User Stories

# Manage Visibility of Control Block Items Based on User and Application Data

Type: CONTROL

User Story: Manage Visibility of Control Block Items Based on User and Application Data  
  
Detailed description:   
As a user, I want the system to dynamically manage the visibility of specific items in the control block based on user roles and application data, so that I can have a streamlined and relevant user interface.  
  
Acceptance criteria:  
1. When the form is initialized, the system should:  
 - Maximize the main window.  
 - Navigate to the 'application\_no' field in the control block.  
 - Hide the following items in the control block:  
 - RECEIPT  
 - Merged\_by  
 - merged\_on  
 - CP\_MERGING  
 - Merg\_req\_dt  
 - STATUS  
 - MERG\_SUB\_STATUS  
 - MERG\_APPROVER\_ID  
 - AUTO\_BBU\_FLAG  
 - PENNY\_DROP\_SUCCESS  
 - PENNY\_DROP\_FAIL  
 - If the user ID starts with 'P00' or 'UU', the 'lan\_no' item should be visible.  
 - Check the 'physical\_policy\_bond' status from the 'azbj\_annuity\_prod\_det' table for the current application number:  
 - If 'physical\_policy\_bond' is 'Y', set the 'PHYSICAL\_COPY\_OPTED' item to 'PHYSICAL\_COPY\_OPTED' and make it visible.  
 - If 'physical\_policy\_bond' is 'N', set the 'PHYSICAL\_COPY\_OPTED' item to 'DIGITAL\_COPY\_OPTED' and make it visible.  
  
Definition of Done:  
- The form initializes correctly and performs all specified actions.  
- The visibility of items in the control block is managed as per the user role and application data.  
- The system retrieves and processes data from the 'azbj\_annuity\_prod\_det' table accurately.  
- All acceptance criteria are met without any errors or issues.  
  
DB queries for Table reference CRUD operations only (With Usage):  
```sql  
SELECT physical\_policy\_bond  
INTO v\_physical\_policy\_bond  
FROM azbj\_annuity\_prod\_det  
WHERE appln\_no = :control.application\_no AND ROWNUM = '1' AND physical\_policy\_bond IS NOT NULL;  
```

# View and Manage Receipt, Application, and Policy Details

Type: CONTROL

Detailed description: As a user, I want to view and manage various details related to receipts, applications, and policies within a centralized interface. This interface should allow me to see information such as reconciliation status, event descriptions, proposal numbers, scrutiny numbers, and other relevant details. Additionally, I should be able to perform actions like searching for status, clearing fields, viewing tele-verification history, and downloading information.  
  
Acceptance criteria:  
1. The interface should display the following details:  
 - Reconciliation status  
 - Event description  
 - Application number  
 - Receipt number  
 - Proposal number  
 - Scrutiny number  
 - Current status  
 - Sub-status  
 - Verification status and user-date  
 - Merged details (by whom, on what date, request date, status)  
 - Policy status  
 - Dispatch status and dates  
 - Delivery date  
 - KYC status  
 - Customer type  
 - Family discount  
 - Wealth segment  
 - Existing application details  
 - Various flags (e.g., PIVC, GSIP, SSS, ISNP)  
  
2. The interface should include buttons for the following actions:  
 - View status  
 - Clear fields  
 - View PAN approved details  
 - Cancel  
 - Exit  
 - View tele-verification history  
 - View EIA details  
 - View medical status  
 - Download information  
 - View email log  
 - View DE linked reasons  
 - View requirement details  
 - View UW (NonConfidential) details  
  
3. The interface should ensure that certain fields are only visible or editable based on specific conditions (e.g., flags being set).  
  
Definition of Done:  
- The interface is implemented and tested to ensure all specified details are displayed correctly.  
- All buttons perform their intended actions without errors.  
- Visibility and editability of fields are correctly managed based on the specified conditions.  
- The interface is user-friendly and meets the requirements specified in the acceptance criteria.  
  
DB queries for Table reference CRUD operations only (With Usage):  
- Not applicable as the provided XML content does not include specific SQL queries or table references that can be directly executed in the database.

# Format Receipt Number to 10 Characters with Leading Zeros

Type: CONTROL

Title: Format Receipt Number to 10 Characters with Leading Zeros  
  
Acceptance Criteria:  
1. When the user enters a receipt number and moves to the next field, the system should automatically pad the receipt number with leading zeros to make it 10 characters long.  
2. After padding the receipt number, the system should automatically move the cursor to the next field, which is the property number field.  
  
Definition of Done:  
- The receipt number is consistently formatted to 10 characters with leading zeros.  
- The cursor moves to the property number field after the receipt number is formatted.  
- The functionality is tested and verified to work as expected.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Implement Scrutiny Number Field

Type: CONTROL

Title: Implement Scrutiny Number Field  
  
Acceptance Criteria:  
1. When the user enters a scrutiny number and presses the designated key to move to the next item, the system should automatically navigate to the next item in the form.  
2. The scrutiny number field should accept a maximum of 30 characters.  
3. The scrutiny number field should be left-justified and have a white background with black text.  
4. The prompt for the scrutiny number field should be displayed above the field, aligned to the center, and styled in bold.  
  
Definition of Done:  
1. The scrutiny number field is implemented and visible on the form.  
2. The field accepts input and allows navigation to the next item upon pressing the designated key.  
3. The field and its prompt are styled according to the specified criteria.  
4. The functionality is tested and verified to ensure it meets the acceptance criteria.

