

**Software Design Specification**

**for**

**Disease Diagnosis System**

**Version 1.0**

Date: January 8, 2020

**Supervisor:**  **Group Members:**

Sir Ahsan Shah Kareema Batool (2018174)

Umair Ayaz Aslam (2018484)

Lalain Aslam (2018091)

Revision History:

|  |  |  |
| --- | --- | --- |
| ***Revision History*** | ***Date*** | ***Comments*** |
| 1.00 | 4th January, 2020 | First Version |
| 2.00 |  |  |
|  |  |  |

Document Approval:

The following document has been accepted and approved by the following:

|  |  |  |
| --- | --- | --- |
| ***Signature*** | ***Date*** | ***Name*** |
|  |  |  |
|  |  |  |
|  |  |  |

List of Contents

[1. INTRODUCTION 6](#_Toc402273396)

[1.1. PURPOSE 6](#_Toc402273397)

[1.2. PRODUCT SCOPE 6](#_Toc402273398)

[1.3. OVERVIEW 6](#_Toc402273399)

[2. THE OVERALL DESCRIPTION 7](#_Toc402273400)

[2.1. PRODUCT PERSPECTIVE 7](#_Toc402273401)

[3. WORK BREAKDOWN STRUCTURE](#_Toc402273402) 8

[4. Design](#_Toc402273403) 9

[4.1 ARCHITECTURAL DESIGN](#_Toc402273404) 9

[4.2. Why we choose layered Architecture Design?](#_Toc402273405) 10

[4.3. MODULE IDENTIFICATION](#_Toc402273406) 10

[5. 4+1 ARCHITECTURE VIEW MODEL](#_Toc402273407) 11

[5.1. Use Case View](#_Toc402273408) 12

[5.2. Logical View:](#_Toc402273409) 19

[5.3. Development View](#_Toc402273410) 20

[5.4. Process View](#_Toc402273411) 21

[5.5. Physical View](#_Toc402273412) 22

[5.6. User Interface Design](#_Toc402273413) 23

List of Figures

[Figure 1 Work Breakdown Structure 8](#_Toc61019968)

[Figure 2 Layered Architecture Design 9](#_Toc61019969)

[Figure 3 4+1 Architecture View 11](#_Toc61019970)

[Figure 4 Use Case for user's input 12](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019971)

[Figure 5 Use case for training ML models 13](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019972)

[Figure 6 Use case for Performance Accuracy Computation 14](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019973)

[Figure 7 Use case for Preprocessing of user symptoms 15](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019974)

[Figure 8 Use case for Disease Prediction 16](#_Toc61019975)

[Figure 9 Use case for Suggesting Suitable Hospitals 17](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019976)

[Figure 10 Use case for Managing Databases 18](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019977)

[Figure 11 Context Diagram 19](file:///C:\Users\Abdul%20Rauf\Downloads\SDS.docx#_Toc61019978)

[Figure 12 Component Diagram 20](#_Toc61019979)

[Figure 13 Activity Diagram 21](#_Toc61019980)

[Figure 14 Physical Diagram 22](#_Toc61019981)

[Figure 15 23](#_Toc61019982)

[Figure 16 23](#_Toc61019983)

[Figure 17 24](#_Toc61019984)

[Figure 18 24](#_Toc61019985)

[Figure 19 25](#_Toc61019986)

[Figure 20 25](#_Toc61019987)

**List of Tables**

[Table 1: Terms used in this document and their description 6](#_Toc61021184)

[Table 2: Table for User’s input 12](#_Toc61021185)

[Table 3: Table for training ML models 13](#_Toc61021186)

[Table 4: Table for Performance Accuracy Computation 14](#_Toc61021187)

[Table 5: Table for processing of User Symptoms 15](#_Toc61021188)

[Table 6: Table for disease prediction 16](#_Toc61021189)

[Table 7: Table for suggesting suitable hospitals 17](#_Toc61021190)

[Table 8: Table for managing databases 18](#_Toc61021191)

# INTRODUCTION

## PURPOSE

The purpose of this document is to detail the implementation of requirements defined in the software requirement specification by different architectural models and patterns. This depicts the different views of the intended product which is provides diagnosis of diseases based on symptoms that are provided by the users through deep learning. Furthermore, it showcases a list of suitable hospitals based on user’s budget and locality.

## PRODUCT SCOPE

This is a personalized diagnosis software that gets the symptoms from the users and suggests a list of diseases based on the given symptoms using various machine learning classification and regression models. These models have been trained and tested on UCI machine learning repository datasets. For each model, it computes the performance accuracy using different parameters such as cross validation folds. It also maintains various DB of users, hospitals, and diseases. Additionally, it provides a list of hospitals based on the budget and the locality of the user.

Table 1: Terms used in this document and their description

|  |  |
| --- | --- |
| Name | Description |
| Agile | A development style where solutions evolve through collaboration between self-organizing cross functional teams. |
| Database (DB) | It is a collection of information that is organized for easy access. |
| Deep Learning | AI function that mimics the working of human brain in data. |
| Machine Learning models | Techniques that are used to implement deep learning. |
| Performance accuracy | A metric for evaluating Machine Learning models. |
| Plan-driven | A development style that attempts to plan and anticipate all the features of a software product. |

## OVERVIEW

This software design specification covers the DD system. This product is basically a disease detection application that takes the symptoms from the user and diagnoses using multiple classification ad regression models. These models are trained on dataset which consists of various diseases along their symptoms. Additionally, it provides a list of suitable hospitals based on user’s budget and locality.

# THE OVERALL DESCRIPTION

The description of the proposed system is given below. This description defines the product in different directions.

## 2.1. PRODUCT PERSPECTIVE

The need for this product arose from the ongoing pandemic that has put a lot of burden on the healthcare facilities. This product decreases the need of going out to the hospitals for a specific treatment by diagnosing the disease and suggesting the suitable hospitals from the comfort of your home. This is a self-contained product which comprises of the following components:

* Hospital:

It is a DB for hospitals that includes hospital name, hospital location, average expenses etc.

* Disease:

It is a DB that has all the diseases as rows and symptoms as columns. It contains all the values in binary numbers where 0 represents fail (disease not found) and 1 represents hit (disease found).

* User:

It is a DB that stores the patient name, patient age, gender, contact information, home address, budget, city etc.

* Machine Learning Techniques:

Classification and regression models such as LR, KNN, RF, MNB, SVM, and MLPNN etc.

* Preprocessing of data:

1. Preprocessing of dataset:

The component that preprocesses the entities present in disease DB.

1. Preprocessing of user input:

The component that preprocesses the inputs entered by user.

* Model Training:

The component that is responsible for training the machine learning techniques specified above.

* Model Testing:

This component tests the models on 10% of datasets from disease DB.

* Performance Accuracy Test:

This computes the performance accuracy of each machine learning models using the cross validation folds and selects the best model amongst all.

