Team (1)
Mohamad Agha
Ali Ibrahim
David Pinos
Daven Minhas

Woo Woo Network

Project Description

The Woo Woo Network is a platform focused on helping people who have experienced pregnancy loss via spiritual healing.

Since spiritual healing “Energy Medicine” is a scarce service, the platform proposed by our project sponsor Kimberlee Klein aims to provide a place where healers can provide their services to clients who need them. The website will connect clients with a variety of options for spiritual healers who will aid them with their healing process.

The project aims to develop a complete website that allows clients to register, search and find healers, schedule appointments and pay for them.

The target user groups include the clients and the healers.

User Groups

Clients:

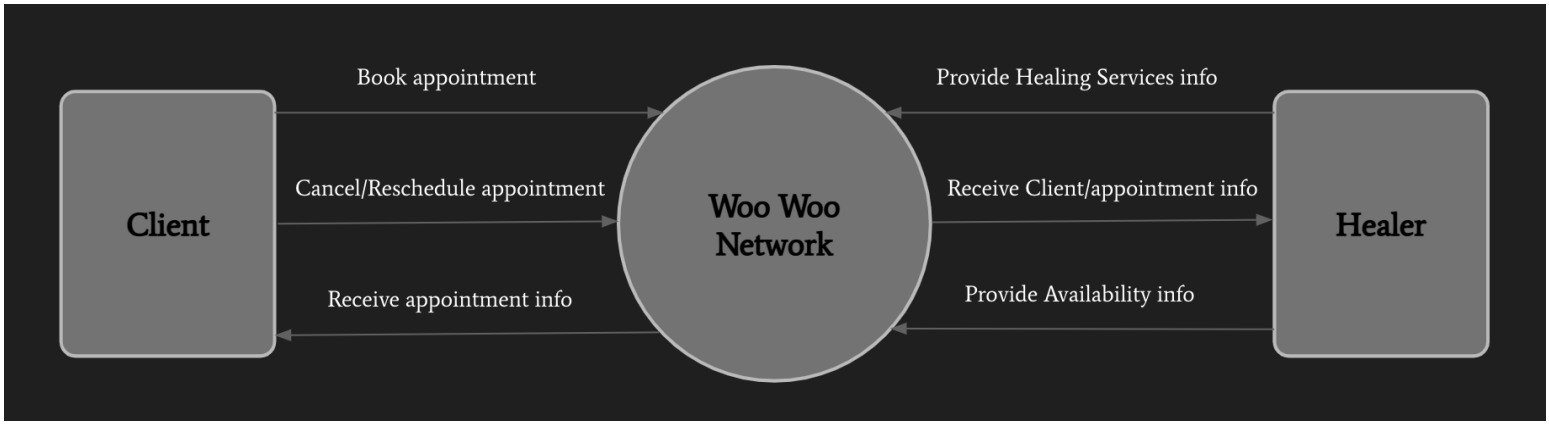
- Browse the Healer dashboard, search for healers, and select desired service.
- Booking appointments.
- User appointments, history, rescheduling
- Payment
- Manage Journal page

Healers:

- View, request to reschedule, or cancel client appointments.
- View recent reviews.
- Manage availability.
- Add, update, or delete services.
- View history of appointments.

