1. Problem

For many people, visiting the doctor can be tedious and repetitive from filling out similar forms and questionnaires with each visit. With patients who see various doctors and have complicated medical histories, documentation consumes a large portion of the appointment time.

2. Background Information

Currently, hospitals have their patients fill out forms using pencil and paper prior to their appointments. Afterwards a hospital worker has to enter all this data into their database. This takes up a significant amount of the patient's time as well as the employee's time. To remedy this problem, our team has decided to develop software that allows the patient to store their information to a database, and it will allow doctors to view and add info about the patient's medical history. When the patient goes to another hospital, the patient can share their database entry with their doctor saving both the users time and money.

3. Environment and System Models

Our System will be client based with a remote database server. The client will communicate directly with the server. This leads to multiple security vulnerabilities, but allows us to have a faster development speed.

The client will allow users to login and fill out a form. They then can give doctors access to their profile. Doctors will be able to look at their patients info and medical history. They will also be able add to the clients medical history. Clients can review this info and add or change the info when the doctor makes a mistake.

The client will be developed in java. The ui will be done using swing.

3.1 Client

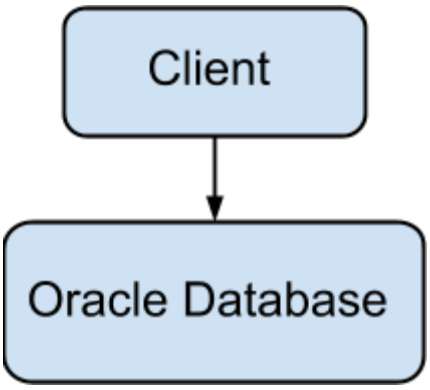
3.1.1 This will be coded in java and will communicate with a database

3.1.2 It will have a GUI for filling out and reviewing data

3.2 Database

3.2.1 This will store all of the patient data

3.2.2 It will also store account data and links between patients and doctors



4. Functional Requirements

4.1 Interaction between client and database

- Client application is able to send and receive data from the database
- Client application will then use this data to allow users to login, access, or modify their profiles

4.2 User to client interaction

- User can create an account as either a doctor or a patient..
- Doctor
    - User can manage and lookup patient information
    - User can edit their information
- Patient
    - User can lookup his own information on record
    - User can edit and save his info
    - User is able to search for a doctor
    - User is able to authorize a doctor to see his profile

5. Non-Functional Requirements:

Usability

The program should be very user friendly. Tabs and fields are clearly labeled and it will be easy to navigate through the program.

Availability

The database server should always be available, client side should be able to make the connection upon the launch of program.

Maintainability

The application will have sufficient documentation and organized structure so that a qualified person is able to maintain the software in the future.

Reusability
This program will be designed with the reusability of its components in mind. Related data and relevant operations will be grouped together to provide as much reusability as possible.

Reliability
If the system is to crash, patient and doctor data that has already been entered and saved will not be affected.

6. Use Cases

Case: User wants to create an account

1. User will launch the executable
2. User will click on the create account button
3. User will click a button to choose either a patient account or a medical provider (doctor) account
4. User will create a unique user name and a password that is between eight and fifteen characters long
5. User will click on the enter button and be sent to their profile page.

Case: User signs in to view program features

1. User signs in to their medical form with account username and password
2. Upon successful login, user sees the profile page

Case: User wants to enter data to their medical history

1. Upon successful login, user will see patient profile
2. User can enter information in the input boxes and save

Case: User wants to give his or her doctor permission to access their profile

1. User will sign in with username and password
2. Upon successful login user will see his or her patient profile
3. The patient will click on the ‘My Doctors’ tab
4. The patient will enter the username of their doctor into a text box and re-enter their password to give their doctor permission to view their profile.

Case: User wants to look up patient’s drug allergies

1. User logins in with username and password

2. User will go to their patient list and select the patient he or she wants to see
3. From here the doctor will be able to click on the drug allergy tab and be able to and view the drug allergies for that particular patient.

Case: User wants to look up a doctor

1. User signs in as patient or doctor
2. User clicks on the Doctor tab where he can search a specific doctor through some filters like Doctor’s name and specialization