ADDING INTELLIGENCE

REQUIREMENT DOCUMENT

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

Table of Contents

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

Revision History

Name	Date	Reason For Changes	Version

REQUIREMENT DOCUMENT

1 Introduction

1.1 Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Project Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here. An SRS that specifies the next release of an evolving product should contain its own scope statement as a subset of the long-term strategic product vision.>

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2 Admin Module

The **Admin Module** in the Jewellery ERP system is designed to provide comprehensive control over system configuration, user management, and security. This module enables administrators to define user roles, set access permissions, and manage user accounts, ensuring that each user has appropriate access to modules and data based on their role. Additionally, it offers settings for configuring master data, including product categories, pricing, tax configurations, and vendor details, which form the backbone of the ERP's operations. The Admin Module also includes tools for monitoring system performance, managing backups, and configuring workflows, providing a secure and streamlined environment that supports the unique needs of the jewellery industry. This centralized control empowers administrators to maintain system integrity, streamline operations, and ensure data accuracy across the ERP.

2.1 Currency

The **Currency Form** in the Jewellery ERP system is designed to simplify the management of currency-related data. This form enables users to create, edit, delete, and view details of currencies used across various business operations. By centralizing currency information, the Currency Form ensures accuracy and consistency in financial transactions, including sales, purchases, and reporting. This functionality supports seamless multi-currency handling, enhances compliance with international financial standards, and improves operational efficiency within the ERP system.

2.1.1 Data View

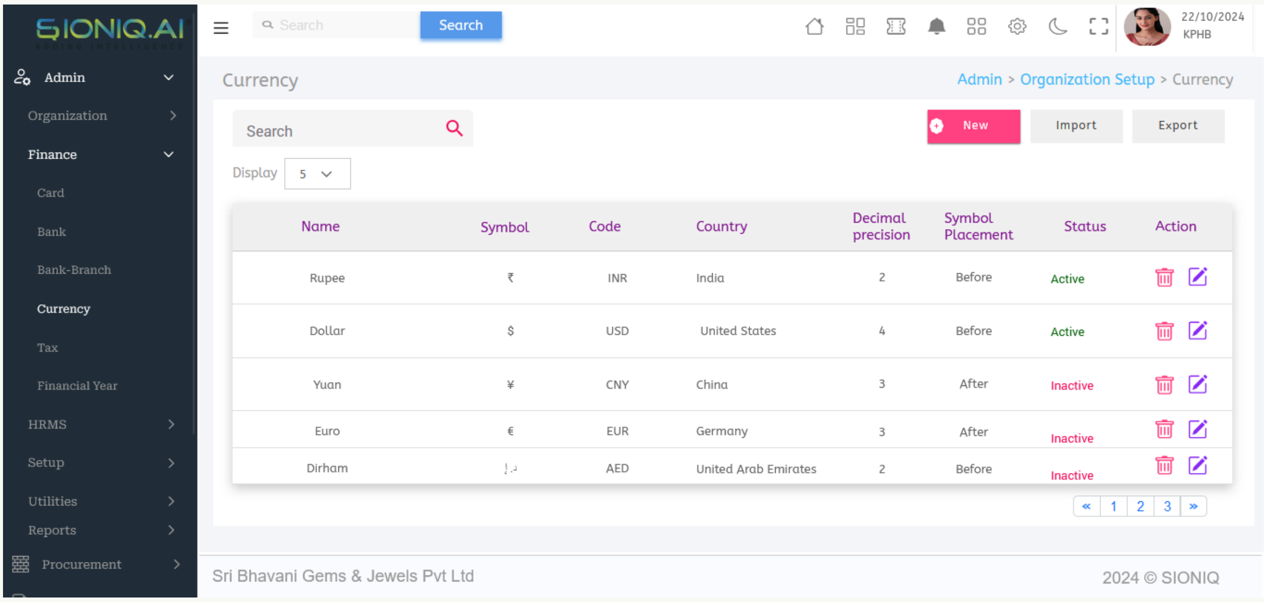
2.1.1.1 Purpose

This section provides a comprehensive overview of all currency details configured in the system. Users can easily access detailed information about each currency, including its code, name, and associated exchange rates, enabling efficient navigation and informed decision-making. The layout ensures quick retrieval of existing records, supporting seamless financial management and operational accuracy.

2.1.1.2 User Type

Registered Users, Admins.

2.1.1.3 Screen Layout



2.1.1.4 UI Elements

- **Search Option**
  - **Description:** A search bar located at the top of the data view section to filter currency displayed in the table.
  - **Functionality:**
    - Enables users to search by Currency Name, Code, or Country.
    - Dynamically updates the displayed table to show only results that match the search query.
- **Display Dropdown**
  - **Description**: A dropdown control to set the number of rows shown in the table at once.
  - **Options**: Common choices (e.g., 5, 10, 25, 50, 100 rows per page).
  - **Functionality**:
    - Allows users to control the number of currency records displayed per page.
    - Updates the table view immediately based on the selected number of rows.
- **Buttons**
  - **Add Button**
    - **Label**: "New"
    - **Functionality**: Opens the Currency Creation Form to add a new currency record.
  - **Export Button**
    - **Label**: "Export" (or Icon button as specified)
    - **Functionality**: Allows users to export the displayed currency data in various file formats (Excel).
    - **Export Options**:
      - Exports all or filtered data based on the current table view and search filter.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Import Button**
  - **Label**: "Import" (or Icon button as specified)
  - **Functionality**:
    - Redirects to the Import Form page, where users can upload currency data.

- **Tabular View**
  - **Description**: Displays a structured view of all currency details registered in the system.
  - **Columns**:
    - **Name:** Displays the name of the currency (e.g., US Dollar, Euro).
    - **Symbol:** Shows the symbol representing the currency (e.g., $, €).
    - **Code:** Displays the currency code (e.g., USD, EUR).
    - **Country:** Indicates the country associated with the currency.
    - **Decimal Precision:** Specifies the number of decimal places used for the currency.
    - **Symbol Placement:** Indicates whether the symbol is placed before or after the value (e.g., $100 or 100€).
    - **Status**: Displays the active/inactive status of the currency.
      - **Status Values**:
      - **Active**: If the checkbox is checked in the form.
      - **Inactive**: If the checkbox is unchecked in the form.
    - **Action**:
      - **Delete**: Deletes the selected currency record (permission-based).
        - **Functionality**: Triggers a delete confirmation prompt before proceeding.
      - **Edit**: Allows modification of the currency’s information (permission-based).
        - **Functionality**: Opens the Currency Creation Form populated with the selected Currency data for editing.

- **Paging for Table View**
  - **Description:** Includes pagination controls at the bottom of the table for navigating through multiple pages of currency records.
  - **Functionality:**
    - Displays navigation buttons (e.g., "Previous", "Next") to move between pages.
    - Shows page numbers to allow users to jump directly to specific pages.
    - Updates the displayed currency records based on the selected page.

2.1.1.5 Functional Requirements

None.

2.1.1.6 Data Requirements

None.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.1.1.7 Non-Functional Requirements

None.

2.1.1.8 Configurations

- **Functionality:**
  - **Permission Settings**: Restricts the visibility and accessibility of buttons and actions based on user roles or permissions.
- **Available Permissions:**
  - **Add Currency Permission**: Grants authorized users access to the "Add Currency" button for creating new currency records.
  - **Edit Permission**: Enables authorized users to access the **Edit** button in the **Action** column, allowing modification of existing currency records.
  - **Delete Permission**: Provides authorized users access to the **Delete** button in the **Action** column, enabling deletion of currency records.
- **Permission Assignment:** Admins or authorized users can assign permissions to individual users or user roles.

2.1.1.9 Flowchart

None.

2.1.1.10 Additional Notes

None.

2.1.2 Create Currency

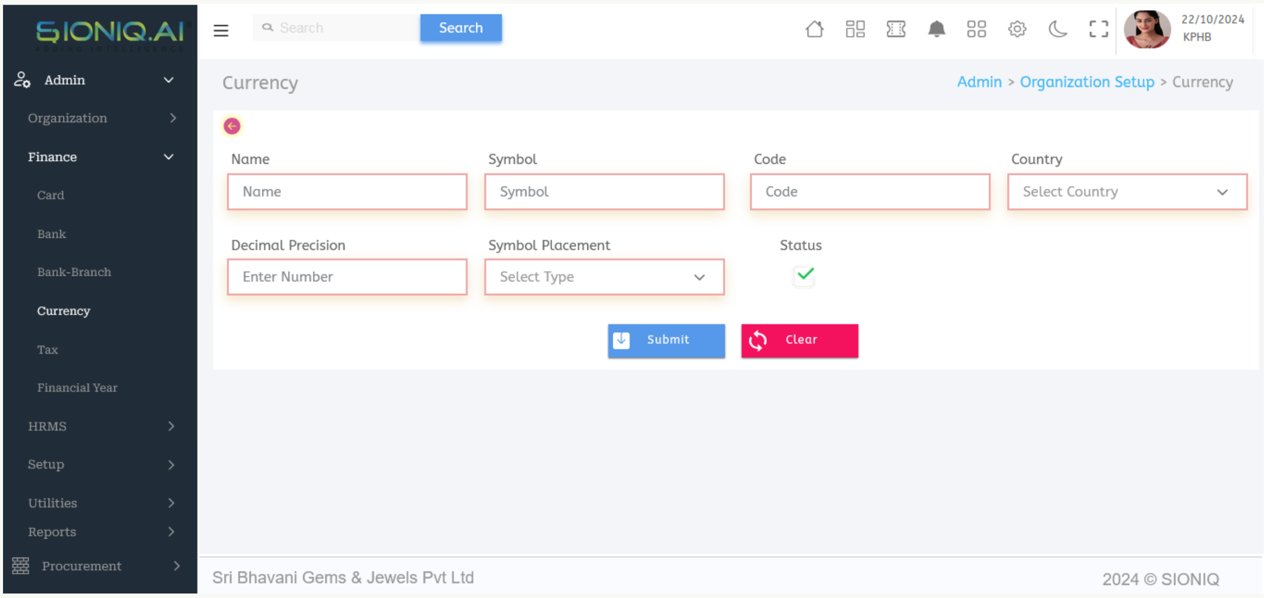
2.1.2.1 Purpose

This section allows users to add a new currency to the ERP system. Users can enter essential details such as the currency name, symbol, code, country, decimal precision, and symbol placement. This process ensures accurate and consistent currency data entry for seamless financial operations and reporting.

2.1.2.2 User Type

Registered Users, Admins.

2.1.2.3 *Screen Layout*



2.1.2.4 *UI Elements*

- **Name**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Name"
  - **Editable:** Yes
  - **Description:** Allows users to enter the full name of the currency (e.g., US Dollar, Euro). This input will be used to identify the currency in the system.
  - **Validation:**
    - **Required:** Yes
    - **Character Limit:** 50 characters
    - **Allowed Characters:** Letters, spaces, and special characters.
- **Symbol**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Symbol"
  - **Editable:** Yes
  - **Description:** Allows users to enter the symbol representing the currency (e.g., $, €).
  - **Validation:**
    - **Required:** Yes

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Character Limit:** 5 characters.

- **Code**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Code"
  - **Editable:** Yes
  - **Description:** Allows users to enter the ISO 4217 currency code (e.g., USD, EUR).
  - **Validation:**
    - **Required:** Yes
    - **Character Limit:** 3 characters
    - **Allowed Characters:** Uppercase letters only.

- **Country**
  - **Type**: Dropdown
  - **Label**: Country
  - **Editable**: Yes
  - **Description**: Select the country associated with the currency.
  - **Options**: Import Country
    - **Source**: Country Master

- **Decimal Precision**
  - **Type:** Text Input
  - **Label:** "Decimal Precision"
  - **Editable:** Yes
  - **Description:** Specifies the number of decimal places used for the currency.
  - **Validation:**
    - **Required:** Yes
    - **Allowed Values:** Integer (0-6).

- **Symbol Placement**
  - **Type:** Dropdown
  - **Label:** "Symbol Placement"

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Editable:** Yes
- **Description:** Indicates whether the currency symbol is placed before or after the value.
- **Options:**
  - **Before**
  - **After**
- **Validation:**
  - **Required:** Yes

- **Is Active**
  - **Type:** Checkbox
  - **Label:** "Is Active"
  - **Editable:** Yes
  - **Default Value:** Checked (True)
  - **Behavior:**
    - **Checked:** The currency is marked as active.
    - **Unchecked:** The currency is inactive but remains in the database.
- **Submit Button**
  - **Type:** Button
  - **Label:** "Submit"
  - **Editable:** No
  - **Description:** Submits the filled form data to the database.
  - **Validation:** Ensures all mandatory fields are filled and validated.
- **Clear Button**
  - **Type:** Button
  - **Label:** "Clear"
  - **Editable:** No
  - **Description:** Resets or clears all form fields, enabling a fresh start.
- **Previous Button**
  - **Type:** Button

- **Label:** "Previous"
- **Editable:** No
- **Description:** Closes the Currency Creation Form and returns the user to the main Currency table view, discarding unsaved changes.

2.1.2.5 Functional Requirements

- **Actions**
  - **Submit Button Click**
    - **Action:** Triggers form submission if all required fields (Name, Symbol, and Code) contain valid values.
    - **Outcome:**
      - **Create Operation:** If it's a new entry, saves the new currency information.
- Validation
  - **Mandatory Fields**
    - **Fields:** Name, Symbol, Country and Code are required fields.
    - **Behavior:**
      - Users must enter values in each mandatory input field.
      - If any mandatory field is left empty, the form submission is blocked, and an error message is displayed (details in the "Error Messages" section).
- Error Messages
  - **Submit Attempt with Missing Values**
    - **Condition:** If one or more required fields are missing a value when the user clicks Submit.
    - **Behavior:** The system displays a specific error message indicating which field(s) are missing values.
    - **Message Examples:**
      - "Name is required" if the Name field is empty.
      - "Symbol is required" if the Symbol field is empty.
      - "Code is required" if the Code field is empty.
      - "Country is required" if the Country field is not selected.

2.1.2.6 Data Requirements

- **Country**
  - **Source:** Country Master table
  - **Usage:**
    - **Populate the Country dropdown**: The Country dropdown in the Currency Creation Form is dynamically populated with data from the Country Master table.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Validate the Country**: Ensures that the selected country is valid and exists in the Country Master table.

- **Details:**
  - The dropdown lists all available countries from the Country Master table.

2.1.2.7 Non-Functional Requirements

None.

2.1.2.8 Configurations

None.

2.1.2.9 Flowchart

None.

2.1.2.10 Additional Notes

None.

2.1.3 Update Currency

2.1.3.1 Purpose

The Edit Currency section allows authorized users to update existing currency records, ensuring the accuracy and relevance of financial data in alignment with evolving international standards and business requirements.

2.1.3.2 User Type

Registered Users, Admins.

2.1.3.3 Screen Layout

Same Create Currency Layout.

2.1.3.4 UI Elements

Same Create Currency Elements.

2.1.3.5 Functional Requirements

- **Initiate Update Process**
  - **Description:** The update process is initiated by clicking the Edit button in the Action column of the currency table.
  - **Functionality:**

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Edit Button Action:**
  - Loads the selected currency's existing details (Name, Symbol, Code, Country, Decimal Precision, Symbol Placement, and Is Active status) into the Currency Creation Form.
  - Pre-populates the form fields with the current values of the selected currency record to enable editing.
- **Expected Outcome:** The Currency Creation Form is populated with the selected currency's current data, ready for modification.

- **Modify Currency Details**
  - **Description:** Allows users to change one or more fields in the currency record.
  - **Fields Available for Update:**
    - **Name:** Modify the currency's full name (e.g., US Dollar, Euro).
    - **Symbol:** Update the currency symbol (e.g., $, €).
    - **Code:** Update the ISO 4217 currency code (e.g., USD, EUR).
    - **Country:** Change the associated country.
    - **Decimal Precision:** Update the number of decimal places used for the currency.
    - **Symbol Placement:** Modify the placement of the symbol (Before or After).
    - **Is Active:** Set the active status of the currency to either "Active" or "Inactive."
  - **Expected Outcome:** Users can update all editable fields as required before saving the record.

- **Form Validation (Update Mode)**
  - **Purpose:** Ensures that updated information meets validation requirements before submission.
  - **Validation Criteria:**
    - **Mandatory Fields:** Name, Symbol, and Code must not be empty.
    - **Specific Field Validation:**
      - **Name:** Up to 50 characters, allows letters, spaces, and certain special characters.
      - **Symbol:** Up to 5 characters, allows any printable character.
      - **Code:** Exactly 3 uppercase letters.
    - **Error Handling:** If validation fails, display specific error messages to guide the user (e.g., "Currency Code must be unique").
  - **Expected Outcome:** All required fields are validated, and any errors are shown to the user before submitting the update.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Submit Updated Data**
  - **Description:** Saves the modified data to the database upon successful validation.
  - **Functionality:**
    - **Submit Button:**
      - Saves the updated currency data to the database and refreshes the table to reflect changes.
      - Displays a success message to confirm the update (e.g., "Currency details have been successfully updated").
  - **Expected Outcome:** The updated currency information is stored in the database, and the user receives confirmation of the successful update.
- **Cancel Update**
  - **Purpose:** Allows users to discard changes if they choose not to update the currency record.
  - **Functionality:**
    - **Cancel Option:** Users can cancel the update operation by closing the form or clicking a "Previous" button.
  - **Expected Outcome:** The form closes without saving changes, leaving the original currency record intact.
- **Confirmation Message**
  - **Purpose:** Notifies the user of the successful completion of the update operation.
  - **Functionality:**
    - **Display Message:** Upon a successful update, display a confirmation message (e.g., "Currency details have been successfully updated").
  - **Expected Outcome:** The confirmation message assures the user that the changes were saved.

2.1.3.6 Data Requirements

None.

2.1.3.7 Non-Functional Requirements

None.

2.1.3.8 Configurations

None.

2.1.3.9 Flowchart

None.

2.1.3.10 Additional Notes

None.