# Proposal Number Field Functionality

Type: CONTROL

Title: Proposal Number Field Functionality  
  
Acceptance Criteria:  
- When the user enters a proposal number and presses the key to move to the next item, the focus should shift to the "Scrutiny Number" field.  
- The proposal number field should accept a maximum of 10 characters.  
- The proposal number field should display text in mixed case.  
- The proposal number field should have a white background and black text.  
- The proposal number field should be positioned correctly on the form as specified.  
  
Definition of Done:  
- The proposal number field is implemented and functional.  
- The field accepts a maximum of 10 characters.  
- The field allows mixed case input.  
- The field has the specified visual properties (white background, black text).  
- The field is correctly positioned on the form.  
- The focus shifts to the "Scrutiny Number" field when the user moves to the next item.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Exit Form Without Validation

Type: CONTROL

Title: Exit Form Without Validation  
  
User Story:  
As a user, I want to be able to exit the current form without validating any unsaved changes, so that I can quickly leave the form when necessary.  
  
Acceptance Criteria:  
1. The "Exit" button should be visible and labeled as "Exit".  
2. When the "Exit" button is pressed, the form should close immediately without performing any validation checks on unsaved data.  
3. The form should exit smoothly and return the user to the previous screen or main menu.  
  
Definition of Done:  
- The "Exit" button is implemented and functional.  
- Pressing the "Exit" button closes the form without any validation prompts.  
- The user is returned to the previous screen or main menu upon exiting the form.  
  
Block Name: CONTROL

# Exit Button Navigation to Tele-Verification History

Type: CONTROL

Detailed description: As a user, I want to be able to exit the current form and navigate to the tele-verification history section when I press the "Exit" button, so that I can review the tele-verification history details.  
  
Acceptance criteria:  
- When the "Exit" button is pressed, the system should navigate to the tele-verification history section.  
- The tele-verification history section should be displayed on the screen.  
  
Definition of Done:  
- The "Exit" button is functional and correctly navigates to the tele-verification history section.  
- The tele-verification history section is displayed without any errors.  
- The navigation should be seamless and should not require any additional user actions.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Manage and Display Reason Comments for Application Number

Type: CONTROL

Detailed description: As a user, I want to view and manage the reasons and comments associated with a specific application number so that I can keep track of the details and updates related to the application.  
  
Acceptance criteria:  
1. When the "DE LINKED Reasons" button is pressed, the system should display the "Reasons" section.  
2. The system should navigate to the "Reason Comments" section.  
3. The system should clear any existing data in the "Reason Comments" section.  
4. The system should fetch and display all reason comments related to the specific application number in the "Reason Comments" section, ordered by the comment date.  
5. The system should populate the "Reason Comments" section with the following details for each record:  
 - Reason  
 - Comments  
 - User ID  
 - Comment Date  
6. The system should navigate to the first record in the "Reason Comments" section after populating the data.  
  
Definition of Done:  
- The "DE LINKED Reasons" button is functional and triggers the display of the "Reasons" section.  
- The "Reason Comments" section is navigable and displays the correct data related to the application number.  
- The data in the "Reason Comments" section is cleared before new data is populated.  
- The data fetched from the database is correctly displayed in the "Reason Comments" section, ordered by the comment date.  
- The system navigates to the first record in the "Reason Comments" section after data population.  
  
DB queries for Table reference CRUD operations only(With Usage):  
```sql  
SELECT FROM azbj\_deqc\_reason\_comments   
WHERE APPLN\_NO = :CONTROL.APPLICATION\_NO   
ORDER BY comment\_date;  
```  
- This query is used to fetch all reason comments related to the specific application number, ordered by the comment date.

# View PAN Approval Details

Type: CONTROL

Title: View PAN Approval Details  
  
Acceptance Criteria:  
1. When the "PAN Approved Details" button is pressed, the system should check if the application number is provided.  
2. If the application number is provided, the system should:  
 - Retrieve and display the approval details from the database, including the approved user, user name, approval date, approval time, and status.  
 - Display a message if no details are found for the given application number.  
3. If the application number is not provided, the system should display a message prompting the user to enter the search criteria.  
  
Definition of Done:  
- The "PAN Approved Details" button functionality is implemented and tested.  
- The system retrieves and displays the correct approval details based on the application number.  
- Appropriate messages are displayed when no details are found or when the application number is not provided.  
- The feature is reviewed and approved by stakeholders.  
  
DB queries for Table reference CRUD operations only(With Usage):  
```sql  
SELECT APPROVED\_USER,  
 az\_pk2\_general.getusername(APPROVED\_USER) AS user\_name,  
 TRUNC(CREATE\_DATE) AS apprv\_date,  
 TO\_CHAR(CREATE\_DATE, 'hh:mi:ss am') AS apprv\_time,  
 (CASE  
 WHEN PAN\_STD\_FLAG = 'Y' THEN 'Standard'  
 WHEN PAN\_STD\_FLAG = 'N' THEN 'Non-Standard'  
 WHEN PAN\_STD\_FLAG = 'R' THEN 'Review'  
 END) AS status  
FROM azbj\_supervisor\_appr\_det  
WHERE AGE\_PROOF = 'PC'  
 AND NVL(PAN\_STD\_FLAG, 'NA') <> 'NA'  
 AND application\_no = :application\_no  
ORDER BY CREATE\_DATE DESC;  
```

# View Tele Verification and Central Call History

Type: CONTROL

Title: View Tele Verification and Central Call History  
  
Acceptance Criteria:  
1. When the "Tele verification history" button is pressed, the system should:  
 - Retrieve and display the tele verification history from the `balic\_history\_details` table for the given application number, ordered by feedback time in descending order.  
 - Retrieve and display the central call history from the `azbj\_central\_call\_dtls` table for the given application or proposal number, ordered by modify date in descending order.  
 - Handle any exceptions and display appropriate error messages if data retrieval fails.  
  
