ZiHealth

CMSC 495 6380

Khalilov, Santos, Bell, Gani, Levy, Arriaga

Final Documentation

Table of Contents

Team Roles

Project Lead - Ziya Khalilov
Database Lead - Alfred Conrad Santos
Backend/Website Development Lead - Darryl Bell
Documentation Lead - Waleed Gani
AWS Lead - Ari Levy
QA Lead - Kristen Arriaga

Project Overview

Our software aims to simplify the process of managing medical records by allowing hospitals to input patient information just once, reducing the workload and increasing efficiency. The system provides doctors with easy access to a patient's entire healthcare records on one platform, making it more convenient for them to provide better care. Additionally, the records can be easily converted into encrypted zip files for secure and efficient transfer between hospitals.

The Case Scenario

1. The system utilizes a secure login process, requiring users to provide a valid username and password in order to access the web interface.

2. Administrators will have the ability to input new patient and staff information, create new medical files, and search for existing patients and medical records. Medical staff will have similar capabilities, with the exception of being able to input new staff members. Patients will only be able to view their own medical records.

3. The forms within the system are designed to be user-friendly and self-explanatory, allowing users to easily input and modify data as necessary.

4. Upon completion of any form, users will have the option to submit the data, which will then be sent to the underlying database for storage.

5. The database system is designed to timestamp each entry and automatically assign the data to the appropriate fields for easy retrieval and organization.

6. Once all necessary modifications and reviews have been completed, users can securely log out of the system through the logout button, ensuring the protection of sensitive information.

Terminology

Medical Records
A patient's medical history, clinical findings, diagnostic test results, pre- and postoperative treatment, patient progress, and medication are all explained in detail in their medical records.
If notes are properly documented, they will support the doctor's assessment of the efficacy of the treatment.

Patient
Any person who receives medical treatment from trained experts is referred to as a patient.

Inpatient Care
Patients who must be admitted to a hospital due to their condition receive inpatient care. The development of extensive out-patient clinics and advancements in modern medicine ensure that patients are only admitted to a hospital when they are critically ill or have suffered significant bodily injuries.

Ambulatory/Outpatient Care
Medical care that is given on an outpatient basis is known as ambulatory care, and it includes services for diagnosis, observation, consultation, treatment, intervention, and rehabilitation.
Even when delivered outside of hospitals, this treatment may involve cutting-edge medical technology and procedures.

Patient Bed
A bed that is routinely maintained and staffed for the accommodation and full-time care of a succession of inpatients and is located in wards or a section of the hospital where ongoing medical care for inpatients is provided is defined as a hospital bed. This word is typically used as a placeholder for a patient currently occupying a spot.

Medical Staff
Medical staff personnel are those licensed healthcare workers (physicians, nurses, allied health professionals, and others) who are permitted to provide medical care within a healthcare facility by state legislation and hospital bylaws.

Database
A database is a collection of data that has been organized to make it simple to manage and update. Data records or files containing information, such as sales transactions, customer information, financial data, and product information, are often stored in computer databases.

Front-End
Front-end web development is the process of creating a website's graphical user interface using HTML, CSS, and JavaScript so that visitors can view and interact with it.

Back-End
Working on server-side software, or what you can't see on a website, is what back-end development entails. By concentrating on databases, back-end logic, application programming interfaces (APIs), architecture, and servers, back-end developers make sure the website functions properly.

AWS
A subsidiary of Amazon, Amazon Web Services, Inc. offers metered, pay-as-you-go on-demand cloud computing platforms and APIs to people, businesses, and governments.

Cloud Computing
The on-demand availability of computer system resources, in particular data storage and processing power, without direct active supervision by the user is known as cloud computing.
Functions in large clouds are frequently dispersed over several sites, each of which is a data center.

Technical Stack

Container
Docker

Source Control
Git/GitLab

Database
Amazon RDS for mySQL

Backend
Python Flask

Frontend
HTML5
CSS
Bootstrap

System Requirements (for Docker Desktop; Requires WSL2)

- Windows 11 64-bit: Home or Pro version 21H2 or higher, or Enterprise or Education version 21H2 or higher.
- Windows 10 64-bit: Home or Pro 21H1 (build 19043) or higher, or Enterprise or Education 20H2 (build 19042) or higher.
- Enable the WSL 2 feature on Windows.
    - The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
        - 64-bit processor with Second Level Address Translation (SLAT)
        - 4GB system RAM
        - BIOS-level hardware virtualization support must be enabled in the BIOS settings

Milestones

User Interface Completed - Bell/Khalilov
This milestone refers to the completion of the user interface design and development. This includes the layout, design, and functionality of the pages and forms within the system. This will be a crucial step in ensuring a user-friendly and efficient experience for the end-user.

Backend/Database Completed - Santos/Bell
This milestone refers to the completion of the backend and database development. This includes the creation and implementation of the database schema, as well as the development of the code to interact with the database. This will ensure that the system is able to store and retrieve data efficiently and effectively.

Full Code Connection/Functioning in AWS - Levy/Khalilov/Arriaga
This milestone refers to the full integration of the front-end and back-end code, as well as the successful deployment of the system on the AWS platform. This step will ensure that the system is fully functional and accessible to users through the specified URL.

Testing and Documentation Completed - Gani/Arriaga
This milestone refers to the completion of testing and documentation for the system. This includes the development of test cases and the execution of those tests to ensure that the system is functioning as intended. Additionally, this step will involve the creation of documentation such as user manuals, system diagrams, and code commenting to ensure that the system is easily understandable and maintainable by future developers.

Figure 1: Entity-Relationship Model

Timeline

Phase 1, Project Planning/Database Creation - 2 Weeks

- Jan 24 - Project Overview/Plan Completed, Technologies Chosen, Team Member Roles Assigned
- Jan 31- Login Page and Patient Registration Page Completed, Test Plan Created, Database Tables and Fields Created

Phase 2, User Facing Pages Completed - 2 Weeks

- Feb 7- Employee Record Page Completed
- Feb 14 - Medical File Page Completed

Phase 3, Functioning Application, Testing, and Documentation - 3 Weeks

- Feb 21 - Code Review; Hospital Search Page Completed
- Feb 28 - Final Connection of all pages/front-end/back-end; Testing of data transfer to database
- March 1-6 - Final Testing for all aspects of project; Final Documentation/User Guide Completed
- March 7 - Final Project Completion

Statement of Goals

Our software aims to simplify the process of managing medical records by allowing hospitals to input patient information just once, reducing the workload and increasing efficiency. The system provides doctors with easy access to a patient's entire healthcare records on one platform, making it more convenient for them to provide better care. Additionally, the records can be easily converted into encrypted zip files for secure and efficient transfer between hospitals. Hospital staff can access bed information, patient information, and staff information. Patients are able to view their own medical records and no other information. This application will be hosted on AWS to ensure scalability and reliability.

Functional Description

- Log In - The initial page a user encounters is a login page. The user is prompted to provide a username and associated password to continue further into the site

- Home - After logging in, the user is taken to a home page. From here, the user has access to the full menu of other pages. On this page, the user ID, user role, user name, and user address is displayed.

- New Patient - This page can be accessed via the menu bar on any page after login. It provides a form that allows the user to create a new patient for the system. It includes name, SSN, admission date, insurance, physician name, patient type, patient address, and space to create and verify a new password. This information is stored on the database to later be accessed from the Search Page.

- New Staff - This page can be accessed via the menu bar on any page after login. This page provides a form for a new staff member to be registered into the system. It includes fields for name, SSN, address, qualifications, certification expiration date, phone number, email address, payroll information, personal details, clock in/out times, as well as space to create and verify a new password. This information is stored on the database to later be accessed from the Search Page.

- New Medical File - This page can be accessed via the menu bar on any page after login. This is where patient information will be recorded and stored. It is a simple form that allows the user to input patient information including patient ID, visit date, bed number, admitted date, discharged date, diagnosis, treatment, medications administered, medical notes, and billing amount. This information is stored on the database to later be accessed from the Search Page by either medical staff or that specific patient.

- Search - This page can be accessed via the menu bar on any page after login. This allows for staff to search through staff records as well as patient records. Patient's will be able to search for their own medical records. This searches the database for all information associated with the name of the staff/patient. There is a drop down menu to specify whether to search staff or patient records.

