

Tab 1

Functional Requirements Specification (FRS)

Version: 1.0

Date: 2025-05-31

1. Purpose

This document outlines the functional requirements for the development of an AI Enabled Governance, Risk, and Compliance (GRC) platform. It defines what users expect the system to do in order to effectively manage and support their GRC-related activities.

The requirements are described from a user's perspective and are not dependent on any specific hardware or software platform. The goal is to ensure that the system meets business needs and user expectations, regardless of the underlying technology used to implement it.

2. Solution Overview

The proposed GRC (Governance, Risk, and Compliance) platform is intended to empower end users—without any programming skills—to **configure, automate, and manage their organizational processes**. These may include risk assessments, audits, compliance tracking, and more.

3. Key Functional Features

The platform will include the following core components:

3.1. **Solution Builder (SB)**

Allows users to create and configure end-to-end GRC solutions tailored to their organization's needs.

3.2. **Assessment Module (AM)**

Enables the design and execution of surveys, questionnaires, or risk assessments, with flexible templates and scoring logic.

3.3. **Workflow Designer (WD)**

Provides a drag-and-drop interface to automate process flows, approvals, and actions based on user-defined logic.

3.4. **Reports and Dashboards (RD)**

Allows users to generate reports, track KPIs, and create custom dashboards for real-time visibility into compliance and risk posture.

3.5. **Access Control (AC)**

Supports role-based access to ensure that users only see or modify data relevant to their responsibilities.

3.6. **Notifications (Email and SMS)**

Triggers alerts, reminders, and escalations via email or SMS, based on configurable events or due dates.

3.7. **Integrations (INT)**

Offers API connectors and integration points to sync with other enterprise systems (e.g., HRMS, ERP, SIEM, etc.).

3.8. **AI & ML Capabilities (AI)**

The platform will incorporate artificial intelligence and machine learning features to enhance risk and compliance operations. These capabilities will include:

- 3.8.1. **Predictive Analytics:** Automatically identify potential risks or compliance issues based on historical patterns and trends.
- 3.8.2. **Smart Recommendations:** Suggest actions, controls, or mitigations based on similar past cases or contextual inputs.
- 3.8.3. **Natural Language Processing (NLP):** Extract key information from unstructured data (e.g., emails, documents, comments) for analysis.
- 3.8.4. **Automated Classifications:** Categorize data such as risks, incidents, or findings using machine learning models.

3.9. **AI Studio**

A key innovation in the platform will be the **AI Studio**, a dedicated space that allows advanced users to build and customize their own AI Agents. This empowers organizations to tailor AI capabilities to their specific business needs without deep technical expertise.

Features of the AI Studio will include:

- 3.9.1. **Agent Builder Interface:** A user-friendly interface for designing AI workflows, decision trees, and logic-based agents.
- 3.9.2. **Prebuilt Templates:** Ready-to-use templates for common AI use cases (e.g., risk scoring, anomaly detection).

- 3.9.3. **Prompt Customization:** Ability to write and test natural language prompts for generative AI tasks (e.g., summarizing incidents, drafting reports).
- 3.9.4. **No-code/Low-code Experience:** Allows configuration of AI behavior with minimal technical input.
- 3.9.5. **Data Source Integration:** AI agents can be trained or configured to pull from forms, records, or external systems as needed.
- 3.9.6. **Monitoring & Governance:** Includes usage tracking, audit trails, and controls to ensure ethical and compliant use of AI.

Think of the AI Studio as a sandbox where users can create "virtual assistants" to help automate analysis, decision-making, and communication across their GRC processes.

4. Conceptual Model (Excel Analogy)

To help visualize how the platform works:

Workbook = A GRC Solution

Worksheet = An Application or Form within the solution

Row = A Record (e.g., a risk item, control, incident)

Column = A Field or Data Point (e.g., risk level, due date, owner)

This structure offers a familiar layout, allowing users to think of building and using GRC solutions as easily as managing an Excel file.

5. Key Design Considerations

To ensure usability, scalability, and security, the platform will adhere to the following design principles:

5.1. Scalability

Must support large datasets (up to 5 million records) without performance degradation.

5.2. User Experience (UX)

The user interface should be simple, intuitive, and self-explanatory, enabling easy navigation between modules.

5.3. Data Integrity

The system must ensure consistency and accuracy of stored data, with proper validation and audit trails.

5.4. Security Standards

Must follow industry best practices and frameworks (e.g., ISO 27001, OWASP Top 10) to

safeguard user data.

5.5. **Performance Optimization**

System response time (UI loading, record saving, process execution) should be fast even under full data load and concurrent user activity.

5.6. **Seamless Navigation**

Users should be able to move between screens, modules, and features quickly and intuitively.

6. FUNCTIONAL REQUIREMENTS

The following sections lists all **functional requirements** intended to guide the design and development of the configurable **GRC (Governance, Risk, and Compliance) platform**. This platform is designed to be flexible, modular, and scalable, supporting a wide range of organizational needs.

The solution components and expected functionalities of the product are outlined in detail below. Sample user interface views are provided at the end of this document to demonstrate typical user experiences.

6.1. User Types

The system is designed keeping in mind **three distinct user roles** with different access and capabilities:

6.1.1. **Admin Users:**

Users with administrative rights who configure and manage solutions according to the business needs. They have full control over user management, roles, access controls, workflows, integrations, and system administration.

6.1.2. **End Users:**

Users who utilize the configured solutions to perform their day-to-day operational tasks. Their access and capabilities are controlled based on roles and permissions set by Admin users.

6.1.3. **Product Owners:**

Users responsible for managing product licensing, controlling the visibility and availability of applications, and overseeing deployment settings and product-wide configurations.

FR ID	Requirement																								
(Admin) Solution Builder- SB																									
FR-SB 001	The Admin user shall have the ability to create multiple Solutions and Applications within the platform. Each Application must have a unique identifier and can be associated (tagged) with one or more Solutions simultaneously.																								
FR-SB 002	Users shall be able to add multiple Applications or Forms to the platform. Each Application or Form must have a unique identifier to ensure distinct identification within the system.																								
FR-SB 003	<p>Users shall be able to add an unlimited number of fields within each Application/Form. The platform must support the following field types with the described characteristics:</p> <table> <thead> <tr> <th>Field Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>ID / Tracking ID</td> <td>System-generated field with configurable prefix and/or suffix to uniquely identify records.</td> </tr> <tr> <td>Text / String</td> <td>Standard alphanumeric text field.</td> </tr> <tr> <td>Integer</td> <td>Whole number field.</td> </tr> <tr> <td>Unsigned Integer (Uint)</td> <td>Whole number field that only accepts non-negative integers.</td> </tr> <tr> <td>Decimal / Number</td> <td>Numeric field supporting decimal values. System should allow merging Integer, Uint, Decimal field types into a unified “Number” type with internal configuration options to define precision and constraints.</td> </tr> <tr> <td>Phone Number</td> <td>Field for phone numbers. May be merged with Text/String field after assessing benefits and limitations.</td> </tr> <tr> <td>IP Address (IPv4 & IPv6)</td> <td>Field to accept and validate IP addresses for both IPv4 and IPv6 formats.</td> </tr> <tr> <td>Date & Time</td> <td>Fields to capture dates, times, or combined timestamp values.</td> </tr> <tr> <td>Dropdown - Single Select</td> <td>Dropdown field allowing selection of one option from predefined choices.</td> </tr> <tr> <td>Dropdown - Multi Select</td> <td>Dropdown field allowing selection of multiple options from predefined choices.</td> </tr> <tr> <td>First Created</td> <td>System-generated field capturing the timestamp when a record is first created.</td> </tr> </tbody> </table>	Field Type	Description	ID / Tracking ID	System-generated field with configurable prefix and/or suffix to uniquely identify records.	Text / String	Standard alphanumeric text field.	Integer	Whole number field.	Unsigned Integer (Uint)	Whole number field that only accepts non-negative integers.	Decimal / Number	Numeric field supporting decimal values. System should allow merging Integer, Uint, Decimal field types into a unified “Number” type with internal configuration options to define precision and constraints.	Phone Number	Field for phone numbers. May be merged with Text/String field after assessing benefits and limitations.	IP Address (IPv4 & IPv6)	Field to accept and validate IP addresses for both IPv4 and IPv6 formats.	Date & Time	Fields to capture dates, times, or combined timestamp values.	Dropdown - Single Select	Dropdown field allowing selection of one option from predefined choices.	Dropdown - Multi Select	Dropdown field allowing selection of multiple options from predefined choices.	First Created	System-generated field capturing the timestamp when a record is first created.
Field Type	Description																								
ID / Tracking ID	System-generated field with configurable prefix and/or suffix to uniquely identify records.																								
Text / String	Standard alphanumeric text field.																								
Integer	Whole number field.																								
Unsigned Integer (Uint)	Whole number field that only accepts non-negative integers.																								
Decimal / Number	Numeric field supporting decimal values. System should allow merging Integer, Uint, Decimal field types into a unified “Number” type with internal configuration options to define precision and constraints.																								
Phone Number	Field for phone numbers. May be merged with Text/String field after assessing benefits and limitations.																								
IP Address (IPv4 & IPv6)	Field to accept and validate IP addresses for both IPv4 and IPv6 formats.																								
Date & Time	Fields to capture dates, times, or combined timestamp values.																								
Dropdown - Single Select	Dropdown field allowing selection of one option from predefined choices.																								
Dropdown - Multi Select	Dropdown field allowing selection of multiple options from predefined choices.																								
First Created	System-generated field capturing the timestamp when a record is first created.																								

