

# **SOFTWARE REQUIREMENTS SPECIFICATION**

**FOR**

## **DIABETES DETECTION WEB APP**

**VERSION 1.0**

**PREPARED BY-PICO ZEN**

<b>Priyal</b>	101903525	ppriyal_be19@thapar.edu
<b>Aditi</b>	101903527	aaditi_be19@thapar.edu
<b>Manya Thakur</b>	101903531	mthakur1_be19@thapar.edu

**SUBMITTED TO:**  
MS. DEEPALI BHAGAT

**DATE CREATED:**  
SEPTEMBER 05, 2021

**THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, PATIALA**

# TABLE OF CONTENTS

TABLE OF CONTENTS	II
REVISION HISTORY	II
1. INTRODUCTION	1
1.1 PURPOSE	1
1.2 DOCUMENT CONVENTIONS	1
1.3 INTENDED AUDIENCE AND READING SUGGESTIONS	1
1.4 PROJECT SCOPE	2
1.5 REFERENCES	2
2. OVERALL DESCRIPTION	2
2.1 PRODUCT PERSPECTIVE	2
2.2 PRODUCT FEATURES	3
2.3 USER CLASSES AND CHARACTERISTICS	3
2.4 OPERATING ENVIRONMENT	10
2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS	11
2.6 USER DOCUMENTATION	11
2.7 ASSUMPTIONS AND DEPENDENCIES	11
3. SYSTEM FEATURES	11
3.1 SYSTEM FEATURE 1	11
4. EXTERNAL INTERFACE REQUIREMENTS	14
4.1 USER INTERFACES	14
4.2 HARDWARE INTERFACES	14
4.3 SOFTWARE INTERFACES	14
4.4 COMMUNICATIONS INTERFACES	14
5. OTHER NONFUNCTIONAL REQUIREMENTS	15
5.1 PERFORMANCE REQUIREMENTS	15
5.2 SAFETY REQUIREMENTS	15
5.3 SECURITY REQUIREMENTS	15
5.4 SOFTWARE QUALITY ATTRIBUTES	15
6. OTHER REQUIREMENTS	15

## REVISION HISTORY

NAME	DATE	REASON FOR CHANGES	VERSION
PRIYAL ADITI MANYA THAKUR	08-09-2021	INITIAL RELEASE	1.0

# **1. Introduction**

## **1.1 Purpose**

The purpose of the Diabetes Detection Web App is to gather and process the user's basic information which will include things like BMI, insulin levels, age, blood pressure, skin thickness and pregnancies (if any) and then predict the likelihood of that particular user having diabetes.

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters, features and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. The overall description also contains purpose, features, interfaces, constraints for operation, reaction to different inputs, scope of the product.

## **1.2 Document Conventions**

Font : Times New Roman

Text colour : Black

Bold Font : Emphasizing heading, subheading, section, subsection, titles

Underline: Main heading

## **1.3 Intended Audience and Reading Suggestions**

This document is intended for potential developers, consumers, testers, project managers and marketing staff to understand the purpose, mechanism, requirements, scope and operations of the said project.

The reading suggestions are as follows:

**Developers:** need to become intimately familiar with the SRS so that they can review project's capabilities and understand where their efforts should be targeted to improve or add more features to it (design and code the application – it sets the guidelines for future development).

**Project managers:** also needs to become intimately familiar with the SRS as they have to keep check of project milestones and deadlines.

**End users:** need to become familiar with features and software, external requirements. Users can then verify the functional requirements and approve the project according to mentioned criteria.

**Project testers:** need an understanding of the system features to develop meaningful test cases and give useful feedback to the developers. They can use this document as a base for their testing strategy as some bugs are easier to find using a requirements document. This way testing becomes more methodically organized.

**Marketing staff:** need to become accustomed to the various product features in order to effectively advertise the product i.e., overall description. The marketing staff also needs to understand the external interface requirements to sell the product

The SRS is written in the standard IEEE format. The readers should read the whole SRS sequentially to gain a proper understanding of the project.

