FINAL REPORT

FARMINGDALE ALLIANCE MEDICAL

Making Medical Documentation Transparent

By: Andrew Bach, Ryan Bocklet, D'Andre Desir, Tika Dhakal, and John Fanara

URL for launched website: https://healthcareproxy-df31e.web.app/

Table of Contents

.

Team Farmingdale Alliance Medical

***Writing a Problem Statement* Worksheet**

Original problem or focusing question	Health care systems are difficult to navigate for patients and medical professionals. How can we improve the hospital staff and patient experience?
Stakeholders who are most affected by the problem	Patients and medical professionals
Type of problem	Mismanagement of information systems. Legacy software in use. Patients have a difficult time using these software systems.
Suspected cause of the problem	Outdated systems. Lack of understanding of technology. Lack of funding.
Goal for improvement and long-term impact	Easy access. Increase use of communication systems.
Proposal for addressing the problem	Create a system for the patient and medical team that is simple to use.

Problem statement	Problem Statement: Navigating health care systems poses challenges for both patients and medical professionals, leading to inefficiencies and frustration. The issue at hand is the cumbersome experience associated with hospital staff and patient interactions. Stakeholders, including patients and medical professionals, are significantly impacted by this problem. The type of problem primarily stems from mismanagement of information systems, compounded by the use of outdated legacy software that patients find difficult to use. Suspected causes include reliance on outdated systems, insufficient understanding of technology, and limited funding for system upgrades. To address these challenges, the goal is to improve access and enhance communication systems for both patients and medical teams, thereby streamlining the overall hospital experience. The proposed solution involves developing a user-friendly system tailored to the needs of patients and medical professionals, ensuring simplicity and efficiency in navigation and interaction. This initiative aims to foster a more seamless and satisfactory experience for all involved parties within the healthcare ecosystem.

