NOVA

Health Care Application and Digital Assistant

Software Design Specification

Version 1.0

Name: Abhirupa Mitra
Reg No: 17BCE0437
Course Code: CSE3001 Software Engineering
Slot: L39+L40

CONTENTS

1. Introduction
    1.1. Purpose
    1.2. Scope
    1.3. Definitions, Acronyms and Abbreviations
    1.4. References
2. Decomposition Description
    2.1. Module Decomposition
    2.2. Concurrent Process Decomposition
    2.3. Data Decomposition
3. Dependency Description
    3.1. Intermodule Dependency
        3.1.1 Independent Modules
        3.1.2 Dependent Modules
    3.2. Interprocess Dependencies
4. Interface Description:
    4.1 Module Interfaces
        4.1.1 User Login/ SignUp Interface Design
        4.1.2 Edit/Alter Medical Database
        4.1.3 Make/ View Appointments
        4.1.4Chatbot

5. Class Diagram For The HealthCare Application:

6.Sequence and Collaboration Diagrams

- 6.1 View Medical Profile
- 6.2 View Appointment
- 6.3 Diagnose Using Integrated API
- 6.4 Make Appointments to Nearby Hospitals

7. Detailed Descriptions and Activity Models For Major Operations:

- 7.1. Login/Signup:
- 7.2 Edit/Alter Database
- 7.3 Disease Diagnosis and chatbot communication
- 7.4 Make/View Appointments

Preface

This document presents the Software Design Specification for the HealthCare Application. The major sections of the document address the system decomposition by module, concurrent process, and data entity. The system dependencies are also described.

Section 2, Decomposition Description, gives a view of the whole system design including concurrent processes and data entities that are common amongst all system modules. An important discussion of how what kind of API will be integrated for the diagnosis of disease has also been included. This discussion includes a UML Class Diagram that depicts the entire system.

Section 4, Interface Description, goes into detail about the user interface for each module of the E-Stamp software. This is followed by an important discussion of the processes implemented in logic for each module of the system.

Section 5, Detailed Design, extends the design discussion found in Section 2 and describes the design for each system module in more detail. A UML Class diagram is included for each module design discussion. This is followed by a description of the data requirements for each module and the design of those data elements.
Introduction

1. Introduction

1.1 Purpose

The purpose of the Software Design Specification is to describe the specific design of the Health Care Project. The specification includes an overview of the design alongwith software module decomposition

This document provides a detailed description of each software module's design. For each module, a user interface design and class diagram design is given. As well, a process description is described for each module. It is in the process description that the details of what logic will need to be implemented are given.

1.2 Scope

It is within the scope of the Software Design Specification to describe the specific system design of the Health Care Application. This would include user interface design, object-oriented class design, process design, and data design. Any specific detail that is needed about the standards or technology used to design the software are within the scope of this document.

It is outside the scope of this document to describe the functioning of the human body, and its immune system, and other health related discussions. This document also refrains from discussin the algorithm that is put into work for symptom analysis

1.3 Definitions, Acronyms, and Abbreviations

Definition, Acronym, or Abbreviation	Description
API	Application Program Interface

2. Decomposition Description

2.1 Module Decomposition

The HealthCare application has been decomposed into the following modules:

- User Login/Signup and Data Authentication module: Allows the user to login/signup using his credentials
- Edit/Alter Medical database module: This allows the user to feed the database with details about his medical history.
- Make/View Appointments Module: This module allows to track all nearby health centres and allows the application to make appointments on the behalf of the user.
- Chatbot Module: This module involves a basic chatbot powered by IBM Watson.

2.2 Concurrent Process Decomposition

The Healthcare Application consists of 2 major components, the medical database of the user and the nearby hospitals as stakeholders. The design of the interface used by the hospital staff to approve appointments are beyond the scope of the current project.

The hospital staff will gain access to the medical history of the patient once they approve the said patient’s appointment to that particular hospital.

2.3 Data Decomposition

The only major data component is the medical information for each user of the applications

Medical History Information:

User Name:The username of the user

Systolic pressure: The systolic pressure of the user

Diastolic Pressure: The diastolic pressure of the user

Body Mass Index: The Body Mass Index Ratio

Glucose Level: The glucose level of the user

Pulse Rate: No, of heartbeats per minute for the user

Age: How many years old the user is

3. Dependency Description

3.1 Inter-module Dependencies

3.1.1 Independent Modules

The following modules are independent and do not rely on other modules to initiate them or provide them data:

- User Login/ Signup Module
- Edit/Alter Medical Database

3.1.2 Dependent Modules

The following modules are dependent on one another for their functioning:

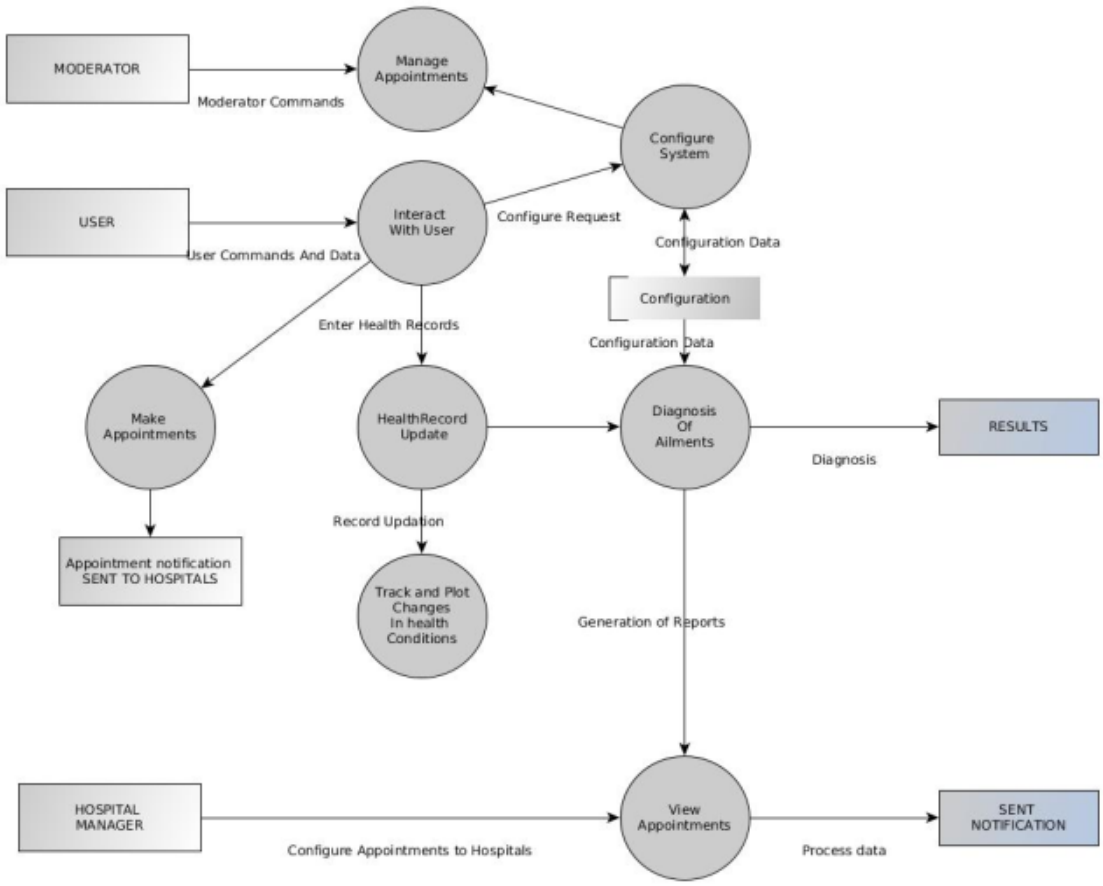
- Make/View Appointments Module: This module depends on the Edit/Alter medical database module for medical history of a person.
- Chatbot Module: This module also depends on the Edit/Alter medical database module for medical history of a person, to send to the the health centre while making appointments.

