

Software Engineering Department  
ORT Braude College

Capstone Project Phase A – 61998

Project No.: **23-1-D-8**

**Life-Style Builder**

March – 2023

Authors:

Shahar Almog – 206217606

Yehonatan Ben Shitrit – 315687442

GIT:

https://github.com/yonibsh145/Healthy-life-style-supported-by-doctors

**Abstract**

The healthy lifestyle programs are very popular now. People want to lose weight, prevent from diabetes, or simply feel good. Such programs collect information about users, connect them with specialist and have many other activities. But most of the existing tools are mainly directed for patient usage. There are no robust tools directed for doctors or other healthy lifestyle specialists that convenient for building and manage of such programs.

In this project we are going to develop a new web app providing a technical support for building and realization of healthy lifestyle programs, making it quick and easy. Our program will be with two interfaces: for doctors and for patients. The doctor's part will be a healthy life-style program builder, whereas the patient part will be a life-tracker application, realizing the life-style programs.

Table of Contents

**1. Introduction4**

**2.Related work and Background 5**

2.1Background5

2.2Related Works6

2.2.1 MyFitnessPal6

2.2.2 FatSecret6

2.3Agile Development7

2.4Client-Server7

2.5Cloud Computing8

3.**Expected Achievements**9

3.1Project Goals9

3.2Unique Features9

4.**Engeeniring Process**10

4.1Process10

4.2Requierments11

4.3Technology Choices12

4.3.1BackEnd - NodeJS & MongoDB13

4.3.2 Frontend - ReactJS13

4.4Diagrams14

4.4.1 Use case14

4.4.2 Class Diagram15

4.4.3 Activity Diagrams16

4.5User Interface20

5.**Evaluation/Verification Plan**26

5.1Unit Testing26

5.2Functionality Testing27

6.References29

1. **Introduction**

The lifestyle of children, adolescents, and adults poses serious health challenges, in our generation, with technological development, we move less and eat more junk.

What constitutes an unhealthy lifestyle, causes weight gain and disease. Control and manage our health, there are many healthy-life style applications in the market. They help people lose weight, and save their level of sugar, blood pressure, and so on. There are many doctors advising patients to keep a healthy lifestyle, they are sent to various applications aimed at improving their lifestyle. But there is no correspondent application that helps doctor to build healthy lifestyle programs. In Addition to building a plan for the doctor, there are no options for monitoring the patient. In the project, we are going to create a new app that makes the building of healthy lifestyle programs. The app will allow doctors to create or reuse an elementary block of lifestyle actions and construct different individual programs from them. Will provide possibilities for automatic adaptation of the directives for specific patients, as well as feedback and results tracking. Using the healthy lifestyle program will also be realized on the patient's side. The main challenge of the project is developing corresponding functionality of the doctor and patient sides, UX/UI, and quick, safe, and scalable tool with good protection of personal data. The app is going to implicate web technologies, the front-end will be realized with angular or react-native, and the back-end will be done using node JS language. The data will be stored with the MongoDB system.

**2.Related work and Background**

**2.1 Background**

**-** Overweight pregnant women

With obesity as a worldwide epidemic, perinatal overweight ness and obesity have been widely considered. Approximately 50% of women experience excessive gestational weight gain[[1]](#footnote-1).  Unhealthy lifestyle patterns are critical factors influencing perinatal overweightness and obesity. Behavioral change depends on the discontinuation of an unhealthy lifestyle and the formation of a new healthy lifestyle[[2]](#footnote-2). Pregnancy is a crucial stage to remaining healthy for the sake of pregnant women and their unborn babies. To obtain a balanced diet, pregnant women should eat a variety of food, including fruits, vegetables, rice, meat, and milk, and their alternatives. With regard to physical activity, pregnant women should walk 10,000 steps a day (4-5 miles, depending on stride length) or do a minimum of 30-min moderate physical activities for 5 to 7 days a week. Mobile apps create new opportunities to set behavioral goals, provide healthy lifestyle counseling, and facilitate self-monitoring of pregnant women’s goal-directed behavior.

**-** Lifestyle among adolescents and students

European adolescents and students tend to have low levels of physical activity and eat unhealthy foods, and the prevalence of overweight and obesity has increased, which poses a public health challenge. the prevalence of overweight and obesity within this group has increased in many countries. Children and adolescents at ages 9, 11, 12, and 15 years found that physical activity levels decreased as children enter adolescence[[3]](#footnote-3). The annual overweight rates are increasing continuously, which demonstrates the need for public intervention. Mobile apps play an important role in their daily lives, suggesting their potential to be used in health-promoting strategies.

**-** Diabetes

The number of people with diabetes worldwide has more than doubled during the past 20 years. One of the most worrying features of this rapid increase is the emergence of type 2 diabetes in children, adolescents, and young adults. Lifestyle factors related to obesity, eating behavior, and physical activity play a major role in the prevention and treatment of type 2 diabetes. Lifestyle intervention studies have consistently shown that quite modest changes can reduce the progression from IGT to diabetes by 50–60%.

- Sleeping affection on mental health

Most adults need at least 7 hours of sleep each night. While this may be fine for a day or two, not getting enough sleep over time can lead to serious health problems and make certain health problems worse. Adults who sleep less than 7 hours each night are more likely to say they have had health problems, including heart attack, asthma, and depression. Some of these health problems raise the risk of heart disease, heart attack, and stroke. These health problems include high blood pressure, diabetes, and obesity.

**2.2 Related works**

In terms of similar work and the development of the internet capabilities and functionality of cell phones. Today there are many apps that deal with managing a healthy lifespan on the one hand or communication apps between doctors and patients on the other hand. One of the first works in the field is a language application in 2005, which at the time was only a website on the Internet, its purpose was to track the owners of another company who were running the activities of his company through their sports activities, soon the purpose of the website was changed to improve and maintain a healthy lifestyle.

**2.2.1** **MyFitnessPal** - MyFitnessPal is an app that allows users to record food consumed and monitor their food composition through a food database. The food database is compiled by both MyFitnessPal (based in America) and users (who live globally). The primary aim of the app is to monitor energy intake and exercise output to help you achieve a range of nutrition, fitness and weight goals. Additional features of the app allow for weight and exercise monitoring and analysis of energy intake distribution through the day, monitoring of limited micronutrients (sodium, potassium, vitamin A, vitamin C, calcium, iron) and access to an online community.

**2.2.2** **FatSecret** - FatSecret is a MHealth app for self-monitoring food, physical activity, and weight. A bar scanner helps track packaged foods. Nutritional intake (Calories, Protein, Fat and Carbohydrate) is included in nutritional analysis. Calories expended through exercise and body weight is also provided. FatSecret also displays the data by meal and snack eaten so that the provider can view meal patterns. For consumers and patients, it also has library of recipes, journals, groups, and forums. FatSecret Professional allows clinicians to view patient tracking information online and view data using various visual summaries. Providers will receive a daily email summarizing calories, fat, carbs, protein, exercise, and weight for all patients/participants who tracked their data the day before. Providers can use the app to write encouraging or informational messages to their patients. Both can be used on smartphone (iPhone or Android), tablets and/or web.