User Interface Design

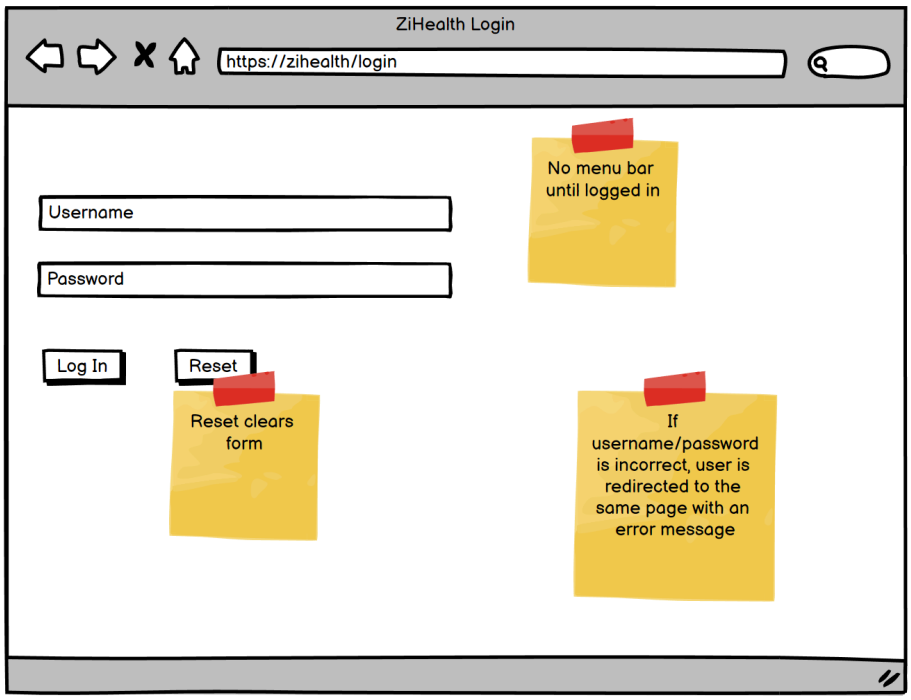


Figure 1: Login Page

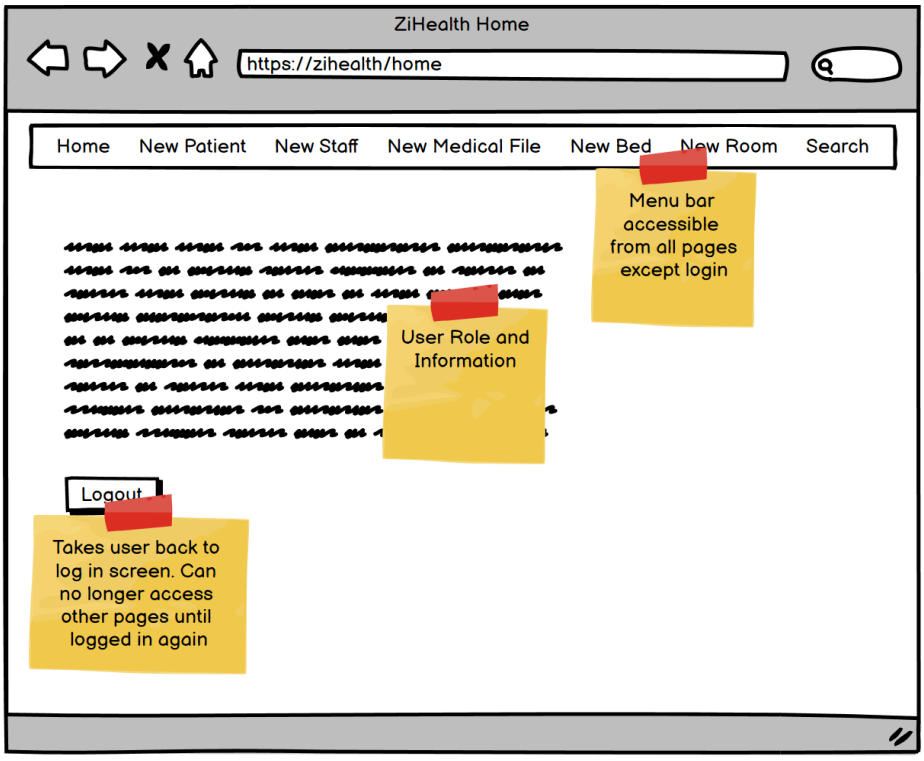


Figure 2: Home Page

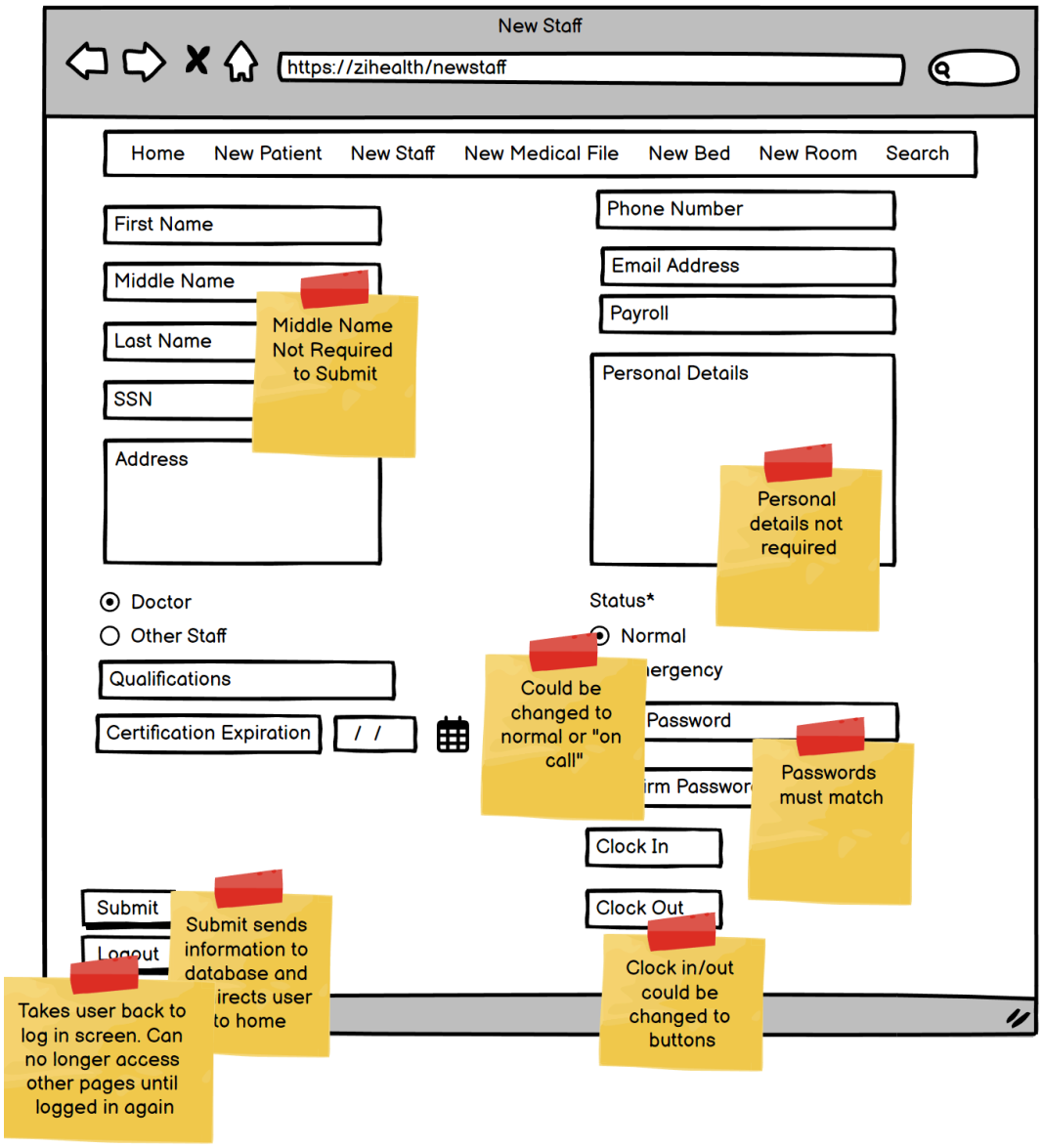


Figure 3: New Staff Page

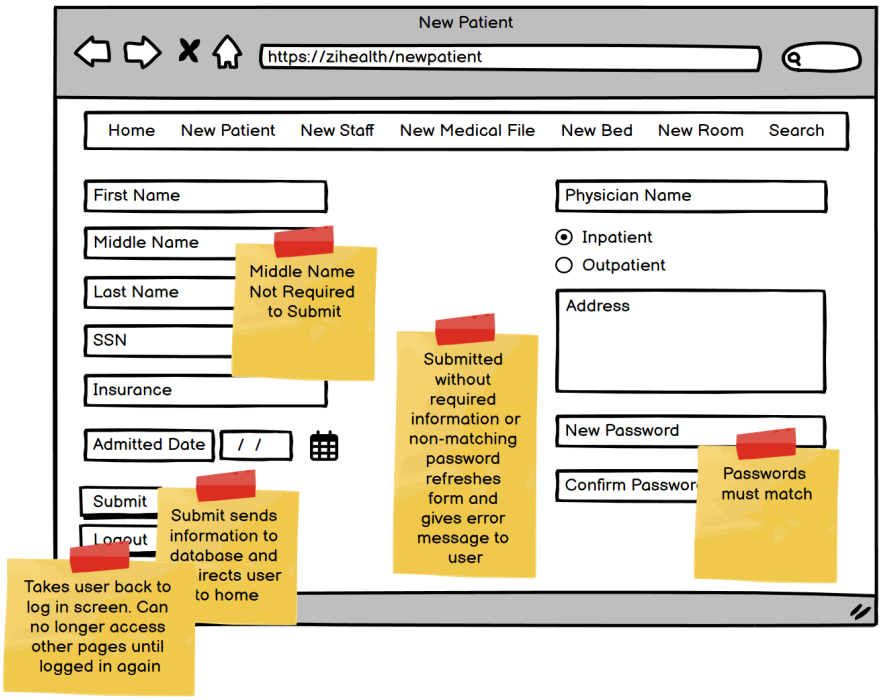


Figure 4: New Patient Page

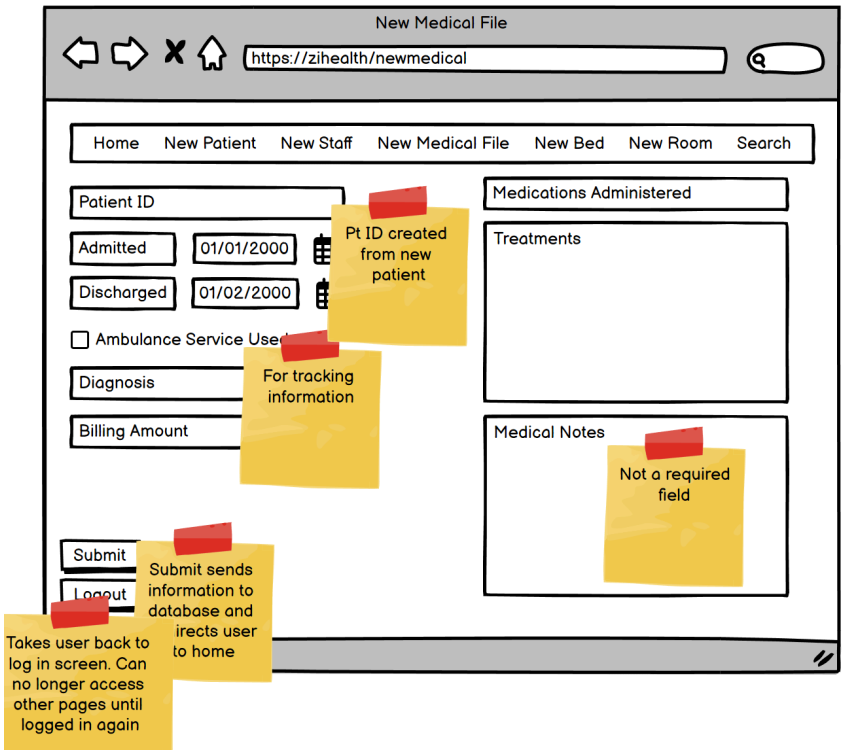


Figure 5: New Medical Record

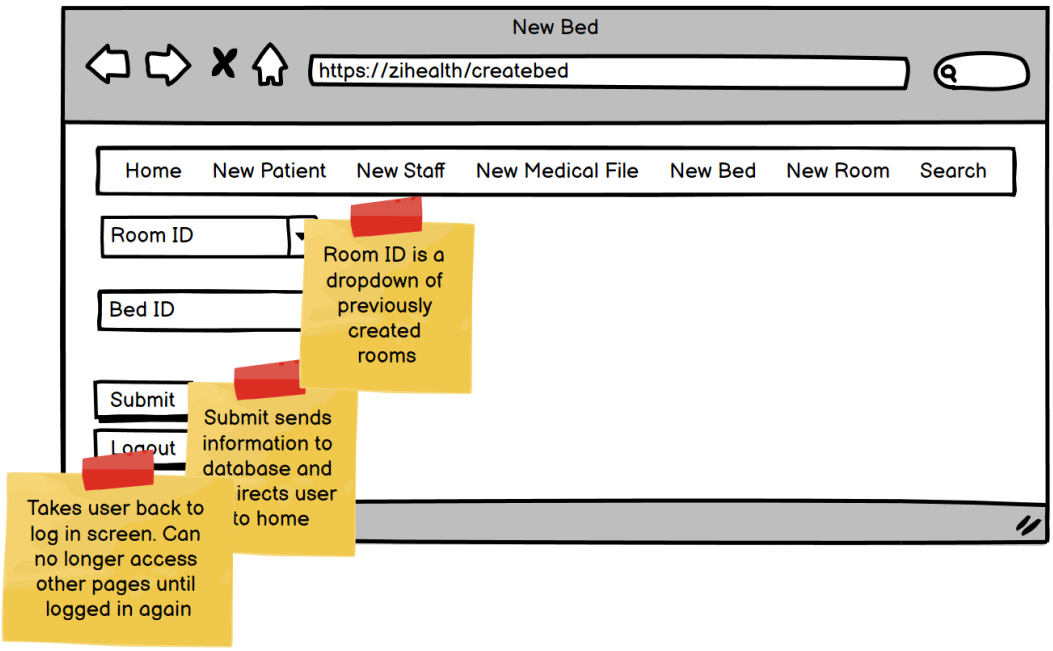


Figure 6: Create Bed

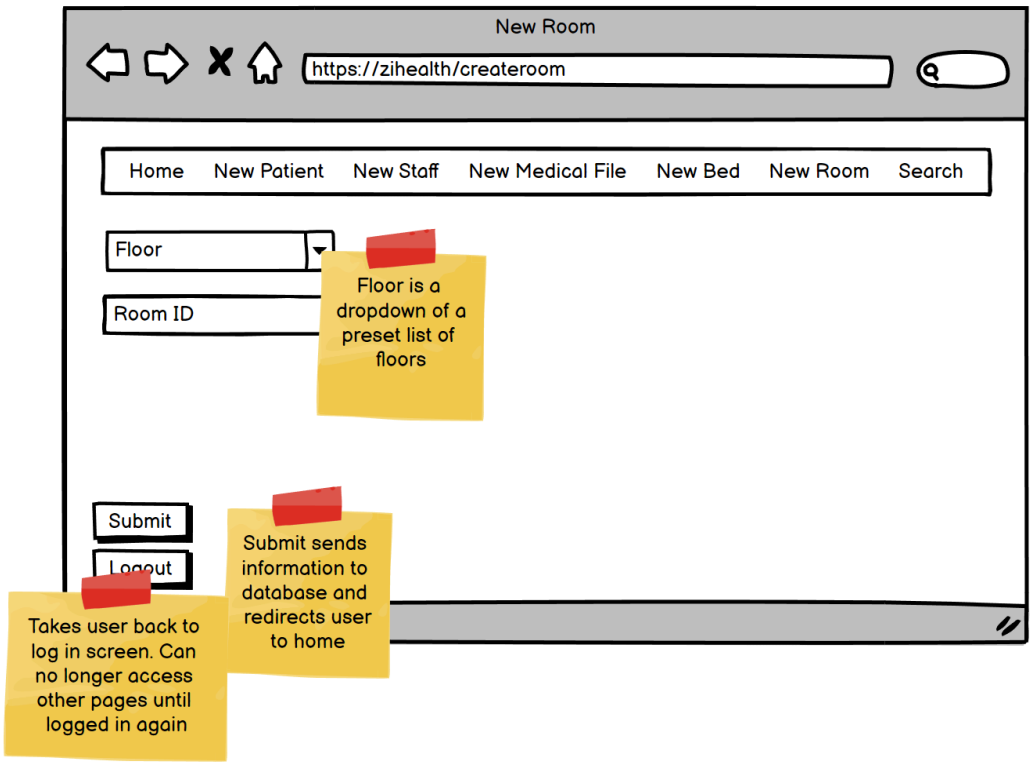


