**HEALTHMATE**

**Software Design**

**CSCI-P465/565 (Software Engineering I)**

**Project Team**

**Sakshi Sitoot**

**Ratan Vadapalli**

**Vikram Kolla**

**Himani Patil**

**Satwik Kulkarni**

1. **Introduction:**

Healthmate is a web application that enables users to track various aspects of their wellness and fitness, as well as access workout programs and resources online. These web applications can be used to help individuals improve their physical and mental health by providing content and resources to track their progress and achieve their wellness goals.

The system has three different views for the clients, fitness professionals and the admins. The core features of the application include Login, Registration, Search, Dashboard, Chat, Recommendations, Subscriptions, Rating and Reviews.

* 1. **System Description:**

Healthmate helps people ensure they are leading a healthy lifestyle and keeps them connected to experts in the field and helps people monitor their health. This project is chosen to help people overcome their sedentary lifestyles. This eliminates the geographical barrier if people want to connect with some professional who lives far away.

* 1. **Design Evolution:**

The design solution is centered around the needs and goals of the target users, who are people interested in improving their physical and mental health. It emphasizes a clean and simple user interface, with clear calls-to-action and visual cues to guide the user. This can help ensure that the web application is easy to use and understand, which can increase user engagement and satisfaction. The web application is designed to be flexible and customizable, allowing users to create personalized workout plans and set their own goals. The web application includes features that allow users to connect with fitness professionals and receive support and motivation. This can help increase user engagement and make the process of improving one's physical and mental health more enjoyable and sustainable.

* + 1. **Design Issues:**

There are few design issues, requirements, and constraints that can drive the design of the software:

* Data privacy and security: It is crucial to make sure that the web application complies with industry standards for data privacy and security because users will be disclosing sensitive information about their physical and mental health.
* User experience: The programming language, framework, and hosting infrastructure used to create the web application should be able to support the desired features and functionality while also adhering to financial and resource restrictions.
* User experience: The programming language, framework, and hosting infrastructure used to create the web application should be able to support the desired features and functionality while also adhering to financial and resource restrictions.
* Scalability: The web application should be built to support a sizable number of users and be scalable as the user base expands.
  + 1. **Candidate Design Solutions:**
* Modular Design: This technique entails segmenting the web application into more manageable, standalone modules that can be created and independently evaluated. This can make it simpler to design web applications, decrease errors, and make it simpler to upgrade and manage them over time.
* User-centered design: This design solution puts the user at the center of the design process, considering their needs, preferences, and goals. It involves conducting user research and testing to ensure that the web application is designed to meet the specific needs of the target users.
  + 1. **Design Solution Rationale:**

The development process may be more flexible if it uses a modular design approach. The ability to work on various modules concurrently by various people is known as parallel development. By doing this, the development process can be sped up and the time to market shortened. Code reuse may also benefit from this design. If a certain module is created, it may be readily used in other components, which can reduce the amount of time and money spent on development.

* 1. **Design Approach:**
     1. **Methods:**

Object-oriented design: This helps organizing the design into a set of objects, which can be used to represent different components and functions of the system. It makes the design more modular and scalable and can facilitate reuse of code.

* + 1. **Standards:**
* User interface (UI) design standards: Layout, typography, color scheme, and other visual features of the design are constrained by UI design standards. A consistent and user-friendly interface will be made by adhering to UI design principles.
* Naming conventions: Classes, functions, and variables will be named according to the application's naming standards. Following naming standards can make the code easier for other developers to read and maintain.
* Security standards: Security standards offer recommendations for creating web apps that are safe and shield user data from assaults or unauthorized access. The web application may be made safe to use and user privacy can be protected by adhering to security standards.
  + 1. **Tools:**
* Version control systems: Changes to the code and design files are being managed using version control tools like Git. With the use of Git, developers will be able to work together on the project and keep track of changes made to the code or design. It provides versions of the code and design files as well as change logs.
* Integrated development environments (IDEs): Developers have a workspace to write, test, and debug code using the IDEs like Microsoft Studio Code and Atom. These tools aid programmers in streamlining their work, enhancing the quality of their code, and controlling dependencies. These tools produce output that includes the source code for the application.

1. **System Architecture:**
   1. **System Design:**

Database

Backend

Frontend

DBMS

HTTPS

API

OAuth

Authentication Services

* 1. **External Interfaces:**

We are currently using Google authentication’s interface. We have been using this interface to secure our logins, accessing Google only when we have a successful username/password entry pair.

1. **Component Design:**
   * 1. **Homepage**
     2. Description:

The Homepage is the landing page on our website before logging in. Users can register a new account, log in using Google, or utilize their existing credentials at the homepage login button to login. Upon successful login, the dashboard will be displayed. If they wish to create a new account, they are sent to the page where they enter account details. They can enter their Gmail account and password if they log in using Google.

* + 1. Responsible Development Team Member:

Ratan Tejaswi Vadapalli

* + 1. **Login Page**
    2. Description:

Users can often access restricted or customized material on a website through the login page. The login screen will include the following fields: Username/Email, Password, Forget Password Link, Sign in Button, New Account or Sign-Up Button. Login to social media

* + 1. Responsible Development Team Member:

Himani Patil

* + 1. **Registration Page**
    2. Description:

Users can register for a new account and sign up on a website's registration page to gain access to the site's personalized features and content. The parts of a registration page are Username/Email Field, Password Field, Confirm Password Field, Personal Information Fields such as name, birthdate, and gender, Contact Information Fields, Terms and Conditions Checkbox, "Create Account" or "Sign Up" Button.

* + 1. Responsible Development Team Member:

Sakshi Sitoot

**Revision History**

|  |  |  |
| --- | --- | --- |
| Revision | Date | Change Description |
| 1.1 | 02/19/2023 | Created Initial System Design Document with Initial Components of the Software |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |