TASK 1B

Callum Macdonald

Contents

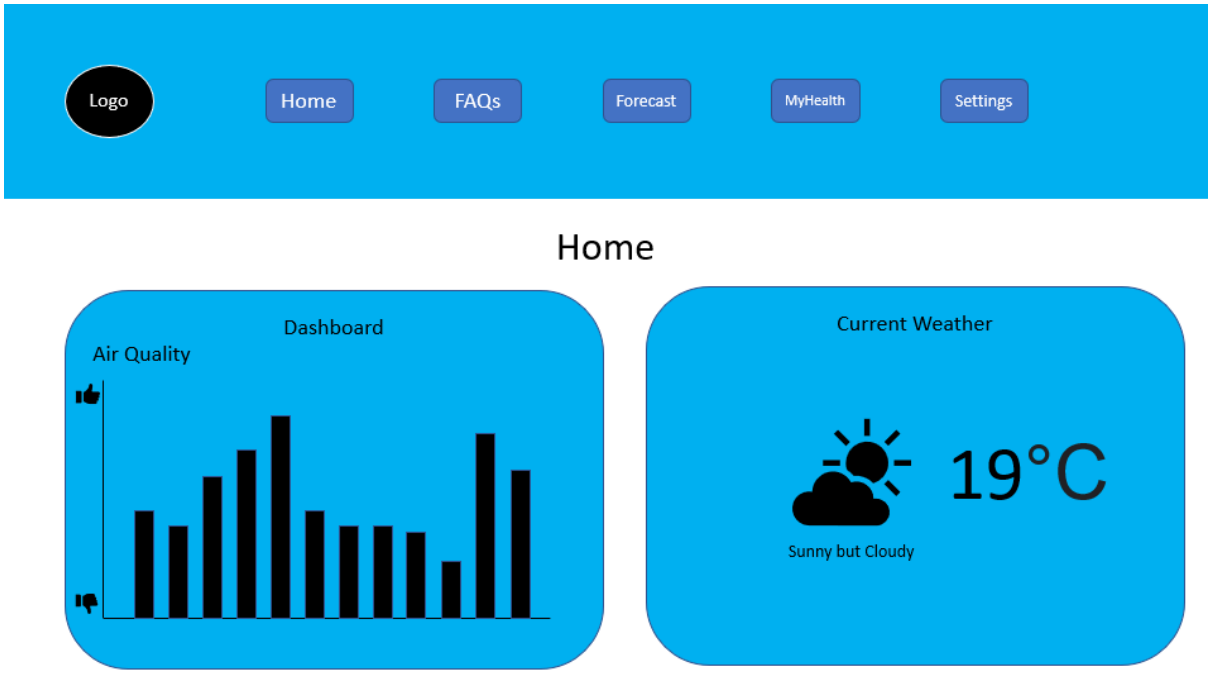
Task 1

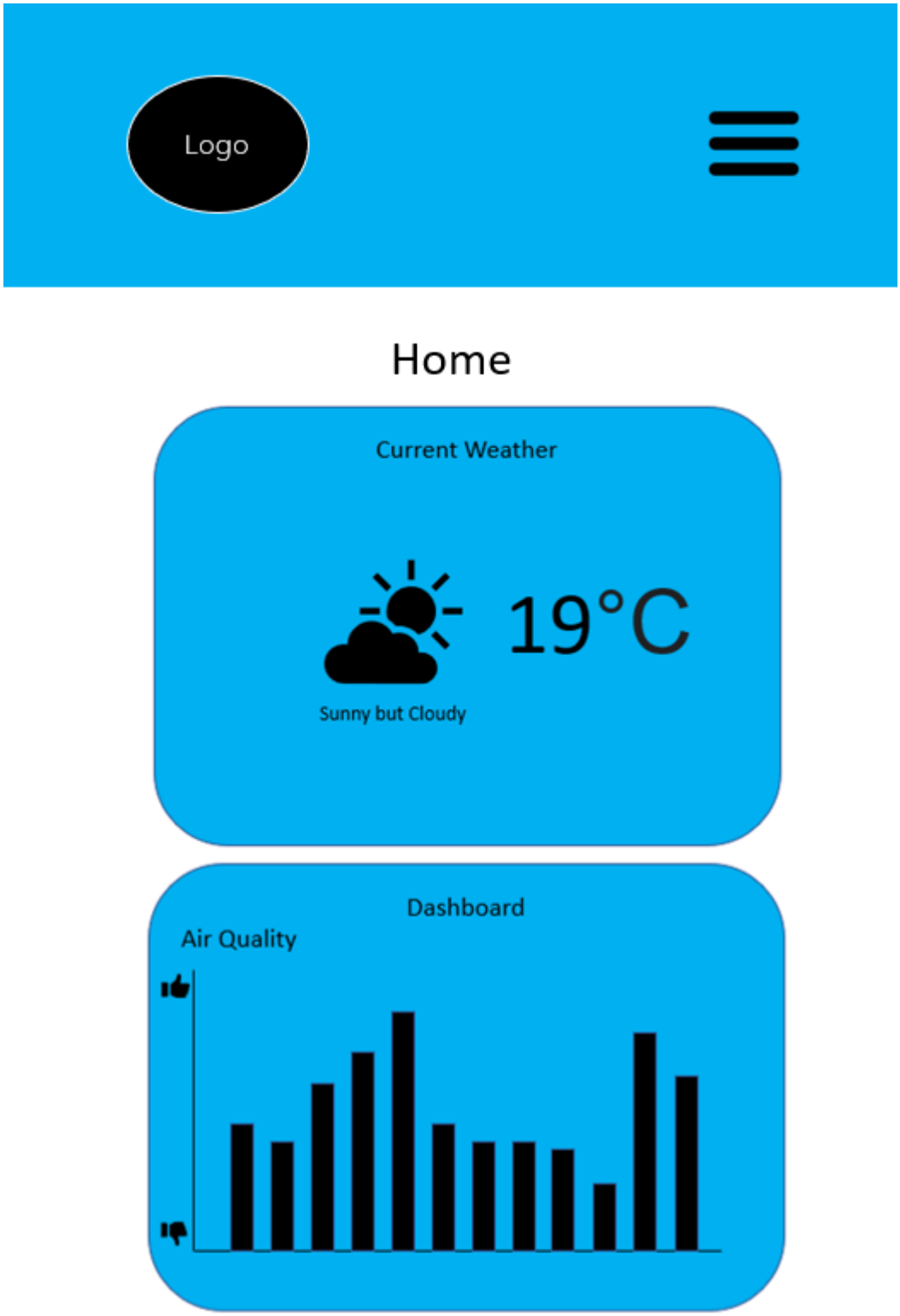
Activity B – Design Documents

Visual / Interface Designs

A description of all visual / interface designs can be found in Task 1 A (ii) within my proposal. Any section with more than one image means it includes both the design for a desktop and mobile interface. If not specified otherwise, assume one image for a section is the same design for both. This will only be the case for 3 pages. These pages being the login page, register page, and mobile navigation. The login page and register page will remain the same for both desktop and mobile users. Mobile navigation is self-explanatory.

Home Page


Logo
Home
FAQs
Forecast
MyHealth
Settings
Home
Dashboard
Air Quality
Current Weather
19°C
Sunny but Cloudy



Frequently-Asked-Questions (FAQs) Page

FAQs

Question:
How do I deal with extreme weather temperatures?