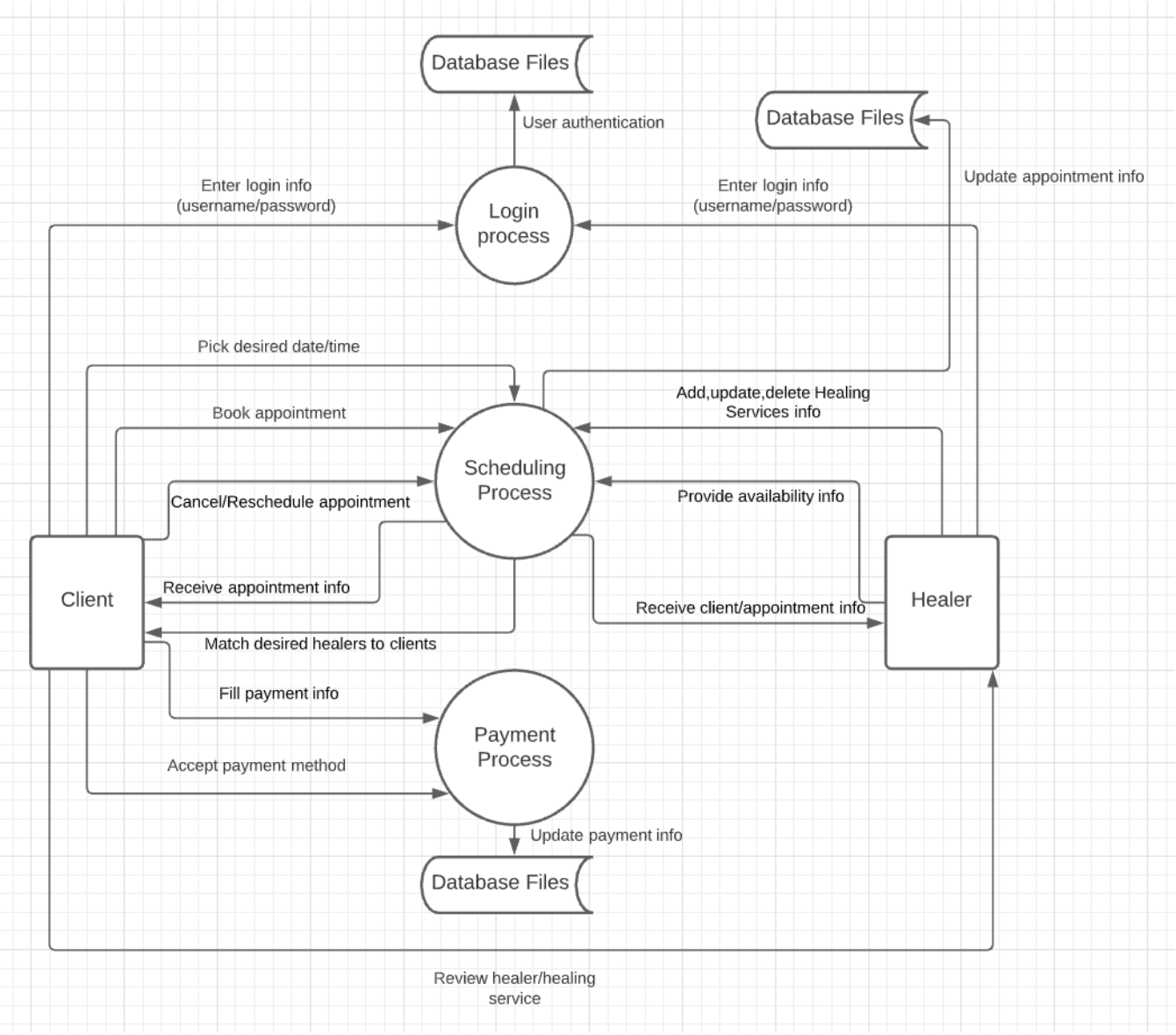
Both user groups can sign-up/login, as well as manage their user profile pages.

DFD Level 0



This diagram encapsulates the basic function of the website. Clients can book, cancel or reschedule appointments, and they receive appointment information when one is booked. Healers provide their services and availability information, and they receive their client and appointment information when a booking is made. Mentioned processes are aimed to be delivered by the Peer Testing #1 milestone.