	<p>Last Updated System-generated field capturing the timestamp of the last modification of the record.</p> <p>Currency Numeric field to capture monetary values with configurable currency symbols or codes (prefix/suffix, e.g., \$, €, USD).</p> <p>Discussion / Comments Field enabling users to add comments or have threaded discussions within the record.</p> <p>User List Field allowing selection of users from the system.</p> <p>Record Permission / Access Control Field controlling access at the form level. Persons or groups selected here will be granted form access. Supports manual or automated selection based on defined form parameters or conditional logic. Detailed specification is available in Section [XXX].</p> <p>Relation / Cross Reference Field linking records to other applications/forms. Connected records from linked applications should be visible as a list view within the parent record. Details described in Section [XXX].</p> <p>Attachment Field supporting file attachments of various formats to be linked with records.</p> <p>Image Field supporting image uploads and display within records.</p> <p>E-Signature Field enabling electronic signature capture and validation within the form.</p> <p>Record Status System field to capture the current status of the record (to be further defined).</p> <p>Email Templates (Chase Request) Fields to support automated email templates for notifications or reminders related to the record.</p> <p>Auditing / History Log System capability to track and log changes at the record and field level, capturing who made changes and when.</p>
<p>Notes:</p> <ul style="list-style-type: none"> • Detailed properties and configuration options for each field type will be specified in subsequent requirements. • The system should allow flexible customization of fields, including adding, removing, and editing field attributes, except where system fields are mandatory. 	

FR-SB
004

Common Field Properties

All fields in the system (except system-generated fields with predefined behavior) must support the following configurable properties:

Property	Description
Field Name	User-friendly name of the field displayed on forms and reports.
Field ID	Unique internal identifier used by the system to reference the field.
Description	Optional textual description providing additional information about the field's purpose or usage.
Status	Indicates if the field is Active or Inactive ; inactive fields are hidden or disabled from end users.
Access Control	Defines which users, roles, or groups have permission to view or edit the field. Supports role-based and conditional access rules.
Unique	Indicates whether field values must be unique across all records in the application.
Key	Marks the field as a key field used for record identification or linking.
Required / Mandatory	Specifies if the field must have a value entered before record save.
Calculated or Scripted	Allows field values to be auto-calculated using scripts or formulas. Applicable only to specific field types.
Default Value	Pre-populated value set when a new record is created, can be static or dynamic (e.g., current date).
Help Text	Optional help tooltip or icon displayed near the field to assist end users with guidance or instructions.
Calculate When Script Runs	Defines when the calculation or script should run (e.g., on record create, update, or manually triggered). Applicable to calculated/scripted fields.

	<p>Bulk Updates Indicates if this field can be updated in bulk operations across multiple records.</p> <p>Max Length Maximum allowed length for text-based fields (where applicable).</p> <p>Inline Edit Specifies if the field supports inline editing directly from list or grid views.</p> <p>Enable Audit History Enables tracking and logging of changes made to this field's value, including timestamps and user info.</p> <p>Note:</p> <ul style="list-style-type: none"> Additional field-specific properties may apply based on the field type and are defined within individual field sections. System fields (e.g., First Created, Last Updated) may have restrictions on some properties and are defined in separate sections. 																
FR-SB 005	<p>In addition to the common properties, fields have type-specific settings:</p> <table> <thead> <tr> <th style="text-align: center;">Field Type</th><th style="text-align: center;">Specific Properties & Description</th></tr> </thead> <tbody> <tr> <td>ID / Tracking ID</td><td>System number field with configurable prefix/suffix (e.g., IN-01) and starting number. Can be default key field for applications.</td></tr> <tr> <td>Text / String</td><td>Default common properties apply.</td></tr> <tr> <td>Integer, Uint, Decimal</td><td>Default common properties apply.</td></tr> <tr> <td>Phone Number</td><td>Default common properties apply.</td></tr> <tr> <td>IP Address (IPv4 & IPv6)</td><td>Default common properties apply.</td></tr> <tr> <td>Date & Time</td><td>Default value can be set to current date/time, a specific future date, or fixed date.</td></tr> <tr> <td>Dropdown (Single & Multi-Select)</td><td>Configurable as Global (usable across all apps) or Local (restricted to current app). Display options: dropdown, radio buttons, checkboxes, or popup. Supports min/max selection constraints.</td></tr> </tbody> </table>	Field Type	Specific Properties & Description	ID / Tracking ID	System number field with configurable prefix/suffix (e.g., IN-01) and starting number. Can be default key field for applications.	Text / String	Default common properties apply.	Integer, Uint, Decimal	Default common properties apply.	Phone Number	Default common properties apply.	IP Address (IPv4 & IPv6)	Default common properties apply.	Date & Time	Default value can be set to current date/time, a specific future date, or fixed date.	Dropdown (Single & Multi-Select)	Configurable as Global (usable across all apps) or Local (restricted to current app). Display options: dropdown, radio buttons, checkboxes, or popup. Supports min/max selection constraints.
Field Type	Specific Properties & Description																
ID / Tracking ID	System number field with configurable prefix/suffix (e.g., IN-01) and starting number. Can be default key field for applications.																
Text / String	Default common properties apply.																
Integer, Uint, Decimal	Default common properties apply.																
Phone Number	Default common properties apply.																
IP Address (IPv4 & IPv6)	Default common properties apply.																
Date & Time	Default value can be set to current date/time, a specific future date, or fixed date.																
Dropdown (Single & Multi-Select)	Configurable as Global (usable across all apps) or Local (restricted to current app). Display options: dropdown, radio buttons, checkboxes, or popup. Supports min/max selection constraints.																

	<p>Currency Option to add currency prefix or suffix (e.g., \$, €).</p> <p>Discussion / Comments Supports threaded comments or discussions.</p> <p>User List Links to Users module; supports lookup from system users or groups; min/max selection controls.</p> <p>Record Permission / Access Control Links to users and groups; controls min/max selections. Determines form-level access.</p> <p>Relation / Cross Reference Links to records from other applications; shows linked records in a list view inside the parent record.</p> <p>Attachment Max file size and number of attachments configurable; display attributes (filename, size, type, downloads).</p> <p>Image Configurable display width, height, max file size.</p> <p>Record Status System field with values like New (default on creation), Updated; read-only.</p> <p>Email Templates (Chase Requests) Compose email messages with recipients from user list or permission fields; select predefined templates.</p> <p>Auditing / History Log Track changes at the field level for all or selected fields; support linking audit logs to records.</p>												
FR-SB 006	<p>1) Filters define the types of logical conditions that can be applied to fields for purposes such as searching, reporting, workflow conditions, and other business logic implementations.</p> <table> <thead> <tr> <th>Field Type</th> <th>Supported Filter Operations</th> <th>Notes</th> </tr> </thead> <tbody> <tr> <td>Text / String</td> <td>Equals, Does Not Equal, Contains, Does Not Contain</td> <td>Case insensitive search</td> </tr> <tr> <td>Numeric (Integer, Uint, Decimal,Currency , Phone No)</td> <td>Equals, Does Not Equal, Less Than, Greater Than, Between</td> <td>Numeric comparisons, phone no treated as numeric or string based on config</td> </tr> <tr> <td>Date / Time</td> <td>Equals, Does Not Equal, Current, Last, Next, Greater Than, Less Than, Between, After,</td> <td>Support for relative dates (Current, Last, Next) and absolute</td> </tr> </tbody> </table>	Field Type	Supported Filter Operations	Notes	Text / String	Equals, Does Not Equal, Contains, Does Not Contain	Case insensitive search	Numeric (Integer, Uint, Decimal,Currency , Phone No)	Equals, Does Not Equal, Less Than, Greater Than, Between	Numeric comparisons, phone no treated as numeric or string based on config	Date / Time	Equals, Does Not Equal, Current, Last, Next, Greater Than, Less Than, Between, After,	Support for relative dates (Current, Last, Next) and absolute
Field Type	Supported Filter Operations	Notes											
Text / String	Equals, Does Not Equal, Contains, Does Not Contain	Case insensitive search											
Numeric (Integer, Uint, Decimal,Currency , Phone No)	Equals, Does Not Equal, Less Than, Greater Than, Between	Numeric comparisons, phone no treated as numeric or string based on config											
Date / Time	Equals, Does Not Equal, Current, Last, Next, Greater Than, Less Than, Between, After,	Support for relative dates (Current, Last, Next) and absolute											