Answer:
Lorem Ipsum

Question:
Can I have information on environmental health conditions and seasonal allergies?

Answer:
Lorem Ipsum

Question:
Can I have a risk assessment for home environments?

Answer:
Lorem Ipsum

Question:
Lorem Ipsum

Answer:
Lorem Ipsum

Question:
Lorem Ipsum

Answer:
Lorem Ipsum

Question:
Lorem Ipsum

Answer:
Lorem Ipsum

FAQs

Question:
How do I deal with extreme weather temperatures?

Answer:
Lorem Ipsum

Question:
Can I have information on environmental health conditions and seasonal allergies?

Answer:
Lorem Ipsum

Question:
Can I have a risk assessment for home environments?

Answer:
Lorem Ipsum

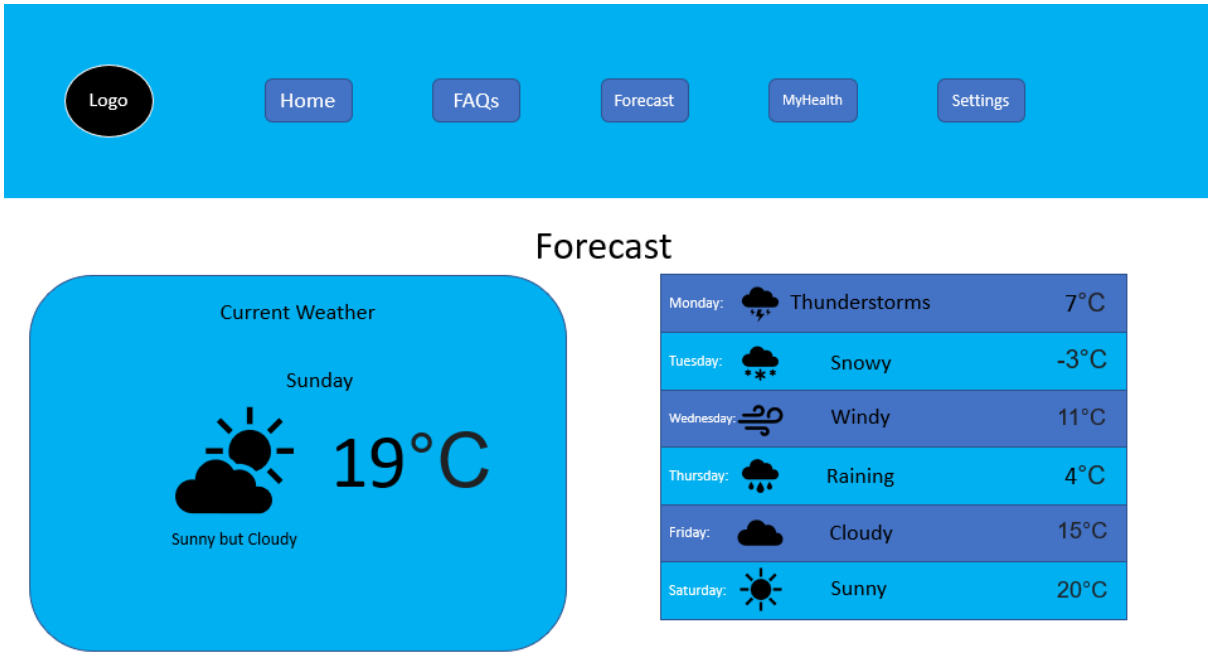
Question:
Lorem Ipsum

Answer:
Lorem Ipsum

Question:
Lorem Ipsum

Answer:
Lorem Ipsum

Forecast Page



Logo

Forecast

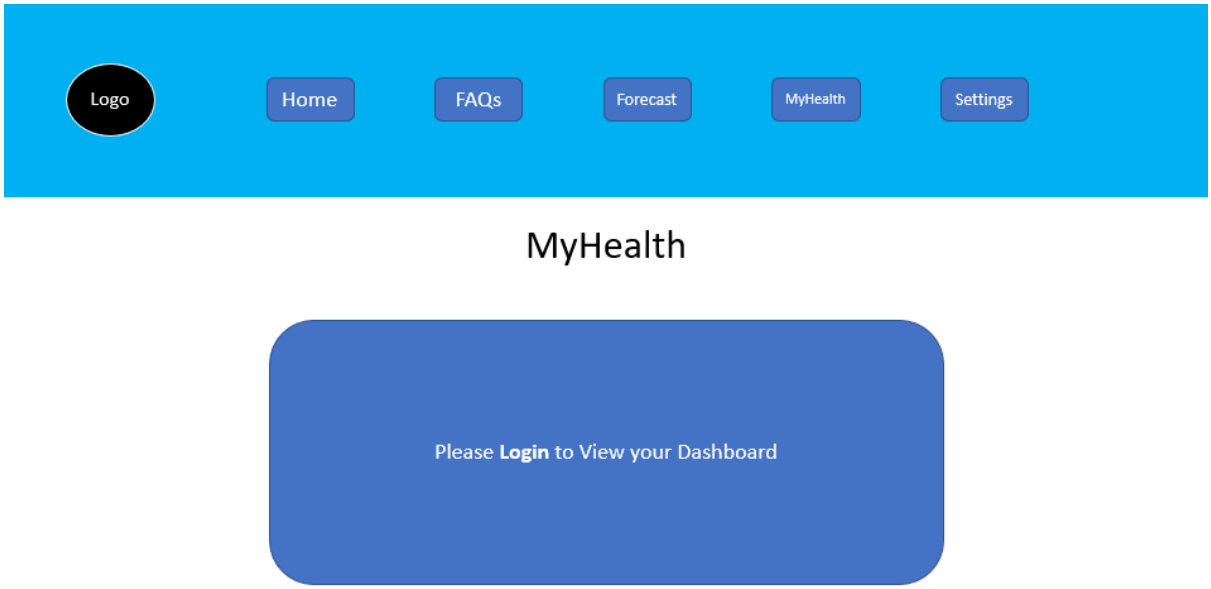
Current Weather

19°C

Sunny but Cloudy

Monday:	Thunderstorms	7°C
Tuesday:	Snowy	-3°C
Wednesday:	Windy	11°C
Thursday:	Raining	4°C
Friday:	Cloudy	15°C
Saturday:	Sunny	20°C