## 1.4 Project Scope

This Web app will be open-source software freely available to all and hence will enable common audience to have a diabetes check without needing to visit a doctor. Based on the parameter's user enters, it predicts whether it is benign or malignant by learning from the parameters.

The three categories of users in this software are:

- Admin
- User
- Guest

## 1.5 References

<http://en.wikipedia.org/wiki/SoftwareEngg>.

<http://ieee.org>

<http://www.slideshare.net/udaschand940/software-requirementspecification-srs2>

## 2. Overall Description

### 2.1 Product Perspective

Our web application uses the power of machine learning to incorporate diabetes detection algorithms with as much accuracy as possible. Usual identifying process for diabetic patients needs more time and money which cannot be afforded by everyone, so this web application will be developed and used free-of-cost which will have the capability to predict whether the patient has diabetes or not. Our main aim of this exploration is to build a web application based on the higher prediction accuracy of some powerful machine learning algorithm.

The user will be able to access the web app via the internet. The website will be hosted on cloud-based web servers. The user will require to register and then login. After successful login the user enters his data. The details required by the application will be composed of:

- 1: Pregnancies:** Number of times pregnant
- 2: Glucose:** Plasma glucose concentration 2 hours in an oral glucose tolerance test.
- 3: Blood Pressure:** Diastolic blood pressure (mm Hg)
- 4: Skin Thickness:** Triceps skinfold thickness (mm)
- 5: Insulin:** 2-Hour serum insulin (mu U/ml)
- 6: BMI:** Body mass index (weight in kg/ (height in m) <sup>2</sup>)
- 7: Diabetes Pedigree Function:** Diabetes pedigree function
- 8: Age:** Age (years)
- 9: Outcome:** Class variable (0 or 1) 268 of 768 are 1, the others are 0

All variables are self-known or available in sample blood tests.

The application will be pretrained on various machine learning algorithms like logistic Regression, Support Vector Machines, Decision Forest, Gradient Boosting, neural networks with pre-processed inputs. The algorithm providing the best predictions will be chosen and used for predictions. The prediction models on getting user input will process the information accordingly and present the results accordingly. If the user has registered, the results will be displayed and saved in the database which could be accessed again by the user later on as well.

## 2.2 Product Features

The web app should be provided with following features:

1. **User Authentication:** The web app must be able to authenticate the user by matching credentials already registered by the user. If the credentials don't match then the user won't be able to log in and an error message will be displayed.
2. **Reliable database:** It must be able to store the information provided by a registered user. Other users must not be able to access the database. Only the owner of information should be able to access their information and the admin can access the database for system management purposes from time to time.
3. **Reliable operation and maximally accurate results:** It must be able to analyse the information, improvise the prediction model and predict a reliable enough outcome.
4. **History:** Users can request to view their history to keep a track of their health records. History access of individuals must also be restricted to the particular individual only and no other user must be able to access the history of other users.
5. **Printing:** Users can request for a print out of their diabetes report.

## 2.3 User Classes and Characteristics

There are 3 user/stakeholders of this system namely:

- Guest
- Admin
- User

### Characteristics of user types:

- **Guest:** Mostly app- illiterate people who want to understand the internal working of the app and its features and functioning. They can only view the details about the app and not save his/her record or produce a report.
- **User:** People who actually make use of the app to meet their requirements and expectations from this app. They have to register beforehand and then login to access all the available functions. These are app-literate people and don't have much difficulty in using the system.
- **Admin:** The admin will have all the access to edit and modify the website. He can maintain the website or keep track of things going around the website.

### Characteristics of interfaces:

- **User interface:**  
The user interface will consist of a login or signup page wherein the user will enter the credentials and move on to the home page. Here the user will be able to enter the required details to check whether they are diabetic or not. The result will also be displayed on this interface.