	<p>Prior</p> <p>Dropdow (Single Multi-select)</p> <p>Cross Reference (Link to other application)</p> <p>IP Address (IPv4 & IPv6)</p> <p>Tracking ID</p> <p>User List / Record Permission</p>	<p>comparisons</p> <p>Equals, Does Not Equal, Contains, Does Not Contain</p> <p>Equals, Does Not Equal, Contains, Does Not Contain</p> <p>Equals, Does Not Equal, Between, Contains, Does Not Contain</p> <p>Equals, Does Not Equal</p> <p>Equals, Does Not Equal, Contains, Does Not Contain</p>	<p>Filters operate on predefined field values</p> <p>Operates on key field record IDs of referenced records</p> <p>Supports IP ranges, individual IPs, or no selection</p> <p>Match against tracking IDs in the application</p> <p>Operates on users or groups; supports no selection</p>
FR-SB 007	<p>Field values within the platform can be populated in two ways:</p> <ul style="list-style-type: none"> • Manual input by the end user or via integration data uploads • Calculation or Script driven, for automated or complex data manipulations <p>Calculations / Formulas</p> <ul style="list-style-type: none"> • Calculations are expressions that operate on values within one or multiple fields, either within the same application or across connected applications. • The concept is analogous to formulas in Excel worksheets. • The platform shall support a comprehensive set of functions similar to Excel formulas. (A detailed list of supported formulas will be provided in a separate document.) • There must be an option to recalculate fields manually or automatically as required. • Users shall be able to define and control the sequence in which calculations and scripts are executed within any application. • Supported field types for calculations include: 		

- Date
- Unsigned Integer (Uint)
- Integer (Int)
- Decimal
- String/Text
- Multi-select dropdown
- Cross Reference (automatically attach records from other applications based on conditions)

Script-Based Fields

- Scripts can assign or modify values in **any field type** and are used where calculations alone cannot fulfill complex logic requirements.
- Scripts enable advanced functionality such as:
 - Querying system tables and pushing results as new or updated records in other applications
 - Adding new records in linked applications
 - Performing Insert, Update, or Delete operations on any table
 - Getting and setting field values dynamically
- Script execution contexts include:
 - **UI Scripts** – Used primarily for controlling form layout elements or dynamic field values on the user interface.
 - **Server-side Scripts** – Used for backend processing tasks, such as automating record creation or updates triggered by conditions on saved records.

Use Cases for Scripts

- Automatically create a related record in another application when a record is saved and meets specific criteria.
- Modify field visibility or validation rules dynamically on the UI based on user inputs.
- Perform complex data transformations or integrations that cannot be handled by standard calculations.

FR-SB
008

The **Record Permission** field is a fundamental component for managing access to individual application records. This field controls who can view or modify a specific record within the application.

Field Type and Source

- The Record Permission field is a **dropdown or multi-select** type field.
- It is always linked to the **Users** and **Groups** defined in the Access Control module.
- The field dynamically displays all available users and groups configured within the Access Control module as selectable options.
- All users and groups created and maintained in the Access Control module will be visible and selectable within this field.

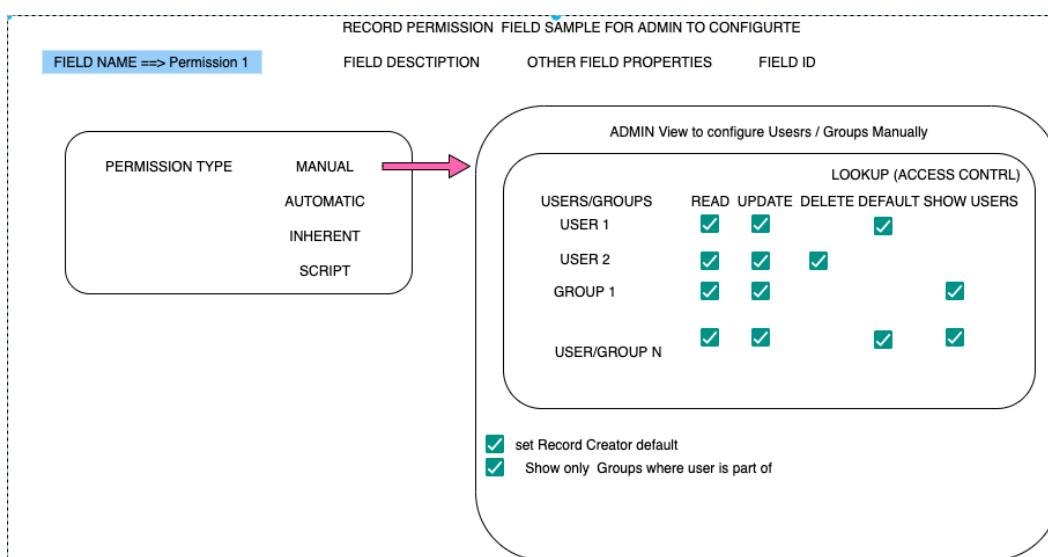
Access Rights Configuration

- Admins shall have the ability to configure granular permissions for each user or group selected within this field.
- Permissions available to configure include:
 - **Read** — Allows the user or group to view the record
 - **Update** — Allows the user or group to modify the record
 - **Delete** — Allows the user or group to delete the record

Populating Record Permission Field

The Record Permission field values can be populated by the following sources:

- **Manual Selection by Users:**
 - End users with appropriate permissions can manually select users or groups to assign record permissions.
 - The list of users and groups available for selection can be configured and filtered by Admins to control what options end users see.



End User view for above configuration



Automatic Rule-Based Conditions for Record Permission

The system shall support **dynamic, rule-based assignment** of users and groups to the Record Permission field based on the content of the record.

Rules will be defined by the Admin to automatically modify user/group access rights when certain conditions on the record's fields are met.

If multiple rules apply to a record, the system shall **aggregate all applicable rules** and grant the user or group the **highest level of access** among those rules.

Users and groups assigned via rules will be **automatically populated** into the Record Permission field without manual intervention.

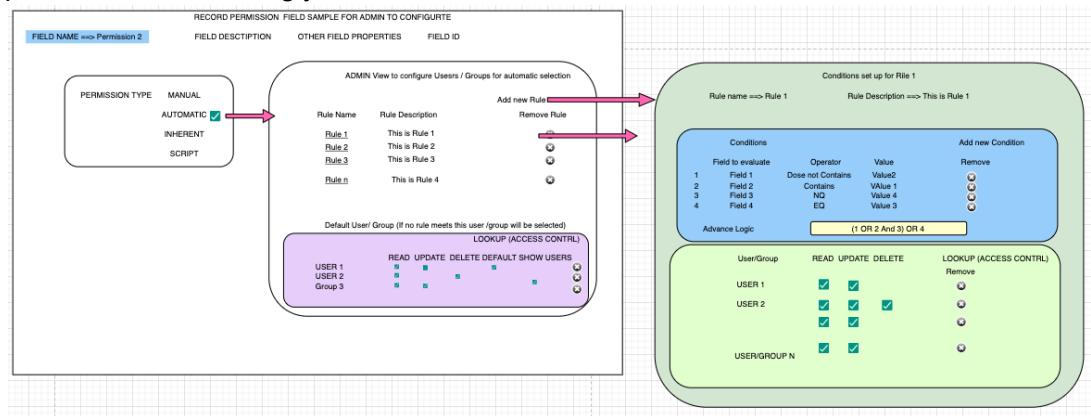
Admins shall have the ability to create, modify, and delete multiple rules that apply to the Record Permission field.

Rules should support complex conditions using logical operators such as AND, OR, and nested conditions to accurately model access control scenarios.

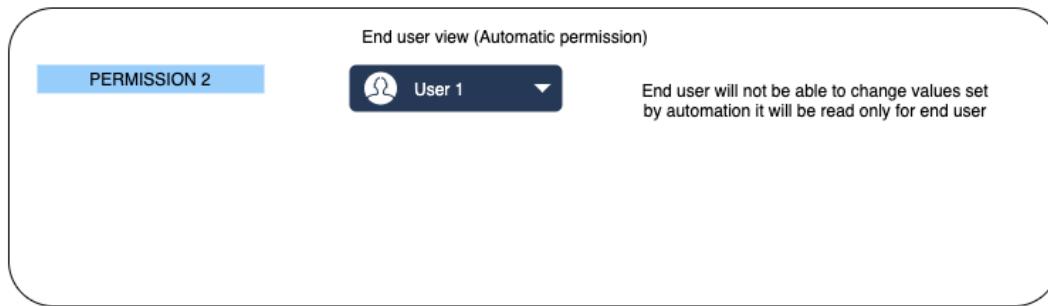
The rule configuration interface will allow Admins to specify:

- Target users or groups
- Conditions based on any field in the record
- Access rights to grant (Read, Update, Delete)

The system shall evaluate these rules **in real-time** during record creation or update to dynamically adjust permissions accordingly.