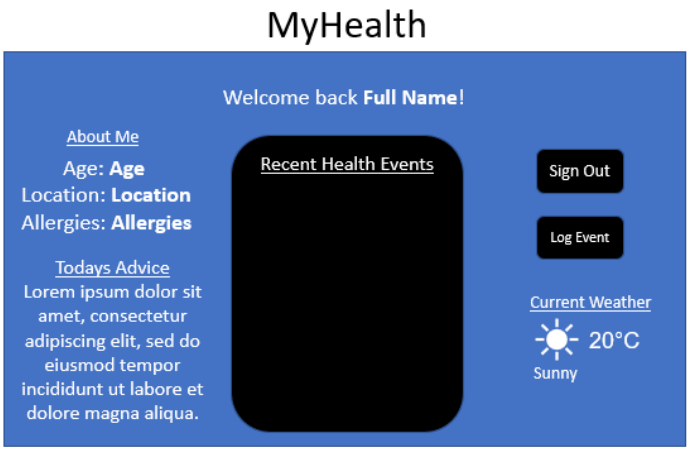
MyHealth (Pre-Login) Page

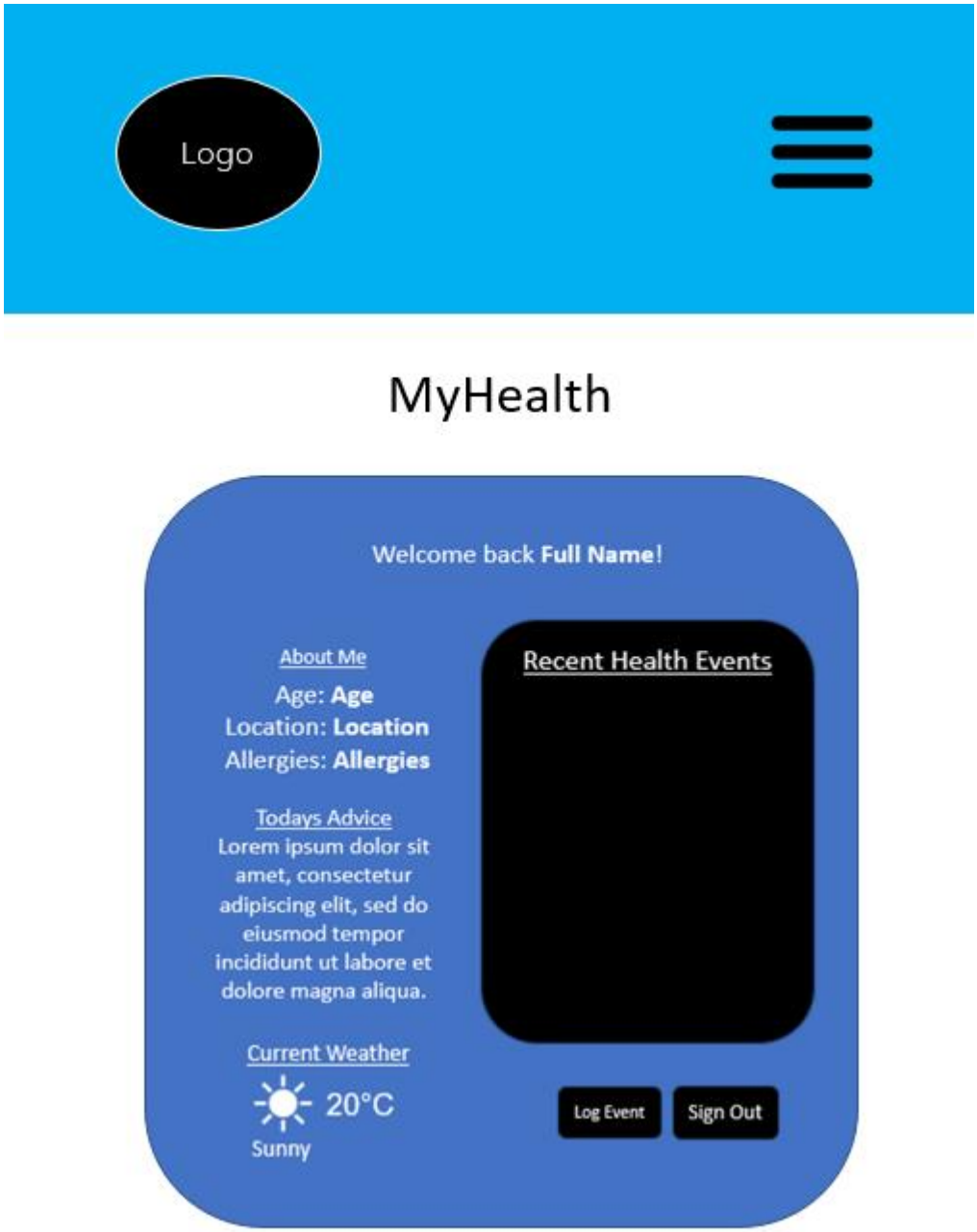


MyHealth

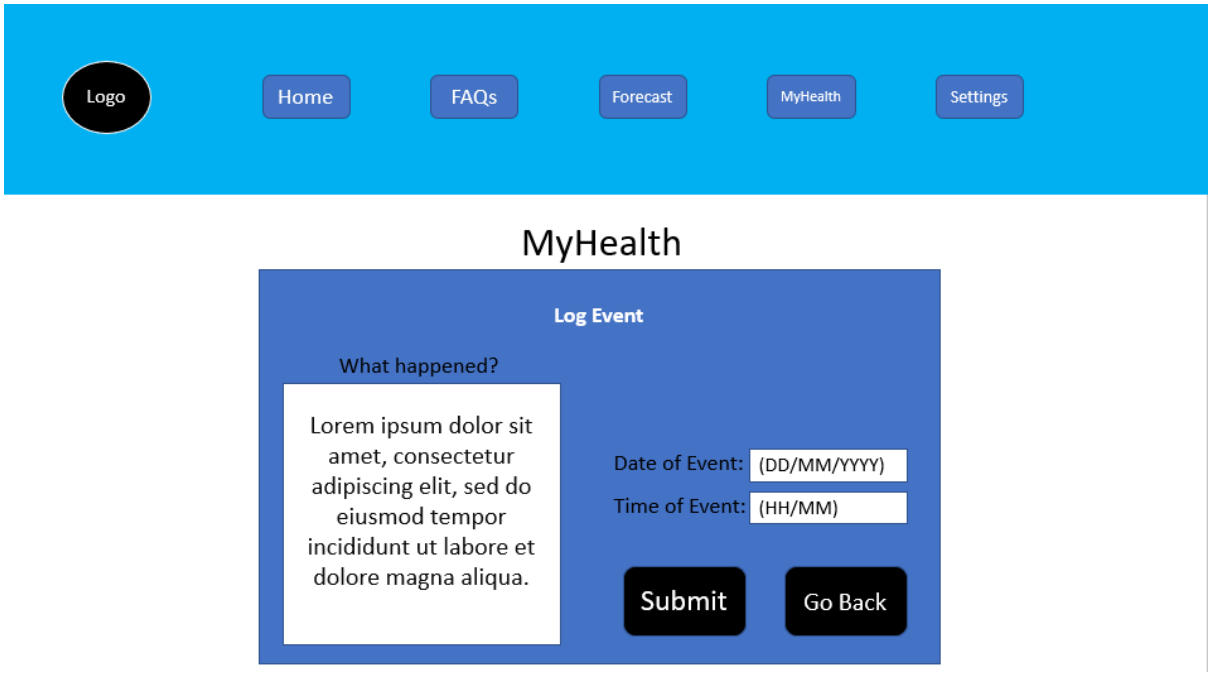
Please **Login** to View your Dashboard

MyHealth (Post-Login) Page




Logo
MyHealth
Welcome back **Full Name**!
About Me
Age: **Age**
Location: **Location**
Allergies: **Allergies**
Todays Advice
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Current Weather
20°C
Sunny
Recent Health Events
Log Event
Sign Out

MyHealth (Log Event) Page



MyHealth

Log Event

What happened?