Definition of Done:  
- The tele verification history and central call history are displayed correctly when the button is pressed.  
- The data is retrieved from the respective tables and displayed in the correct order.  
- Appropriate error messages are shown if there are any issues during data retrieval.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Retrieve tele verification history:  
 ```sql  
 SELECT user\_id, feedback\_time, feedback, tele\_ver\_status, user\_comment  
 FROM balic\_history\_details  
 WHERE appl\_no = :control.application\_no  
 ORDER BY feedback\_time DESC;  
 ```  
  
- Retrieve central call history:  
 ```sql  
 SELECT request\_no, modify\_date, userid, main\_status, sub\_status, approval\_remarks  
 FROM azbj\_central\_call\_dtls  
 WHERE (proposal\_no = :control.prop\_no OR application\_no = :control.application\_no)  
 ORDER BY modify\_date DESC;  
 ```

# Clear Form and Hide Specific Fields on Clear Button Press

Type: CONTROL

Title: Clear Form and Hide Specific Fields on Clear Button Press  
  
Acceptance Criteria:  
1. When the "Clear" button is pressed, the form should be cleared without validation.  
2. The following fields should be hidden if they are currently visible:  
 - RECEIPT  
 - Merged\_by  
 - merged\_on  
 - CP\_MERGING  
 - Merg\_req\_dt  
 - STATUS  
 - email\_time  
 - email\_id  
 - email\_flag  
 - success\_mails  
 - MERG\_SUB\_STATUS  
 - CKYC\_EDIT  
 - EKYC\_EDIT  
 - MERG\_APPROVER\_ID  
 - AUTO\_BBU\_FLAG  
 - APPROVED\_REASON  
 - EIA\_CASE  
 - diff\_state  
 - AUTO\_DE2\_CASE  
 - INFO\_SHARE\_LETTER  
 - PHOD  
3. The field 'PHOD' should be set to null.  
4. The focus should move to the 'REASON\_COMMENTS.EXIT' field, and the 'WHEN-BUTTON-PRESSED' trigger should be executed.  
  
Definition of Done:  
- The form is cleared without validation when the "Clear" button is pressed.  
- All specified fields are hidden if they are visible.  
- The 'PHOD' field is set to null.  
- The focus moves to the 'REASON\_COMMENTS.EXIT' field, and the 'WHEN-BUTTON-PRESSED' trigger is executed.  
- The functionality is tested and verified to work as expected.

# Exit Button Functionality

Type: CONTROL

Title: Exit Button Functionality  
  
Acceptance Criteria:  
- When the "Exit" button is pressed, the system should navigate to the medical status section.  
- The call history view should be hidden upon pressing the "Exit" button.  
  
Definition of Done:  
- The "Exit" button successfully navigates the user to the medical status section.  
- The call history view is hidden when the "Exit" button is pressed.  
- The functionality is tested and verified to work as expected.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database CRUD operations.

# View Requirement Details by Proposal Number

Type: CONTROL

Title: View Requirement Details by Proposal Number  
  
Acceptance Criteria:  
1. If the proposal number is not entered, an alert should be displayed prompting the user to enter the proposal number.  
2. Upon entering a valid proposal number and pressing the button, the system should:  
 - Clear the current block of any previous data.  
 - Retrieve and display the request types, descriptions, and raised dates associated with the entered proposal number.  
 - The data should be fetched from two different tables (`wip\_azbj\_med\_uw` and `azbj\_med\_uw`) and combined.  
 - The results should be ordered by the `recdstat` field in ascending order.  
3. If any error occurs during the process, the system should handle it gracefully and not crash.  
  
Definition of Done:  
- The button should be functional and perform the described actions.  
- The alert should be displayed if the proposal number is not entered.  
- The data should be correctly retrieved and displayed in the specified order.  
- Error handling should be implemented to manage any unexpected issues.  
  
DB queries for Table reference CRUD operations only (With Usage):  
```sql  
-- Query to fetch data from wip\_azbj\_med\_uw and azbj\_med\_uw tables  
SELECT rec.   
FROM (  
 SELECT a.testno AS REQ\_TYPE,   
 UPPER(TRIM(b.sys\_desc)) AS sys\_desc,   
 a.called\_date AS REQ\_RAISED\_DATE,   
 recdstat  
 FROM wip\_azbj\_med\_uw a  
 JOIN azbj\_system\_constants b ON b.sys\_code = a.testno  
 WHERE b.sys\_type IN ('FR\_REQ', 'MED\_TEST')  
 AND contract\_id = azbj\_pk0\_acc.get\_contract\_id(:control.prop\_no)  
 UNION ALL  
 SELECT a.testno AS REQ\_TYPE,   
 UPPER(TRIM(b.sys\_desc)) AS sys\_desc,   
 a.called\_date AS REQ\_RAISED\_DATE,   
 recdstat  
 FROM azbj\_med\_uw a  
 JOIN azbj\_system\_constants b ON b.sys\_code = a.testno  
 WHERE b.sys\_type IN ('FR\_REQ', 'MED\_TEST')  
 AND contract\_id = azbj\_pk0\_acc.get\_contract\_id(:control.prop\_no)  
) rec  
ORDER BY NVL(recdstat, 'N') ASC;  
```

# Manage Comments and Remarks for Policy and Application Numbers

Type: CONTROL

Detailed description: As a user, I want to be able to view and manage comments related to a specific policy number or application number, so that I can ensure all necessary information is reviewed and updated accordingly.  
  
