**Document Status: Approved**

**Approved by: Management Representative**

**Released by : Mentor Labs**

**Revision / Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver.** | **Date** | **Changed by** | **Modifications** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**List of Abbreviations**

DFD **D**ata **F**low **D**iagram

E-R **E**ntity **R**elationship

RS **R**equirement **S**pecification

**Table of Contents**

**1.** **Introduction 3**

**1.1** **Purpose 3**

**1.2** **Scope 3**

**1.3** **Definitions, Acronyms and Abbreviations 3**

**1.4** **References 3**

**1.5** **Coordinators 3**

**2.** **Software Requirements 3**

**2.1** **Operational Requirements 3**

**2.2** **Functional Requirements 4**

**2.3** **User Interface Requirements 7**

**2.4** **External Interface Requirements 18**

**3.** **Project Execution Related Requirements 18**

**3.1** **Development Environment 18**

Technologies to be used for development of the Application: **18**

**3.2** **Design requirements 18**

**3.3** **Coding requirements 18**

**3.4** **Error handling requirements 18**

**3.5** **Resource Requirements 18**

Technologies to be used for development of the Application: **18**

**3.6** **Risk management 18**

**3.7** **Training Requirements 18**

**4.** **Testing Requirements 18**

**5.** **Quality Assurance Activities 19**

**6.** **Software Acceptance Criteria 19**

**7.** **Deliverables 19**

**7.1** **List of Deliverables 19**

**7.2** **Delivery 19**

**8.** **Requirements Acceptance Statement 19**

# 1.Introduction

## 1.1 Purpose

The ***Provider Finder & Network Lookup*** project is designed to enable insurance members and users to efficiently search for healthcare providers—such as doctors and hospitals—who are within their insurance networks. The goal is to minimize out-of-network visits that can result in unexpected high costs, and to provide clarity and transparency regarding insurance coverage for medical services. The system empowers users to locate in-network providers based on specialty, location, and insurance plan, ensuring informed decision-making for medical care

## 1.2 Scope

The scope of this project involves the development of a comprehensive system for the efficient search, verification, and management of healthcare providers within an insurance network, including the following functional areas:

* **Search Functionality**: Users can search and filter healthcare providers by specialty, geographic location, and accepted insurance plans. Advanced filters such as provider ratings, distance, and availability will be supported, with results displayed in both list and interactive map views.
* **Provider Profiles:** Structured and detailed profiles for doctors, hospitals, and clinics including specialties, qualifications, experience, ratings/reviews, offered services, and accepted insurance plans.
* **Map Integration:** Integration with mapping APIs to display search results geographically, support location-based filtering, and provide directions to provider locations.
* **Admin and Provider Management:** Tools for administrators to add, update, validate, and remove provider details, manage provider affiliations, and maintain network accuracy.
* **Insurance Team :** Insurance Team Manages the Customer and Insurance Plans which involves to add,update,validate and remove the Customer and insurance Plans details.

## 1.3 Definitions, Acronyms and Abbreviations

**Provider Finder and Network Lookup:**  
A system that allows customers to search, find, and review healthcare providers (such as hospitals and doctors), confirm their network status, and view available doctors,hospitals and insurance plans.

**Provider:**  
An individual healthcare professional (doctor, specialist) or hospital offering medical services registered in the system.

**Customer:**  
An individual logged on the platform looking for healthcare providers or hospitals within a network and insurance plan.

**Admin:**  
A user with privileges to manage the entire system, including providers, doctors.

**Insurance Team**:  
A dedicated role or team responsible for registering and verifies customers with insurance-related plans

**Insurance Plans**:  
Different insurance products/plans listed in the system.

## 1.4 References

*UHC*

## 1.5 Coordinators

1. Roopesh

# 2 Software Requirements

## 2.1 Operational Requirements

**2.1.2 Customer**

* Login and fetch the Current Location of Customer
* Filter the Providers Locations
* Navigations for Proivders Locations
* Filtering Providers and Doctors details
* View the Providers and Doctors Profiles
* Providing Rating and Reviews

**2.1.2 Provider**

* View the connected Doctors Status
* Update their Availability Status
* View the Ratings and Reviews for thier profile
* Review Providers Performance based on the Ratings

**2.1.3 Admin**

* Login and Viewing Providers and Doctors
* Managing Providers details
* Managing Doctor details

**2.1.4 Insurance Team**

* Login and Viewing Customer and Insurance Plans
* Managing Customer details
* Managing Insurance Plans details

## 2.2 Functional Requirements

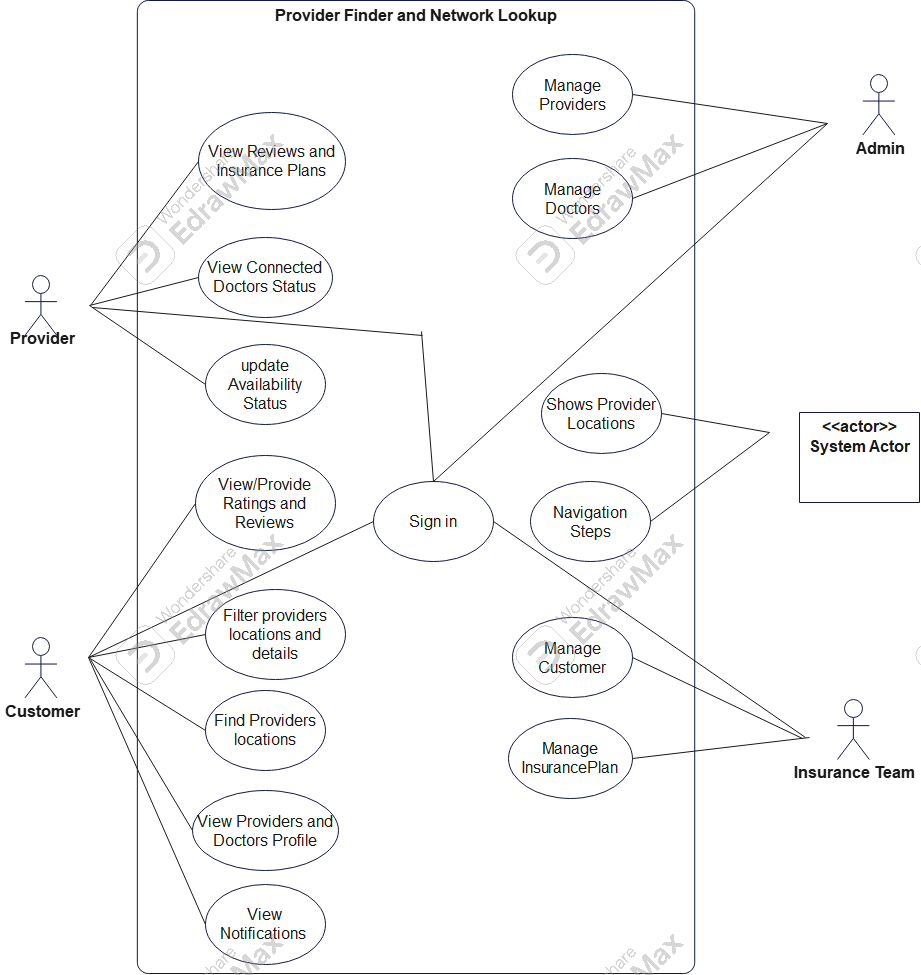


FIGURE 2.1 : Use Case Diagram For Provider Finder and Network Lookup

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-1* | |
| *Source/Trigger* | Customer | |
| *Description* | *Input* | Customer email,password and Insurance details |
| *Process* | Verify Credentials |
| *Output* | Customer dashboard upon successfull login |
| *Validation Method* | *Data validation (email format, password strength, required fields)* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-2* | |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Providers Name* |
| *Process* | *Verify the provider name* |
| *Output* | *Displays the Providers* |
| *Validation Method* | *validates the providers name* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-3* | |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Apply filters on provider* |
| *Process* | *Filter the Providers* |
| *Output* | *Displays the list of providers location based on filters* |
| *Validation Method* | *verifies filters on Providers* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-4* | |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Apply filters on Doctors* |
| *Process* | *Filters the Doctors details* |
| *Output* | *Displays the list of Doctor details* |
| *Validation Method* | *Verifies Filters on Doctors* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-5* | |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Rating and Reviews* |
| *Process* | *Providing rating and reviews for providers and doctors* |
| *Output* | *Updates the Providers and Doctors Rating* |
| *Validation Method* | *Verify and Validates the reviews to be alphanumeric* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-1-6* | |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Provider Details* |
| *Process* | *Views the notification* |
| *Output* | *Finds the location of Providers* |
| *Validation Method* | *Validates to get notification for particular customer* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-2-1* | |
| *Source/Trigger* | *Provider* | |
| *Description* | *Input* | *Provider email and password* |
| *Process* | *Verify Credentials* |
| *Output* | *Displays the Provider dashboard* |
| *Validation Method* | *Validates the credentials* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-2-2* | |
| *Source/Trigger* | *Provider* | |
| *Description* | *Input* | *Availability Status* |
| *Process* | *Updates the status* |
| *Output* | *Updates the Provider availability Status* |
| *Validation Method* | *Validates the status to Available or unAvailable* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-2-3* | |
| *Source/Trigger* | *Provider* | |
| *Description* | *Input* | *Rating and insurance plans* |
| *Process* | *Shows the reviews and plans* |
| *Output* | *Displays the reviews and insurance plans of provider* |
| *Validation Method* | *Validates reviews and ratings to be alphanumeric* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-3-1* | |
| *Source/Trigger* | *Insurance Team* | |
| *Description* | *Input* | *Customer details* |
| *Process* | *Perform CRUD operations on Customer* |
| *Output* | *Manages the Customer Details* |
| *Validation Method* | *Data validation (email format, password strength, required fields);* | |