End user View



Inherent Access and Cross-Referenced Record Permissions

- The Record Permission field shall support **inheriting access rights** from related or cross-referenced records in connected applications.
- Users shall have the option to **select permissions from linked Record Permission fields** in related applications, allowing hierarchical or inherited access control.
- The system shall aggregate permissions from both the current record and linked records, ensuring that users/groups receive the **highest effective access level** across all related permissions.

- This inherited access mechanism enables cascading permission models where access granted on a parent or related record propagates to child or linked records as appropriate.

Value Population via Scripts

- The Record Permission field shall support being populated or modified programmatically via **scripts**.
- Scripts can dynamically add or remove users and groups from this field based on custom logic, data processing, or integration events.
- Detailed scripting guidelines and capabilities will be specified in a dedicated scripting section (refer to Section XXX).

Multiple Record Permission Fields per Application

- An application may contain **multiple Record Permission fields**, each potentially controlling access to different aspects or components of the record.
- The system shall evaluate all Record Permission fields assigned to a record and provide the **highest level of access** granted to a user or group if they appear in multiple fields.
- This multi-field approach allows granular and flexible access control configurations within a single application or form.

FR-SB
009

Form Builder: Frontend Application/Form Layout Designer

Objective:

Admin users should be able to design and configure the frontend layout of applications/forms using an intuitive drag-and-drop interface. This will allow flexible arrangement of form sections, fields, and tabs to tailor the user experience.

Functional Requirements:

- **Section Management**
Admin users shall be able to add multiple **Sections** in a form layout. Sections act as containers grouping related fields together.
- **Tabs within Sections**
Admin users shall be able to add **Tabs** inside Sections to organize fields or subsections under tabbed views for better UI clarity.
- **Drag & Drop Fields**
Admin users shall be able to **drag and drop fields** from the available field list onto any Section or Tab in the form layout.

	<ul style="list-style-type: none"> ○ Fields should be movable within the layout, including moving between Sections and Tabs. ○ Admin users shall be able to reorder fields by dragging and dropping to preferred positions. <ul style="list-style-type: none"> ● Remove Fields from Layout Admin users shall be able to remove/move fields off the layout, effectively hiding them from the frontend form but keeping them available in the application. ● Adjustable Layout Widths Admin users shall be able to adjust the width of Sections and individual fields to control how much horizontal space they occupy on the form. ● Cross Reference/List Values Sections Admin users shall be able to add special List Value Sections that display related records (cross-reference) from linked applications within the form layout. <ul style="list-style-type: none"> ○ These sections should allow configuration of display style, sorting, and filtering of referenced records.
FR-SB 010	Should be able to create front end Views based on users or groups
FR-SB 011	<p>Objective: Admin users shall be able to create and configure event-driven actions that control the behavior and interaction of forms and fields based on user activities on the UI.</p> <p>Event Triggers Supported: Admin users shall be able to configure actions on the following UI events:</p> <ul style="list-style-type: none"> ● On Form Load — Actions triggered when the form is initially loaded. ● On Field Change — Actions triggered when the value of a specified field changes. ● On Field Edit — Actions triggered when a user starts editing a specific field. ● On Record Save — Actions triggered when the user saves the form record. <hr/> <p>Conditional Logic: For On Field Change and On Field Edit events, admins should be able to define conditions for triggering actions, including:</p> <ul style="list-style-type: none"> ● Logical OR conditions between multiple fields. ● Conditions based on specific field values (e.g., if Field2 = "Yes"). ● Combination of multiple field-based conditions to control when actions run.

Available Actions to Configure on Events:

- **Make Field Mandatory** — Set one or more fields as required dynamically.
 - **Make Section Mandatory** — Set all fields within a section as required.
 - **Hide Section** — Hide an entire section from the user interface dynamically.
 - **Hide Field** — Hide individual fields dynamically.
 - **Set Field Values** — Programmatically set values in fields (supported types: text, date, multi-select, numeric, etc.).
 - **Clear Field Values** — Clear values in one or more specified fields.
 - **Send Notification** — Trigger notifications or email alerts to configured users or groups.
 - **Run Database Queries** — Execute predefined queries on the backend to insert, update, or delete records in other applications.
 - **Run Script** — Execute custom scripts to control advanced UI properties or behaviors, such as hiding Save/Edit buttons, modifying layout dynamically, or other advanced control.
-

Additional Notes:

- Admin users shall be able to configure multiple actions per event trigger.
- The sequence of action execution should be configurable if multiple actions are defined for a single event.
- There should be a user-friendly interface to build conditions and select actions without requiring coding skills.
- Actions triggered should provide feedback or logging for audit and debugging purposes.

FR-SB
012**Calculations in Application Fields****Objective:**

Admin users shall be able to define calculations/formulas for fields within an application to automatically compute values based on other field inputs or related application data.

Overview of Calculations:

At application level there should be control if calculation has to run real time without saving record or it has to be triggered on save

- **Calculation Definition:**

Admins can create calculation expressions that operate on values from one or multiple fields within the same application or from connected (related) applications.

- **Formula Support:**

Calculations shall support a wide range of functions similar to those available in spreadsheet tools (e.g., Excel), including arithmetic operations, logical comparisons, string manipulations, date functions, and aggregation.

- **Supported Field Types for Calculations:**

Calculations can be applied to the following field types:

- Date
- Integer (Int)
- Unsigned Integer (Uint)
- Decimal
- String/Text
- Multi-select Dropdown
- Cross Reference (linked records from other applications)

- **Calculation Execution:**

- Calculations can be set to run automatically when input fields change or on specific triggers/events.
- Admins should have the option to manually trigger recalculation on demand.
- Multiple calculations can be sequenced and prioritized within an application.

- **User Interface for Calculation Configuration:**

- Admin users will have an intuitive UI to build and edit calculation expressions without requiring advanced coding skills.
- There will be options to reference fields by name or ID, use constants, and apply functions.
- Error handling and validation should be provided to avoid invalid formulas.

- | | |
|--|--|
| | <ul style="list-style-type: none">• Calculation Output:
The calculated value will be stored in the target field and visible to end users, with options to make these fields read-only or editable depending on configuration. |
|--|--|

Additional Functional Details:

- **Calculation Formula Library:**
A predefined list of supported formulas and functions will be provided (to be detailed in a separate document).
- **Dependencies and Triggers:**
Calculations will track dependencies on input fields and automatically update outputs when dependent field values change.
- **Integration with Scripts:**
For complex logic that cannot be handled by calculations alone, scripts can be used to complement or override calculation results.

FR-SB 013	
--------------	--

Server-Side Script Execution and Management

Objective:

Admin users shall be able to configure and execute server-side scripts to perform backend operations, data updates, and system refresh tasks either manually or on a scheduled basis.

Functional Details:

- **Script Capabilities:**
Admins can write and manage scripts that run on the server side to perform the following actions:
 - Update records in one or multiple applications based on business logic.
 - Run complex queries on the existing database and use the query results to update records.
 - Trigger recalculations on records, with special focus on date fields or any fields requiring recalculation.
 - Refresh access control logic to dynamically update user/group permissions and record-level access.
 - Refresh or update other fields based on custom logic or dependencies.
- **Execution Modes:**

	<ul style="list-style-type: none"> ○ Manual Execution: Admins can manually trigger the execution of scripts at any time via the admin interface. ○ Scheduled Execution: Scripts can be scheduled to run automatically at defined intervals (e.g., hourly, daily, weekly). <ul style="list-style-type: none"> ● Script Management: <ul style="list-style-type: none"> ○ Admins should be able to create, edit, enable/disable, and delete scripts. ○ Scripts will support versioning and audit trail to track changes. ○ Proper error handling and logging must be provided to capture script execution status and errors. ● Security and Access: <ul style="list-style-type: none"> ○ Only authorized Admin users can create or modify server-side scripts. ○ Scripts should execute with appropriate permissions to ensure security and data integrity.
FR-SB 014	Autosave save option for records at application level can be enabled or disabled
FR-SB 015	<p>Form Action Buttons</p> <p>Requirement: Admin users should be able to create custom buttons on application forms.</p> <p>Details:</p> <ul style="list-style-type: none"> ● Buttons can be added with configurable labels such as Approve, Reject, Save, or any custom action name. ● Each button can be linked to specific actions or workflows, including but not limited to: <ul style="list-style-type: none"> ○ Triggering form save or submit. ○ Initiating approval or rejection workflows. ○ Running scripts or backend processes. ● Buttons should be configurable to appear conditionally based on form data or user roles.