Figure 7: Create Room

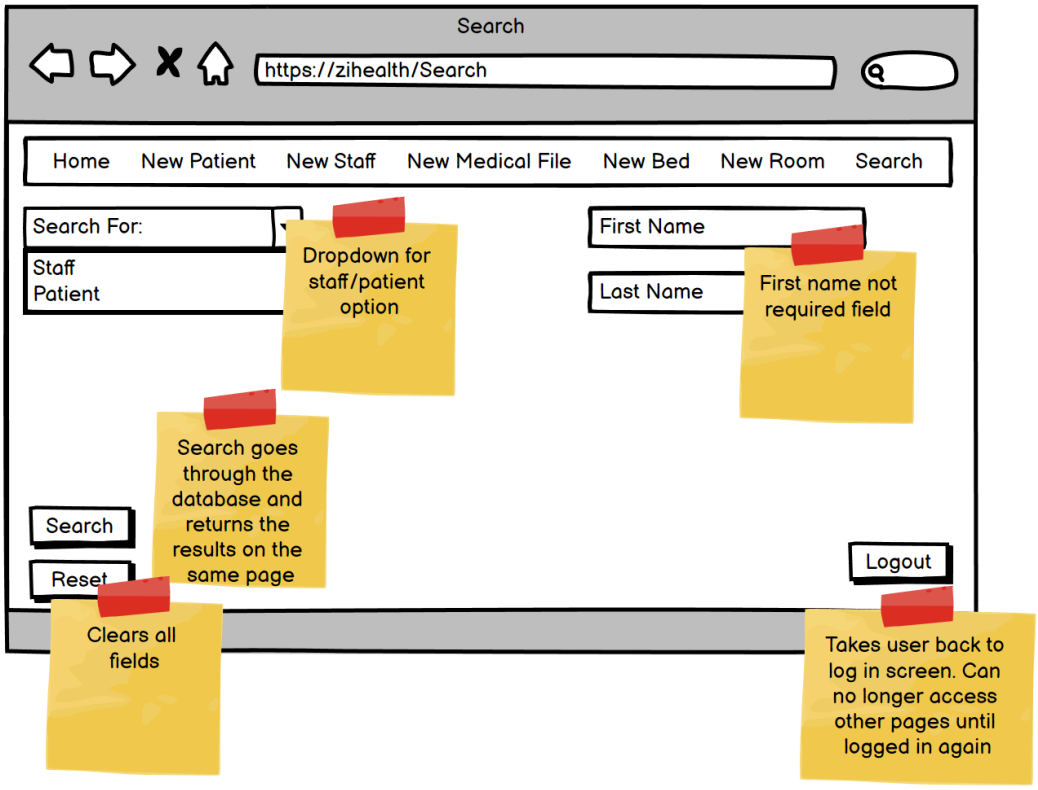


Figure 8: Search Page

Milestones

- Login and Patient Registration Pages Completed - 01/31/2023
- Employee Record Page Completed - 02/07/2023
- Medical File Page Completed - 02/14/2023
- New Bed, New Room, and Search Pages Completed, Alpha Application - 02/17/2023
- All code connected on AWS, Technical Documentation Completed - 02/28/2023
- Testing Completed, Initial Release of Application - 03/07/2023