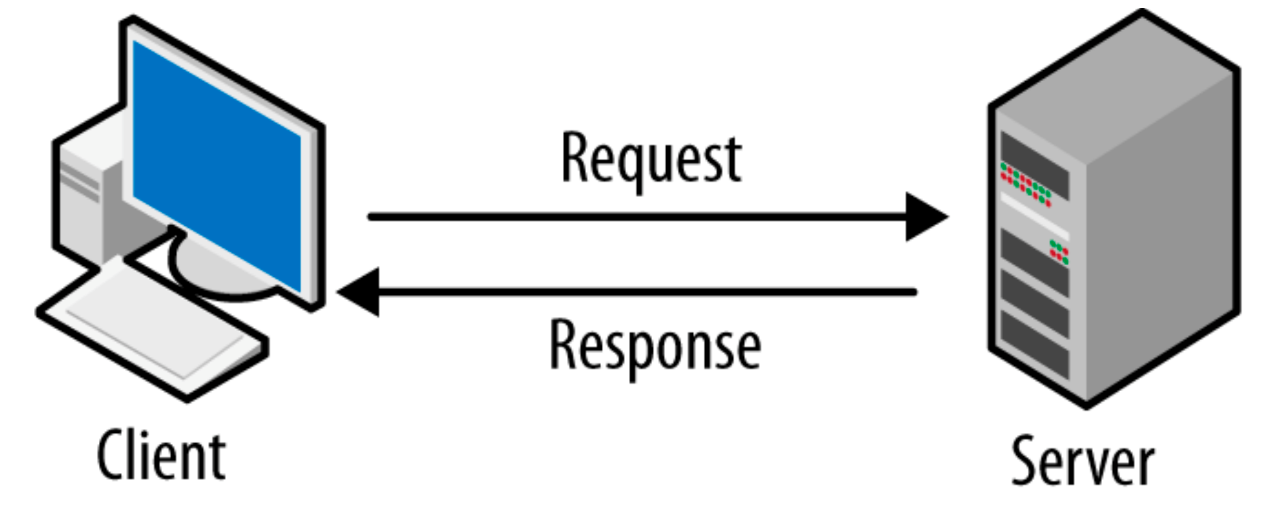
**2.3 Agile Development**

For the development stage, we had to choose the most suitable software development methodology for our project, we have researched and analyzed a couple of project development methodologies such as Agile and Waterfall. our conclusion is that Agile is the most appropriate development methodology for our project.  
Agile methodology is an iterative approach to project management and software development, which breaks down projects into small pieces. These project pieces are completed in work sessions that are often called sprints. Sprints generally run anywhere from a few days to a few weeks. These sessions run from the initial design phase to testing and quality assurance (QA).   
The main benefit of Agile development methodology lies on the flexible approach to product development, the development team are responsive to changes, even at the last minute, and can adapt to it without much disruption.  
Hence, the use of the Agile methodology will allow us to release an initial version of the system.

**2.4 Client – Server**

Client-server denotes a relationship between cooperating programs in an application, composed of clients initiating requests for services and servers providing that function or service.

Clients, also known as service requesters, are pieces of computer hardware or server software that request resources and services made available by a server.

A server is a device or computer program that provides functionality for other devices or programs. Any computerized process that can be used or called upon by a client to share resources and distribute work is a server.

**What is the Client-Server Model?**  
  
The client-server model, or client-server architecture, is a distributed application framework dividing tasks between servers and clients, which either reside in the same system or communicate through a computer network or the Internet.  
The client relies on sending a request to another program in order to access a service made available by a server.  
The server runs one or more programs that share resources with and distribute work among clients.  
The client server relationship communicates in a request–response messaging pattern and must adhere to a common communications protocol, which formally defines the rules, language, and dialog patterns to be used. Client-server communication typically adheres to the TCP/IP protocol suite.   
TCP protocol is the best way to distribute application data into packets that networks can deliver, transfers packets to and receives packets from the network, and manages flow control and retransmission of dropped or garbled packets.   
IP is a connectionless protocol in which each packet traveling through the Internet is an independent unit of data unrelated to any other data units.  
Client requests are organized and prioritized in a scheduling system, which helps servers cope in the instance of receiving requests from many distinct clients in a short space of time.  
  
**Benefits of Client-Server Computing**

* A single server hosting all the required data in a single place facilitates easy protection of data and management of user authorization and authentication.
* Resources such as network segments, servers, and computers can be added to a client-server network without any significant interruptions.
* Data can be accessed efficiently without requiring clients and the server to be in close proximity.
* All nodes in the client-server system are independent, requesting data only from the server, which facilitates easy upgrades, replacements, and relocation of the nodes.

**2.5 Cloud Computing**

Cloud computing is the delivery of computing services—including servers, storage, databases, networking, software, analytics, and intelligence—over the Internet (“the cloud”) to offer faster innovation, flexible resources, and economies of scale. To check our application, we will rent virtual computers that will run our application.

**3. Expected Achievements**

**3.1 Project Goals**

We are going to develop a user-friendly and cross-platform application allowing:

1.Building and sharing healthily lifestyle programs.

2.Tracking of the program's implementation and feelings.

3.Interaction between specialists and users.

**3.2 Unique Features**

1.The app will contain medical information about the patients.

2.The app will be able to automatically adjust the lifestyle programs to the health conditions of the patients.

3. The app will alarm the specialist and the patient when user’s health changes required special attention.

**4. Engineering Process**

**4.1 Process**

We've created a well-organized work plan that covers all the processes, starting with the learning process and progressing through the development process until we arrive at a final product that fits the system's objectives.  
  
At first, we had to analyze the problem that needed to be solved. our first task was to precisely define the problem, in other words, what is the problem that needs to be addressed, who is the product we are developing for, and how important is it to find a solution?

To achieve all the goals, we conducted a survey with potential stakeholders:

What is the current app in the market offers, and if it been used daily base.

What are missing in the lead apps for healthy lifestyle for people who suffering from medical condition.

Who people that will thinks about concerting all the specialist in one platform.

Following that, we thoroughly defined the system's requirements.  
  
The next step was to look into potential solutions, we had to investigate the various factors about each of the potential solutions, we examined the advantages and the disadvantages which are relevant to each solution.   
  
our project, we realized that Agile is the most appropriate development methodology for our project.

**4.2 Requirements**:

**Functional:**

|  |  |
| --- | --- |
| The system will store data in database | 1 |
| The system will be able to connect between patient and Specialist | 2 |
| The system will be able to create a new program/library | 3 |
| The system should be able to use in exists program | 4 |
| The system should support in publish programs in a social network | 5 |
| The system will be based on client-server module | 6 |
| The system shall have the option to rate programs | 7 |
| The system shall have an option of chat between patient and Specialist | 8 |
| The system would have option to search program by filters | 9 |
| The system will send notifications to patient for complete their daily goals | 10 |
| The system shall have option for the user to mark the goals which he completed | 11 |
| The system will be able to create a new user in the system | 12 |
| The system allows the specialist the ability to approve/deny users to from using his programs | 13 |
| The system allows the user to look on the future and past goals via the calendar | 14 |

**Non-functional:**

|  |  |
| --- | --- |
| 1 | The system will have friendly user interface |
| 2 | The system will be easy to work with it |
| 3 | The system will supply connection to social network of the user |
| 4 | Rating a program will be by star 1-5 and to add comments |
| 5 | The system will show every patient his daily goals |
| 6 | Only specialist can create a new program |
| 7 | User can add his physical features for his profile |
| 8 | Every user can login as patient or specialist |
| 9 | To run a several program in parallel required approval of all specialist |

**4.3 Technology choices**:

The problems we are facing in the implementation of the system described above, are numerous and very robust. The first obstacle of note would be the Server-Side or as it called the Back-End. The collection of code and logic that will reside on the server and will provide the engine of the system itself.

A creative approach to this problem is an optimistic User Interface that ‘hopes’ that the queries and update requests it sent has gone through and shows the updated information without waiting for the response of a server. This, coupled with asynchronous API on the server, can prove itself as a seemingly fast and frugal product that is updated instantly.

Another problem is the platform choice. Social networks are in dire need of users (the network will not be very social otherwise). Meaning that the network should be available on as many platforms as possible (Mobile, PC)

This can be solved by implementing the correct framework. A web-app is a sufficient solution to a multi-platform application, considering that modern devices can at least run a simple web browser. In the scope of the project, we should implement the web-app in Android and Website variant to prove the concept. But the same can be done with any modern smart device.

According to the 2018 Mary Meeker report[[4]](#endnote-1), the number of hours spent on the internet is still increasing each year, but the split between desktop and mobile is becoming more and more pronounced. In 2016, Americans were spending 3 hours or more per day on mobile (that’s 10 times more than in 2008) and just 2.2 hours per day on a desktop or laptop (no change since 2008).

The number of mobile app downloads each year has been steadily increasing. In 2017, there were 178 billion app downloads. That number is projected to grow to 205 billion this year, and 258 billion in 2022—a 45 percent increase over five years.

With this information in mind, our system should be available on both mobile and desktop devices.

In order to keep the project doable in a short time most of these features will be implemented by existing open-source frameworks and libraries.

The problem of scalability, speed, and framework reduce us to use specific tools. For speed of development, we decided our development stack will consist of only one language, JavaScript (JS). In the late years, JavaScript has become a booming industry standard[[5]](#endnote-2). With the introduction of NodeJS and the V8 google interpreter, the ability to run scripts written in JavaScript server-side has become possible and the allure of single full-stack language has fueled many developers to create many frameworks to create full-fledged web-apps. Ranging from front-end to back-end solutions.