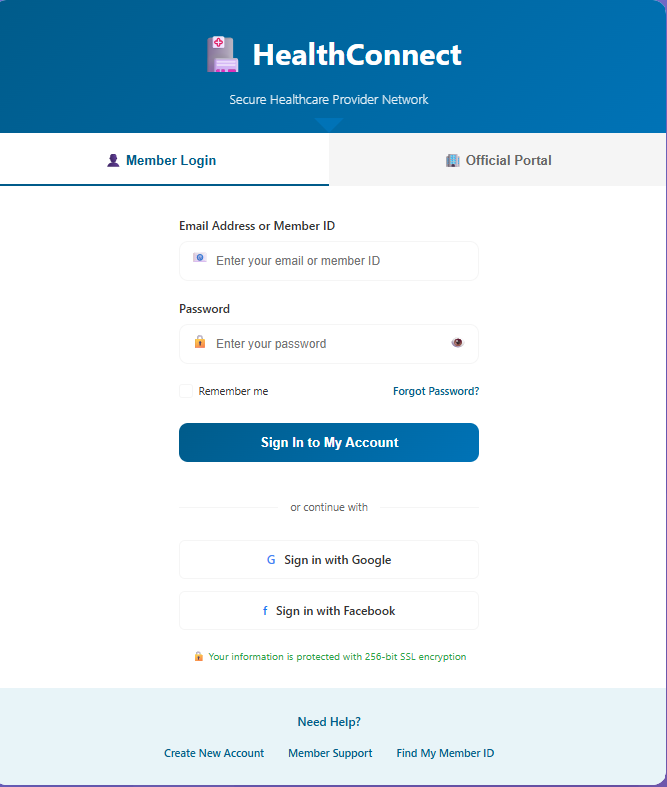
|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-3-2* | |
| *Source/Trigger* | *Insurance Team* | |
| *Description* | *Input* | *Insurance Plans* |
| *Process* | *Perform CRUD on Insurance plans* |
| *Output* | *Manages the Insurance Plans details* |
| *Validation Method* | *Data validation (email format, password strength, required fields);* | |

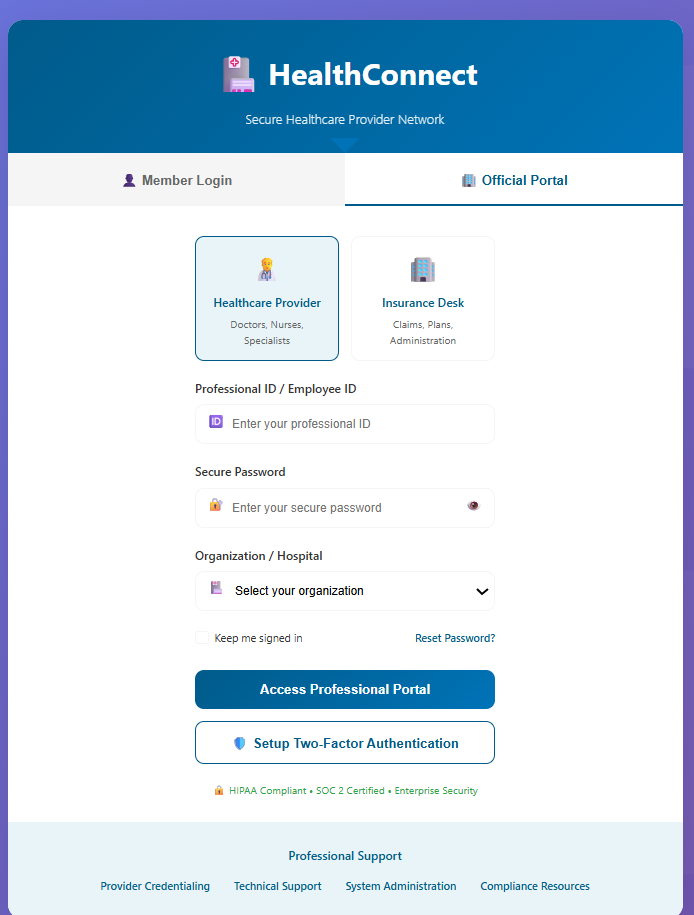
|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-4-1* | |
| *Source/Trigger* | *Admin* | |
| *Description* | *Input* | *Provider Details* |
| *Process* | *Perform Crud on Providers* |
| *Output* | *Manages the Provider details* |
| *Validation Method* | *Data validation (email format, password strength, required fields);* | |

|  |  |  |
| --- | --- | --- |
| Requirement Tag ID | *RS-4-2* | |
| *Source/Trigger* | *Admin* | |
| *Description* | *Input* | *Doctor Details* |
| *Process* | *Perform Crud on Doctors* |
| *Output* | *Manages the Doctor Details* |
| *Validation Method* | *Data validation (email format, password strength, required fields);* | |