Acceptance criteria:  
1. If the application number starts with 'AG' and the current state is 'FR-AR' and the substate is 'FR-AR\_Int', the system should:  
 - Navigate to the 'uw' section.  
 - Clear any existing data in the 'uw' section.  
 - Retrieve and display remarks from the `azbj\_bbu\_qc\_ques\_answers` table where the application number matches and the answer is not equal to the standard answer.  
 - Display each remark in a new record within the 'uw' section.  
2. If the proposal number is not provided, the system should:  
 - Display an alert message prompting the user to enter the proposal number.  
 - Halt further processing until the proposal number is entered.  
3. If the proposal number is provided, the system should:  
 - Navigate to the 'uw' section.  
 - Clear any existing data in the 'uw' section.  
 - Retrieve and display comments from the `azbj\_uw\_comments` table where the policy number matches and the flag is 'N'.  
 - Display each comment in a new record within the 'uw' section.  
4. In case of any errors during the process, the system should display the error message.  
  
Definition of Done:  
- The user can successfully view and manage comments related to a specific policy number or application number.  
- The system handles both scenarios (with and without proposal number) as described in the acceptance criteria.  
- Appropriate error messages are displayed in case of any issues.  
- The functionality is tested and verified to ensure it meets the requirements.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Retrieve comments for a specific policy number:  
 ```sql  
 SELECT comments  
 FROM azbj\_uw\_comments  
 WHERE policy\_no = :control.prop\_no  
 AND FLAG = 'N';  
 ```  
  
- Retrieve remarks for a specific application number:  
 ```sql  
 SELECT remarks  
 FROM azbj\_bbu\_qc\_ques\_answers  
 WHERE application\_no = :control.application\_no  
 AND answer <> stp\_answer;  
 ```

# View Scrutiny Status

Type: CONTROL

Title: View Scrutiny Status  
  
Acceptance Criteria:  
1. When the "Scrutiny Status" button is pressed, the system should display the scrutiny status view.  
2. The system should navigate to the scrutiny status block and clear any existing data.  
3. The system should fetch all records from the `azbj\_scrutiny\_del\_det` table where the `application\_no` matches the current application number.  
4. For each record fetched, the system should populate the scrutiny status block with the deletion comment, user ID, and comment date.  
5. The system should iterate through all records and display them in the scrutiny status block.  
  
Definition of Done:  
- The scrutiny status view is displayed upon pressing the "Scrutiny Status" button.  
- The scrutiny status block is cleared and populated with relevant data from the `azbj\_scrutiny\_del\_det` table.  
- All records are iterated and displayed correctly in the scrutiny status block.  
  
DB queries for Table reference CRUD operations only(With Usage):  
```sql  
SELECT   
FROM azbj\_scrutiny\_del\_det   
WHERE application\_no = :control.application\_no;  
```  
- This query is used to fetch all records related to the current application number from the `azbj\_scrutiny\_del\_det` table.

# Navigate to Search Functionality on Key Press

Type: CONTROL

Title: Navigate to Search Functionality on Key Press  
  
Acceptance Criteria:  
- When the user presses the key to move to the next item, the system should automatically navigate to the search functionality for received items.  
- The search functionality should be located in the control block.  
  
Definition of Done:  
- The navigation to the search functionality is triggered by the key press event.  
- The user is able to access the search functionality for received items without any errors.  
- The functionality is tested and verified to ensure smooth navigation.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database-specific queries.

# Exit Email Log Details View

Type: CONTROL

Title: Exit Email Log Details View  
  
User Story:  
As a user, I want to be able to exit the email log details view and return to the main control screen when I press the "Exit" button, so that I can navigate back to the main control screen easily.  
  
Acceptance Criteria:  
1. When the "Exit" button is pressed, the email log details view should be hidden.  
2. The focus should then shift to the main control screen.  
  
Definition of Done:  
1. The "Exit" button is functional and hides the email log details view.  
2. The focus is successfully shifted to the main control screen after the "Exit" button is pressed.  
3. The feature is tested and verified to work as expected.  
  
Block Name: CONTROL

# Download Activity Tracker Data to Excel

Type: CONTROL

Detailed description: As a user, I want to be able to download the activity tracker data into an Excel file so that I can easily view and analyze the data offline.  
  
Acceptance criteria:  
1. When the "Download" button is pressed, the system should navigate to the activity tracker data block.  
2. The system should count the number of records in the activity tracker data block.  
3. The system should create a new Excel file and add a worksheet to it.  
4. The first row of the worksheet should contain the headers: "Activity", "ActivityUser", and "ActivityDate".  
5. Subsequent rows should contain the corresponding data from the activity tracker:  
 - "Activity" column should display the status.  
 - "ActivityUser" column should display the user.  
 - "ActivityDate" column should display the date in the format 'DD-Mon-YYYY HH24:MI:SS'.  
6. The system should save the Excel file to the local path 'c:\' with a filename based on the current date and time.  
7. The system should notify the user with the file path once the file is saved.  
8. The system should open the saved Excel file automatically for the user to view.  
  
Definition of Done:  
- The "Download" button functionality is implemented and tested.  
- The Excel file is created with the correct data and format.  
- The file is saved to the specified path and the user is notified.  
- The saved file opens automatically for the user to view.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided logic does not include direct database CRUD operations.

