Virtual Health Circles
(VHC)

Requirements Report

Capstone COSC 499
Dawson Psychological Services
Dawson Group (B)

Pamal Mangat
Client Liaison

Sam Finnigan-Griffin
Tech Lead

Paul Zapote
Project Manager

Table of Contents:

High Level Project Description:

The aim of this application is to improve people's health whilst encouraging them to learn more about their own health. The target user groups for the application are the client's patients, the general public, and the client himself (*Dr. Kim Dawson*). The client will be able to view each patient's health progress and data as they continue to work with the application. The target users will begin by completing a questionnaire that determines where the user's current health state is, and how it can be presented in different perspectives such as graphs, charts, or concentric circles. These perspectives give the user a better understanding of how the different health domains interact and influence one another, and how the user can improve in certain domains. The five domains of focus in the application are *Physical, Mental, Emotional, Spiritual, and Social.* The application will also have incentives so that the user will keep improving, and be motivated to continue using the app through email reminders. These incentives and alongside other features will be discussed in greater detail throughout this report.

The application is intended to be a free-to-use service provided through the internet for any devices running modern javascript support and web browsers. The application's front-end development will be implemented with React, whilst the back-end development will be handled with Python 3.8. Majority of the work performed on the application will be hosted on github, please see the link below:

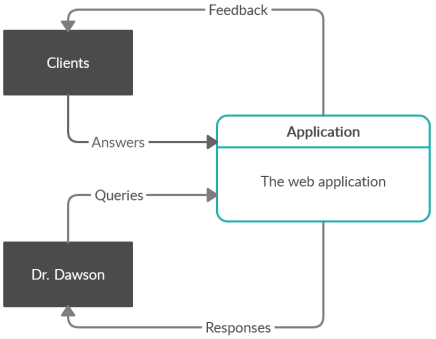
(*Link is private access only*)
https://github.com/SamFinni/DawsonB

The goal of the developing team is to hand a finished high-level prototype of the application running on a test database. The test database that we will be utilizing for the project will be the student database provided to the students of UBC. In addition, if the client wishes to port the application to another paid database service after the final submission, then we will implement with that in mind.

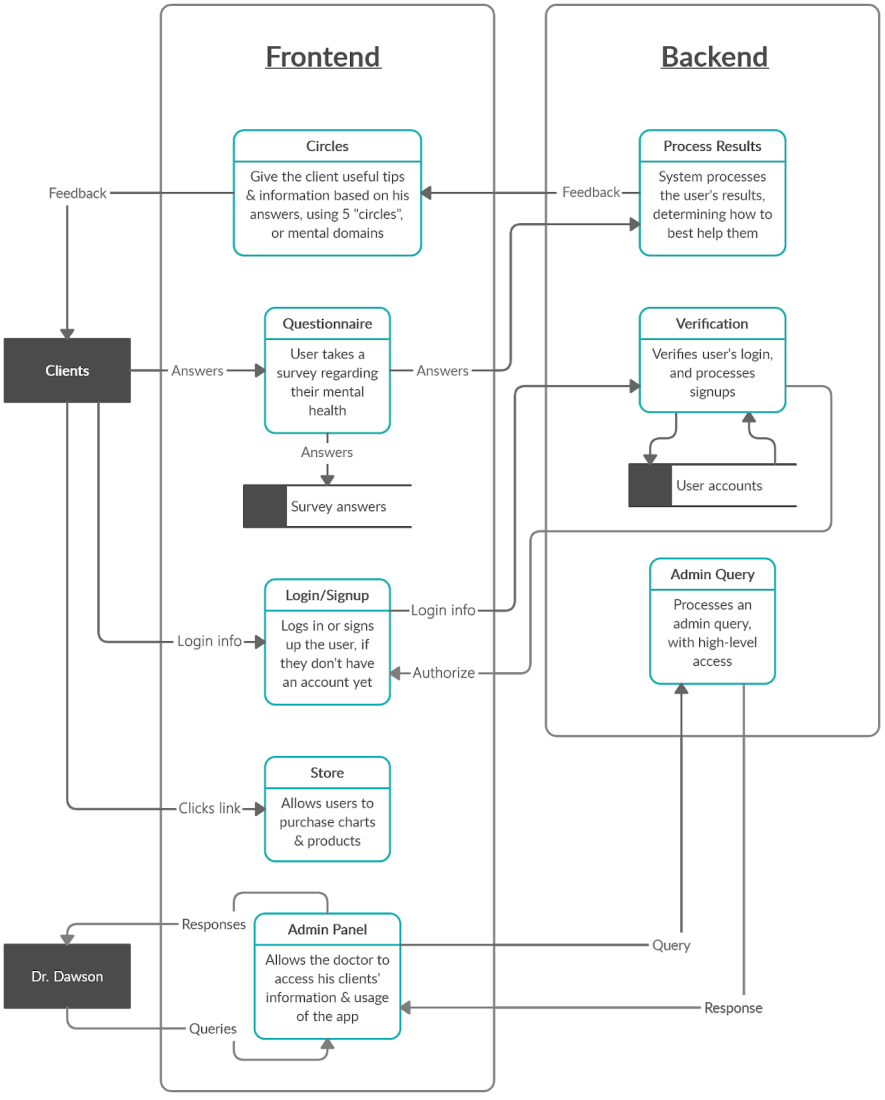
** *All images used within the report are available in the conclusion section.*

