**Document Status: Approved**

**Approved by: Management Representative**

**Released by : Mentor Labs**

**Revision / Document History**

| **Ver.** | **Date** | **Changed by** | **Modifications** |
| --- | --- | --- | --- |
|  |  |  |  |
| 1.0.0 | 22/07/2025 | Shiva Prasad P | Project Analysis |
|  |  |  |  |
|  |  |  |  |

**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**

# **Introduction**

## **Purpose**

The purpose of the Billing & Premium Payment Portal project is to provide a secure, user-friendly online system that enables insurance members to view, manage, and pay their monthly premiums conveniently and safely. This portal is designed to streamline the entire premium payment process, support auto-pay features, provide instant receipts and payment history, and integrate with major payment gateways like Razorpay for secure transactions.

## **Scope**

The scope of this project covers the end-to-end experience and supporting infrastructure for insurance members and administrators to manage premium payments securely and efficiently through a web portal.

Customer Functions: To login,view invoices , pay premiums for invoices, autopay settings, payment history, receive notifications

Insurers : Issue Invoices, Payment History of all customers, Send Remainders like Policy expiry remainders, Pay By Cash, invoice history of all customers.

## **Definitions, Acronyms and Abbreviations**

Not Applicable

## **References**

*Not Applicable*

## **Coordinators**

1. Shiva Prasad.P

# **Software Requirements**

## **Operational Requirements**

**Customer Functions:**

Secure login and profile management.

Online viewing invoices.

Making one-time premium payments via integrated gateways (Razorpay/PayPal).

Setting up, managing, and disabling automated (“auto-pay”) premium payments.

Accessing comprehensive payment history.

Receiving automated payment reminders(mail and sms).

Receiving automated Invoicereminders(mail).

Receiving automated Autopay reminders.