# Validate and Retrieve Application Number based on BI Number

Type: CONTROL

Title: Validate and Retrieve Application Number based on BI Number  
  
Acceptance Criteria:  
1. When a BI Number is entered, the system should check if the BI Number is not null.  
2. If the BI Number is not null, the system should query the `azbj\_proposal\_appln\_det\_ext` table to find the corresponding application number.  
3. If an application number is found, it should be displayed in the application number field.  
4. If no application number is found, the application number field should remain empty.  
  
Definition of Done:  
- The BI Number validation logic is implemented and tested.  
- The system correctly retrieves and displays the application number if it exists.  
- The application number field remains empty if no corresponding application number is found.  
- All acceptance criteria are met and verified through testing.  
  
DB queries for Table reference CRUD operations only(With Usage):  
```sql  
SELECT APPLN\_NO  
INTO V\_APPLN\_NO  
FROM azbj\_proposal\_appln\_det\_ext  
WHERE BI\_NO = :CONTROL.BI\_NUMBER;  
```

# View Proposal Medical Status

Type: CONTROL

Title: View Proposal Medical Status  
  
Acceptance Criteria:  
1. When the "Proposal Medical Status" button is pressed, the system should navigate to the "Customer Call History" section.  
2. The system should clear any existing data in the "Customer Call History" section.  
3. The system should retrieve records from the `azbj\_phub\_req\_feedback` table where the `prop\_no` matches the current proposal number and the `sub\_status` is not in ('AD', 'AC', 'DA', 'MED').  
4. The retrieved records should be ordered by `date\_of\_follow` and `follow\_time`.  
5. For each retrieved record, the system should populate the following fields in the "Customer Call History" section:  
 - `call\_date` with the value of `sysdt`  
 - `caller\_id` with the value of `caller\_id`  
 - `status` with the value of `status`  
 - `sub\_status` with the value of `sub\_status`  
 - `proposal\_medical\_status` with the value of `medical\_status`  
6. After processing all records, the system should display the first record in the "Customer Call History" section.  
7. If an error occurs during this process, the system should display an error message with the details of the error.  
  
Definition of Done:  
- The "Proposal Medical Status" button functionality is implemented as described.  
- The system correctly navigates to and updates the "Customer Call History" section based on the retrieved data.  
- Error handling is implemented to display appropriate error messages.  
- The feature is tested and verified to ensure it meets the acceptance criteria.  
  
DB queries for Table reference CRUD operations only(With Usage):  
```sql  
-- Retrieve records from azbj\_phub\_req\_feedback table  
SELECT   
FROM azbj\_phub\_req\_feedback  
WHERE prop\_no = :control.prop\_no  
 AND sub\_status NOT IN ('AD', 'AC', 'DA', 'MED')  
ORDER BY date\_of\_follow, follow\_time;  
```

# Search and Display Application Status

Type: CONTROL

User Story: Search and Display Application Status  
  
Detailed Description:  
As a user, I want to search for application details using various criteria such as application number, receipt number, proposal number, scrutiny number, or BI number. Upon entering the search criteria, the system should validate the input and display relevant information about the application, including its current status, associated flags, and other pertinent details.  
  
Acceptance Criteria:  
1. The system should validate that at least one search criterion is provided.  
2. If the application number is provided, it should be padded to 10 characters.  
3. The system should check for the existence of the application in various tables and retrieve associated details.  
4. If the application is found, the system should display the following:  
 - Current status of the application.  
 - Flags indicating specific conditions (e.g., EIA case, Auto DE2 case, etc.).  
 - Details about the policy dispatch status.  
 - Information about any associated competition or agent details.  
 - Any additional comments or remarks related to the application.  
5. If no records are found for the provided search criteria, an alert should be displayed to the user.  
  
Definition of Done:  
- The search functionality is implemented and integrated into the system.  
- The system validates the input and displays appropriate messages or results based on the search criteria.  
- All relevant details and flags are displayed correctly for the found application.  
- The system handles exceptions and errors gracefully, providing meaningful feedback to the user.  
- The functionality is tested and verified to ensure it meets the acceptance criteria.  
  
DB Queries for Table Reference CRUD Operations:  
- Retrieve application details from `azbj\_phub\_tracker`, `azbj\_phub\_scrutiny\_prop`, `azbj\_phub\_status\_tracker`, and other relevant tables.  
- Check for specific conditions and flags in tables like `azbj\_eia\_details`, `azbj\_annuity\_prod\_det`, `azbj\_random\_med\_call\_dtls`, etc.  
- Update or insert records as necessary based on the search results and conditions.  
  
Explanation of Oracle Form Logic:  
- The logic involves validating the input search criteria and ensuring at least one criterion is provided.  
- It includes padding the application number to a specific length if provided.  
- The system performs multiple checks and queries across various tables to gather all relevant information about the application.  
- Based on the retrieved data, the system sets various properties and flags to display the appropriate information to the user.  
- The logic also handles exceptions and provides feedback to the user if no records are found or if any errors occur during the process.

# View Email Log History for Application or Proposal Number

Type: CONTROL

Title: View Email Log History for Application or Proposal Number  
  
Acceptance Criteria:  
1. If the application number or proposal number is not provided, the system should display a warning message: "Please enter Application Number or Proposal Number and press Status to proceed."  
2. Upon providing the application or proposal number, the system should:  
 - Retrieve the contract ID associated with the provided proposal number.  
 - Fetch the email ID related to the contract ID.  
 - Display the email log history, including:  
 - The type of document (Policy Bond or EIA Letter).  
 - The status of the email (Sent Success or Sent Failure).  
 - The timestamp of when the email was updated.  
 - Count and display the number of successful emails sent.  
  