# WORK BREAKDOWN STRUCTURE

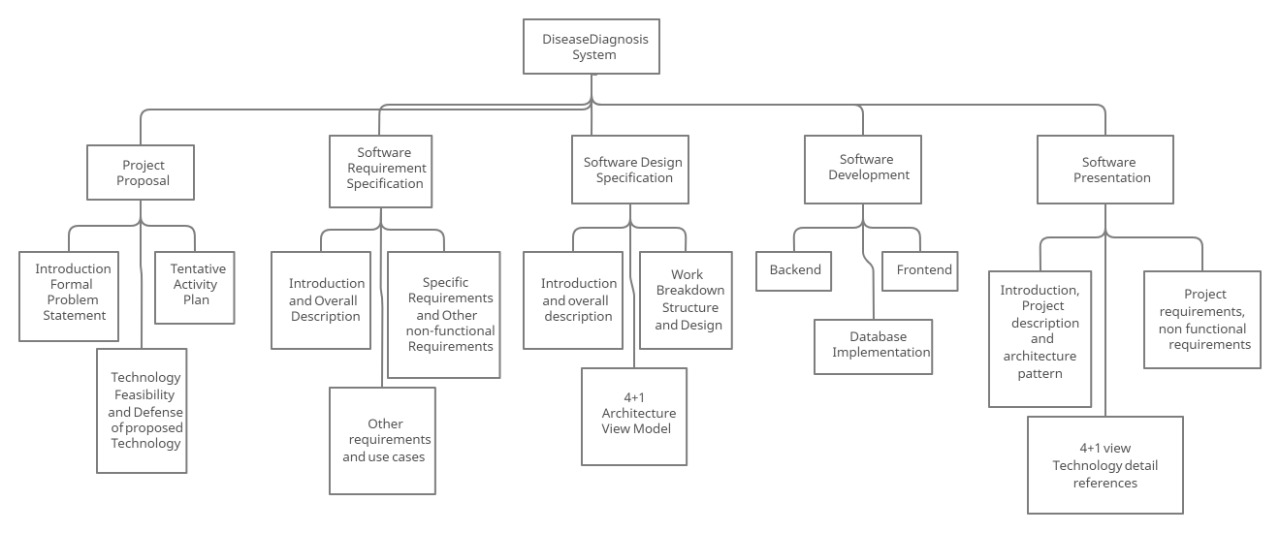


Figure 1 Work Breakdown Structure

This is the work breakdown of DD system on the basis of project phases. Here, each project phase is divided into set of tasks where each set is assigned to a specific person in the development team. The project is being designed using both plan-driven and agile approach working in parallel. It is plan-driven because of extensive documentation and at the same time, it is also agile as documentation and coding was being done in parallel making it as a hybrid approach.

# Design

## LAYERED ARCHITECTURAL DESIGN

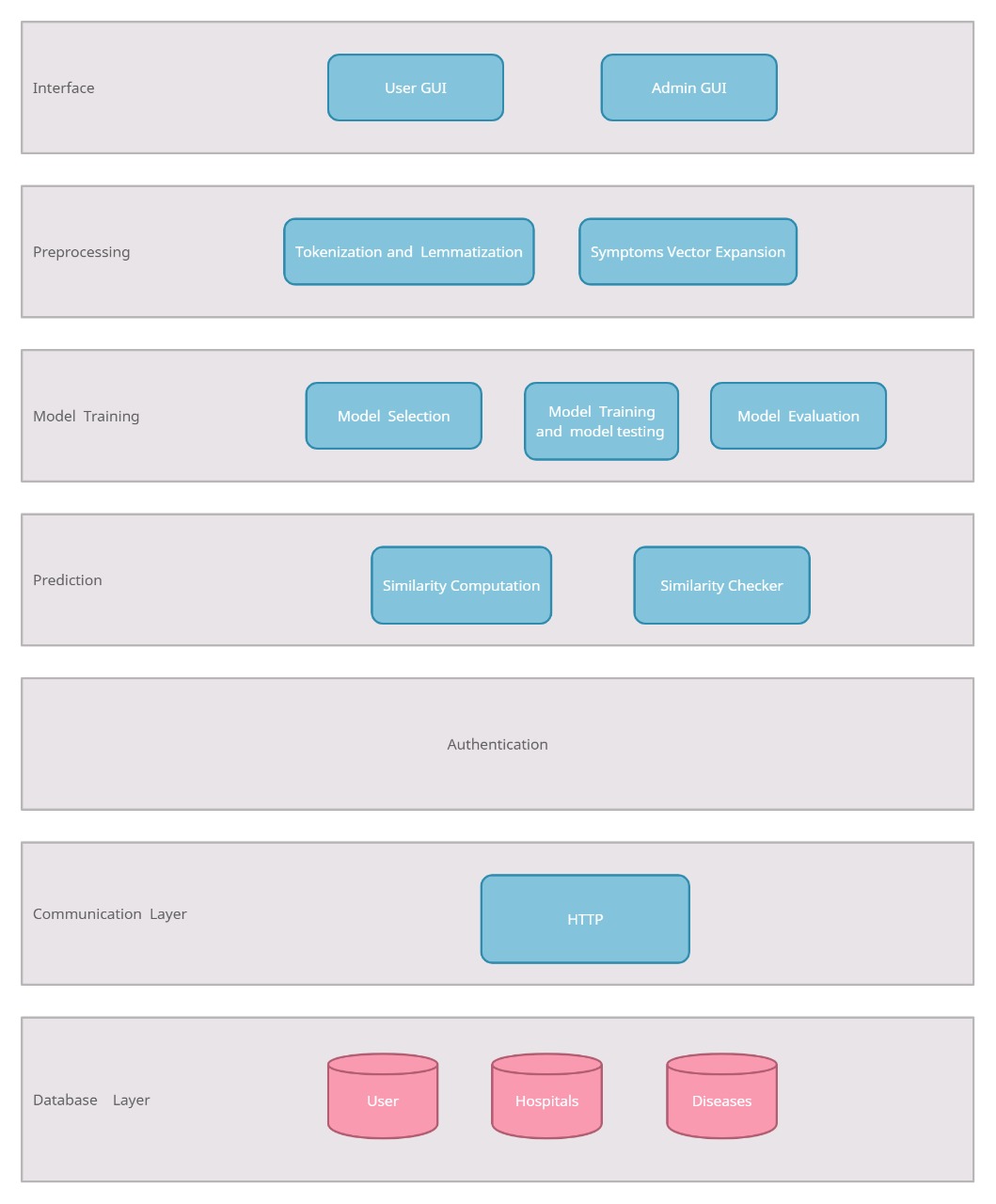


Figure 2 Layered Architecture Design

## Why we choose layered Architecture Design?

* It provides security of data by keeping it in the core layer.
* It supports the incremental development of the system in different layers.
* Each layer is a set of related services thus the architecture provides high degree of cohesion within the layers.
* Each layer may use only lower layer which constrains the amount of coupling.
* It supports reusability and interchangeability as each layer is cohesive and is coupled only to lower layers.

## MODULE IDENTIFICATION

The identified module of DD system are written below:

User:

* User personal information form
* User symptoms
* Diseases list
* Hospitals list

Admin:

* Login
* User database
* Hospitals database
* Disease database
* Logout

# 4+1 ARCHITECTURE VIEW MODEL

In this section, you draw the architecture using the views defined in the “4+1” model.

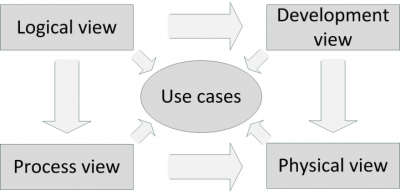


Figure 3 4+1 Architecture View

## Use Case View

This is a list of use-cases that represent major functionality of the final system:

1. User’s Input

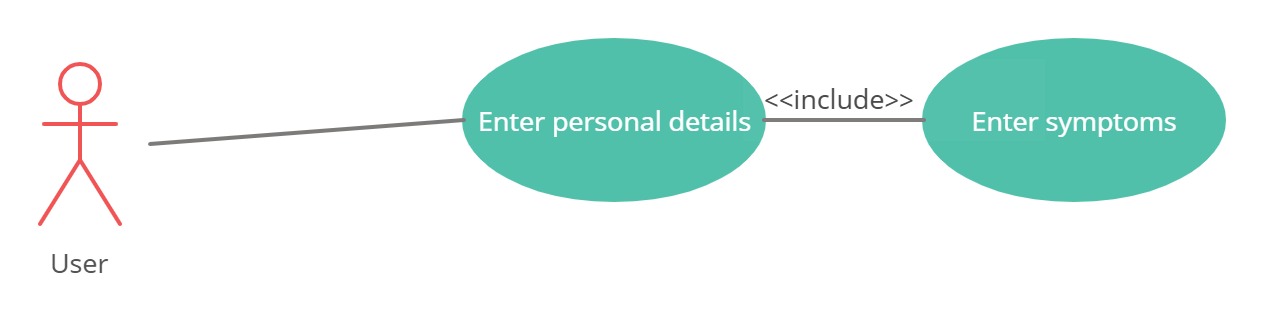


Figure 4 Use Case for user's input

Table 2: Table for User’s input

|  |  |
| --- | --- |
| **Use Case ID** | 1 |
| **Use Case** | User’s input |
| **Description** | The user inserts his/her personal details and symptoms. |
| **Actor** | User |
| **Trigger** | User clicks on “Enter details” and “Enter symptoms”. |
| **Primary Scenario** | User inputs his/her personal information and symptoms that are saved for later processing. |
| **Exceptional Scenario** | None |

1. Training Machine Learning Models:

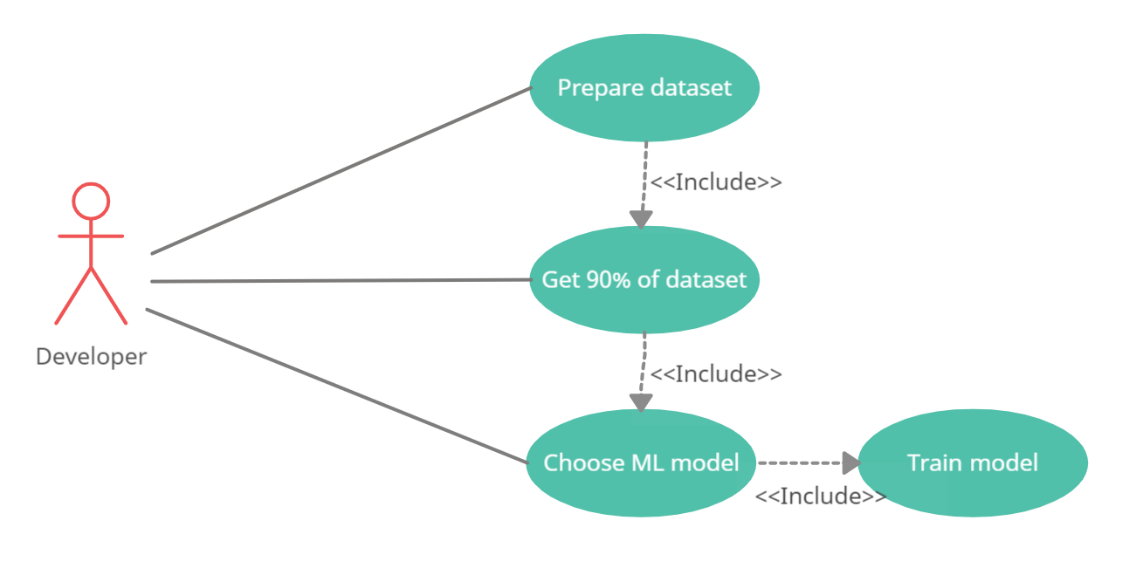


Figure 5 Use case for training ML models

Table 3: Table for training ML models

|  |  |
| --- | --- |
| **Use Case ID** | 2 |
| **Use Case** | Training machine learning models |
| **Description** | A set of models are trained on dataset. |
| **Actor** | Developer |
| **Trigger** | These models are included using different machine learning libraries and packages. |
| **Primary Scenario** | A set of classifiers and regressors get trained on 90% of data from dataset used in code. |
| **Exceptional Scenario** | None |

1. Performance Accuracy Computation:

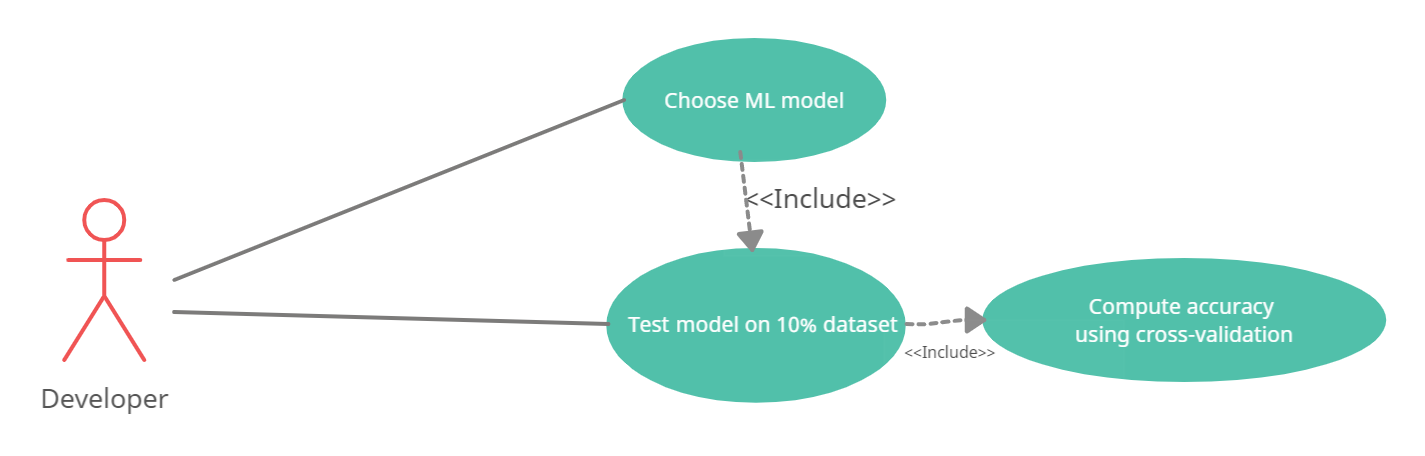


Figure 6 Use case for Performance Accuracy Computation

Table 4: Table for Performance Accuracy Computation

|  |  |
| --- | --- |
| **Use Case ID** | 3 |
| **Use Case** | Performance accuracy computation |
| **Description** | The performance accuracy of each machine learning model is computed. |
| **Actor** | Developer |
| **Trigger** | A performance computation metric is included within the code. |
| **Primary Scenario** | For each of the models, performance accuracy is computed using cross-validation folds. |
| **Exceptional Scenario** | None |

