

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo, Chris Poellabauer

Start time: 3:30 PM

End time: 4:00 PM

After discussion, the velocity of the team was estimated to be 18.

The product owner chose the following user stories to be done during the next sprint. They are ordered based on their priority.

- User Story 29: Prepare for Showcase.
- User Story 30: Document Remaining Code.

The team members indicated their willingness to work on the following user stories.

- Jordan Bravo
 - User Story 29: Prepare for Showcase.
 - User Story 30: Document Remaining Code.
- Jonathan Oviedo
 - User Story 29: Prepare for Showcase.
 - User Story 30: Document Remaining Code.
- Nicolas Bello
 - User Story 29: Prepare for Showcase.
 - User Story 30: Document Remaining Code

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.

Deployment Diagram

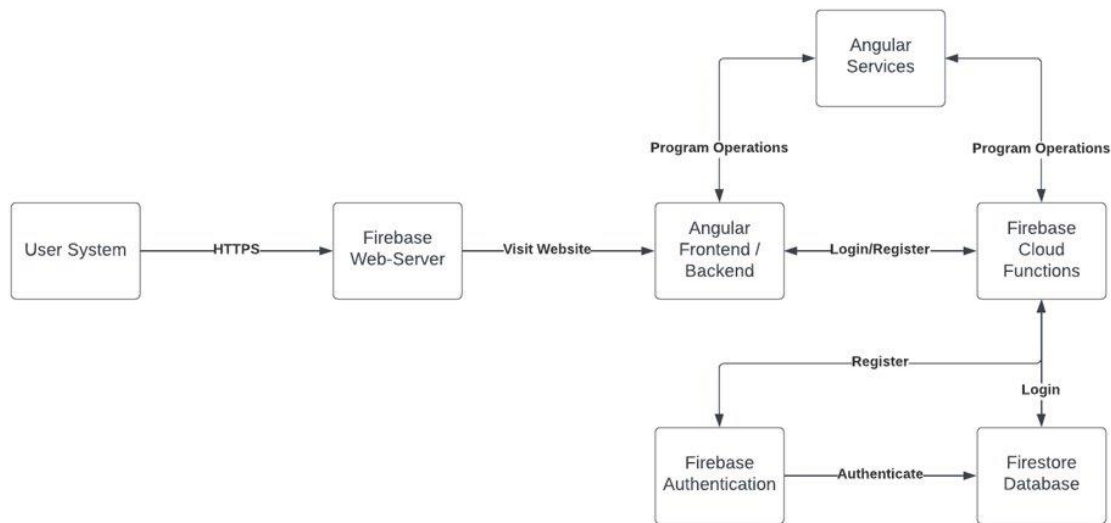


Figure 1 - How services are split apart and deployed

Architectural Patterns

The architectural patterns used for this project are:

Client-Server Architecture

The functions of our system fit into a client-server architecture because Google's Firebase hosts the project's website on their servers while providing other back-end services (BaaS) like storing user information in the Firestore Database. The UI/UX design and frontend implementation cater to the client side, where users interact with the application. This includes features like the landing page, log in, dashboard, user signup, and other user-facing functionalities. The client (web app) interacts with the server (back-end services/Firebase) by sending requests for data, authentication, and other operations. The server processes these requests, retrieves or manipulates data as needed, and sends back the appropriate responses to the client.

Microservices Architecture

There are multiple different services that are offered on our website because of the inherent use of Angular and Firebase. Angular allows us to follow a component-based structure

that inherently supports a modular base design, with each of our components providing a different service. With the use of Firebase, we are able to split off different back-end services such as user login and registration and other similar services. The features that we have implemented so far such as the: dashboard, user login/registration, calendar, contacts, and homepage are loosely coupled following the microservices paradigm.

System Validation

The system validation process for the NeuroTask project encompassed comprehensive assessments and tests across all facets of the application to ensure its functionality, usability, security, and scalability aligned with predetermined benchmarks. The validation procedures aimed to guarantee a seamless and robust user experience while meeting the project's defined objectives.

Rigorous testing was performed with backend services to ensure proper communication between the web app and Firebase, validating data storage, retrieval, and manipulation. Extensive validation of user-centric features, including login, signup, dashboard functionalities, contact management, and authentication mechanisms, ensured their reliability and responsiveness. Thorough usability testing and UI inspections verified the design's intuitiveness, consistency, and responsiveness across devices, guaranteeing an engaging and user-friendly interface.

Validation tests assessed the compatibility and consistent experience of NeuroTask across various platforms. Validation of authentication procedures, including email verification, login via Gmail, route guards, user authentication across web pages, and password reset functionalities ensured robust security measures.