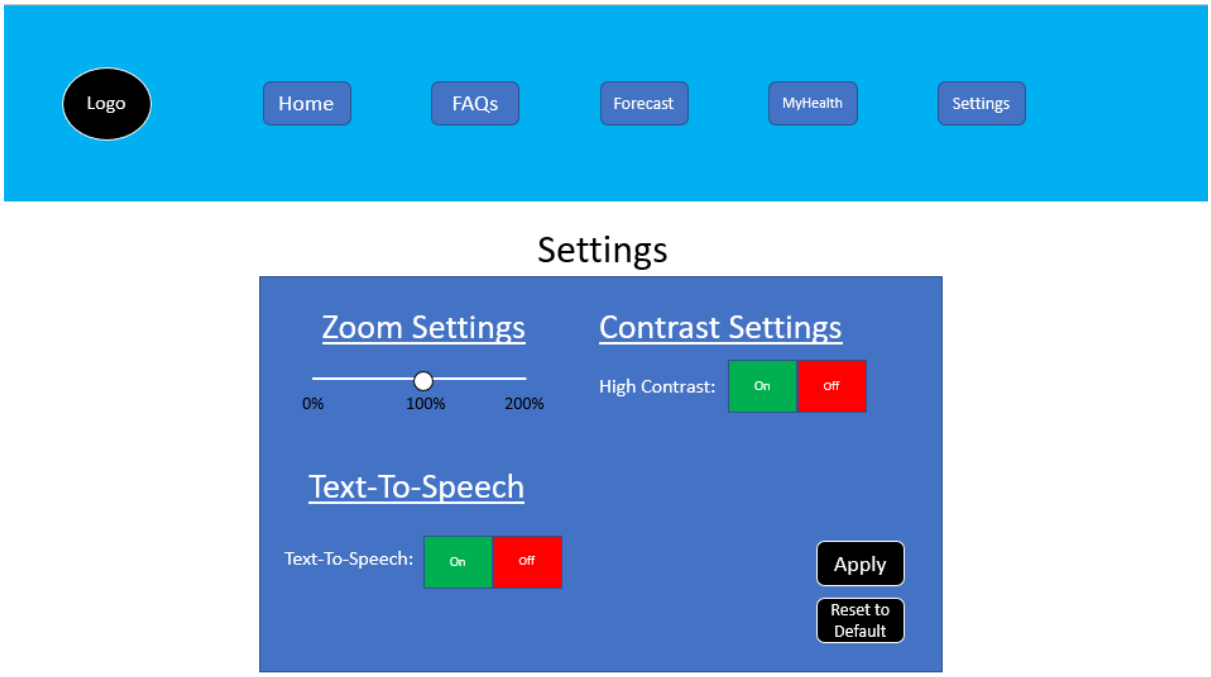
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

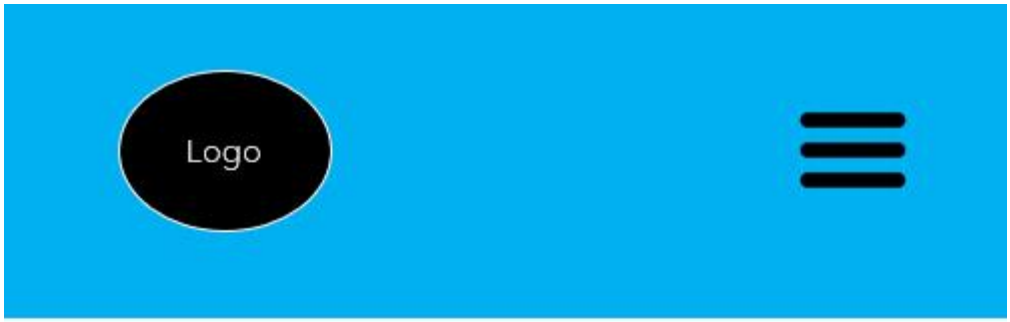
Date of Event: (DD/MM/YYYY)

Time of Event: (HH/MM)