Overview of System Architecture:

Data Flow Diagram (DFD) - Level 0



Data Flow Diagram (DFD) - Level 1



Functional Requirements:

Peer Testing #1 - December 2nd, 2020

- Users should be able to delete, save or redo any section of the questionnaire at any time.
- Users should have the ability to choose which domains they wish to omit or focus their attention on; or which ones they wish to leave out.
- Users should have the ability to choose the color-scheme of the five domains provided.

Peer Testing #2 - March 3rd, 2021

- The application should present the user with appropriate links to activities in which the user can improve their domain ratings.
- Users should be able to customize and edit their avatar to their liking.
- Settings must be provided for the user to choose the frequency of reminders being sent out.

Final Submission - April 8th, 2021

- Users should be given the option to be directed to the web-shop.
  (Create link between application and web-shop)

Non-Functional Requirements:

Peer Testing #1 - December 2nd, 2020

- Users should be able to distinguish and see the perspective of the overlapping domain circles.

Peer Testing #2 - March 3rd, 2021

- Data must be hosted on a Canadian Database to ensure PIPEDA regulations are being followed. (*Data must remain within Canada*)
- Ensure that the data is safe and efficient to access.
- The database must follow the five database attributes:
  Reliability, Performance, Maintainability, Scalability, Usability

Final Submission - April 8th, 2021

- The application must ensure that it can support multiple-users simultaneously.

Target Components:

Peer Testing #1 - December 2nd, 2020

*** PT1 is mostly concerned with completing the front-end graphical design for the application itself.*

- Have a test database setup and running.
- Questionnaire should be completed and finalized.
- List of activities for each domain are finalized.
- Application provides links to activities to improve their health
- Users have the ability to modify/redo any section of the questionnaire.
- Users can set preferences on which health domains to focus on improving.
- Have visual representation of user's health from questionnaire

Peer Testing #2 - March 3rd, 2021

*** PT2 is mostly concerned with completing the back-end work & the avatar incentive feature.*

- Have push notifications working. (*With user preferences in mind*)
- Users should be able to customize and edit their avatar to their liking.

Final Submission - April 8th, 2021

*** Final submission is mainly concerned with ensuring that the applications core functionalities are working, so the only concern for final submission is ensuring the application will work on any future database the client wishes to utilize.*

- Ensure application works for any given SQL database.

Environmental Constraints:

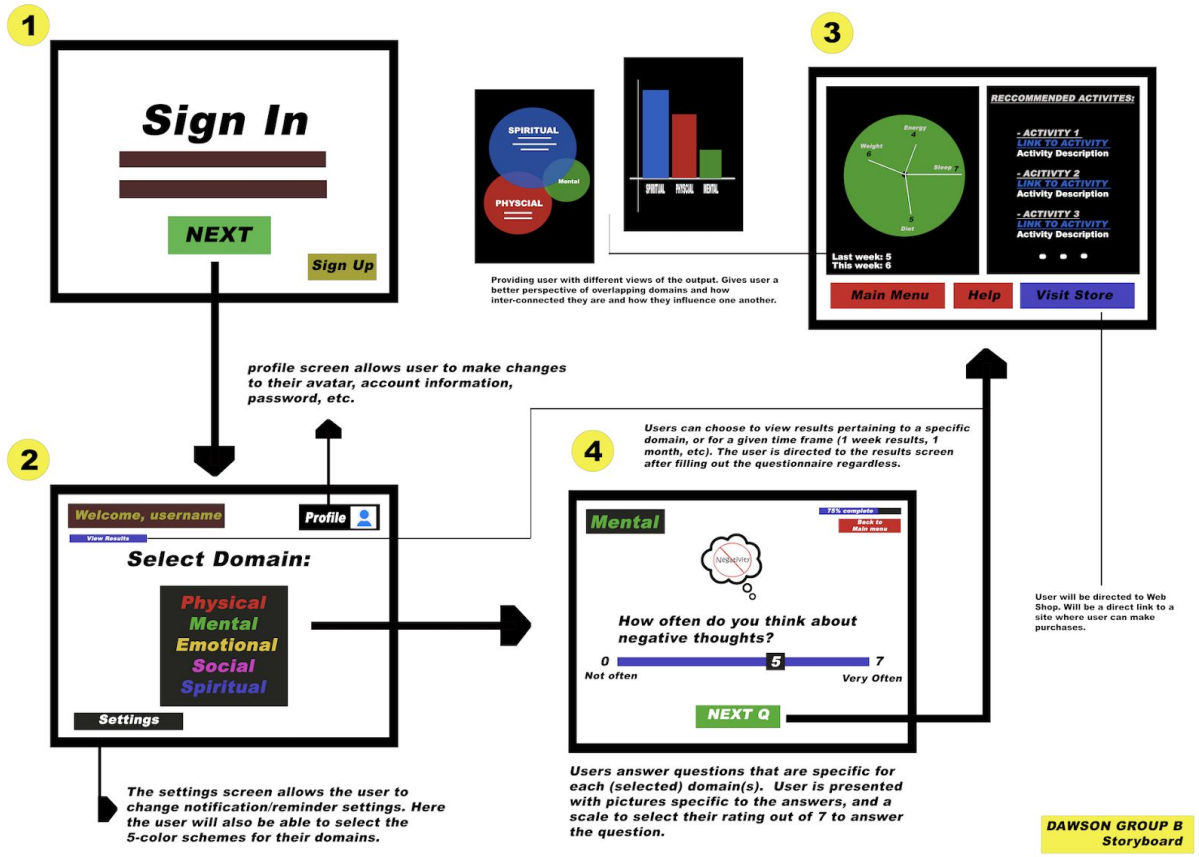
This application will be free-to-use, however users will need a device that is connected to the internet in order to use a web browser. In addition, the data that is collected must be hosted on a Canadian Database to ensure PIPEDA* regulations are being followed.

**PIPEDA - Personal Information Protection and Electronic Documents Act. Sets out the ground rules for how businesses must handle personal information.*

Avatar Feature:

We will be incorporating an avatar feature into our project. This feature will give users the option to customize and create their avatar to their personal liking. This feature will give users the option to incorporate their style and express themselves to the other community of users. This avatar feature gives users an incentive to be more inclined to use the application on a daily basis, and have a sense of '*belonging*' within the VHC community. The avatars will be a basic shoulder up character in which the user can add accessories such as clothes, hats, glasses, etc to their 8-bit avatar.

Output:



Testing Strategy:

We will use a combination of two main testing methods. The first will be manual testing, done by each member as they're programming and testing new features. This will help ensure both the functionality and frontend visuals are working as expected.

Since manual testing tends to miss certain edge cases - more so as program complexity increases - we will also be implementing a second testing method, called 'Test-driven development,' or TDD. TDD's main focus is creating automated tests for a feature *before* programming the functionality. This helps with two core aspects of good development – ensuring added code meets requirements as it's being coded, and allowing automated testing, and therefore automated regression testing.

Essentially, TDD is used by creating and running a test function, which will call functions of different features with a certain input, and then compare the result with the expected output. It doesn't need to know what process is being done on the inputs, only that the output matches what is expected. This will be used for every component and function in our application, with the help of *Jest*, *React Test Library*, and *Cypress* for the frontend, and with Python's built-in *unittest* module for the backend.

To ensure continuous integration, we will use GitHub. Commits will be done by each dev at least once per development day. Features will be broken down into atomic tasks, allowing each part of the feature to be developed on its own branch. This then allows for frequent Pull Requests to update the central 'develop' branch, allowing us to regularly ensure that the constant additions to the code base mesh well together. Pull Requests will only be accepted after all automated tests have been run, and must be reviewed & accepted by another member of the team.