USER STORY BACKLOG

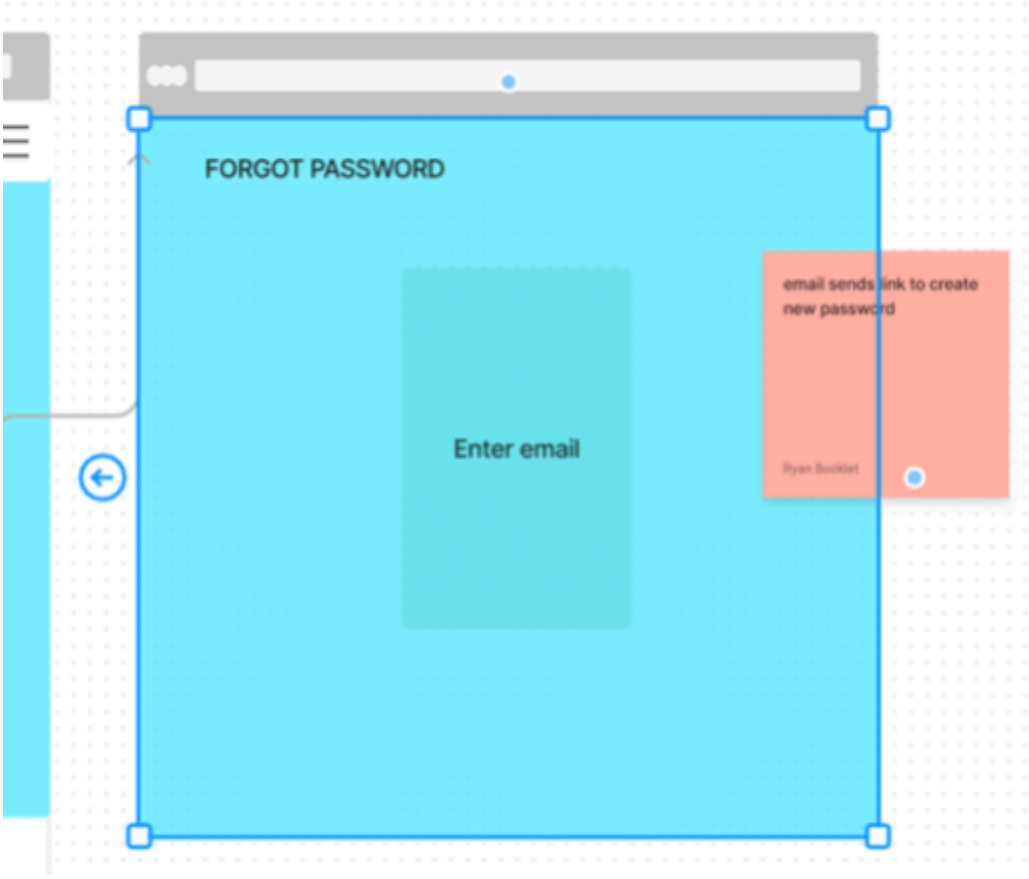
User Stories First Draft

User Stories Second Draft

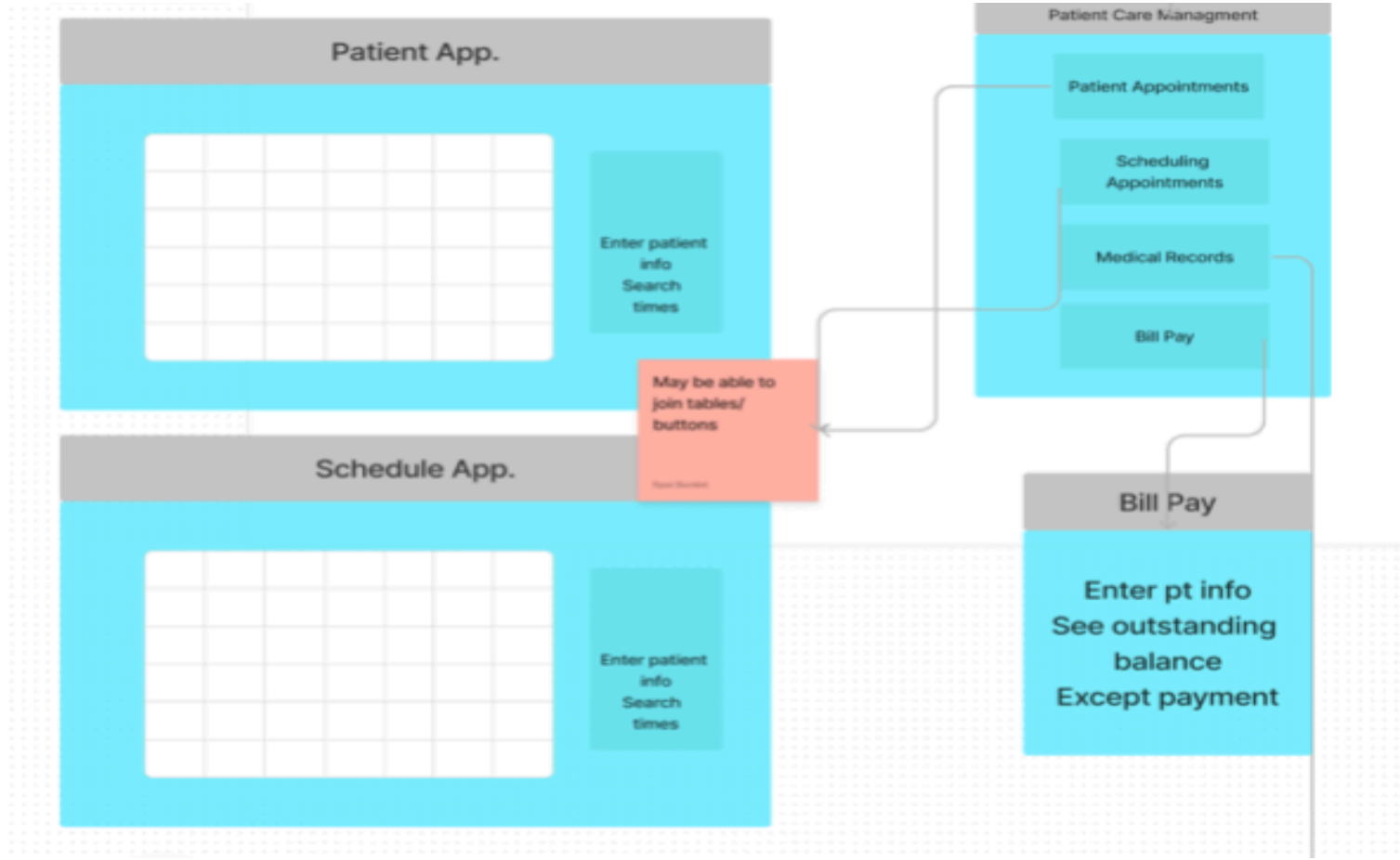
UX/UI

Initial design template


Website Wireframe
Date
Time
Product name or logo
USER LO...
USER NAME
PASSWORD
Portal ...
GO
FORGOT PASSWORD?
NEW USER