2.1.4 Remove Currency

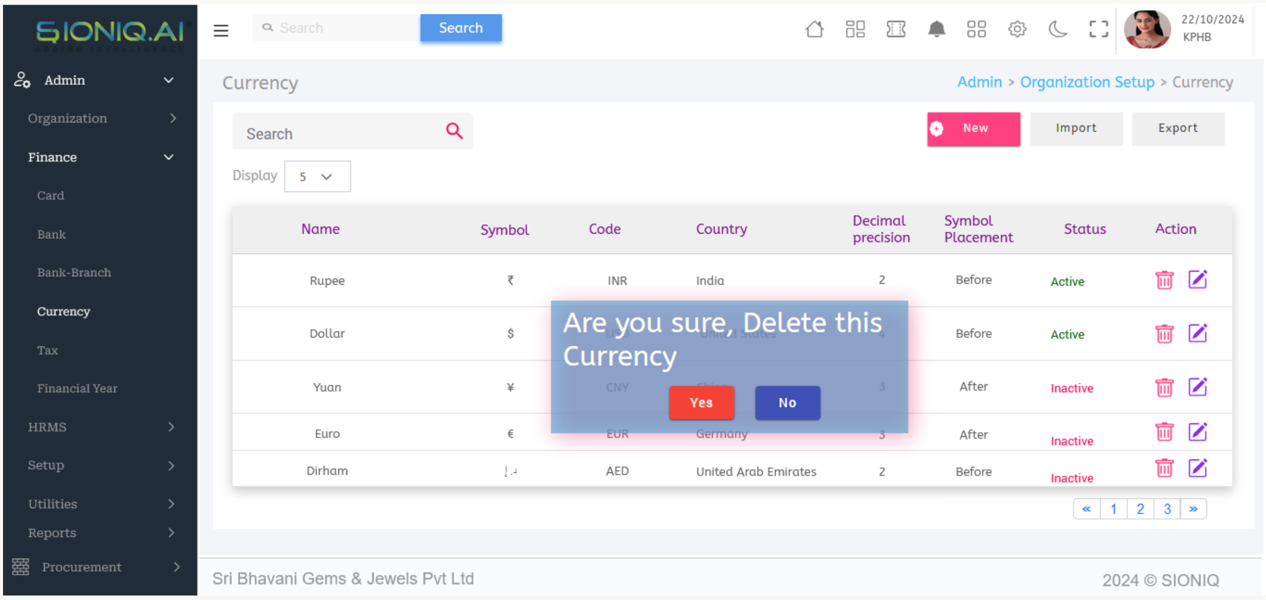
2.1.4.1 Purpose

This section allows for the safe removal of a currency from the ERP system, typically when the currency is no longer relevant to business operations. The action may be restricted if there are dependencies, such as existing records in other modules linked to that currency, and safeguards are in place to ensure data integrity across the system.

2.1.4.2 User Type

Registered Users, Admins.

2.1.4.3 Screen Layout



2.1.4.4 UI Elements

None.

2.1.4.5 Functional Requirements

- **Access Control**
  - **Purpose:** Ensure that only authorized users can perform the delete operation.
  - **Functionality:**

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Delete Permission:** The Delete button is only accessible to users who have been granted delete permissions in the Configuration Section.

- **Initiate Delete Action**
  - **Description:** The delete process begins when an authorized user clicks the Delete button in the Action column of the currency table.
  - **Functionality:**
    - **Delete Button Action:**
      - Prompts a delete confirmation message to prevent accidental deletion.
- Delete Confirmation Prompt
  - **Purpose:** Provide a confirmation prompt to ensure the user wants to proceed with deletion.
  - **Functionality:**
    - **Confirmation Message:** When the Delete button is clicked, display a warning message:
      - "Are you sure you want to delete this currency?"
    - **Options:**
      - **Yes:** Confirms the deletion and proceeds with removing the currency record.
      - **No:** Cancels the delete action without any changes to the data.
  - **Expected Outcome:** The currency record is deleted only if the user confirms by selecting "Yes."
- Delete Execution
  - **Description:** Once confirmed, the system proceeds to update the currency record in the database to mark it as inactive (status = 0).
  - **Functionality:**
    - **Data Removal:** Updates the selected currency record in the database with status = 0.
    - **Dependencies Check:**
      - Before deletion, the system checks for any dependencies or associations (e.g., if the currency is linked to any transactions or records).
      - If dependencies exist, display a message (e.g., "Cannot delete currency as it is associated with active records.") and halt the delete process.
- **Success Message**
  - **Purpose:** Inform the user of the successful completion of the delete action.
  - **Functionality:**

    - **Display Message:** Upon successful deletion, display a success message such as "Currency has been successfully deleted."
  - **Expected Outcome:** The user sees a confirmation that the currency record was deleted.
- **Refresh Table View**
  - **Description:** Automatically updates the currency table view to reflect the changes made by the deletion.
  - **Functionality:**
    - **Refreshes the Table View:** Removes the deleted currency record from the list.
    - **Pagination Update:** Adjusts the pagination if necessary, ensuring the remaining records are displayed correctly.

2.1.4.6 Data Requirements

None.

2.1.4.7 Non-Functional Requirements

None.

2.1.4.8 Configurations

None.

2.1.4.9 Flowchart

None.

2.1.4.10 Additional Notes

None.

2.2 Taxation

The **Tax Form** in the Jewellery ERP system is designed to simplify the entry and management of tax-related information. This form enables users to create, edit, delete, and view details about various tax configurations applicable within the jewellery business. By centralizing tax data, the Tax Form ensures consistent application of tax rules across modules such as sales, purchases, and inventory management. This functionality supports accurate tax calculations, compliance with regional tax regulations, and seamless integration into financial reporting within the ERP system.

2.2.1 Data View

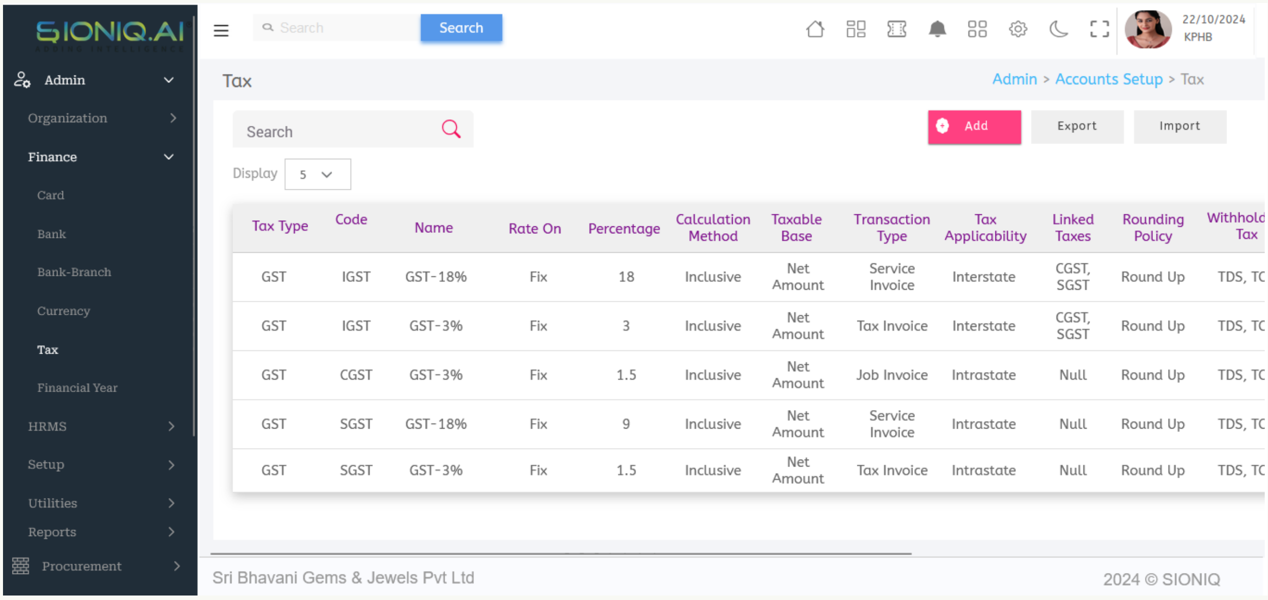
2.2.1.1 Purpose

This section provides an overview of all tax configurations currently registered in the system. Users can view detailed information for each tax record, enabling quick access to essential tax data. This supports informed decision-making, ensures compliance with tax regulations, and facilitates streamlined navigation across the system.

2.2.1.2 User Type

Registered Users, Admins.

2.2.1.3 Screen Layout



2.2.1.4 UI Elements

- **Search Option**
  - **Description:** A search bar located at the top of the data view section to filter tax records displayed in the table.
  - **Functionality:**
    - Enables users to search by **Tax Type**, **Code**, or **Name**.
    - Dynamically updates the table to show only results matching the search query.
- **Display Dropdown**
  - **Description:** A dropdown control to set the number of rows shown in the table at once.
  - **Options:** Common choices (e.g., 5, 10, 25, 50, 100 rows per page).
  - **Functionality:**

- Allows users to control the number of tax records displayed per page.
- Updates the table view immediately based on the selected number of rows.

- **Buttons**
  - **Add Tax Button**
    - **Label:** "New"
    - **Functionality:** Opens the Tax Creation Form to add a new tax record.
  - **Export Button**
    - **Label:** "Export" (or Icon button as specified)
    - **Functionality:** Allows users to export the displayed tax data in various file formats (e.g., Excel).
    - **Export Options:**
      - Exports all or filtered data based on the current table view and search filter.
  - **Import Button**
    - **Label:** "Import" (or Icon button as specified)
    - **Functionality:** Redirects to the Import Form page, where users can upload tax data.
- **Tabular View**
  - **Description:** Displays a structured view of all created tax records.
  - **Columns:**
    - **Tax Type:** The category or type of tax (e.g., GST, VAT).
    - **Code:** A unique alphanumeric identifier for the tax.
    - **Name:** The full name of the tax.
    - **Rate On:** Specifies what the tax rate is applied to (e.g., Fix, Slab).
    - **Percentage:** The percentage rate of the tax.
    - **Taxable Base:** Describes the base amount on which the tax is computed (e.g., Net Amount, Gross Amount).
    - **Tax Applicability:** Shows where the tax is applicable (e.g., within state, interstate).
    - **Transaction Type:** The type of transactions to which the tax applies (e.g., sales, purchases).
    - **Linked Taxes:** Lists any additional taxes associated with this tax.
    - **Calculation Method:** Indicates how the linked tax is calculated (e.g., additive, inclusive).
    - **Rounding Policy:** Specifies how tax amounts are rounded (e.g., up, down, nearest).

- **Withholding Tax:** Indicates if the tax includes a withholding component (e.g., tds, tcs).
- **Country**: Displays the country associated with the tax.
- **State**: Shows the state associated with the tax.
- **Filing Frequency:** Describes how often the tax must be filed (e.g., monthly, quarterly).
- **Effective Date:** The date the tax becomes effective.
- **Status:** Displays whether the tax is active or inactive.
  - **Active:** If the checkbox is checked in the form.
  - **Inactive:** If the checkbox is unchecked in the form.
- **Action:**
  - **Delete:** Deletes the selected tax record (permission-based).
    - **Functionality:** Triggers a delete confirmation prompt.
  - **Edit:** Allows modification of a tax's information (permission-based).
    - **Functionality:** Loads the selected tax's data into the Tax Creation Form for editing.

- **Paging for Table View**
  - **Description:** Includes pagination controls at the bottom of the table for navigating through multiple pages of tax records.
  - **Functionality:**
    - Displays navigation buttons (e.g., "Previous", "Next") to move between pages.
    - Shows page numbers to allow users to jump directly to specific pages.
    - Updates the displayed tax records based on the selected page.

2.2.1.5 Functional Requirements

None.

2.2.1.6 Data Requirements

None.

2.2.1.7 Non-Functional Requirements

None.

2.2.1.8 Configurations

- Functionality:

- **Permission Settings**: Controls the visibility and accessibility of buttons and actions based on user roles or permissions.

○ Available Permissions:

- **Add Tax Permission**: Grants access to the "Add Tax" button for creating new tax records.
- **Export Permission**: Grants access to the Export button for exporting tax data.
- **Import Permission**: Grants access to the Import button for importing tax data.
- **Edit Permission**: Allows users to access the Edit button in the Action column to modify tax records.
- **Delete Permission**: Allows users to access the Delete button in the Action column to delete tax records.

○ Permission Assignment:

- **Admins or Authorized Users**:
  - Can assign permissions to individual users or user roles.
  - Ensure proper role-based access control to maintain data integrity and security.

2.2.1.9 Flowchart

None.

2.2.1.10 Additional Notes

None.

2.2.2 Create Tax

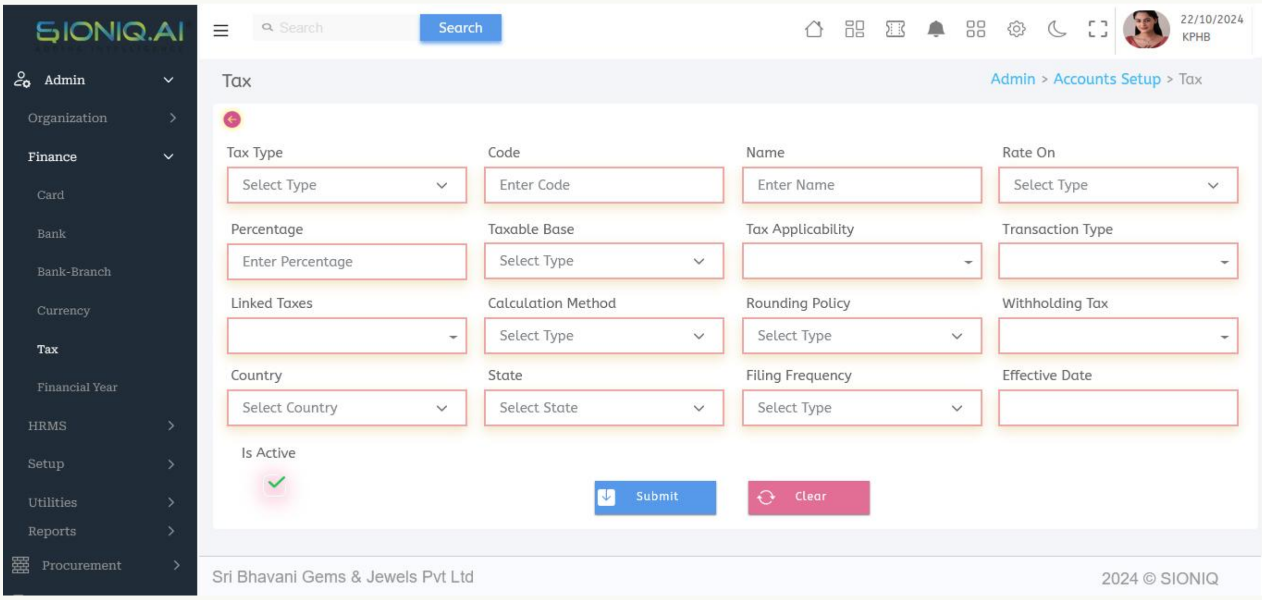
2.2.2.1 Purpose

The Create Tax Form allows users to define and configure tax details within the ERP system. Users can input key information such as tax name, tax type (e.g., GST, VAT, Sales Tax), tax rate, effective date, and applicable regions or categories. This process ensures accurate tax calculations, compliance with regulatory requirements, and consistent application across all transactions for operational and reporting accuracy.

2.2.2.2 User Type

Registered Users, Admins.

2.2.2.3 Screen Layout



2.2.2.4 UI Elements

- **Tax Type**
  - **Type:** Dropdown
  - **Label:** "Tax Type"
  - **Editable:** Yes
  - **Description:** Allows users to select the type of tax from predefined options imported from the Tax Type table.
  - **Validation:**
    - **Required**: Yes
- **Code**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Code"
  - **Editable:** Yes
  - **Description:** Field to input a unique tax code for identification within the system.
  - **Validation:**
    - **Required**: Yes
    - **Character Limit**: 10 characters

- **Allowed Characters**: Alphanumeric only

- **Name**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Name"
  - **Editable:** Yes
  - **Description:** Field to input the name of the tax for easy identification.
  - **Validation:**
    - **Required**: Yes
    - **Character Limit**: 50 characters
    - **Allowed Characters**: Letters, spaces, and special characters
- **Rate On**
  - **Type:** Dropdown
  - **Label:** "Rate On"
  - **Editable:** Yes
  - **Options:** Flex, Slab
  - **Description:** Determines whether the tax rate is flexible or slab-based.
  - **Validation:**
    - **Required**: Yes
- **Percentage**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Percentage"
  - **Editable:** Yes
  - **Description:** Field to input the tax percentage rate.
  - **Validation:**
    - **Required**: Yes
    - **Format**: Numeric values only
- **Taxable Base**
  - **Type:** Dropdown

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Label:** "Taxable Base"
- **Editable:** Yes
- **Options:** Net Amount, Gross Amount
- **Description:** Allows users to define whether the tax applies to the net or gross amount.
- **Validation:**
  - **Required**: Yes

- **Transaction Type**
  - **Type:** Dropdown
  - **Label:** "Transaction Type"
  - **Editable:** Yes
  - **Options:** International, Domestic, Interstate, Intrastate
  - **Description:** Specifies the type of transaction the tax applies to.
  - **Validation:**
    - **Required**: Yes
- **Tax Applicability**
  - **Type:** Dropdown
  - **Label:** "Tax Applicability"
  - **Editable:** Yes
  - **Options:** Service Invoice, Sales Invoice, Job Invoice, Purchase Invoice
  - **Description:** Defines the types of invoices the tax is applicable to.
  - **Validation:**
    - **Required**: Yes
- **Linked Taxes**
  - **Type:** Dropdown
  - **Label:** "Linked Taxes"
  - **Editable:** Yes
  - **Options:** Values imported from the Tax table
  - **Description:** Enables linking of this tax to other predefined taxes.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Validation:**
  - **Required**: No

- **Calculation Method**
  - **Type:** Dropdown
  - **Label:** "Calculation Method"
  - **Editable:** Yes
  - **Options:** Inclusive, Exclusive
  - **Description:** Defines whether the tax is calculated inclusively or exclusively.
  - **Validation:**
    - **Required**: Yes

- **Rounding Policy**
  - **Type:** Dropdown
  - **Label:** "Rounding Policy"
  - **Editable:** Yes
  - **Options:** Round Up, Down, Nearest Whole Number
  - **Description:** Specifies the rounding policy for tax calculations.
  - **Validation:**
    - **Required**: Yes

- **Withholding Tax**
  - **Type:** Dropdown
  - **Label:** "Withholding Tax"
  - **Editable:** Yes
  - **Options:** Values imported from the Tax table filtered by Withholding Tax type
  - **Description:** Links a withholding tax type if applicable.
  - **Validation:**
    - **Required**: No

- **Country**
  - **Type:** Dropdown

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Label:** "Country"
- **Editable:** Yes
- **Options:** Values imported from the Country table
- **Description:** Defines the country the tax is applicable to.
- **Validation:**
    - **Required**: Yes

- **State**
    - **Type:** Dropdown
    - **Label:** "State"
    - **Editable:** Yes
    - **Options:** Values imported from the State table
    - **Description:** Defines the state the tax is applicable to.
    - **Validation:**
        - **Required**: No

- **Filing Frequency**
    - **Type:** Dropdown
    - **Label:** "Filing Frequency"
    - **Editable:** Yes
    - **Options:** Monthly, Quarterly, Annually
    - **Description:** Defines how frequently the tax needs to be filed.
    - **Validation:**
        - **Required**: Yes

- **Effective Date**
    - **Type:** Text Input (Text Editor)
    - **Label:** "Effective Date"
    - **Editable:** Yes
    - **Description:** Field to specify the effective date of the tax.
    - **Validation:**

- **Required**: Yes
- **Format**: Date format (DD/MM/YYYY)

- **Is Active**
  - **Type:** Checkbox
  - **Label:** "Is Active"
  - **Editable:** Yes
  - **Default Value:** Checked (True)
  - **Behavior:**
    - **Checked**: Tax is active and applied in calculations.
    - **Unchecked**: Tax is inactive but stored in the database.
- **Submit Button**
  - **Type:** Button
  - **Label:** "Submit"
  - **Editable:** No (Standard Button)
  - **Description:** Submits the completed form to the database.
  - **Validation:**
    - Ensures all mandatory fields are validated.
  - **Behavior:**
    - **Successful Submission**: Saves data to the database.
    - **Failed Validation**: Displays error messages for corrections.
- **Clear Button**
  - **Type:** Button
  - **Label:** "Clear"
  - **Editable:** No (Standard Button)
  - **Description:** Resets all fields in the form to their default state.
  - **Behavior:**
    - Resets fields with a confirmation prompt to avoid accidental data loss.
- **Previous Button**

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Type:** Button
- **Label:** "Previous Icon"
- **Editable:** No (Standard Button)
- **Description:** Closes the Tax Creation Form and returns to the Tax table view, discarding unsaved changes.