Appendix A – UML Diagrams

Use Case Diagrams

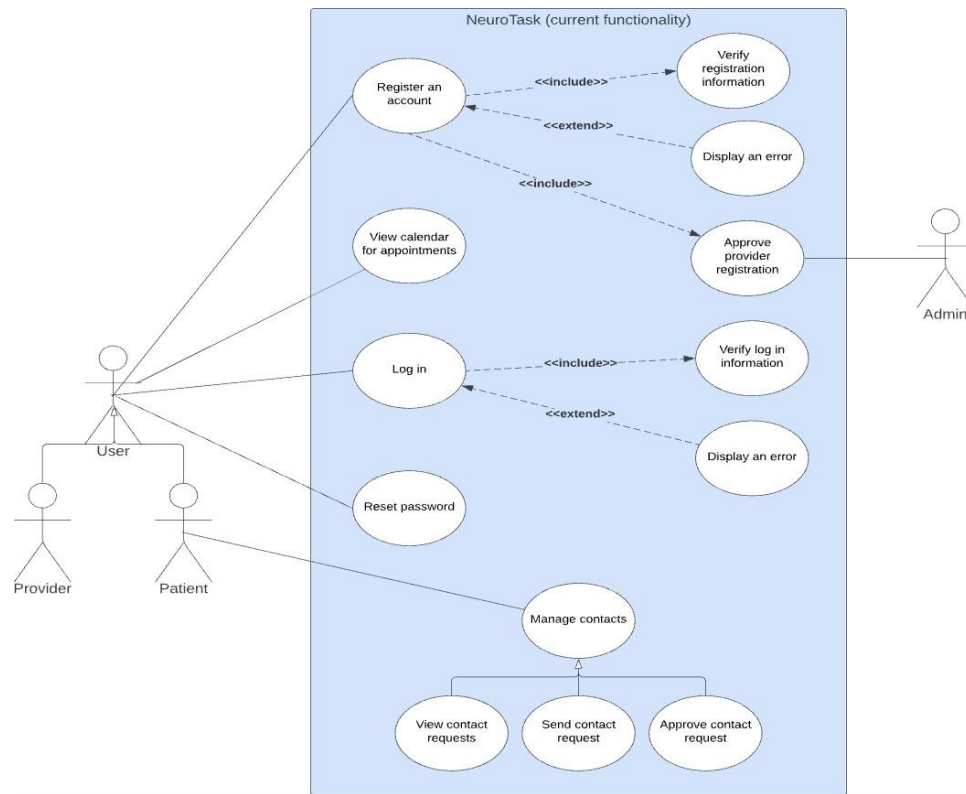


Figure 2 – Use case current functionality

Sequence Diagrams