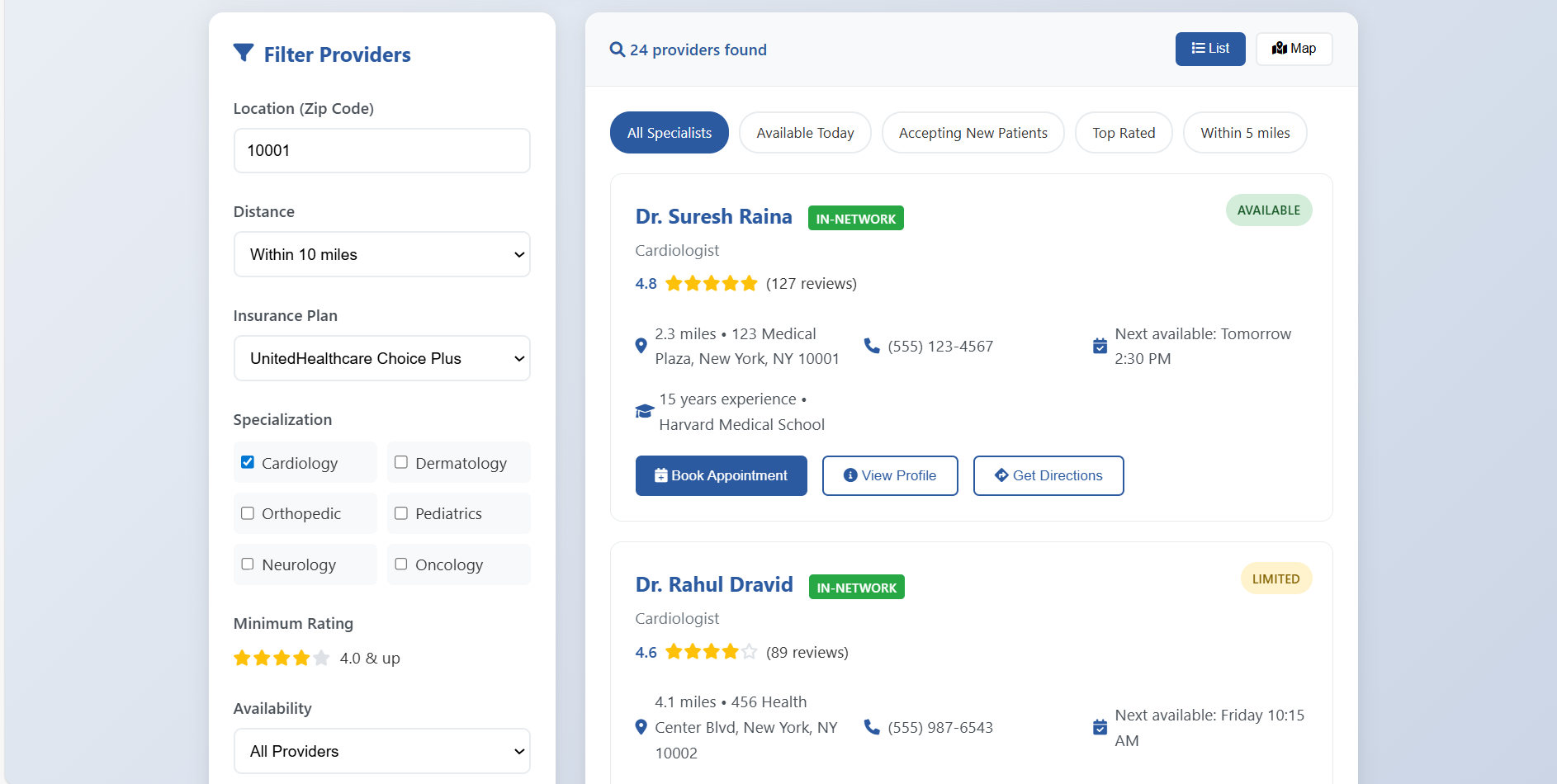
## 2.3 User Interface