2.2.2.5 Functional Requirements

- **Actions**
  - **Submit Button Click**
    - **Action:** Triggers the form submission if all required fields (Tax Type, Code, Name, Percentage, Taxable Base, Transaction Type, Filing Frequency, and Effective Date) contain valid values.
    - **Outcome:**
      - **Create Operation:** Saves a new tax record if it is a new entry.
      - **Update Operation:** Updates an existing tax record if the form is being used to edit a tax.
  - **Clear Button Click**
    - **Action:** Resets all fields in the form to their default state.
    - **Outcome:**
      - Prompts the user for confirmation before clearing all input fields to avoid accidental data loss.
  - Previous Button Click
    - **Action:** Discards unsaved changes and returns the user to the Tax table view.
    - **Outcome:**
      - Closes the Tax Creation Form without saving changes.
- **Validation**
  - **Mandatory Fields**
    - **Fields:** Tax Type, Code, Name, Percentage, Taxable Base, Transaction Type, Filing Frequency, and Effective Date are required fields.
    - **Behavior:**
      - Users must provide values in each mandatory field.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- Form submission is blocked if any mandatory field is left empty, with an error message indicating the missing fields.

- **Field-Specific Validation**
  - **Code:** Must be alphanumeric with a maximum of 10 characters.
  - **Percentage:** Must be numeric and within a valid range (e.g., 0-100).
  - **Effective Date:** Must follow a valid date format (e.g., DD/MM/YYYY).

- **Error Messages**
  - **Submit Attempt with Missing Values**
    - **Condition:** If one or more required fields are missing a value when the user clicks Submit.
    - **Behavior:** The system displays a specific error message for each missing field.
    - **Message Examples:**
      - "Tax Type is required" if the Tax Type field is empty.
      - "Code is required" if the Code field is empty.
      - "Percentage is required" if the Percentage field is empty.
  - **Invalid Field Values**
    - **Condition:** If a field contains invalid data (e.g., non-numeric input in the Percentage field).
    - **Behavior:** The system displays an error message specifying the issue with the invalid field.
    - **Message Examples:**
      - "Percentage must be a numeric value between 0 and 100."
      - "Effective Date must follow the DD/MM/YYYY format."
  - **Behavior on Successful Submission**
    - The form saves the data to the database and redirects the user to the Tax table view.
    - A success message is displayed:
      - Example: "Tax successfully created."
  - **Behavior on Failed Validation**
    - Form submission is blocked, and error messages are displayed to guide the user in correcting the input fields.

2.2.2.6 Data Requirements

- **Tax Type**
  - **Source:** Tax Type Table
  - **Description:** The type of tax (e.g., GST, VAT, Sales Tax) applicable to the transaction.
  - **Usage:** This field is populated from the Tax Type Table and must be selected during the form submission process.
- **Linked Taxes**
  - **Source:** Tax Table
  - **Description:** Allows the selection of other taxes linked to the current tax.
  - **Usage:** This field is populated from the Tax Table and can be edited to link other relevant taxes.
- **Withholding Tax**
  - **Source:** Tax Table (Filtered by Withholding Tax Type)
  - **Description:** Specifies whether this tax includes withholding tax, and if so, links to the appropriate withholding tax type.
  - **Usage:** Populated from the Tax Table with a filter based on the Withholding Tax category.
- **Country**
  - **Source:** Country Table
  - **Description:** Specifies the country where the tax is applicable.
  - **Usage:** This field is populated from the Country Table, and the user must select a country for the tax.
- **State**
  - **Source:** State Table
  - **Description:** Specifies the state within the selected country where the tax is applicable.
  - **Usage:** Populated from the State Table based on the selected country.

2.2.2.7 Non-Functional Requirements

None.

2.2.2.8 Configurations

None.

2.2.2.9 Flowchart

None.

2.2.2.10 Additional Notes

None.

2.2.3 Update Tax

2.2.3.1 Purpose

The Edit Tax section enables authorized users to modify existing tax records to ensure accurate and up-to-date information, supporting the ERP's compliance with evolving international tax regulations and business requirements.

2.2.3.2 User Type

Registered Users, Admins.

2.2.3.3 Screen Layout

Same Create Tax Layout.

2.2.3.4 UI Elements

Same Create Tax Layout.

2.2.3.5 Functional Requirements

- **Initiate Update Process**
  - **Description:** The update process is initiated by clicking the Edit button in the Action column of the tax table.
  - **Functionality:**
    - **Edit Button Action:**
      - Loads the selected tax record's existing details (Tax Name, Tax Type, Tax Percentage, Effective Date, and Status) into the Tax Update Form.
      - Pre-populates the form fields with the current values of the selected tax record to enable editing.

- **Expected Outcome:** The Tax Update Form is populated with the selected tax record's current data, ready for modification.

- **Modify Tax Details**
  - **Description:** Allows users to change one or more fields in the tax record, such as Tax Name, Tax Type, Tax Code, Tax Percentage, Effective Date, and Status, among others.
  - **Fields Available for Update:**
    - **Tax Name:** Modify the name of the tax.
    - **Tax Type:** Update the type of tax (e.g., VAT, GST, Service Tax).
    - **Tax Code:** Update the unique identifier for the tax.
    - **Taxable Base:** Specify whether the taxable amount is based on the Net Amount or Gross Amount.
    - **Tax Applicability:** Indicate whether the tax is applicable for Interstate, Intrastate, or international transactions.
    - **Transaction Type:** Specify the type of transaction (e.g., Service, Job, Purchase, or Sale).
    - **Linked Taxes:** Specify related taxes (e.g., CGST, SGST).
    - **Calculation Method:** Define whether the tax calculation is Inclusive or Exclusive (e.g., CGST Including).
    - **Rounding Policy:** Set the rounding policy (e.g., Round Up, Round Down, or Nearest Value).
    - **Withholding Tax:** Specify withholding tax details (e.g., TDS).
    - **Country:** Specify the country where the tax is applicable.
    - **Filing Frequency:** Define the frequency of tax filing (e.g., Monthly, Quarterly).
    - **Tax Percentage:** Change the tax rate (e.g., 5%, 12%).
    - **Effective Date:** Update the date when the tax becomes effective.
    - **Status:** Set the active status of the tax to either "Active" or "De-active."
  - **Expected Outcome:** Users can update all editable fields as required before saving the record.

- Form Validation (Update Mode)
  - **Purpose:**
    Ensures that updated information meets validation requirements before submission.
  - **Validation Criteria:**
    - **Mandatory Fields:** Tax Name, Tax Type, and Tax Percentage must not be empty.
    - **Specific Field Validation:**

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

        - **Tax Name:** Up to 50 characters, allows letters, spaces, and certain special characters.
        - **Tax Percentage:** Must be a numeric value between 0 and 100.
    - **Error Handling:** If validation fails, display specific error messages to guide the user (e.g., "Tax Name must be unique").
  - **Expected Outcome:** All required fields are validated, and any errors are shown to the user before submitting the update.

- **Submit Updated Data**
  - **Description:** Saves the modified data to the database upon successful validation.
  - **Functionality:**
    - **Submit Button:**
      - Saves the updated tax data to the database and refreshes the table to reflect changes.
      - Displays a success message to confirm the update (e.g., "Tax details have been successfully updated").
  - **Expected Outcome:** The updated tax information is stored in the database, and the user receives confirmation of the successful update.

- **Cancel Update**
  - **Purpose:** Allows users to discard changes if they choose not to update the tax record.
  - **Functionality:**
    - **Cancel Option:** Users can cancel the update operation by closing the form or clicking a "Previous" button.
  - **Expected Outcome:** The form closes without saving changes, leaving the original tax record intact.

- **Confirmation Message**
  - **Purpose:** Notifies the user of the successful completion of the update operation.
  - **Functionality:**
    - **Display Message:** Upon a successful update, display a confirmation message (e.g., "Tax details have been successfully updated").
  - **Expected Outcome:** The confirmation message assures the user that the changes were saved.

2.2.3.6 Data Requirements

None.

2.2.3.7 Non-Functional Requirements

None.

2.2.3.8 Configurations

None.

2.2.3.9 Flowchart

None.

2.2.3.10 Additional Notes

None.

2.2.4 Remove Tax

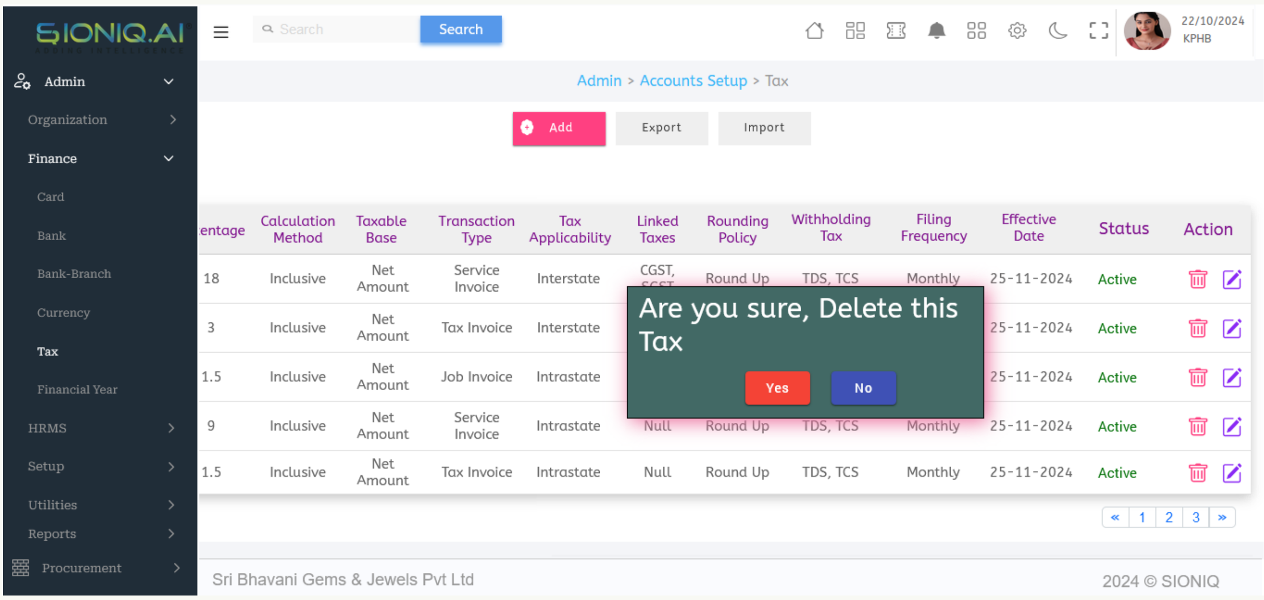
2.2.4.1 Purpose

This section allows for the safe removal of a tax from the ERP system, typically when the tax is no longer applicable to business operations. The action may be restricted if there are dependencies, such as existing records in other modules linked to that tax, and safeguards are in place to ensure data integrity across the system.

2.2.4.2 User Type

Registered Users, Admins.

2.2.4.3 Screen Layout



2.2.4.4 UI Elements

None.

2.2.4.5 Functional Requirements

- **Access Control**
  - **Purpose:** Ensure that only authorized users can perform the delete operation**.**
  - **Functionality:**
    - **Delete Permission:** The Delete button is only accessible to users who have been granted delete permissions in the Configuration Section.
- **Initiate Delete Action**
  - **Description:** The delete process begins when an authorized user clicks the Delete button in the Action column of the tax table.
  - **Functionality:**
    - **Delete Button Action:**
      - Prompts a delete confirmation message to prevent accidental deletion.
- **Delete Confirmation Prompt**
  - **Purpose:** Provide a confirmation prompt to ensure the user wants to proceed with deletion.
  - **Functionality:**
    - **Confirmation Message:** When the Delete button is clicked, display a warning message:
      - " Are you sure you want to delete this tax?"
    - **Options:**
      - **Yes:** Confirms the deletion and proceeds with removing the tax record.
      - **No:** Cancels the delete action without any changes to the data.
  - **Expected Outcome:** The tax record is deleted only if the user confirms by selecting "Yes."
- **Delete Execution**
  - **Description:** Once confirmed, the system proceeds to update the tax record in the database to mark it as inactive (status = 0).
  - **Functionality:**
    - **Data Removal:** Updates the selected tax record in the database with status = 0.
    - **Dependencies Check:**

- Before deletion, the system checks for any dependencies or associations (e.g., if the tax is linked to any active tax, invoices, transactions, or records).
- If dependencies exist, display a message (e.g., "Cannot delete tax as it is associated with active records.") and halt the delete process.

- **Success Message**
  - **Purpose:** Inform the user of the successful completion of the delete action.
  - **Functionality:**
    - **Display Message:** Upon successful deletion, display a success message such as "Tax has been successfully deleted."
  - **Expected Outcome:** The user sees a confirmation that the tax record was deleted.
- **Refresh Table View**
  - **Description:** Automatically updates the tax table view to reflect the changes made by the deletion.
  - **Functionality:**
    - **Refreshes the Table View:** Removes the deleted tax record from the list.
    - **Pagination Update:** Adjusts the pagination if necessary, ensuring the remaining records are displayed correctly.

2.2.4.6 Data Requirements

None.

2.2.4.7 Non-Functional Requirements

None.

2.2.4.8 Configurations

None.

2.2.4.9 Flowchart

None.

2.2.4.10 Additional Notes

None.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.3 Financial Year

The Financial Year section in the ERP system is designed to centralize and streamline the management of fiscal periods across various companies. This section allows users to create, edit, delete, and view financial year details, organized by parent companies, ensuring clear and accessible financial data.

2.3.1 Data View

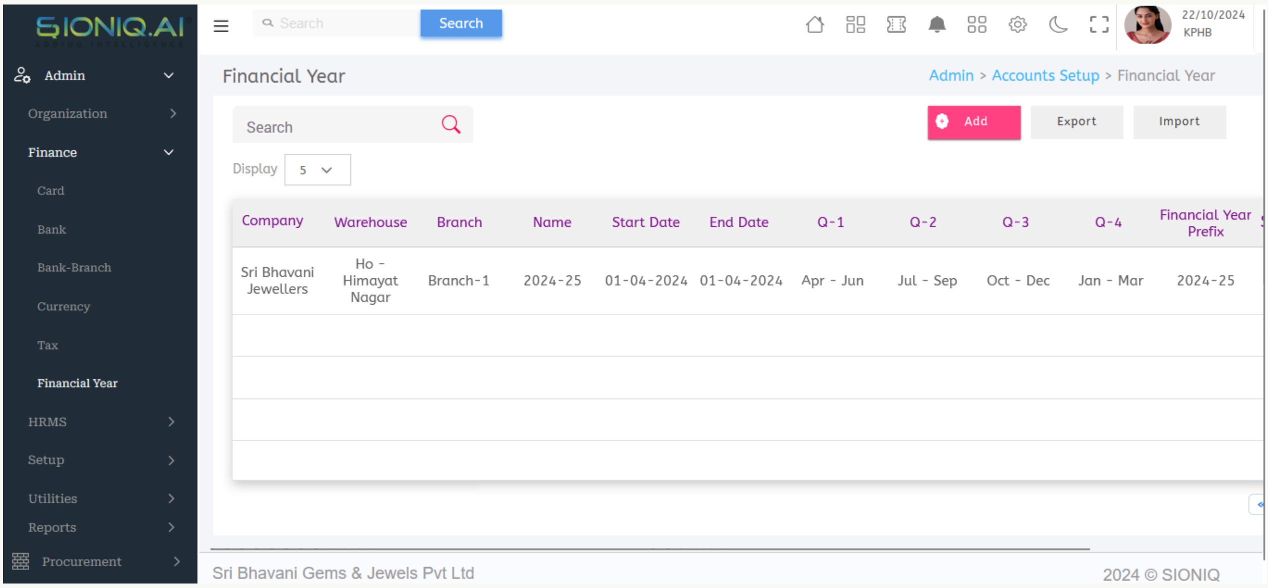
2.3.1.1 Purpose

The Financial Year Data View provides a comprehensive overview of all financial years registered within the ERP system. This view displays key details, such as fiscal year start and end dates, status (active or inactive), and associated companies, in an organized, tabular format.

2.3.1.2 User Type

Registered Users, Admins.

2.3.1.3 Screen Layout



2.3.1.4 UI Elements

- **Search Option**
  - **Description**: A search bar positioned at the top of the data view section, enabling filtering of financial year records.
  - **Functionality**:
    - Allows users to search by financial year name, company, warehouse, or branch association.
    - Dynamically updates the displayed table to show results matching the query.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Display Dropdown**
  - **Description**: A dropdown to control the number of rows shown in the table at a time.
  - **Options**: Common choices (e.g., 5, 10, 25, 50, 100 rows per page).
  - **Functionality**:
    - Enables users to select the number of financial year records shown per page.
    - Updates the table view instantly based on the chosen value.
- **Buttons**
  - **Add Financial Year Button**
    - **Label**: "New"
    - **Functionality**: Opens the Financial Year Creation Form for adding a new financial year record.
  - **Export Button**
    - **Label**: "Export" (or Icon button as specified)
    - **Functionality**: e displayed financial year data in formats such as Excel.
    - **Export Options**:
      - Includes all data or filtered data based on the current table view and search filter.
  - **Import Button**
    - **Label**: "Import" (or Icon button as specified).
    - **Functionality**:
      - Redirects users to an Import Form page for batch uploading of financial year data.
- **Tabing View**
  - **Description**: Displays a structured table format for all created financial years, enabling users to review and manage records effectively .
  - **Columns**:
    - **Company**: Displays the name of the company associated with the financial year.
    - **Warehouse**: Shows the warehouse associated with the financial year.
    - **Branch**: Indicates the branch linked to the financial year.
    - **Name**: Displays the financial year's designated name.
    - **Start Date**: Shows the starting date of the financial year.
    - **End Date**: Indicates the ending date of the financial year.
    - **Quarter 1**: Displays the start and end dates for the first quarter.
    - **Quarter 2**: Displays the start and end dates for the second quarter.
    - **Quarter 3**: Displays the start and end dates for the third quarter.
    - **Quarter 4**: Displays the start and end dates for the fourth quarter.
    - **Financial Year Prefix**: Shows the prefix associated with the financial year for identification purposes.
    - **Status**: Displays whether the financial year is active or inactive.
      - **Active**: Indicates that the financial year is currently in use.
      - **Inactive**: Indicates that the financial year is not in use.

