**SMART HEALTH SYSTEM**

Software Requirements Specification

**Group - 4 (5394.251)**

**Prepared by**

**Sriram, Putta**

**Sai Sandeep Potluri**

**Ganesh, Tummala**

**Kaushik, Ravuri**

**Sai Venkat Goud, Marri**

Table of Contents

[1. Introduction 1](#_Toc63688982)

[1.1. Purpose 1](#_Toc63688983)

[1.2. Scope 1](#_Toc63688984)

[1.3. References 1](#_Toc63688985)

[1.4. Overview 1](#_Toc63688986)

[2. Overall Description 2](#_Toc63688987)

[2.1. Product Perspective 2](#_Toc63688988)

[2.2. Product Functions 2](#_Toc63688989)

[**2.2.1.** **Signup** 2](#_Toc63688990)

[**2.2.2.** **Login** 2](#_Toc63688991)

[**2.2.3.** **Update Information** 2](#_Toc63688992)

[**2.2.4.** **Forgot Password** 2](#_Toc63688993)

[**2.2.5.** **Symptom Analyzer** 2](#_Toc63688994)

[**2.2.6.** **Payment Gateway** 3](#_Toc63688995)

[**2.2.7.** **Doctor Appointment Scheduler** 3](#_Toc63688996)

[**2.2.8.** **BMI Calculator** 3](#_Toc63688997)

[**2.2.9.** **Admin Functionality** 3](#_Toc63688998)

[2.3. User Characteristics 3](#_Toc63688999)

[2.4. Assumptions and Dependencies 3](#_Toc63689000)

[3. Specific Requirements 4](#_Toc63689001)

[3.1. Functional Requirements 4](#_Toc63689002)

[**3.1.1.** **Sign Up** 4](#_Toc63689003)

[**3.1.2.** **Account Login** 4](#_Toc63689004)

[**3.1.3.** **Update Information** 5](#_Toc63689005)

[**3.1.4.** **Forgot Password** 6](#_Toc63689006)

[**3.1.5.** **Symptom Analyzer** 6](#_Toc63689007)

[**3.1.6.** **Check-out and Payment** 6](#_Toc63689008)

[**3.1.7.** **Doctor Appointment Scheduler** 7](#_Toc63689009)

[**3.1.8.** **BMI Calculator** 7](#_Toc63689010)

[**3.1.9.** **Admin Functionality** 7](#_Toc63689011)

[3.2. Non-functional Requirements 7](#_Toc63689012)

[**3.2.1.** **Performance** 7](#_Toc63689013)

[**3.2.2.** **Reliability** 7](#_Toc63689014)

[**3.2.3.** **Availability** 8](#_Toc63689015)

[**3.2.4.** **Security** 8](#_Toc63689016)

[**3.2.5.** **Usability** 8](#_Toc63689017)

[**3.2.6.** **Software Quality Attributes** 8](#_Toc63689018)

# **Introduction**

## **Purpose**

The purpose of this document is to specify the requirements of Smart Health System (A Virtual Diagnostic system). The software requirement specification will describe the functionality, performance, external interfaces, attribute and any design constraint on the implementation. The intended audience of this document are the development team members of the project who will be implementing the project and the stakeholders who are investing in the design of the system.

## **Scope**

The scope of the project is to create and implement a virtual diagnosis system an provide with the required suggestions along with the doctor's availability and scheduling appointments for the Users(patients). The system will allow a customer to use our application to identify user’s disease based on their identifiable/known symptoms. The customer will be charged according to the pricing policy of the system. System will allow authorized admin users to login and update the system’s usage rates, view customer details, add/update the required data for customers and analyze the tools feedback data. This intelligent system makes use of AI inference engine to identify corresponding diseases as per user’s input of symptoms.

## **References**

[1] IEEE Software Engineering Standards Committee, "IEEE STD 830-1998, IEEE Recommended Practice for software Requirement Specifications", October 20, 1998

[2] Artificial Intelligence (to be updated with author names)

[3] Data Gathering Sites (to be updated accordingly)

## **Overview**

The following sections of the SRS is organized as follows:

**Section 2** will give the overall description of each functionality found in the system.

**Section 3** will give more technical details of the system functionalities.

# **Overall Description**

## **Product Perspective**

The Smart Health system is not a self-contained application. It allows the user to interact with the database to store information of the customer and help to retrieve data related to customer.

## **Product Functions**

### **Signup**

**Description:** This function allows the user to sign up to the application. They need to enter their email id, password, contact details and his medical identification number. Once they have signed up, they can use these details to login.