Requirements

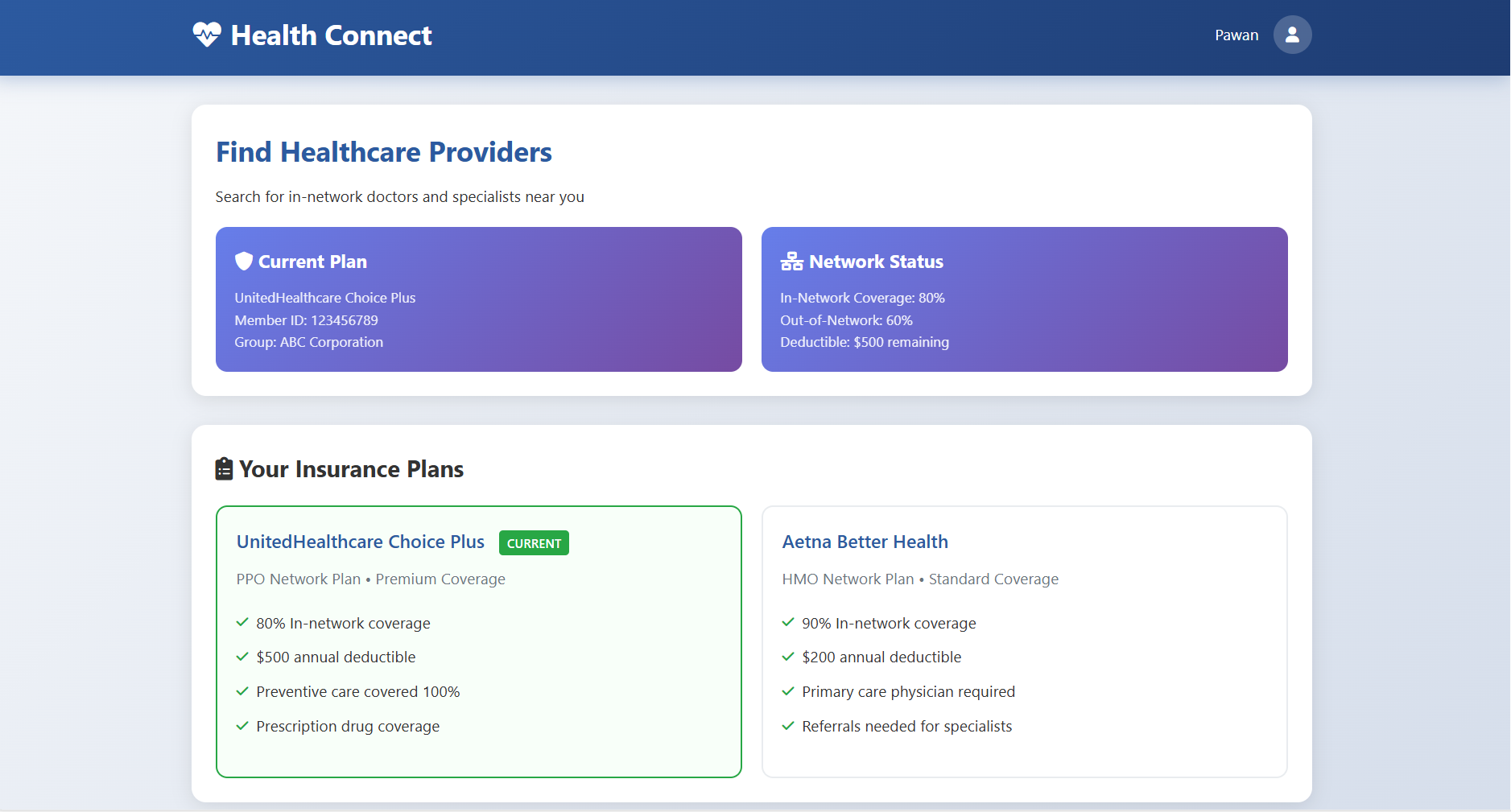
Login Pages :

