IT314 – Software Engineering

Group 25

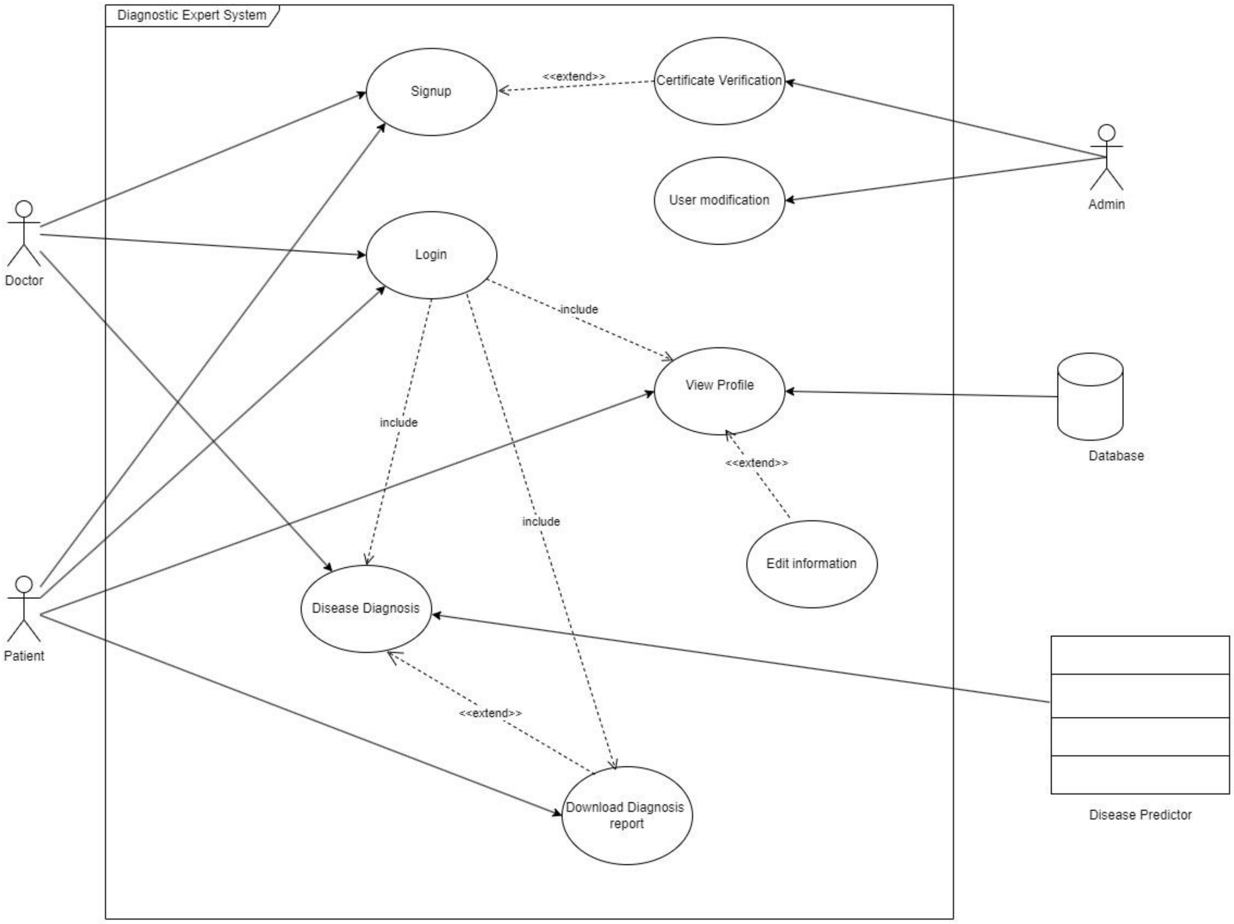
Diagnosis Expert System

Table of content

Introduction: Software requirement specification

Here is the documentation of the Diseases Diagnosis expert system. Below is the Use Case Diagram which depicts the components of the system, such as the actors like doctor, admin and patient with that the predictor model and database are also connected. This whole system shows the working and the basic features of the system. Following is its documentation, showcasing a detailed explanation. Furthermore the Functional and non-functional requirements are attached with the user stories.