1. Preprocessing of User Symptoms:

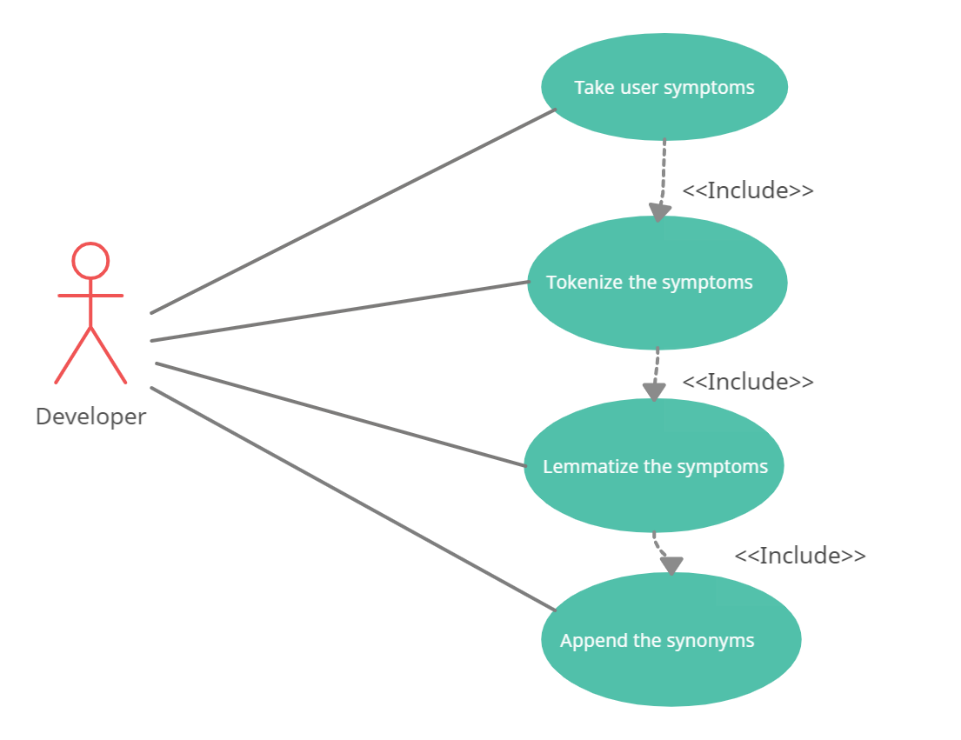


Figure 7 Use case for Preprocessing of user symptoms

Table 5: Table for processing of User Symptoms

|  |  |
| --- | --- |
| **Use Case ID** | 4 |
| **Use Case** | Preprocessing of user symptoms |
| **Description** | Symptoms entered by user are tokenized and lemmatize for further processing. |
| **Actor** | Developer |
| **Trigger** | Functions for tokenization and lemmatization are called within the code. |
| **Primary Scenario** | The user entered symptoms are preprocessed to produce a symptom vector for each user. |
| **Exceptional Scenario** | None |

1. Disease Prediction:

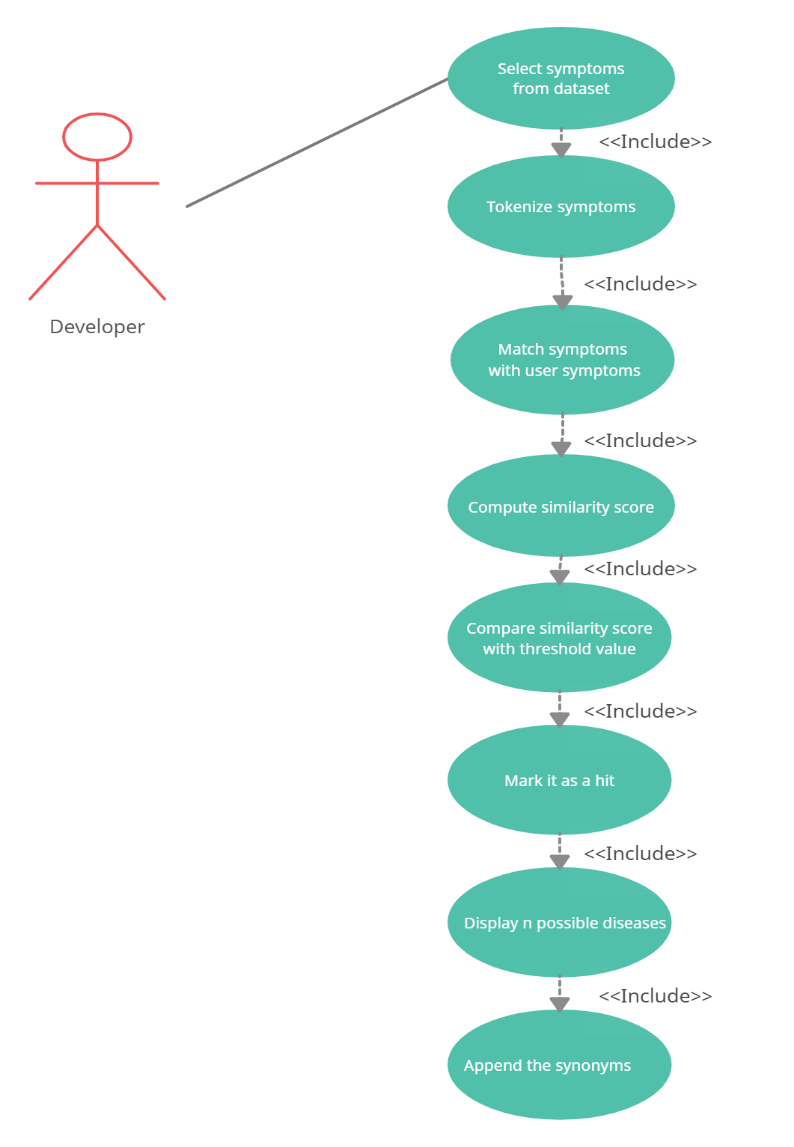


Figure 8 Use case for Disease Prediction

Table 6: Table for disease prediction

|  |  |
| --- | --- |
| **Use Case ID** | 5 |
| **Use Case** | Disease prediction |
| **Description** | A set of possible diseases are predicted based on user entered symptoms. |
| **Actor** | Developer |
| **Trigger** | A function for predicting diseases is called in the code. |
| **Primary Scenario** | The trained models predict a list of diseases based on user symptoms by matching and comparing. |
| **Exceptional Scenario** | None |