FORGOT PASSWORD
Enter email
email sends link to create new password

YOUR LOGOTYPE



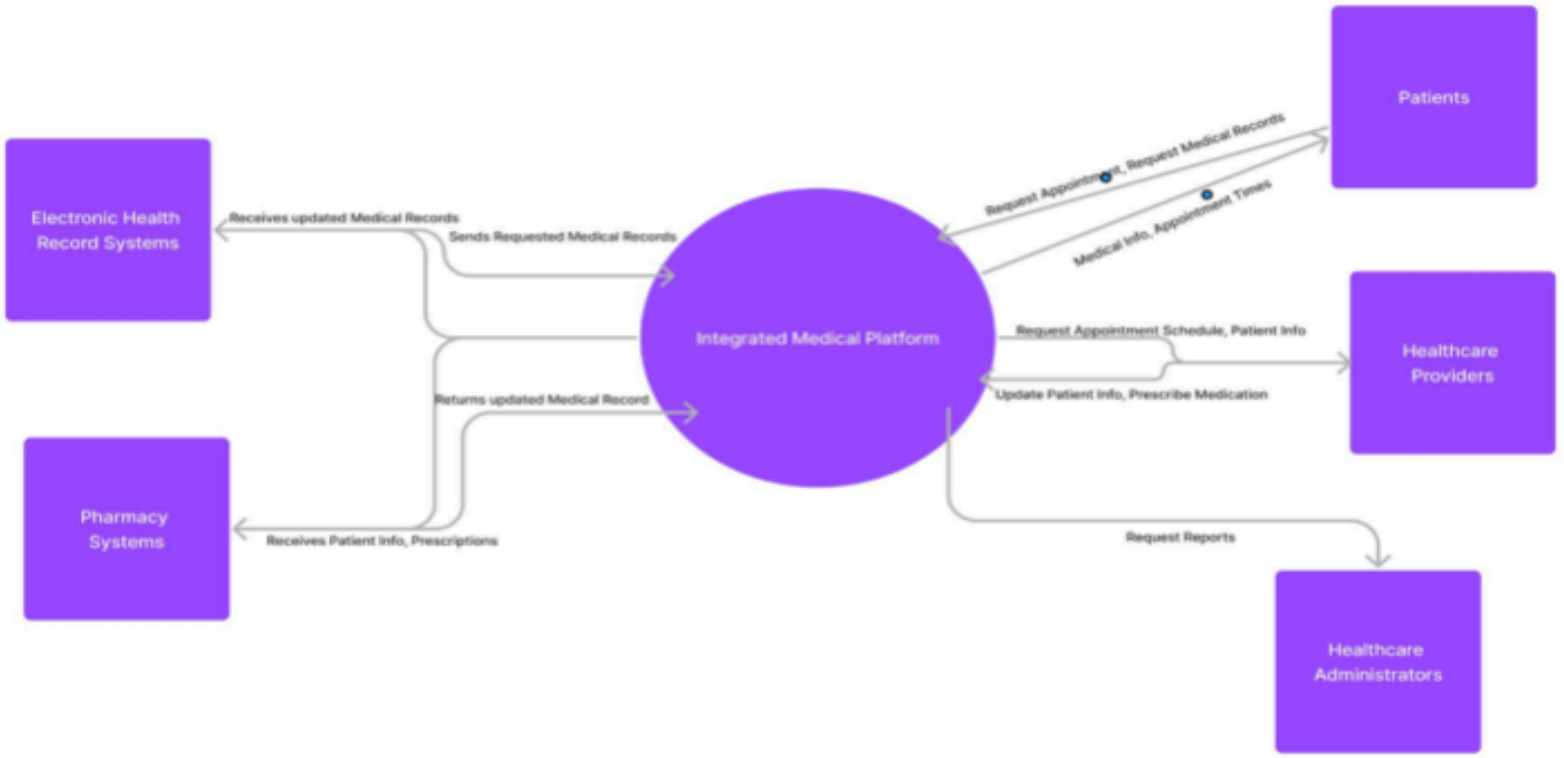
YOUR LOGOTYPE

Completed System Images

Use the link below for a walk through of the entire system.