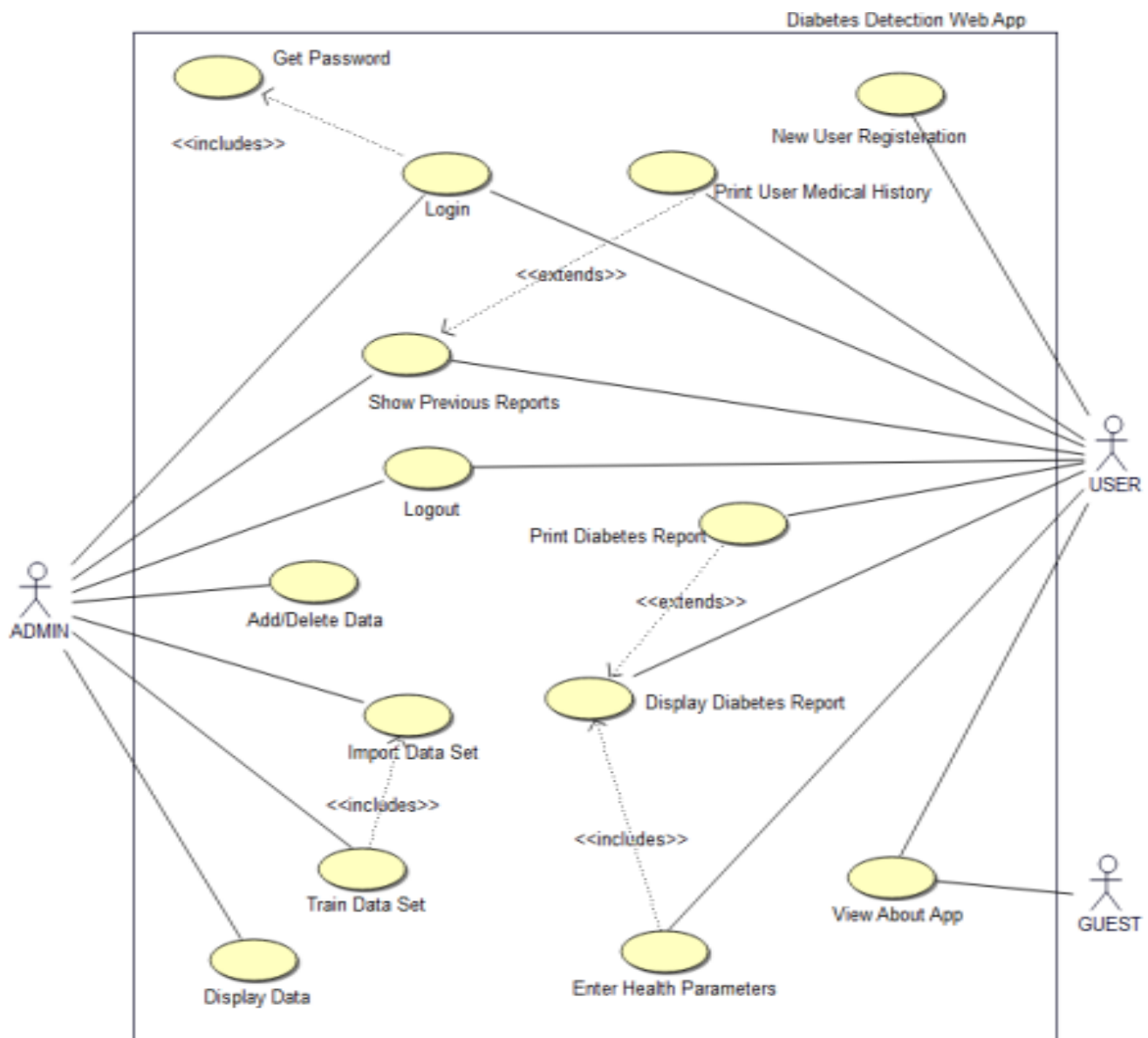
Along with this, the user also has an option to visit pages where information about diabetes is displayed and the general working of how we predict the result is displayed.