3.2 Inter-process Dependencies

As described earlier, the main process is the medical user history and database where the user’s medical records are stored. These records are then further accessed and used later for diagnosis purposes.

3.3 Data Dependencies

The following data flow diagram shows the data dependencies:



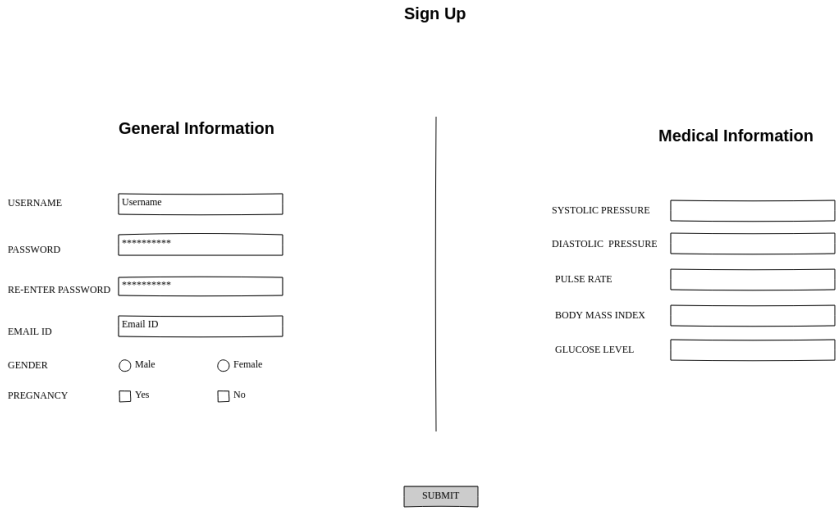
Data Flow Diagram

4. Interface Description:

4.1 Module Interfaces

4.1.1 User Login/ SignUp Interface Design:

4.1.1.1 User Interface Design



4.1.1.2 Description

The login interface is the first page that appears on opening the application. The user can either login from this page, or if he doesn't have an account he can opt to create one, in which case he is redirected to another page which allows him to signup.

4.1.2 Edit/Alter Medical Database:

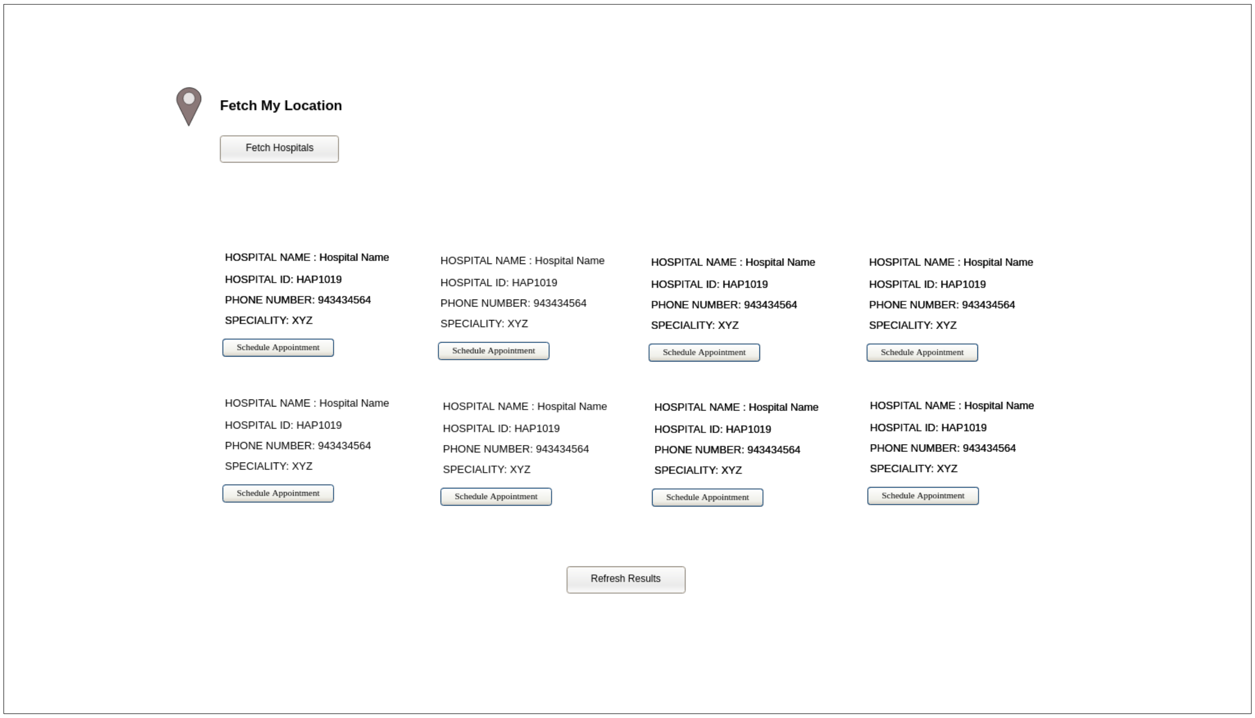
4.1.2.1 User Interface Design

4.1.2.2 Description:

This interface display the medical history of the user. It also shows a graph depicting the trends in the changes of the user's health. It provides the user with two options, ie, to either make an appointment or two talk to a digital assistant.