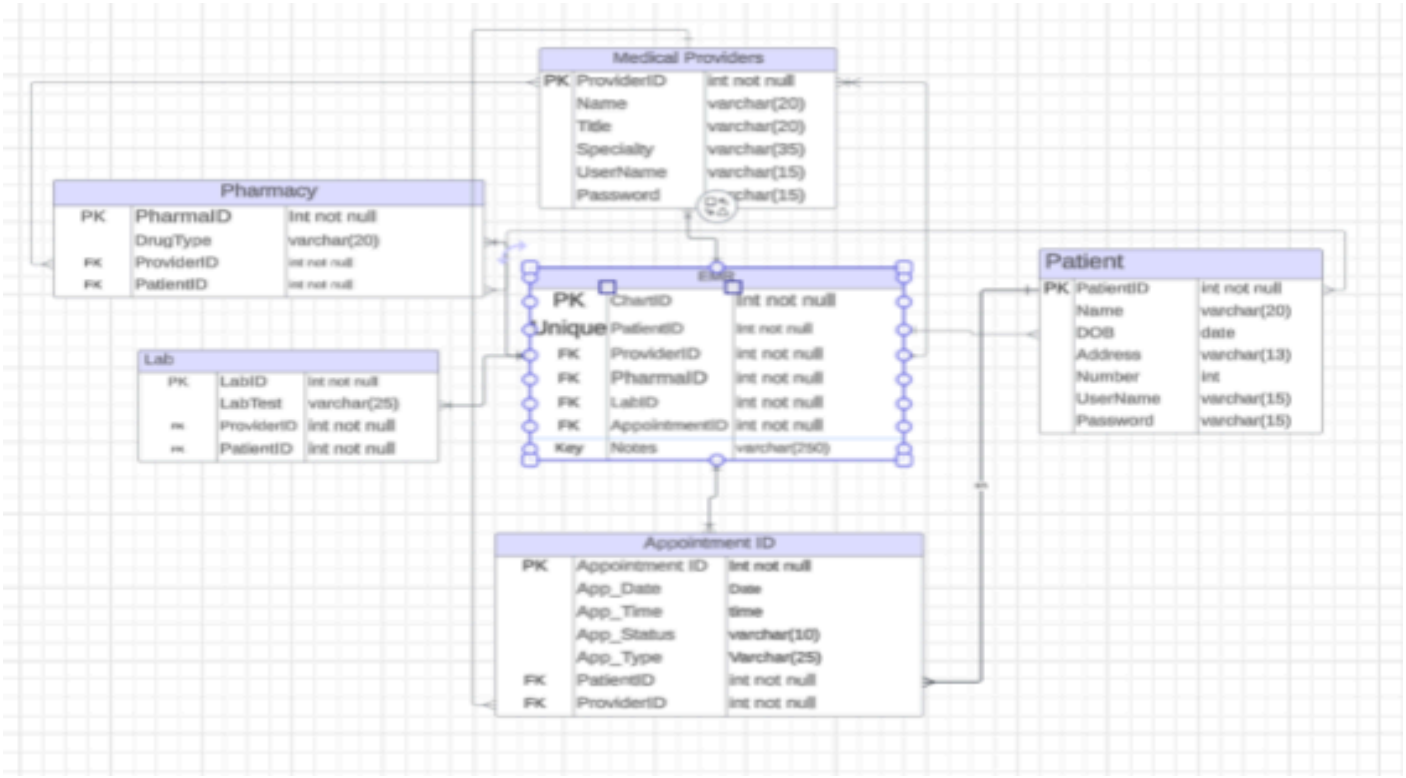
https://docs.google.com/presentation/d/1YCRBBxoz-Jf4pE2bm6ZsaC1qO_OLGA7X/edit?usp=sharing&ouid=115272483510807820494&rtpof=true&sd=true