Use case Diagram



Use case Documentation

1: Register as Patient

Name: Register as Patient

Goal: - A new user should be able to create his/her personal account.

Actor: - Patient, database

Precondition: - The user needs to open the website.

Trigger: User clicks on the ‘Sign up’ button from Homepage.

Main Flow: -

1. User needs to select the option to “register as patient”.
2. User submits the details such as – Name, DOB, email ID, contact number, gender.
3. When the user is registered, an email will be sent to the registered email ID with a unique patient ID.

Alternate Flow: -
In case of any kind of system failure at any step:
- The user will need to re-register.

2.a If email ID is not in proper format:
- The system should display a message indicating an incorrect email format.
2.b If the contact number is not written in proper format, then the system should display a message indicating incorrect number format.
3.a If mail is not received, it would imply that user is not yet registered and will need to re-register.

Postcondition: - The user is registered successfully and is logged in to his/her account and directed to the homepage of the website.

Special Requirements: - The system needs to be stable so that the registration process is completed without any difficulties.
- The mail should be received within 5 minutes so that the patient can use it.
- The patient information should be stored in the database quickly.

2: Register as Doctor

Name: Register as Doctor
Goal: - A new user should be able to create his/her personal account.

Actor: - Doctor, Database
Precondition: - The user needs to have URL of the website and should visit it.
Trigger: User clicks on the ‘Sign up’ button from Homepage.

Main Flow: -
1. User needs to select the doctor option to register as doctor.
2. User fills in the details such as – Name, DOB, email ID, contact number, gender, speciality, and scanned copy of Doctor Certificate.
3. When the user is registered, an email will be sent to the registered email ID with a unique doctor ID.

Alternate Flow: -
In case of any kind of system failure in between:
- The user will need to re-register.

2.a If email ID is not in proper format:
- The system should display a message indicating an incorrect email format.
2.b If the contact number is not written in proper format, then the system should display a message indicating incorrect number format.
3.a If mail is not received, it would imply that the user is not yet registered and will need to re-register.

Postcondition: - The user form is sent for verification and is redirected to the homepage of the website.
Special Requirements: - The system needs to be stable so that the registration process is completed without any difficulties.
- The doctor documents such as the Doctor's certificate should be verified manually within 24 hours.
– The mail should be received within 24 hours so that the doctor can use it.

- The doctor information should be stored in the database quickly once verified.

3: To verify the user from admin.

Name: To verify the user.

Goal: - To verify the doctor's account on the website.

Actor: - Admin

Precondition:

- The doctor have already filled the detailed asked.
- The admin is authenticated.
- The admin has opened the website and logged in.

Trigger: Admin clicks on the profile button.

Main Flow: -

1. The admin opens the list of the registered doctors.
2. The admin view the account details and verify it.
3. After the verification the account is ready to be used.
4. The admin signs out of the verification process.

Alternate Flow: -

- If a admin enters incorrect login details, the website displays an error message.
- On having technical failure or poor connection, the website will show an error message.
- If the user is not authentic the admin discard the request to create account.

Postcondition:

- Admin has successfully verified the details.
- The admin has the option to log in again in future to access the account details to verify them.

4: Login

Name: - Login
Goal: - A registered user should be able to login to his/her personal account.
Actor: - Doctor, Patient
Precondition: - The user needs to have the URL of the website and an existing account.
Trigger: User clicks on the Login button from Homepage.

Main Flow: -

1. User enters a unique user ID and password.
2. User logs in.

Alternate Flow: -
In case of any kind of system failure at any step:

- The user will need to login again.