4.1.3 Make/ View Appointments:

4.1.3.1 User Interface Design:

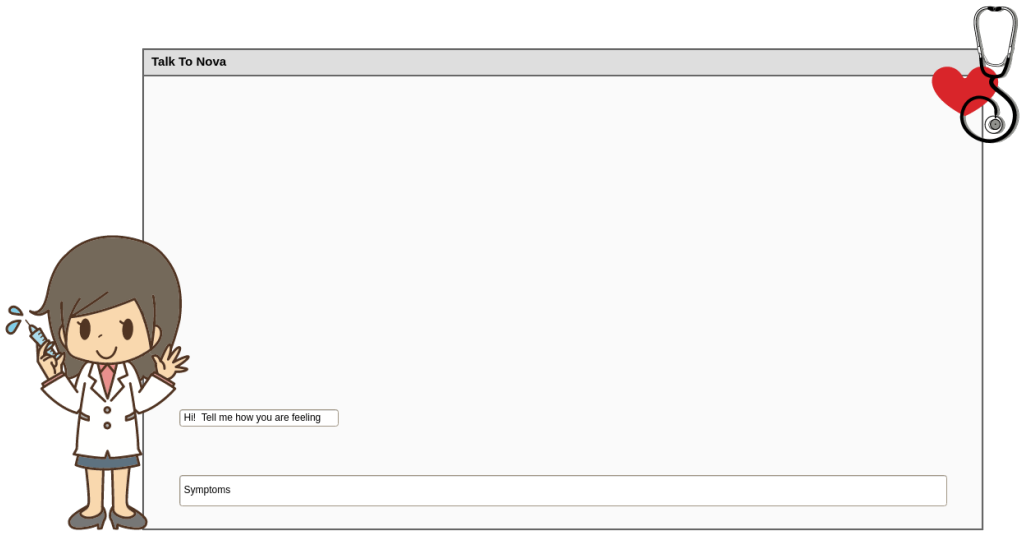


4.1.3.2 Description:

This interface allows the users to geolocate has current location and find out all the health centres that are present in a 1 km radius and gives out a list for the same. The user can opt to contact the health centre himself, or choose to allow the app to make the booking for himself.