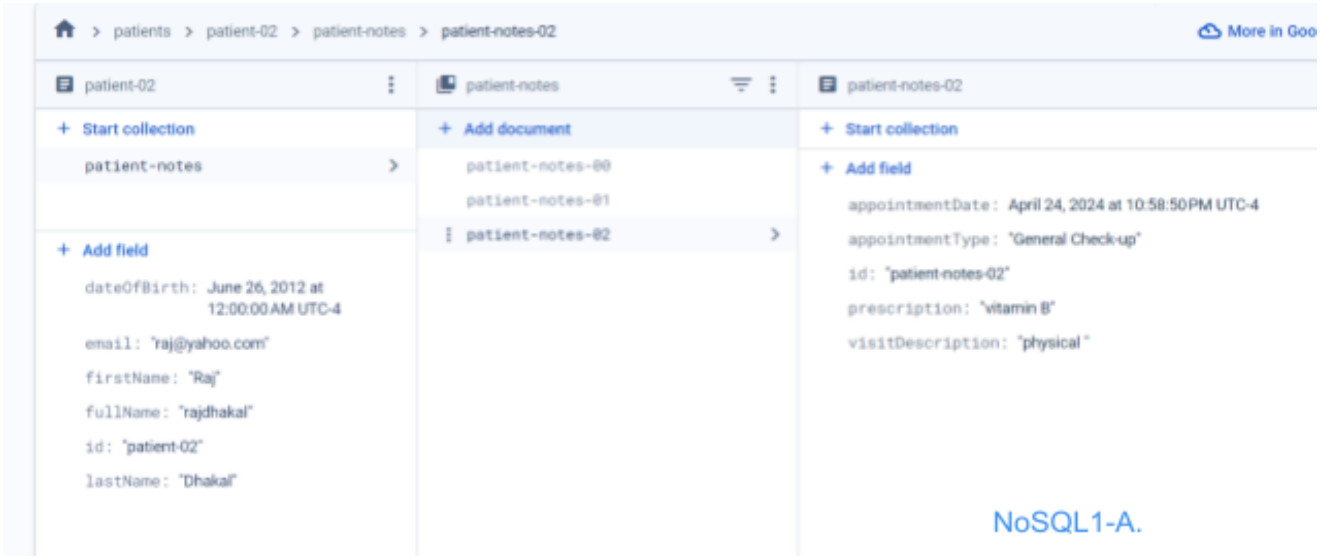
Context Diagram



Database

Initially the design was for an SQL database. After researching ideas for our working environment, we decided to use a NoSQL database. See image NoSQL1-A below for an example of a patient's stored record.





NoSQL1-A.

Retrospectives

IdeaBoardz - SR PROJECT

The team retrospective enabled the team to reflect on the past few weeks. We shared our individual ideas and unique experiences. An important aspect of team building is learning about the perspectives of our teammates. This creates a stronger team because you can approach problems from different mindsets.

Keep Doing-

- Working together
- Communication
- Creating user stories of medical documentation and patient care

Start Doing

- Reading full assignment description
- Narrowing scope of project

Stop Doing

- Stop meeting so close to deadline

Action items

- Read full assignment description

The meetings these past two weeks have provided an opportunity for the team to collectively review our progress and experiences. By sharing our individual perspectives and insights we were able to come up with some things that we should keep doing, start, or stop doing. Doing this helps team dynamics and allows us to visualize what needs to be done.

Keep Doing-

Working together

Narrowing the scope of the project

Start Doing-

Reading the full assignment descriptions

Stop Doing-

Stop meeting so close to the deadline

Less of-

Getting sidetracked

Teammates missing meetings

More of-

Troubleshooting sessions to make sure everything works as intended

Action Items-

Conduct regular check-ins with each other

Over the last few weeks, our team gatherings have offered a chance to collectively assess our advancements and troubles in this project. Through exchanging our unique perspectives, we've identified areas to maintain, start, or stop, which can help our team develop into the coming weeks.