- **Action**:
  - **Edit**: Opens the financial year form pre-filled with existing data for modifications.
  - **Delete**: Deletes the financial year record (permissions-based).
- **Functionality**: Prompts a confirmation message before deletion to prevent accidental changes.

- **Paging for Table View**
  - **Description**: Provides pagination controls at the bottom of the table, facilitating navigation through large volumes of financial year records.
  - **Functionality**:
    - **Navigation Buttons**:
      - **Previous**: Moves to the preceding page of records.
    - **Page Numbers**:
      - Displays clickable page numbers, allowing users to jump directly to a specific page.
      - Highlights the current page for clarity.

2.3.1.5 Functional Requirements

Refer to the above **UI Elements** section for specific functionalities associated with each element.

2.3.1.6 Data Requirements

None.

2.3.1.7 Non-Functional Requirements

None.

2.3.1.8 Configurations

- **Functionality**:
  - **Permission Settings**: Defines the visibility and accessibility of various buttons and actions based on user roles or permissions. This ensures that only authorized users can access specific functionalities like adding, editing, deleting, or exporting financial year records.
- **Available Permissions**:
  - **Add Financial Year Permission**:
    - **Description**: Grants users the ability to access and use the "Add Financial Year" button to create new financial year records.
  - **Export Permission**:
    - **Description**: Grants users the ability to access the Export button to download financial year data in formats such as Excel or CSV for reporting or analysis purposes.
  - **Import Permission**:

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

        - Description: Grants users the ability to access the Import button to upload financial year data from external files into the system for batch processing.
    - **Edit Permission**:
        - **Description**: Allows users to access the "Edit" button in the Action column to modify existing financial year records. This is essential for updating details like start/end dates, quarters, and status.
    - **Delete Permission**:
        - Description: Allows users to access the "Delete" button in the Action column to remove financial year records from the system, subject to validation and confirmation.
- **Permission Assignment**:
    - **Role-Based Permissions**: Admins or authorized users can assign specific permissions to individual users or roles. This can be done via a permissions management interface, where the admin can select users or user roles and assign permissions based on business needs.
    - **Granular Control**: Permissions can be assigned at the financial year level or on a broader scope for the entire ERP system. This ensures the proper level of access control across different areas of the system.

2.3.1.9 Flowchart

None.

2.3.1.10 Additional Notes

None.

2.3.2 Create Financial Year

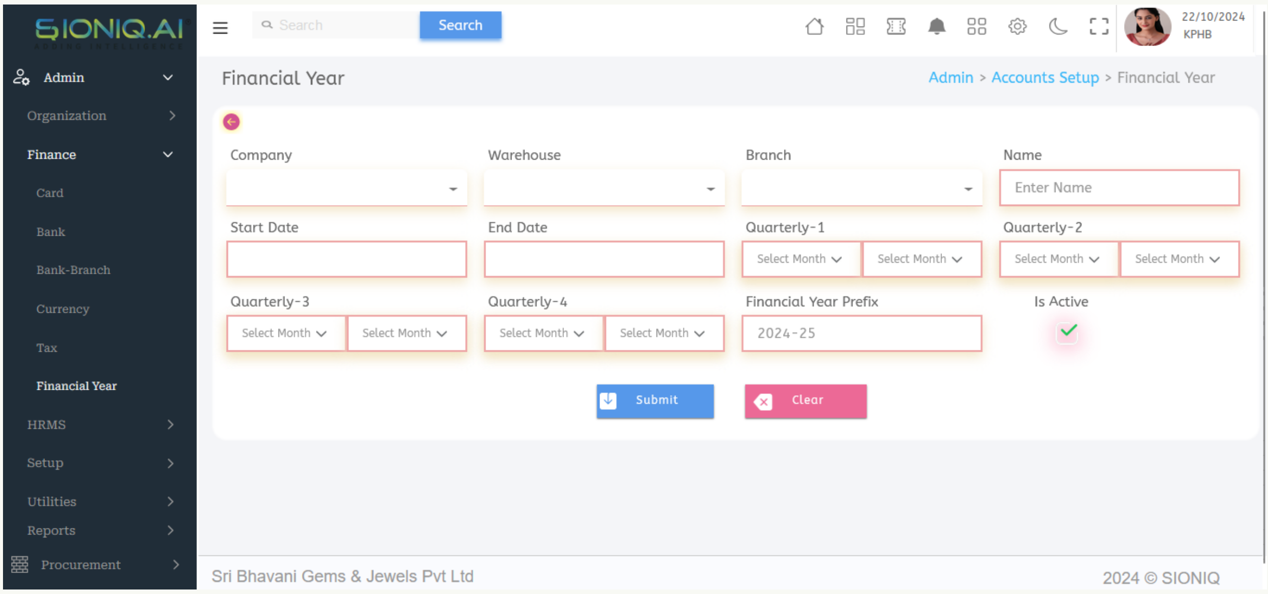
2.3.2.1 Purpose

This section allows users to add a new financial year to the ERP system. Users can enter key information such as the financial year name, start date, end date, quarters, and associated company details. This process ensures structured and consistent data entry for operational and reporting purposes, supporting effective financial planning and decision-making.

2.3.2.2 User Type

Registered Users, Admins.

2.3.2.3 Screen Layout



2.3.2.4 UI Elements

- **Company**
  - **Type:** Multi Dropdown
  - **Label:** "Company"
  - **Editable:** Yes
  - **Options:** Values imported from the Company table.
  - **Description:** Allows selection of one or more companies to associate with the financial year.
  - **Validation:**
    - **Required:** Yes
- **Warehouse**
  - **Type:** Multi Dropdown
  - **Label:** "Warehouse"
  - **Editable:** Yes
  - **Options:** Values imported from the Warehouse table.
  - **Description:** Allows selection of one or more warehouses linked to the financial year.
  - **Validation:**
    - **Required:** No

- **Branch**
  - **Type:** Multi Dropdown
  - **Label:** "Branch"
  - **Editable:** Yes
  - **Options:** Values imported from the Branch table.
  - **Description:** Allows selection of one or more branches associated with the financial year.
  - **Validation:**
    - **Required:** No
- **Name**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Name"
  - **Editable:** Yes
  - **Description:** Defines the name of the financial year (e.g., "FY2024").
  - **Validation:**
    - **Required:** Yes
    - **Character Limit:** 20 characters.
    - **Allowed Characters:** Alphanumeric and spaces.
- **Start Date**
  - **Type:** Date Picker
  - **Label:** "Start Date"
  - **Editable:** Yes
  - **Description:** Specifies the starting date of the financial year.
  - **Validation:**
    - **Required:** Yes
    - **Format:** Valid date format (e.g., DD/MM/YYYY).
- **End Date**
  - **Type:** Date Picker
  - **Label:** "End Date"

- **Editable:** Yes
- **Description:** Specifies the ending date of the financial year.
- **Validation:**
  - **Required:** Yes
  - **Format:** Valid date format (e.g., DD/MM/YYYY).
  - **Dependency:** Must be greater than the Start Date.

○ **Quarterly-1**

- **Type:** Month Range Selector
- **Label:** "Quarterly-1"
- **Editable:** Yes
- **Description:** Allows selection of the month range for the first quarter of the financial year.
- **Validation:**
  - **Required:** No

○ **Quarterly-2**

- **Type:** Month Range Selector
- **Label:** "Quarterly-2"
- **Editable:** Yes
- **Description:** Allows selection of the month range for the second quarter of the financial year.
- **Validation:**
  - **Required:** No

○ **Quarterly-3**

- **Type:** Month Range Selector
- **Label:** "Quarterly-3"
- **Editable:** Yes
- **Description:** Allows selection of the month range for the third quarter of the financial year.
- **Validation:**
  - **Required:** No

○ **Quarterly-4**

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Type:** Month Range Selector
- **Label:** "Quarterly-4"
- **Editable:** Yes
- **Description:** Allows selection of the month range for the fourth quarter of the financial year.
- **Validation:**
  - **Required:** No

- **Financial Year Prefix**
  - **Type:** Text Input (Text Editor)
  - **Label:** "Financial Year Prefix"
  - **Editable:** Yes
  - **Description:** Specifies the prefix for the financial year (e.g., "FY").
  - **Validation:**
    - **Required:** No
    - **Character Limit:** 10 characters.
    - **Allowed Characters:** Alphanumeric.

- **Is Active**
  - **Type:** Checkbox
  - **Label:** "Is Active"
  - **Editable:** Yes
  - **Default Value:** Checked (True)
  - **Behavior:**
    - **Checked:** The financial year is active and used in calculations and reporting.
    - **Unchecked:** The financial year is inactive but stored in the database.

- **Submit Button**
  - **Type:** Button
  - **Label:** "Submit"
  - **Editable:** No (Standard Button)
  - **Description:** Submits the completed form to the database.

- **Validation:**
  - Ensures all mandatory fields are validated.
- **Behavior:**
  - **Successful Submission:** Saves data to the database.
  - **Failed Validation:** Displays error messages for corrections.

- **Clear Button**
  - **Type:** Button
  - **Label:** "Clear"
  - **Editable:** No (Standard Button)
  - **Description:** Resets all fields in the form to their default state.
  - **Behavior:**
    - Resets fields with a confirmation prompt to avoid accidental data loss.

- **Previous Button**
  - **Type:** Button
  - **Label:** "Previous Icon"
  - **Editable:** No (Standard Button)
  - **Description:** Closes the Financial Year Creation Form and returns to the Financial Year table view, discarding unsaved changes.

2.3.2.5 Functional Requirements

Refer to the above **UI Elements** section for specific functionalities associated with each element.

2.3.2.6 Data Requirements

- **Data Sources and Field Mapping**
  - **Company**
    - **Source:** Company Table.
    - **Fields Required:**
      - Company Name: The name of the company.
      - Company ID: The unique identifier for the company.
    - **Usage:**

        - Links the financial year to specific companies for accurate financial reporting and operations.
        - Ensures multi-company support within the ERP system.
- **Warehouse**
    - **Source:** Warehouse Table.
    - **Fields Required:**
        - Warehouse Name: The name of the warehouse.
        - Warehouse ID: The unique identifier for the warehouse.
    - **Usage:**
        - Associates the financial year with relevant warehouses to manage inventory and operations.
- **Branch**
    - **Source:** Branch Table.
    - **Fields Required:**
        - Branch Name: The name of the branch.
        - Branch ID: The unique identifier for the branch.
    - **Usage:**
        - Links the financial year to branches for operational and financial tracking.

○ **Mandatory Fields**

- **Company**: Selection of at least one company is required to proceed with financial year creation.
- **Warehouse**: Selection of at least one warehouse is required to ensure proper association.
- **Branch**: Selection of at least one branch is mandatory to complete the financial year setup.

○ **Validation and Data Usage**

- **Validation Rules:**
    - Each mandatory field must be selected before submission.
    - Dropdowns are populated dynamically from respective tables to ensure accurate and updated options.
- **Data Usage:**
    - Ensures proper mapping of financial years to organizational entities like companies, warehouses, and branches for reporting and operational accuracy.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- Facilitates seamless data integration across the ERP system.

- **Optional Data**
  - Additional fields like Name, Start Date, End Date, Quarterly ranges, and Financial Year Prefix can be used for descriptive and operational purposes but do not affect the core linkage to Company, Warehouse, and Branch.

2.3.2.7 Non-Functional Requirements

None.

2.3.2.8 Configurations

None.

2.3.2.9 Flowchart

None.

2.3.2.10 Additional Notes

None.

2.3.3 Update Financial Year

2.3.3.1 Purpose

The "Update Financial Year" section allows authorized users to modify existing financial year records in the ERP system. This ensures that the financial year data remains accurate, up-to-date, and in line with evolving business needs and compliance requirements. Users can edit details such as the financial year name, start and end dates, quarter information, and associated company, warehouse, and branch data. The update process helps maintain the integrity of the ERP system by ensuring that all financial year records reflect current business conditions and support seamless financial reporting and operations.

2.3.3.2 User Type

Registered Users, Admins.

2.3.3.3 Screen Layout

Same Create Brand Layout.

2.3.3.4 UI Elements

Same Create Brand Layout.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.3.3.5 Functional Requirements

- **Initiate Update Process**
  - **Description:** The update process is initiated by clicking the Edit button in the Action column of the Financial Year table.
  - **Functionality:**
    - **Edit Button Action:**
      - Loads the selected Financial Year's existing details (Start Date, End Date, Fiscal Year Code, Status, and other relevant fields) into the Financial Year Update Form.
      - Pre-populates the form fields with the current values of the selected record to enable editing.
  - **Expected Outcome:** The Financial Year Update Form is populated with the selected financial year's current data, ready for modification.
- **Modify Financial Year Details**
  - **Description:** Allows users to change one or more fields in the financial year record, ensuring alignment with organizational requirements.
  - **Fields Available for Update:**
    - **Start Date:** Update the starting date of the financial year.
    - **End Date:** Update the ending date of the financial year.
    - **Fiscal Year Code:** Modify the unique identifier for the financial year.
    - **Description:** Update additional notes or descriptions about the financial year.
    - **Period Status:** Change the status of the financial year (e.g., "Open," "Closed").
    - **Status:** Set the active status of the financial year to either "Active" or "De-active."
  - **Expected Outcome:** Users can update all editable fields as required before saving the record.
- **Form Validation (Update Mode)**
  - **Purpose:** Ensures that updated information meets validation requirements before submission.
  - **Validation Criteria:**
    - **Mandatory Fields:** Start Date, End Date, and Fiscal Year Code must not be empty.
    - **Specific Field Validation:**
      - **Start Date and End Date:** Must be valid dates, with the End Date occurring after the Start Date.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

        - **Fiscal Year Code:** Unique code with up to 10 alphanumeric characters.
    - **Error Handling:** If validation fails, display specific error messages to guide the user (e.g., "Fiscal Year Code must be unique").
- **Expected Outcome:** All required fields are validated, and any errors are shown to the user before submitting the update.

- **Submit Updated Data**
  - **Description:** Saves the modified financial year data to the database upon successful validation.
  - **Functionality:**
    - **Submit Button:**
      - Saves the updated financial year data to the database and refreshes the table to reflect changes.
      - Displays a success message to confirm the update (e.g., "Financial year details have been successfully updated").
  - **Expected Outcome:** The updated financial year information is stored in the database, and the user receives confirmation of the successful update.

- **Cancel Update**
  - **Purpose:** Allows users to discard changes if they choose not to update the financial year record.
  - **Functionality:**
    - **Cancel Option:**
      - Users can cancel the update operation by closing the form or clicking a "Previous" button.
    - **Expected Outcome:** The form closes without saving changes, leaving the original financial year record intact.

- **Confirmation Message**
  - **Purpose:** Notifies the user of the successful completion of the update operation.
  - **Functionality:**
    - **Display Message:**
      - Upon a successful update, display a confirmation message (e.g., "Financial year details have been successfully updated").
  - **Expected Outcome:** The confirmation message assures the user that the changes were saved.

2.3.3.6 Data Requirements

None.

2.3.3.7 Non-Functional Requirements

None.

2.3.3.8 Configurations

None.

2.3.3.9 Flowchart

None.

2.3.3.10 Additional Notes

None.

2.3.4 Remove Financial Year

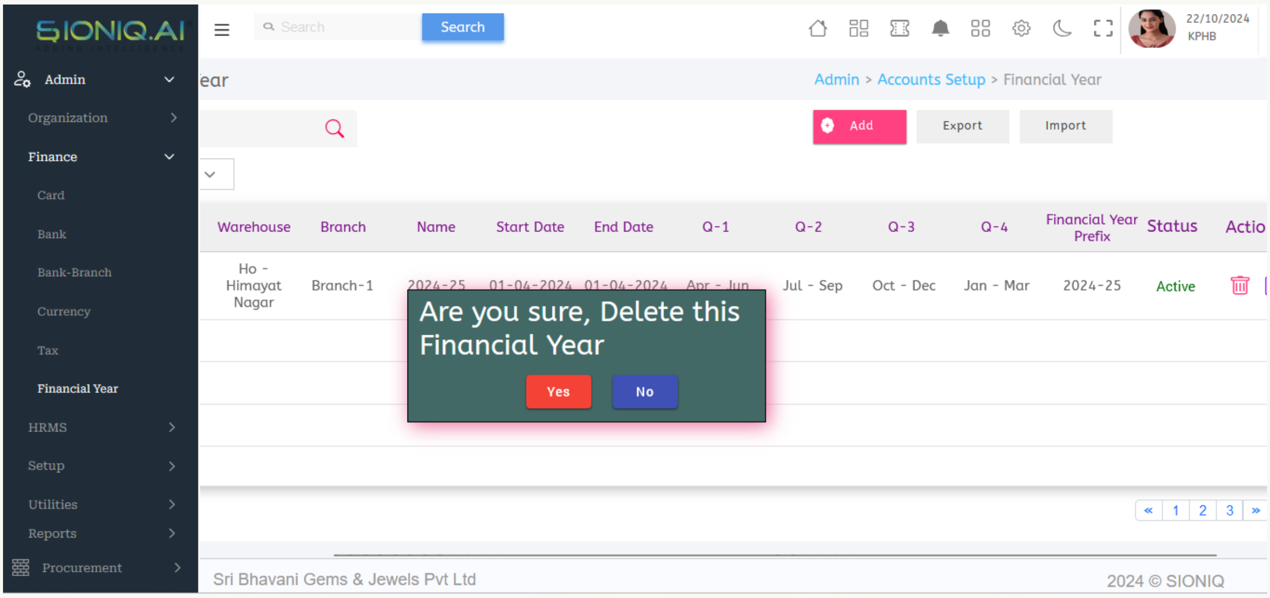
2.3.4.1 Purpose

This section allows for the safe removal of a brand from the ERP system, typically when the brand is no longer relevant to business operations. The action may be restricted if there are dependencies, such as existing records in other modules linked to that brand, and safeguards are in place to ensure data integrity across the system.

2.3.4.2 User Type

Registered Users, Admins.

2.3.4.3 Screen Layout



2.3.4.4 UI Elements

None.

2.3.4.5 Functional Requirements

- **Access Control**
  - **Purpose:** Ensure that only authorized users can perform the delete operation.
  - **Functionality:**
    - **Delete Permission:** The Delete button is only accessible to users who have been granted delete permissions in the Configuration Section.
- **Initiate Delete Action**
  - **Description:** The delete process begins when an authorized user clicks the Delete button in the Action column of the financial year table.
  - **Functionality:**
    - **Delete Button Action:**
      - Prompts a delete confirmation message to prevent accidental deletion.
- **Delete Confirmation Prompt**
  - **Purpose:** Provide a confirmation prompt to ensure the user wants to proceed with deletion.
  - **Functionality:**
    - **Confirmation Message:** When the Delete button is clicked, display a warning message:
      - "Are you sure you want to delete this financial year?"
    - **Options:**
      - **Yes:** Confirms the deletion and proceeds with removing the financial year record.
      - **No:** Cancels the delete action without any changes to the data.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Expected Outcome:** The financial year record is deleted only if the user confirms by selecting "Yes."