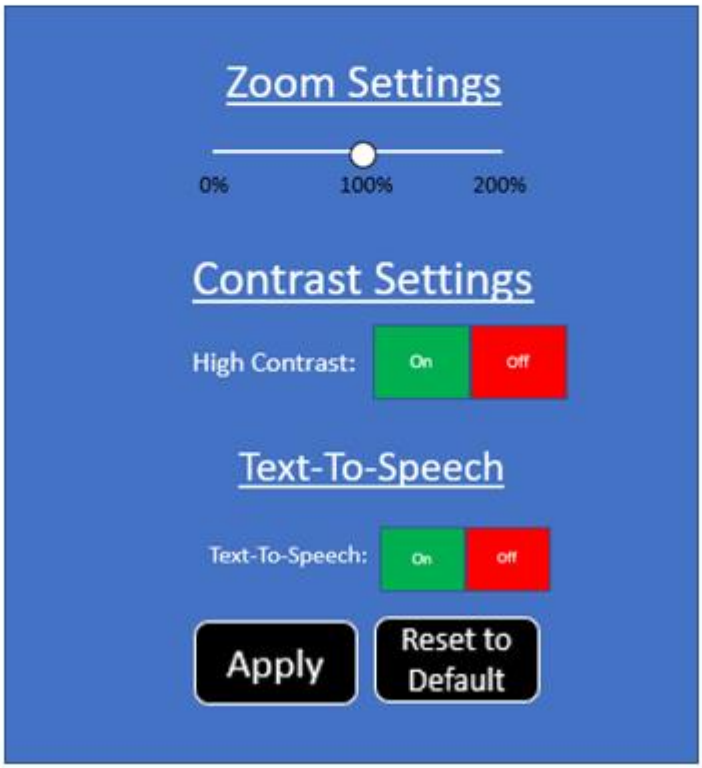
Submit Go Back

Settings Page

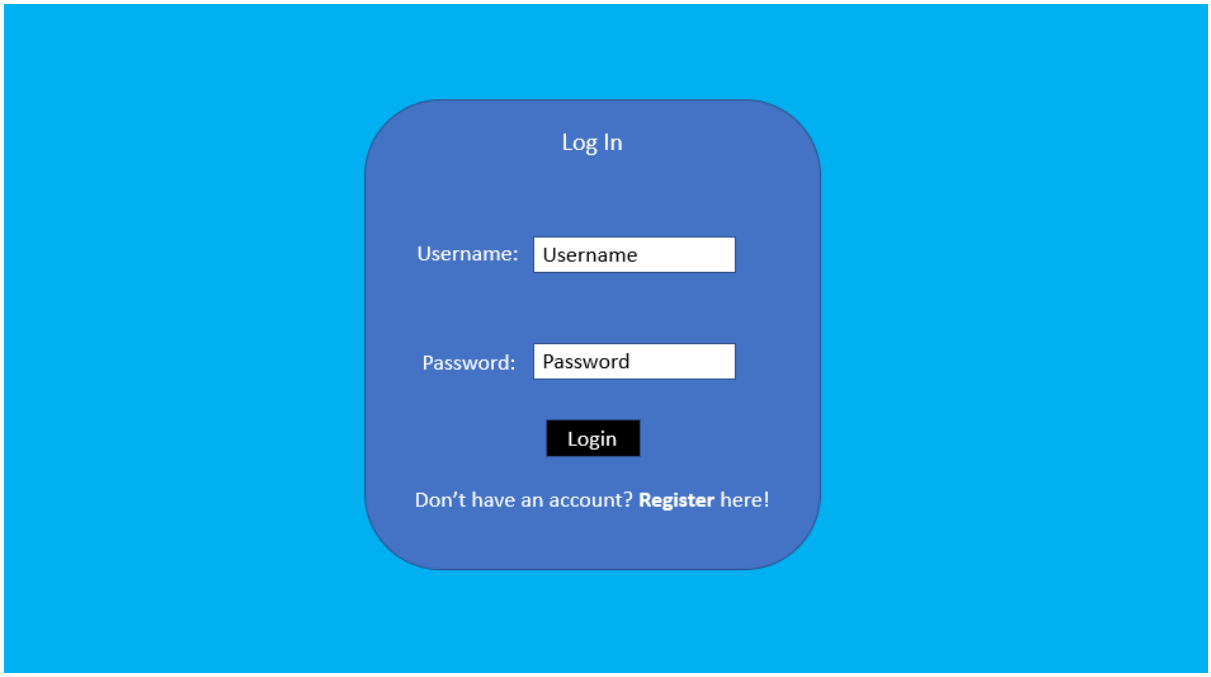



Logo

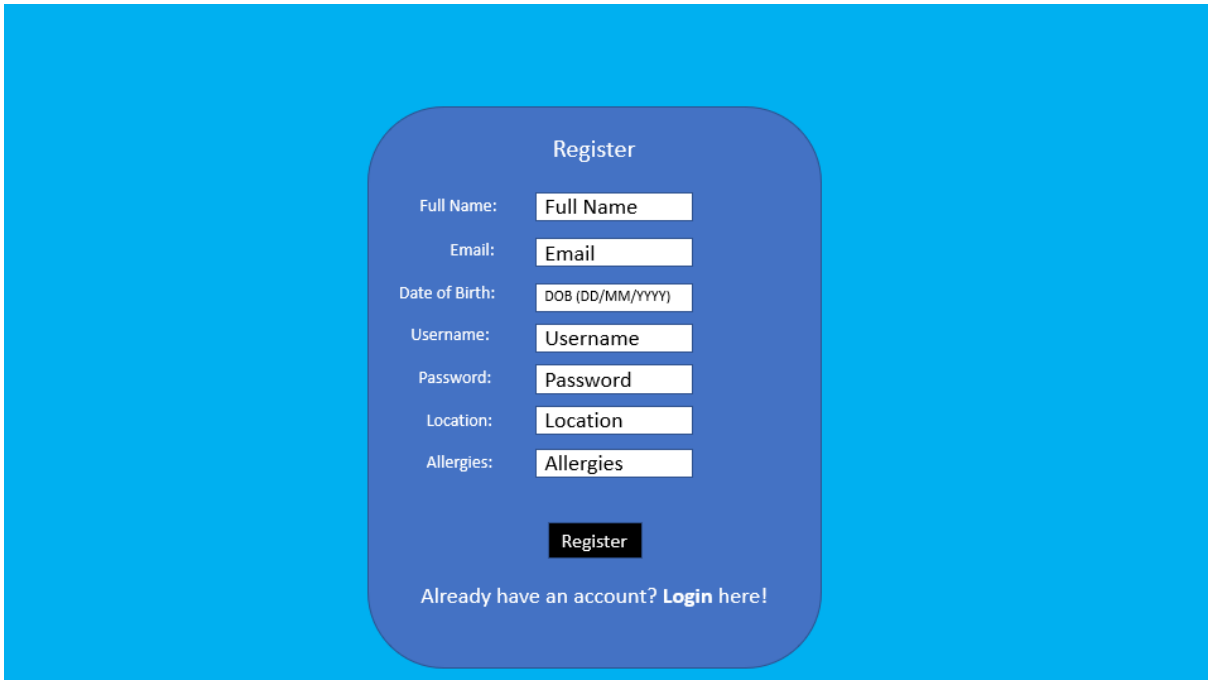
Settings


Zoom Settings
0%
100%
200%
Contrast Settings
High Contrast:
On
Off
Text-To-Speech
Text-To-Speech:
On
Off
Apply
Reset to Default

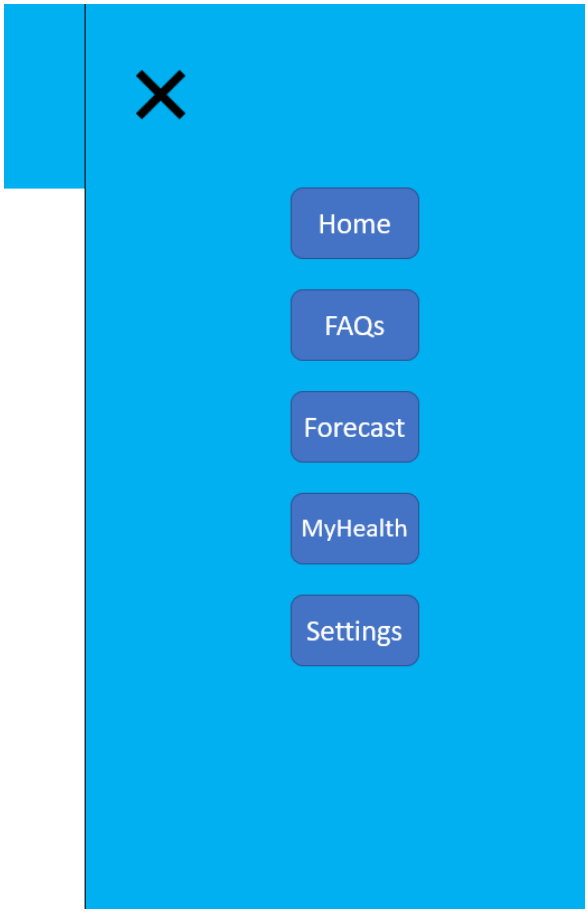
Login Page



Register Page



Mobile Navigation



Data Requirements

Each page will have their own specific data requirements.

However, I would like to note that there is potential for a weather database table to store and retrieve weather information to be displayed to the user but this could be dismissed with the use of APIs.

Here is a list of the following pages and the data requirements for each one:

Login Page

To access the login page, the user must go to the MyHealth dashboard without being logged in. This will prompt the user with the option to login and view their personal dashboard.

On the login page, the user will have two fields to input; their username and password. These will be presented as two textboxes which will be stored as variables called username and password to be checked in the database using backend code / queries that the user can't see. The variables 'username' and 'password' will both be stored as strings.

The way that the login page will check if the user's inputs are valid is that it will take both the variables 'username' and 'password' and it will check if they exist within a database called 'users'. If the username is in the database 'users', then it will check if the variable 'password' is equal to the password that is stored with that username in that database.

If the password is not the same as the one kept in the database, then the website will send an error message to the user that says something along the lines of 'username or password was incorrect, please try again'. In the error message, we cannot specify what the user got wrong as this is a cyber security issue and we would be giving a cyber-attacker information on if they are getting closer to hacking an account.

Below is a **mock-up** of the 'users' database. Please note this is subject to change as it could require more fields or tables in order for better functionality.

Users.db

Primary Key	Field Name	Data Type
🔑	UserID	int
	Full Name	string
	Email	string
	Date Of Birth	string
	Username	string
	Password	string
	Location	string
	Allergies	string

While it says that the date of birth is kept as a string, this will be done so that the user's age can be calculated from today's date back to the date of birth they inputted. When displaying their age on their MyHealth Dashboard, it will be displayed as an integer.

Within the 'users' database, there is a field called 'UserID', this will be an integer that is auto-incremented so that the user has an ID unique to them that can be used to identify them.