Test Plan

Login Page
- Attempt to log in with both fields blank
- Attempt to log in with name blank, correct password
- Attempt to log in with a multi-word username, separated by a space, correct password
- Attempt to log in with appropriate name, no password
- Attempt to log in with correct username, incorrect password
- Click Reset button after inputting characters into username and password
- Attempt to log in with correct username, correct password

Home Page
- Verify information matches appropriate user information
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

New Patient
- Attempt to submit a form with no fields filled out
- Attempt submit a form with all required fields, no other fields filled out
- Attempt to submit a form with all fields filled
- Verify in database that the information from the previous three attempts has been saved correctly
- Attempt to submit a form with a SSN that is too long
- Attempt to submit a form with a SSN that is too short
- Attempt to submit a form with a SSN that has additional characters (hyphens) and spaces
- Attempt to submit a form with date as 01-12-1111
- Attempt to submit a form with date as 01121111
- Attempt to submit a form with date as 01 12 1111
- Attempt to submit a form with date as 1/12/1111
- Attempt to submit a form with date as 01/2/1111
- Attempt to submit a form with date as 01/12/11112
- Attempt to submit a form with address on one line
- Attempt to submit a form with address on multiple lines
- Attempt to submit a form with password and password retype different
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button

- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

New Staff

- Attempt to submit a form with no fields filled out
- Attempt to submit a form with 40 characters in the name field
- Attempt submit a form with all required fields, no other fields filled out
- Attempt to submit a form with all fields filled
- Verify in database that the information from the previous three attempts has been saved correctly
- Attempt to submit a form with a SSN that is too long
- Attempt to submit a form with a SSN that is too short
- Attempt to submit a form with a SSN that has additional characters (hyphens) and spaces
- Attempt to submit a form with date as 01-12-1111
- Attempt to submit a form with date as 01121111
- Attempt to submit a form with date as 01 12 1111
- Attempt to submit a form with date as 1/12/1111
- Attempt to submit a form with date as 01/2/1111
- Attempt to submit a form with date as 01/12/11112
- Attempt to submit a form with address on one line
- Attempt to submit a form with address on multiple lines
- Attempt to submit a form with an invalid email address
- Attempt to submit a form with past date in the clock in
- Attempt to submit a form with future date in the clock out
- Attempt to submit a form with password and password retype different
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

New Medical File

- Attempt to submit a form with no fields filled out
- Attempt submit a form with all required fields, no other fields filled out
- Attempt to submit a form with all fields filled
- Attempt to submit a form with date as 01-12-1111
- Attempt to submit a form with date as 01121111
- Attempt to submit a form with date as 01 12 1111
- Attempt to submit a form with date as 1/12/1111
- Attempt to submit a form with date as 01/2/1111

- Attempt to submit a form with date as 01/12/11112
- Attempt to submit with a non-numeric character under billing amount
- Attempt to submit form with non alpha-numeric characters under Disease Name
- Attempt to submit form with 300 characters under medication name
- Verify all things have saved into the database correctly
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

New Bed

- Attempt to submit a form with no fields filled out
- Attempt to submit a form with all fields filled
- Attempt to submit a form with a field with 40 characters
- Attempt to submit a form with a field with a special character
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

New Room

- Attempt to submit a form with no fields filled out
- Attempt to submit a form with all fields filled
- Attempt to submit a form with a field with 40 characters
- Attempt to submit a form with a field with a special character
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Log Out

Search

- Attempt to submit a form with no fields filled out
- Attempt to submit a form with all fields filled

- Attempt to submit a form with a field with 40 characters
- Attempt to submit a form with a field with a special character
- Submit forms using the user drop down
- Submit forms using the medical file drop down
- Click Home button
- Click New Patient button
- Click New Staff button
- Click New Medical File button
- Click New Bed button
- Click New Room button
- Click Search button
- Click Reset

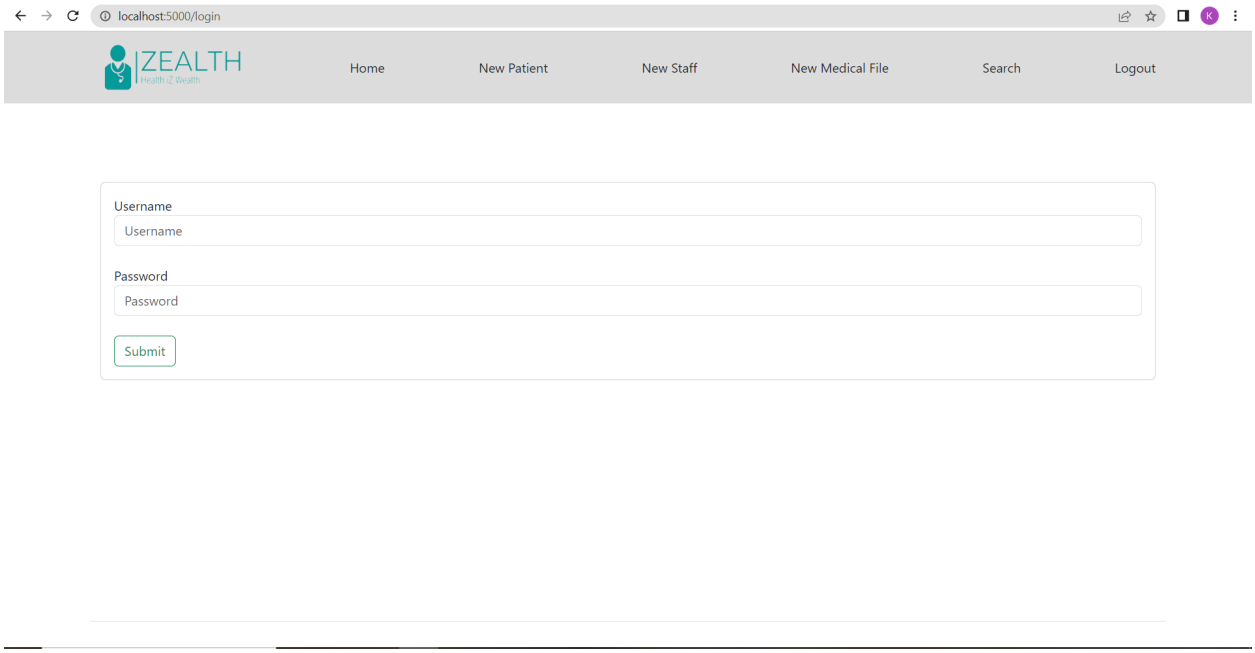
Zealth User Guide

Our software aims to simplify the process of managing medical records by allowing hospitals to input patient information just once, reducing the workload and increasing efficiency. The system provides doctors with easy access to a patient's entire healthcare records on one platform, making it more convenient for them to provide better care. Additionally, the records can be easily converted into encrypted zip files for secure and efficient transfer between hospitals.

The navigation system on each page, with the exception of the Login page, offers a user-friendly and intuitive interface for navigating through the different sections of the system. The top of the page features a set of tabs that are labeled Home, New Patient, New Staff, New Medical File, New Bed, New Room, and Search. Depending on the user's account type, some of these tabs may not be visible to the user. By clicking on one of the tabs, the user is redirected to the corresponding webpage, making it easy for them to access the information or functionality they need quickly and efficiently. This design ensures that users can navigate the system with ease and access the information they need quickly, while also providing a logical and consistent structure to the

Login Page

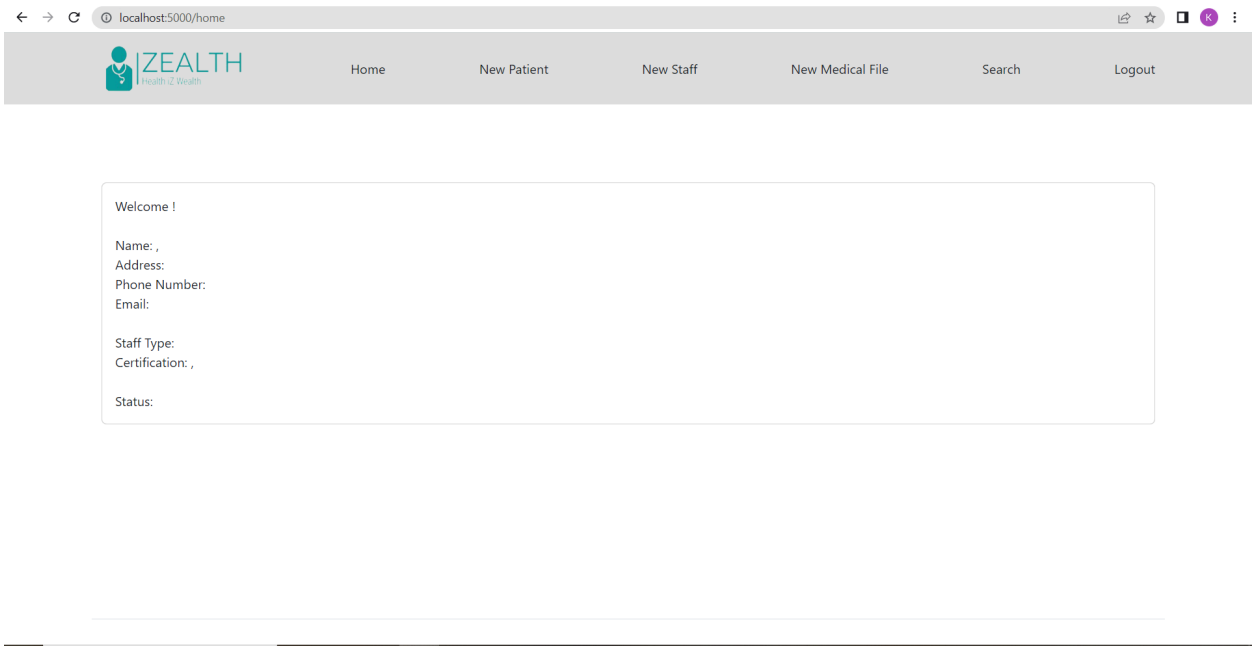
The system's login page features the logo and allows users to enter their username and password for authentication. The text fields for login name and password are labeled clearly, making it easy to understand. There is a "submit" button at the bottom of the page. If either field is left blank, an error is thrown and the user is redirected to the login page. If the authentication fails, the user is sent back to the login page. A successful login redirects the user to the Home page.