1. Suggesting Suitable Hospitals:

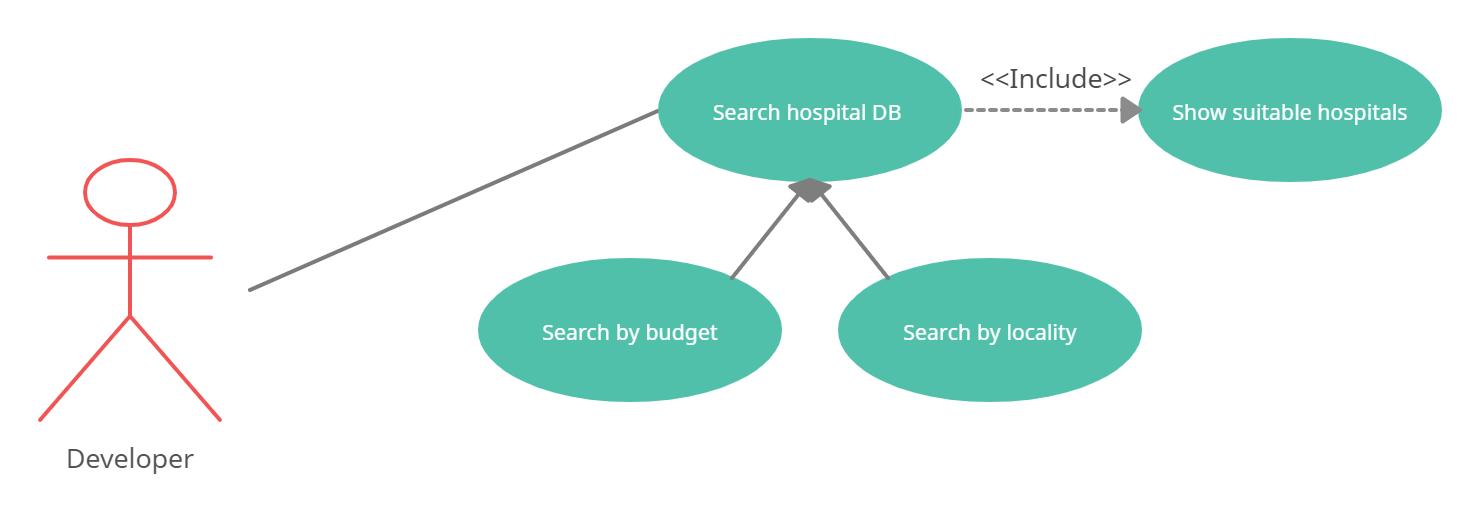


Figure 9 Use case for Suggesting Suitable Hospitals

Table 7: Table for suggesting suitable hospitals

|  |  |
| --- | --- |
| **Use Case ID** | 6 |
| **Use Case** | Suggesting suitable hospitals |
| **Description** | A list of all suitable hospitals is displayed. |
| **Actor** | Developer |
| **Trigger** | A query to search all hospitals based on user’s budget and locality is implemented in the code. |
| **Primary Scenario** | The hospital DB is searched on the basis of budget and locality and a list of suitable hospitals is displayed. |
| **Exceptional Scenario** | None |

1. Managing Databases:

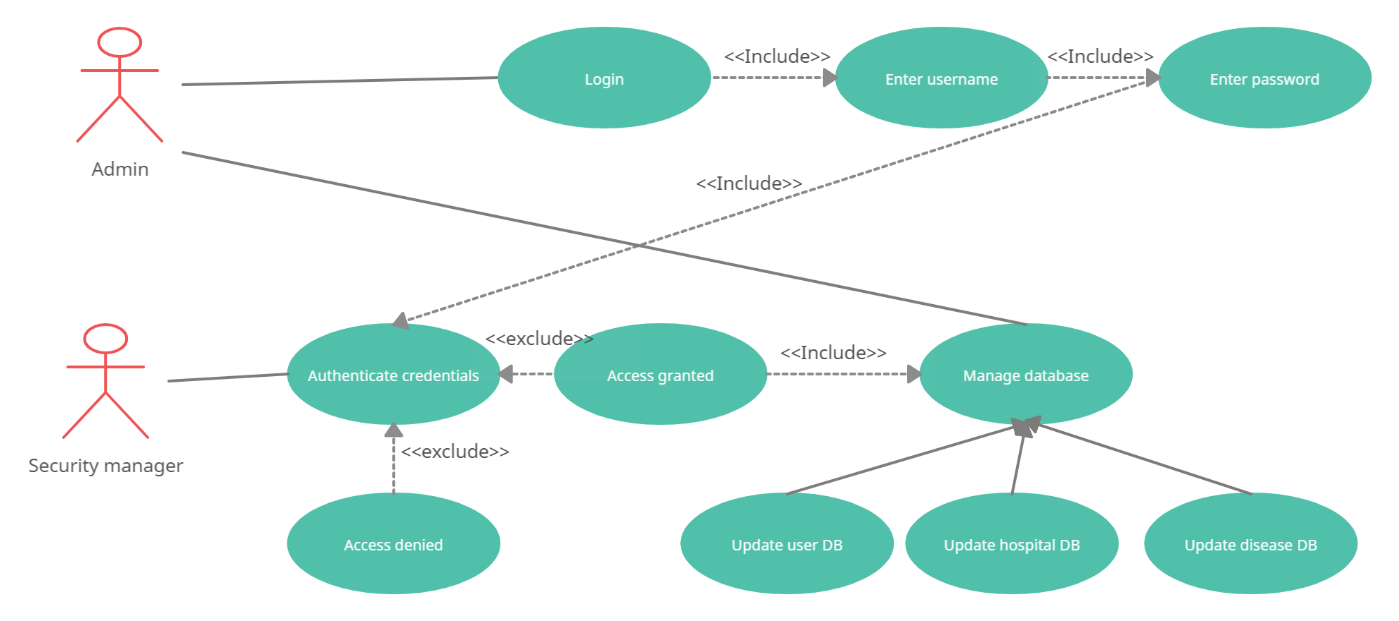


Figure 10 Use case for Managing Databases

Table 8: Table for managing databases

|  |  |
| --- | --- |
| **Use Case ID** | 7 |
| **Use Case** | Managing databases |
| **Description** | Admin manages all the databases implemented in the product. |
| **Actor** | Admin |
| **Trigger** | Admin can select three different interfaces to update all three databases. |
| **Primary Scenario** | Admin checks and updates databases. |
| **Exceptional Scenario** | None |