- **Delete Execution**
  - **Description:** Once confirmed, the system proceeds to update "Status=0" int the financial year record from the database.
  - **Functionality:**
    - **Data Removal:** Updated the selected brand record from the database [status=0].
    - **Dependencies Check:**
      - Before deletion, the system checks for any dependencies or associations.
      - If dependencies exist, display a message (e.g., "Cannot delete financial year as it has associated financial year.") and halt the delete process.
- **Success Message**
  - **Purpose:** Inform the user of the successful completion of the delete action.
  - **Functionality:**
    - **Display Message:** Upon successful deletion, display a success message such as " financial year has been successfully deleted."
  - **Expected Outcome:** The user sees a confirmation that the record was deleted.
- **Refresh Table View**
  - **Description:** Automatically updates the financial year table view to reflect the changes made by the deletion.
  - **Functionality:**
    - Refreshes the table view, removing the deleted financial year record from the list.
    - Pagination Update: Adjusts the pagination if necessary, ensuring the remaining records are displayed correctly.

2.3.4.6 Data Requirements

None.

2.3.4.7 Non-Functional Requirements

None.

2.3.4.8 Configurations

None.

2.3.4.9 Flowchart

None.

2.3.4.10 Additional Notes

None.

2.4 Bank

The Bank Form in the Jewellery ERP system is designed to streamline the entry, management, and tracking of bank-related data. This form allows users to create, edit, delete, and view details of various banks. By centralizing bank information, the form ensures consistency in financial transactions, account management, and reconciliation across multiple modules. This supports efficient operations, accurate financial reporting, and compliance with global banking standards, enabling seamless management of financial processes within the ERP system.

2.4.1 Data View

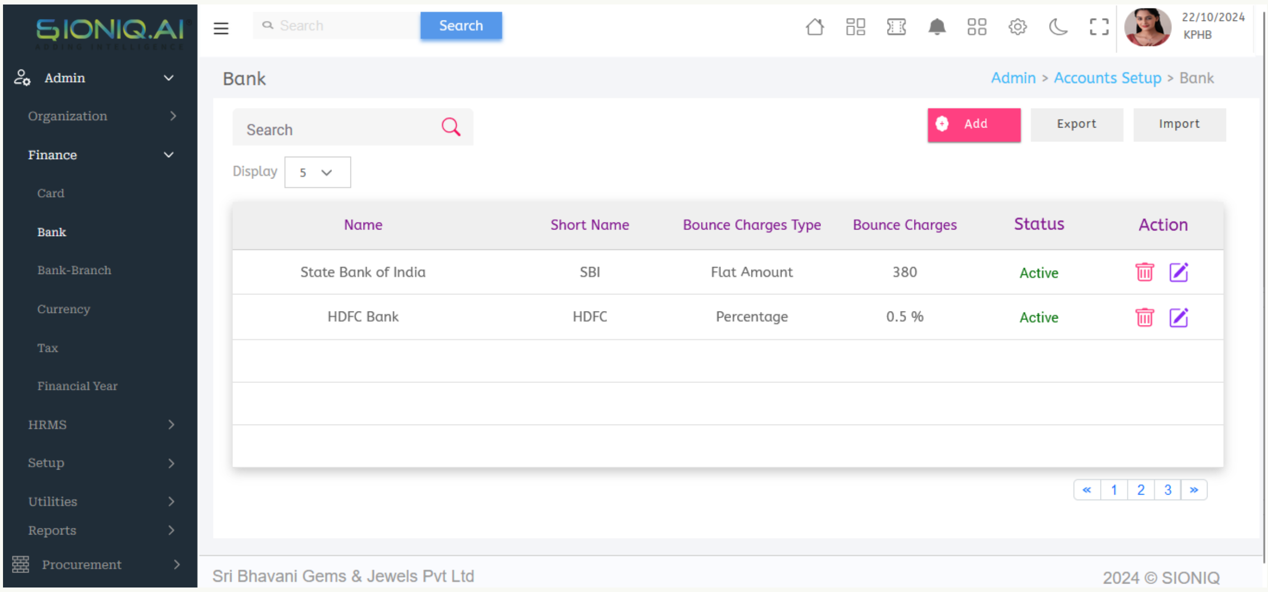
2.4.1.1 Purpose

The Warehouse Table Data View offers a detailed and organized overview of all warehouses registered in the ERP system, providing essential information about each warehouse's location, capacity, and operational status. This view enables users to quickly locate and assess individual warehouses, including details like address, city, state, country, and active status. The table supports filtering options, allowing users to refine their search by warehouse location or status for efficient navigation. With action buttons for editing and deleting entries, users can easily manage warehouse records, keeping the system updated to reflect current logistics and inventory needs.

2.4.1.2 User Type

Registered Users, Admins.

2.4.1.3 Screen Layout



2.4.1.4 UI Elements

- **Search Option**
  - **Description**: A search bar located at the top of the data view section to filter the bank records displayed in the table.
  - **Functionality**:
    - Enables users to search by attributes such as Name, Short Name.
    - Dynamically updates the displayed table to show only results that match the search criteria.
- **Display Dropdown**
  - **Description**: A dropdown to control the number of rows shown in the table at a time.
  - **Options**: Common choices such as 5, 10, 25, 50, or 100 rows per page.
  - **Functionality**:
    - Allows users to set how many bank records are displayed per page.
    - The table updates immediately based on the selected number of rows.
- **Button**
  - **Add Bank Button**
    - **Label**: "New"
    - **Functionality**: Opens the Bank Creation Form to add a new bank record to the system.
  - **Export Button**
    - **Label**: "Export" (or icon button as specified)
    - **Functionality**:
      - Allows users to export the displayed bank data in various file formats, such as Excel.
      - Exports either all or filtered data based on the current view and search filter.
  - **Import Button**
    - **Label**: "Import" (or icon button as specified)
    - **Functionality**:
      - Redirects users to the Import Form page, where bank data can be imported from external files.
- **Tabular View**
  - **Description**: Displays a structured view of all registered bank.
  - **Columns**:

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Name:** Displays the full name of each bank.
- **Short Name:** Shows the abbreviated name for quick reference.
- **Bounce Charges Type:** Indicates the type of bounce charges applicable.
- **Bounce Charges:** Displays the charges applied for bounced transactions.
- **Status**: Displays the active status of each bank, with values such as "Active" or "Inactive."
    - **Status**:
    - "Active" if the checkbox is checked in the form.
    - "Inactive" if the checkbox is unchecked in the form.
- **Action**:
    - **Edit**: Opens the bank Creation Form populated with the bank's details for editing.
    - **Delete**: Triggers a delete confirmation prompt to remove the selected bank record from the system (permission-based).

- **Paging for Table View**
    - **Description**: Includes pagination controls at the bottom of the table to navigate through multiple pages of bank records.
    - **Functionality**:
        - Provides navigation buttons (e.g., "Previous" and "Next") to move between pages.
        - Shows page numbers, allowing users to jump directly to a specific page.
        - Updates the displayed bank records based on the selected page, making it easy to manage large datasets.

2.4.1.5 Functional Requirements

Refer to the above **UI Elements** section for specific functionalities associated with each element.

2.4.1.6 Data Requirements

None.

2.4.1.7 Non-Functional Requirements

None.

2.4.1.8 Configurations

- **Functionality**:
    - **Permission Settings**: Controls the visibility and accessibility of buttons and actions based on user roles or permissions.

- **Available Permissions**:
  - **Add Bank Permission**: Grants access to the "Add bank" button.
  - **Export Permission**: Grants access to the Export button for exporting data.
  - **Import Permission**: Grants access to the Import button for importing data.
  - **Edit Permission**: Allows users to access the Edit button in the Action column to modify records.
  - **Delete Permission**: Allows users to access the Delete button in the Action column to delete records.
- **Permission Assignment**: Admins or authorized users can assign permissions to individual users or user roles.

2.4.1.9 Flowchart

None.

2.4.1.10 Additional Notes

None.

2.4.2 Create Bank

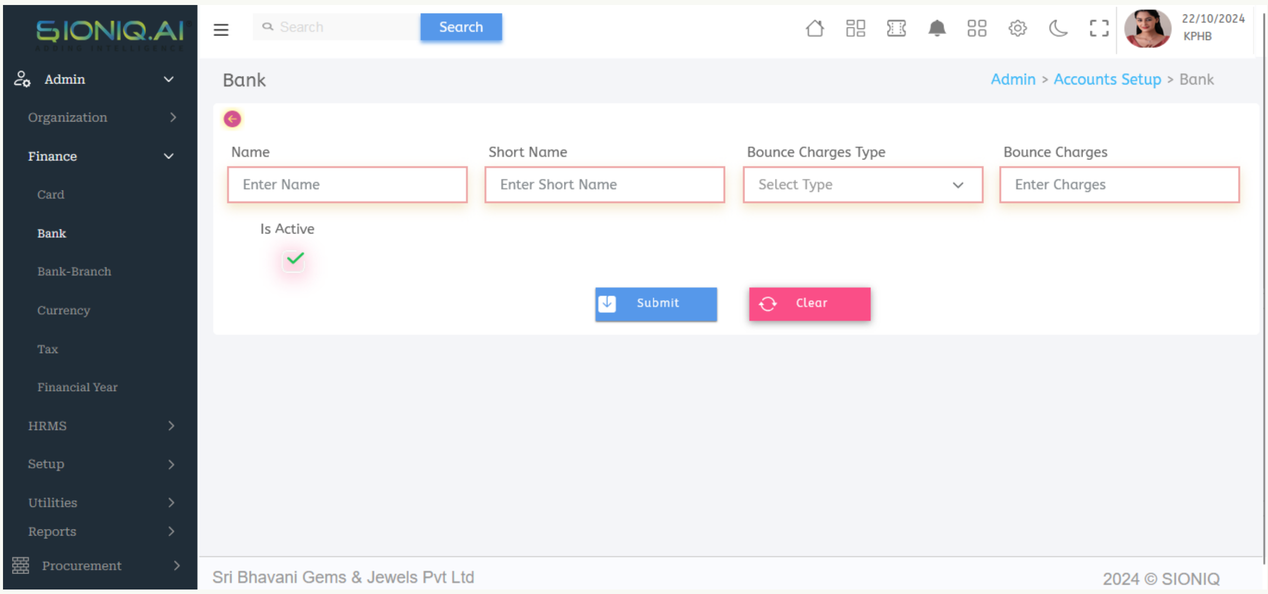
2.4.2.1 Purpose

This section allows users to add a new bank to the ERP system. Users can enter key information such as the bank name, short name, bounce charges type, bounce charges, and operational status. This process ensures consistent data entry for financial management and reporting purposes.

2.4.2.2 User Type

Registered Users, Admins.

2.4.2.3 Screen Layout



Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.4.2.4 UI Elements

- **Bank Name**
  - **Type**: Text Input (Text Editor)
  - **Label**: “Name”
  - **Editable**: Yes
  - **Description**: Input field for entering the bank full name.
  - **Validation**:
    - **Required**: Yes
    - **Character Limit**: 50 characters
    - **Allowed Characters**: Letters, spaces, and special characters (e.g., apostrophes).
- **Bank Short Name**
  - **Type**: Text Input (Text Editor)
  - **Label**: “Short Name”
  - **Editable**: Yes
  - **Description**: Input for an abbreviated name of the bank.
  - **Validation**: Required, limited to 10 characters.
    - **Required**: Yes
    - **Character Limit**: 10 characters
    - **Allowed Characters**: Uppercase letters only (e.g., HDFC)
- **Bounce Charges Type**
  - **Type**: Dropdown
  - **Label**: Bounce Charges Type
  - **Editable**: Yes
  - **Description**: Select the bounce Charges Type used in transactions related to this bank.
  - **Validation**: Required
  - **Options**: Percentage, Flat Amount.
- **Bounce Charges**
  - **Type**: Text Input (Text Editor)
  - **Label**: Bounce Charges

- **Editable**: Yes
- **Description**: bounce Charges used in transactions related to this bank.
- **Validation**: Required, limited to 5 characters.
  - **Required**: Yes
  - **Character Limit**: 5 characters
  - **Allowed Characters**: Numbers only.

- **Status**
  - **Type**: Checkbox
  - **Label**: Is Active
  - **Editable**: Yes
  - **Default Value**: Checked (True)
  - **Behavior**:
    - **Checked**: The bank is active and visible in the system.
    - **Unchecked**: The bank is inactive but retained in the database.

- **Submit Button**
  - **Type**: Button
  - **Label**: Submit
  - **Editable**: No (Standard Button)
  - **Description**: A button that submits the filled form details to the database.
  - **Validation**:
    - Ensures all required fields are completed and validated.
  - **Behavior**:
    - If validation is successful, the form data is submitted to the database.
    - If validation fails, displays an error message indicating the fields that require correction.

- **Clear Button**
  - **Type**: Button
  - **Label**: “Clear”
  - **Editable**: No (Standard Button)
  - **Description**: Resets all form fields to default values.

- **Validation**:
  - Not required; clearing the form does not require validation checks.
- **Behavior**:
  - Reset Form Fields: Clears all entered data in the form fields, resetting them to their default values.
  - Confirmation Prompt: Prompts the user for confirmation to avoid accidental data loss.

○ **Previous Button**

- **Type**: Button
- **Label**: "Previous Icon"
- **Editable**: No (Standard Button)
- **Description**: Closes the Bank Creation Form and returns the user to the main Bank table view, discarding any unsaved changes.
- **Behavior**: Cancels any unsaved changes and navigates back.
- **Validation**:
  - **Required**: No

2.4.2.5 Functional Requirements

○ **Actions**

- **Submit Button Click**
  - **Action**: Triggers form submission if all required fields (Bank Name, Bank Short Name) contain valid values.
  - **Outcome**:
    - **Create Operation**: If it's a new entry, saves the new Bank information to the database.
    - **Update Operation**: If editing an existing entry, updates the bank information.

○ **Validation**

- **Mandatory Fields**
  - **Fields**: Bank Name, Bank Short Name, are required fields.
  - **Behavior**:
    - Users must enter valid values in each mandatory input field.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- If any mandatory field is left empty, the form submission is blocked, and an error message is displayed (details in "Error Messages" section).

- **Field Format Validation**:
  - **Bank Short Name**: Maximum 10 characters, letters only.
  - **Bounce Charges**: Numeric values only, maximum 5 digits.

- **Error Messages**
  - **Submit Attempt with Missing Values**
    - **Condition**: If one or more required fields are missing a value when the user clicks Submit.
    - **Behavior**: The system displays specific error messages indicating which field(s) are missing values.
    - **Message Examples**:
      - "Bank Name is required" if the Bank Name field is empty.
      - "Bank Short Name is required" if the Bank Short Name field is empty.
  - **Validation Errors**
    - **Condition**: If any field does not meet the required format or constraints.
    - **Behavior**: The system displays an error message specifying the field and the expected input.
    - **Message Examples**:
      - "Short Name cannot exceed 10 characters."
      - "Bounce Charges must be numeric format."

- **Behavior**
  - **Successful Submission**
    - If all required fields are valid, the form data is saved in the database, and a success confirmation is displayed.
  - **Form Reset**
    - Clicking the **Clear Button** resets all form fields to default values after user confirmation.
  - **Navigation**
    - Clicking the **Previous Button** discards unsaved changes and navigates back to the Bank Table View.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.4.2.6 Data Requirements

None.

2.4.2.7 Non-Functional Requirements

None.

2.4.2.8 Configurations

None.

2.4.2.9 Flowchart

None.

2.4.2.10 Additional Notes

None.

2.4.3 Update Bank

2.4.3.1 Purpose

The Bank Edit section allows authorized users to modify existing bank records to maintain accurate and up-to-date information. This ensures that the ERP system remains aligned with evolving financial requirements, operational standards, and business needs.

2.4.3.2 User Type

Registered Users, Admins.

2.4.3.3 Screen Layout

Same Create Bank layout.

2.4.3.4 UI Elements

Same Create Bank layout.

2.4.3.5 Functional Requirements

- **Initiate Update Process**
  - **Description:** The update process begins when the user clicks the **Edit** button in the Action column of the bank table.
  - **Functionality:**
    - **Edit Button Action:**

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- Loads the selected bank existing details (e.g., Bank Name, Short Name, Bounce Charges, etc.) into the **Bank Creation Form**.
- Pre-populates all form fields with the current values of the selected bank record to enable editing.

- **Expected Outcome:** The **Bank Creation Form** is populated with the selected bank current data, ready for modification.

- **Modify Bank Details**
  - **Description:** Allows users to edit one or more fields of the bank record, such as Name, Short Name, Bounce Charges Type, and Bounce Charge.
  - **Fields Available for Update:**
    - **Name:** Update the full name of the bank.
    - **Short Name:** Change the abbreviated name for the bank.
    - **Bounce Charges Type:** Update the selected bounce charges type.
    - **Bounce Charges:** Update the bounce charges.
    - **Status:** Change the active status to "Active" or "De-active."
  - **Expected Outcome:** Users can update any editable field(s) as required before saving the changes.

- **Form Validation (Update Mode)**
  - **Purpose:** Ensures that updated information meets required validation rules before submission.
  - **Validation Criteria:**
    - **Mandatory Fields:** The following fields must not be empty:
      - Bank Name
      - Bank Short Name
    - **Specific Field Validation:**
      - **Bank Name:** Up to 50 characters, allows letters, spaces, and certain special characters.
      - **Short Name:** 2 to 10 uppercase letters.
      - **Bounce Charges Type:** Must select an option from the dropdown.
      - **Bounce Charges:** Numeric values only, maximum 5 digits.
    - **Error Handling:** If validation fails, display specific error messages to guide the user, such as:
      - "Bank Name is required."

- "Short Name must be 2-10 uppercase letters."

- **Expected Outcome:** All required fields are validated, and errors are displayed if any criteria are not met.

○ **Submit Updated Data**

- **Description:** Saves the modified data to the database upon successful validation.
- **Functionality:**
  - **Submit Button:**
    - Saves the updated bank details to the database and refreshes the bank table to reflect the changes.
    - Displays a success message confirming the update, such as: "Bank details have been successfully updated."
- **Expected Outcome:** The updated bank record is stored in the database, and the user is notified of the successful update.

○ **Cancel Update**

- **Purpose:** Allows users to discard changes if they choose not to update the bank record.
- **Functionality:**
  - **Cancel Option:** Users can cancel the update process by either closing the form or clicking the **Previous** button.

○ **Expected Outcome:** The form closes without saving any changes, leaving the original bank record intact.

○ **Confirmation Message**

- **Purpose:** Notifies the user that the update operation was successfully completed.
- **Functionality:**
  - **Display Message:** Upon successful submission, a confirmation message is displayed, such as: "Bank details have been successfully updated."
- **Expected Outcome:** The confirmation message assures the user that the updated data has been saved successfully.

2.4.3.6 Data Requirements

None.

2.4.3.7 Non-Functional Requirements

None.

2.4.3.8 Configurations

None.

2.4.3.9 Flowchart

None.

2.4.3.10 Additional Notes

None.