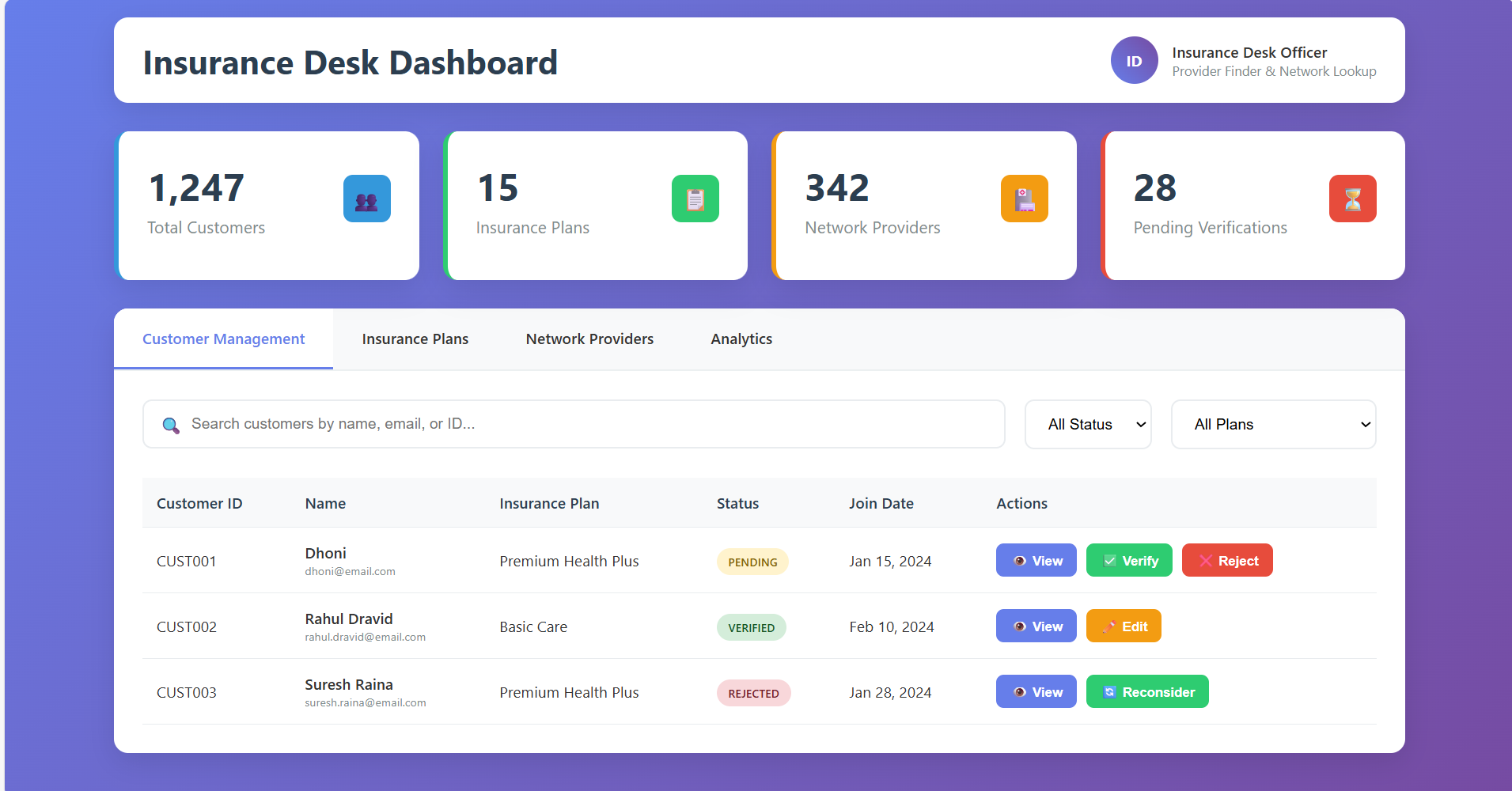


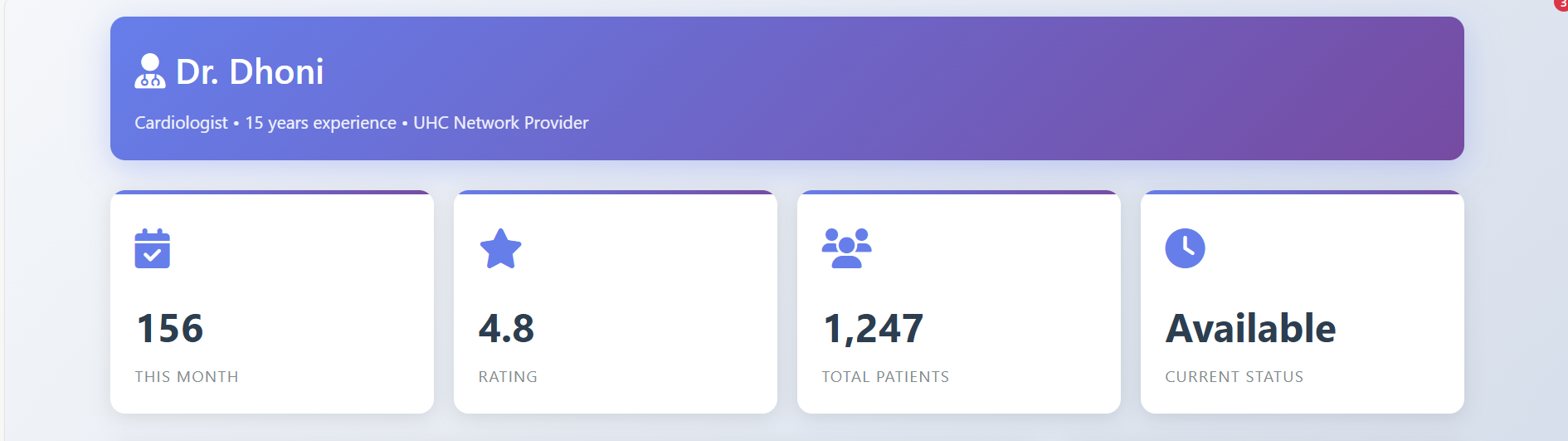


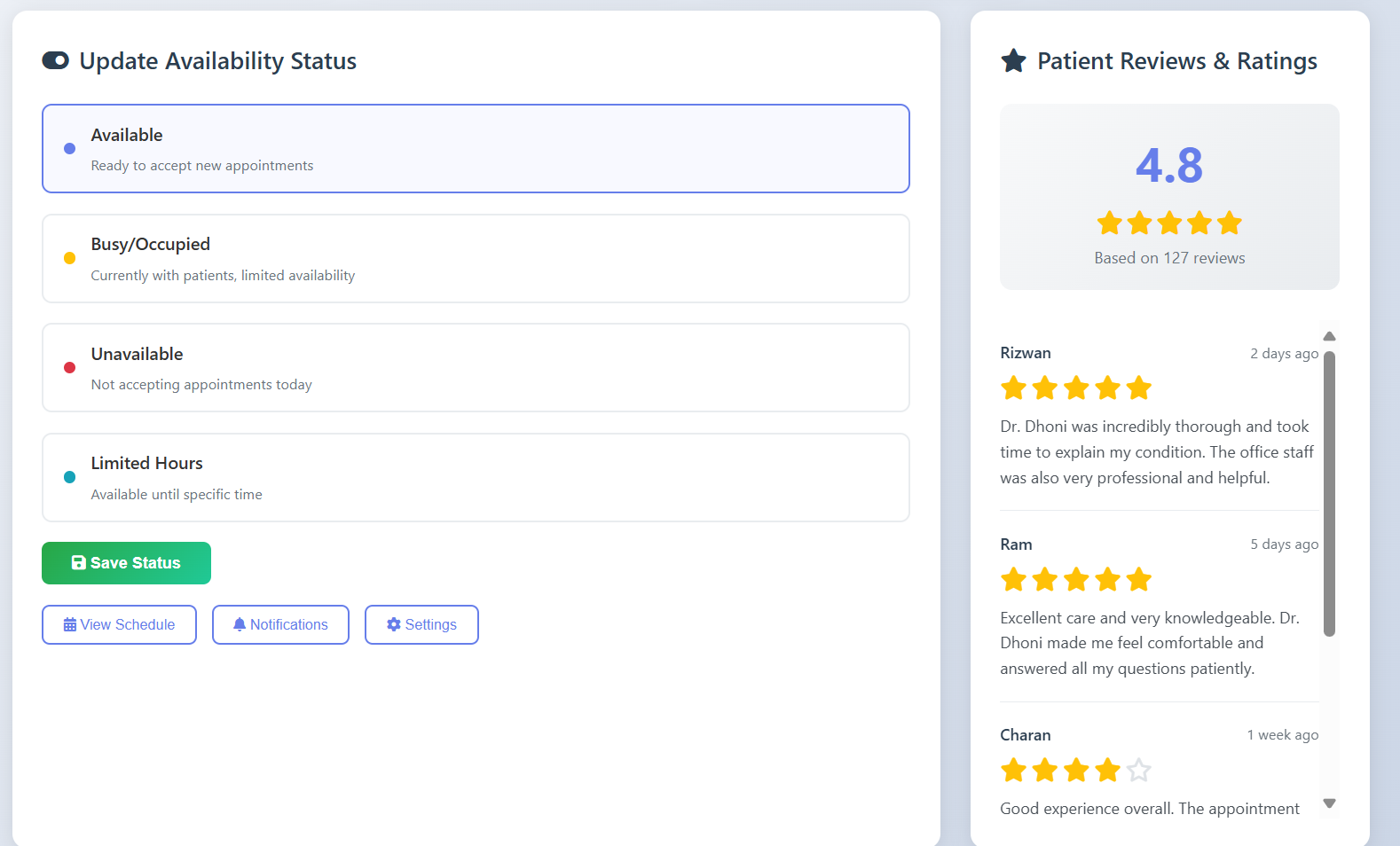
Customer

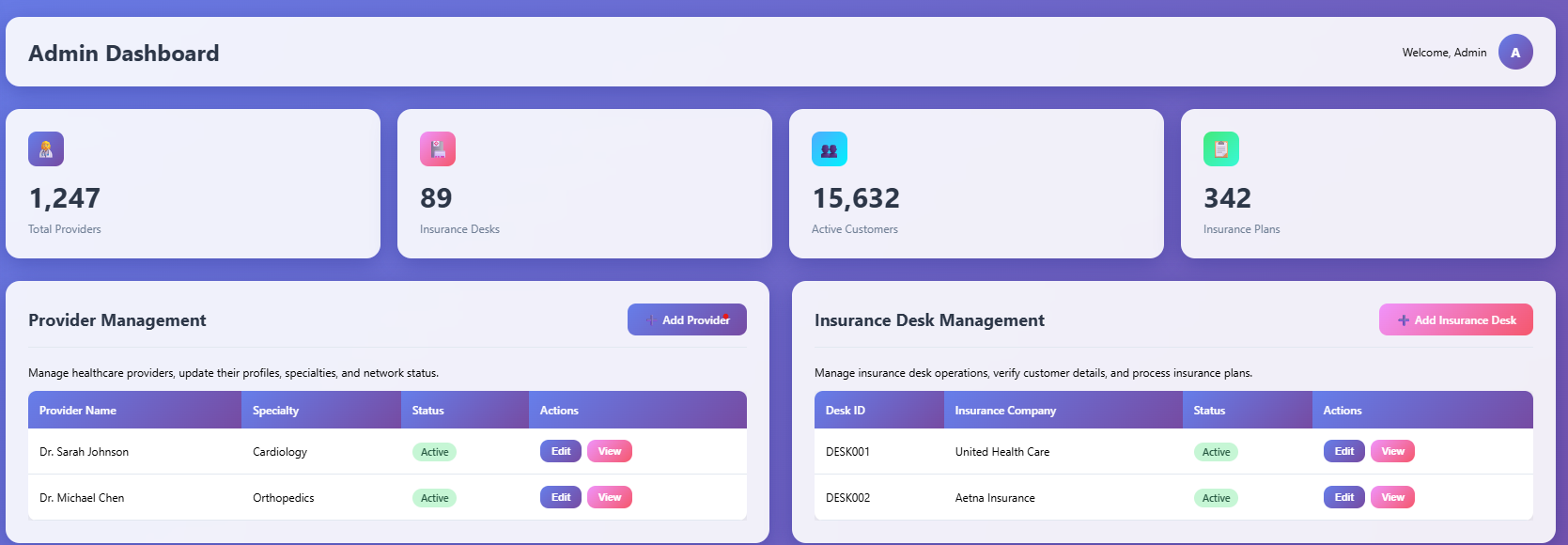












## 2.4 External Interface Requirements

NA

# 3. Project Execution Related Requirements

## 3.1 Development Environment

## Technologies to be used for development of the Application:

* Frontend:  
  React.js for build of dynamic, responsive, and user-friendly web portals accessible by Admin patients, hospital , insurers
* Backend:  
  Java Spring Boot framework using microservices architecture to ensure modularity, scalability, and ease of maintenance.
* Database:  
  MongoDB for flexible, scalable document-based storage to handle complex, nested data like patient records, claims details, and document attachments efficiently.
* Github copilot for developing the microservices

Software tools that can be used for development:

● Spring Tool Suite

● Enterprise Architecture/EdrawMax

● Visual studio code

* MongoDB