**4.3.1 Back-End - NodeJS & MongoDB**

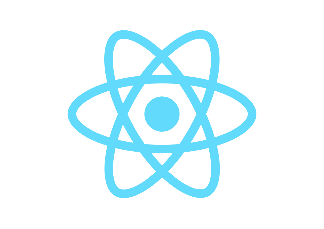
Running JavaScript is made easy with the NodeJS interpreter, which is creating a single-threaded async process on the system which parses JavaScript files and runs them, implementing many popular features like standard output/input and file read/write. Another feature worthy of note is the “npm packages” which allow functionality similar to python modules, by adding a JavaScript Node Package, a collection of open source code supported and written by the community[[6]](#endnote-3) is added to your project.

MongoDB is a Database Management system that allows speed, security, and scalability. That makes it the perfect candidate for a back end Database System. With the added perk of simple integration under a NodeJS process. A single running thread will run on the server as the daemon for control over data. Allowing the server to also provide other services and undercut extra operation costs.

With the Node JS Runner in the background, a simple Python script can take over the task of the comparison between user profiles and feature selection.

When a user queries the DBMS for new feed elements, the NodeJS process will receive a vector in the query. The server will compare the received vector with the activity collection vectors. Finally, it will return the most compatible feed elements.

**4.3.2 Front-End - ReactJS**

ReactJS[[7]](#endnote-4) is a User Interface front-end JavaScript framework built by Facebook. React is currently maintained by a community of developers, private and corporate. The framework is a base for a single-page web application. And used for creating Simple User Interface components and stitching them together.

React will allow extensive code reuse and modularity while having the open-source community support of the code.

**4.4 Diagrams**

**4.4.1 Use case diagram**

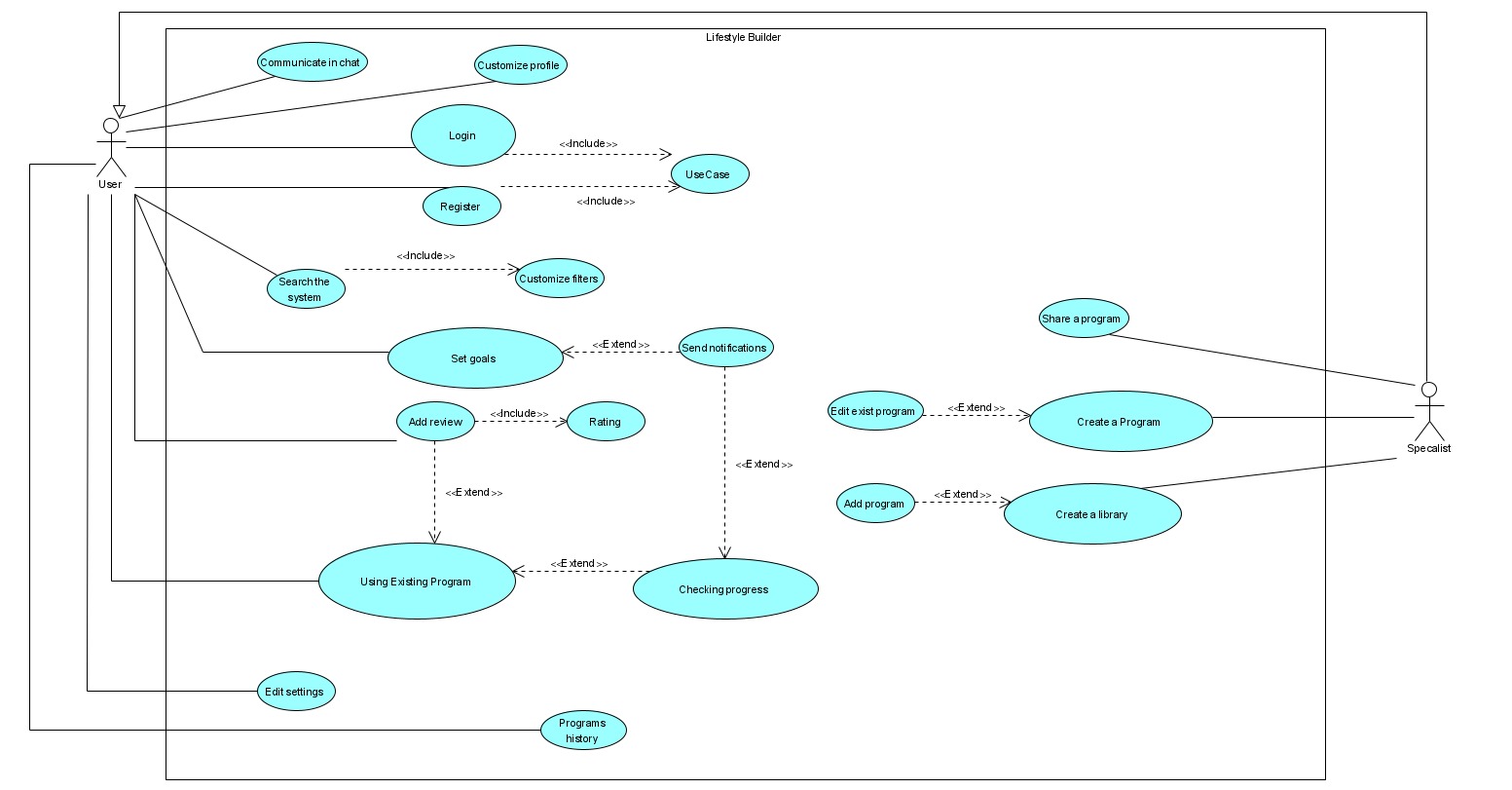


Fig1: Use Case diagram

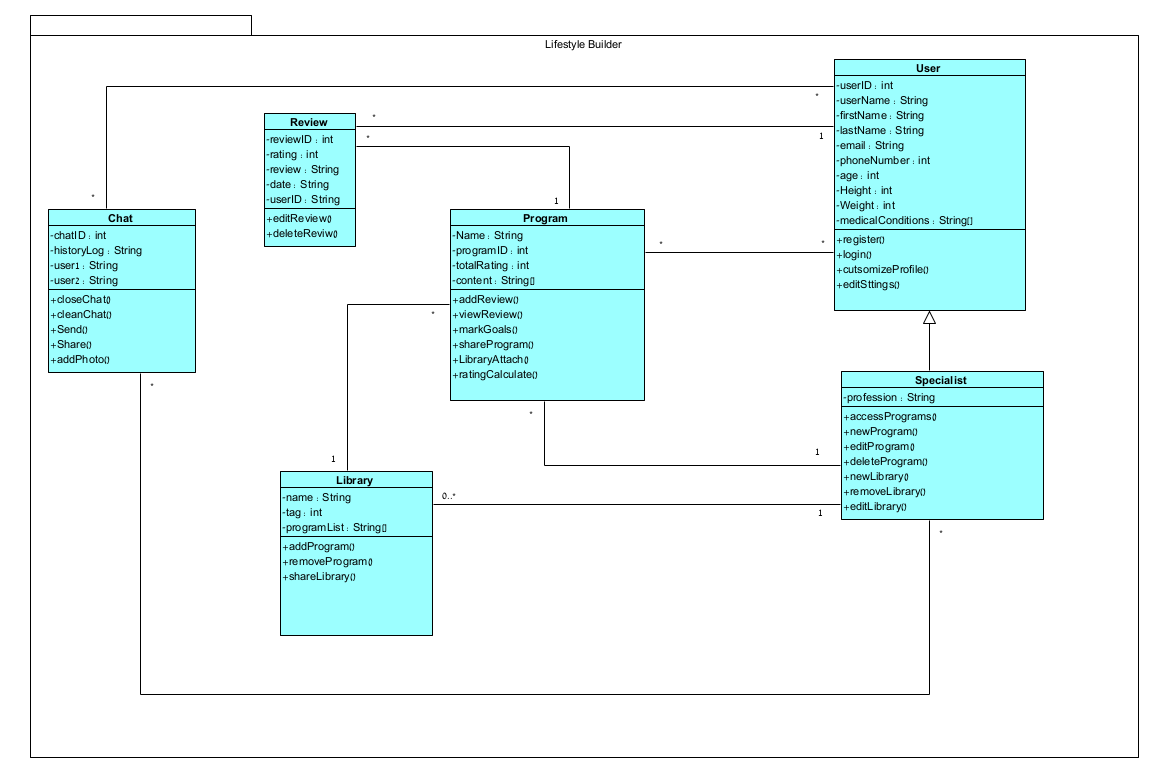
**4.4.2 Class diagram**

Fig2: Class diagram

**4.4.3 Activity diagram**

**Create Program**:

1. The Specialist login into the application
2. The system validates the login and presents the homepage to the specialist.
3. The specialist decides to create a new program.
4. The specialist adding content to the program.
5. The program is being saved in the database.
6. The specialist needs to choose if he wants to save the program in an existing library
7. The answer is "Yes" jumped to step 10.
8. The answer is "No " the specialist is creating a new library.
9. The new library is saved in the database.
10. The program is saved in the chosen library.
11. The system will ask if the specialist wants to share the program.
12. The answer is "No" jump to step 14.
13. The answer is "Yes" the specialist is sharing the program through any social media.
14. End of process.