1. If user ID is incorrect or password is incorrect, the user will be asked to re-enter valid credentials to log in or to sign-in (make a new account).

Postcondition: - The user is directed to the home page of the website.

5: Disease Diagnosis

Name: - Disease Diagnosis
Goal: - Doctor uses the ML model to predict a disease.
Actor: - Doctor, Disease Predictor
Precondition: - Doctor is logged in to his/her verified account.
Trigger: - Patient contacts doctor.
User: - Doctor

Main Flow: -

1. User enters patient ID.
2. User gives input of the relevant symptoms of the disease in the form.
3. The input details are used to predict the disease and the prediction is obtained.
4. The doctor writes his/her own conclusion and prescription based on the predictor and personal experience.
5. The diagnosis is updated in the database.
6. Patient Report is generated.

Alternate Flow: -

If system failure occurs:

- the system should reset back to normal without having any anomalies.

On the steps 1,2 & 3:

- the user should be able to go back to the previous step.

1. If patient ID is not found:

The patient is requested to make an account and the process is put to halt till the patient account is not created.

2. If incorrect input is given in any field, the user should be able to rectify the error before submitting it for disease prediction.

2.b The system should prompt the user when unexpected input is given.

6.a If a patient report is not generated, the user should be able to manually do this step.

Postcondition: - The system returns to the dashboard, ready for the next diagnosis.

6: To view profile

Name: To view profile

Goal: - To view a profile from the website.

Actor: - Patient/Doctor

Precondition:

- The Patient/Doctor has a valid account on the website.
- The Patient/Doctor has logged in to the account

Trigger: User clicks on the 'Name' button from Dashboard.

Main Flow: -

1. The Patient/Doctor opens the website and navigates to login.
2. The Patient/Doctor accesses his account by entering the correct username and password.
3. The website directs the Patient/Doctor to the dashboard.
4. The website process the request from the records(database)
5. The website display the details that they have filled while creating the account
6. Patient logs out of the account for security and privacy reasons.

Alternate Flow: -

- If a patient enters incorrect login details, the website displays an error message.
- On having technical failure or poor connection, the website will show an error message.
- If the patient's/Doctor's record is not verified by admin it shows the error message

Postcondition:

- Patient/Doctor has successfully viewed their profile.
- The Patient/Doctor has the option to log in again in future to access his records.

7: To download medical report online

Name: To download medical report online

Goal: - To download a medical report from the website.

Actor: - Patient and doctor

Precondition:

- The patient has a valid account on the website.

- The patient has already provided his medical data.
- The doctor has predicted his diseases and has given a conclusion.
- The doctor has a valid account on the website.
- The doctor should have predicted the diseases.

Trigger: User clicks on the ‘Download’ button from Dashboard.

Main Flow: -

1. The patient/doctor opens the website and navigates to login.
2. The patient/doctor accesses his account by entering the correct username and password.
3. The website directs the patient to the dashboard.
4. The website process the request from the records(database)
5. As the report is ready, it is displayed to the patient to save/download.
6. Patient/doctor logs out of the account for security and privacy reasons.

Alternate Flow: -

- If a patient/doctor enters incorrect login details, the website displays an error message.
- On having technical failure or poor connection, the website will show an error message.
- If the patient’s record is not verified or predicted by doctor then it shows the error message

Postcondition:

- Patient/doctor has successfully download the medical report.
- The patient/doctor has the option to log in again in future to access his records.

Functional and Non-Functional requirements

Functional Requirements for the Disease Diagnosis Website

1. Disease Identification: The website shall employ a machine-learning model to recognize and diagnose diseases based on user-provided symptoms and medical history.

2. Data Storage: The system should securely store user-provided medical data and diagnostic results in a structured and organized manner.

3. Health Report Generation: The website shall generate comprehensive health reports for users and treatment recommendations from doctors.

4. User Registration and Authentication: Users should be able to create accounts, log in securely, and manage their personal information.