4.1.4 Chatbot

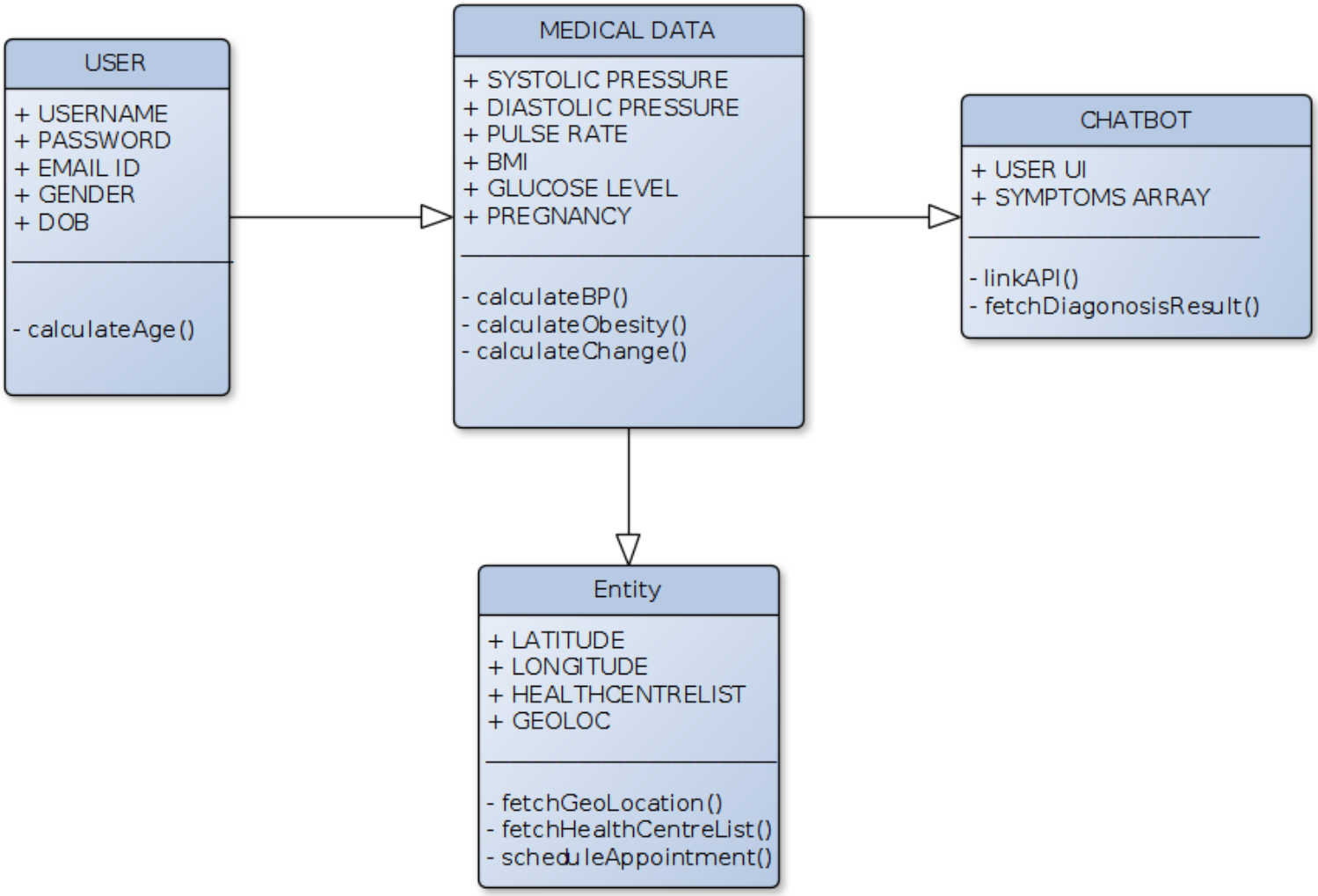
4.1.4.1 User Interface Design



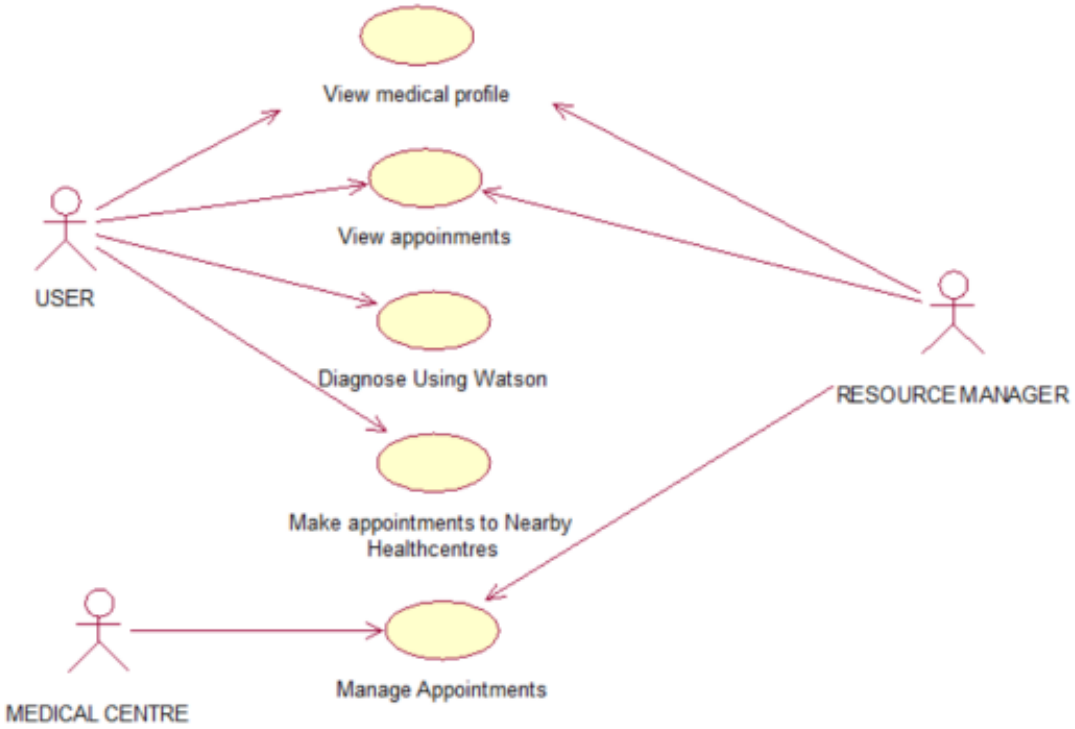
4.1.4.2 Description

The interface will allow us to communicate with the assistant, which is a simple chatbot. The chatbot can be integrated with an API, which can predict diseases from symptoms.

5. Class Diagram For The HealthCare Application:



6. Use-Case Diagrams:

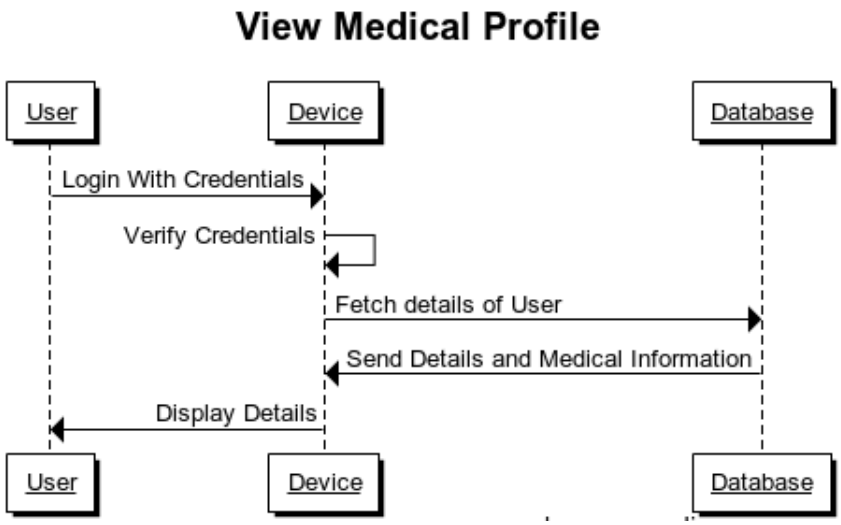


6.Sequence and Collaboration Diagrams

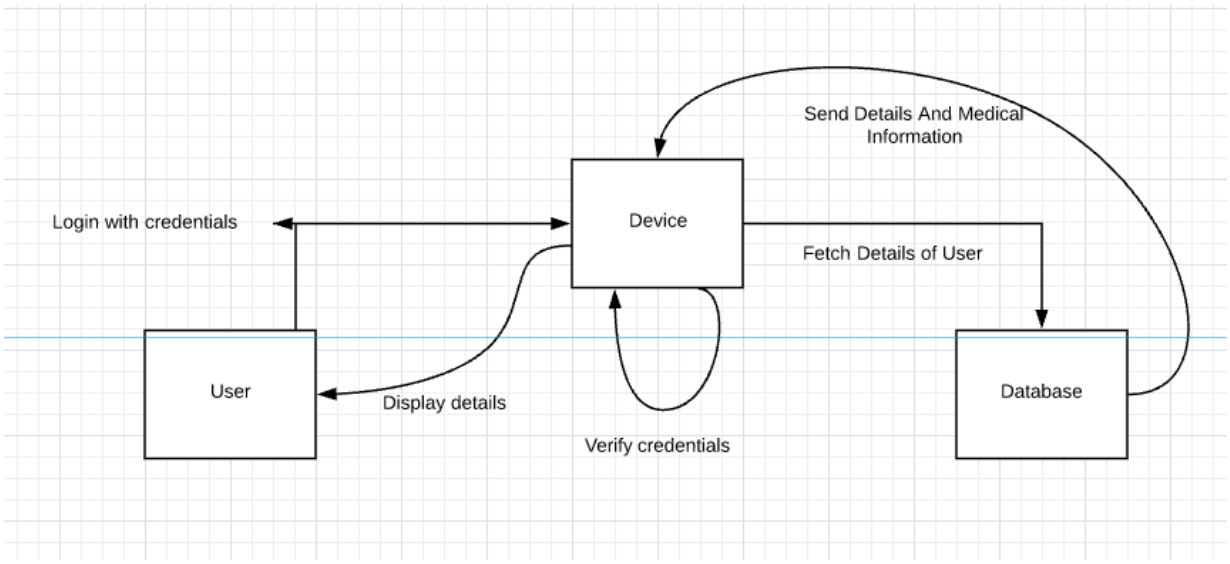
For Given Components In the Use case Diagram:

6.1 View Medical Profile

Sequence Diagrams

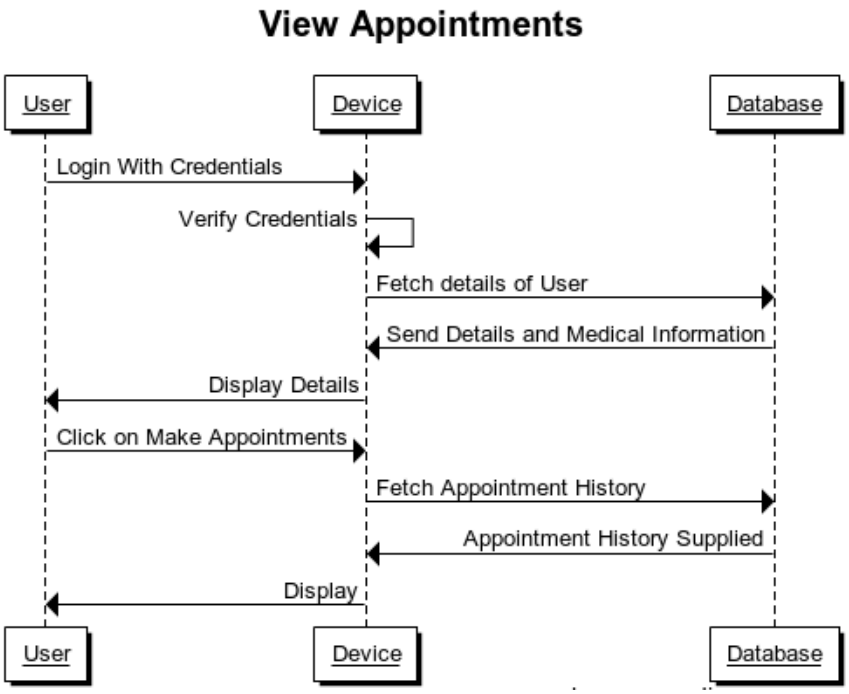


Collaboration Diagram

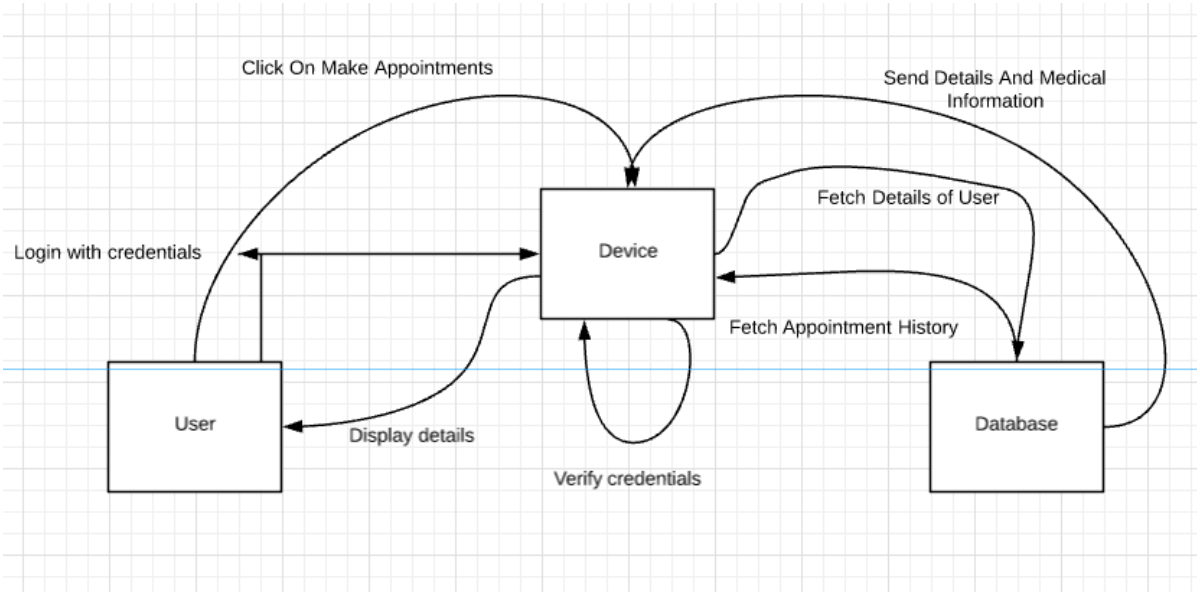


6.2 View Appointments

Sequence Diagrams

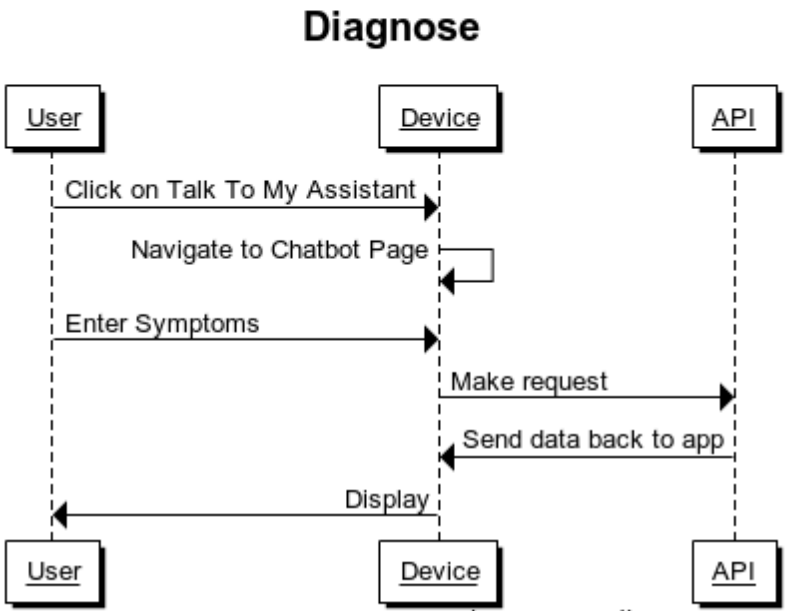


Collaboration Diagram

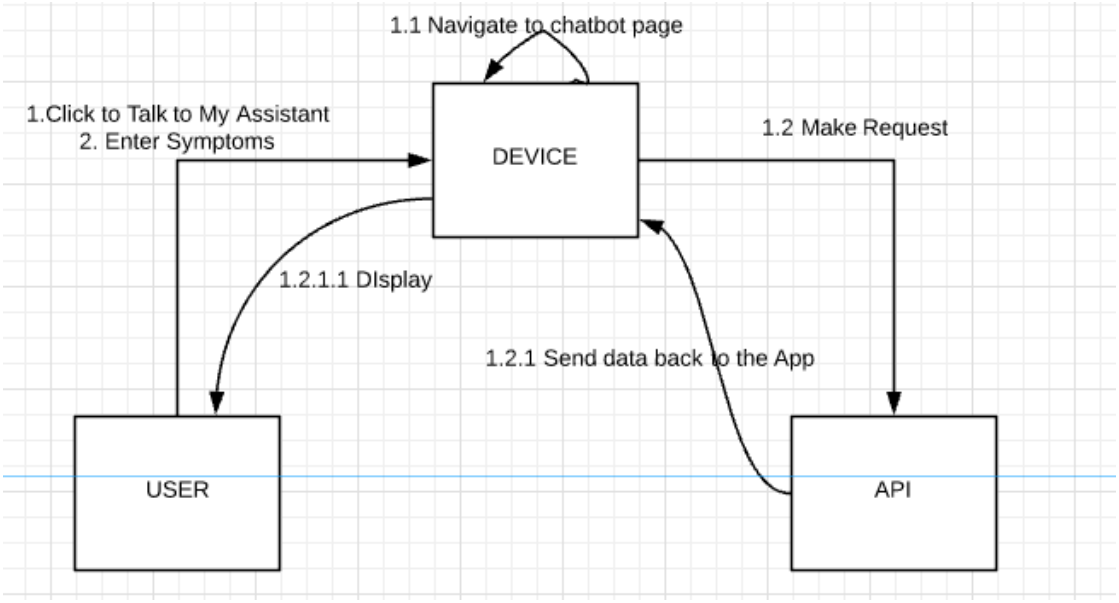


6.3 Diagnose Using Integrated API

