BMI/BFP Design Document

Context and Background

The purpose of this web application is to allow urban people with obesity and for models calculate their BMI and BFP (Body Fat Percentage) respectively. This will help urban residents to know if they are obese or not and also will help let models know if they're in shape for an upcoming model contest. These users will be able to use the website called *wwwFitBonny.com* that will perform these calculations. The platform that will be used is a web application that will store data on a database server.

When the user visits our website, it will prompt them to enter their username and password. They will then be forwarded to the homepage that will have the selection of doing a BMI or a BFP. They will be redirected to the BMI or BFP webpage to get their result by entering their weight in pounds and height in inches or age in years. They will then be notified if they are underweight, normal or obese in the case of BMI users. If its BFP users, they will be notified of their excess fat percentage. Lastly, if the user is not pleased with their result there is a chat feature where they can seek advice from experts or from other users in the chat.

Goals and Non-goals

The goals of the website are:

1. An input box on the homepage to take the username and password.

2. BMI and BFP buttons will be provided for the user to choose which they want to calculate.

3.Clicking on the BMI button will take the user to the BMI webpage where the webpage will prompt the user to enter their weight and height.

4.The webpage will return the calculated BMI along with Underweight, Normal or Obese.

5.Clicking on the BFP button will take the user to the BFI webpage where the webpage will prompt the user to enter their weight and height.

6.The webpage will return the calculated BFP along with the calculated BFP which states how much percentage weight over or under.

7.A chat box will be provided at the end of the BFP webpage and the BMI webpage

The Non-Goals of this website are:

1. The logo of the website will be posted at the top of homepage.
2. The website will be chat history will be stored in the database.
3. The users' results will be stored in the database server.

Design

The website will use two external services to accomplish the goal. We will be using MongoDB Atlas to store all our user's username and passwords along with using Replit.com to host our website.

Getting Started:

When the user first loads the website, they will be prompted to input their username and password. See the Sequence diagram below:

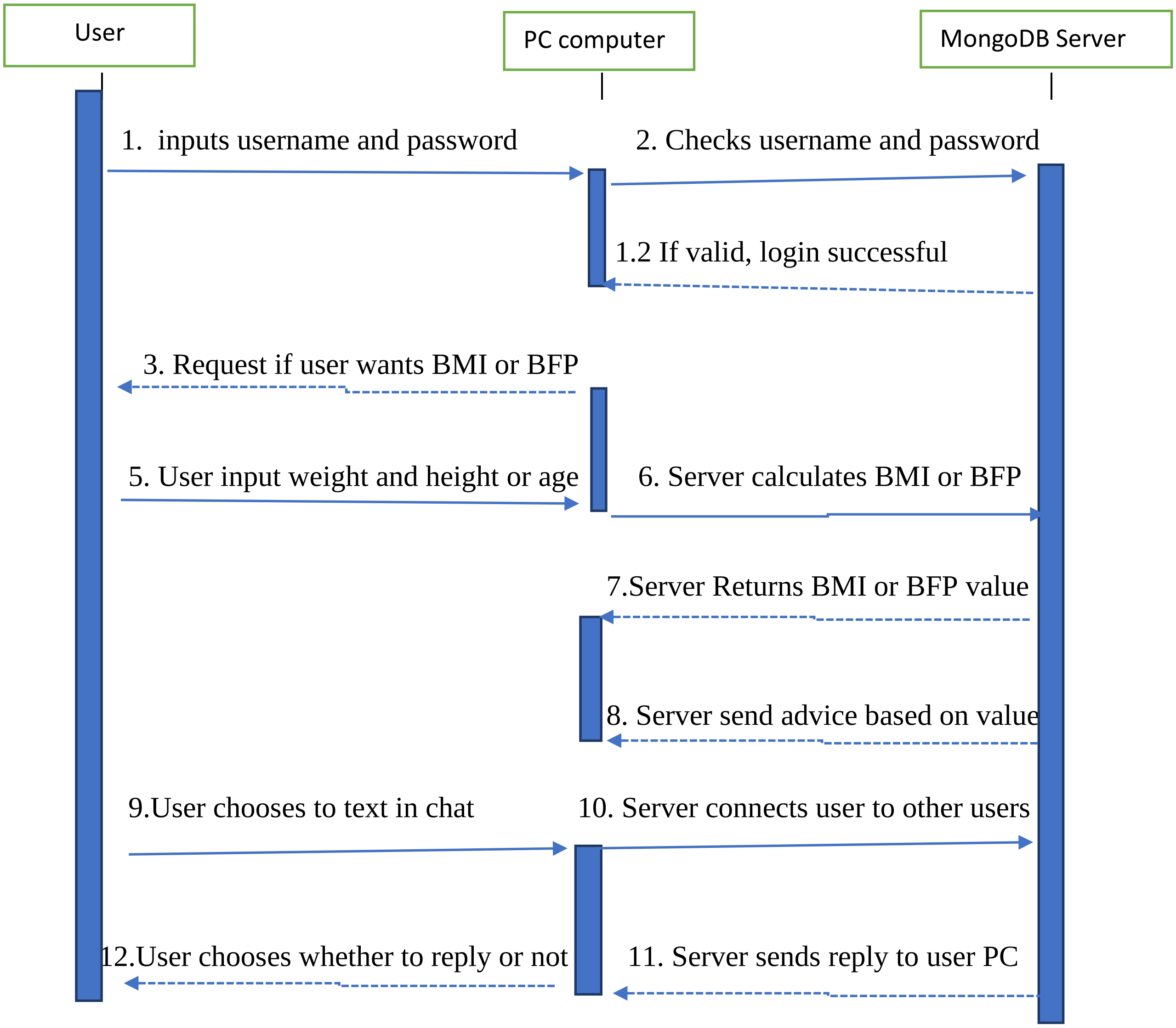
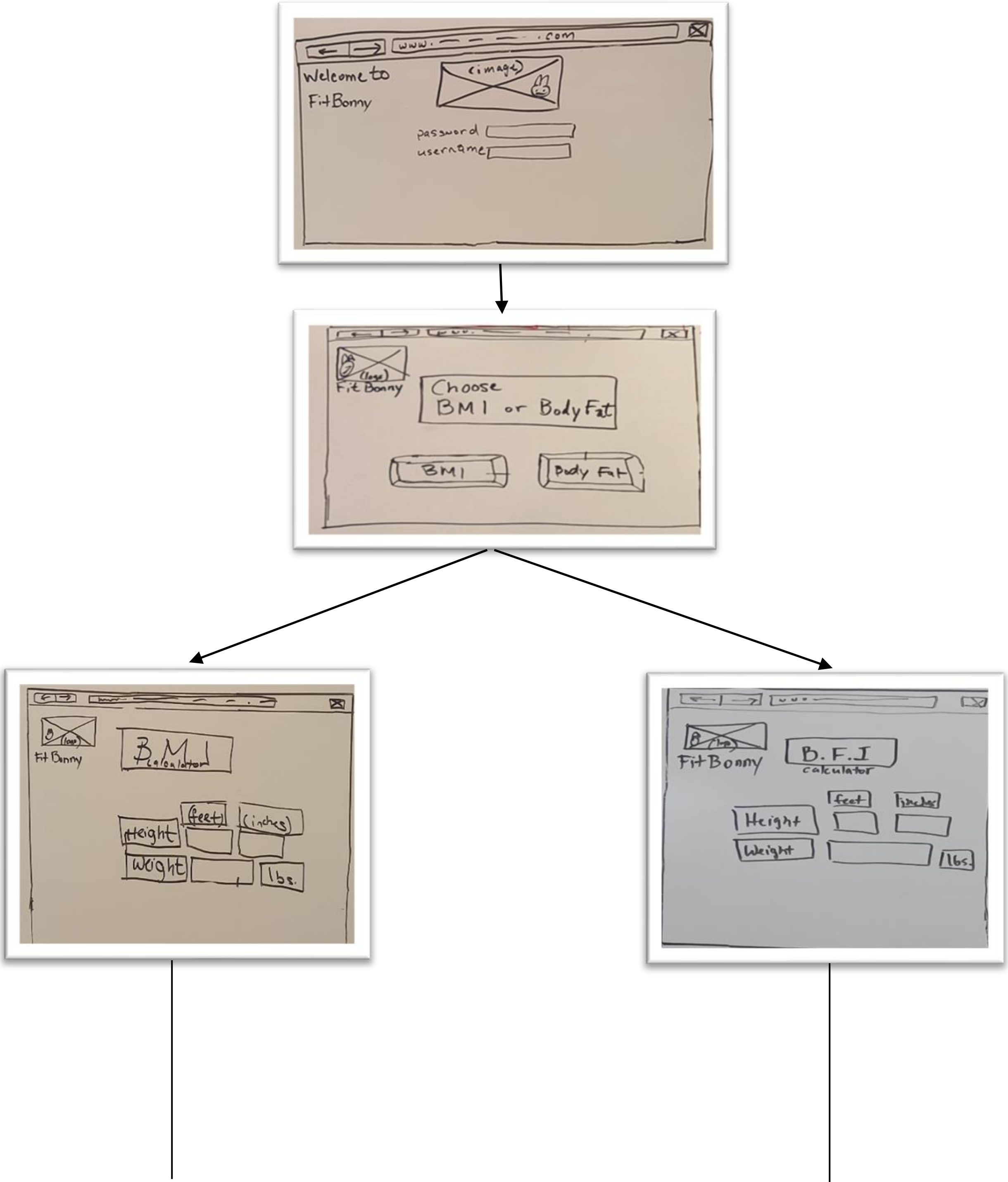

User
PC computer
MongoDB Server
1. inputs username and password
2. Checks username and password
1.2 If valid, login successful
3. Request if user wants BMI or BFP
5. User input weight and height or age
6. Server calculates BMI or BFP
7.Server Returns BMI or BFP value
8. Server send advice based on value
9.User chooses to text in chat
10. Server connects user to other users
12.User chooses whether to reply or not
11. Server sends reply to user PC

Figure 1. Website Sequence Diagram of requests between web user, and their personal computer and the MongoDB server.