Register Page

For the user to get to the register page, they must click the hyperlink that says register here on the login page.

Once the user has clicked this hyperlink, they will be redirected to the register page where they will be greeted with 7 input fields, they will need to fill out to create an account. The 7 input fields can be seen on the design for the register page. All these fields will be stored as strings.

Like the login page, the code will be error-handled, and the user will not be able to create an account without filling the fields in correctly. This means the user must input their date of birth in the requested format (DD/MM/YYYY), as well as their email containing an '@' and '.com'. As for the locations and allergies field I have not decided whether it will allow for the user to type their location and allergies or select from a dropdown box. It will most likely be a dropdown box to ensure easier

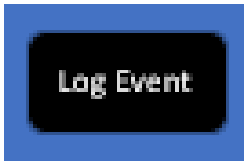
functionality when displaying weather and environment / health decisions based on the user's location and allergies.

One thing the user will not be able to do is use a username that is already in use. This will be error-handled via code that will take the 'username' and check if its already in the 'users' database, if it is then an error message will be displayed to the user saying something along the lines of "Username already exists, please try again!".

Once the user has successfully inputted the fields in the correct formats requested, these fields will be placed into the 'users' database for the user to then login with on the login page. I have already included a **mock-up** diagram of what the 'users' database will look like in the login page's data requirements.

MyHealth (Log Event) Page

For the user to get to the MyHealth (Log Event) Page, they must go to their MyHealth dashboard and press the button that says 'Log Event'



Once the user has clicked this button, they will be redirected to the MyHealth (Log Event) Page, and they can begin logging a health event.

The Log Event page consists of 3 fields and 2 buttons that the user can see. 3 text fields, one for what happened, one for the date of the event, and one for the time of event. There are then 2 buttons that say 'submit' or 'go back'. The submit button will place the user's health event in a database solely for health events, and the go back button will take the user back to their MyHealth Dashboard.

Once again, this page will be fully error-handled, and the user will not be able to submit a health event without filling the text fields out accordingly in their desired formats.

For the 'What Happened?' field, the user can simply input anything they want, as long as it is not left empty, they will be able to submit it. For the 'Date of Event' field, the solution asks it for a date in the format (DD/MM/YYYY). If the user does not follow this format, when they press submit, it will display an error message asking them to try again or that there has been an invalid input. This will be the same for the 'Time of Event' field if they do not input a time that is in the format of (HH/MM) being 0 – 24 hours and 0 – 59 minutes.

The field for 'What Happened?' will be stored as a variable called 'EventDetails'.

The field for 'Date of Event' will be stored as a variable called 'EventDate'.

The field for 'Time of Event' will be stored as a variable called 'EventTime'.

Once the user has successfully inputted these fields in the desired manner, the health event will then be logged into the database for health events. Below is a **mock-up** of how the health event database will look.

Events.db

Key	Field Name	Data Type
Primary Key	EventID	int
Foreign Key	UserID	int
	EventDetails	string
	EventDate	String
	EventTime	String

The database will have the 'UserID' as a foreign key so that it can help link events to users and make the 'Recent Health Events' table on the user's MyHealth Dashboard only have health events they have logged. This protects other user's personal data such as their own private health being displayed to the public by having the foreign key link to only their account to be displayed on only their MyHealth Dashboard.

MyHealth Dashboard Page

Once the user has logged in, they will be redirected to the MyHealth Page where they can view their own personal dashboard. This will contain a section called 'About Me' where it will display their age (as an integer), their location, and if they listed any allergies. These will have all been assigned at the registration of their account.

On their dashboard, there will also be a table called 'Recent Health Events'. This will display any health events that have been logged by the specific user by searching the 'Events' database and looking for any health events that are linked to the same UserID as the one that is currently logged in. If it finds any logged health events by this user, it will display it to the user on their dashboard.

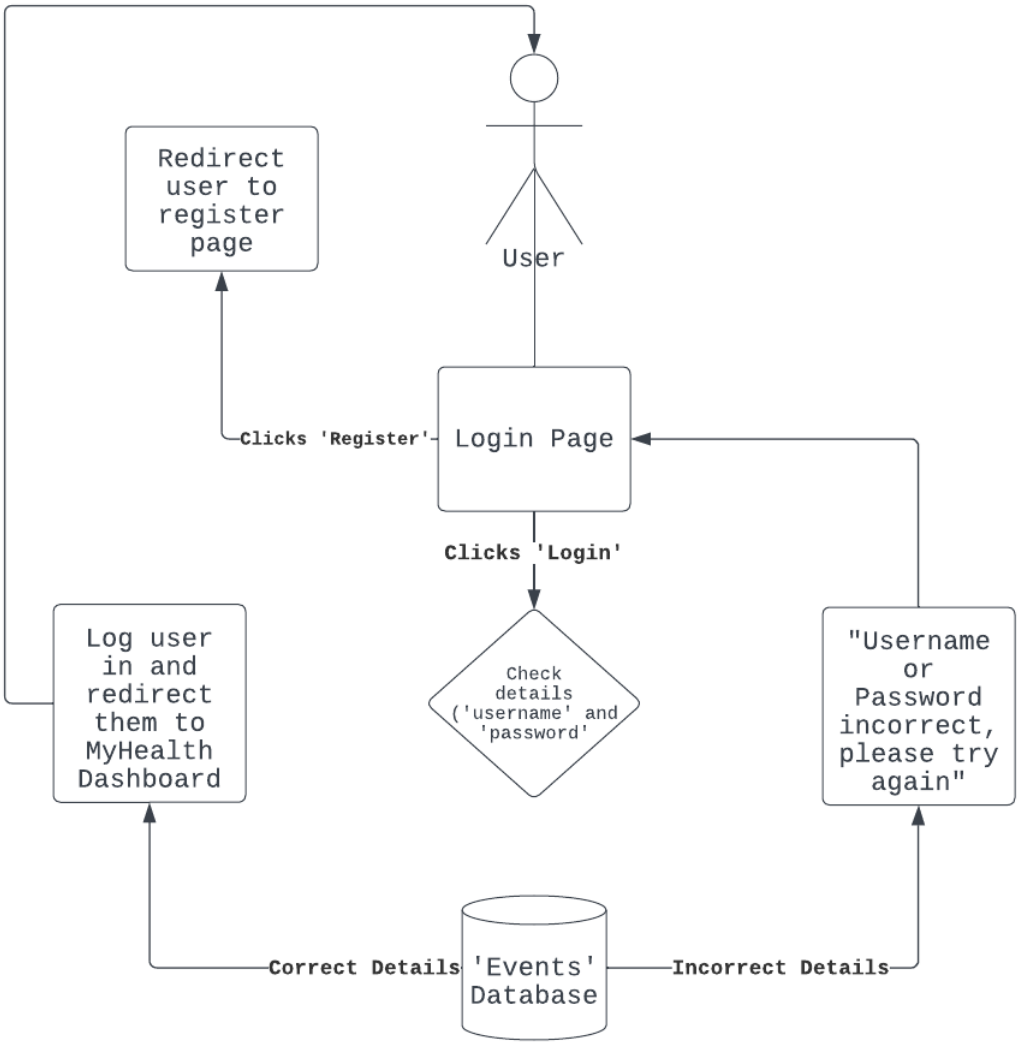
There is also a section called 'Todays Advice'. This section will contain any advice that the user could take based on the weather in their area or their allergies. There is also a 'Current Weather' section that displays the weather at their location which could also determine the advice they are given.