**Insurers:**

Assign Invoices

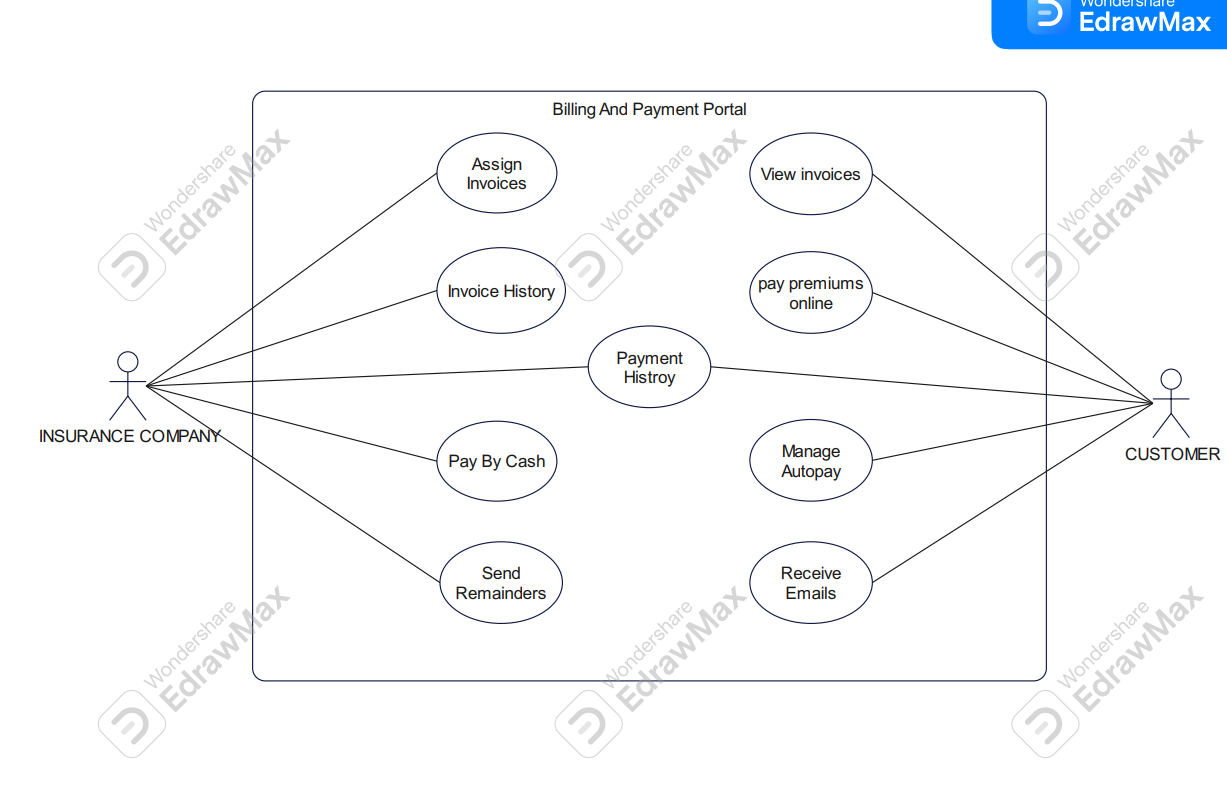
Payment History

Invoice History

Manage Remainders(Send policy expiry mails)

Pay by cash

## **Functional Requirements**

**

*FIGURE 2.1 USE CASE DIAGRAM FOR Billing And Premium Payment Portal*

| Requirement Tag ID | *RS-1-2* | |
| --- | --- | --- |
| *Source/Trigger* | *Insurance company* | |
| *Description* | *Input* | *Invoice details [Invoice History]* |
| *Process* | *Stores all the invoice history of the customers* |
| *Output* | *Operations successful* |
| *Validation Method* |  | |

| Requirement Tag ID | *RS-1-1* | |
| --- | --- | --- |
| *Source/Trigger* | *Insurer* | |
| *Description* | *Input* | *Invoice details [Pay by Cash]* |
| *Process* | *Paying by cash* |
| *Output* | *Operations successful* |
| *Validation Method* |  | |

| Requirement Tag ID | *RS-1-3* | |
| --- | --- | --- |
| *Source/Trigger* | *Insurance company* | |
| *Description* | *Input* | *Payment details [Payment History]* |
| *Process* | *Stores and displays all the payments of the customers* |
| *Output* | *Operations successful* |
| *Validation Method* |  | |

| Requirement Tag ID | *RS-1-4* | |
| --- | --- | --- |
| *Source/Trigger* | *Insurance company* | |
| *Description* | *Input* | *Invoice details [Assign Invoices]* |
| *Process* | *Issue invoices* |
| *Output* | *Operation Successful* |
| *Validation Method* |  | |

| Requirement Tag ID | *RS-1-5* | |
| --- | --- | --- |
| *Source/Trigger* | *Insurance Company* | |
| *Description* | *Input* | *Notification [Policy Expiry Remainders]* |
| *Process* | *Schedule/manage policy expiry reminders* |
| *Output* | *Operation successful* |
| *Validation Method* |  | |

| Requirement Tag ID | *RS-1-6* | |
| --- | --- | --- |
| *Source/Trigger* | *Customers* | |
| *Description* | *Input* | *Payments details [Payment History]* |
| *Process* | *Stores and displays all the payments done by customer* |
| *Output* | *Operation Successful* |
| *Validation Method* |  | |