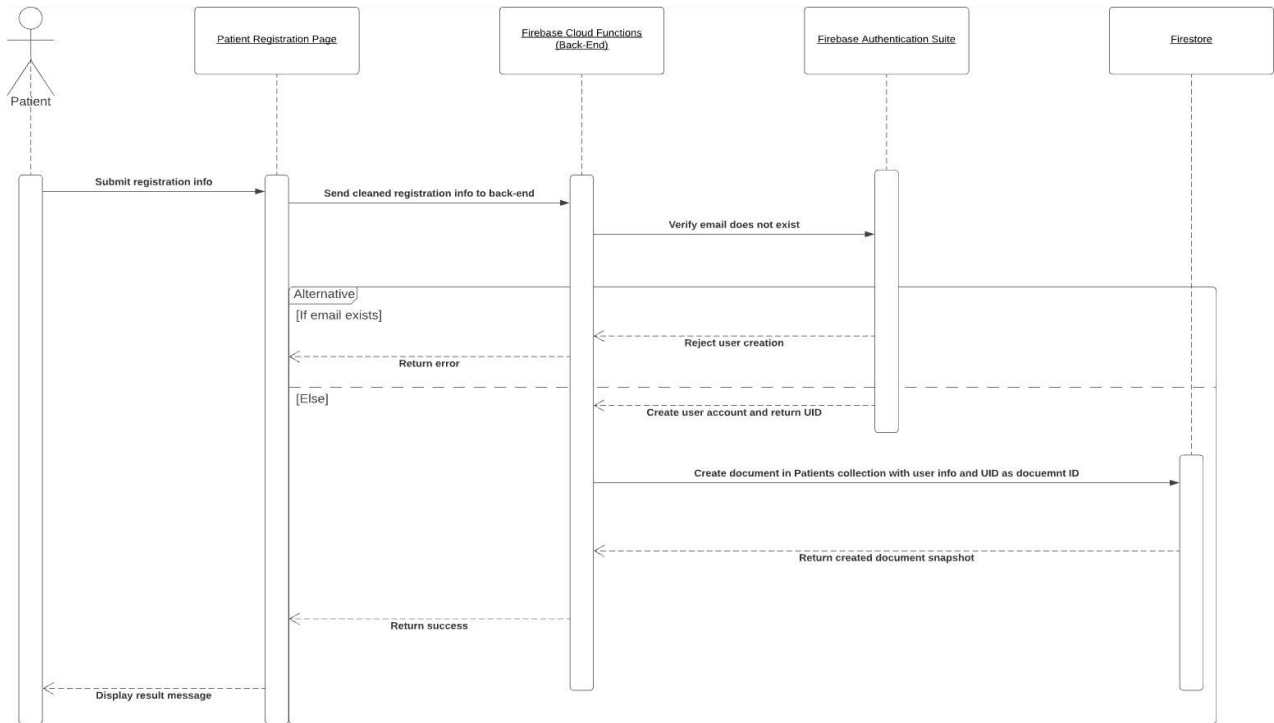


Figure 3 – Patient Registration

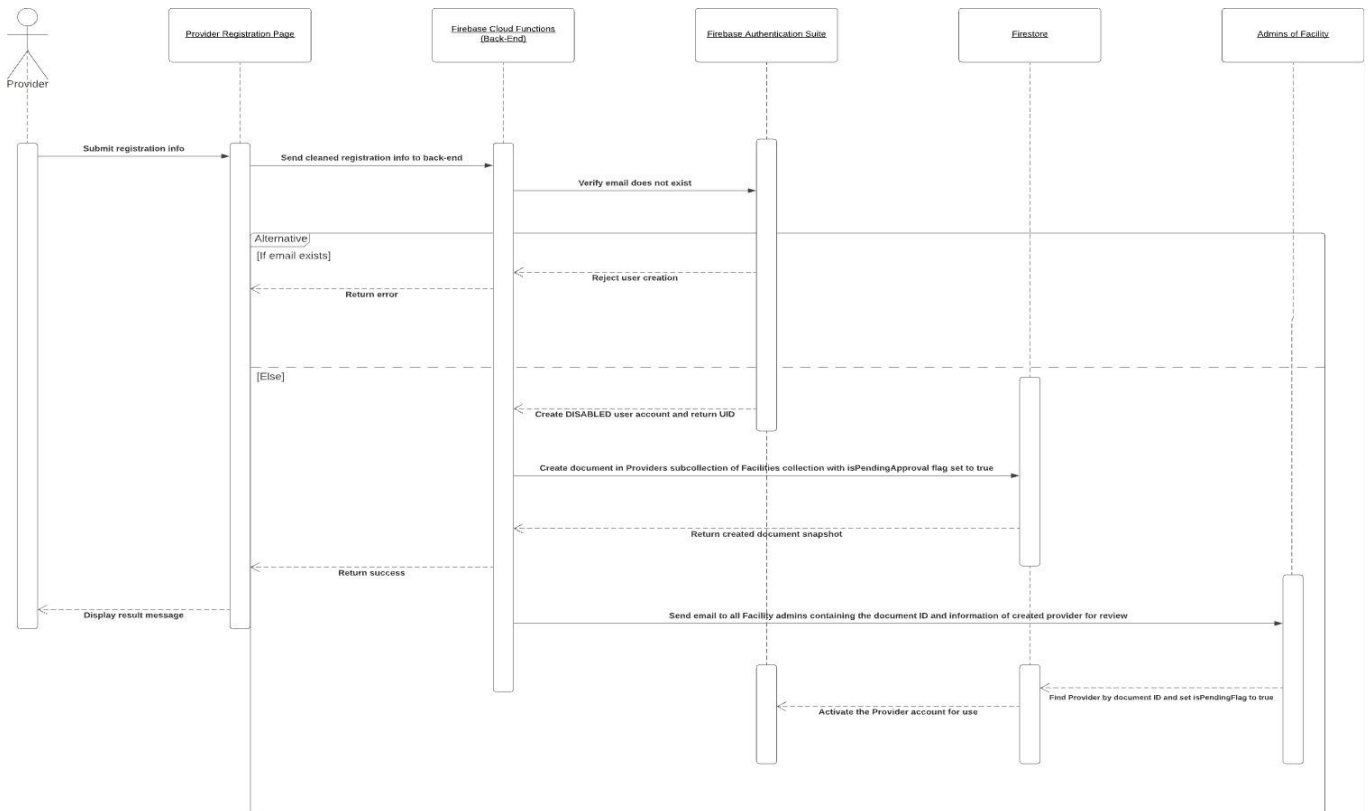
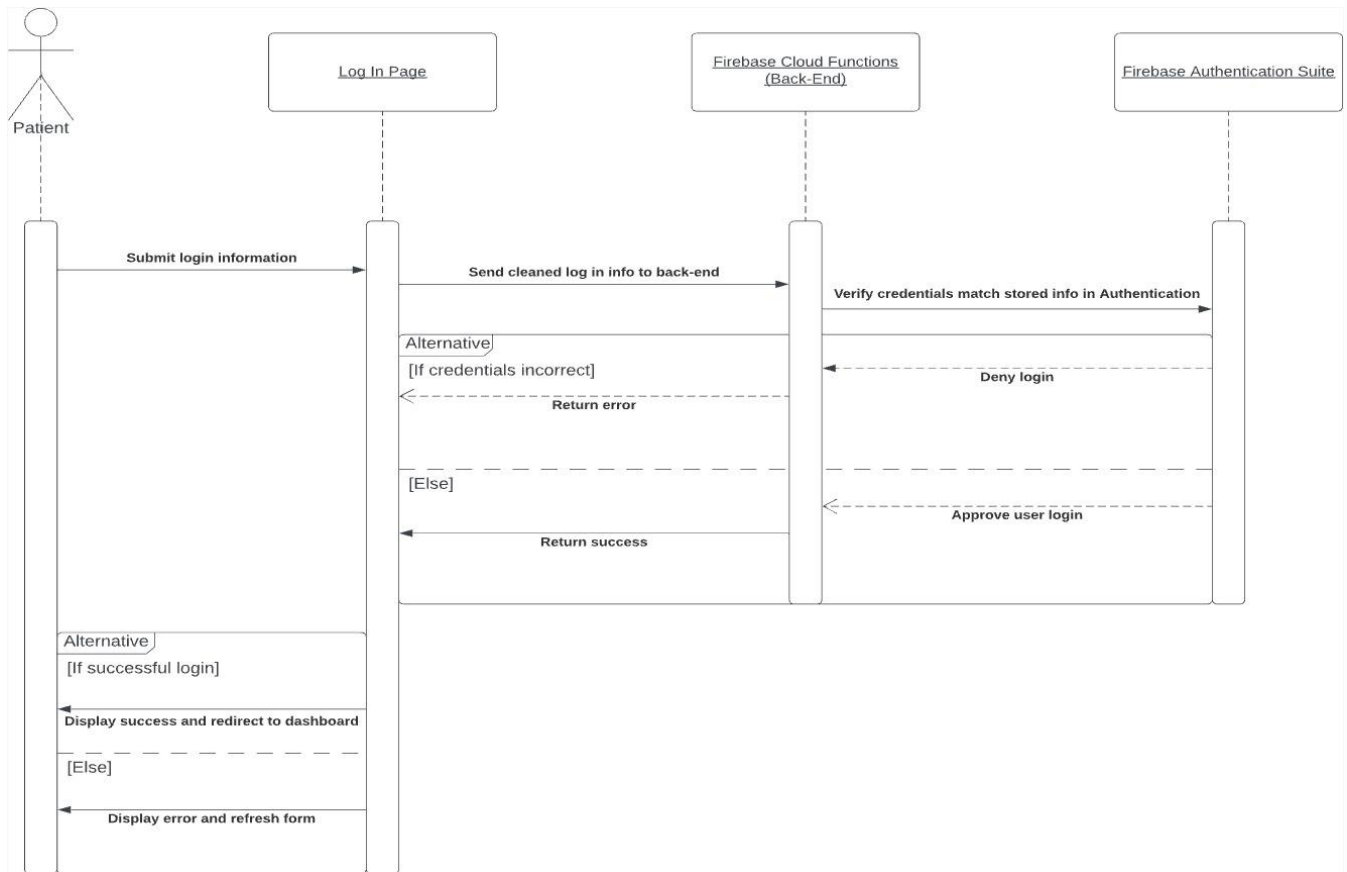