**NOTE:** Prerequisite knowledge is not required for signing up but it is recommended for fresh users to go through the documentation and get a small idea of how things go around in the app.

- **Admin interface:**

The admin will have the ability to import datasets and to add or delete the data. The admin will have a login page similar to that of the users and a home page which helps in manipulation of datasets.

## USE CASE DIAGRAM:



**Use Case Scenario**

Use Case Title:	Login
Use Case Id:	1
Actors:	User, Admin
Description:	Login is used to give access of website to users who are registered.
Pre conditions:	User, Admin must get appropriate user id and password by registration.
Task Sequence	i. User, Admin enter user-id. ii. User, Admin enter password. iii. Entered user-id and password is checked if exists or not. iv. The system directed to corresponding home page for user, admin if user-id and password is correct. v. The system shows error message if authentication fails.
Post conditions:	Corresponding home page for user, admin must be opened.

Use Case Title:	Show Previous Reports
Use Case Id:	2
Actors:	Admin, User
Description:	User can view his previous medical reports and admin can see the previous reports of all the registered users.
Pre conditions:	User must be registered and must have some previous reports.
Task Sequence	i. User must login. ii. Home page for user will open. iii. Select “Show History” option.
Post conditions:	Corresponding user history will be displayed



Use Case ID:	3
Use Case:	Enter Health Parameters
Use Case Purpose:	To get patient medical details like blood pressure, glucose level, age, etc.
Use case Description:	Parameters are taken as input to predict the result and generate the report.
Assumptions:	The users are ready with their medical details
Variations:	No variants are available for this use case
Trigger:	When the user chooses the take test option then he will be asked to enter the health parameters.
Primary Actors:	The primary actor here is the User.
Secondary Actors:	The secondary actor here is the Admin.
Pre-Conditions:	The user must be registered and logged in.
Normal Scenario:	i. User must login. ii. Home page for user will open. iii. User will select the 'Take Test' option. iv. Page will open for the users to input their details. v. User will click on the submit button.
Extension points:	If the user cancels to input the health parameters before clicking the submit button, then the page will refresh. If the user missed to fill any field, then the field will be highlighted and user will be prompted to fill it.
Alternate Scenario:	No alternate scenario.
Post Conditions:	Corresponding user results will be displayed.
Special Requirements: Performance	The result should be displayed within 5 seconds.

Use Case Title:	Print User Medical History
Use Case Id:	4
Actors:	User
Description:	User can print his previous medical details.
Pre conditions:	User must be registered and must have some previous reports.
Task Sequence	i. User must login. ii. Home page for user will open. iii. Select “Show History” option. iv. User will select the ‘Print History’ option.
Post conditions:	Corresponding user history will be printed

## 2.4 Operating Environment

This web app can run on any Windows operating system and any Linux- based GUI operating system with sufficient memory. To support the web application any of the following browsers must be available:

- Apple Safari 7+
- Google Chrome 44+
- Microsoft Internet Explorer 10+
- Mozilla Firefox 40+

### Briefly:

- Front end- HTML
- Operating System- Windows XP/Vista/7/8/10
- Back end- Python, MySQL

## 2.5 Design and Implementation Constraints

End user must have basic knowledge and understanding of the English language.

Knowledge of basic computer usage is required.

Using the latest version of the web browser is recommended.

Since our web application will be supported by a third-party web hosting provider, availability and speed of connection will rely on them.

For developers, who want to understand minute details like implementation require to have basic knowledge of ML algorithms like logistic Regression, Support Vector Machines, Decision Forest, Gradient Boosting and neural networks; HTML, CSS, python and flask.