Home Page

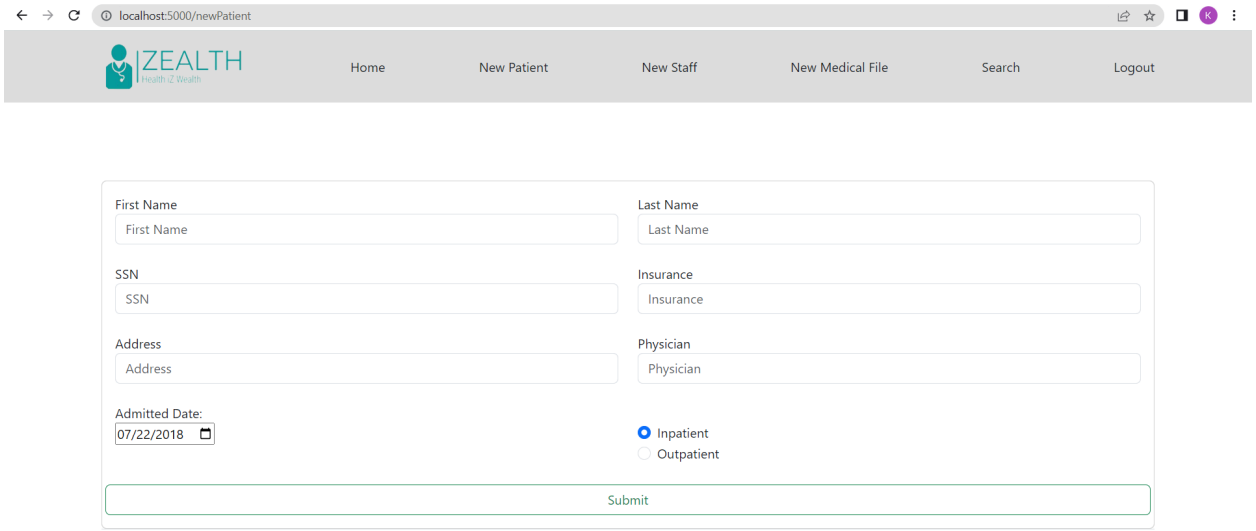
The user may now click on any of the tabs. New patient, new staff, new medical file, search, or logout.

The home page serves as a personalized dashboard for the logged-in user, displaying information specific to their account, such as their name, address, and phone number. It provides a clear overview of their account information and can be accessed quickly.



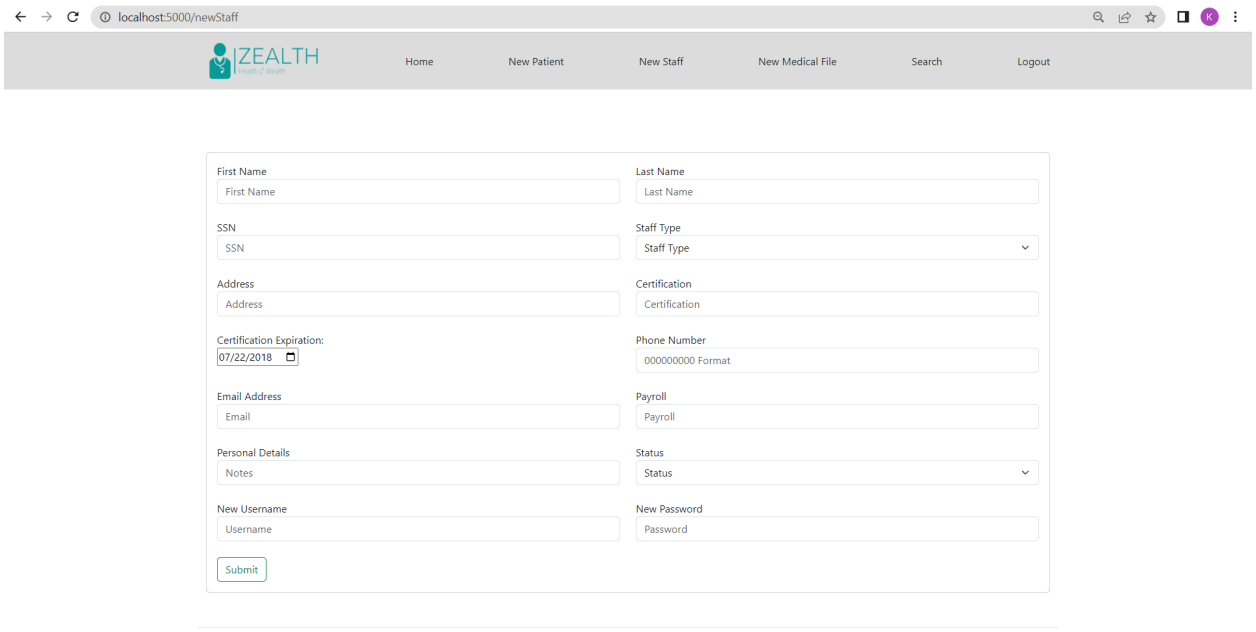
New Patient

If the user selects New Patient they will be directed to the New Patient page. It is a crucial tool for hospital staff to efficiently input and record important information about new patients into the system. The page features a comprehensive form with 8 fields, such as patient name, SSN, admission date, insurance, primary doctor, inpatient/outpatient status, and address.



New Staff

The New Staff page is a platform for adding new employees to the hospital and has fields for personal information such as first and last name with a 30-char max limit per field.



An error message will appear if the character limit is exceeded or if the fields are left blank. The page also features a Staff Type dropdown to specify if the employee is a staff member or a doctor, determining where the information will be stored in the database.

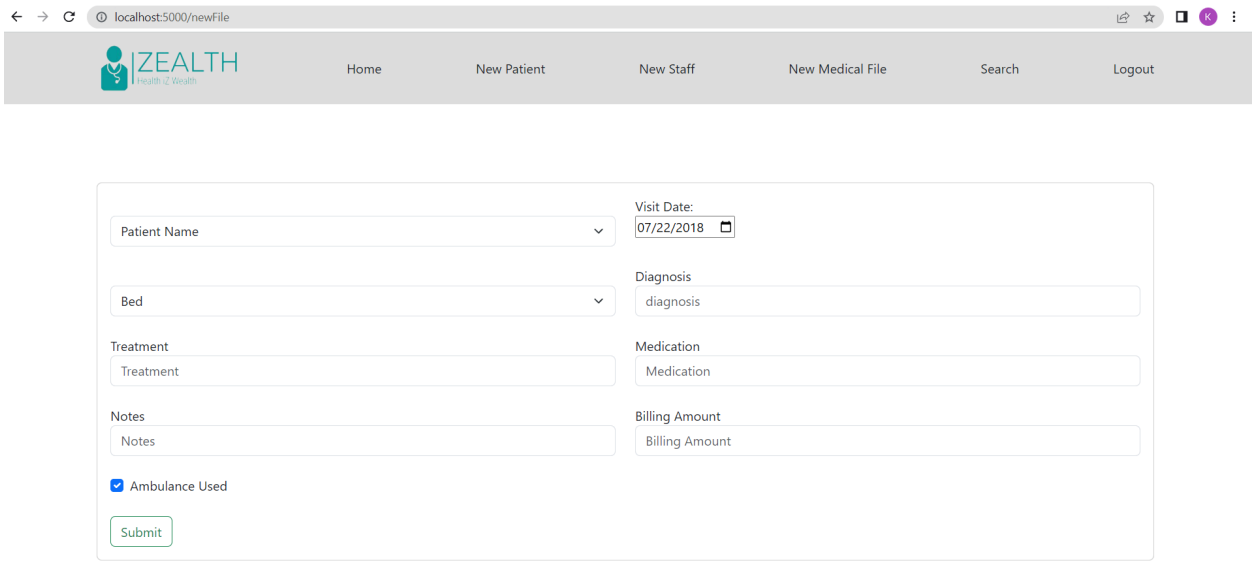
Additionally, the page features a field labeled SSN, which is used to enter the employee's social security number. The format of this field must match the standard format for SSN numbers, and any deviations will result in an error message being displayed to the user.

The page also includes a field labeled Certifications, which is used to enter information about the employee's education and certifications. This field has a maximum character limit of 500 alpha/numeric characters, and any special characters or exceeding of the character limit will result in an error message being displayed.

Other fields on the page include a field labeled Cell Number, which is used to enter the employee's phone number in the format DDDDDDDDDD with no spaces, dashes, or parentheses. An error message will be displayed if the format is not followed. Additionally, the page includes a field labeled Email Address, which is used to enter the employee's email address. This field has a maximum character limit of 50 and any special characters or absence of a " @ " symbol or a ".com/.edu/.net/.gov" at the end of the field will result in an error message.

New Medical File

The medical files page is intended for use by both patients and medical personnel. Patients can only read their own records, while medical personnel can edit and update the information on the page.



The next section, "Date of Visit," is used to record the first day the patient arrived at the hospital for treatment. This field's format must be MM/DD/YYYY, and any other format will result in an error.

The following field contains a checkbox titled "Ambulance Service Used." This field is mostly used for billing and other statistics.