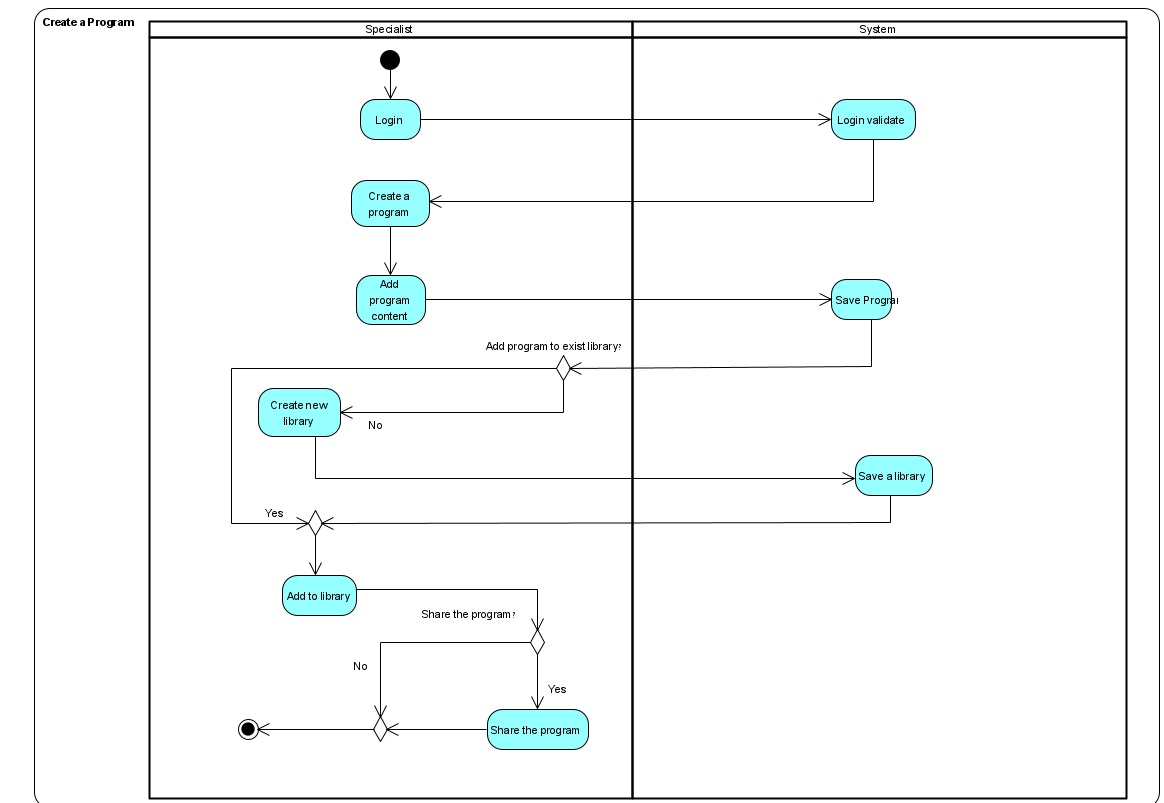


Fig3: Activity diagram 1

**Daily Goals**

1. The user login into the application
2. The system validates the login and presents the homepage to the user.
3. The user checks if he has any active daily program.
4. The answer is "No" jump to step 14.
5. The answer is "Yes" the user selects an active program.
6. The user checks his daily goals.
7. The system uploads his goals from the database.
8. The user attempts a relevant medical assignment.
9. The user documenting the assignment.
10. The data is saved in the database.
11. The system asks the user if he finish the program goals.
12. The answer is "Yes" jump to step 3.
13. The answer is "no" the user mark the finished goals and jumps to step 8.
14. End of process.

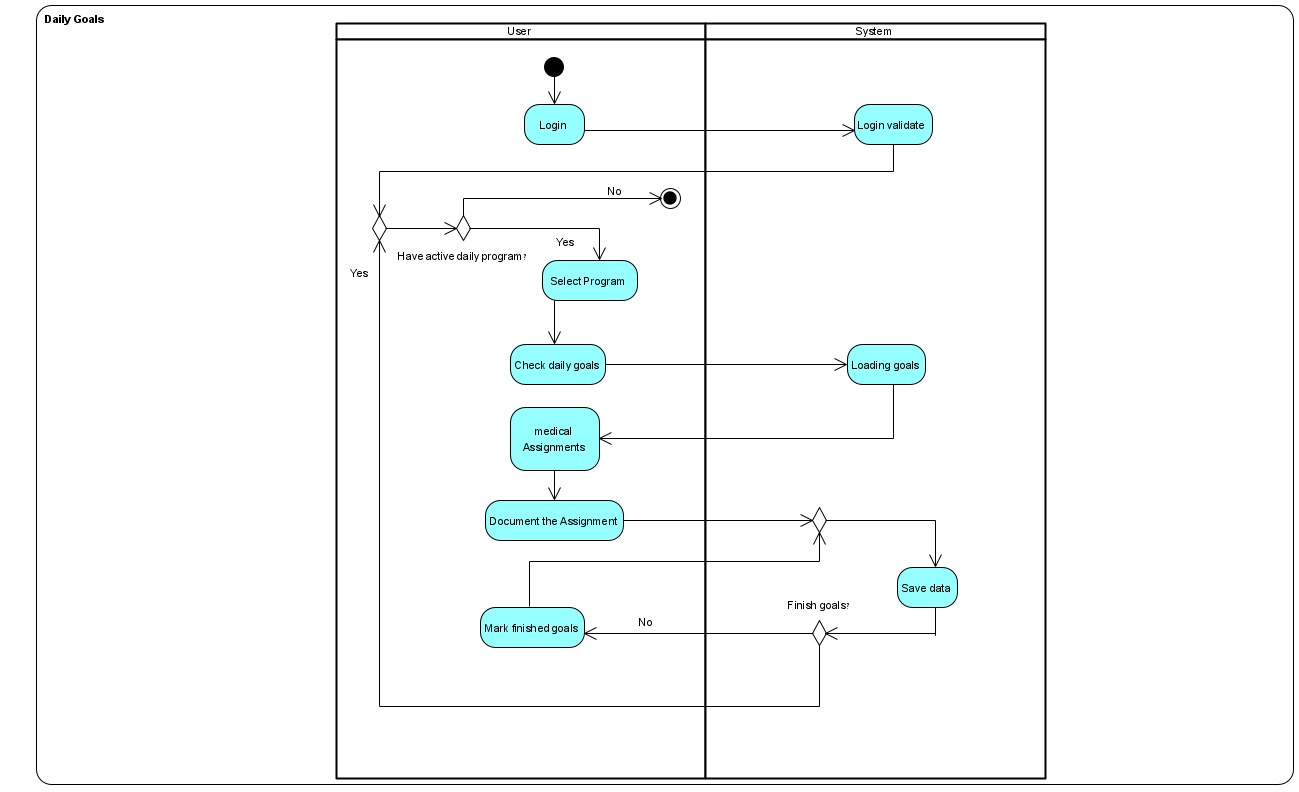


Fig3: Activity diagram 2

**Review a program**

1. The user login into the application.
2. The system validates the login and presents the homepage to the user.
3. The user selects any program.
4. The system asks what he wants to do.
5. The answer is "Nothing" jump to step 17.
6. The answer is "Select other programs" jump to stem 3.
7. The answer is "Watch reviews" jump to step 15.
8. The answer is "Review a program" the user must do two things to continue.
9. First the user must rate the program.
10. Second the user has to review the program.
11. The new total rate of the program is calculated, and the review is saved.
12. The system asks if the user wants to choose another program.
13. The answer is "Yes" jump to step 3
14. The answer is "No" jump to step 15.
15. The user watches the existing program's reviews.
16. After the user finishes with the reviews jump to step 4
17. End of process.