## 2.6 User Documentation

There will be a user manual page on the website on how to use button in the website which guides the guests & users about the step wise step process to navigate & make use of the functions in the website for their better understanding. A YouTube video tutorial will also be provided.

## 2.7 Assumptions, Dependencies and Constraints

The following list presents the constraints, assumptions, dependencies or guidelines that are imposed upon implementation of the diabetes detection web app:

- The product must have a user-friendly interface that is simple enough for all types of users to understand.
- Response time for loading the software and for processing a request should be no longer than five seconds.
- A general knowledge of basic computer skills is required to use the product.
- The central database server should be updated regularly.
- We assume that the health parameters provided by the user are correct.
- The prediction result will depend on the strength of the algorithm though we will use the algorithm which will provide us with best results.
- The external libraries or features used may/may not be deprecated in the future.
- User must be aware about his/her details required by the web application for diabetes detection.

## 3. System Features

### 3.1 User Registration

#### 3.1.1 Description and Priority

The registration process is a must for all the users before they use the application. This module has the highest priority when compared to all the other modules. This model allows the user to register himself at the application and become eligible to become a user of the application.

### **3.1.2 Stimulus/Response Sequences**

This module has text boxes where the user can create his username name and password. If the username is already being used or the password does not match the requirements, then the system will pop a message box and registration will fail. For successful registration the user must enter all the details correctly.

### **3.1.3 Functional Requirements**

It is mandatory to register oneself before using the application, without this no one can use the application.

## **3.2 Login**

### **3.2.1 Description and Priority**

The login form is used by all the users and admin. This module has the second highest priority when compared to all the other modules. This model allows the user to enter his username and password in order to get access to the application as a user and make use of the application and his recorded data.

### **3.2.2 Stimulus/Response Sequences**

This module has text boxes where the user can enter his username name and password. If the necessary information is not provided or if invalid inputs are given by the user, then the system will pop a message box and user login will fail. For successful login the user must enter all the details correctly.

### **3.2.3 Functional Requirements**

Only authorized users are allowed to login. The authorized users are the admin and the general users who are already registered. If invalid user name or password is given the system should inform the user. If unauthorized users try to access then it should not allow the user to access the application.

## **3.3 View App Details**

### **3.3.1 Description and Priority**

This feature can be used by anyone whether that person is registered or not. This feature talks about the app details - how the app would work, operations it can perform, suitable environment and scope of the application and other details.

### **3.3.2 Stimulus/Response Sequences**

There will be an option on the home page of the application to view about the app. Clicking on that option the user or the guest can view the app details.

### 3.3.3 Functional Requirements

This will guide the users to how to use the app and steps to follow for performing various operations through this application. It will also tell the users about the ML models and other implementation details used to predict the result so that the users can have a better understanding about the app.

## 3.4 Take Diabetes Test

### 3.4.1 Description and Priority

This feature can be used only by the users after getting logged in. Using this feature users can take the diabetes prediction test. Users must be aware of the following details to successfully complete this step:

- **Pregnancies:** Number of times pregnant
- **Glucose:** Plasma glucose concentration 2 hours in an oral glucose tolerance test.
- **Blood Pressure:** Diastolic blood pressure (mm Hg)
- **Skin Thickness:** Triceps skinfold thickness (mm)
- **Insulin:** 2-Hour serum insulin ( $\mu$ U/ml)
- **BMI:** Body mass index (weight in kg/(height in m)<sup>2</sup>)
- **Diabetes Pedigree Function:** Diabetes pedigree function
- **Age:** Age (years)
- **Outcome:** Class variable (0 or 1) 268 of 768 are 1, the others are 0

All variables are self-known or available in sample blood tests.

### 3.4.2 Stimulus/Response Sequences

The user will be asked to provide their health details. If the user misses to enter any detail, then a message will pop up and diabetes detection will fail. If user successfully enters all the details correctly then this detail will be used to predict whether the user has diabetes or not. The algorithm selected in the data testing and training phase will then be used for diabetes detection for the required user. The output generated will finally be displayed to the user.