5. Treatment Recommendations: The website should offer personalized treatment recommendations from doctor and guidelines based on the diagnosed disease.

Non-Functional Requirements for the Disease Diagnosis

1. Security: The system must ensure the confidentiality and integrity of user data, including adherence to privacy regulations and secure encryption of sensitive information.

2. Reliability: The website should be highly dependable, with minimal downtime.

3. Scalability: The system must be able to accommodate a growing number of users, ensuring that it remains efficient as the user base expands.

4. Performance: The website should deliver fast response times and accurate disease diagnoses ensuring a seamless user experience.

5. User Support and Documentation: Provide comprehensive user support, and query section to assist users in using the platform effectively.

6. System Maintenance: Ensure a systematic schedule for system updates, maintenance, and bug fixes to keep the website running smoothly and securely over time

User Stories

User Story #1

Front: As a Doctor, I want to see the percentage likelihood of each disease prediction so that I can assess the risk level.

Back: Acceptance Criteria

- The system should display the percentage likelihood of each disease prediction.
- The percentage likelihood should be calculated based on the user's symptoms and the algorithm's accuracy.
- The system should clearly label the percentage likelihood as an estimation and not a definitive diagnosis.
- The system should provide information on the accuracy of the algorithm used for predictions.

User Story #2

Front: As a doctor, I want to input my own diagnosis to compare with the system's predictions so that I can have a second opinion.

Back: Acceptance Criteria

- The system should allow doctors to input their own diagnosis for comparison.
- The system should highlight any discrepancies between the doctor's diagnosis and the system's predictions.
- The system should provide references and evidence supporting the system's predictions.

User Story #3

Front: As a doctor I want to login on the webpage so that I can use the features of the website.

Back: Acceptance Criteria

Success: Login

Failure:

- Wrong format
- Wrong email/contact/Password
- Network error
- Session time expired.

User Story #4

Front: As a doctor I want to create an account so that I can register myself on a website.

Back: Acceptance Criteria

- On successful verification create my account
- On unsuccessful registration it will show the errors such maybe, network error, service unavailable

User Story #5

Front: As a patient, I want to seek my previous prescriptions given by doctors at different points of time for my disease so that I can track my health and use it for other purposes.

Back: Acceptance Criteria

- The system can have a patient as a different type of user with limited view and operating access.

- The patient will have a unique patient ID using which he/she can login.
- The system will show the patient’s medical history i.e. which doctor was consulted, what prescription was given, at what time it was given etc.

User Story #6

Front: As a healthcare professional, I want to view certain patients' disease prediction history so that I can efficiently use the data for examination or research purposes.

Back: Acceptance Criteria

- The system should permit healthcare professionals to access certain patients’ disease prediction history while ensuring the anonymity of personal information.
- The history should display the date, entered symptoms, disease predictions and doctor’s conclusion for each entry.

User Story #7

Front: As a doctor I want to input data in appropriate units and datatypes so that I can get accurate results.

Back: Acceptance Criteria

- The system will show what datatype it expects in every field and the units in which input should be given.
- If a system is requested to give output using invalid data then it should raise an error.

User Story #8

Front: As a doctor, I want to set up alerts for certain disease predictions so that I can quickly respond to high-risk cases.

Back: Acceptance Criteria

- The system should allow doctors to set up alerts for specific disease predictions.
- The system should alert doctors for high risk disease for a likelihood more than 90%.

User Story #9

Front: As a pharmaceutical company, I want to access aggregated data on disease trends so that I can anticipate the demand for certain drugs.

Back: Acceptance Criteria

- The system should provide aggregated data on disease trends.
- The data should be anonymized and de-identified to protect patient privacy.
- The system should provide visualizations (e.g., graphs, charts) to help analyze the disease trends.

User Story #10

Front: As a relatively young doctor I want to examine my patient's report so that I can classify between near-classified diseases using global data.

Back: Acceptance Criteria.

- Viewers of this category will be considered while making the model.