## Logical View:

Figure 11 Context Diagram

## Development View

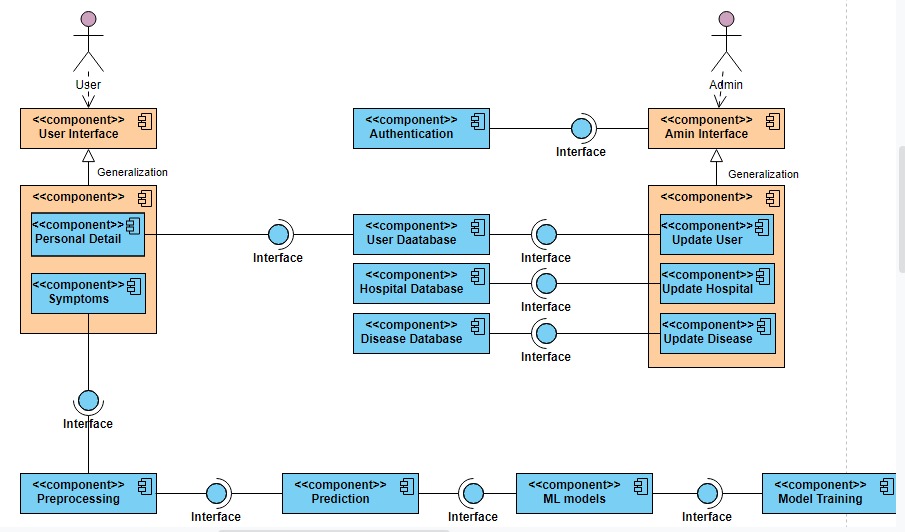


Figure 12 Component Diagram

## Process View

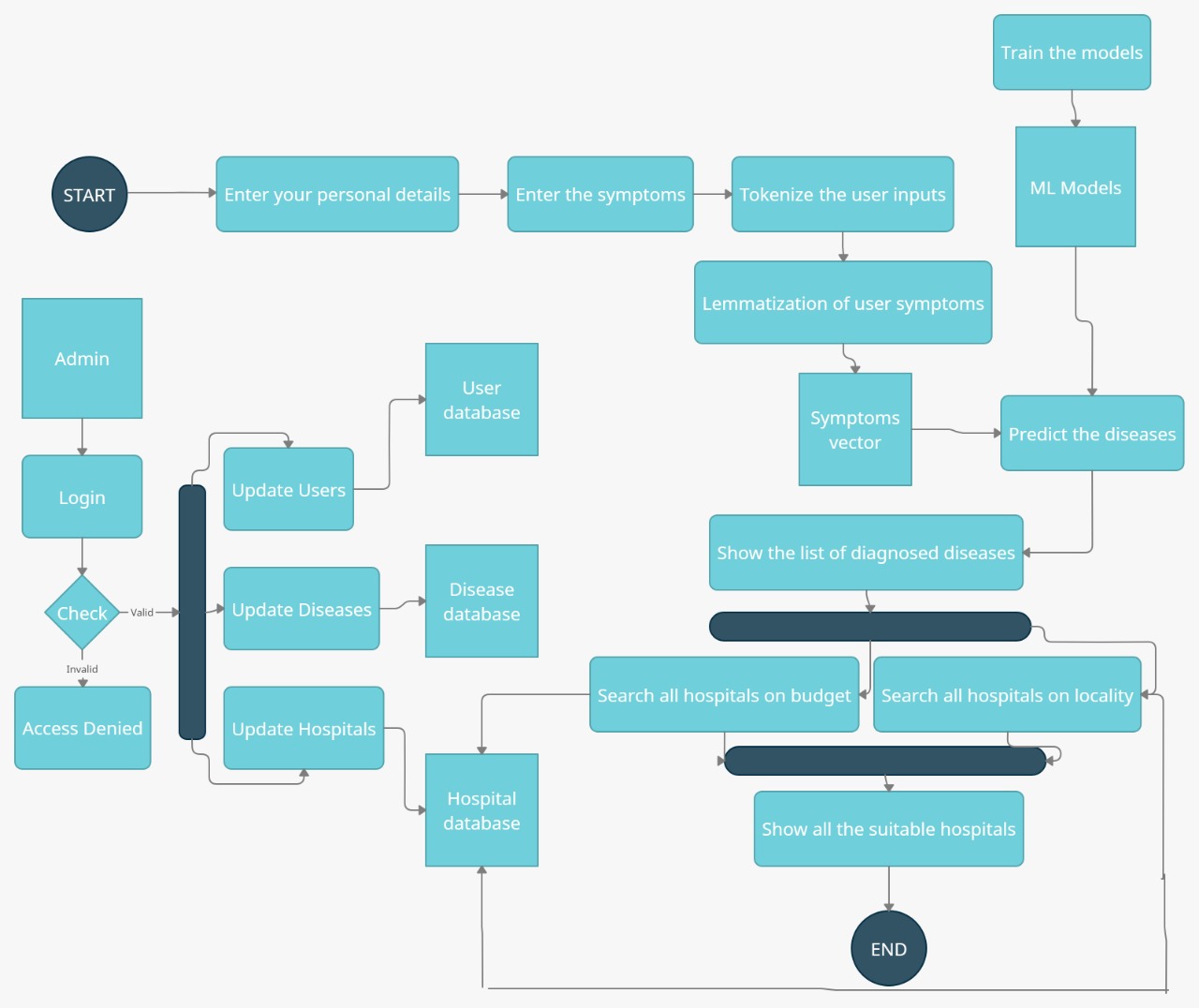