Figure 4 – Provider Registration**Figure 5 – Patient Login**

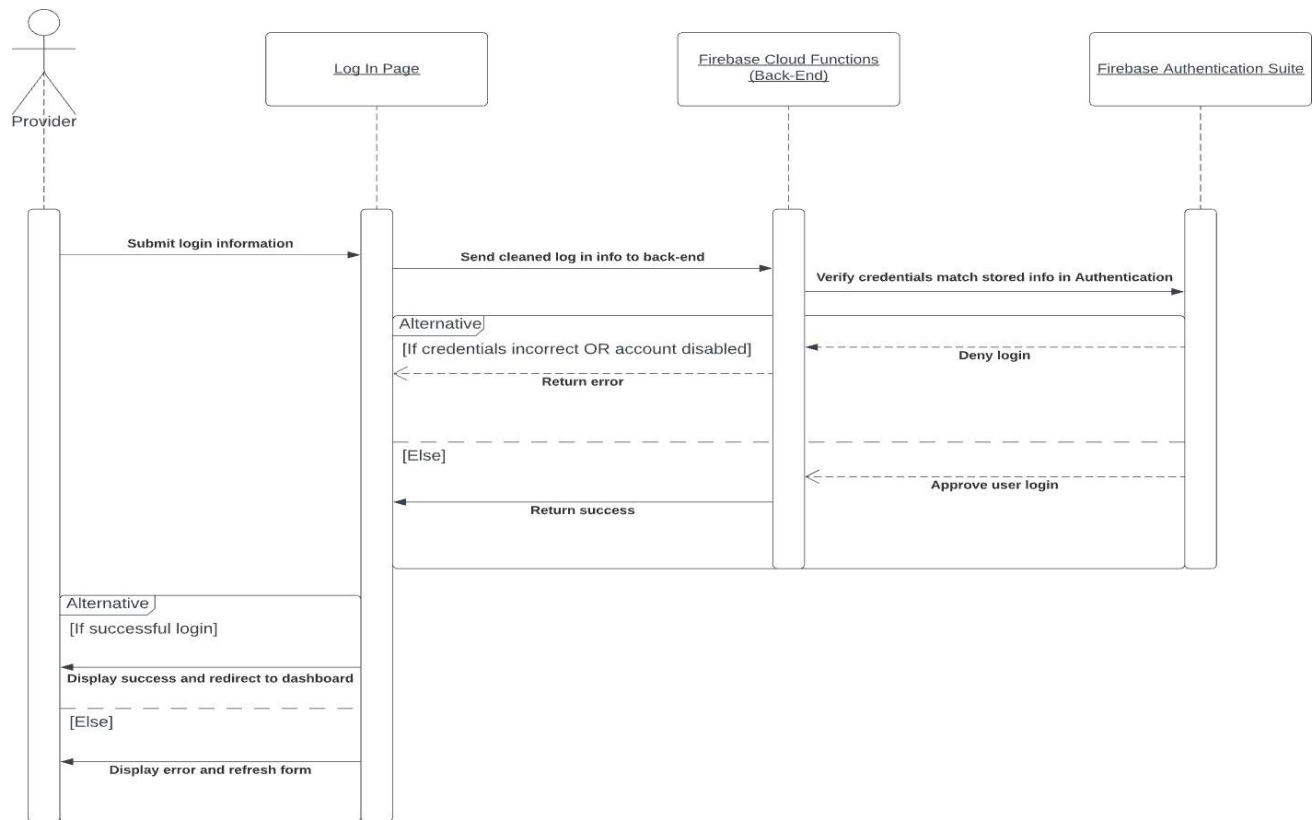


