

3. User Characteristics

There are 3 types of roles in this software.

1. **Admin:** The administration is the main owner of this Medi-Care software. One must understand computers and the internet as well as prior knowledge of operating this software. The administration is responsible for maintaining the software and its requirements. The administration can perform the following functions:
 - Add or change the price of the cost, fees, cabin no., doctor/hospital/ diogenitic canter name, test name, etc. And can delete those if needed.
2. **Patients:** The patient is the main user of this software. They must have the minimum knowledge of the internet. The users/patients should be able to perform the following functions:
 - View, browse, and select a category on the home page.
 - View, add, and update min/max cost, fees, and other requirements.
 - Create a profile, and be able to update it whenever users want to.
 - Check out the list of Hospitals/Doctors/Diagnostic Centres and chose whatever users want.
 - Can Sing-in/log in using username and password.
 - Can comment and review listed hospitals/doctors/Diagnostic Centres.
3. **Hospital/Diogenitic canter IT Staff:** This role is basically from the listed hospital, doctor chamber, and diagnostic centres, Staff must understand computers and the internet as well as prior knowledge of operating this software. This role should be able to perform the following functions:
 - View the patient's appointment and confirm it and listed it in their database.
 - View and reply to Users'/Patients' comments.
 - Give basic information to patients.
 - Can inform to change their requirement to the administration.

4. Functional Requirements

This section contains the requirements for Medi-Care software. The functional requirements, as collected from the users, have been categorized as follows to support the types of user interactions that the system shall have.

1. Patients: View Categories and Search:

The users/patients can see the home page of the online Medi-Care application when they first visit the program. The users will be able to view the different categories e.g. Users can visit the Doctor list by searching doctor's names. And user can understand our application by reading features documents. Users can select categories for searching. And it will show the result of the search.

Functional Requirement:

- 1.1 Users/Patients can view categories on the home page.
- 1.2 Can be able to view different categories.
- 1.3 Select one of the categories.
- 1.4 Can be able to search the hospital/doctor/diagnostic center's name by selecting categories.
- 1.5 View the search items in the list.

Priority Level: High

Precondition: System database must have a List of hospitals/Doctors/ Diagnostic center's names.

Cross-Reference: None.

2 Patients: Select Living Area:

Before selecting the living area user must select the test name he/she chose Diagnostic Center, or select and fill up the current health condition if he/she selects Hospital/Doctor and if he/she knows the disease select the disease. If he/she fill-up current health condition then will show up as a possible disease and can select a living area. In selecting the test, the living area will show directly. Then fill up the user's living area details.

Functional Requirement:

- 2.1 User will write his /her disease name
- 2.2 Application will suggest a disease name whether the user types the disease name.
- 2.3 It is not mandatory to write the disease name if user does not know his current disease.
- 2.4 Then it allows users to select their area.
- 2.5 All post-office names and local area names are displayed when the user clicks the drop-down menu named "Select Your Living Area".

Priority Level: High.

Precondition: System database must have all area names and disease names.

Cross-Reference: None.

3 Patients: Hospital Comparison:

In this section, Medi-Care will provide a suitable hospital for the user. But the users can search hospitals by completing his/her living area. Then the application will provide a list of the hospital to the user(patient) and also shows the best suitable hospital based on the user's filter's cost, distance, cabin fees, etc. there will have a list of how many patients are cured from that particular hospital. There also have reviews and comments from previous patients. So, the user can understand which will be the best.

Functional Requirements:

3.1 Application can provide suitable hospitals based on the user's health condition.

3.2 There will be a search option where users can search hospitals. And able to search it based on cost, distance, medicine fees, cabin fees, and medical test costs.

3.3 Then it will provide a list of the hospitals from the users required.

3.4 A list will be shown to the user where the number of patients who were cured.

3.5 There also have a review and comment portion where patients account for rating hospitals.

Priority Level: High.

Precondition: No prediction.

Cross-Reference: None.

4 Patients: Select Hospital:

In this section, the user can select his/her required hospital to understand which will be good for him/her.

Functional Requirements:

4.1 User can select a hospital.

Priority Level: High

Precondition: Result must be shown.

Cross-Reference: None.

5 Patients: Select Cabin and Confirm:

A list of cabins of that hospital can able to see users. So, they understand how many cabins are available, and the cost of the cabin also show to the users. Users also see the categories of cabins. Then users can select the required and confirm the cabin booking.

There is also the color of the cabin number on site. Greed means the cabin is available and Red means not.

Functional Requirements:

- 5.1 Cabin list will show the users with its categories and price.
- 5.2 Users can book a selected cabin.
- 5.3 There is also have to color in the cabin number where Green is available and Red is not.

Priority Level: High.

Precondition: System must have information about the cabin and save the confirmation of booking.

Cross-Reference: None.