Keep Doing-

Working together

Start Doing-

Narrowing the scope of the project

Stop Doing-

Stop meeting so close to the deadline

Getting sidetracked

Less of-

Teammates missing meetings

More of-

Troubleshooting sessions to make sure everything works as intended

Reading the full assignment descriptions

Action Items-

Conduct regular check-ins with each other

Keep Doing-

Working together

Start Doing-

Stay consistent with the plan instead of changing direction

Add features

Stop Doing-

Stop meeting so close to the deadline

Getting sidetracked

Narrowing the scope of the project

Less of-

Teammates missing meetings

More of-

Troubleshooting sessions to make sure everything works as intended

Reading the full assignment descriptions

Action Items-

Conduct regular check-ins with each other

Keep Doing-

Working together

Start Doing-

Stay consistent with the plan instead of changing direction

Set interim due dates for the team

Stop Doing-

Stop meeting so close to the deadline

Getting sidetracked

Add features

Less of-

Teammates missing meetings

More of-

Troubleshooting sessions to make sure everything works as intended

Reading the full assignment descriptions

Action Items-

Conduct regular check-ins with each other

Keep Doing-

Working together

Start Doing-

Stay consistent with the plan instead of changing direction

Set interim due dates for the team

Stop Doing-

Stop meeting so close to the deadline

Getting sidetracked

Add features

Less of-

Teammates missing meetings

More of-

Troubleshooting sessions to make sure everything works as intended

Reading the full assignment descriptions

Action Items-

Conduct regular check-ins with each other

Team Retrospective

Keep Doing-

Working together

Start Doing-

Stay consistent with the plan instead of changing direction

Set interim due dates for the team

Wrapping things up

Stop Doing-

Stop meeting so close to deadline

Less of-

Teammates missing meetings

More of-

Reading the full assignment descriptions

Action Items-

Conduct regular check-ins with each other

TEAM #2 TEST PLAN	RYAN, D'ANDRE, JOHN,ANDREW,& TIKA				
Feature UserStories	**Given**	**When**	**Then**	**Pass/ Fail**	**Test Note**
1 J	**Given a patient completes the registration process.**	**When the registration is successful**	**Then the patient is added to database and logged in.**	***Pass***	P Registra on
2 J	**Give a patient does not complete the registration process.**	**When the registration is unsuccessful**	**Then the patient receives an error message and registration does not go through.**	***Pass***	P Registra on
3 J	**Given an admin completes the registration process.**	**When the registration is successful**	**Then the admin is added to database and logged in.**	***Pass***	A Registra on

4 J	Given an admin does not complete the registration process.	When the registration is unsuccessful	Then the admin receives an error message and registration does not go through.	Pass
5 J	Given a patient is attempting to log in.	When the patient enters their credentials	Then the patient securely logs in and gains access to the patient info page.	Pass
6 J	Given a patient's login attempt fails.	When the patient receives an error message	Then the user receives an error message.	Pass

7 R	Given an admin is attempting to log in.	When the admin enters their credentials	Then the admin securely logs in and gains access to administrative features.	Pass

8 R	Given an admin login attempt fails.	When the admin inputs the incorrect information	Then clear documentation appears to guide the user.	***Pass***
9 R	Given a patient forgets their password.	When the patient selects the password reset option	Then the patient receives an email with instructions to reset the password	***Pass***
10 R	Given an admin forgets their password	When the admin selects the password reset option	Then the admin receives an email with instructions to reset the password.	***Pass***
11 R	Given an admin is logged into the Farmingdale Alliance Medical app.	When the admin navigates to the patient visit form section	Then the admin can input patient visit information. The form includes fields for capturing patient information such as name, dob, prescription/medication, visit description, appointment type.	***Pass***

12 A	Given an admin is logged into the Farmingdale Alliance Medical app	When the admin is searching for patient records	Then they will have a field to search by last name and date of birth.	***Pass***
13 A	Given an admin has submitted the patient visit form.	When the form submission is successful	Then the admin receives a confirmation message that the report was successfully added to database.	***Pass***
14 A	Given a user is on the home page.	When the patient accesses the main menu or navigation bar	Then the patient can easily find and access information about the mission statement, contact details, and help resources	***Pass***
15 A	Given a user is signed in	When they would like to log off the app	Then they can sign out with the log off button.	***Pass***

16 D	Given a patient is logged in.	When they are brought to the patient information page	Then they can view their logged patient notes	***Pass***
17 D	Given a patient is on the patient information page.	When they select an appointment type from the drop down menu	Then they can filter which patient notes they are able to view.	***Pass***
18 D	Given a patient is on the patient information page.	When the user selects the download PDF button	Then they can download a copy of their records as a pdf file.	***Pass***
19 D	Given a user has a question about the site	When the user selects contact us	Then they are able to leave a message for us to respond to.	***Pass***
20 D	Given a user is on the home page.	When the user clicks the about, mission statement, contact us, or help buttons	Then each button will bring up a new page.	Stretch
21 D	Given a patient is logged on	When searching for a healthcare provider	Then they can click on a map link that brings them to a map that states the location of providers in any area	Future

System Overview

Project version control. Git https://github.com/johnfanara/healthcareproxy

1. Front-end components

i. Design- HTML CSS, most up-to-date versions.

a. HTML5

b. CSS3

ii. Data control- JavaScript

a. React.js

b. Node.js

2. Back-end components

i. Firebase-JavaScript SDK- Included in node.js package.