Definition of Done:  
- The user can input either an application number or a proposal number.  
- The system retrieves and displays the email log history based on the provided number.  
- The system shows a warning message if the required number is not provided.  
- The email log history includes document type, email status, and timestamp.  
- The count of successful emails is displayed.  
  
DB queries for Table reference CRUD operations only (With Usage):  
- Retrieve contract ID:  
 ```sql  
 SELECT azbj\_pk0\_acc.get\_contract\_id(:control.prop\_no) INTO v\_contr\_id FROM dual;  
 ```  
- Fetch email ID:  
 ```sql  
 SELECT customer.azbj\_get\_email\_id(v\_contr\_id) INTO v\_mail\_id FROM dual;  
 ```  
- Fetch email log history:  
 ```sql  
 SELECT DECODE(table\_data, '1', 'Policy Bond', '2', 'EIA Letter') table\_data,  
 mail\_flag,  
 updated\_on  
 FROM (SELECT 1 table\_data,  
 DECODE(mail\_yn, 'Y', 'Sent Success', 'N', 'Sent Failure') mail\_flag,  
 updated\_on  
 FROM azbj\_policy\_copy\_mailer  
 WHERE policy\_ref = :control.prop\_no  
 UNION  
 SELECT 2 table\_data,  
 DECODE(send\_flag, 'Y', 'Sent Success', 'N', 'Sent Failure') mail\_flag,  
 mail\_send\_time updated\_on  
 FROM azbj\_eia\_letter\_gen\_dtls  
 WHERE policy\_no = :control.prop\_no)  
 ORDER BY updated\_on DESC;  
 ```

# View EIA Details on Button Press

Type: CONTROL

Title: View EIA Details on Button Press  
  
Acceptance Criteria:  
1. When the "EIA Details" button is pressed, the system should display the EIA details view.  
2. The system should navigate to the EIA status screen.  
3. The system should clear any existing data in the EIA block.  
4. The system should fetch and display the following details from the `azbj\_eia\_details` table for the given application number:  
 - EIA Status (SCRU)  
 - EIA Account Type  
 - Create User  
 - Create Date  
5. If no data is found for the SCRU status, the EIA status should be set to 'N' and other fields should be cleared.  
6. The system should fetch and display the following details from the `azbj\_eia\_details` table for the given application number:  
 - EIA Status (DEQC)  
 - EIA Account Type  
 - Create User  
 - Create Date  
7. If no data is found for the DEQC status, the EIA status should be set to 'N' and other fields should be cleared.  
8. The system should fetch and display the following details from the `azbj\_eia\_details` table for the given application number:  
 - EIA Status (BBU)  
 - EIA Account Type  
 - Create User  
 - Create Date  
9. If no data is found for the BBU status, the EIA status should be set to 'N' and other fields should be cleared.  
10. The system should fetch and display the following details from the `azbj\_eia\_xml\_file\_dtls` table for the given application number:  
 - EIA Account Creation Status  
 - Create User  
 - Create Time  
11. If no data is found for the account creation status, the EIA status should be set to 'NA' and other fields should be cleared.  
12. The system should fetch and display the following details from the `azbj\_eia\_account\_details` table for the given application number:  
 - EIA Account Upload Status  
 - Status  
 - Create User  
 - Create Date  
13. If no data is found for the account upload status, the EIA status should be set to 'NA' and other fields should be cleared.  
14. The system should fetch and display the following details from the `azbj\_eia\_xml\_policy\_det` table for the given policy reference:  
 - Policy XML Status  
 - Create User  
 - Create Time  
15. If no data is found for the policy XML status, the EIA status should be set to 'NA' and other fields should be cleared.  
16. The system should fetch and display the following details from the `azbj\_eia\_policy\_details` table for the given application number:  
 - Policy XML Upload Status  
 - Status  
 - Create User  
 - Create Date  
17. If no data is found for the policy XML upload status, the EIA status should be set to 'NA' and other fields should be cleared.  
18. The system should fetch and display the following details from the `azbj\_logistics\_shipment\_detail` table for the given policy number:  
 - Logistics Status  
 - Create User  
 - Create Date  
19. If no data is found for the logistics status, the EIA status should be set to 'N' and other fields should be cleared.  
20. If any error occurs during the process, an appropriate error message should be displayed.  
  