Sequence Diagrams

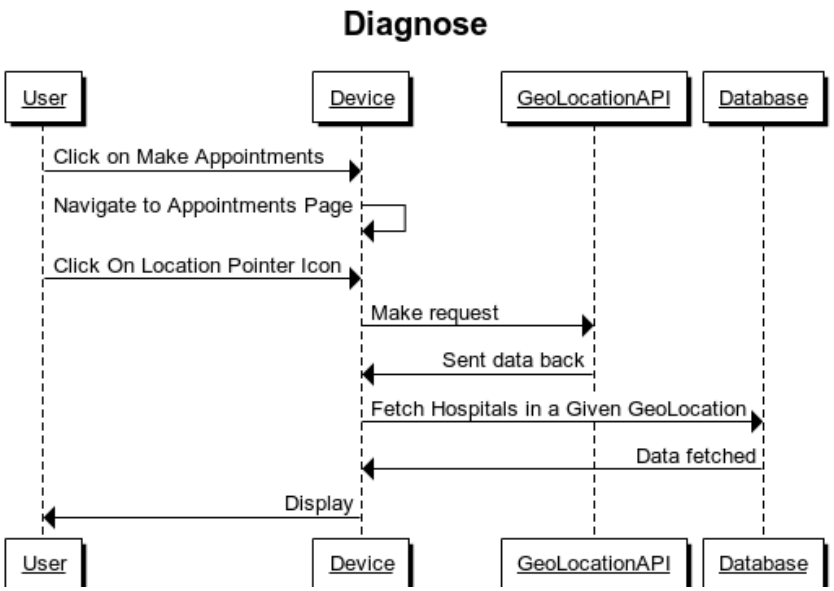


Collaboration Diagram

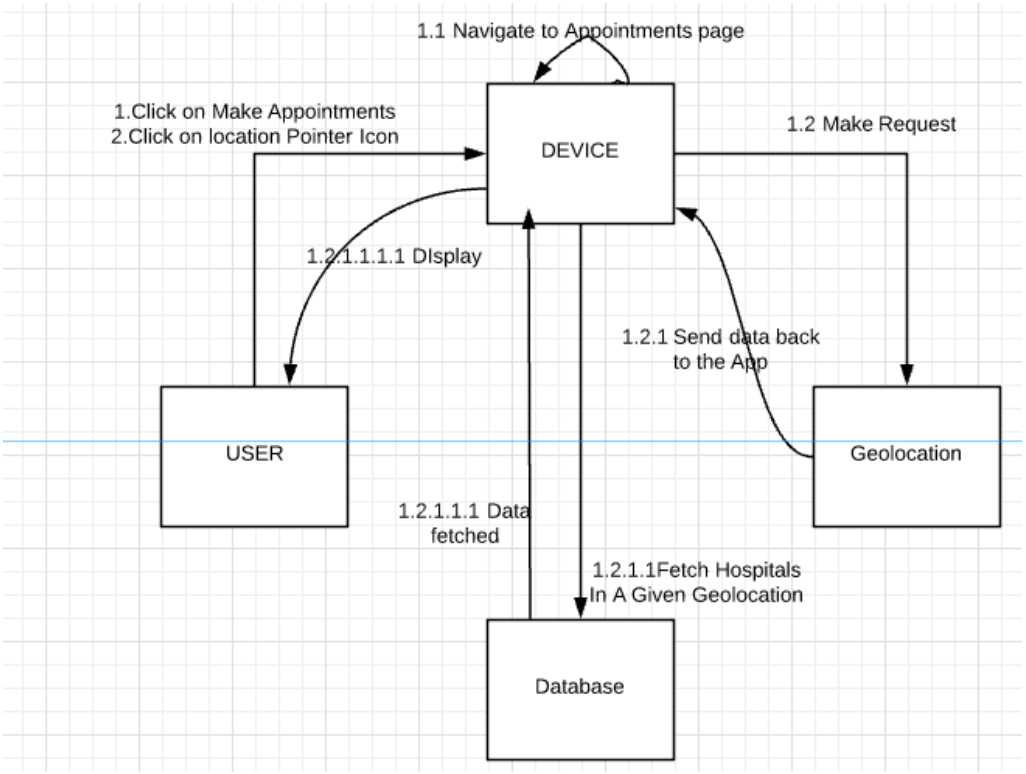


6.4 Make appointments to nearby Health Centers

Sequence Diagrams

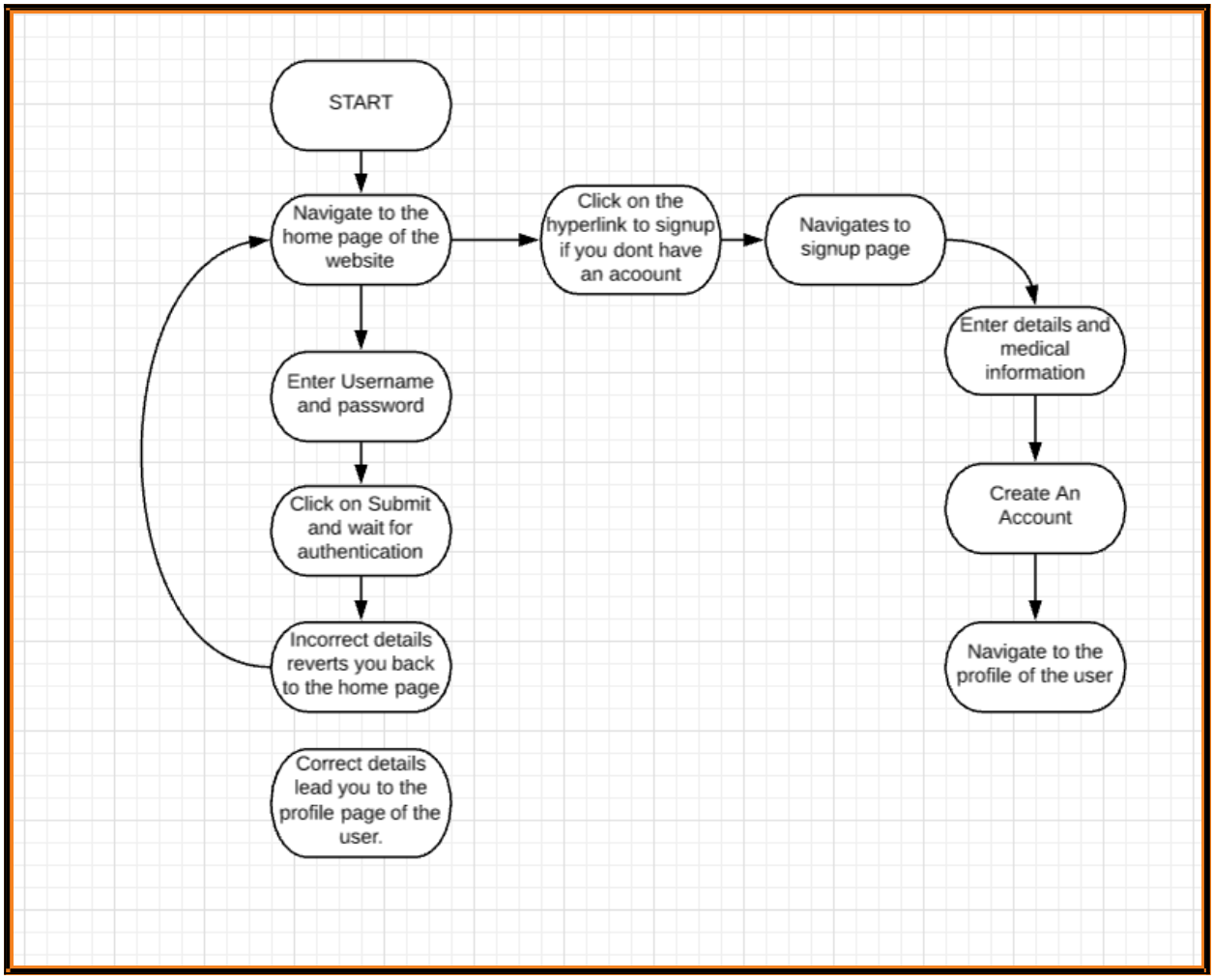


Collaboration Diagram

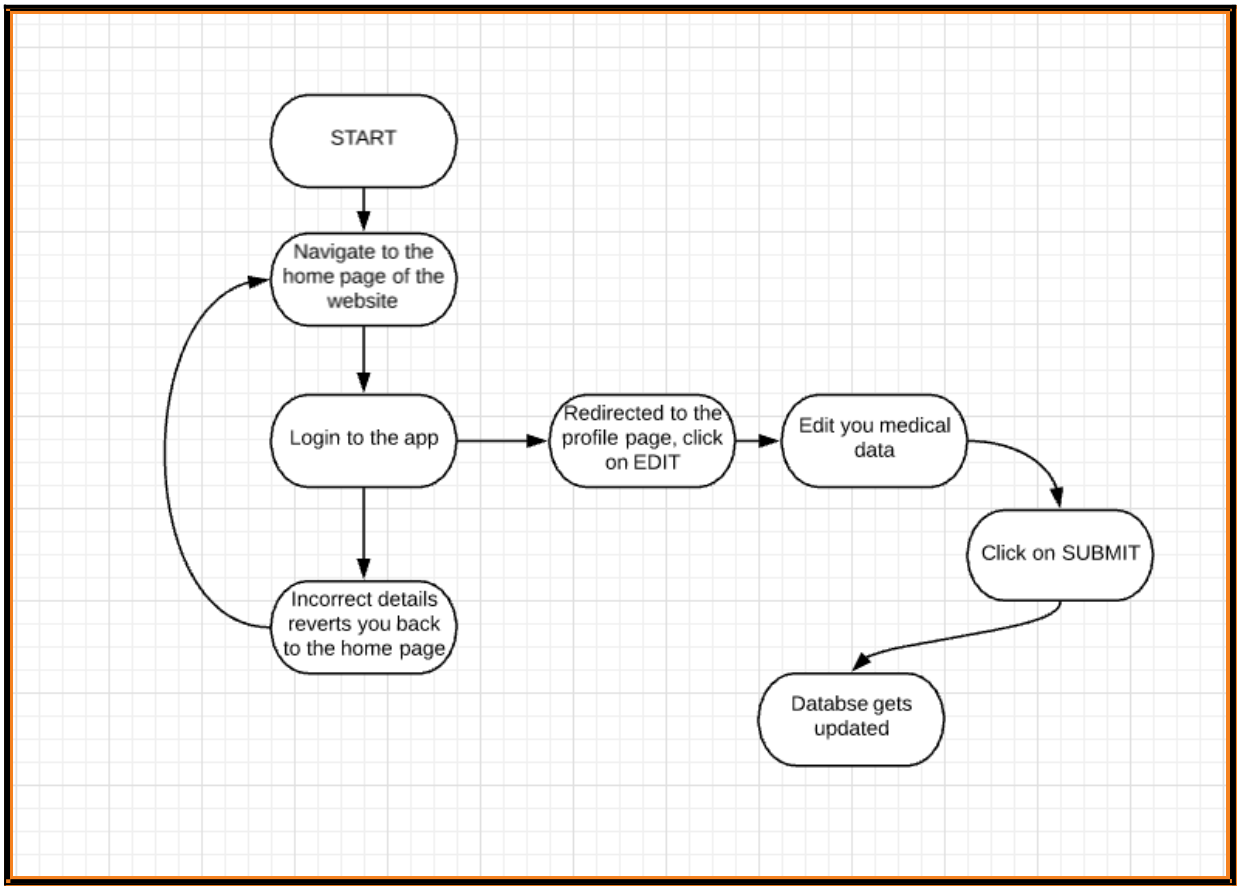


7. Detailed Descriptions and Activity Models For Major Operations:

7.1. Login/Signup:



7.1 Edit/Alter Medical Database Entries:



The application will provide users with an interface to update their health related information such as blood pressure, pregnancy and other information, and constantly keep track of the changes that happen with a progress graph