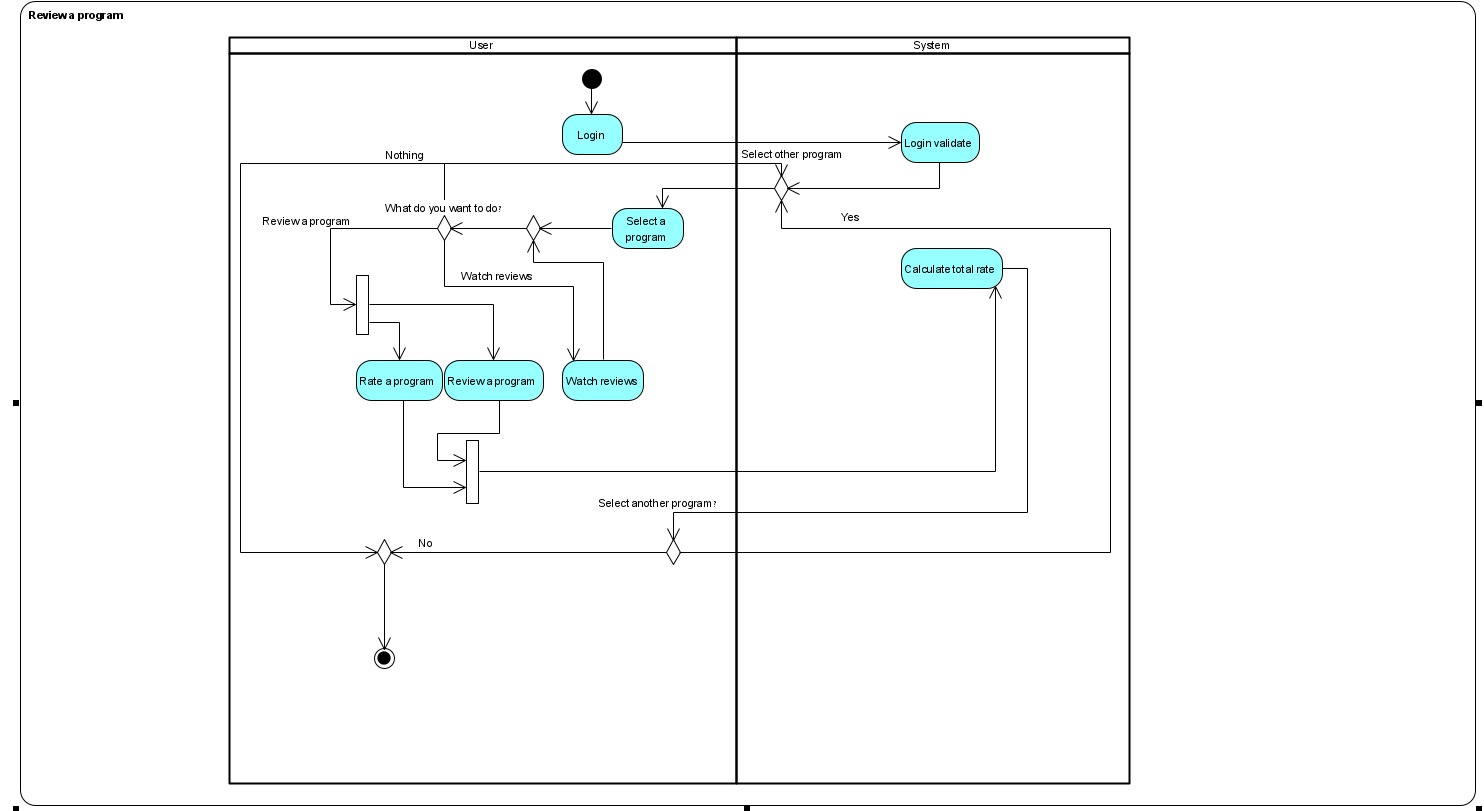


Fig3: Activity diagram 3

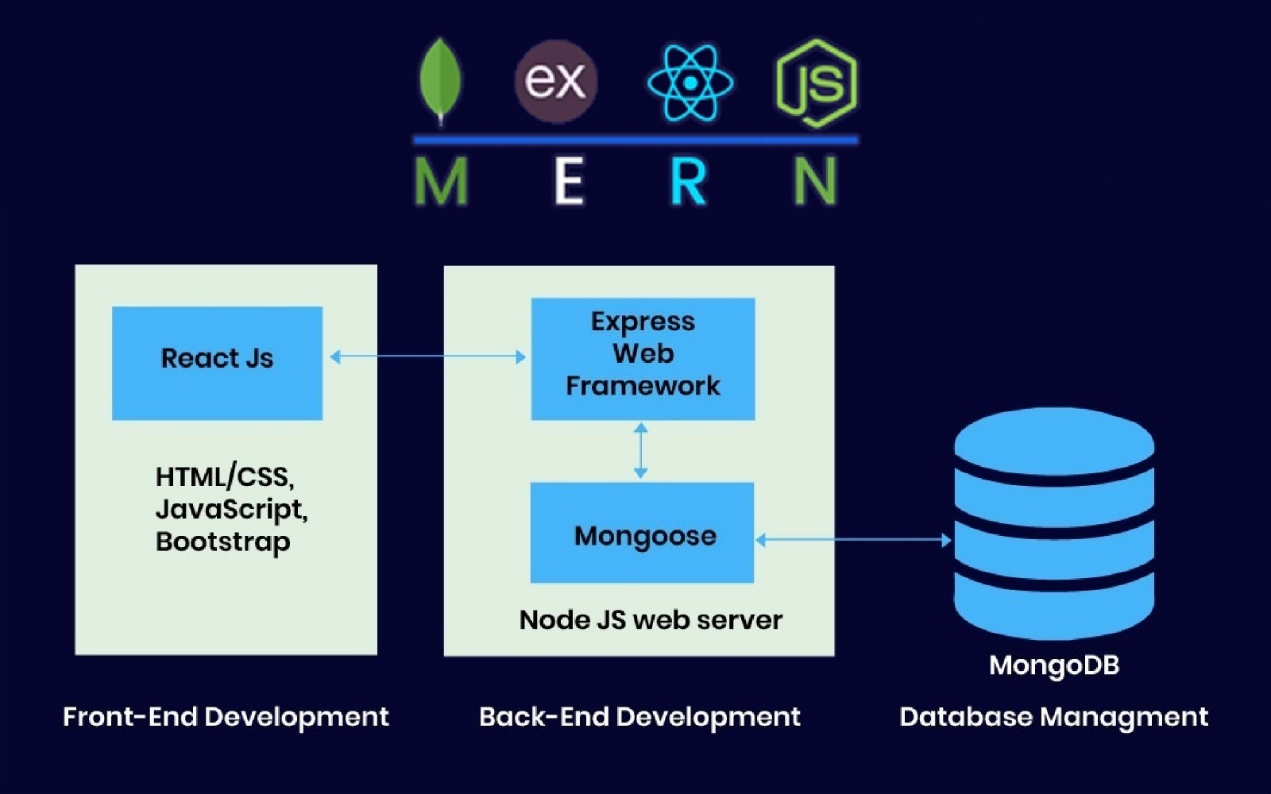


Fig6: Software Architecture Diagram

**4.5 User Interface**

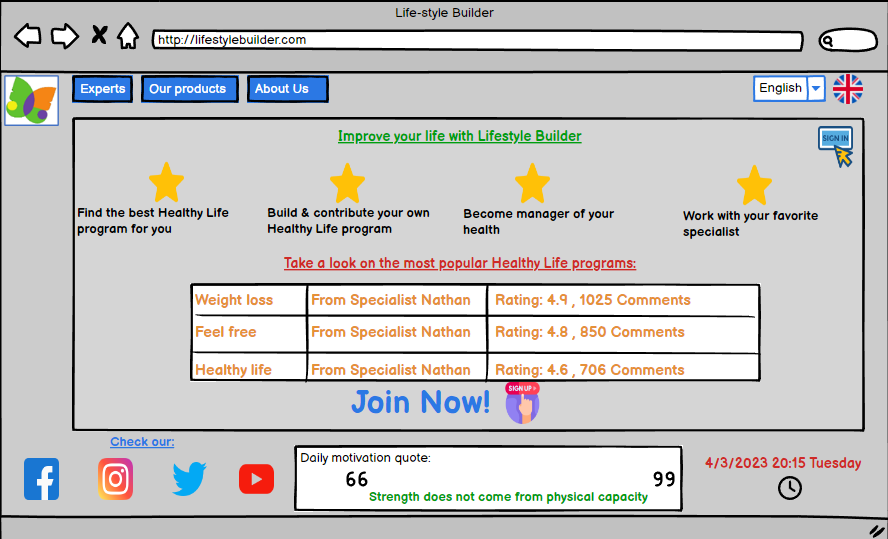


Fig6: Pre login homepage

This is the starting point, Unless the user using cookies and the remember me button.

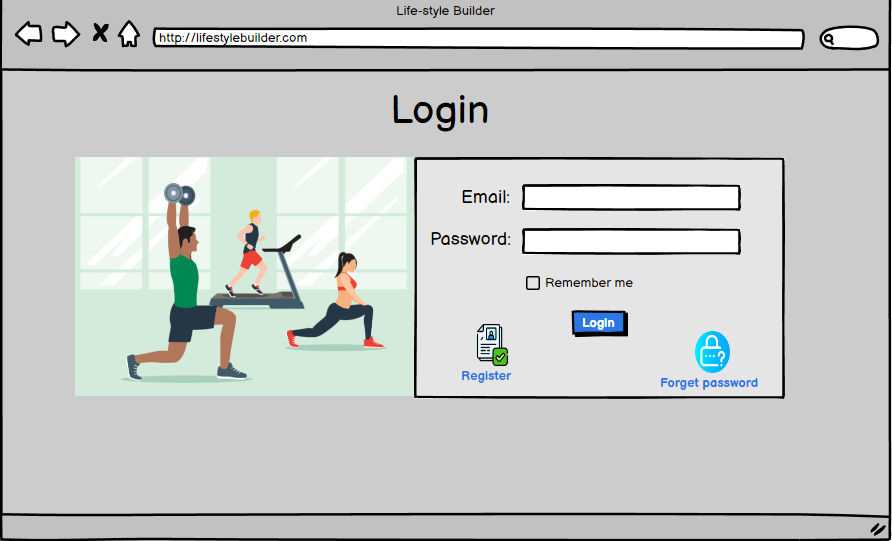


Fig7: Login page

In the registration page all fields or required except the image which is optionally

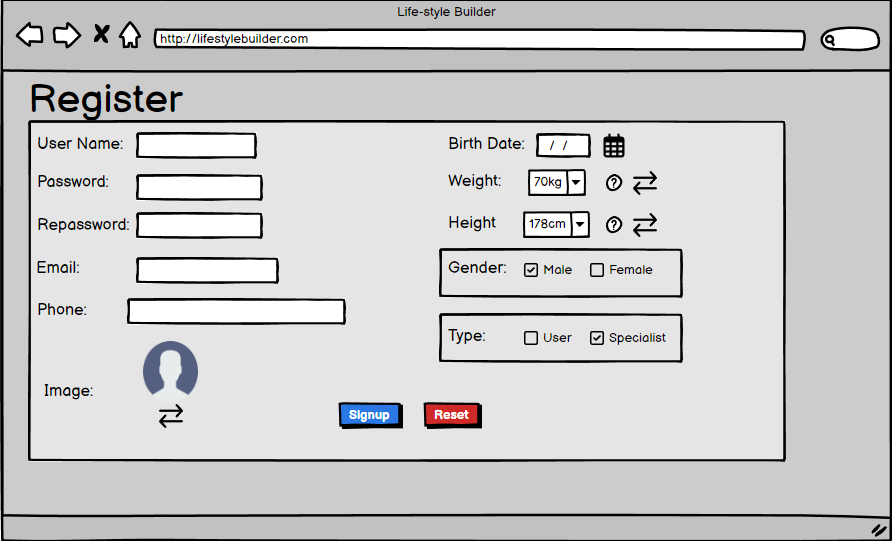
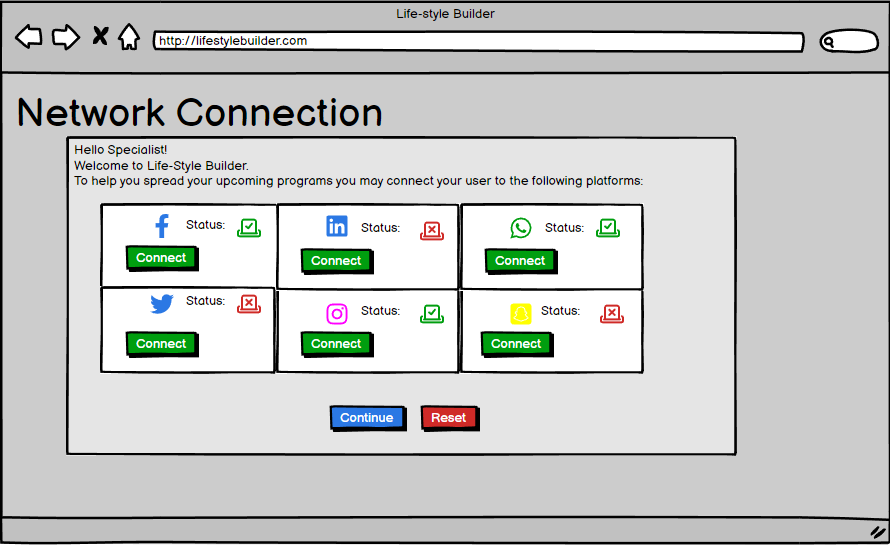