2.4.4 Remove Bank

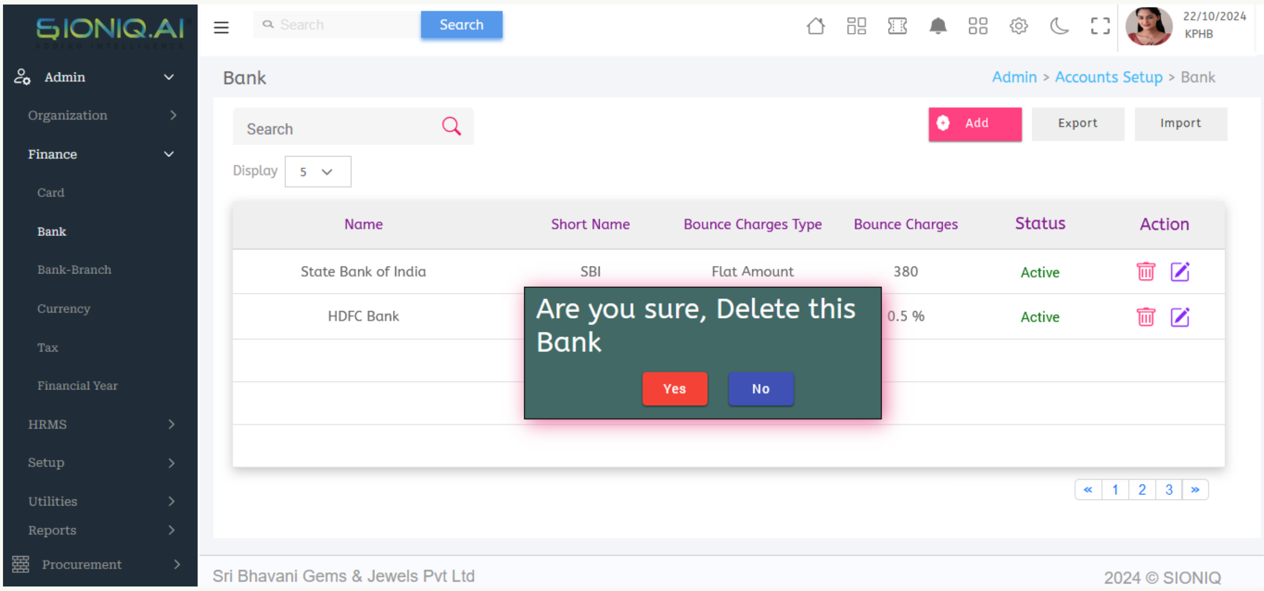
2.4.4.1 Purpose

The Bank Remove functionality allows authorized users to delete existing bank records from the system. This feature ensures that obsolete or incorrect bank data can be effectively managed, supporting accurate record maintenance and financial operations. The deletion process is permission-controlled and includes safeguards such as a confirmation prompt to prevent accidental removal of critical data.

2.4.4.2 User Type

Registered Users, Admins.

2.4.4.3 Screen Layout



Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

2.4.4.4 UI Elements

None.

2.4.4.5 Functional Requirements

- **Access Control**
  - **Purpose**: Ensure that only authorized users can perform the delete operation for bank.
  - **Functionality**:
    - **Delete Permission**:
      - The **Delete** button is only accessible to users with delete permissions configured in the system's user roles and permissions settings.
    - **Expected Outcome**: Unauthorized users cannot access or initiate the delete process.
- **Initiate Delete Action**
  - **Description**: The delete process begins when an authorized user clicks the **Delete** button in the **Action** column of the bank table.
  - **Functionality**:
    - **Delete Button Action**:
      - Prompts the user with a delete confirmation message to prevent accidental deletion.
    - **Expected Outcome**: The system displays a confirmation prompt, and no action is taken unless the user confirms the delete request.
- **Delete Confirmation Prompt**
  - **Purpose**: To ensure the user intends to proceed with deletion and prevent accidental removal of data.
  - **Functionality**:
    - **Confirmation Message**:
      - When the **Delete** button is clicked, the system displays a warning message:
        - "Are you sure you want to delete this bank?"
    - **Options**:
      - **Yes**: Confirms the deletion and proceeds with removing the bank record.
      - **No**: Cancels the delete action without making any changes.
    - **Expected Outcome**: The bank record is deleted only when the user confirms by selecting **Yes**.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Delete Execution**
  - **Description**: After confirmation, the system marks the bank record as deleted in the database.
  - **Functionality**:
    - **Data Removal**:
      - Updates the selected bank record's status to "Status=0" in the database.
    - **Dependencies Check**:
      - Before deletion, the system checks for any dependencies (e.g., associated branches, or transactions).
      - **If dependencies exist**:
        - Displays a message: "Cannot delete bank as it has associated records."
        - Halts the delete process until dependencies are resolved.
    - **Expected Outcome**: The system removes the record only when no dependencies are found.
- **Success Message**
  - **Purpose**: Inform the user that the delete action was successfully completed.
  - **Functionality**:
    - **Display Message**:
      - Upon successful deletion, display a success message such as:
        - "Bank has been successfully deleted."
    - **Expected Outcome**: The user receives confirmation that the record was deleted.
- **Refresh Table View**
  - **Description**: Updates the bank table view to reflect the changes made by the deletion.
  - **Functionality**:
    - **Refresh Table**:
      - Automatically removes the deleted bank record from the list in the table view.
    - **Pagination Update**:
      - Adjusts the pagination if necessary to ensure the remaining records are displayed correctly.
    - **Expected Outcome**: The table reflects the updated bank, with the deleted record no longer visible.

2.4.4.6 Data Requirements

None.

2.4.4.7 Non-Functional Requirements

None.

2.4.4.8 Configurations

None.

2.4.4.9 Flowchart

None.

2.4.4.10 Additional Notes

None.

2.5 Bank Branch

The Bank Branch Summary form in the ERP system is designed to provide a concise overview of bank branch-specific data. This form allows users to access and review essential information related to bank branches, including financial performance, account balances, transaction volumes, and other relevant metrics. By summarizing bank branch data in one place, the form facilitates quick analysis, decision-making, and monitoring of branch operations within the ERP system. This consolidated view enables branch managers to identify trends, allocate resources effectively, and make informed financial decisions. The Bank Branch Summary form plays a crucial role in optimizing branch performance and ensuring seamless management of bank branch-related activities within the ERP system.

2.5.1 Data View

2.5.1.1 Purpose

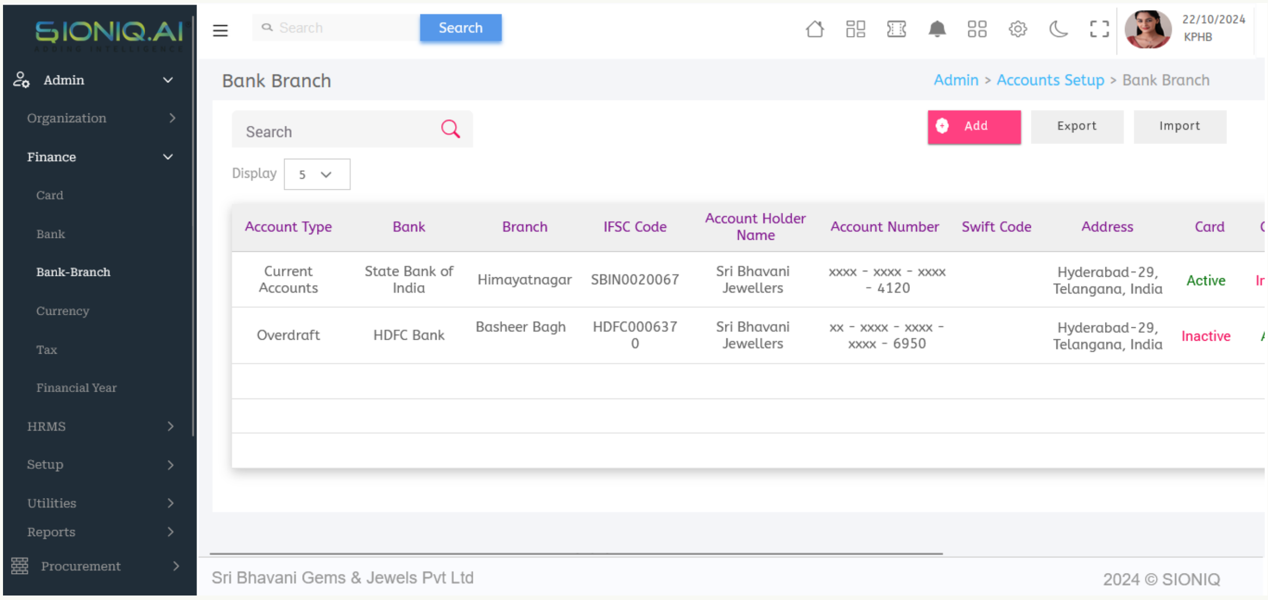
The Bank Branch Table Data View provides a comprehensive and organized overview of all bank branches registered in the ERP system, offering key information about each branch's location, operational status, and associated financial data. This view enables users to quickly locate and assess individual bank branches, including details such as branch name, address, district, state, country, account types, and active status. The table also supports filtering options, allowing users to narrow down their search by branch location, status, or financial metrics for efficient navigation. With action buttons for editing and deleting entries, users can easily manage bank branch records, ensuring that the system remains updated and aligned with the organization's financial and operational requirements.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.5.1.2 User Type

Registered Users, Admins.

2.5.1.3 Screen Layout



2.5.1.4 UI Elements

- **Search Option**
  - **Description**: A search bar located at the top of the data view section to filter the branch records displayed in the table.
  - **Functionality**:
    - Enables users to search by attributes such as Bank Name, Branch Name, Short Name, or Account Type.
    - Dynamically updates the displayed table to show only results that match the search criteria.
- **Display Dropdown**
  - **Description**: A dropdown to control the number of rows shown in the table at a time.
  - **Options**: Common choices such as 5, 10, 25, 50, or 100 rows per page.
  - **Functionality**:
    - Allows users to set how many branch records are displayed per page.
    - The table updates immediately based on the selected number of rows.
- **Button**
  - **Add Branch Button**

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Label**: "New"
- **Functionality**: Opens the Branch Creation Form to add a new branch record to the system.

- **Export Button**
  - **Label**: "Export" (or icon button as specified)
  - **Functionality**:
    - Allows users to export the displayed branch data in various file formats, such as Excel.
    - Exports either all or filtered data based on the current view and search filter.
- **Import Button**
  - **Label**: "Import" (or icon button as specified)
  - **Functionality**:
    - Redirects users to the Import Form page, where branch data can be imported from external files.

- **Tabular View**
  - **Description**: Displays a structured view of all registered bank branches.
  - **Columns**:
    - **Account Type**: Indicates the type of bank account (e.g., Savings, Current).
    - **Bank**: Displays the name of the bank associated with the branch.
    - **Branch**: Shows the name of the bank branch.
    - **IFSC Code**: Displays the IFSC code for the branch, used for electronic transfers.
    - **Account Holder Name**: Displays the name of the account holder associated with the branch.
    - **Account Number**: Shows the unique account number assigned to the branch.
    - **Swift Code**: Displays the SWIFT/BIC code for international transactions, if applicable.
    - **Address**: Provides the address details of the bank branch.
    - **Status**: Displays the active status of each branch, with values such as "Active" or "Inactive."
      - **Status**:
        - "Active" if the checkbox is checked in the form.
        - "Inactive" if the checkbox is unchecked in the form.
    - **Action**:
      - **Edit**: Opens the Branch Creation Form populated with the branch’s details for editing.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Delete**: Triggers a delete confirmation prompt to remove the selected branch record from the system (permission-based).

- **Paging for Table View**
  - **Description**: Includes pagination controls at the bottom of the table to navigate through multiple pages of branch records.
  - **Functionality**:
    - Provides navigation buttons (e.g., "Previous" and "Next") to move between pages.
    - Shows page numbers, allowing users to jump directly to a specific page.
    - Updates the displayed branch records based on the selected page, making it easy to manage large datasets.

2.5.1.5 Functional Requirements

None.

2.5.1.6 Data Requirements

None.

2.5.1.7 Non-Functional Requirements

None.

2.5.1.8 Configurations

- **Functionality:**
  - **Permission Settings**: Controls the visibility and accessibility of buttons and actions based on user roles or permissions.
- **Available Permissions:**
  - **Add Branch Permission**: Grants access to the "Add Branch" button for adding new branch records.
  - **Export Permission**: Grants access to the "Export" button for exporting branch data.
  - **Import Permission**: Grants access to the "Import" button for importing branch data.
  - **Edit Permission**: Allows users to access the "Edit" button in the Action column to modify branch records.
  - **Delete Permission**: Allows users to access the "Delete" button in the Action column to delete branch records.
- **Permission Assignment:**
  - Admins or authorized users can assign permissions to individual users or user roles.

2.5.1.9 Flowchart

None.

2.5.1.10 Additional Notes

None.

2.5.2 Create Branch

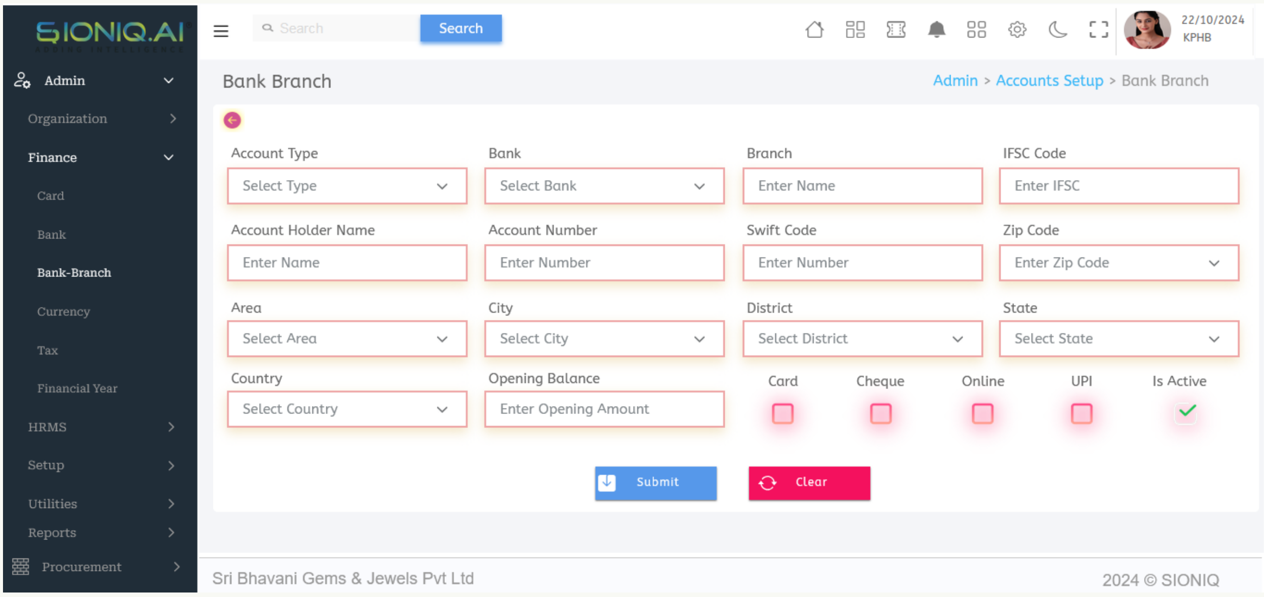
2.5.2.1 Purpose

This section allows users to add a new bank branch to the ERP system. Users can enter key information such as the branch name, short name, account type, bank name, IFSC code, SWIFT code, account holder name, account number, and address. This process ensures consistent data entry for operational and reporting purposes, facilitating accurate banking operations and seamless integration within the ERP system.

2.5.2.2 User Type

Registered Users, Admins.

2.5.2.3 Screen Layout



2.5.2.4 UI Elements

- **Account Type**
  - **Type**: Dropdown (Select)
  - **Label**: “Account Type”
  - **Editable**: Yes

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Description**: Defines the type of bank account for the branch.
- **Validation**: Required
- **Options**: Import Account Type Name's.
  - **Source**: Account Type Table.

- **Bank**
  - **Type**: Dropdown (Select)
  - **Label**: "Bank"
  - **Editable**: Yes
  - **Description**: Specifies the bank associated with the branch.
  - **Validation**: Required
  - **Options**: Values imported from the Bank table.

- **Branch Name**
  - **Type**: Text Input (Text Editor)
  - **Label**: "Name"
  - **Editable**: Yes
  - **Description**: Defines the name or location of the bank branch.
  - **Validation**:
    - **Required**: Yes
    - **Character Limit**: 50 characters
    - **Allowed Characters**: Letters, spaces, and special characters (e.g., apostrophes).

- **IFSC Code**
  - **Type**: Text Input
  - **Label**: "IFSC Code"
  - **Editable**: Yes
  - **Description:** Enter the IFSC code of the bank branch for electronic fund transfers.
  - **Validation**:
    - **Required**: Yes
    - **Format**: Alphanumeric, typically 11 characters.

- **Account Holder Name**
  - **Type:** Text Input

- **Label:** "Account Holder Name"
- **Editable:** Yes
- **Description:** Name of the account holder linked to the bank account.
- **Validation:** Required

- **Account Number**
  - **Type:** Text Input
  - **Label:** "Account Number"
  - **Editable:** Yes
  - **Description:** The unique account number associated with the branch.
  - **Validation:** Required, Numeric only, Length validation (depending on bank requirements).

- **SWIFT Code**
  - **Type:** Text Input
  - **Label:** "SWIFT Code"
  - **Editable:** Yes
  - **Description:** The SWIFT/BIC code for international transfers.
  - **Validation:** Optional, alphanumeric, and typically 8-11 characters long.

- **Zip Code**
  - **Type**: Dropdown
  - **Label**: Zip Code
  - **Editable**: Yes
  - **Description**: Select the zip code for the branch's location.
  - **Options**: Import Zip Code.
    - **Source**: Zip Code Master

- **Area**
  - **Type**: Dropdown
  - **Label**: Area
  - **Editable**: Yes
  - **Description**: Select the area in which the branch is located.

- **Options**: Import Area.
  - **Source**: Area Master

- **City**
  - **Type**: Dropdown
  - **Label**: City
  - **Editable**: Yes
  - **Description**: Select the city associated with the branch location.
  - **Options**: Import City.
    - **Source**: City Master

- **District**
  - **Type**: Dropdown
  - **Label**: District
  - **Editable**: Yes
  - **Description**: Select the district of the branch.
  - **Options**: Import District.
    - **Source**: District Master

- **State**
  - **Type**: Dropdown
  - **Label**: State
  - **Editable**: Yes
  - **Description**: Select the state associated with the branch.
  - **Options**: Import State.
    - **Source**: State Master

- **Country**
  - **Type**: Dropdown
  - **Label**: Country
  - **Editable**: Yes
  - **Description**: Select the country of the branch.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- **Options**: Import Country
    - **Source**: Country Master

- **Status**
    - **Type**: Checkbox
    - **Label**: Is Active
    - **Editable**: Yes
    - **Default Value**: Checked (True)
    - **Behavior**:
        - **Checked**: The branch is active and visible in the system.
        - **Unchecked**: The branch is inactive but retained in the database.