Tech Stack:

Our chosen tech stack for the application is as follows:

Web Application
Frontend: **React**
Backend: **Python**
Database: **MySQL** (UBC test database)
Web server (if needed): **Amazon AWS**

We decided that a website was the best fit for the application, so that it could be accessible from any device with a web browser. React was chosen for the frontend based on our team's current experience, its popularity (and thus libraries available), and its flexibility & power. Python was chosen for the backend, due to its ease of rapid prototyping & design, as well as the number of libraries available. Our application should not require a large amount of net activity, and so the backend server's performance is not a large concern. MySQL was chosen for the DBMS due to our team's experience with it, and because we determined that our data's structure would fit well in an SQL relational database. If we're running the application locally only, we won't have a need for a remote web server. However, if the client decides that he would like the application to be accessible over the internet while we're developing it, then we will use the free tier of AWS to host the site.

Questions:

Common questions received from the requirements video presentation.
*** SurveyMonkey is a template questionnaire the client provided us to use for our implementation.*

What considerations have been made to ensure your product is easily maintainable after it has been handed off to your client?
The application itself will be implemented to ensure that the core-functionalities of the application are satisfied. The project will work the same regardless of whatever database we choose to host the data on. So with this in mind, we will incorporate the application to work with any SQL database provider as long as the core-functionalities of the application are met.

What languages/tools are you using in your tech stack beyond a web-based app and what are the advantages of this?
We will be utilizing React for the front-end development. Python will be used for back-end development. Python is such a versatile programming language that makes it easy for developers to access and retrieve information from databases.

What is the difference between patient and public users?
A patient and a public user will be treated the same way when interacting with the application. Dr. Dawson will be more interested in data collected from his patients rather than the general public. This data will be more of a priority for him, since his interest is more inclined to that of his patients. However, it is worth mentioning that it is still important to collect information from all users since it may become of interest later on for the doctor.

How will you implement reminders for the user?
The reminders being sent out to the users will be done through email. Since we will have limited access to the user's device through a web-browser, we will implement an automatic-email reminder system which is incorporated to remind the user based on the frequency of email reminders they have selected. In addition, we will try to implement notification features if we can get access to the user's device permission (*So this way the user can get the notifications directly to their devices*).

Will you have your client and or end users test your product to ensure it fits their needs?
Aside from the testing done from the developers, the only other individual involved in the testing period will be Dr. Dawson himself. We will allow him to test the final product acting as a user himself, and allow him to get a general feeling of the application. With hopes that he likes the final product, we can talk further steps about launching the application using an encrypted safer option such as Amazon Web Services or Microsoft Azure, and port the application for final use. In addition, the intent isn't to give him a ready-to-launch product (as prof. Bowen stated), instead it's to give them a high-level functioning prototype.

With Dawson being the only person who may access the database, is he familiar with database administration? Also, will your group have some sort of dummy database to use in development?

Yes, we will be utilizing either ubc's student database, or a free version of Google Firebase for our testing period. In addition, Dr. Dawson will be the only individual with access to the data collected from the users. For the testing period, the developers will be able to see the data; However when we hand the product over to Dr. Dawson with a working version, he will be the only individual who has access to the database into whichever database he feels fit. (*We are also open to the idea of porting the database for him also, since the application will be built in mind to incorporate any database; as long as the core features work*).

Why are you blurring out things I am trying to read? How many users are multiple users (accessing database/website)? 2, 10, 1000? What are the domains?

The blurring in the video presentation was done intentionally to ensure that the viewer focuses on the current bulletin point being discussed by the narrator. This was to encourage people to actually sit through and listen to all the points in the video; rather than just skimming through the video and reading everything that was present on the screen. In addition, we classify multiple users as any occasion where there is more than 1 user accessing the website. Also, the five-domains were listed in the video near the start (*by the bottom*) and at the start of this report.

What Canadian server will you use? How are you going to support multiple users? When using survey monkeys, does this survey program insure the user's privacy?

We aren't actually using SurveyMonkey's site with our application. We are instead taking what is provided on surveyMonkey and incorporating it into building our questionnaire. SurveyMonkey is more of an intended tool to help us design our site, we aren't directly linking surveyMonkey into our application. Also, the Canadian server we plan on utilizing is the ubc's student database for the testing periods of development. The final product will run on a database of the client's liking.

Will the web store be built by your team, or built on an existing platform?