The server will check the values entered against what is stored in MongoDB, if they are correct, they will be redirected to the homepage. If not, they will have to try again. Secondly, the user is will see two buttons that will be labelled ‘BMI’ and ‘BFP’ that they will have to choose between. See the wire frame diagram below:



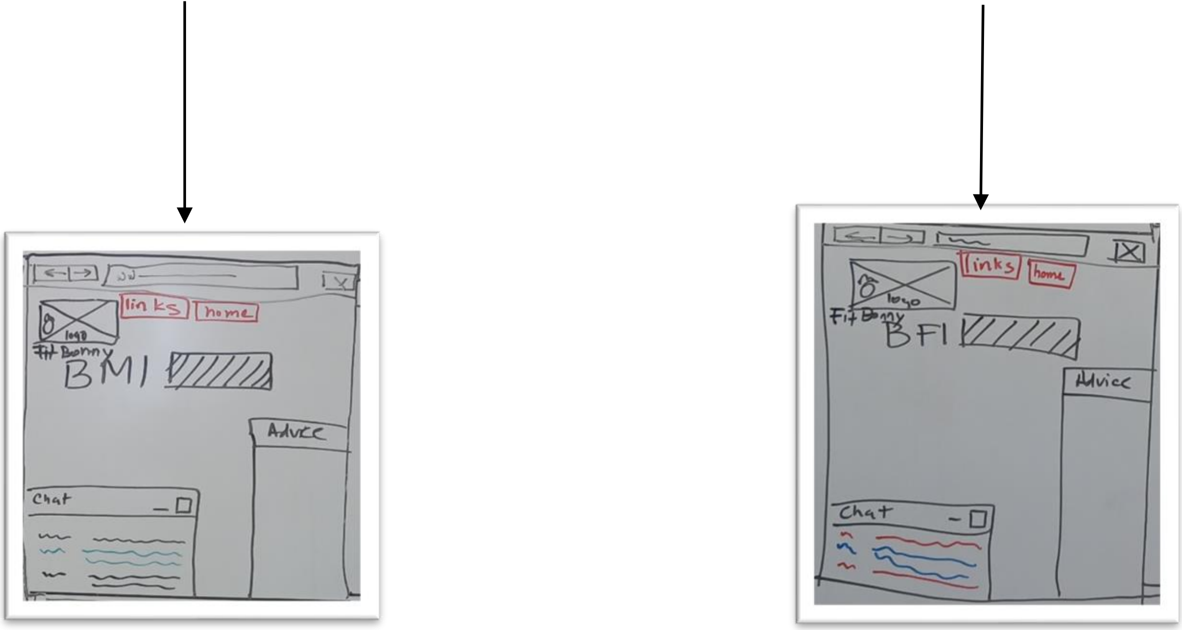


Figure 2. Wire Frame Diagrams

BMI/BFP Calculations

For calculation of the BMI or the BFP, the user will have to input their weight in pounds and their height in feet/inches. An API call will be made to the MongoDB Server to calculate the selected BMI = (708*weight/heights^2) and the BFP = (1.39*BMI) + (0.16*age) - (10.34*gender) - 9

The server then returns the calculated result to the user personal computer along with some advice based on the calculated value.

Chat Feature

The user can choose to discuss their result with other users in the chat box or take expert advice from the box below. If the users choose to use the chat feature, then the API will send the users remarks to the MongoDB server then the server will connect the user with other users using the feature at that moment.

Alternative Approach

One alternative approach would be to implement our program into a mobile application using Android studio.

Pros

1. Ease of access by simply tapping on the mobile app on their phone to access everything in one place in oppose to every time potential customers want to access the webpage; they need to use a search engine.

2. It would let the user engage with our application faster and more efficiently.

Cons

1. Its more expensive to build and maintain than a web application.

2. Compatibility with different platforms (example: iOS and Android) usually means designing and building the app from scratch for each platform.

Another alternative approach would be to host our own database server to store all our users' usernames, passwords and results.

Pros

1. Will not have to pay or wait to access our own database.

Cons

1. More costly to build and maintain our own Database