Figure 6 – Provider (Medical Expert) Login

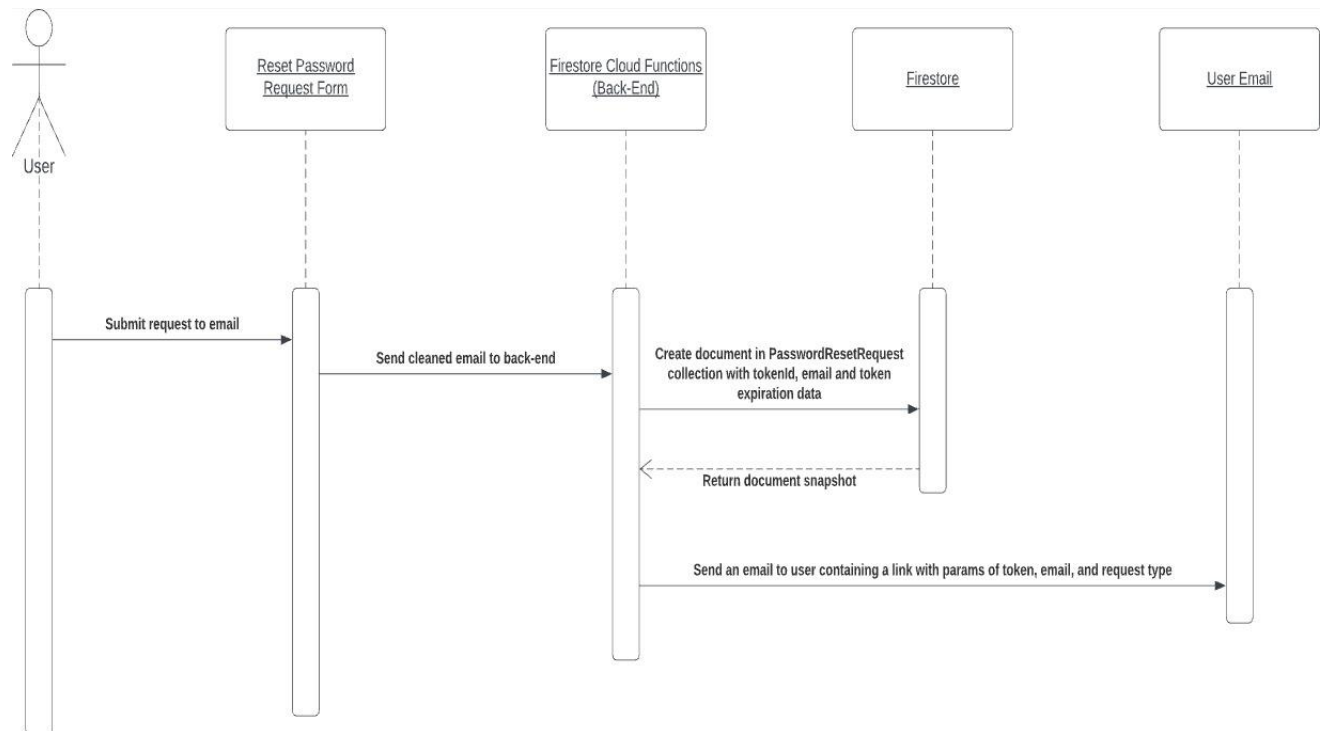