Functional Requirements

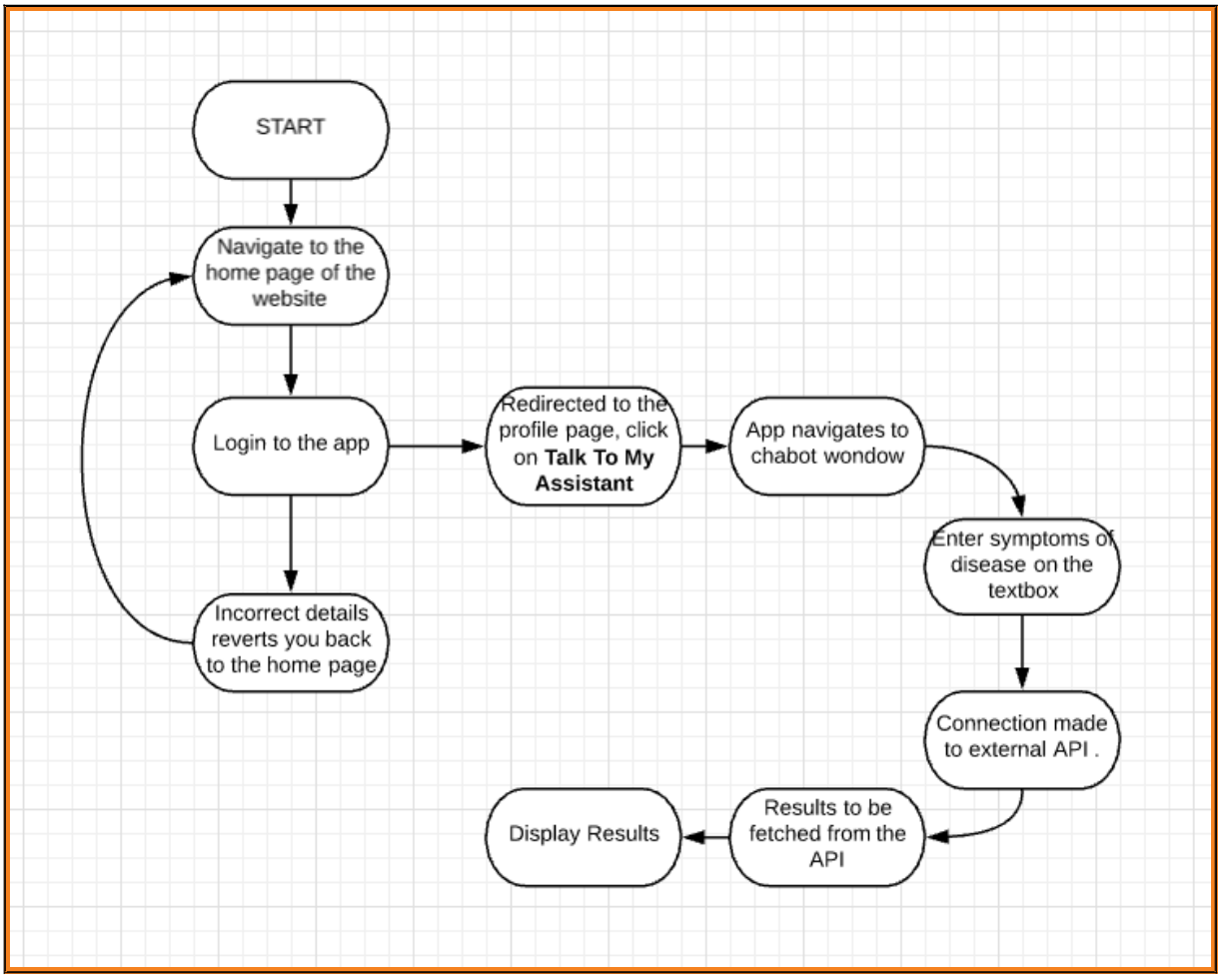
Purpose: Maintaining a record of health information for users and continuously track changes

Input: Constants fed by the user, based on his health information.

Processing: The data entered by the user will be stored in the database, and also processed to notify irregularities in the changes

Output: An output/ prediction that shows the changes trends of the user's health conditions in the form of textual output or graph plots.

7.3 Disease diagnosis and Chatbot Communication:



The software helps diagnosis ordinary ailments from symptoms with the help of an external API.

Functional Requirements

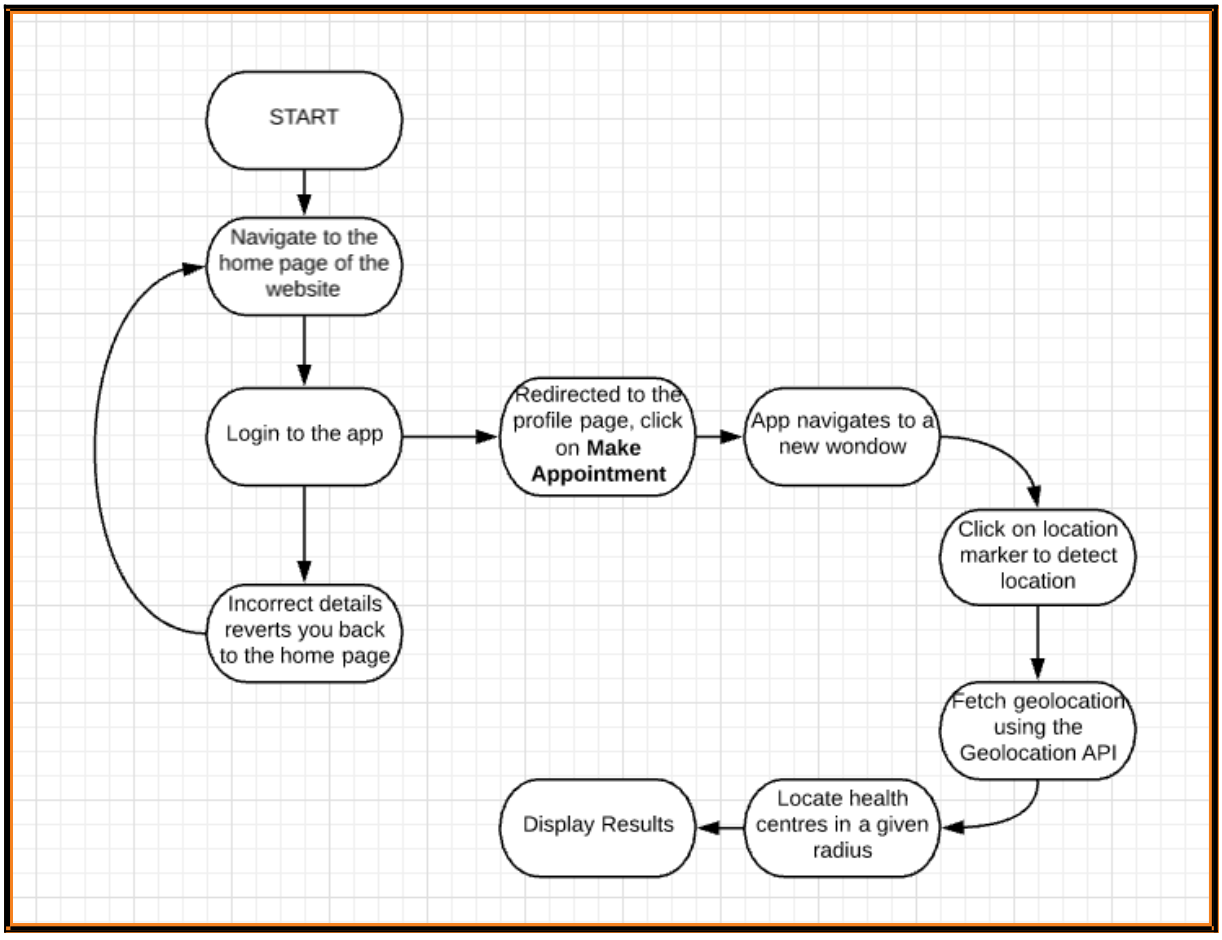
Purpose: Diagnosis of ailments with the help of an external API

Input: Comma separated symptoms in the form of textual entry

Processing: The feed thus generated is sent to an external API for processing.

Output: The output obtained from the API is displayed

7.4 Make Appointments



This feature will capture the user’s device location and locate hospitals around a said kilometre radius. After location of the hospitals, the user is free to make appointments in any one of these hospitals. The user can avail the contact details and directly contact the aforementioned health centres .

Functional Requirements

Purpose: To facilitate users to make appointments to nearby hospitals in case of emergency conditions.

Input: User click on locate hospitals, to generate list of nearby hospitals, then choose a hospital where the appointment should be made.

Processing: The Google Geolocation API is used to track the user’s location and also facilitate making appointments to the hospitals.

Output: A list of nearby hospitals is provided along with their contact details.