The "Bed Name" field must match the name of one of the hospital's available beds.

The “Diagnosis" section is used for diagnosis and permits alpha/numeric characters up to 200 characters. If the field contains more than 200 characters or special characters, the user will see an error message.

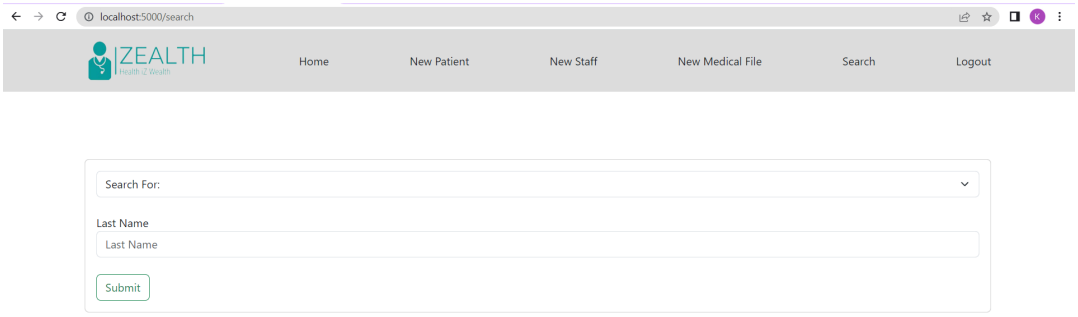
The "Medication" text area can hold up to 200 alpha/numeric characters.
If special characters are entered, an error notice will appear informing the user that this field contains unlawful characters.

The Notes area is a text box where the medical team can put any additional information on the patient's condition or treatment. This text box can accept up to 1000 alpha/numeric characters and will only throw an error if one of these conditions is not met.

The final field on the medical files page is labeled Submit, and when clicked, it will save all of the information submitted into the patient's medical record.

Search

The Search page is a powerful tool that allows users to quickly and easily locate specific individuals or medical files within the hospital's system. The page is designed to be user-friendly and intuitive, making it simple for both patients and medical staff to navigate.

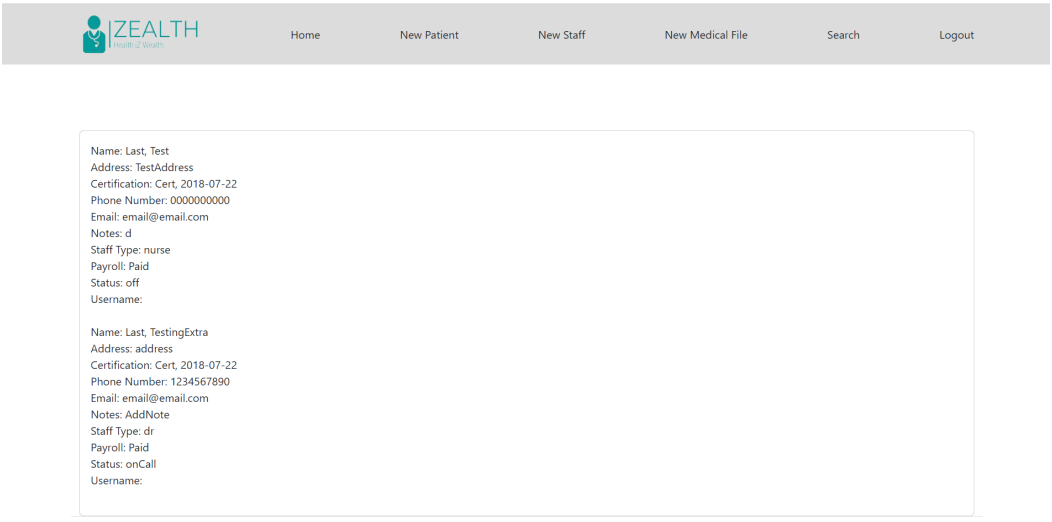


The first field on the page, labeled "Search for", is a drop-down box that allows users to specify whether they are searching for a user or a medical file. This ensures that the search results are relevant and accurate.

The field, labeled "Last Name", is used to input the name of the individual or file being searched for. These fields are designed to follow the same rules as the name fields in previous sections, and will only accept alphabetical characters.

At the bottom of the page there is a "Submit" button. This submits the information entered into the fields above and checks for any errors. If no errors are found, the program will search the database for the requested information and display the results.

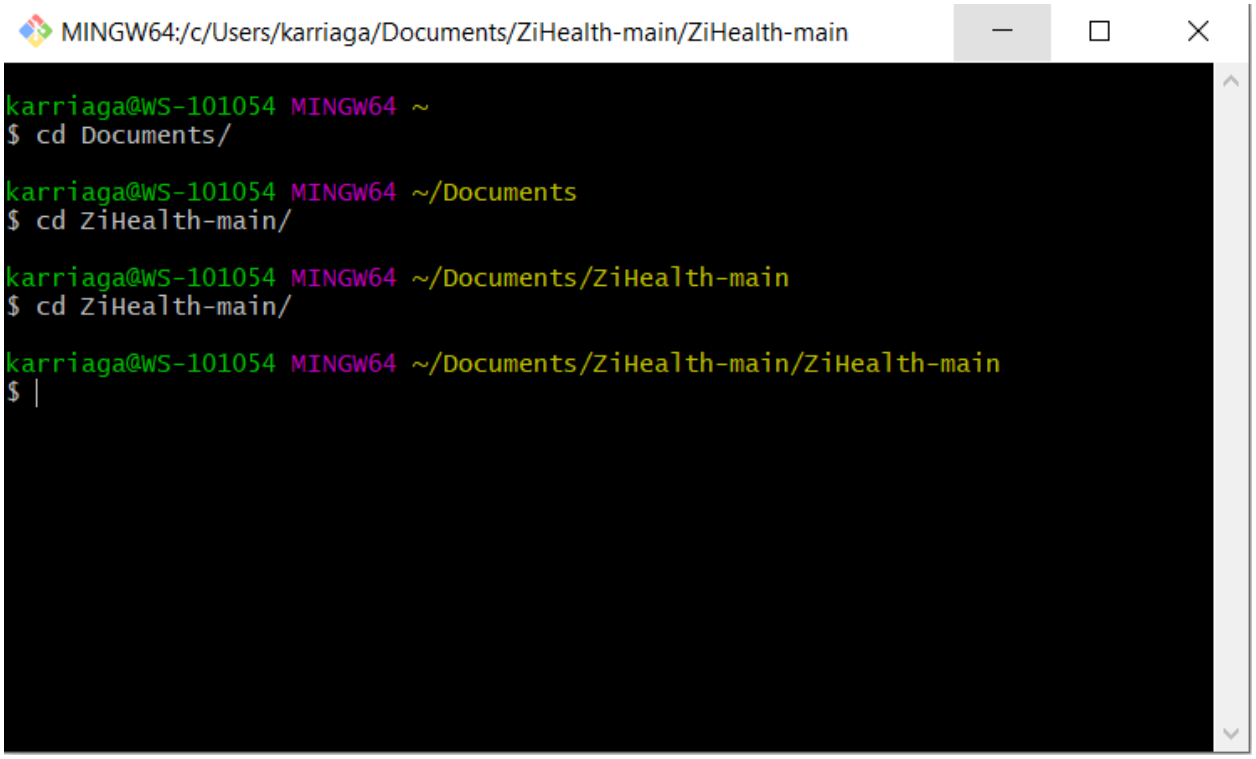
Once the search is complete, the system will display the relevant files or individuals that match the search criteria, making it easy for users to find the information they need.



Zealth Quick Launch Guide

This application is containerized in Docker. We have chosen to use GitBash to work with to start our Docker container and to use for this guide, however there are other methods for building and running the application. The container is connected to our Amazon RDS, so *all data is saved in the cloud*, not locally, so be aware of what information you place in the application when testing.

Required:
GitBash https://gitforwindows.org/
Docker Desktop https://www.docker.com/products/docker-desktop/



Use GitBash to navigate into the application folder. In this example, ZiHealth is located inside the Documents folder. Using the command ***cd [foldername]*** allows the user to enter into the filesystem.

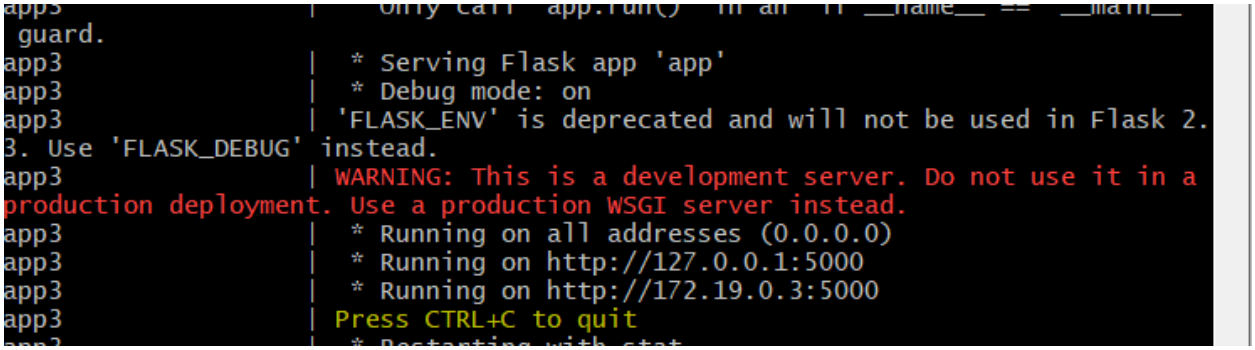
```
karriaga@WS-101054 MINGW64 ~/Documents/ZiHealth-main/ZiHealth-main
$ docker compose build
#1 [internal] load build definition from dockerfile
#1 transferring dockerfile: 257B done
#1 DONE 0.0s

#2 [internal] load .dockerignore
#2 transferring context: 2B done
#2 DONE 0.0s

#3 [internal] load metadata for docker.io/library/python:3.10.2
#3 DONE 0.6s

#4 [1/7] FROM docker.io/library/python:3.10.2@sha256:3204faabc2f0b5e0939bdb8b290
79a2a330c38dee92a22482a9ed449c5649a55
#4 resolve docker.io/library/python:3.10.2@sha256:3204faabc2f0b5e0939bdb8b29079a
2a330c38dee92a22482a9ed449c5649a55 done
#4 DONE 0.0s
```

```
04064350d
#12 writing image sha256:1f519629902c7c5e4f9de0d857158313dd979a882c44a2b4effd4ca
04064350d done
#12 naming to docker.io/library/zihealth-main-app done
#12 DONE 0.0s

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and l
earn how to fix them

karriaga@WS-101054 MINGW64 ~/Documents/ZiHealth-main/ZiHealth-main
$
```