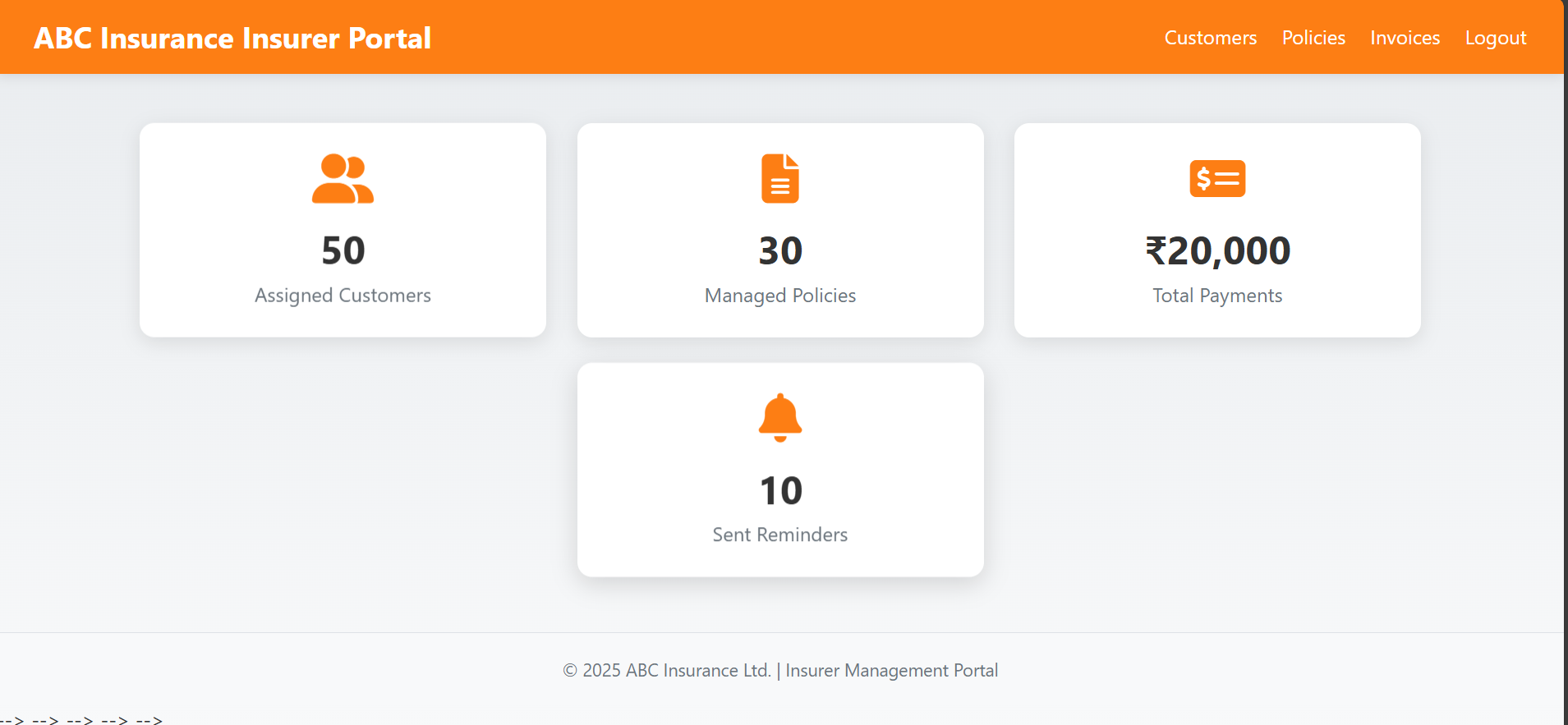
| Requirement Tag ID | *RS-1-6* | |
| --- | --- | --- |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Secure Login [Login]* |
| *Process* | *Login credentials,Authentication* |
| *Output* | *Login successful* |
| *Validation Method* |  | |