FR-SB 016	<h2>Cross Reference / Relational Records Between Applications</h2> <p>Requirement: Users should be able to create relationships between records of two different applications using Cross Reference or relational fields.</p> <p>Details:</p> <ul style="list-style-type: none"> • Cross Reference fields allow linking of records from one application to another, enabling navigation and aggregation of related data. • Parent records should display linked child records in a list or summary view. • Relationship configurations should support one-to-one, one-to-many, and many-to-many mappings.
FR-SB 017	<h2>Use Case Management</h2> <p>Requirement: Admin users should be able to create and manage Use Cases, which are logical groupings of Solutions.</p> <p>Details:</p> <ul style="list-style-type: none"> • A Use Case is a dedicated page or workspace aggregating all elements of selected Solutions and Applications tied to a specific business scenario or process. • Features include: <ul style="list-style-type: none"> ○ Linking Solutions: Admins can select and link existing Solutions to a Use Case. ○ Ordering: Admins can change the display order of linked Solutions for user convenience. ○ Dashboards & Reports: Option to link relevant dashboards and reports to the Use Case page ○ Homepage Customization: Admins can customize the Use Case homepage using simple scripts or UI widgets to enhance user experience. ○ Access Control: Ability to restrict or grant access to Use Cases based on user roles or groups ○ Mandatory Setting: Option to mark Use Cases as mandatory (must be accessed) or optional for users.
FR-SB 018	<h2>Manage Use Cases</h2> <p>Requirement:</p>

Admin users should be able to **create, configure, and manage Use Cases**, which are logical groupings of Solutions. A **Solution** is a grouping of Applications.

Detailed Functional Requirements:

Feature	Description
Link Available Solution	Admin users should be able to link one or more existing Solutions to a Use Case . Solutions contain grouped applications and can be reused across multiple use cases.
Change Display Order of Solutions	Admins should have the ability to define the display sequence of linked solutions in the Use Case view for better user navigation and experience.
Link Dashboards & Reports	Ability to link relevant dashboards and reports (either prebuilt or user-defined) to the Use Case. These can provide insights into the performance, status, or data analytics of the applications within the Use Case.
Customize Use Case Homepage	Admins should be able to customize the homepage or landing screen of a Use Case using: <ul style="list-style-type: none"> - Simple scripts (e.g., JavaScript, JSON logic) - UI widgets (e.g., counters, summary views, charts) - Informational sections or announcements.
Access Control	The system should allow configuration of access control for each Use Case. This includes: <ul style="list-style-type: none"> - Granting or restricting visibility based on users, roles, or groups. - Read-only or full-access permissions.
Mandatory or Optional Visibility	Admins should have the ability to define whether a Use Case is: <ul style="list-style-type: none"> - Mandatory: Visible and accessible to all users in a specific role/group. - Optional: Users may choose to add it to their dashboard or ignore it.

(Admin) Assessment Module - AM

Overview:

The **Assessment Module** enables organizations to initiate and manage structured assessments, surveys, self-reviews, control evaluations, compliance checks, and policy/process audits. It is designed to be highly flexible and configurable.

and is **based on the existing Form Builder module** with additional functionalities specific to assessment workflows.

FR-AM 001	The assessment module must allow Admins to create assessment templates with predefined questions, sections, and logic
FR-AM 002	Admin should be able to create an assessment form same as application (defined in above section). One extra feature required is Question type field
FR-AM 003	<p>Requirement: The Assessment module should include a configurable "Question Type" field, which defines the expected format of responses for each assessment question.</p> <hr/> <h2>Supported Question Types</h2> <p>The Question Type field will support the following formats:</p> <ol style="list-style-type: none">1. Multi-Select Questions<ul style="list-style-type: none">○ Users can select from predefined options using dropdowns, radio buttons, or checkboxes.2. Text Questions<ul style="list-style-type: none">○ Open-ended questions where the answer is entered as free text input.3. Date Fields<ul style="list-style-type: none">○ Responses are captured as calendar dates (e.g., for setting due dates or event dates).4. Integer/Numeric Fields<ul style="list-style-type: none">○ Answers must be provided as numbers (e.g., ratings, counts, percentages).5. Attachment Field<ul style="list-style-type: none">○ Requires the user to upload a file as the answer (e.g., policy documents, evidence). <hr/> <h2>Configurability</h2> <ul style="list-style-type: none">● Admin users should be able to define and manage available question types within the platform settings● Each question type should be configurable in terms of:<ul style="list-style-type: none">○ Validation rules

	<ul style="list-style-type: none"> ○ Mandatory/optional setting ○ Default values ○ UI behavior (e.g., hide/show based on logic)
FR-AM 004	<p>Admin will create a question library where all questions will be created. Below are question types:</p> <ol style="list-style-type: none"> 1) Multi Select question i.e answer of question is dropdown) 2) Text questions i.e answer for this type of question is text input 3) Date fields to set some dates 4) Integer/numeric fields 5) Reference fields <p>User should be able to connect these questions to respective laws or compliances hosted in other applications</p>
FR-AM 005	<p>Requirement: Each question in the assessment module should support conditional visibility rules, allowing the system to dynamically show or hide questions based on user responses.</p> <hr/> <p>Key Capabilities:</p> <ul style="list-style-type: none"> ● Admin Configurability: Admin users should be able to define logic that controls the visibility of other questions based on the response to a particular question. ● Trigger Types: The logic should be triggered by: <ul style="list-style-type: none"> ○ User selecting a specific value (for dropdown, multiselect, radio button) ○ Text input (e.g., contains>equals specific text) ○ Numeric condition (e.g., >, <, = a value) ○ Date input (e.g., before/after a certain date) ● Supported Actions: <ul style="list-style-type: none"> ○ Show specific question(s) ○ Hide specific question(s) ○ Show/hide sections of the assessment form

	<ul style="list-style-type: none"> ● Chained Logic Support: <ul style="list-style-type: none"> ○ Visibility rules can apply recursively (e.g., Question 1 shows Question 2, and Question 2 can then trigger visibility for Question 3). ● Multiple Conditions: <ul style="list-style-type: none"> ○ Admin should be able to use logical operators (AND/OR) to define complex visibility rules. ● UI Indication: <ul style="list-style-type: none"> ○ Hidden questions should not be visible or accessible to the end user until triggered. ○ Once visible, the questions should function as normal form fields.
FR-AM 006	Each question can be linked to records from predefined applications such as Controls, Risks, Authoritative Sources, etc.
FR-AM 007	When a question is selected from the Question Library and added to an assessment, a snapshot of the question at that point in time is saved in the questionnaire. Any subsequent changes to the Question Library will not affect questions already included in existing assessments.
FR-AM 008	Users should be able to create assessments using predefined question templates.
FR-AM 009	<p>The platform should support automatic triggering of assessments based on the following conditions:</p> <ul style="list-style-type: none"> ● Date-based triggers (e.g., six months after the completion of a previous assessment). ● Specific field values within an assessment (e.g., a status field marked as "Reassess").
FR-AM 010	<p>While auto-triggering an assessment, the following configuration options should be available:</p> <ul style="list-style-type: none"> ● Copy last responses: Option to copy responses from the previous assessment — either all at once or selectively for each question. ● Include comments and attachments: Ability to carry forward comments and attachments from the previous responses. ● Send blank assessment: Option to initiate a completely new assessment with no prior data. ● Use latest template/questionnaire: Option to apply the most recent version of the template or questionnaire, if available. ● Set responder and approver: Ability to predefine or auto-assign the responder and approver for the