Fig8: Registration

After succeed to register the Specialist next step is to connect is social media so he can share his libraries and programs

Fig9: Network Connection

Home page shown as a Specialist, His home screen consists of recommended programs, reviews given to programs he has prepared, search and preparation of new programs, etc.

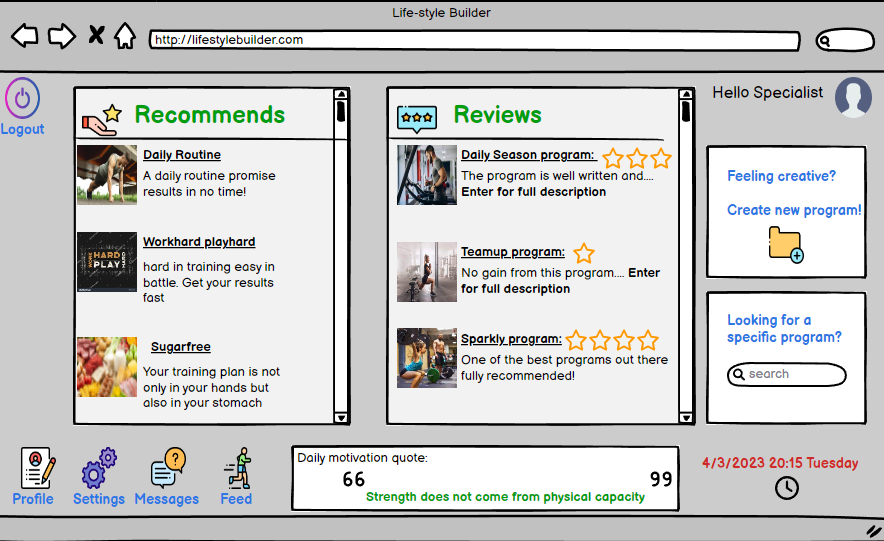


Fig10: Home Page - Specialist

Home page shown as a User, His home screen consists of recommended programs, list of popular specialists, active daily programs, etc.