6 Patients: Doctor Comparison:

In this section, Medi-Care will provide suitable Doctors for the user. But the users can search for Doctors by completing his/her living area. Then the application will provide a list of the doctor to the user and also shows the best suitable doctor based on the user's filter's cost, distance, appointment fees, etc. There also have reviews and comments from previous patients. So, the user can understand which will be the best.

Functional Requirements:

- 6.1 Provide suitable doctors based on the user's health condition or disease.
- 6.2 There will be a search option where users can search for Doctors. And able to search it based on cost, distance, and appointment fees.
- 6.3 Then will provide a list of the doctors from the users required.
- 6.4 There also have a review and comment portion where patients account for rating hospitals.

Priority Level: High.

Precondition: No prediction.

Cross-Reference: None.

7 Patients: Select Doctor:

In this section, the user can select his/her required doctor to understand which will be good for him/her.

Functional Requirements:

- 7.1 User can select a Doctor.

Priority Level: High

Precondition: Result must be shown.

Cross-Reference: None.

8 Patients: Diagnostic Center Comparison:

Users can search for Diagnostic Center by completing his/her living area. Then the application will provide a list of the Diagnostic Center to the user and also shows the best suitable Diagnostic Center based on the user's filter's cost, distance, etc. There also have reviews and comments from previous patients. So, the user can understand which will be the best.

Functional Requirements:

- 8.1 There will be a search option where users can search Diagnostic Center. And able to search it based on cost, distance, and fees.
- 8.2 Then it will provide a list of the Diagnostic Center from the users required.
- 8.3 There also have a review and comment portion where patients account for rating hospitals.

Priority Level: High.

Precondition: No prediction.

Cross-Reference: None.

9 Patients: Select Diagnostic Center:

In this section, the user can select his/her required Diagnostic Center to understand which will be good for him/her.

Functional Requirements:

- 9.1 User can select a Diagnostic Center.

Priority Level: High

Precondition: Result must be shown.

Cross-Reference: None.

10 Patients: Information Given and Confirmation:

The patient must give his information after confirming Booking Cabin or Doctor Appointments or Diagnostic Center. Users can create an account for avoiding further filling up the information. In giving information part user must have to fill-up Name, Gender, Blood Group, and NID, User can payment with a credit card or mobile banking system.

Functional Requirements:

- 10.1 Patient has to give his/her Name, Gender, NID, and Blood Group.
- 10.2 Can create an account.
- 10.3 Users can pay with a credit card or mobile banking.

Priority Level: High

Precondition: Result must be shown.

Cross-Reference: None.

11 Admin: View, Update, Delete, and Response:

The administration has a big role in this application. Here admin can update, view, delete or add any Hospital/ Doctor/Diagnostic center or add cost, appointment time, etc. to the web page. And able to respond to both patients' and IT staff's complaints or other information via chat systems.

Functional Requirements:

11.1 Can update, view, delete and add hospital/doctor/diagnostic center list.

11.2 Can update, view, delete and add hospital/doctor/diagnostic center costs and others things.

11.3 Able to respond to both patient and IT staff messages via the chat system.

Priority Level: High.

Precondition: System database to restore account and categories' information.

Cross-Reference: None.

12 IT Staff: View, Reply and Confirm:

IT staff main from that particular listed Hospital, Doctor, and Diagnostic Center. Staff can view patient information, his/her request, booking, or appointment in their database system. And also, able to reply to any information via the chat system.

Functional Requirements:

12.1 View patients' information and add it to their database.

12.2 Reply to any information via the chat system.

Priority Level: High.

Precondition: System database has account information.

Cross-Reference: None.

13 Login/ User Authentication:

Selected those 3-types of users who can log in to this platform. The Patient can create an account.

Functional Requirement:

13.1 The users can log in or register using the user authentication form.

13.2 The users cannot log in or register if the information is incomplete or invalid.

13.3 The patient can create an account.

Priority Level: High.

Precondition: System database to restore account information.

Cross-Reference: None.

5. Non-Functional Requirements:

- **Performance:** The website's load time should not be more than one second for users.
- **Reliability:** Applicants can access their search results 98% of the time without failure.
- **Availability:** Users can search for anything from those categories every week at any time during the day. So, users can get huge benefits from using this application.
- **Maintainability:** If any problem faces from using this application, we can solve it in a few hours.
- **Recoverability:** If a major incident happens on the website, the business must take measures to go back to being fully operational within three days.
- **Capacity:** Up to 100 users can request at a time in the current state. And store 100,000 users' data can store.
- **Serviceability:** Users can request any problem and able to get any solution they needed.
- **Security:** Only the users with the role "admin" can view the patient's request, their profiles, and their verified phone number. And users in the role of "patient" can log in with is username or phone number. If staff can log in with the given account.
- **Data integrity:** The system shall maintain data integrity by keeping backups of all updates to the database for every record transaction.
- **Interoperability:** The website must follow a service-oriented architecture.
- **Usability:** The website's interface has to be user-friendly and easy to use.

6. Diagram

7. UX Design