**Rationale:** It allows user to use our system in a way that suits him/her.

### **Login**

**Description**:This function allows the user to login to the application by entering their email id and password which they have used while signing up. If they have not signed up, they cannot login to this application. If they forgot their password, they can reset their password.

**Rationale**: This function grants user access to Smart Health System.

### **Update Information**

**Description:** This function allows the user to sign into the application and update the contact details.

**Rationale:** It allows the user to maintain his information up to date.

### **Forgot Password**

**Description:** This function allows the user to reset his password.

**Rationale:**A new link is sent to the user to reset his password.

### **Symptom Analyzer**

**Description:** This function allows the user to interact with the application by constantly answering the questions in a yes or no format.

**Rationale:**This function enables the user to identify the disease.

### **Payment Gateway**

**Description:** This function allows the customer to Check-out. The customer arrives at check-out kiosk and swipes same credit card that was used during check-in. The customer will be settled from pre-authorized transaction and receipt is displayed.

**Rationale:**This function allows customer to avail the system services*.*

### **Doctor Appointment Scheduler**

**Description:** This function allows the customer to view the available dates and times of the corresponding doctors they want to make an appointment and allow them to schedule an appointment once all the parameters are met.

**Rationale:** This will help the user to book an appointment with the doctor at his feasible times.

### **BMI Calculator**

**Description:** This function allows the user to input their weight and height in order to calculate their BMI (Body Mass Index).

**Rationale:** This offers a convenience to the user to check for their body’s health condition.

### **Admin Functionality**

**Description:** This function allows the authorized person to login into the system and update the data and collect the system’s feedback.

**Rationale:** This helps the developers to improve the system further and provide more flexibility for the end users.

## **User Characteristics**

The user is expected to have knowledge about how to use a computer and use a web application and know how to schedule appointments by processing online payments

## **Assumptions and Dependencies**

1. The application needs to have an active internet connection and a web browser to run.

2. The user should have a user account to access smart health system services.

# **Specific Requirements**

## **Functional Requirements**

### **Sign Up**

|  |  |
| --- | --- |
| **Description** | Please see section 2.2.1 |
| **Inputs** | The following information is required   1. First Name 2. Last Name 3. Email Address 4. Password with length of 6 characters with at least one number and one capital letter 5. Confirm Password 6. Date of Birth 7. Phone number. 8. Department |
| **Optional Inputs** | Middle Initial, Address Line1 (mailing), Address Line2, State, Country, Zip code, and Billing address. |
| **Source** | All the data will be entered by the user. |
| **Output** | An account will be created for the user. User is assigned a medical Identification number and user information is saved. |
| **Destination** | A new database entry will be made at the web server with all user account details. |
| **Requires** | None |
| **Pre-Condition(s)** | The user is presented with the web-interface to create a new account. The user is not currently registered with the website. |
| **Post-Condition(s)** | New user account created on webserver and assigned a student identification number. |

### **Account Login**

|  |  |
| --- | --- |
| **Description** | Please see section 2.2.2 |
| **Inputs** | The following information is required from the user:  i. Login Username: The users enter their username used while registration.  ii. Password: The password should match the corresponding username for successful login. |
| **Source** | The user will provide the information needed for this function. |
| **Output** | The user will be logged in successfully once correct username and corresponding password is entered or if users enters incorrect password message such as “Invalid Username or Password” is displayed. |
| **Destination** | Error message will be displayed in the same login page. |
| **Requires** | The users need to have a valid account. |
| **Pre-Condition(s)** | The user is displayed with the login page on the website. |
| **Post-Condition(s) (Successful Login)** | The user is displayed with Smart Health System home page. |
| **Post-Condition(s) (Invalid Login)** | The user is displayed with login error message. |

### 

### **Update Information**

|  |  |
| --- | --- |
| **Description** | Please see section 2.2.3 |
| **Inputs** | The following information is required from the user  i. Username: The users enter their username.  ii. Password: The password should match the corresponding student identification number for successful login.  iii. Information to be updated such as mailing address, password, name. |
| **Source** | Database will be searched for the unique username via login and user will be provided a web-interface to update their information. The user will provide the information needed to log on to the system and information to be updated. |
| **Output** | Information can be updated. |
| **Destination** | User account table in the database on the webserver. |
| **Requires** | A valid log in information to access the website. |
| **Pre-Condition(s)** | The user is displayed with the update profile option on the web-interface. |
| **Post-Condition(s)** | The user can check the updated information on their profile page. |