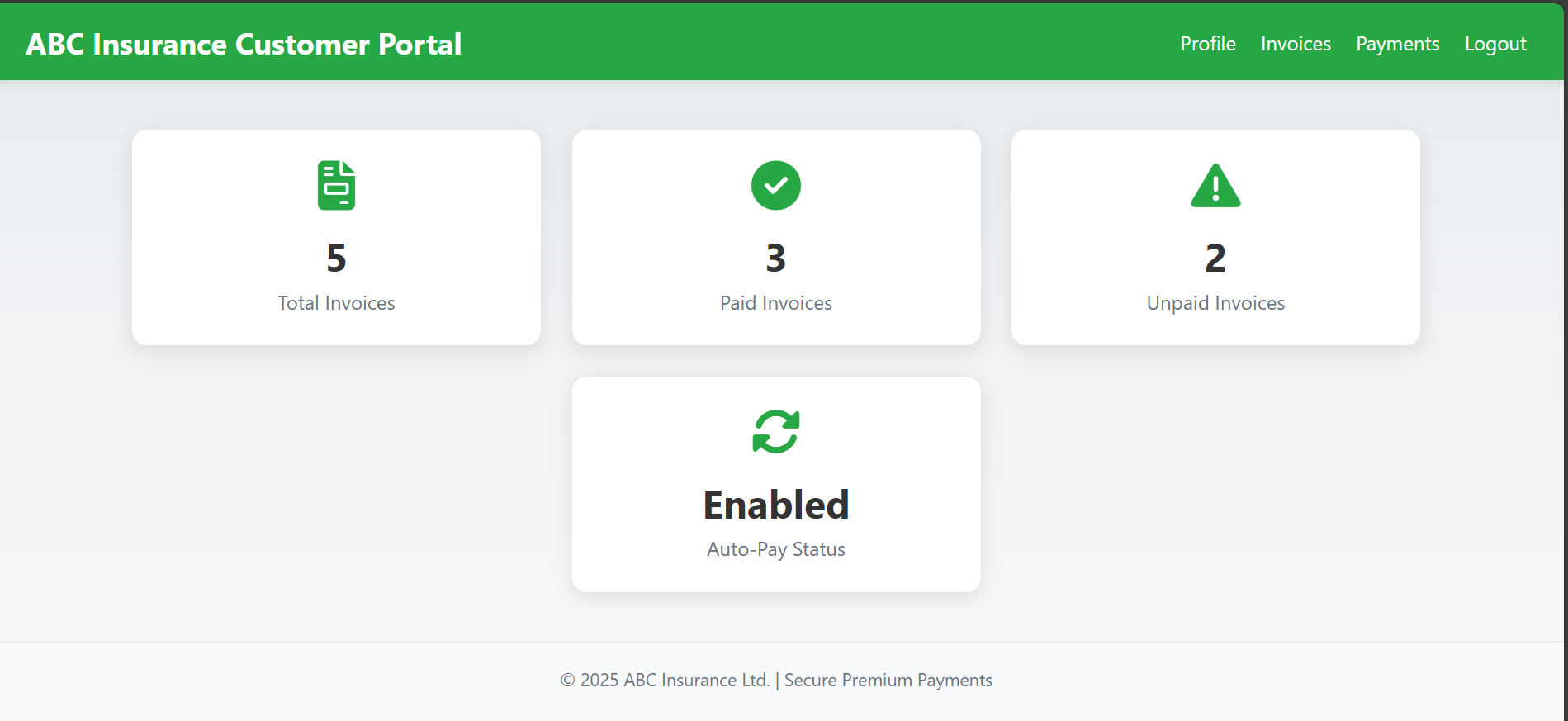
| Requirement Tag ID | *RS-1-7* | |
| --- | --- | --- |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Invoice details [Pay Premiums]* |
| *Process* | *Pay Premiums for Invoices using razorpay* |
| *Output* | *Payment success / failed* |
| *Validation Method* |  | |