From inside the folder, run the command ***docker compose build***

```
karriaga@WS-101054 MINGW64 ~/Documents/SchoolStuff/ZiHealth (main)
$ docker compose up
time="2023-03-07T17:41:49-05:00" level=warning msg="Found orphan containers ([ki
bana3 elastic3]) for this project. If you removed or renamed this service in you
r compose file, you can run this command with the --remove-orphans flag to clean
 it up."
Container zihealth-adminer-1  Created
Container zihealth-mysql-1  Created
Container app3  Created
Attaching to app3, zihealth-adminer-1, zihealth-mysql-1
zihealth-mysql-1    | 2023-03-07 22:41:50+00:00 [Note] [Entrypoint]: Entrypoint
script for MySQL Server 5.7.41-1.el7 started.
zihealth-mysql-1    | 2023-03-07 22:41:50+00:00 [Note] [Entrypoint]: Switching t
o dedicated user 'mysql'
zihealth-mysql-1    | 2023-03-07 22:41:50+00:00 [Note] [Entrypoint]: Entrypoint
script for MySQL Server 5.7.41-1.el7 started.
zihealth-mysql-1    | '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.soc
k'
zihealth-mysql-1    | 2023-03-07T22:41:50.818909Z 0 [Warning] TIMESTAMP with imp
licit DEFAULT value is deprecated. Please use --explicit_defaults_for_timestamp
server option (see documentation for more details).
```

From the same location, run the command ***docker compose up***



When you see this appear in the logs, the application is running. To access the page, navigate to localhost:5000 on the browser.

Username: UniqueAdmin
Password: password

As another reminder, *data is saved in the cloud*, so our team has the ability to see all data placed into these forms. *Please do not use your personal information*, like SSN, when testing this application.

When you are done using the application, press ***CTRL+C*** on GitBash to quit the application.

Test Results

Login Functions

Attempt to navigate to home page without logging in	Unable to access home page without logging in, redirects user to login page	RESULT AS EXPECTED
Attempt to navigate to search page without logging in	Unable to access search page without logging in, redirects user to login page	RESULT AS EXPECTED
Attempt to login with no password	Unable to submit the form, a notification appears above the password bar	RESULT AS EXPECTED
Attempt to login with a correct username and incorrect password	Login page reloads with an error message of a non-matching username and password	RESULT AS EXPECTED
Login with a correct username and password	User is redirected to the home page	RESULT AS EXPECTED
After creating a new staff member, attempt to log in using those credentials	User is redirected to the home page, all new staff member information is appropriately displayed, not including the SSN	RESULT AS EXPECTED

Logout Function

Logout from home page	Session is ended, user is redirected to the login page	RESULT AS EXPECTED
Attempt to re-enter home page without logging back in	Unable to access home page without logging in, redirects user to login page	RESULT AS EXPECTED

New Staff

Attempt to post the form without the first name filled out	Unable to submit the form, a notification appears above the name bar	RESULT AS EXPECTED
Attempt to set the	Unable to submit the form,	RESULT AS EXPECTED

certification expiration to 2050	a notification appears above the date bar	
Attempt to submit the form with a phone number that does not match the 0000000000 format	Form is reloaded, an error message appears that states a phone number must be in a 0000000000 format	RESULT AS EXPECTED - Used number “123-”
Attempt to set a username that already exists	Form is reloaded, an error message appears that states username already exists	RESULT AS EXPECTED
Submit a form with the status drop down set to “On-Call”	Form is submitted correctly, can be verified by logging in as that staff member and checking the home page states “On-Call”	RESULT AS EXPECTED

Home Page

Navigate to home page after logging in using New Staff created credentials	Home page loads correctly, displaying the staff information that was initially entered into the New Staff Page	RESULT AS EXPECTED
Navigate to New Patient Page using navigation bar	New Patient Page Opens	RESULT AS EXPECTED
Navigate to New Staff Page using navigation bar	New Staff Page Opens	RESULT AS EXPECTED
Navigate to New Medical File Page using navigation bar	New Medical File Page Opens	RESULT AS EXPECTED
Navigate to Search Page using navigation bar	Search Page Opens	RESULT AS EXPECTED

New Patient

Attempt to submit a form with no address	Unable to submit the form, a notification appears above the address bar	RESULT AS EXPECTED

Attempt to submit a form with the SSN in a format other than 000000000	Form is reloaded, an error message appears that states a phone number must be in a 000000000 format	RESULT AS EXPECTED- Used number “4333”
Submit a form with the radio box set to Inpatient	Form is submitted correctly, can be verified by searching for a patient with that last name and noting the “Inpatient” wording on their search result	RESULT AS EXPECTED

New File

Attempt to fill out the form with no previous “New Patients” added	No names appear in the drop down menu, form is unable to be submitted	RESULT AS EXPECTED
Attempt to submit form with no bed chosen	Unable to submit the form, a notification appears above the bed bar	RESULT AS EXPECTED
Attempt to submit a non numerical billing amount	Form is reloaded, an error message appears that states a billing amount must be numerical	FAILED, CODE DOES NOT APPROPRIATELY CHECK FOR NUMERICAL AMOUNT
Submit form with “Ambulance Used” checked	Form is submitted correctly, can be verified by searching for a file with that last name and noting the “Ambulance Used” is noted to be True on their search result	RESULT AS EXPECTED

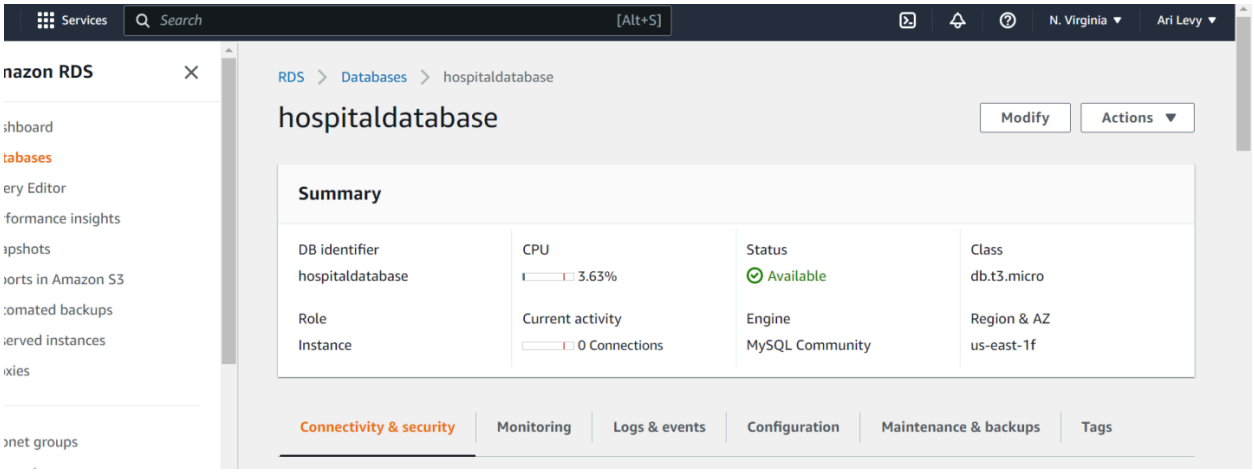
Search

Attempt to submit form with no “Search For” dropdown chosen	Unable to submit the form, a notification appears above the bed bar	RESULT AS EXPECTED
Attempt to search through staff with a correct last name	Form is submitted correctly, view page loads with staff information for any staff member with that	RESULT AS EXPECTED