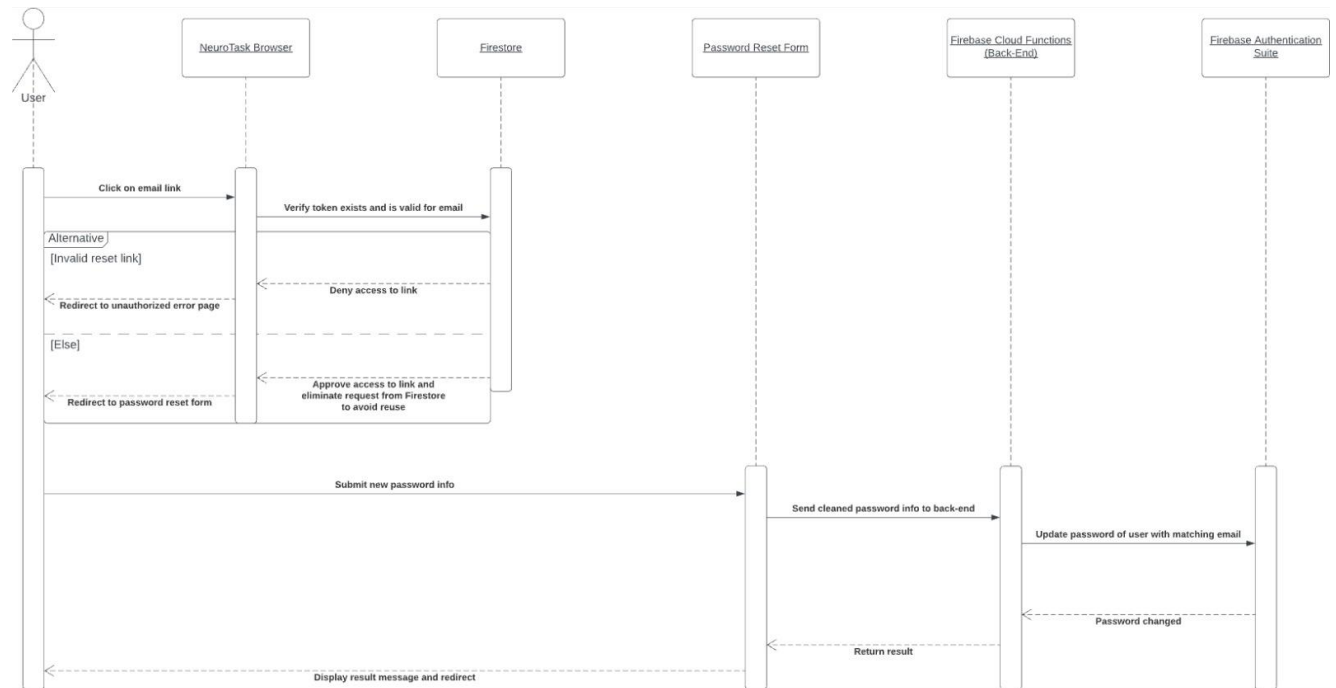
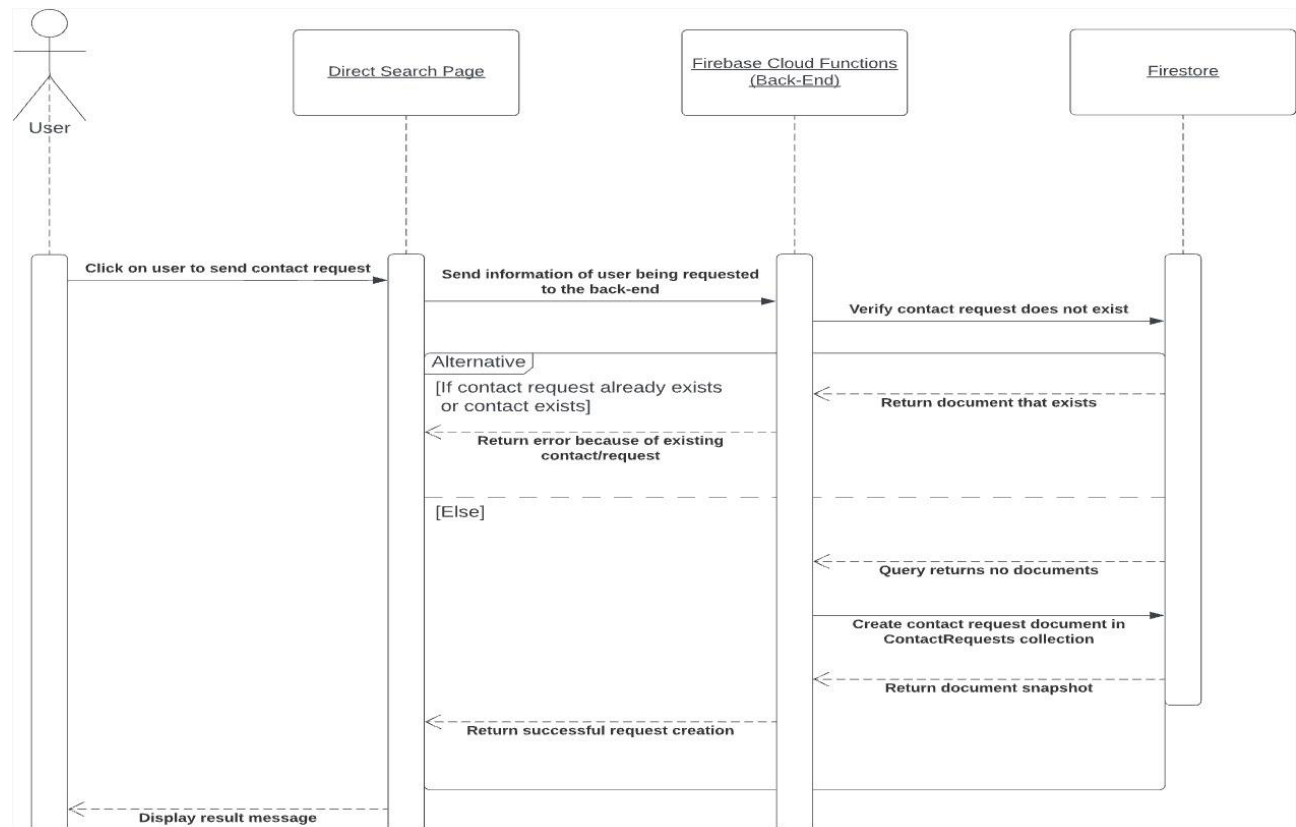
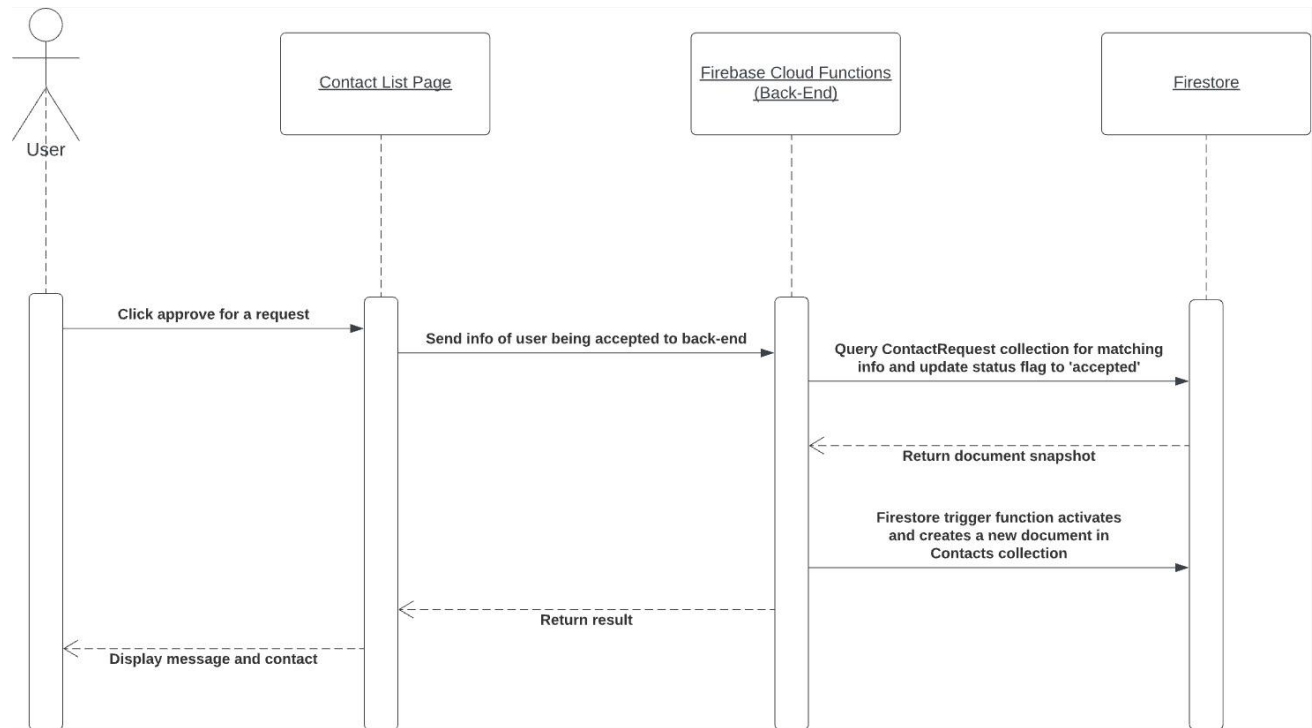