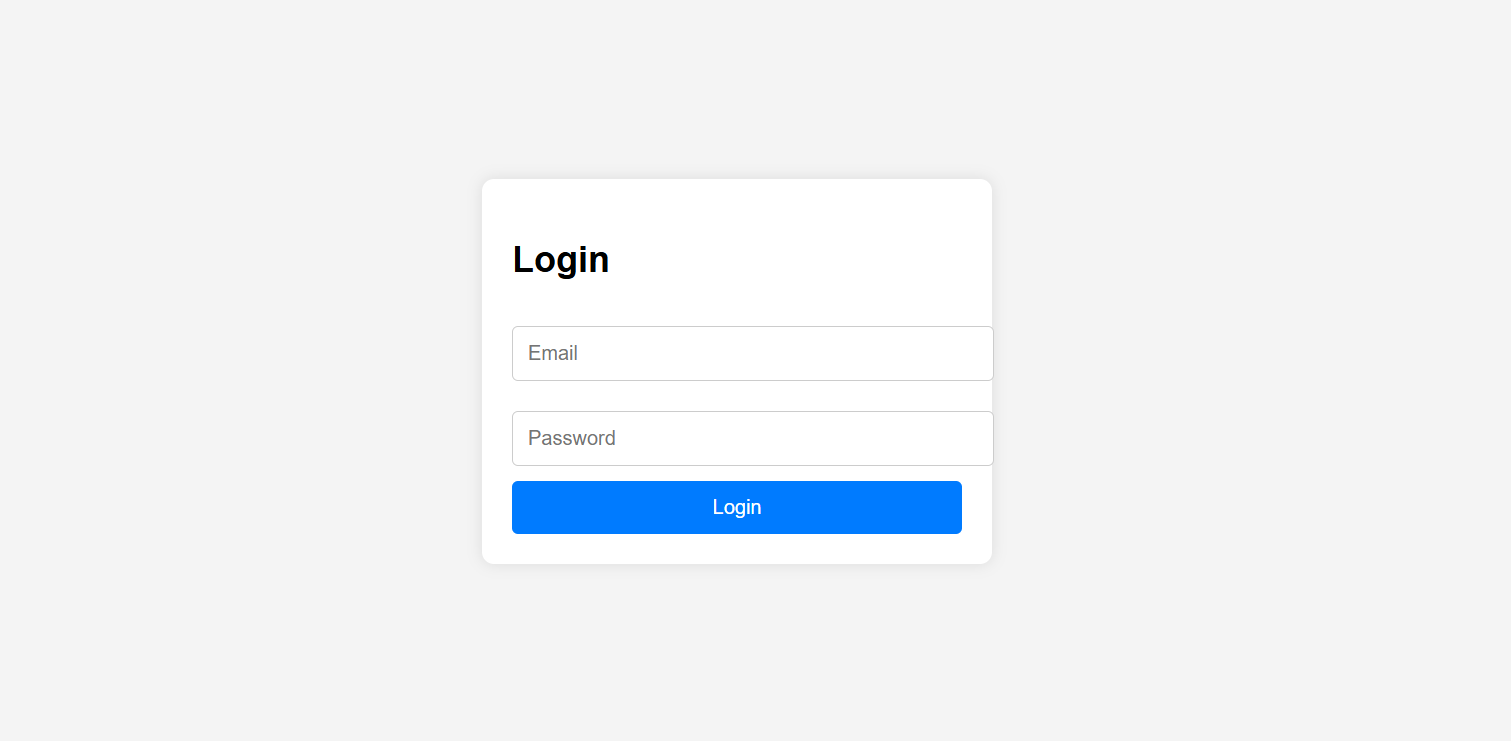
| Requirement Tag ID | *RS-1-8* | |
| --- | --- | --- |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Auto-pay instructions [Autopay]* |
| *Process* | *Enable or Disable Autopay* |
| *Output* | *Auto-pay status updated* |
| *Validation Method* |  | |