	triggered assessment.												
FR-AM 011	It should be possible to trigger a new assessment automatically based on the response or outcome of a previous assessment.												
(Admin) Workflow Designer - WD													
Requirement:													
The system must support flexible and configurable Workflow Management that determines how records or objects progress through a business process. Workflows can range from simple linear flows to complex conditional flows involving multiple stages, approvals, and actions.													
FR-WD 001	<p>1) Functional Description:</p> <table> <thead> <tr> <th style="text-align: center;">Feature</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Workflow Definition</td><td>Admin users should be able to define workflows that represent the lifecycle of a record or business object. Workflows define how a record moves from one state/stage to another.</td></tr> <tr> <td>Simple to Complex Workflow Support</td><td>The platform must support both: <ul style="list-style-type: none"> - Simple workflows (e.g., Submit → Review → Approve) - Complex workflows with conditional logic, parallel approvals, rework loops, time-based triggers, and escalation paths. </td></tr> <tr> <td>Visual Workflow Design</td><td>Workflows will be visually designed, with representations of stages, transitions, decision points, and actions. These will be supported using drag-and-drop components and editable diagrams (referenced from tools such as Visio or similar).</td></tr> <tr> <td>Workflow Visualization</td><td>Workflow designs should be exportable or viewable as Visio-style diagrams for better clarity and communication with stakeholders. These diagrams should illustrate the full process lifecycle for business and IT alignment.</td></tr> <tr> <td>Menu and UI-Based Configuration</td><td>Workflows can be created and modified through intuitive menu-driven interfaces, avoiding the need for custom coding in most cases. Admins can access workflow menus and configuration options directly from the UI.</td></tr> </tbody> </table>	Feature	Description	Workflow Definition	Admin users should be able to define workflows that represent the lifecycle of a record or business object. Workflows define how a record moves from one state/stage to another.	Simple to Complex Workflow Support	The platform must support both: <ul style="list-style-type: none"> - Simple workflows (e.g., Submit → Review → Approve) - Complex workflows with conditional logic, parallel approvals, rework loops, time-based triggers, and escalation paths. 	Visual Workflow Design	Workflows will be visually designed , with representations of stages, transitions, decision points, and actions. These will be supported using drag-and-drop components and editable diagrams (referenced from tools such as Visio or similar).	Workflow Visualization	Workflow designs should be exportable or viewable as Visio-style diagrams for better clarity and communication with stakeholders. These diagrams should illustrate the full process lifecycle for business and IT alignment.	Menu and UI-Based Configuration	Workflows can be created and modified through intuitive menu-driven interfaces , avoiding the need for custom coding in most cases. Admins can access workflow menus and configuration options directly from the UI.
Feature	Description												
Workflow Definition	Admin users should be able to define workflows that represent the lifecycle of a record or business object. Workflows define how a record moves from one state/stage to another.												
Simple to Complex Workflow Support	The platform must support both: <ul style="list-style-type: none"> - Simple workflows (e.g., Submit → Review → Approve) - Complex workflows with conditional logic, parallel approvals, rework loops, time-based triggers, and escalation paths. 												
Visual Workflow Design	Workflows will be visually designed , with representations of stages, transitions, decision points, and actions. These will be supported using drag-and-drop components and editable diagrams (referenced from tools such as Visio or similar).												
Workflow Visualization	Workflow designs should be exportable or viewable as Visio-style diagrams for better clarity and communication with stakeholders. These diagrams should illustrate the full process lifecycle for business and IT alignment.												
Menu and UI-Based Configuration	Workflows can be created and modified through intuitive menu-driven interfaces , avoiding the need for custom coding in most cases. Admins can access workflow menus and configuration options directly from the UI.												

	<p>Stage Management</p> <p>Each stage should support configurations such as:</p> <ul style="list-style-type: none"> - Entry and exit criteria - Field-level rules (e.g., make mandatory, show/hide fields) - Notifications (email, in-app) - Role-based access - Time limits and escalations <p>Record Status Tracking</p> <p>Every record involved in a workflow should reflect its current workflow stage, history, and next possible actions. This status should be visible to both users and auditors.</p> <p>Workflow Triggers</p> <p>Workflows should allow event-based triggers, such as:</p> <ul style="list-style-type: none"> - On record creation - On field value change - On status update - On due date - On escalation <p>Audit Trail</p> <p>Every transition or action within the workflow should be logged for auditing and traceability. The history should include who took the action, when, and any relevant comments or data changes.</p> <p>Role-Based Transitions</p> <p>Each transition between workflow stages can be restricted to specific roles or users, ensuring only authorized users can perform certain actions.</p> <p>Workflow Reusability</p> <p>Workflows should be reusable across multiple applications, either as-is or with slight modifications (cloned/copy workflows).</p>
--	--

(Admin) Notification - NT

FR-NT 001	<p>Requirement: The platform must support the ability to send notifications to both internal users (within the platform) and external users (via email or other channels). Notifications can be configured as mandatory (must be sent) or optional (configurable by admins or users).</p> <hr/> <p>Functional Description:</p> <table border="1"> <thead> <tr> <th>Feature</th><th>Description</th></tr> </thead> </table>	Feature	Description
Feature	Description		

	Internal Notifications	Notifications must be visible to users within the platform via an in-app notification center, dashboard widget, or alerts section.
	External Notifications	Notifications must be sent to external users through email. In the future, support for other channels (e.g., SMS, Slack, Teams) can be considered.
	Notification Recipients	Notifications can be sent to: <ul style="list-style-type: none">Users selected in any User List fieldGroups defined in the Access Control moduleUsers identified through scripts or workflow triggersExternal email addresses manually added
	Triggering Events	Notifications should be configurable to trigger on various system events , including (but not limited to): <ul style="list-style-type: none">Record creationField value changeWorkflow stage changeManual action (e.g., click of a button)Scheduled time-based triggers
	Notification Content	Each notification should support: <ul style="list-style-type: none">Subject line (dynamic)Message body (HTML or rich text)Merge fields (to insert dynamic values from record fields)Attachments, if applicable
	Templates	Admin users should be able to create and manage notification templates that can be reused across the platform. These templates can be selected in workflows, scripts, or actions.
	Mandatory vs Optional Notifications	Admin should be able to configure each notification as: <ul style="list-style-type: none">Mandatory → Always sent, cannot be disabled by usersOptional → Users may opt in/out or adjust frequency
	Audit Logging	The system should keep an audit log of sent notifications, including timestamp, recipients, content, and delivery status.
	Preview & Test	Admins should be able to preview the notification output and send test notifications before enabling it for actual use.
	Language & Localization	Notifications should support multi-language templates or localization, if required by business users.
	Notification Delivery Status	The system should track and optionally display delivery status (e.g., delivered, failed, opened), especially for emails.
	UI Component	Notifications should be accessible from the user interface via a notification bell icon , inbox-style page , or task center for in-app alerts.

FR-NT 003	<p>Requirement: The platform must allow Admin users to create and manage notification templates that can be reused across the system for emails, alerts, and system-generated messages. Templates must support both layout customization and dynamic content population using record field values.</p> <hr/> <p>Functional Description:</p> <table> <thead> <tr> <th style="text-align: center;">Feature</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Template Creation</td> <td>Admin users should be able to create new notification templates through a template management interface.</td> </tr> <tr> <td>Template Components</td> <td>Each template should support the following elements: <ul style="list-style-type: none"> Header Image (e.g., organization logo or banner)Footer Image (e.g., branding or disclaimer)Subject Line (static or dynamic using field tokens)Body (rich text or HTML, including inline images and styles) </td> </tr> <tr> <td>Dynamic Field Insertion</td> <td>Admin users should be able to insert field values from application records into the subject line, body, and other parts of the email using placeholder tokens (e.g., <code>{ {Record.ID} }</code>, <code>{ {User.Name} }</code>, <code>{ {Form.FieldName} }</code>).</td> </tr> <tr> <td>Merge Field Support</td> <td>Platform should support commonly used merge fields, including: Record IDApplication NameUser Names or EmailsAny field value in the formRecord Status, Created Date, etc.</td> </tr> <tr> <td>Template Preview</td> <td>Admins should be able to preview the full template with example record data before saving or applying it.</td> </tr> <tr> <td>Reusable Templates</td> <td>Templates should be selectable from a dropdown list when configuring notification rules in workflows, scripts, field-level actions, or system events.</td> </tr> <tr> <td>Multi-Language Support</td> <td>(Optional/Future) Should allow language-specific versions of a template for localization.</td> </tr> </tbody> </table>	Feature	Description	Template Creation	Admin users should be able to create new notification templates through a template management interface.	Template Components	Each template should support the following elements: <ul style="list-style-type: none"> Header Image (e.g., organization logo or banner)Footer Image (e.g., branding or disclaimer)Subject Line (static or dynamic using field tokens)Body (rich text or HTML, including inline images and styles) 	Dynamic Field Insertion	Admin users should be able to insert field values from application records into the subject line , body , and other parts of the email using placeholder tokens (e.g., <code>{ {Record.ID} }</code> , <code>{ {User.Name} }</code> , <code>{ {Form.FieldName} }</code>).	Merge Field Support	Platform should support commonly used merge fields, including: Record IDApplication NameUser Names or EmailsAny field value in the formRecord Status, Created Date, etc.	Template Preview	Admins should be able to preview the full template with example record data before saving or applying it.	Reusable Templates	Templates should be selectable from a dropdown list when configuring notification rules in workflows, scripts, field-level actions, or system events.	Multi-Language Support	(Optional/Future) Should allow language-specific versions of a template for localization.
Feature	Description																
Template Creation	Admin users should be able to create new notification templates through a template management interface.																
Template Components	Each template should support the following elements: <ul style="list-style-type: none"> Header Image (e.g., organization logo or banner)Footer Image (e.g., branding or disclaimer)Subject Line (static or dynamic using field tokens)Body (rich text or HTML, including inline images and styles) 																
Dynamic Field Insertion	Admin users should be able to insert field values from application records into the subject line , body , and other parts of the email using placeholder tokens (e.g., <code>{ {Record.ID} }</code> , <code>{ {User.Name} }</code> , <code>{ {Form.FieldName} }</code>).																
Merge Field Support	Platform should support commonly used merge fields, including: Record IDApplication NameUser Names or EmailsAny field value in the formRecord Status, Created Date, etc.																
Template Preview	Admins should be able to preview the full template with example record data before saving or applying it.																
Reusable Templates	Templates should be selectable from a dropdown list when configuring notification rules in workflows, scripts, field-level actions, or system events.																
Multi-Language Support	(Optional/Future) Should allow language-specific versions of a template for localization.																