3. Database components

i. Firebase Realtime Database

Environment- platforms, libraries, simulators, etc. Needed for application.

1.ii) Data control- JavaScript

a. React.js- React.js is a JavaScript library that aids in dynamic web/app development.

b. Node.js- Node.js is a framework that allows JavaScript to be executed outside of the browser for communication with a database or other third-party systems.

2.i) Firebase-JavaScript SDK- Included in node.js package.

This tool kit or “software development kit” allows communication between our Google Firebase and our Node.js application. It is included with the installation of Node.js.

3.i) Firebase Realtime Database-

Our data is being stored in a NoSQL database. This back end as a service also includes things like user authentication, storage, notifications, email communication and analytics.

Citations

1. Date Picker, React.js library used in admin Report form (line 118) and patient registration form (line 106)
```
<DatePicker
            selected={dateOfBirth}
            onChange={date => setDateOfBirth(date)}
            dateFormat="MM/dd/yyyy"
            placeholderText="Date of Birth"
            showYearDropdown={true}
            scrollableYearDropdown
            yearDropdownItemNumber={60}
            Required
          />
          </div>
```

2. Google.com-Images & videos are downloaded from open sources. These are stored in the images folder and are altered to suit the needs of the webpage. The four image buttons on the home page and the mp4 video that plays when the page loads.
3. MDBootsrap.com Custom React footer: https://mdbootstrap.com/docs/react/navigation/footer/

```
import React from 'react';
import { MDBFooter, MDBContainer, MDBRow, MDBCol, MDBIcon } from 'mdb-react-ui-kit';

export default function App() {
  return (
    <MDBFooter bgColor='light' className='text-center text-lg-start text-muted'>
      <section className='d-flex justify-content-center justify-content-lg-between p-4 border-bottom'>
        <div className='me-5 d-none d-lg-block'>
          <span>Get connected with us on social networks:</span>
        </div>

        <div>
```

```
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="facebook-f" />
    </a>
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="twitter" />
    </a>
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="google" />
    </a>
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="instagram" />
    </a>
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="linkedin" />
    </a>
    <a href='' className='me-4 text-reset'>
      <MDBIcon fab icon="github" />
    </a>
  </div>
</section>

<section className=''>
  <MDBContainer className='text-center text-md-start mt-5'>
    <MDBRow className='mt-3'>
      <MDBCol md="3" lg="4" xl="3" className='mx-auto mb-4'>
        <h6 className='text-uppercase fw-bold mb-4'>
          <MDBIcon icon="gem" className="me-3" />
          Company name
        </h6>
        <p>
```

```
    Here you can use rows and columns to organize your footer content. Lorem ipsum dolor sit amet,
    consectetur adipisicing elit.
  </p>
</MDBCol>

<MDBCol md="2" lg="2" xl="2" className='mx-auto mb-4'>
  <h6 className='text-uppercase fw-bold mb-4'>Products</h6>
  <p>
    <a href='#!' className='text-reset'>
      Angular
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      React
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      Vue
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      Laravel
    </a>
  </p>
</MDBCol>

<MDBCol md="3" lg="2" xl="2" className='mx-auto mb-4'>
```

```
  <h6 className='text-uppercase fw-bold mb-4'>Useful links</h6>
  <p>
    <a href='#!' className='text-reset'>
      Pricing
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      Settings
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      Orders
    </a>
  </p>
  <p>
    <a href='#!' className='text-reset'>
      Help
    </a>
  </p>
</MDBCol>

<MDBCol md="4" lg="3" xl="3" className='mx-auto mb-md-0 mb-4'>
  <h6 className='text-uppercase fw-bold mb-4'>Contact</h6>
  <p>
    <MDBIcon icon="home" className="me-2" />
    New York, NY 10012, US
  </p>
  <p>
```

```
              <MDBIcon icon="envelope" className="me-3" />
              info@example.com
            </p>
            <p>
              <MDBIcon icon="phone" className="me-3" /> + 01 234 567 88
            </p>
            <p>
              <MDBIcon icon="print" className="me-3" /> + 01 234 567 89
            </p>
          </MDBCol>
        </MDBRow>
      </MDBContainer>
    </section>

    <div className='text-center p-4' style={{ backgroundColor: 'rgba(0, 0, 0, 0.05)' }}>
      © 2021 Copyright:
      <a className='text-reset fw-bold' href='https://mdbootstrap.com/'>
        MDBootstrap.com
      </a>
    </div>
  </MDBFooter>
 );
}
```

Firebase code is generated by google firebase NoSQL database for interaction with the application. It is account specific.