Figure 13 Activity Diagram

## Physical View

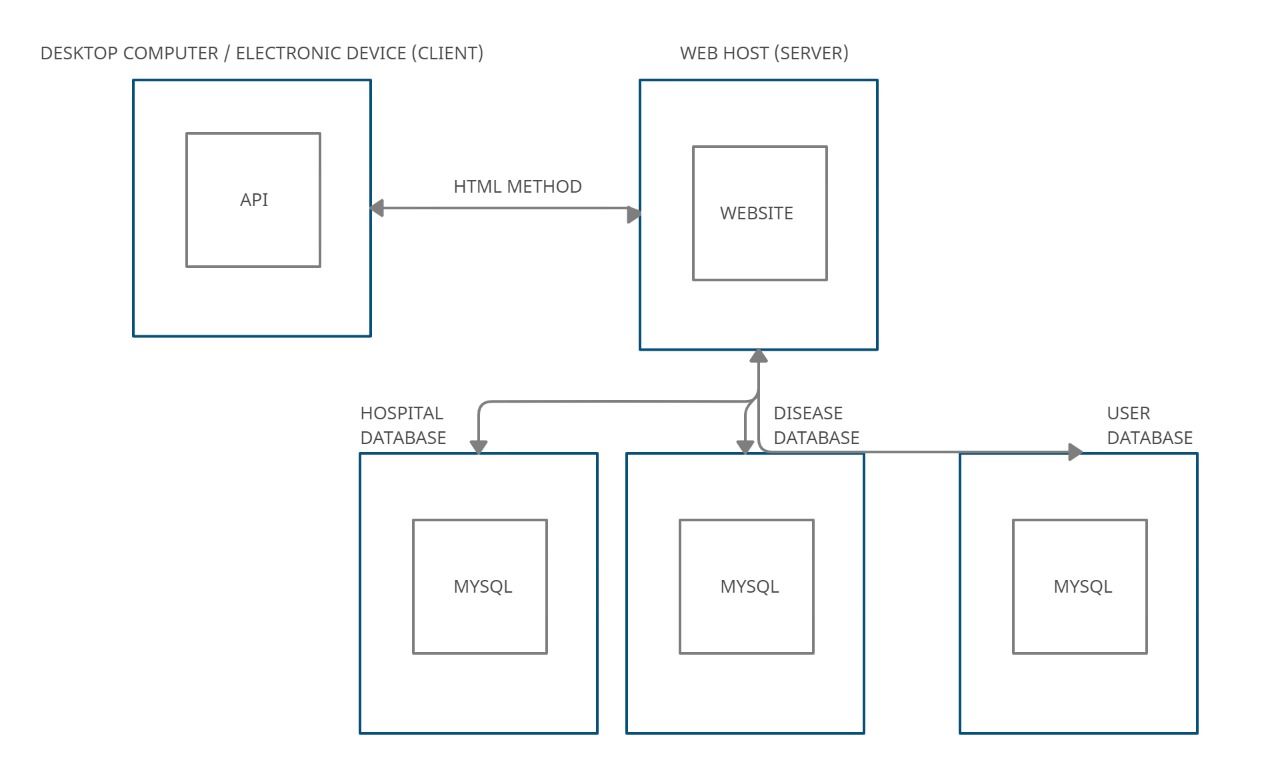


Figure 14 Physical Diagram

## User Interface Design

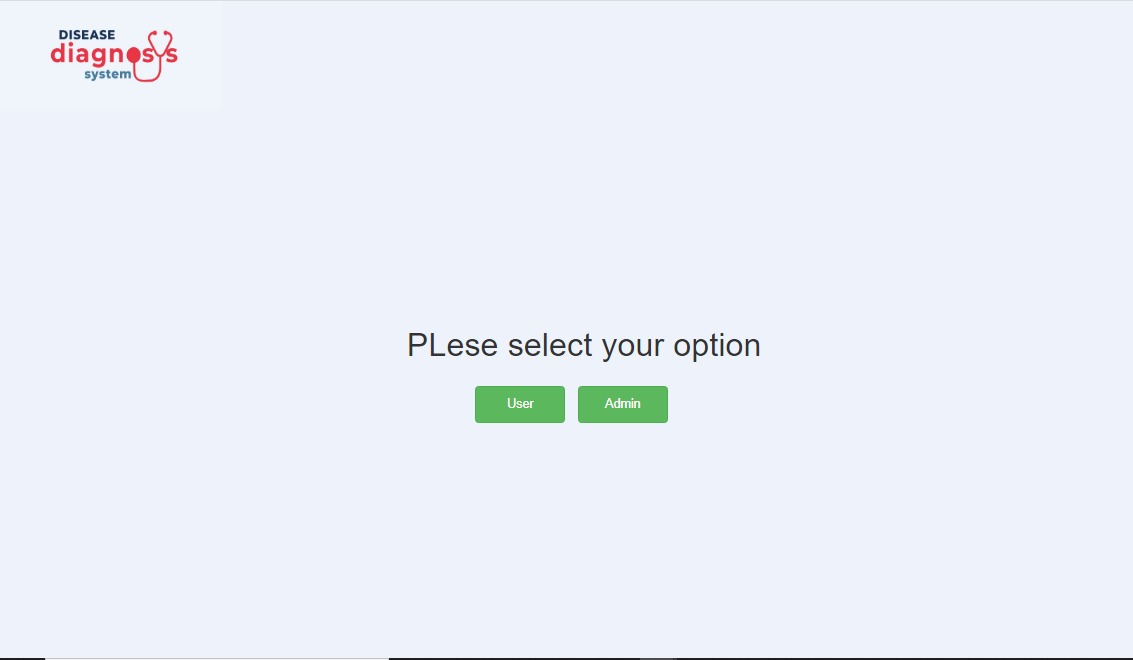


Figure 15