### **Forgot Password**

|  |  |
| --- | --- |
| **Description** | Please see section 2.2.4 |
| **Inputs** | Registered Email address |
| **Source** | Database will be searched for the username entered and if found, a new password will be sent to the associated email address. |
| **Output** | New password can be set-up. |
| **Destination** | User account table in the database on the webserver. |
| **Requires** | A valid email address is used while creating the user account on the website. |
| **Pre-Condition(s)** | The user is displayed with forgot password option on the web-interface. |
| **Post-Condition(s)** | The user can access the website with the student identification number and password sent to their corresponding email address. |

### **Symptom Analyzer**

|  |  |
| --- | --- |
| **Title** | **Description** |
| **Description** | See Section 2.2.2 for Description |
| **Inputs** | Symptoms of the user |
| **Source** | The AI Interface engine will look for symptoms entered by user. |
| **Output** | The probable disease based in symptoms entered by the user |
| **Destination** | None. |
| **Pre**-**condition** | The customer has logged into the system. |
| **Post**-**condition** | Doctors list displayed on screen |
| **Side**-**effects** | None. |

### **Payment Gateway**

|  |  |
| --- | --- |
| **Title** | **Description** |
| **Description** | See Section 2.2.4 for Description |
| **Inputs** | 16-digit credit/debit card number. If the customer is Frequent Parker, then customer is given an option to redeem points. |
| **Source** | All inputs are provided by the customer. |
| **Outputs** | Receipt is generated for customer. |
| **Destination** | Printed receipt is generated. |
| **Pre**-**condition** | User has scheduled a doctor appointment. |
| **Post-condition** | Customer is charged for scheduling a doctor appointment. |
| **Side**-**effects** | None. |

### **Doctor Appointment Scheduler**

|  |  |
| --- | --- |
| **Title** | **Description** |
| **Description** | See Section 2.2.4 for Description |
| **Inputs** | 16-digit credit/debit card number. If the customer is Frequent Parker, then customer is given an option to redeem points. |
| **Source** | All inputs are provided by the customer. |
| **Outputs** | Receipt is generated for customer. |
| **Destination** | Printed receipt is generated. |
| **Pre**-**condition** | User has scheduled a doctor appointment. |
| **Post-condition** | Customer is charged for scheduling a doctor appointment. |
| **Side**-**effects** | None. |

### **BMI Calculator**

|  |  |
| --- | --- |
| **Title** | **Description** |
| **Description** | See Section 2.2.4 for Description |
| **Inputs** | Height, Weight and Gender |
| **Source** | All inputs are provided by the customer. |
| **Outputs** | A BMI Index is displayed |
| **Destination** | Printed receipt is generated. |
| **Pre**-**condition** | User has scheduled a doctor appointment. |
| **Post-condition** | Customer is charged for scheduling a doctor appointment. |
| **Side**-**effects** | None. |

### **Admin Functionality**

|  |  |
| --- | --- |
| **Title** | **Description** |
| **Description** | See Section 2.2.5 for Description |
| **Inputs** | Customer information to register/update a frequent parker, updated parking rates to change prices |
| **Source** | All inputs are provided by the employee(admin) |
| **Outputs** | Admin can register frequent parker customer, update a frequent parker’s information, update parking rates and analyze parking statistics |
| **Destination** | The frequent parker is registered/updated and the parking rates are updated |
| **Pre**-**condition** | Admin has the credentials to access admin dashboard. |
| **Post-condition** | Customer is able to use frequent parker program after registration. If the price is updated, billing is done based on the update |
| **Side**-**effects** | None. |

## **Non-functional Requirements**

### **Performance**

The System is accessible from any web browser. The performance of the system depends on user’s internet connection and computer system. Concurrent access to the website doesn’t affect the system’s overall performance

### **Reliability**

In the event of system crash/failure, a backup system will be made available.

### **Availability**

The website is available to users 22 hours a day. The remaining two hours are considered for breakdowns, server issues and other maintenance activities.

### **Security**

To access personal information or make any changes, users must log into the system. Passwords and credit card information should be protected to prevent theft.

### **Usability**

The system interface should not require more than 15 minutes’ worth of tutorial to use. The user should not need to consult the tutorial every time they use the system interface. The users should be able to use the functions provided by the system website with no or minimal errors.

### **Software Quality Attributes**

#### **Maintainability**

All the system updates should be on system side and users should not be required to perform any maintenance activity.

#### **Robustness**

The estimated loss of data in case of system failure shall be less than one percent. Regular backups will be taken to keep the system robust.

*Software Requirements:*

* Mongo DB
* Express.js
* Node.js
* React JS

*Hardware Requirements:*

* RAM minimum 4 GB
* Mac, windows or linux machine
* The application will run over the internet, hence all the hardware shall require connecting internet.