- **Submit Button**
    - **Type**: Button
    - **Label**: Submit
    - **Editable**: No (Standard Button)
    - **Description**: A button that submits the filled form details to the database.
    - **Validation**:
        - Ensures all required fields are completed and validated.
    - **Behavior**:
        - If validation is successful, the form data is submitted to the database.
        - If validation fails, displays an error message indicating the fields that require correction.

- **Clear Button**
    - **Type**: Button
    - **Label**: “Clear”
    - **Editable**: No (Standard Button)
    - **Description**: Resets all form fields to default values.
    - **Validation**:
        - Not required; clearing the form does not require validation checks.
    - **Behavior**:

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

- Reset Form Fields: Clears all entered data in the form fields, resetting them to their default values.
- Confirmation Prompt: Prompts the user for confirmation to avoid accidental data loss.

- **Previous Button**
  - **Type**: Button
  - **Label**: "Previous Icon"
  - **Editable**: No (Standard Button)
  - **Description**: Closes the Branch Creation Form and returns the user to the main Branch table view, discarding any unsaved changes.
  - **Behavior**: Cancels any unsaved changes and navigates back.
  - **Validation**:
    - **Required**: No

2.5.2.5 Functional Requirements

- **Actions**
  - **Submit Button Click**
    - **Action**: Triggers form submission if all required fields (Bank, Branch Name, and Account Type) contain valid values.
    - **Outcome**:
      - **Create Operation**: If it's a new entry, saves the new Branch information to the database.
      - **Update Operation**: If editing an existing entry, updates the branch information.
- **Validation**
  - **Mandatory Fields**
    - **Fields**: Branch Name, Bank, Account Type are required fields.
    - **Behavior**:
      - Users must enter valid values in each mandatory input field.
      - If any mandatory field is left empty, the form submission is blocked, and an error message is displayed (details in the "Error Messages" section).
  - **Field Format Validation**
    - **IFSC Code**: Alphanumeric, typically 11 characters.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

  - **Account Number**: Numeric values only.
  - **Swift Code**: Alphanumeric, and typically 8-11 characters long.

- **Error Messages**
  - **Submit Attempt with Missing Values**
    - **Condition**: If one or more required fields are missing a value when the user clicks Submit.
    - **Behavior**: The system displays specific error messages indicating which field(s) are missing values.
    - **Message Examples**:
      - "Branch Name is required" if the Branch Name field is empty.
      - "Bank is required" if the Bank field is not selected.
      - "Account Type is required" if the Account Type field is not selected.
  - **Validation Errors**
    - **Condition**: If any field does not meet the required format or constraints.
    - **Behavior**: The system displays an error message specifying the field and the expected input.
    - **Message Examples**:
      - "IFSC Code cannot exceed 11 characters."
      - "The account number must consist of numeric values only."
      - "Please select account Type."
- **Behavior**
  - **Successful Submission**
    - If all required fields are valid, the form data is saved in the database, and a success confirmation is displayed.
  - **Form Reset**
    - Clicking the Clear Button resets all form fields to default values after user confirmation.
  - **Navigation**
    - Clicking the Previous Button discards unsaved changes and navigates back to the Branch Table View.

2.5.2.6 Data Requirements

- **Data Source**: The data required for the "Create Bank Branch" form is either user-provided or imported from relevant system tables to ensure consistency and accuracy.
- **Account Type**
  - **Source**: Account Type Master Table
  - **Usage**: Specifies the type of account associated with the branch.
  - **Details**: Contains a list of all account type registered in the system, enabling users to associate with branch.
- **Bank**
  - **Source**: Values imported from the bank table.
  - **Usage**: Links the branch to a specific bank.
  - **Details**:
    - Contains a list of all banks registered in the system, enabling users to associate a branch with a specific bank.
- **Zip Code Data**
  - **Source**: Zip Code Master Table
  - **Usage**: Populates the "Zip Code" dropdown in the Branch Creation form.
  - **Details**: Contains a list of zip codes associated with the selected area.
- **Area Data**
  - **Source**: Area Master Table
  - **Usage**: Populates the "Area" dropdown in the Branch Creation form.
  - **Details**: Contains a list of areas filtered based on the selected city.
- **City Data**
  - **Source**: City Master Table
  - **Usage**: Populates the "City" dropdown in the Branch Creation form.
  - **Details**: Contains a list of cities filtered based on the selected district.
- **District Data**
  - **Source**: District Master Table

- **Usage**: Populates the "District" dropdown in the Branch Creation form.
- **Details**: Contains a list of districts filtered based on the selected state.

- **State Data**
  - **Source**: State Master Table
  - **Usage**: Populates the "State" dropdown in the Branch Creation form.
  - **Details**:
    - Contains a list of states available in the system.
    - Dynamically filters based on the selected country.

- **Country Data**
  - **Source**: Country Master Table
  - **Usage**: Populates the "Country" dropdown in the Branch Creation form.
  - **Details**: Contains a complete list of countries in the system, used to filter states dynamically.

2.5.2.7 Non-Functional Requirements

None.

2.5.2.8 Configurations

None.

2.5.2.9 Flowchart

2.5.2.10 Additional Notes

None.

2.5.3 Update Branch

2.5.3.1 Purpose

The **Bank Branch Edit** section allows authorized users to modify existing bank branch records to maintain accurate and up-to-date information. This ensures that the ERP system remains aligned with evolving banking operations, compliance requirements, and business needs.

2.5.3.2 User Type

Registered Users, Admins.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.5.3.3 Screen Layout

Same Create Branch layout.

2.5.3.4 UI Elements

Same Create Branch Elements.

2.5.3.5 Functional Requirements

- **Initiate Update Process**
    - **Description**: The update process begins when the user clicks the Edit button in the Action column of the branch table.
    - **Functionality**:
        - **Edit Button Action**:
            - Loads the selected branch's existing details (e.g., Account Type, Bank, Branch Name, IFSC Code, Account Number, Account Holder Name, Swift Code, etc.) into the Branch Creation Form.
            - Pre-populates all form fields with the current values of the selected branch record to enable editing.
    - **Expected Outcome**: The Branch Creation Form is populated with the selected branch's current data, ready for modification.
- **Modify Branch Details**
    - **Description**: Allows users to edit one or more fields of the branch record, such as Name, Address, Account Type, Account Number, IFSC Code, Account Holder Name, Swift Code and more.
    - **Fields Available for Update**:
        - **Account Type**: Modify the associated Account Type.
        - **Bank**: Change the associated Bank.
        - **Name**: Update the full name of the branch.
        - **IFSC Code**: Change the IFSC Code of the branch.
        - **Account Holder Name**: Change the Account Holder Name.
        - **Account Number**: Change the Account Number.
        - **Swift Code**: Change the swift code.
        - **Address**: Update the full address, including Country, State, District, City, and Area.
        - **Status**: Change the active status to "Active" or "De-active."

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Expected Outcome**: Users can update any editable field(s) as required before saving the changes.

- **Form Validation (Update Mode)**
  - **Purpose**: Ensures that updated information meets required validation rules before submission.
  - **Validation Criteria**:
    - **Mandatory Fields**: The following fields must not be empty:
      - **Account Type**
      - **Bank**
      - **Branch Name**
    - **Specific Field Validation**:
      - **Branch Name**: Up to 50 characters, allows letters, spaces, and certain special characters.
      - **Account Number**: Numeric only.
    - **Error Handling**: If validation fails, display specific error messages to guide the user, such as:
      - "Branch Name is required."
      - "Account Number must be Numeric only."
      - "Account Type must be selected."
  - **Expected Outcome**: All required fields are validated, and errors are displayed if any criteria are not met.

- **Submit Updated Data**
  - **Description**: Saves the modified data to the database upon successful validation.
  - **Functionality**:
    - **Submit Button**:
      - Saves the updated branch details to the database and refreshes the branch table to reflect the changes.
      - Displays a success message confirming the update, such as: "Branch details have been successfully updated."
  - **Expected Outcome**: The updated branch record is stored in the database, and the user is notified of the successful update.

- **Cancel Update**

- **Purpose**: Allows users to discard changes if they choose not to update the branch record.
- **Functionality**:
    - **Cancel Option**: Users can cancel the update process by either closing the form or clicking the Previous button.
- **Expected Outcome**: The form closes without saving any changes, leaving the original branch record intact.

- **Confirmation Message**
    - **Purpose**: Notifies the user that the update operation was successfully completed.
    - **Functionality**:
        - **Display Message**: Upon successful submission, a confirmation message is displayed, such as: "Branch details have been successfully updated."
    - **Expected Outcome**: The confirmation message assures the user that the updated data has been saved successfully.

2.5.3.6 Data Requirements

None.

2.5.3.7 Non-Functional Requirements

None.

2.5.3.8 Configurations

None.

2.5.3.9 Flowchart

None.

2.5.3.10 Additional Notes

None.

2.5.4 Remove Branch

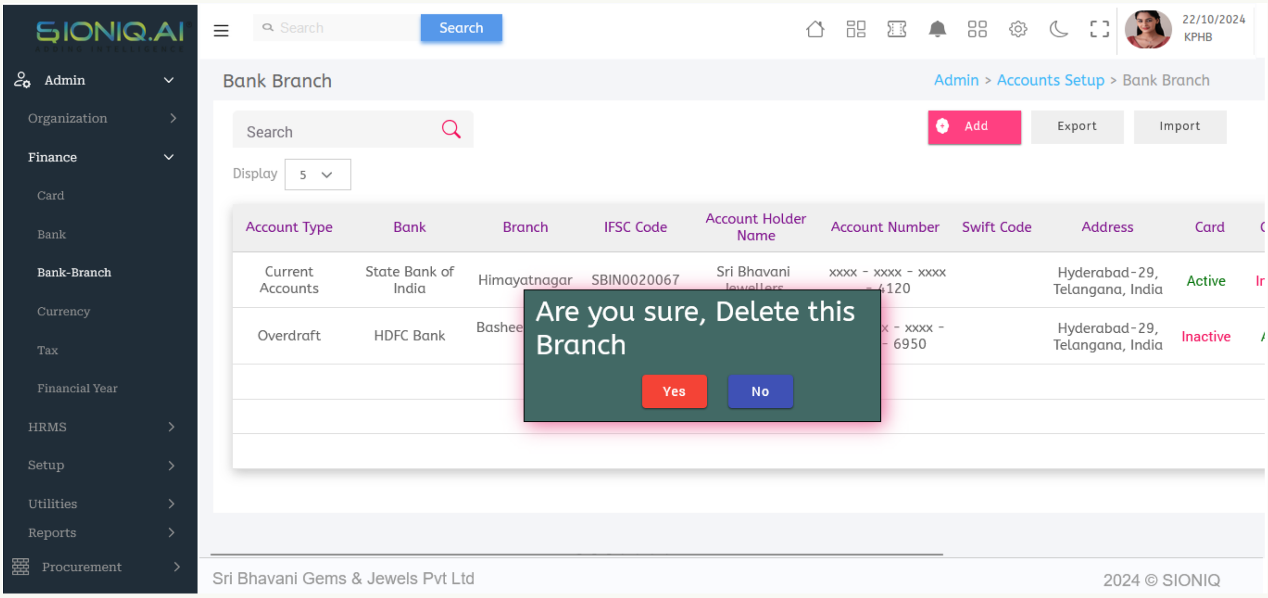
2.5.4.1 Purpose

The Bank Branch Remove functionality allows authorized users to delete existing bank branch records from the system. This feature ensures that obsolete or incorrect bank branch data can be effectively managed, supporting accurate record maintenance and operational efficiency. The deletion process is permission-controlled and includes safeguards such as a confirmation prompt to prevent accidental removal of critical data.

2.5.4.2 User Type

Registered Users, Admins.

2.5.4.3 Screen Layout



2.5.4.4 UI Elements

None.

2.5.4.5 Functional Requirements

- **Access Control**
  - **Purpose**: Ensure that only authorized users can perform the delete operation for bank branches.
  - **Functionality**:
    - **Delete Permission**:
      - The Delete button is only accessible to users with delete permissions configured in the system's user roles and permissions settings.
    - **Expected Outcome**: Unauthorized users cannot access or initiate the delete process.
- **Initiate Delete Action**
  - **Description**: The delete process begins when an authorized user clicks the Delete button in the Action column of the bank branch table.
  - **Functionality**:

- **Delete Button Action**:
  - Prompts the user with a delete confirmation message to prevent accidental deletion.
- **Expected Outcome**: The system displays a confirmation prompt, and no action is taken unless the user confirms the delete request.

- **Delete Confirmation Prompt**
  - **Purpose**: To ensure the user intends to proceed with deletion and prevent accidental removal of data.
  - **Functionality**:
    - **Confirmation Message**: When the Delete button is clicked, the system displays a warning message:
      - "Are you sure you want to delete this bank branch?"
        - **Options**:
        - **Yes**: Confirms the deletion and proceeds with removing the bank branch record.
        - **No**: Cancels the delete action without making any changes.
    - **Expected Outcome**: The bank branch record is deleted only when the user confirms by selecting Yes.

- **Delete Execution**
  - **Description**: After confirmation, the system marks the bank branch record as deleted in the database.
  - **Functionality**:
    - **Data Removal**:
      - Updates the selected bank branch record's status to "Status=0" in the database.
    - **Dependencies Check**:
      - Before deletion, the system checks for any dependencies (e.g., associated transactions, accounts, or linked entities).
      - **If dependencies exist**:
        - **Displays a message**:
          - "Cannot delete bank branch as it has associated records."
        - Halts the delete process until dependencies are resolved.
    - **Expected Outcome**: The system removes the record only when no dependencies are found.

- **Success Message**

- **Purpose**: Inform the user that the delete action was successfully completed.
- **Functionality**:
  - **Display Message**:
    - Upon successful deletion, display a success message such as:
      - "Bank branch has been successfully deleted."
  - **Expected Outcome**: The user receives confirmation that the record was deleted.

- **Refresh Table View**
  - **Description**: Updates the bank branch table view to reflect the changes made by the deletion.
  - **Functionality**:
    - **Refresh Table**:
      - Automatically removes the deleted bank branch record from the list in the table view.
    - **Pagination Update**:
      - Adjusts the pagination if necessary to ensure the remaining records are displayed correctly.
    - **Expected Outcome**: The table reflects the updated branch, with the deleted record no longer visible.

2.5.4.6 Data Requirements

None.

2.5.4.7 Non-Functional Requirements

None.

2.5.4.8 Configurations

None.

2.5.4.9 Flowchart

None.

2.5.4.10 Additional Notes

None.

2.6 Card

The Card Form in the Jewellery ERP system is designed to streamline the entry, management, and tracking of card-related data. This form allows users to create, edit, delete, and view details of various card types and accounts. By centralizing card information, the form ensures consistency in payment processing, account management, and reconciliation across multiple modules. This supports efficient operations, accurate financial reporting, and compliance with global payment standards, enabling seamless management of financial transactions within the ERP system.

2.6.1 Data View

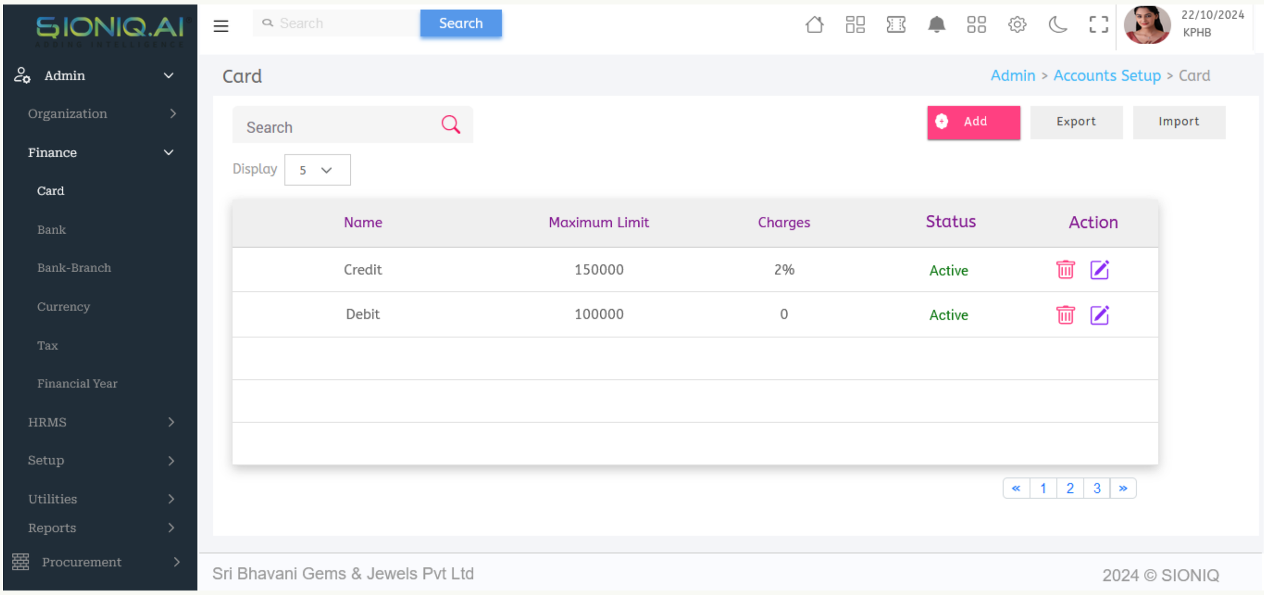
2.6.1.1 Purpose

The Card Data View provides a detailed and organized overview of all card-related records registered in the ERP system, offering essential information about each card type, associated accounts, and operational status. This view enables users to quickly locate and assess individual card records, including details such as card type, issuing bank, account holder name, and active status. The table supports filtering options, allowing users to refine their search by card type or status for efficient navigation. With action buttons for editing and deleting entries, users can easily manage card records, ensuring the system remains updated to reflect current financial and transactional needs.

2.6.1.2 User Type

Registered Users, Admins.

2.6.1.3 Screen Layout



Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

2.6.1.4 UI Elements

- **Search Option**
  - **Description**: A search bar located at the top of the data view section to filter the card records displayed in the table.
  - **Functionality**:
    - Enables users to search by attributes such as *Card Name* or *Status*.
    - Dynamically updates the displayed table to show only results matching the search criteria.
- **Display Dropdown**
  - **Description**: A dropdown to control the number of rows shown in the table at a time.
  - **Options**: Common choices such as 5, 10, 25, 50, or 100 rows per page.
  - **Functionality**:
    - Allows users to set how many card records are displayed per page.
    - The table updates immediately based on the selected number of rows.
- **Buttons**
  - **Add Card Button**
    - **Label**: "New"
    - **Functionality**: Opens the Card Creation Form to add a new card record to the system.
  - **Export Button**
    - **Label**: "Export" (or icon button as specified)
    - **Functionality**:
      - Allows users to export the displayed card data in various file formats, such as Excel.
      - Exports either all or filtered data based on the current view and search filter.
  - **Import Button**
    - **Label**: "Import" (or icon button as specified)
    - **Functionality**:
      - Redirects users to the Import Form page, where card data can be imported from external files.
- **Tabular View**
  - **Description**: Displays a structured view of all registered cards.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

- **Columns**:
  - **Card Name**: Displays the full name of each card.
  - **Maximum Limit**: Indicates the maximum limit assigned to the card.
  - **Charges**: Displays the charges associated with the card.
  - **Status**: Displays the active status of each card, with values such as "Active" or "Inactive."
    - **Status**:
      - "Active" if the checkbox is checked in the form.
      - "Inactive" if the checkbox is unchecked in the form.
  - **Action**:
    - **Edit**: Opens the Card Creation Form populated with the card's details for editing.
    - **Delete**: Triggers a delete confirmation prompt to remove the selected card record from the system (permission-based).