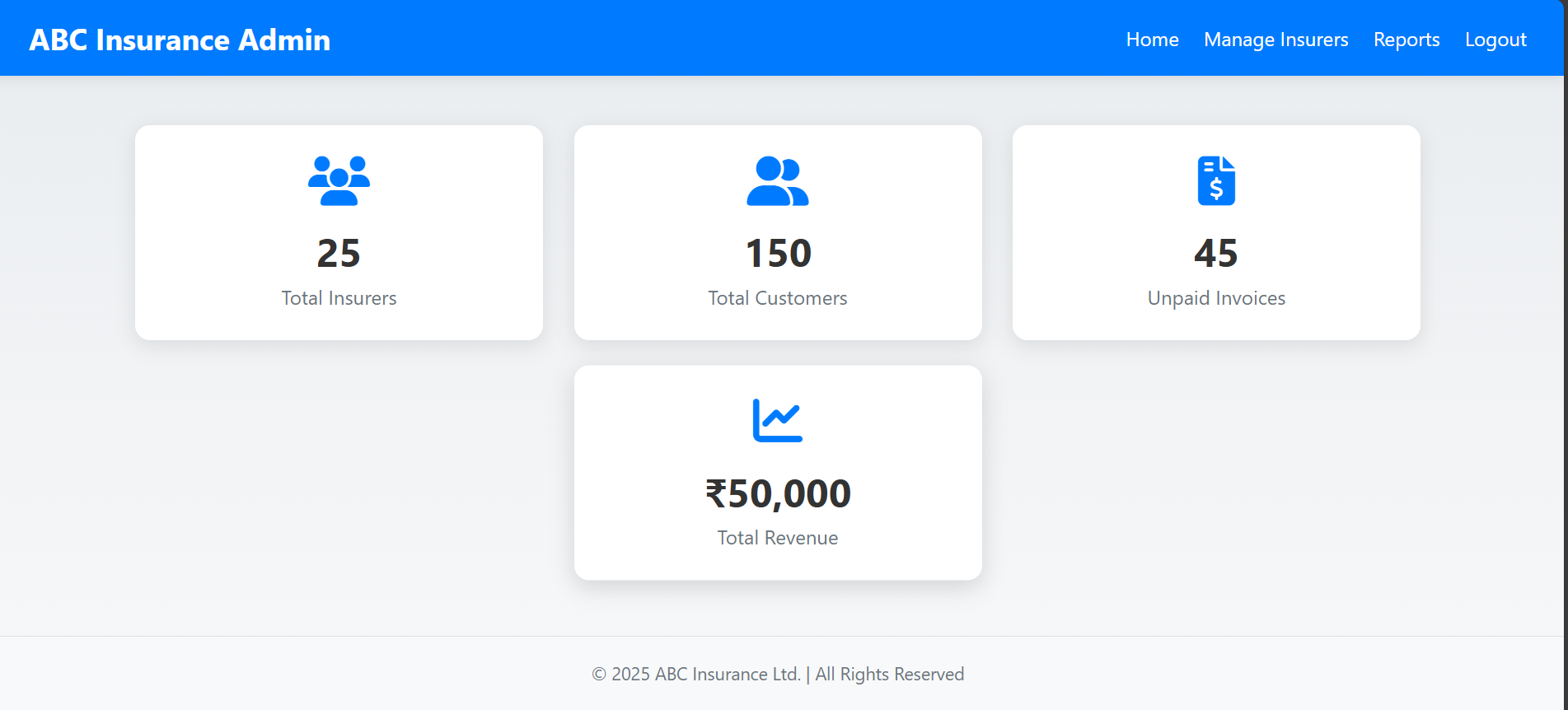
| Requirement Tag ID | *RS-1-9* | |
| --- | --- | --- |
| *Source/Trigger* | *Customer* | |
| *Description* | *Input* | *Notifications [Get Notifications]* |
| *Process* | *Receive notifications* |
| *Output* | *Notifications received* |
| *Validation Method* |  | |