## 3.2 Design requirements

Not Applicable

## 3.3 Coding requirements

Since the entire application is being coded in Java Spring Boot, all the coding standards required by Java are applicable here.

## 3.4 Error handling requirements

NA

## 3.5 Resource Requirements

## Technologies to be used for development of the Application:

● MongoDB: Stores users, medications, reminders, inventory, logs.

● Spring Boot: REST APIs to manage data and notification scheduling.

● React: User interface for management and interaction.

● WebSockets: For sending notifications to the users.

● MUI: for reusable functional styled components.

## 3.6 Risk management

Not applicable.

## 3.7 Training Requirements

## Since the application being developed is greatly dependent on user interaction, the end user should be comfortable with the GUI.

* End user should be given a walkthrough of the working of the entire application along Training should be provided with respect to efficient GUI usage.

# 4. Testing Requirements

The testing phase shall include unit testing to validate each individual module of the system,

Integration testing to check if the system works according to customer requirements after modules’

integration. This is supported by regression testing for efficient results.

The testing is to be carried out manually following both black box testing as well as white box in testing in order to understand the system functionality better.

# 5. Quality Assurance Activities

NA

# 6. Software Acceptance Criteria

User acceptance test is the sole software acceptance criteria.

In order to fulfill this criterion, the system should be executed and tested .The errors found during

system testing should be solved and corrected version of the software should be sent to Customer for final Acceptance Testing.

# 7. Deliverables

## 7.1 List of Deliverables

*Not applicable*

## 7.2 Delivery

Not Applicable

# 8. Requirements Acceptance Statement

*Requirement Acceptance statement consists of agreement to the above mentioned requirements, by*

*The customer & supplier()*