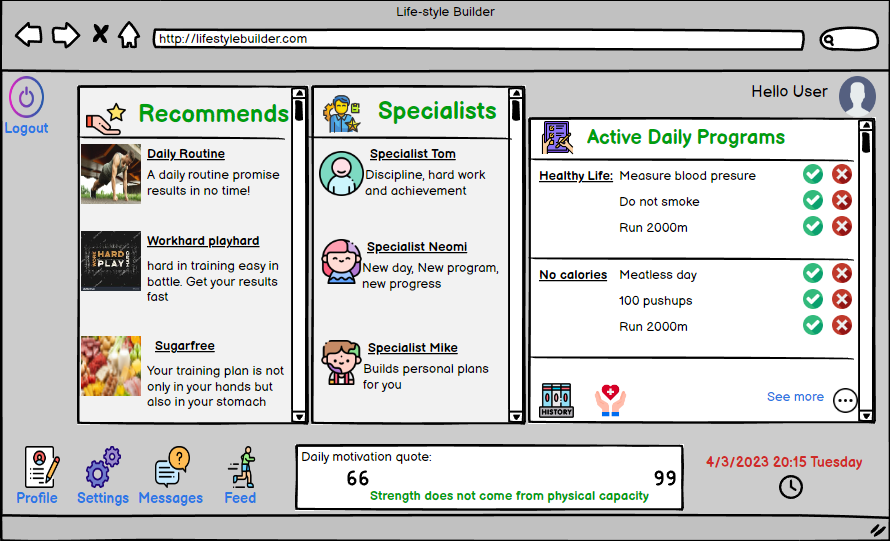


Fig11: Home Page - Specialist

The profile page that is displayed in a default format, that is, how others will see your profile and vice versa. Through the profile you can make contact and see the user's statistics

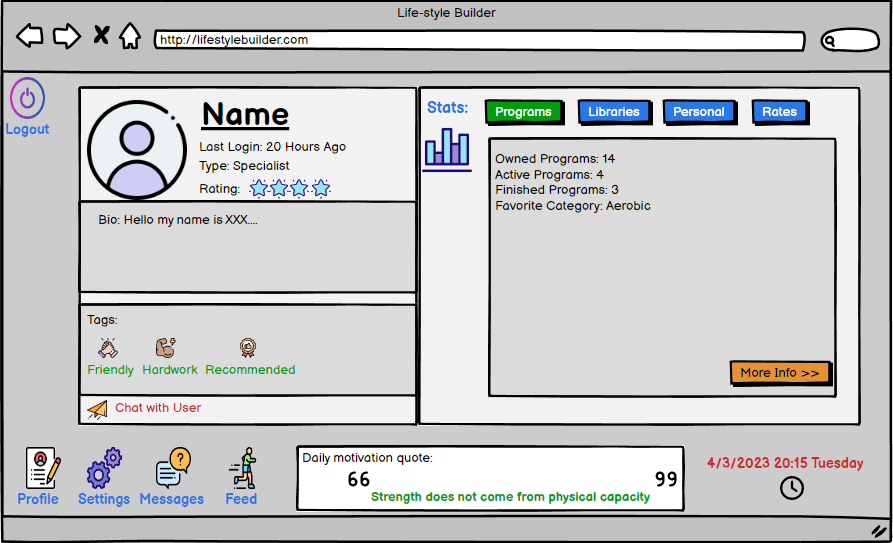


Fig12: Profile

Library page of the user which shown all of his libraries, how much program each library has and if there any active program inside the library.  
Click on one of the libraries to enter the specific library to choose any program.

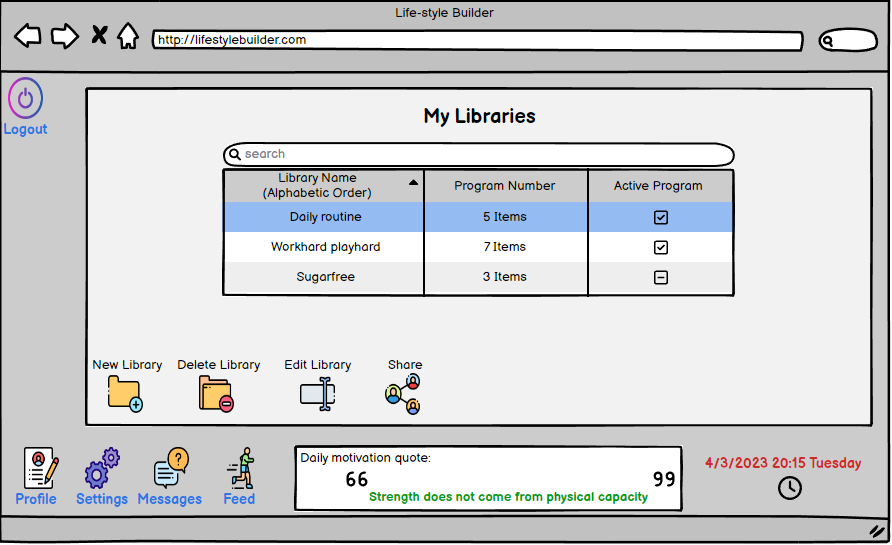


Fig13: Library

Specific library page of the user which shown all of the programs inside of the library, each program has daily goals tab, how long the goals takes and the rating that users gave to the program. Click one of the programs to check his content.

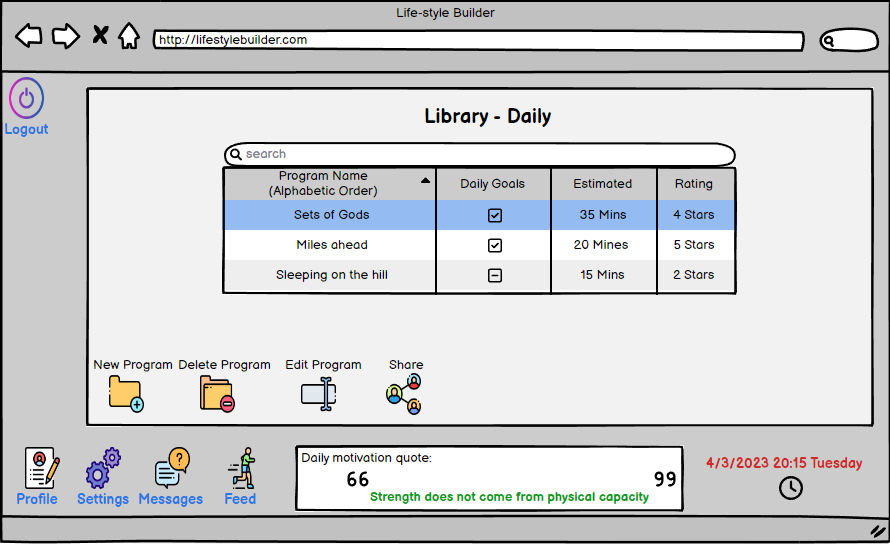


Fig14: Specific Library

Create new program, each program has a name and a type. To the program you can add unlimited tasks, click on add to open a tab that contains options for realization, you need to add type, frequency, time and marking of each part of the program.