The store will simply be a link to an external store that Dr. Dawson may create and manage on his own - Shopify, for example.

Was the client background info necessary?

We felt that the client's information was necessary. It gives the user a better understanding of who we are working for, and what the client's interests are so that we can meet them. While it wasn't required, we felt that it was better to include this information in our video presentation.

Will this product be sold to other psychologists? Or are the client users always going to be clients of Dr. Dawson?

From our understanding, this application is not intended for sale. Instead Dr. Dawson is more interested in offering a free service for people to get a better understanding of their health. The users themselves won't always be client's of Dr. Dawson since the website will be available for public use. The intention of this application is to serve the general public, rather than a small select group of individuals. We will implement this application with the fact that Dr. Dawson may refer his patients to use it, in mind.

It seems like a very niche app, are there plans to expand it?

We don't plan on expanding it past this initial scope. If Dr. Dawson wishes to expand it, he'll have the source code to continue development.

How is the radius of each circle calculated given the length of the "spokes"?

The radius of each circle domain is calculated by taking each question asked in the domain questionnaire (*each question is rated out of 7*), and all the answer's ratings are summed up and divided by the number of questions asked in the questionnaire. This will give the user the average domain rating for each domain. Each spoke within the circle represents an activity or sub-domain within each domain (*eg. Sleep spoke within Physical Domain*)

It seems like you need a web portal for the admin separate from the app. Is this being considered?

We have planned to integrate the admin panel into the web app - it may just be a link somewhere on the homepage (with access authentication), or a link only visible to Dr. Dawson while he's logged in.

How will you ensure there are no data leaks? The data seems like it is sensitive for the patients.

A few different things will help ensure no (sensitive) data leaks. First, we'll be using a database hosted by a large, secure company such as Google. We will also be encrypting stored data, such that the raw data is essentially unusable without encryption keys. Finally, we will be using a back-end server to handle user requests, to ensure that malicious requests are denied.

Some data flow in DFD level 1 is not labeled. What that data should be.
This was a good point - labels have been added to the remaining flow lines in the diagram.

How will the accessibility need of different user groups be accommodated through the app?
Special accessibility concerns will not be covered within our scope. The target user group shouldn't have a higher-than-average percentage of users with accessibility issues. We will try to make it as accessible as possible for average users, though, including a clear, readable, and easy to follow design.

Is there a separate webapp/UI for Dr. Dawson to administer and create the questions? How will he be accessing the results?
The admin panel will allow the doctor to manage the survey questions asked to users, including how the answers to the questions will affect the feedback given to users (circles).

Except Survey monkey Do you consider embedded surveys? And What kind of method will be used to collect the survey data?
We will have the survey built into our application - we are simply taking the questions from an existing Survey Monkey survey. The user will be able to answer the survey on the website.

Since you are building a web app with a shop, there seems to be monetary goal here as well. Has Dr. Dawson discussed if he will be his user's data?
This question seems to be missing a critical word. However, as mentioned above, the shop will be external and handled by the doctor, however he wishes. It will simply show up as a link on our web application.

For uploading features' codes to Github, does all members in your team have to agree or one person with satisfied tests will be ok? Can one user access another user's profile and reading history?
For every Pull Request, one other member of the team will have to review and accept the changes, and Pull Requests should only be created after all automated tests have run successfully. In the app, there will be no sort of connection between users - only between Dr. Dawson and his clients.

What frameworks and libraries will you be using to implement the automated testing mentioned in your presentation? Also, what specific frontend and backend technologies will make up your tech stack?
For testing the frontend (React), we will be using Jest, React Test Library, and Cypress. For the backend (Python), we will use the 'unittest' module. As mentioned, the frontend will be React, backend Python, and the database is going to be a MySQL test database. If the doctor decides to continue using our application, it should be simple for him to switch to any MySQL database provider.

You mention that Dr. Dawson is the only one to have access to the database is there any plan in place for if he passes for example will anyone else need access to this information will they be able to get to it. Or Is there a plan to shut it down. With him being the only one how familiar is he with being a DBA? You mention continuous integration were you planning on using any of the workflows or actions provided by GitHub? Is there also any plan for linters?
The app is designed specifically for Dr. Dawson, but this does raise a good point. It's possible that he'll want to pass it off to someone else in the future, or even have multiple administrators alongside himself. He is not a DBA, but the plan is for him to be able to access anything he needs easily, through a GUI admin panel. We may add a function in the panel to allow him to designate other users as an administrator (and remove them as admins). We have no plans to use GitHub workflows/actions or linters.