- **Paging for Table View**
  - **Description**: Includes pagination controls at the bottom of the table to navigate through multiple pages of card records.
  - **Functionality**:
    - Provides navigation buttons (e.g., *"Previous"* and *"Next"*) to move between pages.
    - Shows page numbers, allowing users to jump directly to a specific page.
    - Updates the displayed card records based on the selected page, making it easy to manage large datasets.

2.6.1.5 Functional Requirements

None.

2.6.1.6 Data Requirements

None.

2.6.1.7 Non-Functional Requirements

None.

2.6.1.8 Configurations

- **Functionality**:

- **Permission Settings**: Controls the visibility and accessibility of buttons and actions based on user roles or permissions.

- **Available Permissions**:
  - **Add Card Permission**: Grants access to the "Add Card" button.
  - **Export Permission**: Grants access to the Export button for exporting data.
  - **Import Permission**: Grants access to the Import button for importing data.
  - **Edit Permission**: Allows users to access the Edit button in the Action column to modify records.
  - **Delete Permission**: Allows users to access the Delete button in the Action column to delete records.
- **Permission Assignment**: Admins or authorized users can assign permissions to individual users or user roles.

2.6.1.9 Flowchart

None.

2.6.1.10 Additional Notes

None.

2.6.2 Create Card

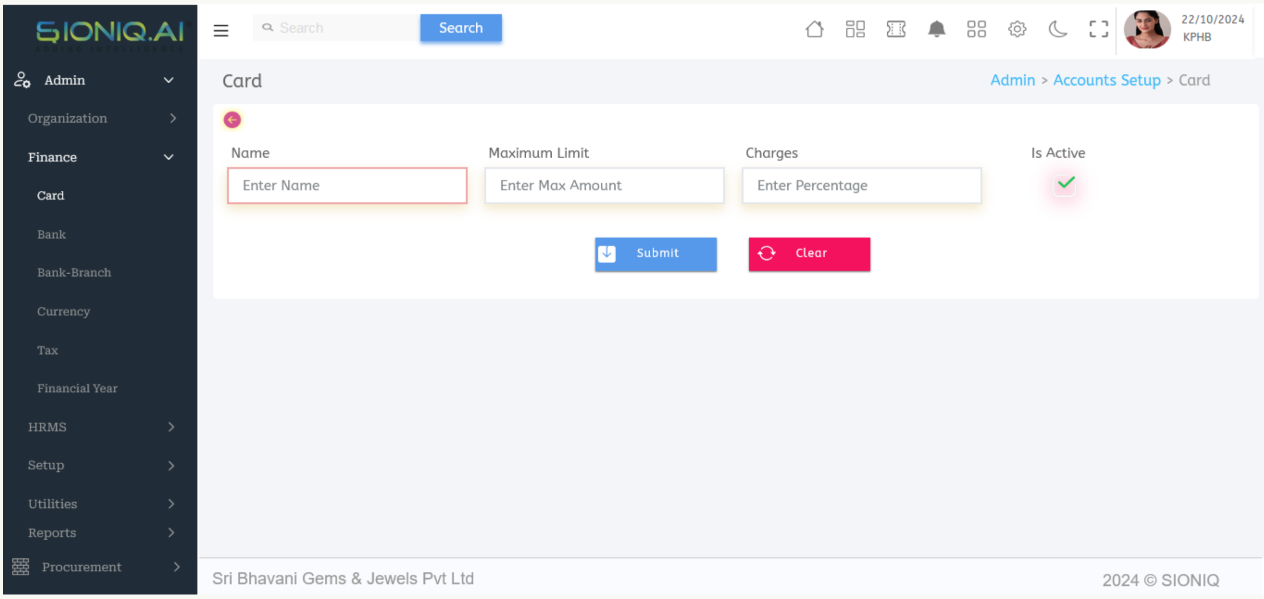
2.6.2.1 Purpose

This section allows users to add a new card to the ERP system. Users can enter key information such as the card name, card type, transaction fee percentage, transaction fee amount, and operational status. This process ensures consistent data entry for financial management and reporting purposes.

2.6.2.2 User Type

Registered Users, Admins.

2.6.2.3 Screen Layout



2.6.2.4 UI Elements

- **Card Name**
    - **Type**: Text Input (Text Editor)
    - **Label**: “Card Name”
    - **Editable**: Yes
    - **Description**: Input field for entering the full name of the card.
    - **Validation**:
        - **Required**: Yes
        - **Character Limit**: 50 characters
        - **Allowed Characters**: Letters, spaces, and special characters (e.g., apostrophes, hyphens).
- **Maximum Limit**
    - **Type**: Text Input (Numeric Editor)
    - **Label**: “Maximum Limit”
    - **Editable**: Yes
    - **Description**: Field to define the maximum transactional or usage limit for the card.
    - **Validation**:
        - **Required**: Yes

- **Allowed Characters**: Numbers only.

- **Charges**
  - **Type**: Text Input (Text Editor)
  - **Label**: "Charges"
  - **Editable**: Yes
  - **Description**: Input for specifying any applicable charges for using the card.
  - **Validation**:
    - **Required**: Yes
    - **Character Limit**: 5 characters
    - **Allowed Characters**: Numbers only.

- **Is Active**
  - **Type**: Checkbox
  - **Label**: "Is Active"
  - **Editable**: Yes
  - **Default Value**: Checked (True)
  - **Behavior**:
    - **Checked**: The card is active and visible in the system.
    - **Unchecked**: The card is inactive but retained in the database.

- **Submit Button**
  - **Type**: Button
  - **Label**: "Submit"
  - **Editable**: No (Standard Button)
  - **Description**: A button that submits the completed form details to the database.
  - **Validation**:
    - Ensures all mandatory fields are completed and validated.
  - **Behavior**:
    - **Successful Submission**: Saves data to the database.
    - **Failed Validation**: Displays an error message indicating the fields requiring correction.

- **Clear Button**
  - **Type**: Button
  - **Label**: "Clear"
  - **Editable**: No (Standard Button)
  - **Description**: Resets all form fields to their default values.
  - **Validation**: None required.
  - **Behavior**:
    - **Reset Form Fields**: Clears all entered data in the form fields, resetting them to their default values.
    - **Confirmation Prompt**: Prompts the user to confirm the action to prevent accidental data loss.
- **Previous Button**
  - **Type**: Button
  - **Label**: "Previous Icon"
  - **Editable**: No (Standard Button)
  - **Description**: Closes the Card Creation Form and returns the user to the main Card table view, discarding any unsaved changes.
  - **Behavior**: Cancels any unsaved changes and navigates back.
  - **Validation**: Not required.

2.6.2.5 Functional Requirements

- **Actions**
  - **Submit Button Click**
    - **Action**: Initiates form submission when all mandatory fields are filled and validated.
    - **Outcome**:
      - **Create Operation**: Saves a new card entry into the database if the form is valid.
      - **Validation**:
        - Ensures that all mandatory fields are populated with valid data.
        - Displays error messages if required fields are missing or contain invalid values.

- **Database Update**: On successful validation, the card details are saved in the system's database.

- **Clear Button Click**
  - **Action**: Clears all the fields in the form to allow the user to start fresh.
  - **Outcome**:
    - **Reset Form Fields**: Resets all input fields to their default states.
    - **Confirmation Prompt**: Prompts the user for confirmation before clearing data to prevent accidental data loss.
- **Previous Button Click**
  - **Action**: Returns the user to the main Card table view without saving changes.
  - **Outcome**:
    - **Navigation**: Closes the form and navigates to the Card table view.
    - **Unsaved Changes**: Discards all unsaved changes.

- **Validation**
  - **Mandatory Fields**
    - **Fields**: Card Name, Maximum Limit, and Charges are required fields.
    - **Behavior**:
      - The user must provide values for all mandatory fields before submitting the form.
      - If any required field is left empty, form submission is blocked, and an appropriate error message is displayed.
  - **Error Messages**
    - **Submit Attempt with Missing Values**
      - **Condition**: If one or more mandatory fields are empty when the user clicks Submit.
      - **Behavior**: The system displays an error message indicating the missing field(s).
      - **Message Examples**:
        - "Card Name is required."
        - "Maximum Limit is required."
        - "Charges is required."
  - **Field Validation**
    - **Card Name**: Must contain letters, spaces, and special characters, limited to 50 characters.
    - **Maximum Limit**: Must be a numeric value.

- **Charges**: Must be a numeric value, limited to 5 characters.

- **Error Handling**
  - **Invalid Input**:
    - Displays validation messages inline or as a popup for fields containing invalid input.
    - Prevents submission until all errors are corrected.
  - **Database Errors**:
    - Displays a user-friendly error message if there is a failure while saving data to the database (e.g., "Failed to save card details. Please try again.").

- **Behavior**
  - **Form State on Successful Submission**:
    - Clears all fields and resets the form after a successful save.
    - Displays a success message (e.g., "Card created successfully.").
  - **Form State on Failed Submission**:
    - Retains entered data in all fields for user correction.
    - Highlights invalid fields for correction and displays appropriate error messages.
  - **Form Reset**
    - Clicking the **Clear Button** resets all form fields to default values after user confirmation.
  - **Navigation**
    - Clicking the **Previous Button** discards unsaved changes and navigates back to the Bank Table View.

2.6.2.6 Data Requirements

None.

2.6.2.7 Non-Functional Requirements

None.

2.6.2.8 Configurations

None.

2.6.2.9 Flowchart

None.

Building 8-2-293-82/A, Plot 1107, Road Number 55, opposite Peddamma Talli Temple, CBI Colony, Jubilee Hills, Hyderabad, Telangana 500033

2.6.2.10 Additional Notes

None.

2.6.3 Update Card

2.6.3.1 Purpose

The Card Edit section allows authorized users to modify existing card records to maintain accurate and up-to-date information. This ensures that the ERP system remains aligned with evolving financial operations, compliance standards, and business needs.

2.6.3.2 User Type

Registered Users, Admins.

2.6.3.3 Screen Layout

Same Create Card layout.

2.6.3.4 UI Elements

Same Create Card layout.

2.6.3.5 Functional Requirements

- **Initiate Update Process**
  - **Description:** The update process begins when the user clicks the **Edit** button in the Action column of the bank table.
  - **Functionality:**
    - **Edit Button Action:**
      - Loads the selected card existing details (e.g., Card Name, Maximum Limit, Charges, etc.) into the **Card Creation Form**.
      - Pre-populates all form fields with the current values of the selected card record to enable editing.
  - **Expected Outcome:** The **Card Creation Form** is populated with the selected card current data, ready for modification.
- **Modify Card Details**
  - **Description:** Allows users to edit one or more fields of the card record, such as Name, Maximum Limit, Charges.
  - **Fields Available for Update:**

- **Name:** Update the full name of the card.
- **Maximum Limit:** Update the Maximum Limit.
- **Charges:** Update the charges.
- **Status:** Change the active status to "Active" or "De-active."

- **Expected Outcome:** Users can update any editable field(s) as required before saving the changes.

- **Form Validation (Update Mode)**

- **Purpose:** Ensures that updated information meets required validation rules before submission.
- **Validation Criteria:**
  - **Mandatory Fields:** The following fields must not be empty:
    - Card Name
    - Maximum Limit
    - Charges
  - **Specific Field Validation:**
    - **Card Name:** Up to 50 characters, allows letters, spaces, and certain special characters.
    - **Maximum Limit:** Must be a numeric value.
    - **Charges:** Numeric values only, maximum 5 digits.
  - **Error Handling:** If validation fails, display specific error messages to guide the user, such as:
    - "Card Name is required."
    - "Maximum Limit must be numeric value.
- **Expected Outcome:** All required fields are validated, and errors are displayed if any criteria are not met.

- **Submit Updated Data**

- **Description:** Saves the modified data to the database upon successful validation.
- **Functionality:**
  - **Submit Button:**
    - Saves the updated card details to the database and refreshes the card table to reflect the changes.

- Displays a success message confirming the update, such as: "Card details have been successfully updated."

- **Expected Outcome:** The updated card record is stored in the database, and the user is notified of the successful update.

- **Cancel Update**
  - **Purpose:** Allows users to discard changes if they choose not to update the card record.
  - **Functionality:**
    - **Cancel Option:** Users can cancel the update process by either closing the form or clicking the **Previous** button.
  - **Expected Outcome:** The form closes without saving any changes, leaving the original card record intact.

- **Confirmation Message**
  - **Purpose:** Notifies the user that the update operation was successfully completed.
  - **Functionality:**
    - **Display Message:** Upon successful submission, a confirmation message is displayed, such as: "Card details have been successfully updated."
    - **Expected Outcome:** The confirmation message assures the user that the updated data has been saved successfully.

2.6.3.6 Data Requirements

None.

2.6.3.7 Non-Functional Requirements

None.

2.6.3.8 Configurations

None.

2.6.3.9 Flowchart

None.

2.6.3.10 Additional Notes

None.

2.6.4 Remove Card

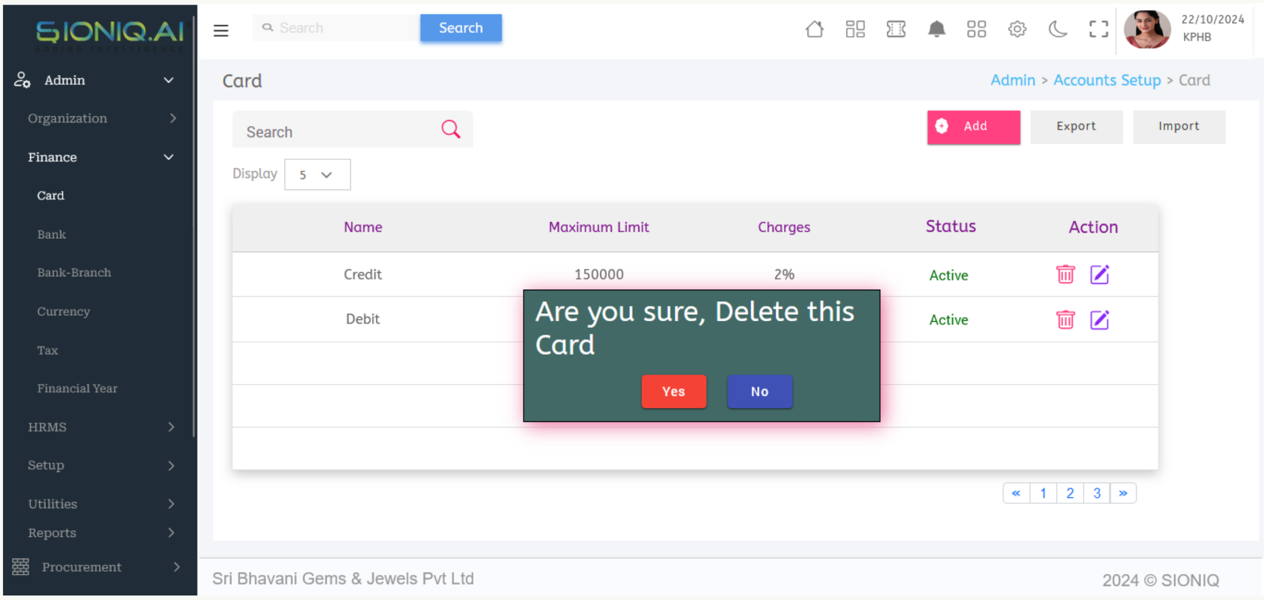
2.6.4.1 Purpose

The Card Remove functionality allows authorized users to delete existing card records from the system. This feature ensures that obsolete or incorrect card data can be effectively managed, supporting accurate record maintenance and operational efficiency. The deletion process is permission-controlled and includes safeguards such as a confirmation prompt to prevent accidental removal of critical data.

2.6.4.2 User Type

Registered Users, Admins.

2.6.4.3 Screen Layout



2.6.4.4 UI Elements

None.

2.6.4.5 Functional Requirements

- **Access Control**
  - **Purpose**: Ensure that only authorized users can perform the delete operation for card.
  - **Functionality**:
    - **Delete Permission**:
      - The **Delete** button is only accessible to users with delete permissions configured in the system's user roles and permissions settings.
    - **Expected Outcome**: Unauthorized users cannot access or initiate the delete process.
- **Initiate Delete Action**

- **Description**: The delete process begins when an authorized user clicks the **Delete** button in the **Action** column of the card table.
- **Functionality**:
  - **Delete Button Action**:
    - Prompts the user with a delete confirmation message to prevent accidental deletion.
  - **Expected Outcome**: The system displays a confirmation prompt, and no action is taken unless the user confirms the delete request.

- **Delete Confirmation Prompt**
  - **Purpose**: To ensure the user intends to proceed with deletion and prevent accidental removal of data.
  - **Functionality**:
    - **Confirmation Message**:
      - When the **Delete** button is clicked, the system displays a warning message:
        - "Are you sure you want to delete this card?"
    - **Options**:
      - **Yes**: Confirms the deletion and proceeds with removing the bank record.
      - **No**: Cancels the delete action without making any changes.
    - **Expected Outcome**: The bank record is deleted only when the user confirms by selecting **Yes**.

- **Delete Execution**
  - **Description**: After confirmation, the system marks the card record as deleted in the database.
  - **Functionality**:
    - **Data Removal**:
      - Updates the selected card record's status to "Status=0" in the database.
    - **Dependencies Check**:
      - Before deletion, the system checks for any dependencies.
      - **If dependencies exist**:
        - Displays a message: "Cannot delete card as it has associated records."
        - Halts the delete process until dependencies are resolved.
    - **Expected Outcome**: The system removes the record only when no dependencies are found.

- **Success Message**
  - **Purpose**: Inform the user that the delete action was successfully completed.
  - **Functionality**:
    - **Display Message**:
      - Upon successful deletion, display a success message such as:
        - "Card has been successfully deleted."
    - **Expected Outcome**: The user receives confirmation that the record was deleted.
- **Refresh Table View**
  - **Description**: Updates the card table view to reflect the changes made by the deletion.
  - **Functionality**:
    - **Refresh Table**:
      - Automatically removes the deleted card record from the list in the table view.
    - **Pagination Update**:
      - Adjusts the pagination if necessary to ensure the remaining records are displayed correctly.
    - **Expected Outcome**: The table reflects the updated card, with the deleted record no longer visible.

2.6.4.6 Data Requirements

None.

2.6.4.7 Non-Functional Requirements

None.

2.6.4.8 Configurations

None.

2.6.4.9 Flowchart

None.

2.6.4.10 Additional Notes

None.

Building 8-2-293-82/A, Plot 1107, Road Number 55,
opposite Peddamma Talli Temple, CBI Colony,
Jubilee Hills, Hyderabad, Telangana 500033

info@sioniqerp.com 040 69 888 999 www.sioniqerp.com