4. Firebase create user with email and password:
   import { getAuth, createUserWithEmailAndPassword } from "firebase/auth";

```
const auth = getAuth();
createUserWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {
    // Signed up
    const user = userCredential.user;
    // ...
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    // ..
  });
```

5. Firebase sign in user with email and password:

```
import { getAuth, signInWithEmailAndPassword } from "firebase/auth";

const auth = getAuth();
signInWithEmailAndPassword(auth, email, password)
  .then((userCredential) => {
    // Signed in
    const user = userCredential.user;
    // ...
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
  });
```

6. Firebase sign out:

```
import { getAuth, signOut } from "firebase/auth";
```

```
const auth = getAuth();
signOut(auth).then(() => {
  // Sign-out successful.
}).catch((error) => {
  // An error happened.
});
```

7. Firebase reset user password:

```
import { getAuth, sendPasswordResetEmail } from "firebase/auth";

const auth = getAuth();
sendPasswordResetEmail(auth, email)
  .then(() => {
    // Password reset email sent!
    // ..
  })
  .catch((error) => {
    const errorCode = error.code;
    const errorMessage = error.message;
    // ..
  });
```

8. Initialize cloud firestore:

```
import { initializeApp } from "firebase/app";
import { getFirestore } from "firebase/firestore";

// TODO: Replace the following with your app's Firebase project configuration
// See: https://support.google.com/firebase/answer/7015592
const firebaseConfig = {
```

FIREBASE_CONFIGURATION

};

// Initialize Firebase
const app = initializeApp(firebaseConfig);

// Initialize Cloud Firestore and get a reference to the service
const db = getFirestore(app);

9. Set firebase database document:
```
import { doc, setDoc } from "firebase/firestore";

// Add a new document in collection "cities"
await setDoc(doc(db, "cities", "LA"), {
  name: "Los Angeles",
  state: "CA",
  country: "USA"
});
```

10. Getting firebase database document:
```
import { doc, getDoc } from "firebase/firestore";

const docRef = doc(db, "cities", "SF");
const docSnap = await getDoc(docRef);

if (docSnap.exists()) {
  console.log("Document data:", docSnap.data());
} else {
```

```
  // docSnap.data() will be undefined in this case
  console.log("No such document!");
}
```

11. Execute firebase database query:

```
import { collection, query, where, getDocs } from "firebase/firestore";

const q = query(collection(db, "cities"), where("capital", "==", true));

const querySnapshot = await getDocs(q);
querySnapshot.forEach((doc) => {
  // doc.data() is never undefined for query doc snapshots
  console.log(doc.id, " => ", doc.data());
});
```

12. Generate PDF file with jsPDF:

```
import { jsPDF } from "jspdf";

// Default export is a4 paper, portrait, using millimeters for units
const doc = new jsPDF();

doc.text("Hello world!", 10, 10);
doc.save("a4.pdf");
```

Conclusion

In conclusion, our project has delved into the intricate landscape of healthcare systems, aiming to alleviate the challenges faced by both patients and medical professionals. Through our collective efforts, we identified the core issue of cumbersome interactions within hospitals, exacerbated by outdated information systems and a lack of technological fluency. Our goal was clear: to enhance access and communication channels, thereby improving the hospital experience.

Throughout this journey, each team member contributed distinctively, reflecting on personal growth and newfound skills. Andrew spearheaded administrative tasks, honing his organization and time management abilities. Tika emphasized the importance of structure and adaptability through user case development. D'Andre showcased proficiency in front-end UI enhancement and collaboration tools like GitHub. Ryan's journey encompassed team management, technical and agile development. John's expertise in backend development, particularly with Firebase, was pivotal in materializing our project vision.

As individuals, we've evolved, equipped with a deeper understanding of healthcare technology and collaborative software development. Our experiences in BCS 430W have not only fortified our technical prowess but also instilled confidence in our ability to tackle future challenges. With this project as a foundation, we step forward, poised to make meaningful contributions to the ever-evolving landscape of healthcare technology.