The final features on the MyHealth Dashboard are the buttons 'Sign Out' and 'Log Event'. Clicking 'Log Event' will redirect the user to the page where they can log any health events. Whereas, clicking 'Sign Out' will end the session that the user started when they logged in. This means they can no longer access any of their data or their dashboard unless they log back in. This is a feature implemented solely for the safety and security of the user's data.

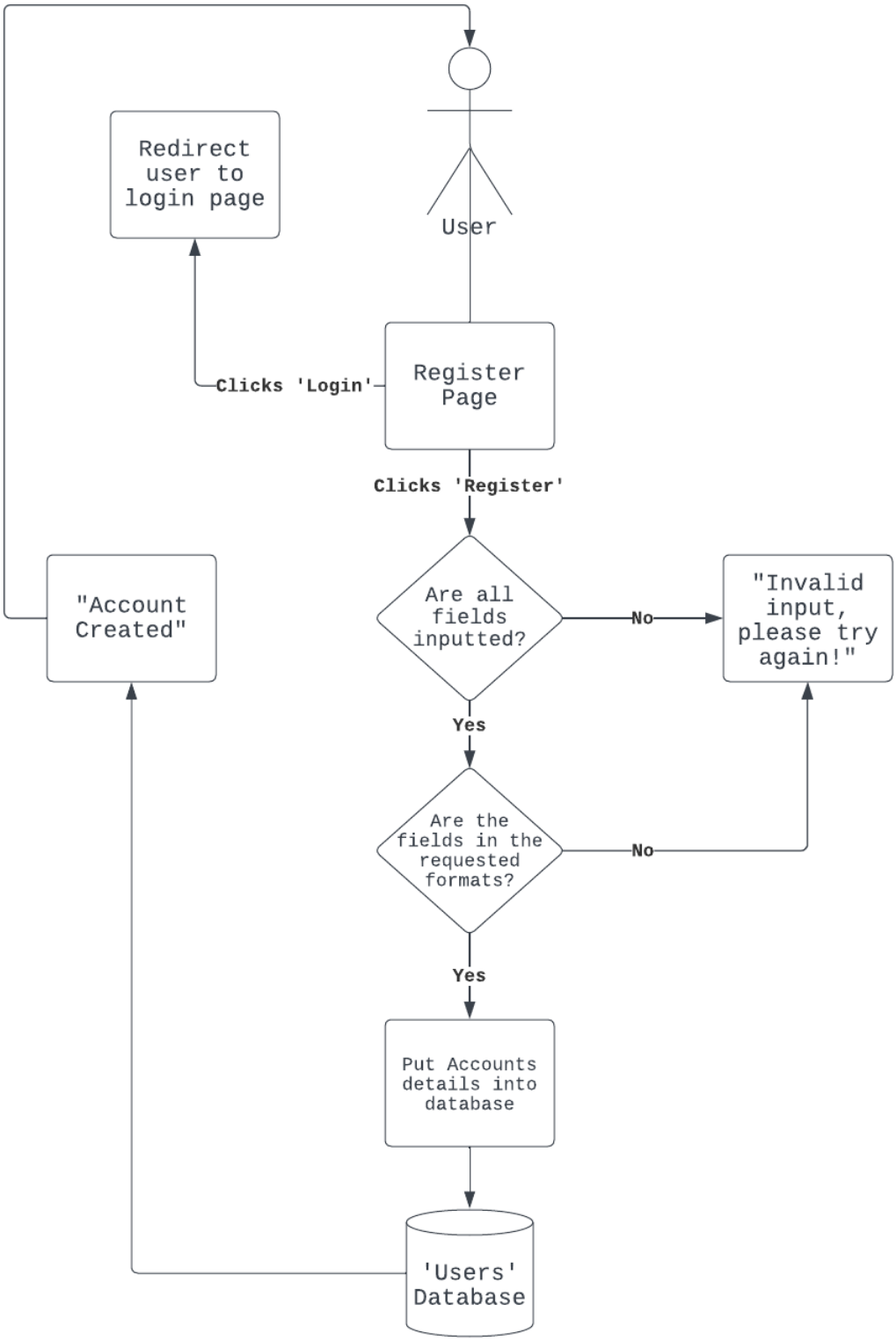
Data Flow Diagrams

Below are diagrams of how the data will flow for each page that uses data.