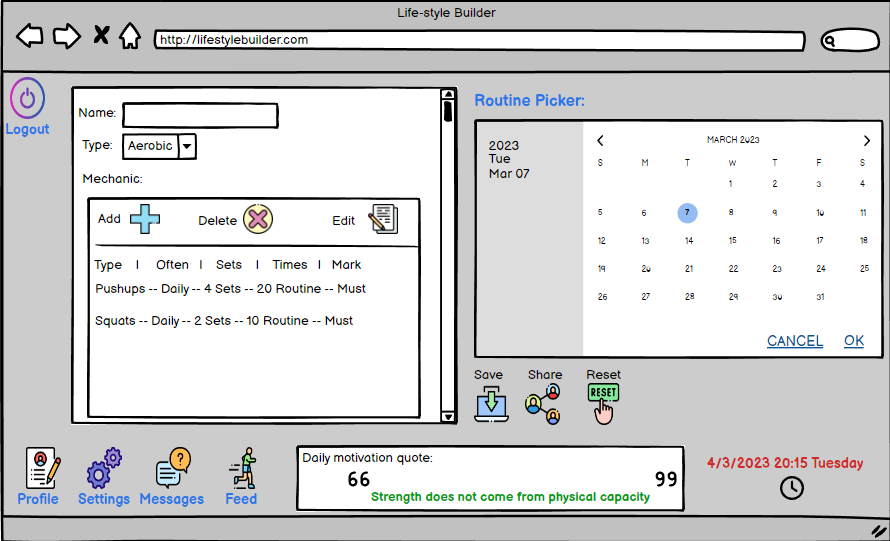


Fig15: New Program

Chat with whoever you desire

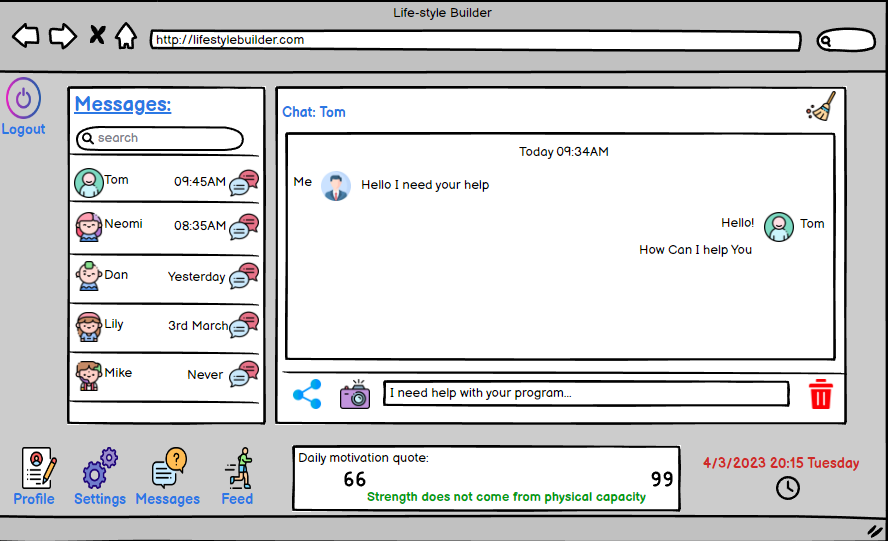


Fig16: Messages

**5. Evaluation/Verification Plan**

To be able to evaluate the performance of our system, we will perform comprehensive tests that include system unit tests and functional tests through which we will verify the integrity of the system, and that it indeed performs and meets the established requirements.

The tests will be performed on threshold conditions and prerequisites as well as on communication between the server and the client.

**5.1 Unit Testing:**

|  |  |  |
| --- | --- | --- |
| **Registration Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1 | Enter the empty required field | Error alert: "Please fill in all the required fields" |
| 2 | Wrong password pattern | Error alert: "Please retype a password that matches the password pattern" |
| 3 | Password verification does not match | Error alert: "The passwords do not matches please retry again" |
| 4 | Wrong email pattern | Error alert: "Please retype the email that matches the email pattern" |
| 5 | Used email | Error alert: "The email you entered is already in use" |

|  |  |  |
| --- | --- | --- |
| **Login Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1 | Wrong email pattern | Error alert: "Please retype the email that matches the email pattern" |
| 2 | Wrong password pattern | Error alert: "Please retype a password that matches the password pattern" |
| 3 | The email and password don't match | Error alert: "The email and the password don't match please try again" |
| 4 | Empty required field | Error alert: "Please fill all the fields to login" |

|  |  |  |
| --- | --- | --- |
| **New Program Testing** | | |
| **Expecting Result** | **The Test** | **No.** |
| Error Alert: "A program has to have at least one objective" | Empty program | 1 |
| Error Alert: "The program must have a name" | Empty program name | 2 |
| Error Alert: "A program with the same name already exists" | Existing library | 3 |

|  |  |  |
| --- | --- | --- |
| **New Library Testing** | | |
| **Expecting Result** | **The Test** | **No.** |
| Error Alert: "The library must have a name" | Empty name | 1 |
| Error Alert: "A library with the same name already exists" | Existing library | 2 |

|  |  |  |
| --- | --- | --- |
| **Review Testing** | | |
| **Expecting Result** | **The Test** | **No.** |
| Error alert: "Review must have a rate" | Empty rating | 1 |
| Error alert: "The review must have at least one word" | Empty review | 2 |

**5.2 Functionality Testing:**

|  |  |  |
| --- | --- | --- |
| **UI Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1. | Buttons check | Every button in the application is working properly |
| 2 | Navigation check | With every redirect, the application is working as intended |
| 3 | Database check | Showing the correct content in the application as the database |
| 4 | Page loading | Make sure that every page time loading is optimistically |
| 1. | Buttons check | Every button in the application is working properly |

|  |  |  |
| --- | --- | --- |
| **Program Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1. | Adding objectives | Adding any objective to the program is shown on the screen |
| 2 | Delete objectives | Removing any objective from the program is shown on the screen |
| 3 | Saving program | The program should be shown in the correct library and the database |
| 4 | Remove program | The program should be removed from the library and the database |
| 1. | Adding objectives | Adding any objective to the program is shown on the screen |

|  |  |  |
| --- | --- | --- |
| **Share Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1. | Platforms validate | The sharing is working in every platform connected to the application |
| 2 | Link validate | The reference link to the program is valid through the shared link |
| 3 | Share connection | The connection between the social media the user profile is working properly |
| 1. | Platforms validate | The sharing is working in every platform connected to the application |
| 2 | Link validate | The reference link to the program is valid through the shared link |

|  |  |  |
| --- | --- | --- |
| **Chat Testing** | | |
| **No.** | **The Test** | **Expecting Result** |
| 1. | Sending message | The message sent from one user to another user needed to be presented on both chat screen |
| 2 | Open new chat | The correct users that chatting is shown |
| 3 | Close chat | The chat is closed, and the screen switched to the last page |
| 4 | Chat History | Chat history is saved even if the chat close and reopen |
| 5 | Delete history | Removing all the chat history between the users |

**6. References**

-FatSecret - https://telehealthvillage.com/InnovationExchange/FatSecret.aspx

-MyFitnessPal Application- <https://www.healthnavigator.org.nz/apps/m/myfitnesspal-app/>

- Client Server <https://www.omnisci.com/technical-glossary/client-server>  
  
**-** Cloud Computing <https://azure.microsoft.com/en-us/overview/what-is-cloud-computing>  
  
**-**Agile Development <https://www.techtarget.com/searchcio/definition/Agile-project-management>

- Natasha Maclnnis, " Population Attributable Risk Fractions of Maternal Overweight and Obesity for Adverse Perinatal Outcomes, Phillippa Lally, "Promoting habit formation"

Phillip R Nadar, " Moderate-to-vigorous physical activity from ages 9 to 15 years"

- Does improving sleep lead to better mental health? A protocol for a meta-analytic review of randomized controlled trials- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5623526/

- Health Navigator Editorial Team. Reviewed By: "Frances Arenhold, Registered Dietitian"- https://www.healthnavigator.org.nz/apps/m/myfitnesspal-app

- Study reference: Turner-McGrievy GM, Wilcox S, Boutté A, et al. The Dietary Intervention to Enhance Tracking with Mobile Devices (DIET Mobile) Study: A 6-Month Randomized Weight Loss Trial [published correction appears in Obesity (Silver Spring). 2017 Dec;25(12 ):2156-2157]. Obesity (Silver Spring). https://www.healthnavigator.org.nz/apps/m/myfitnesspal-app/

-"npm | the of the modern development ...." https://www.npmjs.com/. Accessed 13 May. 2019.

"React." https://reactjs.org/. Accessed 13 May. 2019.-

1. Natasha Maclnnis, " Population Attributable Risk Fractions of Maternal Overweight and Obesity for Adverse Perinatal Outcomes" [↑](#footnote-ref-1)
2. Phillippa Lally, "Promoting habit formation" [↑](#footnote-ref-2)
3. Phillip R Nadar, " Moderate-to-vigorous physical activity from ages 9 to 15 years" [↑](#footnote-ref-3)
4. [↑](#endnote-ref-1)
5. [↑](#endnote-ref-2)
6. [] "npm | the of the modern development ...." <https://www.npmjs.com/>. Accessed 13 May. 2019. [↑](#endnote-ref-3)
7. [] "React." <https://reactjs.org/>. Accessed 13 May. 2019. [↑](#endnote-ref-4)