	<p>Attachment Support Should support dynamic or static file attachments (e.g., generated PDFs, record files).</p> <p>Template Categories Admin can categorize templates for easy management (e.g., Approvals, Alerts, Reminders, Escalations).</p> <p>Versioning (Optional) Templates may include version control, allowing rollback to previous versions.</p> <p>Access Control Admin can control who can create, edit, and apply templates based on roles or permissions.</p>										
FR-NT 004	<p>Requirement: The platform must allow Admin users to attach documents and predefined reports to notifications, which are sent to both internal and external recipients. Supported formats include XLS/XLSX, PDF, CSV, and DOCX.</p> <hr/> <p>Functional Description:</p> <table> <thead> <tr> <th style="text-align: left;">Feature</th> <th style="text-align: left;">Description</th> </tr> </thead> <tbody> <tr> <td>Attachment Types</td> <td>Notifications should support the following types of attachments: Static file attachments (e.g., templates, disclaimers)Dynamically generated reports based on current record or conditionsField-specific files (e.g., uploaded documents from the record)</td> </tr> <tr> <td>Supported Formats</td> <td>The following output formats must be supported: Microsoft Excel (.xls, .xlsx)Adobe PDF (.pdf)CSV (.csv)Microsoft Word (.docx)</td> </tr> <tr> <td>Dynamic Report Inclusion</td> <td>Admin should be able to configure notifications to automatically include system-generated reports (e.g., compliance summary, audit log, task status report) triggered by: Workflow actionsField-level triggersScheduled events</td> </tr> <tr> <td>Parameter-Based Reports</td> <td>Reports included in the notification should support filters or parameters, such as: Record IDUser IDDate RangeApplication ID</td> </tr> </tbody> </table>	Feature	Description	Attachment Types	Notifications should support the following types of attachments: Static file attachments (e.g., templates, disclaimers)Dynamically generated reports based on current record or conditionsField-specific files (e.g., uploaded documents from the record)	Supported Formats	The following output formats must be supported: Microsoft Excel (.xls, .xlsx) Adobe PDF (.pdf) CSV (.csv) Microsoft Word (.docx)	Dynamic Report Inclusion	Admin should be able to configure notifications to automatically include system-generated reports (e.g., compliance summary, audit log, task status report) triggered by: Workflow actionsField-level triggersScheduled events	Parameter-Based Reports	Reports included in the notification should support filters or parameters, such as: Record IDUser IDDate RangeApplication ID
Feature	Description										
Attachment Types	Notifications should support the following types of attachments: Static file attachments (e.g., templates, disclaimers)Dynamically generated reports based on current record or conditionsField-specific files (e.g., uploaded documents from the record)										
Supported Formats	The following output formats must be supported: Microsoft Excel (.xls, .xlsx) Adobe PDF (.pdf) CSV (.csv) Microsoft Word (.docx)										
Dynamic Report Inclusion	Admin should be able to configure notifications to automatically include system-generated reports (e.g., compliance summary, audit log, task status report) triggered by: Workflow actionsField-level triggersScheduled events										
Parameter-Based Reports	Reports included in the notification should support filters or parameters, such as: Record IDUser IDDate RangeApplication ID										

	<p>Template Integration Admin should be able to define which report or file is included in which notification template, optionally using conditions.</p> <p>Link or Attachment Option to include the report as: A direct download linkAn embedded attachment in the email (depending on format and size)</p> <p>Security & Access Control Access to reports and attachments should follow the platform's access control model. If a report contains sensitive data, only authorized users should be able to open/download it.</p> <p>Size Limit Handling If the size of attachments exceeds email limits, system should auto-convert to download link or notify user.</p> <p>Preview and Test Option Admin should have the ability to preview test emails with example attachments before deployment.</p>						
FR-NT 005	<p>Requirement: Each application in the platform should provide centralized control to enable or disable notifications for that specific application.</p> <hr/> <p>Functional Description:</p> <table> <thead> <tr> <th style="text-align: center;">Feature</th> <th style="text-align: center;">Description</th> </tr> </thead> <tbody> <tr> <td>Application-Level Control</td> <td>Each application must include a "Notification Settings" panel accessible by Admin users.</td> </tr> <tr> <td>Toggle Option</td> <td>Admin should be able to toggle notifications ON or OFF globally for all event types (e.g., form submission, approval, rejection, workflow transitions).</td> </tr> </tbody> </table>	Feature	Description	Application-Level Control	Each application must include a " Notification Settings " panel accessible by Admin users.	Toggle Option	Admin should be able to toggle notifications ON or OFF globally for all event types (e.g., form submission, approval, rejection, workflow transitions).
Feature	Description						
Application-Level Control	Each application must include a " Notification Settings " panel accessible by Admin users.						
Toggle Option	Admin should be able to toggle notifications ON or OFF globally for all event types (e.g., form submission, approval, rejection, workflow transitions).						
(Admin) Access Control - AC							
FR-AC 001	<p>Admin users should be able to create new user accounts with the following details:</p> <ul style="list-style-type: none"> • First Name • Middle Name • Last Name 						

	<ul style="list-style-type: none"> • Email • Username • Domain • Locale • Organization details (aligned with LDAP parameters) • Password • Time Zone • Status (Active / Inactive) • Login history (linked to user login history log) • Comments • Groups (user's group memberships) • Roles (roles assigned to user) <p>A separate user table should be maintained with customizable attributes—admins can add or remove fields as needed, except for mandatory ones. Users created here will be referenced across the platform's applications.</p>
FR-AC 002	Admins should have the ability to create new Groups and add users to these Groups. (Refer to Group details section for more information.)
FR-AC 003	<p>Role Assignment</p> <p>Admin should be able to assign one or more roles to a user. (See Roles details section for more information.)</p>
FR-AC 004	Email should be triggered to User once password is set
FR-AC 005	Admin should have option to reset password

FR-AC 006	<p>First Login Password Change</p> <p>The system should provide an option to require users to change their password upon first login. Admins must have the ability to enable or disable this feature.</p>
FR-AC 007	There should be an option to link users to specific security parameters as defined in Security Parameter details (see FR-AC008).
FR-AC 008	The system must support integration with LDAP or Active Directory (AD) using SAML 2.0 for authentication and user management.
FR-AC 009	<p>Security Parameters Management:</p> <p>The system shall provide configurable security parameters to be administered by Admins for both Admin and End Users. The configurable parameters include:</p> <ul style="list-style-type: none"> • Minimum password length • Maximum password length • Requirement for special characters • Uppercase and lowercase character requirements • Password change interval • Password expiry settings • Maximum failed login attempts before account lockout • Automatic account deactivation period • Account reset options after lockout • Restriction on reuse of previous passwords (e.g., last 5 passwords disallowed) • Session timeout duration for users
FR-AC 010	<p>Groups Management:</p> <p>Admins shall be able to create groups and add users to these groups. A default group named EVERYONE shall exist, to which all users are automatically assigned. The EVERYONE group shall be read-only and cannot be modified.</p>
FR-AC 011	A group can be nested within another group, allowing one group to be a member of another.