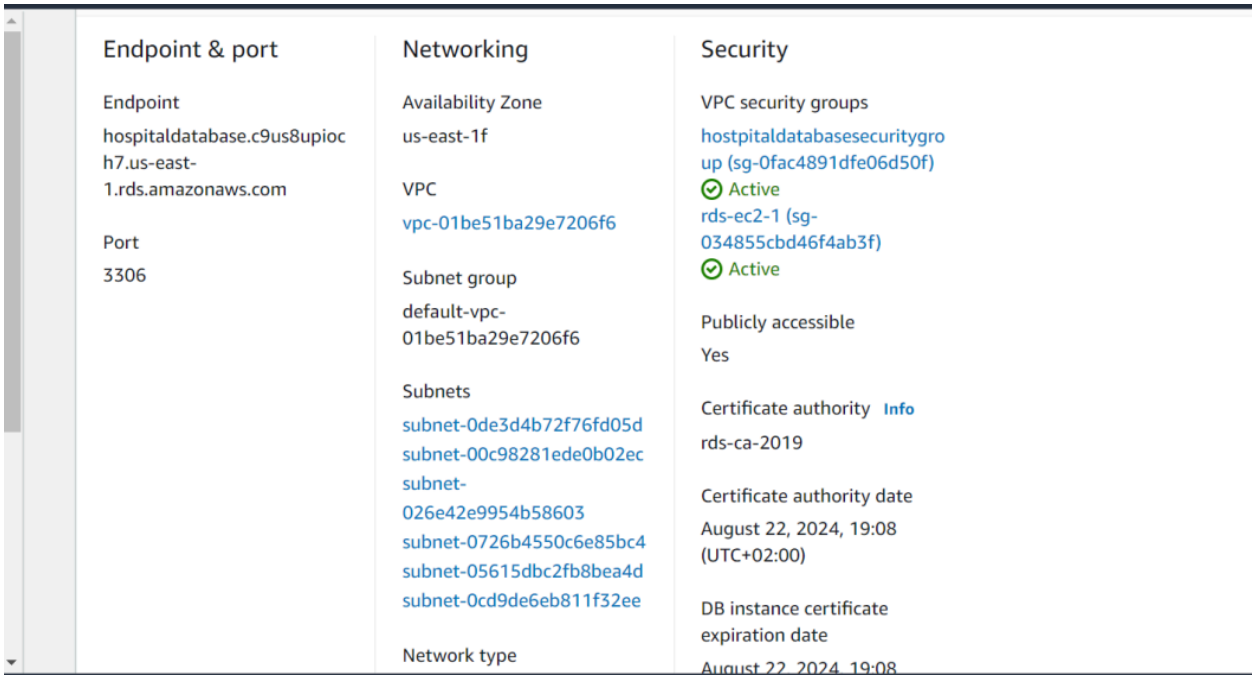
	last name, excluding SSN	
Attempt to search through patients with a correct last name	Form is submitted correctly, view page loads with patient information for any patient with that last name	RESULT AS EXPECTED
Attempt to search through medical files with a correct last name	Form is submitted correctly, view page loads with file information for any patient with that last name	RESULT AS EXPECTED
Attempt to search through patients with an incorrect name	Form is submitted correctly, message of “No patients found” appears	RESULT AS EXPECTED
Attempt to search through medical files with a correct last name that has more than one medical file associated with it	Form is submitted correctly, view page loads with file information for any patient with that last name. Multiple sets of files appear.	RESULT AS EXPECTED

AWS Endpoint for RDS

Hospitaldatabase.c9us8upioch7.us-east-1.rds.amazonaws.com



Amazon RDS - Location of Database



AWS Endpoint

Appendix

Contribution Report

Week 3:

- User Guide - Santos, Levy, Arriaga
- AWS Database Setup - Levy, Santos
- Test Plan - Arriaga
- Final Documentation Draft - Gani
- Backend design, connection with database - Gani, Bell
- Frontend/UI design - Khalilov

Week 4:

- Project Design Documentation Deliverable - Arriaga
- Meeting for AWS Architecture Design - Levy, Arriaga
- AWS Database Setup Completion/MySQL Workbench connected to AWS - Levy, Santos, Bell
- Edits and Approval of Project Design - Khalilov
- Continued Work on Final Project Documentation (Addition of Module Description, Navigating Zealth, and Database Design) - Gani
- Edits to UI interface/submission forms - Khalilov
- Change from manually typing IDs to drop down menus on UI - Khalilov
- Backend code review/testing - Bell

Week 5:

- Testing of html files (in S3 bucket) and lambda functions for AWS research - Ari
- Adjusted backed .java to lambda functions for use of AWS - Ari
- Adding user guide to final documentation - Gani
- Initial front end work; home and search page html draft - Bell
- Database edits to connect to lambda function - Santos
- AWS Research - Ari, Santos, Arriaga
- Code Review - Khalilov
- Documentation Review - Khalilov
- Phase I Documentation - Arriaga

Week 6:

- Database completed, tables created - Santos
- Backend code connected to the database in AWS - Bell, Ari
- Phase II Documentation - Arriaga
- Code Review - Khalilov
- Documentation Review - Khalilov
- Changes to front end to utilize bootstrap - Arriaga
- Addition of Project Aim and Case Scenario to Documentation - Gani
- Initial Discussion of Previous Peer Review and Possible Changes - All Team Members

Week 7:

- Changes to AWS to accommodate Python instead of a Java backend - Santos, Ari
- Start of rewriting backend to Python - Bell, Arriaga
- Phase III Documentation - Gani
- Code and Documentation Review - Khalilov

Week 8:

- Completion of Testing Portion - Arriaga
- Completion of backend to Python - Arriaga, Bell
- Completion of Docker Implementation - Arriaga, Bell
- Front end and Back end integration - Arriaga
- Final polishing of Documentation/Documentation Review - Ari, Santos, Arriaga
- Final Discussion as team member/implementation of any peer review requests - All team members

Weekly Report - Week 7

Original Milestone - Final Connection of all pages/front-end/back-end; Testing of data transfer to database

Schedule - We are currently behind schedule. We have the backend connected to the database, however there is no frontend connection.

Special Problems Encountered - Due to not having our code in any sort of source control, the original frontend files were destroyed with no ability to recover them. The entire frontend needed to be recoded and there was difficulty connecting the new code to the previously coded backend due to different team members working on this.

Changes - There are no major documentation changes. A few minor grammatical issues were corrected. However, all frontend code is different. The backend may be changed and re-coded in Python due to our current trouble. Most group members are most familiar with Python web applications, so a Python backend would be easier to connect to a new frontend as well as easier to do any troubleshooting during the testing phase.

Project Direction - Due to our massive project changes, we have an updated timeline. The new backend and frontend should be completed by Thursday, leaving the connection to the database and AWS to be completed by Saturday. This leaves Sunday and Monday for testing and minor edits, and Tuesday for final documentation and presentation changes.

Weekly Report - Week 8

Original Milestone - Completion of Testing Portion, Backend to Python, Docker Implementation, Frontend and Backend Integration, Final Polishing of Documentation and Documentation Review, and Final Discussion as Team Members

Schedule - This week, the team has made significant progress towards completing all of the original milestones. We are on track with our schedule and optimistic about the completion of the project on time.

Special Problems Encountered - No significant problems encountered

Project Direction - Overall, this week was highly productive, and we are pleased with the progress made. We will continue to work collaboratively and communicate effectively to ensure that the project is delivered on time and meets all requirements.

Lessons Learned:

Creating a hospital website that allows for the storage of patient information requires effective communication, technical skills, sound infrastructure, and adaptability in the face of unexpected challenges.

During the project, the team faced a range of challenges, including diverse technical backgrounds, different timelines, experiences, and strengths, the loss of front-end code, and the need to store HIPAA-compliant patient information securely. Despite these challenges, the team was able to work effectively by recognizing and valuing the individuality of each team member.

The team's diverse technical backgrounds meant that each team member brought different experiences and strengths to the project. This allowed them to collaborate effectively and leverage each other's skills to achieve their goals. Additionally, the team's different timelines required effective time management and communication to ensure that everyone was working towards a common goal.

The loss of front-end code also highlighted the need for individual technical skills and adaptability. The team demonstrated their ability to work flexibly by switching to Python instead of their original plan, and by utilizing AWS to store all patient information securely.

Ultimately, the project taught the team the importance of clear communication, individual technical skills, sound infrastructure, and adaptability in the face of unexpected challenges. By recognizing and valuing the individuality of each team member, the team was able to work effectively and deliver a high-quality solution that met their clients' needs while ensuring the safety and security of sensitive patient information.

Possible Improvements

The next logical step for this project would be to improve the functionality to allow for patients to view their own information and files. Currently, this application is only for staff members to view patient files. It would be beneficial for all patients to be able to search for and view their own health information.

The next improvement we would start on would be the connection between files and patients. Initially, the goal in the project was to have individual files linked to patients so a patient could have multiple associated files, however we did not have the time needed to implement this. Additionally, we would have a set list of beds to pull from to attach patients to so there could be tracking for available and unavailable beds.

Finally, while the database is on the AWS Cloud at this time, it would be part of the project's next steps to place the entire application on the cloud for security and scalability.

For code specific improvements, there is code duplication that could be gotten rid of. The connection to the database happens multiple times and spans too many lines. The code would also benefit from some sort of architecture, like MVC. It would be beneficial to separate out the controllers that speak to the database from the routes that speak to the frontend of the code to keep organization and space between pieces of code. This would allow for better maintainability in the application.

Approach

Our initial approach to this project was as individual team members that had specific and separated roles. We had a person set to code each chunk, a person for AWS, documentation, and to be team leader. The initial members of the group decided on a project and language (java) and members individually did their own work. However, this did not work out long term, and we had to pivot last minute to come up with a functional project. After the loss of the front-end, we realized that although a functional database with the backend written in java, fully hosted on the cloud was our goal, it was not feasible for our current skillset and amount of time. We adjusted to a language that was easier for more people to work through, Python. We also decided to just containerize the application and keep only the database in the cloud. This allowed us to keep the scalability of the database while working locally for the application.

While we were still working under the idea of the entire application being placed on the cloud, we had multiple team members researching and learning the best ways to utilize AWS without massive fees. While this was important for our project at the time, it ended up being very time consuming and confusing for members of the team who had never interacted with AWS. We attempted to take a pre-made application and force it into AWS instead of building an application around the services we wanted to use. While the approach we used does work since AWS is built for all kinds of applications to be migrated, those services often come with a cost and are not “free-tier,” which was an unexpected hurdle. A much better approach, since we were starting from a clean slate application-wise, would have been to choose our services and tailor our application to them.