Definition of Done:  
- The "EIA Details" button should trigger the display of the EIA details view.  
- All relevant EIA details should be fetched and displayed correctly based on the application number and policy reference.  
- Appropriate error handling should be in place to display error messages if any issues occur during data retrieval.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- The following queries are used to fetch data from the respective tables:  
 ```sql  
 SELECT 'Y', EIA\_ACCOUNT\_TYPE, CREATE\_USER, CREATE\_DATE  
 INTO :EIA.EIA\_STATUS\_SCR, :EIA.EIA\_DESC\_SCR, :EIA.EIA\_USER\_SCR, :EIA.EIA\_DATE\_SCR  
 FROM azbj\_eia\_details  
 WHERE APPLICATION\_NO = :control.application\_no AND scrutiny\_no IS NOT NULL AND UPPER(FLAG) = 'SCRU';  
  
 SELECT 'Y', EIA\_ACCOUNT\_TYPE, CREATE\_USER, CREATE\_DATE  
 INTO :EIA.EIA\_STATUS\_QC, :EIA.EIA\_DESC\_QC, :EIA.EIA\_USER\_QC, :EIA.EIA\_DATE\_QC  
 FROM azbj\_eia\_details  
 WHERE APPLICATION\_NO = :control.application\_no AND UPPER(FLAG) = 'DEQC';  
  
 SELECT STATUS\_FLG, EIA\_ACCOUNT\_TYPE, CREATE\_USER, CREATE\_DATE  
 INTO :EIA.EIA\_STATUS\_BBU, :EIA.EIA\_DESC\_BBU, :EIA.EIA\_USER\_BBU, :EIA.EIA\_DATE\_BBU  
 FROM (SELECT top\_indicator STATUS\_FLG, EIA\_ACCOUNT\_TYPE, CREATE\_USER, CREATE\_DATE  
 FROM azbj\_eia\_details  
 WHERE APPLICATION\_NO = :control.application\_no AND UPPER(FLAG) = 'BBU'  
 ORDER BY CREATE\_DATE DESC)  
 WHERE ROWNUM = 1;  
  
 SELECT 'Y', NULL, CREATE\_USER, CREATE\_TIME  
 INTO :EIA.EIA\_STATUS\_ACC\_CREATION, :EIA.EIA\_DESC\_ACC\_CREATION, :EIA.EIA\_USER\_ACC\_CREATION, :EIA.EIA\_DATE\_ACC\_CREATION  
 FROM azbj\_eia\_xml\_file\_dtls  
 WHERE APPLICATION\_NO = :control.application\_no;  
  
 SELECT STATUS\_FLG, STATUS, CREATE\_USER, CREATE\_DATE  
 INTO :EIA.EIA\_STATUS\_ACC\_UPLOAD, :EIA.EIA\_DESC\_ACC\_UPLOAD, :EIA.EIA\_USER\_ACC\_UPLOAD, :EIA.EIA\_DATE\_ACC\_UPLOAD  
 FROM (SELECT 'Y' STATUS\_FLG, STATUS, CREATE\_USER, CREATE\_DATE  
 FROM azbj\_eia\_account\_details  
 WHERE APPLICATION\_NO = :control.application\_no  
 ORDER BY CREATE\_DATE ASC)  
 WHERE ROWNUM = 1;  
  
 SELECT 'Y', NULL, CREATE\_USER, CREATE\_TIME  
 INTO :EIA.EIA\_STATUS\_XML, :EIA.EIA\_DESC\_XML, :EIA.EIA\_USER\_XML, :EIA.EIA\_DATE\_XML  
 FROM azbj\_eia\_xml\_policy\_det  
 WHERE POLICY\_REF = :control.prop\_no;  
  
 SELECT STATUS\_FLG, STATUS, CREATE\_USER, CREATE\_DT  
 INTO :EIA.EIA\_STATUS\_XML\_UPLOAD, :EIA.EIA\_DESC\_XML\_UPLOAD, :EIA.EIA\_USER\_XML\_UPLOAD, :EIA.EIA\_DATE\_XML\_UPLOAD  
 FROM (SELECT 'Y' STATUS\_FLG, STATUS, CREATE\_USER, CREATE\_DT  
 FROM azbj\_eia\_policy\_details  
 WHERE APPLICATION\_NO = :control.application\_no  
 ORDER BY CREATE\_DT DESC)  
 WHERE ROWNUM = 1;  
  
 SELECT 'Y', NULL, SHIPMENT\_CREATE\_USER, TRUNC(SHIPMENT\_CREATE\_DATE)  
 INTO :EIA.EIA\_STATUS\_LOGISTICS, :EIA.EIA\_DESC\_LOGISTICS, :EIA.EIA\_USER\_LOGISTICS, :EIA.EIA\_DATE\_LOGISTICS  
 FROM azbj\_logistics\_shipment\_detail  
 WHERE policy\_no = :control.prop\_no AND shipment\_id LIKE '704%' AND ROWNUM = 1;  
 ```

# View Delivery Date Information

Type: CDA

Title: View Delivery Date Information  
  
Acceptance Criteria:  
- The delivery date field should be displayed in the designated section of the application.  
- The delivery date field should be read-only and not allow any updates or insertions.  
- The delivery date field should be right-justified and have a maximum length of 50 characters.  
- The field should be visually styled with a white background, black text, and use the Tahoma font.  
  
Definition of Done:  
- The delivery date field is visible in the specified section.  
- The field is read-only and cannot be edited by the user.  
- The field adheres to the specified visual styling and formatting requirements.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific SQL queries or table references.

# Task Assignment Management

Type: BLK\_RI\_ASSIGNED

Title: Task Assignment Management  
  
Acceptance Criteria:  
1. The system should allow the user to input the name of the person to whom the task is assigned.  
2. The system should allow the user to input the date when the task was assigned.  
3. The system should provide a button to exit the assignment interface.  
  
Definition of Done:  
1. The user can successfully input and save the name of the person assigned to the task.  
2. The user can successfully input and save the date when the task was assigned.  
3. The user can exit the assignment interface using the provided button.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database-specific queries or operations.

# Exit Button Functionality

Type: BLK\_RI\_ASSIGNED

Title: Exit Button Functionality  
  
Acceptance Criteria:  
1. When the "Exit" button is pressed, the current view should be hidden.  
2. The focus should then move to the search control screen.  
  
Definition of Done:  
- The "Exit" button is functional and performs the required actions.  
- The current view is hidden upon pressing the "Exit" button.  
- The application navigates to the search control screen successfully.  
- The functionality is tested and verified to work as expected.

# Display Customer Call History

Type: CUST\_CALL\_HISTORY

Title: Display Customer Call History  
  
Acceptance Criteria:  
1. The call history should display the following fields:  
 - Call Date  
 - Caller ID  
 - Status  
 - Sub Status  
 - Proposal Medical Status  
2. Each field should be displayed in a readable format with appropriate labels.  
3. The call history should be displayed in a scrollable section that can show up to 4 records at a time.  
4. The fields should be read-only and not allow any modifications.  
  