FR-AC 012	<p>ROLES:</p> <p>CRUD (Create, Read, Update, Delete) access must be defined for both end users and admin functionalities. This will be managed by creating Roles and assigning appropriate CRUD permissions to each Role.</p>																																				
FR-AC 013	<p>Role Creation and Permissions Automation:</p> <p>Admin users should be able to create new roles. When a new application or module is added to the platform, predefined roles and associated permissions for that application should be automatically created in the backend. These roles will cover all functionalities related to the application, such as access to create new records, managing options on the home screen, and other relevant actions.</p> <p>For example, for an application like Incident, roles will be created with permissions as follows:</p> <table border="1"> <thead> <tr> <th>Application</th> <th>Page</th> <th>Create (C)</th> <th>Read (R)</th> <th>Update (U)</th> <th>Delete (D)</th> </tr> </thead> <tbody> <tr> <td>Incident</td> <td>End user record</td> <td>Yes</td> <td>Yes</td> <td>Yes</td> <td>No</td> </tr> <tr> <td>Incident</td> <td>Export Reports</td> <td>Yes</td> <td>Yes</td> <td>Yes</td> <td>No</td> </tr> <tr> <td>Incident</td> <td>Create Reports</td> <td>No</td> <td>Yes</td> <td>Yes</td> <td>No</td> </tr> <tr> <td>Incident</td> <td>Data Import</td> <td>Yes</td> <td>Yes</td> <td>Yes</td> <td>No</td> </tr> <tr> <td>Incident</td> <td>Email link of record</td> <td>N/A</td> <td>Yes</td> <td>N/A</td> <td>N/A</td> </tr> </tbody> </table>	Application	Page	Create (C)	Read (R)	Update (U)	Delete (D)	Incident	End user record	Yes	Yes	Yes	No	Incident	Export Reports	Yes	Yes	Yes	No	Incident	Create Reports	No	Yes	Yes	No	Incident	Data Import	Yes	Yes	Yes	No	Incident	Email link of record	N/A	Yes	N/A	N/A
Application	Page	Create (C)	Read (R)	Update (U)	Delete (D)																																
Incident	End user record	Yes	Yes	Yes	No																																
Incident	Export Reports	Yes	Yes	Yes	No																																
Incident	Create Reports	No	Yes	Yes	No																																
Incident	Data Import	Yes	Yes	Yes	No																																
Incident	Email link of record	N/A	Yes	N/A	N/A																																
FR-AC 014	<p>Multiple Role Creation:</p> <p>Admin users should have the ability to create multiple access roles to suit different user permissions and responsibilities across the platform.</p>																																				
FR-AC 015	<p>Role Access Resolution:</p> <p>If a user is assigned multiple roles, the system should grant the highest level of access available from any of the assigned roles.</p>																																				
FR-AC 016	<p>Conditional Access Control at Record Level:</p> <p>Conditional access control at the end-user record level is required (detailed in Section XXX). For example, if the incident severity is high, access should be automatically granted to a specific group or user; if the severity is low, access should be granted to a different set of users. Admins should be able to define such conditions based on any field in the form, using AND/OR logical operators.</p> <p>Example condition: If (Severity = High) OR (Details = XYZ AND Country = ABC) OR any other combination, then grant access to specific group.</p>																																				
FR-AC 017	<p>Field-Level Access Control:</p>																																				

	<p>Field-level access control is required and should be configurable manually by admins during field configuration. By default, all fields must be visible to users who have access to the corresponding record.</p> <p>Note: The access control flow follows this hierarchy: Access to system → Access to application → Access to record → Access to fields</p> <p>Field-level access should also be dynamically adjustable based on the attributes and conditions of the form.</p>
FR-AC 018	The system should support the bulk upload of users via offline methods, such as importing comma-delimited (CSV) files or other standard data upload formats.
FR-AC 019	<p>User Delegation: Users should have a delegation feature that allows them to select another user to delegate their access and responsibilities for a specified time period (start date to end date). During this timeframe, the delegate user will have the same access and work permissions as the original user..</p>
FR-AC 020	<p>Admin Impersonation: Administrators should have the ability to impersonate any user role to troubleshoot or perform actions on behalf of that user.</p>
(Admin) Integration - INT	
FR-INT 001	<p>System Integration: The platform should support integration with external systems using REST and SOAP APIs, enabling seamless data exchange and interoperability.</p>
FR-INT 002	<p>Data Consumption from External Systems: The system should be capable of consuming data from external systems via supported integration methods such as REST or SOAP APIs.</p>
FR-INT 003	<p>Data Transmission to External Systems: The system should be able to send data to other systems by exposing APIs for external consumption.</p>
FR-INT 004	<p>Data Upload via Files: The system should provide options to upload data into any Application or Module using comma-delimited (CSV) files.</p>
FR-INT 005	<p>Bidirectional Integration Configuration: The system should have the capability to configure bidirectional integrations, enabling both sending and receiving data between systems.</p>
FR-INT 006	<p>Data Upload and Ingestion:</p> <ul style="list-style-type: none"> • The system should support manual data upload/ingestion in CSV format. • The system should support automated data uploads using predefined file structures via automation processes.

FR-INT 007	<p>Email Integration:</p> <ul style="list-style-type: none"> The system should have the capability to read mailbox data and extract information from emails.
FR-INT 008	<p>External Data Injection:</p> <ul style="list-style-type: none"> The system should support injecting data from external sources such as databases and any systems.
FR-INT 009	<p>Integration Scheduling and Reliability:</p> <ul style="list-style-type: none"> The system should allow scheduling of integrations. It should support queuing and retry mechanisms to handle inbound data retrieval in case of failures.
FR-INT 010	<p>Cross-Application Data Operations:</p> <ul style="list-style-type: none"> The system should support running queries in one application and publishing the resulting data into another application within the platform.
FR-INT 011	<p>External Database Query Interface:</p> <ul style="list-style-type: none"> The system should provide a simple user interface where users can enter data source connection details, run queries against external databases, and import the resulting data into the platform.
FR-INT 012	<p>Integration Capabilities: The platform should support flexible, open integration possibilities—enabling both one-way and two-way data exchange with virtually any external system or solution available in the market, such as mailboxes, MS Teams Slack, various chat systems, ERP platforms, asset management systems, and more.</p>
(End Users + Admin) Searching & Reporting - SR	
FR-SR 001	<p>Users should be able to search and generate reports on all records they have access to within the system.</p> <p>Search queries should support complex logic using AND, OR, and IF-ELSE conditions.</p> <p>Users must have the option to select which fields to display in the search results.</p> <p>Search results should be sortable and groupable by any selected field.</p> <p>Users should be able to save search results as reports, provided they have the necessary permissions.</p>

Reports can be saved as **Personal Reports** (accessible only to the creator) or **Open Reports** (shared with other teams or users, access granted by the report creator).

Users should be able to generate visualizations such as maps, pie charts, stacked charts, funnel charts, cylinder charts, and other graph types similar to those available in Excel sheets. Color customization for charts should also be supported.

Reports should support **markers** (e.g., user-defined thresholds) that highlight specific values or conditions within the report.

Search filters must allow conditions based on field values, for example:

- Field 1 contains Value 1
- Field 2 contains Value 2
- Field 3 does not contain Value 3

Search conditions should support logical grouping, such as **(Field 1 AND Field 2) OR Field 3**.

Full filtering options must be available based on field types (as detailed in section XXXX).

Results should support sorting and grouping by any field value.

Users should be able to pull and include data from related or connected applications.

Search results and reports should be exportable in multiple formats including XLS, PDF, Word, CSV, XML, and HTML.

Users should have the ability to add reports to dashboards and display them on the home page (see home page section XXXX).

(End User + Admin) Packaging - PA

FR-PK G001	Packaging option is required to move application structure from one environment to other environment (only structure of application)
FR-PK G002	User should be able to move applications & related components from one environment to other i.e from Test to production
FR-PK G003	User should have option to move individual or all components
FR-PK G004	User should have option to deactivate components in destination environment which are not present in source pkg

Product Owner - PO

FR-PO 005	<p>Product Ownership Controls</p> <ul style="list-style-type: none"> • Product owners should have the ability to control the maximum number of forms/applications that an end user can create. • Product owners should be able to control which applications are visible and accessible to customers. • Product owners should be able to set limits on the number of users that can be created within a specific instance. • Product owners must have the capability to add or remove applications at any time. • Product owners should be able to control the number of instances that can be deployed under a single license key.
End User functionalities - EU	
FR-EU 001	<p>Home Page</p> <ul style="list-style-type: none"> • The system should provide a landing page tailored to the Solutions the user has access to. • Each Solution's landing page should allow the addition of dashboards. <p>Dashboards</p> <ul style="list-style-type: none"> • Users should be able to create dashboards that can contain multiple reports. • Reports can be selected from a dropdown list. • Users should also be able to create custom views or landing pages using scripts. <p>Search & Reports</p> <ul style="list-style-type: none"> • Users must be able to run searches across any application they have access to. • Users should have the option to save their search results as reports. <p>Access Review</p> <ul style="list-style-type: none"> • Users should be able to view their current access rights within the system.

FR-M F001	<h3>System Load & Job Management</h3> <ul style="list-style-type: none"> The system should include an application load balancer to distribute backend load across available servers in multi-server setups. Users (admins) should have the ability to configure load balancing preferences, including prioritizing jobs on specific servers. A real-time dashboard must be available to monitor running scripts and jobs. Admins should have controls to terminate or kill running jobs or scripts as needed. System logs must be captured and maintained for audit and troubleshooting purposes.
FR-M F002	<h3>Scheduling, Time Zones & Monitoring</h3> <ul style="list-style-type: none"> The system should support automated record deletion based on user-defined filters and schedules (daily, weekly, etc.). Users should be able to set their preferred time zones and view all time-related fields accordingly. Daylight Saving Time adjustments should be handled automatically based on the user's time zone. Admins should be able to configure a system calendar to be referenced by date fields where applicable. Debugging tools must be available for events, scripts, and calculations. The product should include monitoring and notification features to alert users/admins about system events or issues. An archiving feature should be considered for long-term data retention or storage optimization. There should be a process load balancer or options to assign specific processes or jobs to run on designated servers.

Sample End User View for Record :