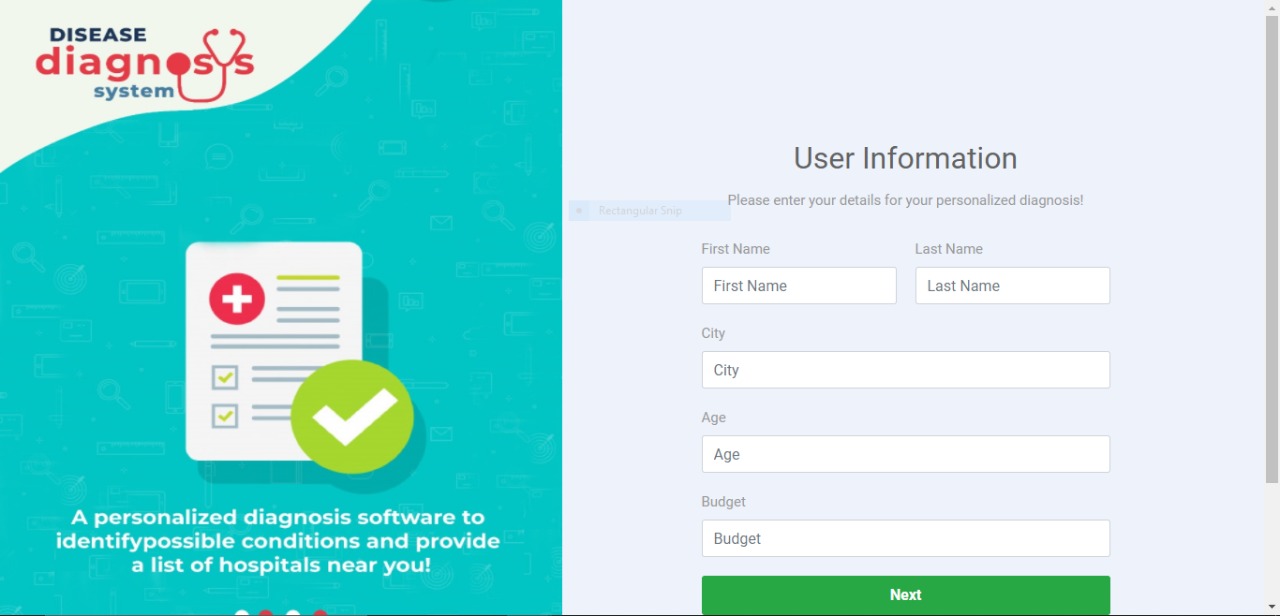


Figure 16



Figure 17

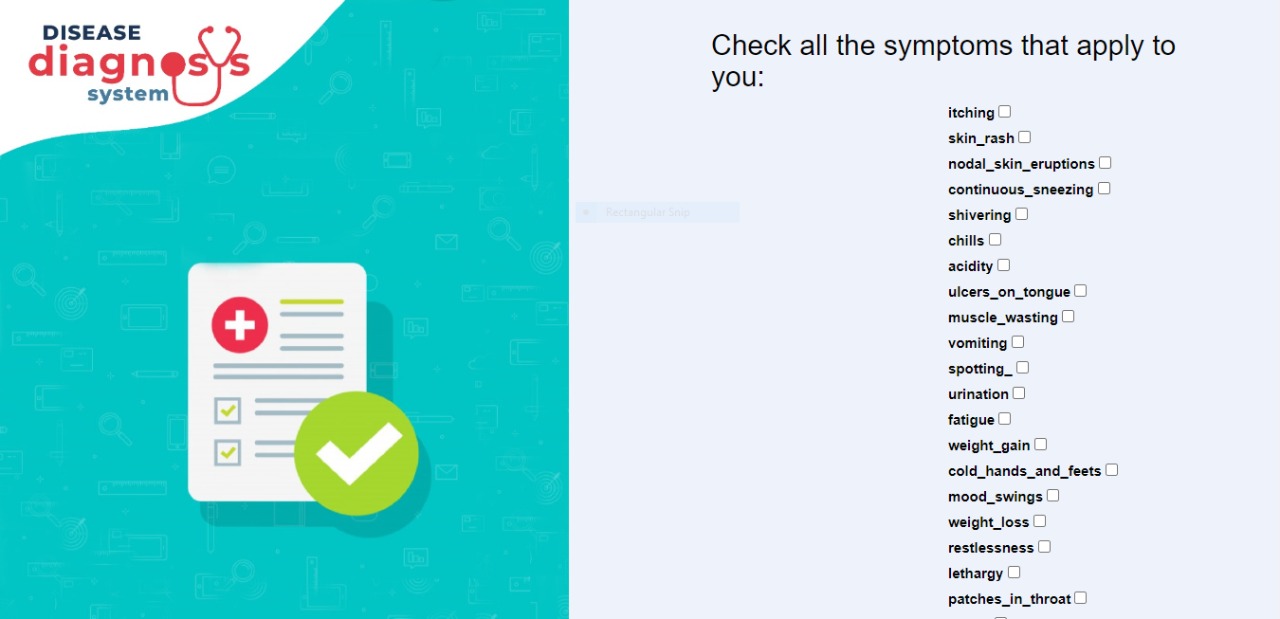


Figure 18

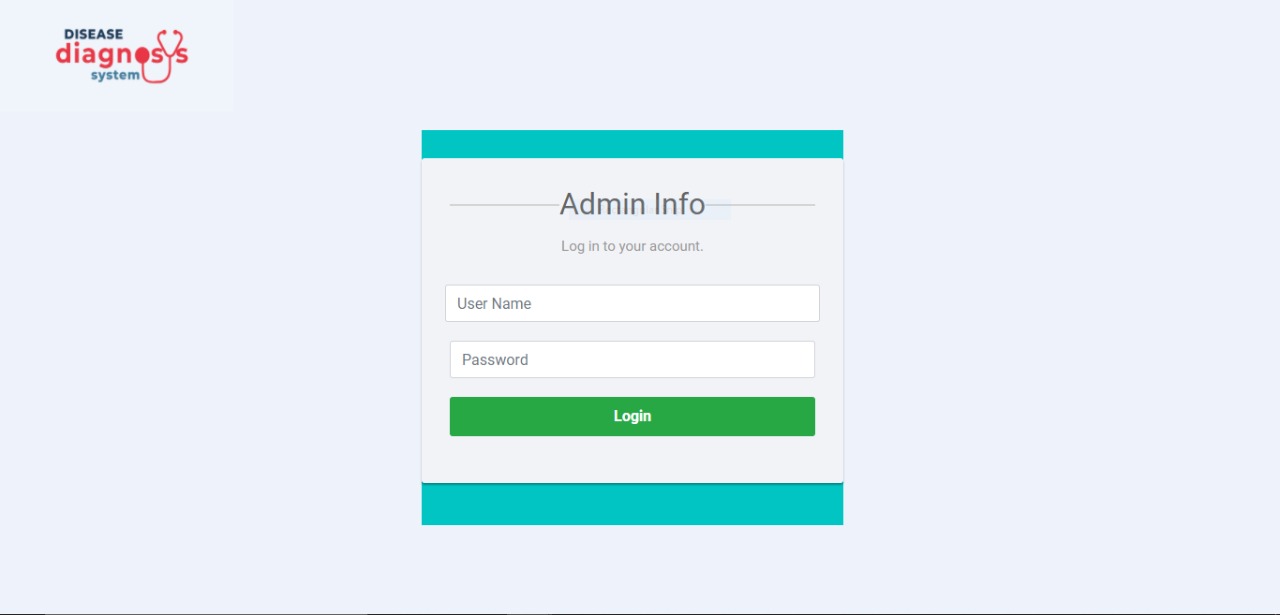


Figure 19

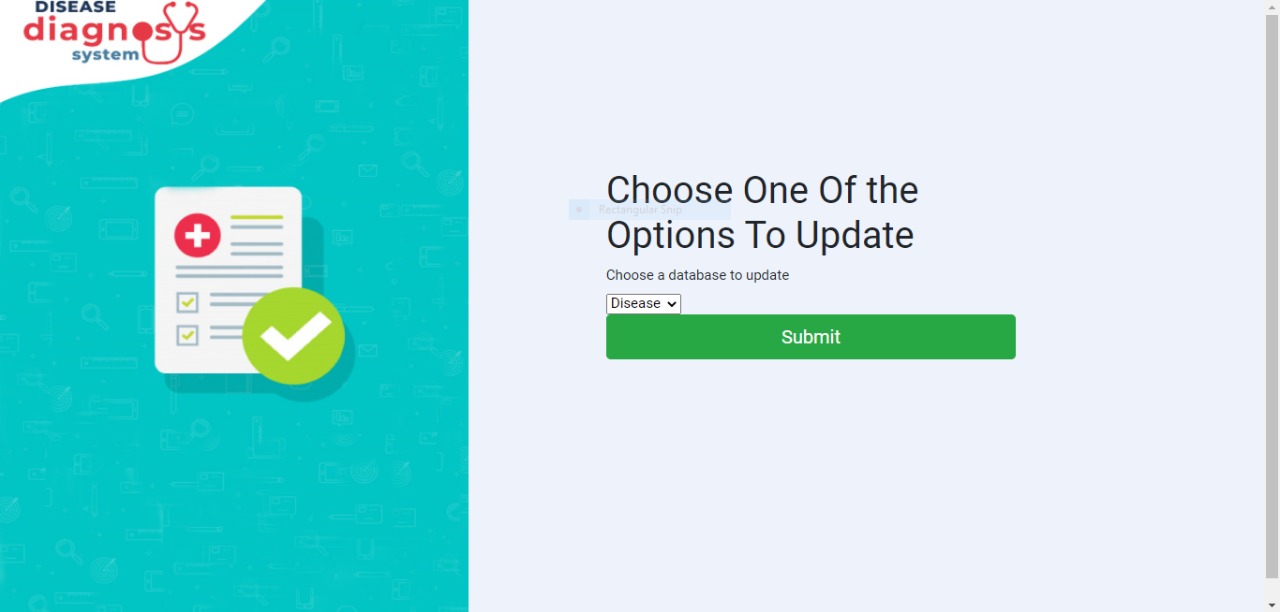


Figure 20