DFD Level 1

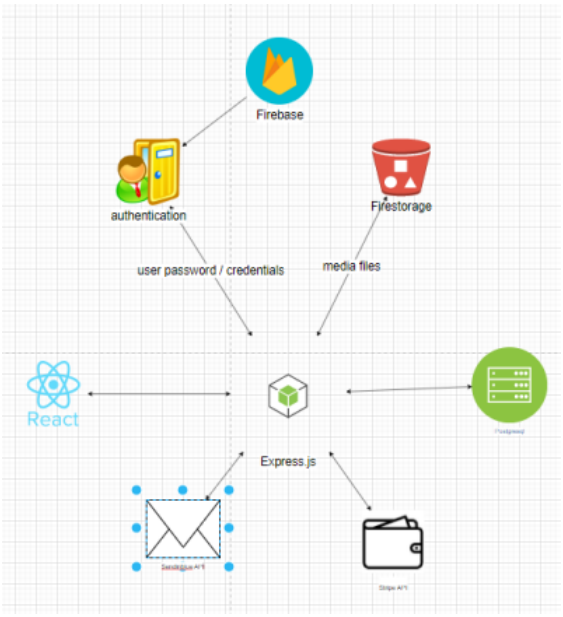


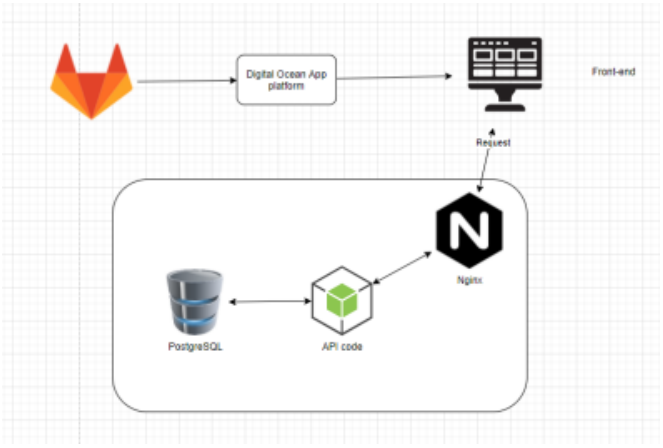
This diagram goes into appointment reservation in more detail. The scheduling process, payment process, and login process all provide their respective functions while updating the database. Mentioned processes including login, scheduling, and payment functionality are aimed to be delivered by the Peer Testing #1 Milestone.

Tech Stack

We will be building on what the previous team did, therefore we will continue to work on the project utilizing the stack they used.

- React and Material UI for Front-end
- Express.js for building API
- Sequelize for sending query to database, SQLite for production and PostgreSQL for development
- Gitlab for Git repository management
- DigitalOcean, GitlabCI and Docker for deployment
- Stripe for payment handling
- Sendinblue for email sending
- Firebase for authentication





Testing Methodology

In order to ensure that our code meets the customer's expectations we will follow a thorough testing process. In order to ensure continuous integration we will use this as a series of steps to follow.

We will begin with unit testing which will allow us to test individual components of our work such as functions in our code. We can use technologies such as Jest or Mocha.

After the different components meet our needs we will use integration testing to make sure that these units run according to the requirements when integrated with each other.

Once the integrated codebase works and the different components of the stack are joined (e.g. back end, front end, database) we will perform performance testing. This will stress test the program and we will identify possible break points. We can use WebLOAD for this step.

Once the system is stress tested ,we perform acceptance testing. We will make sure the clients requirements are met and that the program works as solicited.

Finally we will use regression testing to make sure that as we add functionalities this does not compromise previously tested code. Certain aspects of this testing process can be automated and will be rerun every time new functionalities are added. Selenium can be used for this type of testing.

Functional Requirements

Milestone 1 (Requirements):

The primary goal of this milestone is to present all important project requirements. Providing a high-level description of the project with appropriate DFD diagrams, discussing the Tech Stack that will be used, as well as going over major milestones and most functional and nonfunctional requirements.

Milestone 2 (Peer-testing #1):

The goal of this milestone is to have an efficient and simple system for clients to search for healers. Two features will be implemented to get it working.

-Option for healers to add tags and categories in order to describe their services more accurately.
-Search function that will allow clients to search for healers based on location and categories.

Milestone 3 (Peer-testing #2):

The second milestone will have two goals: automated testing and implementing a help bot. The client has expressed that a help bot is not a mandatory feature, we will only be developing this if time permits.

-Implement automated tests in order to catch any unwanted behaviour and bugs.
-Chatbot that will help clients and healers with any technical problems or confusion. This chatbot will contain mostly hard-coded responses to common questions and problems.

Milestone 4 (Final Product)

For the final product, the main goal is to implement some features that will better the user experience. In order to keep the code well reviewed, we will also be updating testing procedures and improving data validation.