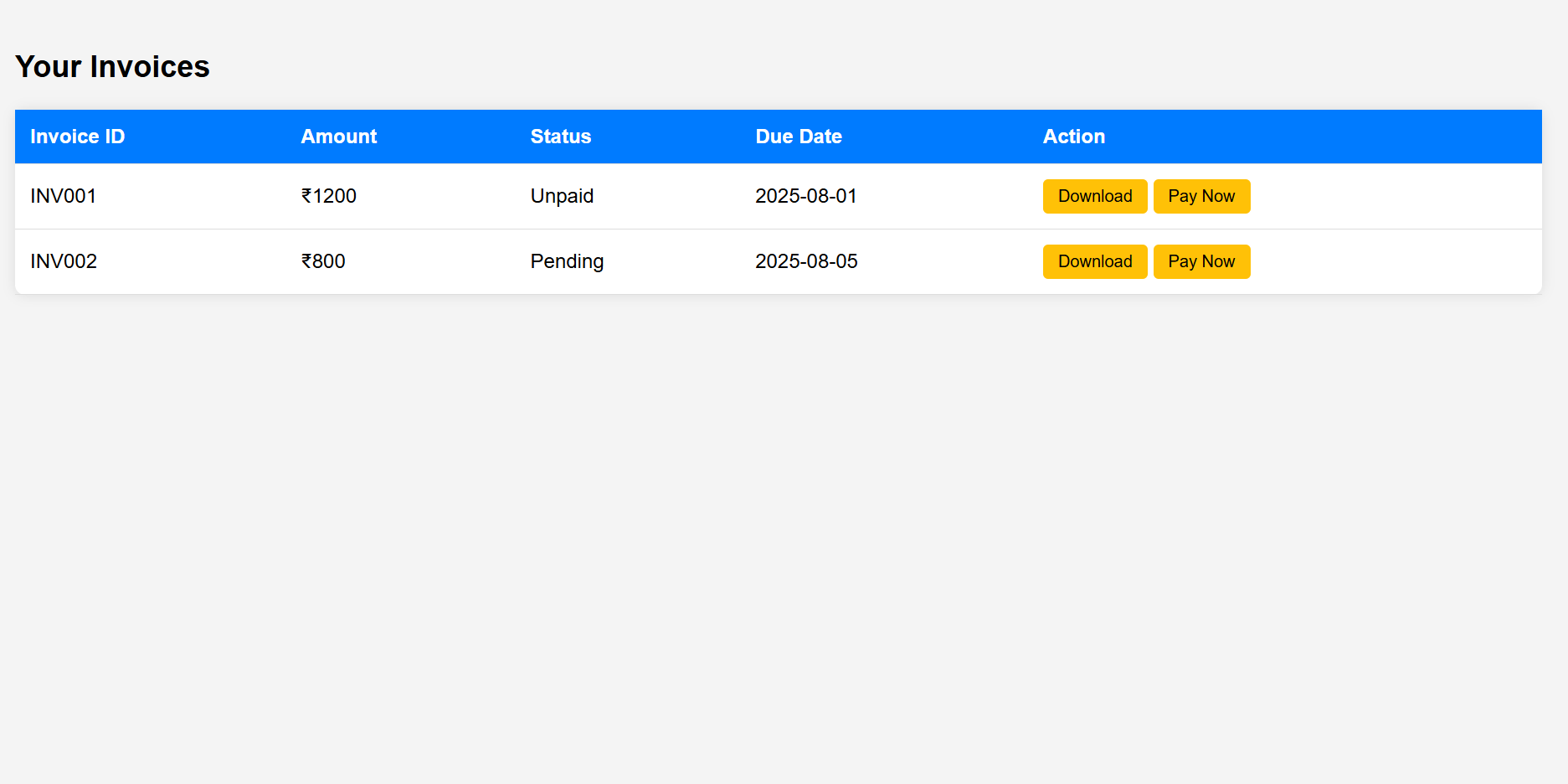
## **User Interface Requirements**











## **External Interface Requirements**

NA

# **Project Execution Related Requirements**

## **Development Environment**

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

* Frontend - [React.js](http://react.js),Node
* Backend - SpringBoot Microservices,RESTful API
* Database - MongoDB (NoSQL)
* Payment Integration - Razorpay API

Software tools that can be used for development:

* IDE - Visual Studio Code,STS
* API Testing: Postman

## **Design requirements**

###### Not applicable.

## **Coding requirements**

*Handled per standard coding conventions for selected technologies. Secure coding practices, proper input validation, and modular design should be enforced.*

## **Error handling requirements**

NA

## **Resource Requirements**

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

* Refer to the technologies listed above in the "Development Environment" section.

## **Risk management**

Not applicable.

## **Training Requirements**

Since the application relies heavily on user interaction, end users should receive a walkthrough and demonstration of all modules.

· Training should focus on efficient and effective use of the UI.

· A training manual or user guide is recommended to facilitate onboarding.

# **Testing Requirements**

The testing phase includes:

· Unit testing: To validate individual modules (using JUnit for backend).

· Integration testing: To ensure system components interact correctly.

· Regression testing: After any new changes/fixes to check existing features. Testing is carried out manually with tools such as Postman and Swagger for black-box (functional) and white-box (code-level) validation.

# **Quality Assurance Activities**

NA

# **Software Acceptance Criteria**

# **Deliverables**

## **List of Deliverables**

*Not applicable*

## **Delivery**

Not Applicable

# **Requirements Acceptance Statement**

Requirements Acceptance statement consists of agreement to above mentioned requirements, by the user & developer.