Will admins have the access to users' profiles?
Yes, Dr. Dawson, as the administrator, will be able to access all of his clients' profiles and app usage. New users will be asked if they are a client when signing up, and non-clients' data will not be visible to him.

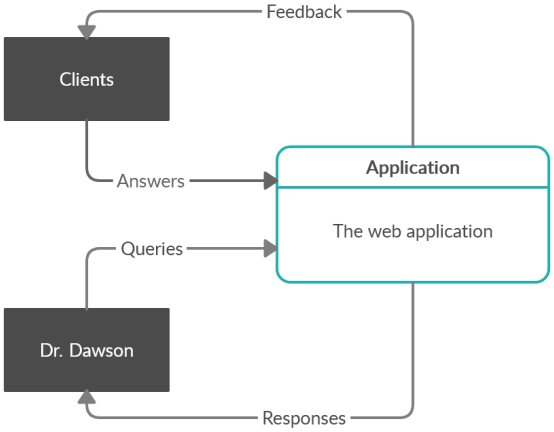
Conclusion:

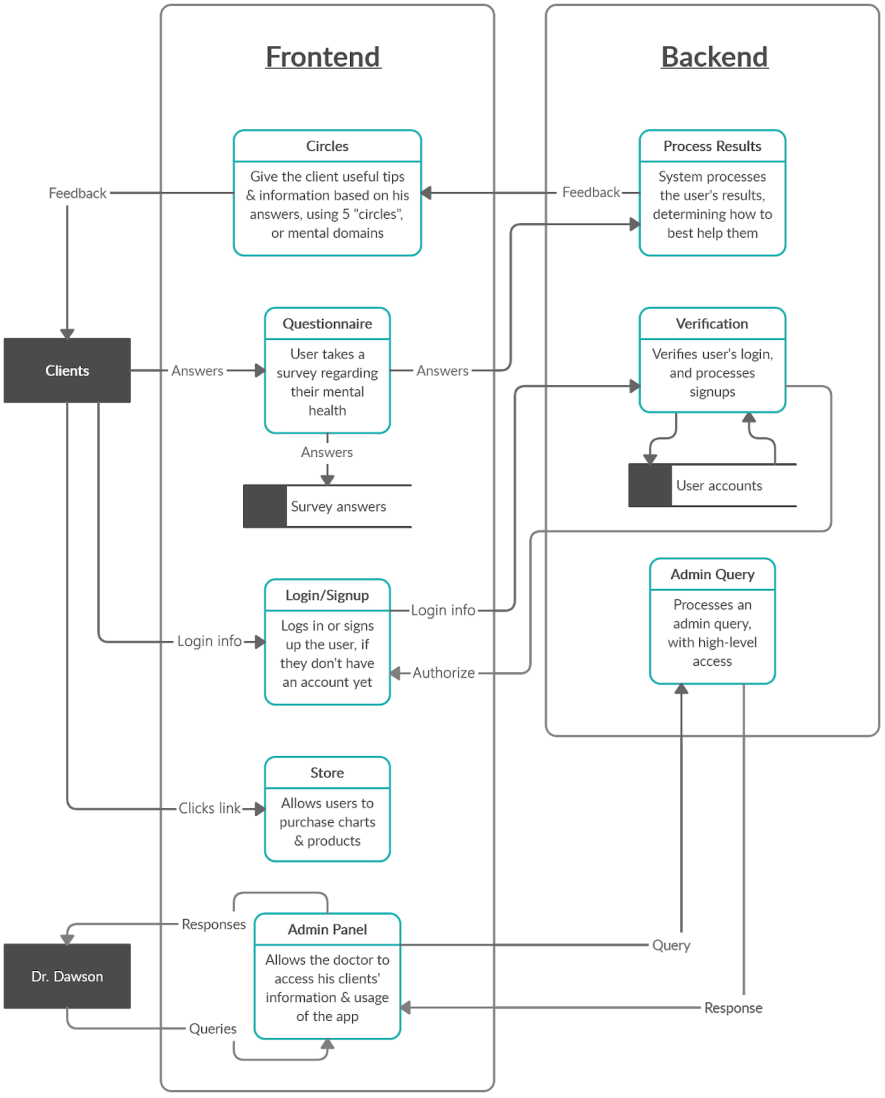
On page three of this report you will find the link to the private gitHub repo where you can find the source code and supporting documentation for VHS. In addition, below you will find the link to the video presentation for VHS in it's very early stages:

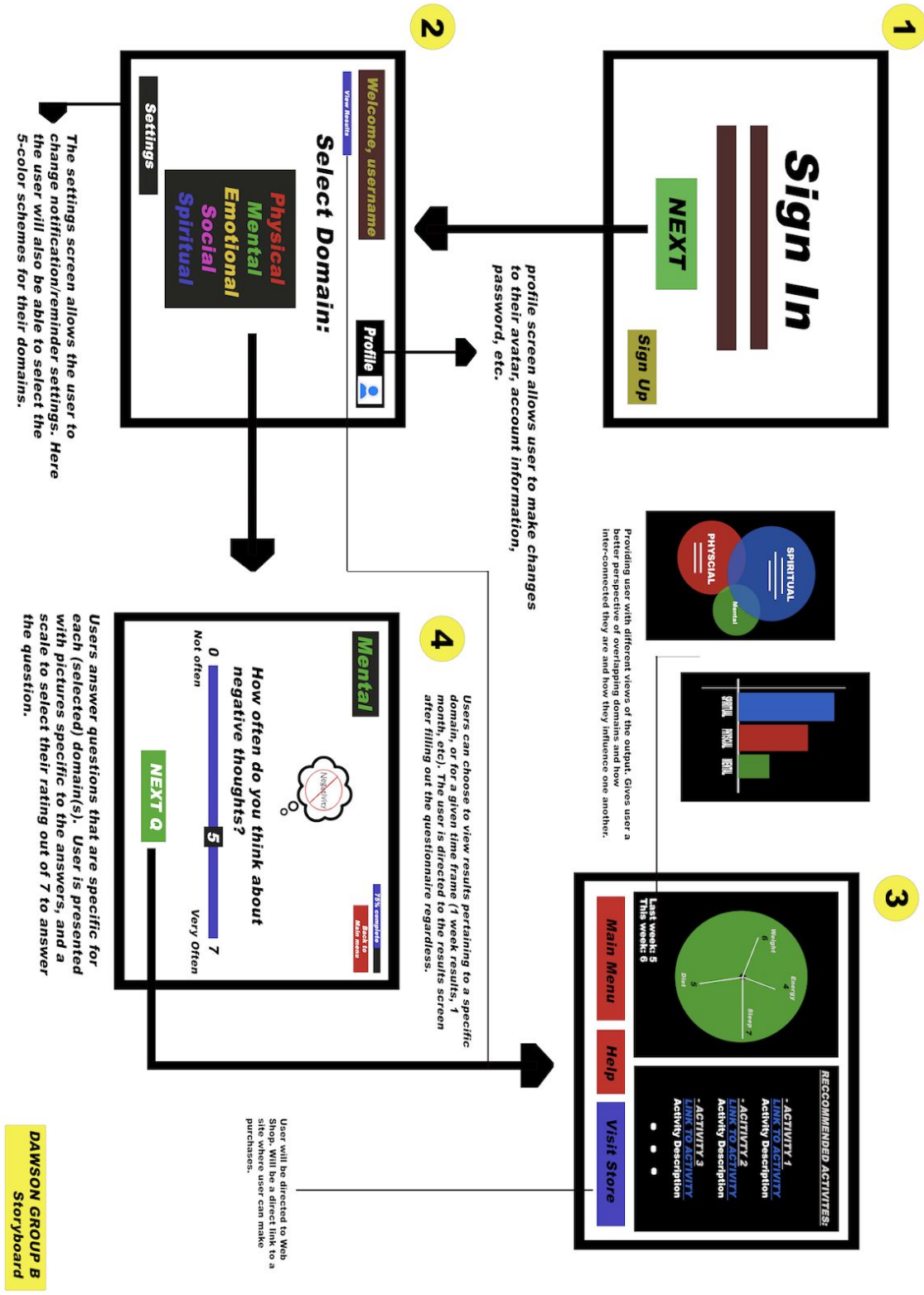
https://drive.google.com/file/d/12pgnbbHfnkAmiLohw3DAbDcZ4u2Xsc8E/view?usp=sharing

** Please note that some of the information within the video has been updated and fixed for this written report. Please refer to this document for up-date material and not the video.

Images used within the report, size has been enlarged for better viewing.







End of Requirements Report.
Thank you