-Calendar highlighting for ease of finding an appointment
-System that will allow clients to rate healers.
-Update our automated testing code in order to test the new features.
-By reviewing the current data flow and properly sanitizing user input, we can improve data validation greatly and in turn increase the security of the website. Data validation is especially required in order to manage the schedule and make sure each time period does not overlap with each other.

NonFunctional Requirements

Some of the non-functional requirements and environmental constraints are as follows: **Security** for personal information, banking details, health records. **Reliability** to prevent failures during booking and transactions. **Useability** for people of all demographics. This will require discussions about ethics as this site will be for those who need help. **Accessibility** will be important in two ways, the first being for those who may not be able to navigate the site in a traditional manner. And secondly accessibility for recurring testing - there must be an easy way to access the site and test minimal updates without hassle. **Capacity** in two elements; database must be able to support all users, site must be able to handle user traffic on hosting side. There should be capacity for **scalability** should the amount of member/site traffic/healers grow beyond the means of what is set up. **Processing** capabilities between team members and local machines. **Maintenance** should be easy and minimal. We will also face **time management** constraints between team members, clients, and time zones. The non-functional requirements will be managed alongside the implementation of our list of functional requirements. Some of the non-functional requirements need client input for more detailed instructions and will be discussed in upcoming meetings.

Conclusion

Woo woo network will be a platform to match customers to healers. We will be using the tech stack that the previous team used, as we will be building on their work. The client has given us instructions as detailed throughout the presentation and they will be met according to our milestone layout. A thorough testing process has been agreed upon, in order to ensure continuous integration and to allow us to add features without tampering with previously tested code.

Questions/Feedback

- How are you planning on implementing performance testing? We will be using technologies such as WebLoad, which will test how our product reacts to stress.

- What methods are you using to implement acceptance testing? For this type of testing we will make sure we have our clients requirements and we will use the results of the other tests to see if they pass the threshold established by the clients requirements.

- Will there be any restrictions on rescheduling appointments with healers? We will have to discuss these technicalities with the client as we have to cater the product to her expectations and business model.

- Is there planned to be a deposit and/or a fee for a reschedule within say 24 hours of the appointment? We have to talk to the client regarding all the technicalities that she expects from the requirements she has provided. We will do so as we implement the features and these questions arise.

- Will the database for the appointment info and the database for the payment info be kept separate for security reasons? It will depend on the client's technical requirements on how they want the payment system set up which we will have to discuss with them.

- who will have access to the payment info database? It will depend on the client's technical requirements on how they want the payment system set up which we will have to discuss with them.

- Can anyone sign up and list themself as a healer? If not, what does the process look like once someone submits a request to set up a healer profile? We will have to discuss these technicalities with the client regarding the requirements for the registration as a healer.

- I liked the detailed description of testing strategies. Maybe mention how you would achieve this technically? What testing framework will you use? As mentioned previously, we will be using multiple technologies such as Jest/Mocha, WebLOAD, and so on to properly test our program.

- It was mentioned that you are building off of another's work who already started this project. It wasn't clear how much the team before had completed? The team before us had set up the environment for the front and back end which we are building on to implement our features.

- The tech stack shows different libraries/languages/frameworks for your project, though how do you plan on implementing data validation? Data validation techniques will be discussed in more detail as we approach that milestone as well as with our project sponsor. Open-source tools could potentially be used.

- What would the differences be between registering as a client or a healer? Would the healer have to have specific credentials? We will work on implementing some way of properly verifying healers. Requirements for healers will be discussed with the client before moving on.

- How do you plan on keeping personal information, banking details, etc more secure on the website

and the database? User credentials are being handled by Firebase. Firebase is a google company and therefore we are reliant on google’s extremely secure services to handle that. Banking details are handled by Stripe, which is also much more secure than saving it in a database because of its large scale.

- There is no mention of payment towards the Healers in your DFD diagram, will they be paid out by the service itself? Depending on the method of payment, the healer would receive the payment directly when it gets paid by the client.

- You are using firebase for authentication and PostgreSQL for your database, fire base itself has easy integration with its NoSQL database alongside its authentication service which can make the process smoother, is there a reason for splitting up the services you use? PostgreSQL was used by the previous team, and therefore it is already well integrated.Firebase works perfectly with the system for now, but if we need to change it we will consider using NoSQL.