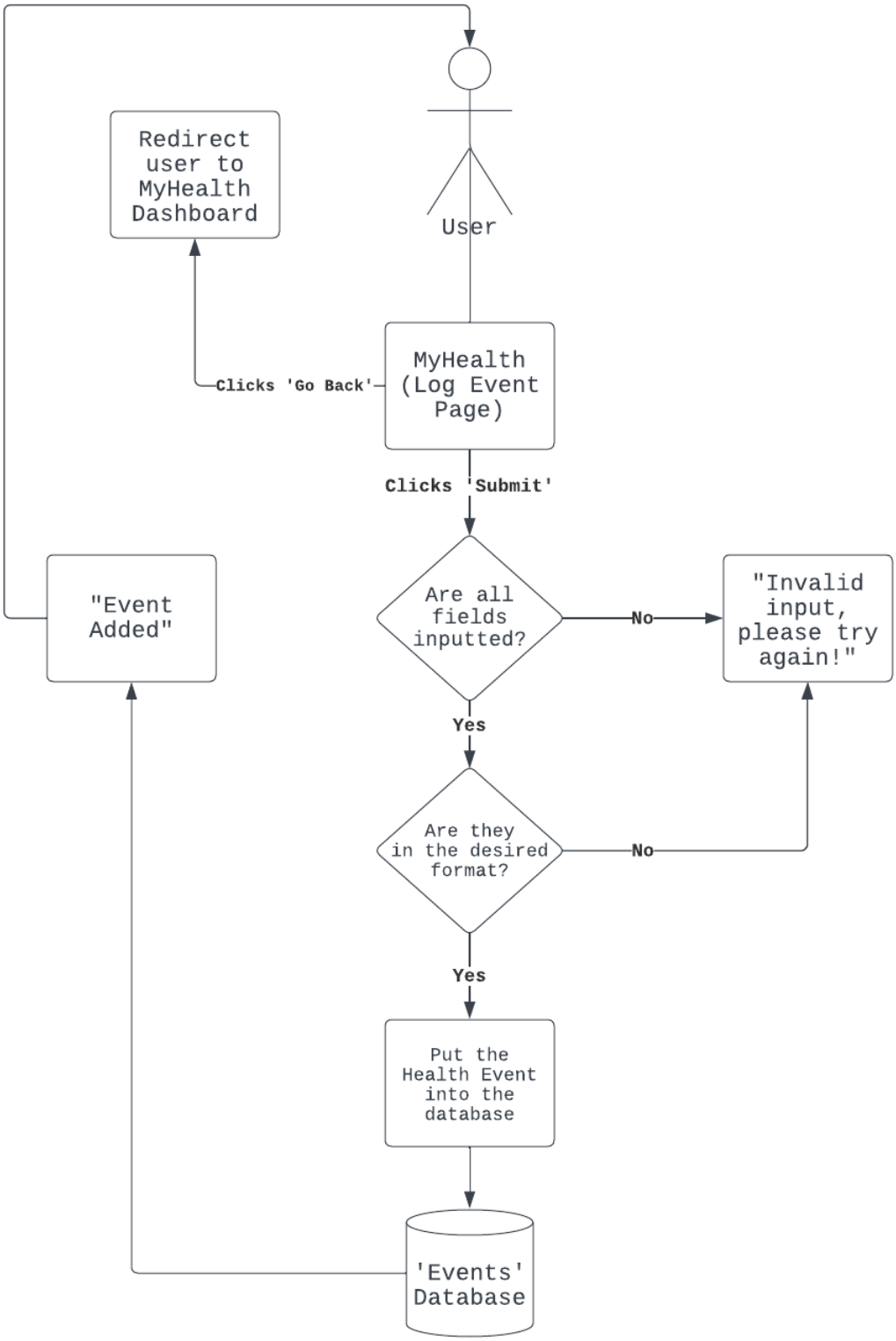
Login Page


Redirect user to register page
User
Clicks 'Register'
Login Page
Clicks 'Login'
Log user in and redirect them to MyHealth Dashboard
Check details ('username' and 'password'
"Username or Password incorrect, please try again"
Correct Details
'Events' Database
Incorrect Details

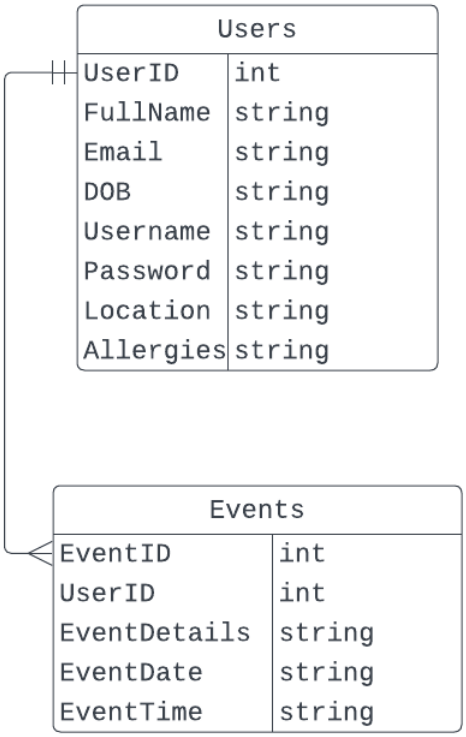
Register Page



MyHealth (Log Event) Page



Entity Relationship Diagram



Algorithms

Login Page

```
BEGIN

username = TEXTBOX
password = TEXTBOX
login = BUTTON
register = HYPERLINK
saved_users = DATABASE



IF login IS CLICKED
      IF username OR password IS EMPTY
            SEND "You have left a field blank! Please Try Again!" TO DISPLAY
      ELSE
            IF username NOT IN saved_users
                  SEND "Invalid Username or Password! Please Try Again!" TO DISPLAY
            ELSE
                  IF username AND password IS EQUAL TO THE USERNAME AND PASSWORD IN saved_users
                        LOG USER IN
                        REDIRECT USER TO MYHEALTH PAGE
                  ELSE
                        SEND "Invalid Username or Password! Please Try Again!" TO DISPLAY

IF register IS CLICKED
      REDIRECT USER TO REGISTER PAGE


END
```

Register Page

```
BEGIN

fullname = TEXTBOX
email = TEXTBOX
dob = TEXTBOX
username = TEXTBOX
password = TEXTBOX
location = TEXTBOX
allergies = TEXTBOX
register = BUTTON
login = HYPERLINK
saved_users = DATABASE



IF register IS CLICKED
      IF fullname OR email OR dob OR username OR password OR location OR allergies IS EMPTY
            SEND "You have left a field blank! Please Try Again!" TO DISPLAY
      ELSE
            IF "@" AND ".com" NOT IN email
                  SEND "Please enter a valid email address!" TO DISPLAY
            ELSE IF dob IS NOT IN THE FORMAT (DD/MM/YYYY)
                  SEND "Please enter a valid Date of Birth in the format (DD/MM/YYYY)!" TO DISPLAY
            ELSE
                  SAVE USER IN saved_users
                  SEND "Account Created!" TO DISPLAY

IF login IS CLICKED
      REDIRECT USER TO REGISTER PAGE


END
```

MyHealth (Log Event) Page

```
BEGIN

event_details = TEXTBOX
date_of_event = TEXTBOX or DATE SELECT
time_of_event = TEXTBOX or TIME SELECT
submit = BUTTON
go_back = BUTTON
health_events = DATABASE


IF submit IS CLICKED
      IF event_details IS EMPTY
            SEND "Please enter the details of the health event!" TO DISPLAY
      ELSE IF date_of_event IS NOT SELECTED OR IN THE FORMAT (DD/MM/YYYY)
            SEND "Please enter a valid date for the health event!" TO DISPLAY
      ELSE IF time_of_event IS NOT SELECTED OR IN THE FORMAT (HH/MM)
            SEND "Please enter a valid time for the health event!" TO DISPLAY
      ELSE
            SAVE HEALTH EVENT IN health_events

IF go_back IS CLICKED
      REDIRECT THE USER TO THE MYHEALTH PAGE

END
```

Test Strategy

Before I begin the test strategy, I have decided to note that any testing that doesn't specify if its black-box testing means it is white box testing and the user testing it has complete knowledge of the design documentation and inside code, as well as how it links to the back-end databases.

I would also like to note that during the testing stages, if any changes are made, it will be followed by regression testing to ensure that no functionality has been altered or broken accidentally. There will also be beta testing to ensure further development and improvement of the solution. Beta testing allows for real users to test the solution and report any bugs or overall features that need fixing or improvement.

Date of test	Component to be tested	Type of test to be carried out	Prerequisites and dependencies

TBA	**Home Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**FAQs Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**Forecast Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**MyHealth Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**Login Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**Register Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**MyHealth Dashboard**	**Functionality Testing**	**The Page is developed fully**
TBA	**MyHealth Log Event**	**Functionality Testing**	**The Page is developed fully**
TBA	**Settings Page**	**Functionality Testing**	**The Page is developed fully**
TBA	**Home Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**FAQS Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**Forecast Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**MyHealth Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**

TBA	**MyHealth Login Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**MyHealth Register Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**MyHealth Dashboard**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**My Health Log Event Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**Settings Page**	**User Acceptance Testing (Blackbox)**	**Dependencies: Any needs specific to the user may impact their acceptance of the digital solution**
TBA	**MyHealth Login Page**	**Integration Testing**	**Dependencies: The user creates an account and attempts to login with either correct or incorrect details**
TBA	**MyHealth Register Page**	**Integration Testing**	**Dependencies: The user attempts to create an account**
TBA	**MyHealth Login Page**	**Unit Testing**	**Dependencies: The user tests each input field by leaving one blank each time to ensure they are error-handled.**
TBA	**MyHealth Register Page**	**Unit Testing**	**Dependencies: The user tests each input field by leaving one blank each time to ensure they are error-handled.**
TBA	**MyHealth Register Page**	**Stress Testing**	**The Page is developed fully and linked with the back-end database**

TBA	**MyHealth Login Page**	**Stress Testing**	**The Page is developed fully and linked with the back-end database**
TBA	**MyHealth Login Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**MyHealth Register Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**MyHealth Dashboard**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**Home Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**FAQs Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**Forecast Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**
TBA	**Settings Page**	**Black-box Testing**	**Have a user with no inside knowledge of the software to test**