Figure 7 – User Password Reset Request

**Figure 8 – User Password Reset****Figure 9 – Send Contact Request**

**Figure 10** – Approve Contact Request

Appendix B – User Interface Design

NeuroTask

USER LOG IN FORM

Email:

Password:

[Forgot Password?](#)

Don't have an account? [Sign Up Now!](#)

Running in emulator mode. Do not use with production credentials.

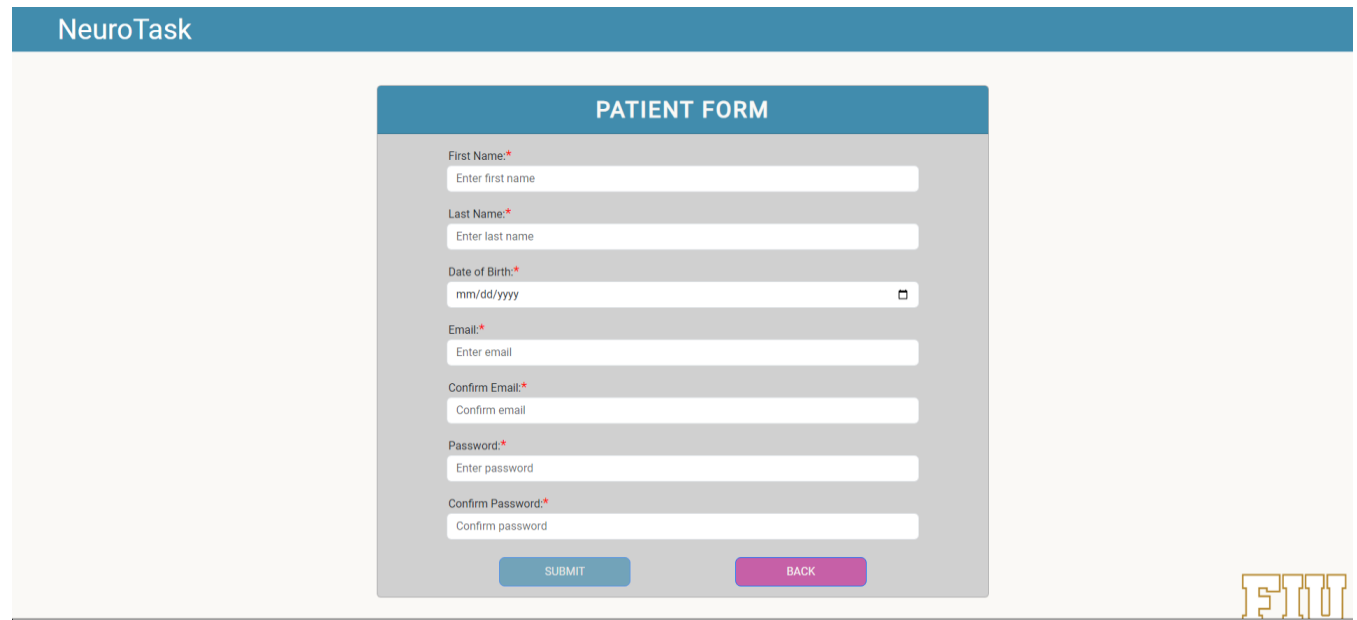
Figure 11 – The User Login Page

NeuroTask

CHOOSE AN OPTION

Running in emulator mode. Do not use with production credentials.

Figure 12 – Options for User Login Page



The image shows a web application interface for a patient sign-up form. The header is a blue bar with the text "NeuroTask". Below the header is a light gray box containing a form titled "PATIENT FORM". The form has the following fields: "First Name:*" with a text input, "Last Name:*" with a text input, "Date of Birth:*" with a date picker (mm/dd/yyyy), "Email:*" with a text input, "Confirm Email:*" with a text input, "Password:*" with a text input, and "Confirm Password:*" with a text input. At the bottom of the form are two buttons: "SUBMIT" (blue) and "BACK" (purple). To the right of the form is a large, stylized "NT" logo. Below the form, a small red text line reads: "Running in emulator mode. Do not use with production credentials."

Figure 13 – Patient Form Sign-up



Figure 14 – Homepage/Dashboard after successful login



Figure 15 – Sidebar navigation after user clicks on it (submenus also displayed)

Future teams should try to use smooth colors and try to follow the color scheme that we have listed within the GitHub documentation.

Appendix C – Sprint Review Reports

Sprint 1 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Jonathan Oviedo, Nicolas Bello

Start time: 5:00 PM

End time: 5:30 PM

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

- User Story 5: Implement Landing Page
- User Story 6: Set up NgRx, Documentation, Guidelines and Initial Commit of Project
- User Story 7: Setting up Firebase + Introducing Fake Data
- User Story 8: Implement Login Page
- User Story 9: Implement Routes Depending on User being Patient or Provider

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.

- None

Sprint 2 Review

Review Meeting Minutes: 30 minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Jonathan Oviedo, Nicolas Bello

Start time: 4:30 PM

End time: 5:00 PM

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

- User Story 5: Implement Landing Page
- User Story 6: Implement Login Page
- User Story 7: Set up NgRx, Documentation, Guidelines and Initial Commit of Project
- User Story 8: Setting up Firebase + Introducing Fake Data
- User Story 9: Implement Routes Depending on User being Patient or Provider

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.

- N/A

Sprint 3 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo

Start time: 5:00 PM

End time: 5:30 PM

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

- User Story 10: Implement Dashboard
- User Story 11: Implement User Sign Up Option Page
- User Story 12: Implement Registration Form
- User Story 13: Redesign Firestore Database to Accommodate for Facilities, Admins, Providers and Patients

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 14: Implement User Drawer
- User Story 15: Implement Email Verification for New Accounts

Sprint 4 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo

Start time: 4:00 PM

End time: 4:30 PM

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

- User Story 16: Implement Backend Logic for New Provider Creation and Handling.
- User Story 17: Create Tests for Existing Functionality.
- User Story 18: Dashboard Responsiveness and Route Guarding.

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 18: Dashboard Responsiveness and Route Guarding
 - While this story was being worked on, we decided to split it into the upcoming sprint because route guarding was yet to be implemented.

Sprint 5 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Nicolas Bello, Jonathan Oviedo

Start time: 4:00 PM

End time: 4:30 PM

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

- User Story 19: MVP of Contacts Page for a User
- User Story 20: Password Forgot Feature
- User Story 21: Route Guards for Route Protection

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 22: Finalize the Side Nav for MVP
 - Side nav requires more work to function within Angular's component-based system and implementation of NGX Bootstrap library.
- User Story 23: Login Via Gmail
 - Due to the complications that arose with the password reset feature and having to transition from Firebase built-in password reset to a custom system, we were not able to prioritize login features with Gmail at this time.