### 3.4.3 Functional Requirements

During the testing and training phase, the application must go through some pre-processing techniques and then apply various machine learning algorithms and pick the algorithm giving the best results. This algorithm will then be used for diabetes prediction for users on our web application. The user can then test their diabetes after entering the required details. The user must enter their details accurately for maximally accurate predictions. Mistakes in entering the details will lead to wrong predictions and that may be harmful for the users.

## 3.5 Show User History

### 3.5.1 Description and Priority

This feature can be used by the respective users. A user must be able to access only his own history and must not be able to access the history information of someone else. The admin can see the health

history of all the users for management purposes but it will be assured to be completely confidential and users need not to worry about their privacy breach.

### 3.5.2 Stimulus/Response Sequences

On the homepage of the application there will be an option to see the user history. Clicking on that option the health history of the user will be displayed and the user can go through it.

### 3.5.3 Functional Requirements

Using this option, the user can see their health history and thus keep a track of their health. The admin can see the health history of all the users and use this to predict the health trend.

## **4. External Interface Requirements**

### **4.1 User Interfaces**

The user interface provided by our application will be very straightforward, simple and subtle in light of a user- friendly interface.

It features input boxes for users to type in the input and buttons for the user to click in. The system will verify the inputs and move forward accordingly.

The product requires very limited graphics usage with just a simple keypad for taking the user input. The product does not require usage of sound or animation.

Sound is not an essential feature but it can be considered for future variants of the system wherein the user will be greeted by his name.

### **4.2 Hardware Interfaces**

COMPONENT	MINIMUM	MAXIMUM
Processor	1.9 GHz x86- or x64- bit dual core processor with SSE2 instruction set	3.3 GHz or faster 64- bit dual core processor with SSE2 instruction set
Memory	2- GB RAM	4- GB RAM or more

### **4.3 Software Interfaces**

For the front end HTML5, CSS and bootstrap will be used for GUI. Flask, python, and machine learning will be used in the backend. A python ML model giving the best predictions out of all other models used for testing in the testing phase will be used to predict the disease stage.

### **4.4 Communications Interfaces**

The following list presents the external interface requirements:

- **HTTPS:** Hypertext Transfer Protocol secure is used to securely transfer the data from server to web browser by using MD5 encryption to increase data transfer security.

## **5. Other Non-functional Requirements**

### **5.1 Performance Requirements**

It is a software that will be available 24/7 365 days of the year providing reliable results of which reports can be shown in hospitals, doctor tests, etc. The user doesn't need any such performance requirement as ML models which will be the backbone for predictions will be carried out and deployed while creating the web app. So the user doesn't need to contribute for any performance requirement except for an up to date browser and an active internet connection with fine speed.

### **5.2 Security Requirements**

Security is our main concern and so the data/files store will be protected and not leaked with other security methods like:

- Maintain confidentiality of User's data.
- Providing login id and strong password to each user
- Clean and easy to use interface.
- Routine backups of DB to prevent any data loss

All in all, security breaches and users' privacy will be considered carefully. Users are recommended to install antivirus software on their systems as a precaution.

### **5.3 Software Quality Attributes**

- **Consistent uptime**  
This system will be able to stay up and running at least 98% of the time. Any downtime would be due to maintenance or upgrades. This downtime also includes any potential failures/crashes.
- **Easy Interface**  
The user interface will be easy to understand and use and no one will have any trouble in handling the system making it extremely user friendly.
- **Flexibility**  
Because of its simplicity, it's flexible enough to incorporate UI and other improvements in the future.
- **Testability**  
The system will be easy to test with the help of the test cases that will be designed during the development of the system.
- **Portability**  
The app will easily work on any operating system with any web browser and proper internet connection.

## **6. Other Requirements**

No other requirements are needed.