Sprint 6 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Jonathan Oviedo, Nicolas Bello

Start time: 4:00 PM

End time: 4:30 PM

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

- User Story 24: Implement Landing Page for Contacts Route
- User Story 25: Implement Page for Pending Contact Requests
- User Story 26: Implement Page for Contact List
- User Story 27: Implement Backend Logic for Contacts Route
- User Story 28: Document Previous Features

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.

Sprint 7 Review

Review Meeting Minutes: 30 Minutes

Project: Web Based Neurological Assessment

Attendees: Jordan Bravo, Jonathan Oviedo, Nicolas Bello

Start time: 4:00 PM

End time: 4:30 PM

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

- User Story 29: Prepare for Showcase.
- User Story 30: Document Remaining Code

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.

Appendix D – User Manuals, Installation/Maintenance Document, Shortcomings/Wishlist Document and other documents

Installation Manuel

The IDE the team used is Visual Studio Code for its flexibility with web development and ability to add extensions. The download for Visual Studio Code can be found [here](#).

We recommend these extensions:

- Angular Language Service by Angular
- Angular Schematics by Cyrille Tuzi
- CSS Peek by Pranay Prakash
- Firebase by toba
- HTML CSS Support by ecmel
- Material Icon Theme by Philipp Kief

Accessing the Repo

1. To gain access to the organization, FIU_MOSAIC, please reach out to the PO ([Dr. Christian Poellabauer](#)) to request access.
- Once access has been granted, you will see the home page of the repo under **NeuroTask_Web**. From here, you can view the main *README.md* that gives a little background on the project and the general overview of its structure.
2. Once you've gained access to the organization, you can now install [Git](#) to manage the workflow of the repo and clone it to your local environment.
- The official Git tutorial on installation for Linux, Windows and Mac can be found [here](#).
- (OPTIONAL) We recommend setting up an SSH key for GitHub to recognize your local environment and bypass security protocols when committing, pulling, etc. To get started, first look at the official tutorial for [generating a new SSH key and adding it to the ssh-agent](#), and then [adding the SSH key to your GitHub account](#).
3. Once you've downloaded Git, you can click on the <> **Code** dropdown button at the top of the repo and copy the URL for the repo.
- If you set up the SSH key, select the SSH url. Otherwise, select the HTTPS url.
4. Go to your CLI (terminal), change directories to your desired project location, and write git clone followed by the just copied URL. This will pull down the most recent version of the main branch into your current directory.

Accessing Firebase Production

To have access to Firebase, please log in with the created admin account:

Email: ntcap2fiu@gmail.com

sprin

Using this account, you can then invite yourself and other members of your team to the Firebase project.

Firebase [website](#)

Installing Dependencies and Packages

Node

At the time of writing, the team has been using different versions of Node with the oldest version being Node v18.7.1 and NPM version 9.6.7. There shouldn't be any issues on using different versions of Node, however, but we do recommend getting any version past the version listed.

Node and NPM are bundled together so you don't need to worry about separate downloads.

You can get the [latest version of Node LTS](#), or you can opt for a Node Version Manager called NVM which allows you to manage different versions of Node on your system.

After download, restart the terminal and run `npm -v` and `node -v` to assure that they downloaded correctly.

Project Dependencies

Once you have Node and NPM downloaded, you can change your terminal directory to the root of the project and run `npm i`. This will download all the dependencies that are inside the root `package.json` file.

After running `npm i` inside the root folder, change your directory in the terminal to the functions folder and run `npm i`. This will install the dependencies that are used to work with the back-end cloud functions from Firebase.

Running Project for First Time

When running the project for the first time, make sure you have these checked:

5. In the functions folder, you have the lib folder. This is the cloud functions compiled for use.
6. In the root `package.json`, you have the start script which runs the command `firebase emulators:exec --import=./seed --project=neurotask-afbd7 --ui 'ng serve'`. More information regarding this script can be found in the official Firebase documentation and the [documentation for seeding](#).
 - The original script will execute the emulators, import the metadata from the seed folder to populate the emulator with data, sets the project configuration to the name of the project, and finally loads up the UI using the `ng serve` Angular command.
 - For more information on

Shortcomings/Wishlist

This semester focused primarily on laying the groundwork and setting the structure of the project. While most of the deliverables announced during backlog refinement were met, there were some corners that were cut to meet expected completion dates and end-of-sprint showcases.

1. Even though we approached the software with a mobile-first approach that is used in web development, not all of the responsive design is at the level that we would want. There are some pages, like the home page on the dashboard when you log in, that is missing a responsive design.
2. Instead of using the built in `window.alert()` method to display alerts, we would've liked to use custom modals that better suit the design of the website and improve user experience.
3. We weren't able to implement an **admin** route for facility admins to manage providers through a UI. Instead, we opted for the admin or database super user to manage the data directly through Firestore. This still works as direct manipulation of the data on Firestore triggers the document listeners in the back-end cloud functions.
4. The side navigation bar used when the user logs in is in a buggy state. We used NgRx to store the state of the side nav and listen to those changes throughout the application. We would like for the side nav to have better usability and more responsiveness in accordance to the state it is in (collapsed or open).
5. The home page was added in the last sprint of development as a refinement to the backlog. Because of this, it is in a very barebones state with no mobile responsiveness.
6. Due to the complexity of medical software and the agreements, records and data needed to populate a users experience, we opted for a simple approach with very essential data to begin with. This includes data like first name, last name, date of birth, email and password.
7. In the contact list route, data is retrieved through an API call on route load and then fed to an onSnapshot listener, allowing for semi-real time updates because of the API call that needs to happen. This can be an issue in discrepancies with the UI and the data in the data base. We didn't have enough time to further expand on this feature and therefore, you still have to refresh the UI to see the data rather than live updates and displaying.
8. Due to the time constraints of the semester and the complexity behind designing and creating software from the ground up, the experience is mostly tailored to patients only. This is clearly noticeable when you go to the contacts search route and can only search facilities for providers. The intended use was for providers to view patients depending on different filters from that of a patient viewing providers.