Definition of Done:  
1. The call history section is implemented and displays the required fields.  
2. The section is scrollable and shows up to 4 records at a time.  
3. All fields are read-only and properly labeled.  
4. The implementation is tested and verified to ensure it meets the acceptance criteria.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Display Call History Details

Type: CENT\_CALL\_HIST

Title: Display Call History Details  
  
Acceptance Criteria:  
1. The call history should display the following fields:  
 - Request Number  
 - Main Status  
 - Sub Status  
 - Approval Remarks  
 - Caller ID  
 - Call Time  
2. Each field should be displayed with appropriate labels and should be read-only.  
3. The call history should be displayed in a grid format with a maximum of 7 records visible at a time.  
4. The grid should have a scrollbar to navigate through the records if there are more than 7 records.  
  
Definition of Done:  
1. The call history grid is implemented and displays the specified fields.  
2. The fields are read-only and properly labeled.  
3. The grid can display up to 7 records at a time and includes a scrollbar for navigation.  
4. The feature is tested and verified to ensure it meets the acceptance criteria.  
  
DB queries for Table reference CRUD operations only (With Usage):  
- Not applicable as the provided XML content does not include specific SQL queries or table references.

# View Requirement Details

Type: REQ

Title: View Requirement Details  
  
Acceptance Criteria:  
1. The requirement details should be displayed in a section with the following fields:  
 - Requirement Type (REQ\_TYPE)  
 - Requirement Description (REQ\_DESC)  
 - Raise Date (RAISE\_DATE)  
2. The fields should be read-only and not allow any updates or insertions.  
3. The fields should be properly aligned and justified as per the design specifications.  
4. The section should have a scrollbar to navigate through the records if there are more than six records.  
  
Definition of Done:  
1. The requirement details section is implemented and displays the fields as specified.  
2. The fields are read-only and do not allow any modifications.  
3. The alignment and justification of the fields match the design specifications.  
4. The scrollbar is functional and allows navigation through the records.  
5. The implementation is tested and verified to meet the acceptance criteria.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific SQL queries or table references.

# Display Approval Details

Type: BLK\_PAN\_APPROVE\_DET

Detailed description: As a user, I want to view the details of approval records in a structured format so that I can easily understand the approval information.  
  
Acceptance criteria:  
1. The system should display the following fields for each approval record:  
 - Approval ID  
 - Approval Name  
 - Approval Date  
 - Approval Time  
 - Approval Decision  
2. The fields should be displayed in a grid format with a maximum of 8 records visible at a time.  
3. There should be a scrollbar to navigate through the records if there are more than 8 records.  
4. The system should provide an "Exit" button to close the approval details view.  
  
Definition of Done:  
- The approval details view is implemented and displays the required fields.  
- The view supports scrolling for more than 8 records.  
- The "Exit" button is functional and closes the view.  
- The user interface is tested and verified for correctness and usability.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Exit PAN Approved Details and Return to Main Hub

Type: BLK\_PAN\_APPROVE\_DET

Title: Exit PAN Approved Details and Return to Main Hub  
  
Acceptance Criteria:  
- When the "EXIT" button is pressed, the current view of the PAN Approved Details should be hidden.  
- The main hub details view should be displayed.  
- The focus should be set to the main control item in the hub details view.  
  
Definition of Done:  
- The "EXIT" button successfully hides the PAN Approved Details view.  
- The main hub details view is displayed without any errors.  
- The focus is correctly set to the main control item in the hub details view.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as there are no CRUD operations mentioned in the provided XML content.

# Display and Interaction of UW Section in Requirement Despatch Module

Type: UW

Detailed description: As a user, I want to view and interact with the "UW" section within the "Requirement Despatch" module, so that I can see the relevant details displayed in a structured format.  
  
Acceptance criteria:  
1. The "UW" section should display up to 6 records at a time.  
2. The section should have a scrollbar to navigate through the records.  
3. The "UW" field should be right-justified, with a maximum length of 2000 characters.  
4. The "UW" field should have a white background and black text.  
5. The "UW" field should not allow insertions or updates.  
6. The "UW" field should be displayed on the "CAN\_HUB\_DETAILS" canvas.  
7. The "Requirement Despatch" window should not allow resizing, maximizing, or minimizing.  
8. The window should not show horizontal or vertical scrollbars.  
  
Definition of Done:  
- The "UW" section is implemented and displays up to 6 records.  
- The scrollbar is functional and allows navigation through the records.  
- The "UW" field is displayed with the specified formatting and restrictions.  
- The "Requirement Despatch" window adheres to the specified constraints.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- No specific DB queries provided in the XML content.

# Tele-Verification History Management

Type: TELE\_VERIFICATION\_HIS

Title: Tele-Verification History Management  
  
Acceptance Criteria:  
1. The system should allow the user to input the following details:  
 - User ID  
 - Date and Time of verification  
 - Tele-verification comments  
 - Tele-verification status  
 - User comments  
2. The date and time should be recorded in the format "dd-mon-yyyy hh:mi:ss".  
3. The tele-verification comments should allow a maximum length of 700 characters.  
4. The tele-verification status should allow a maximum length of 100 characters.  
5. The user comments should allow a maximum length of 300 characters.  
6. The tele-verification history should be displayed in a scrollable view with a maximum of 10 records displayed at a time.  
  
Definition of Done:  
- The user can successfully input and save tele-verification history details.  
- The tele-verification history is displayed in a scrollable view with a maximum of 10 records at a time.  
- The date and time are displayed in the correct format.  
- The input fields for tele-verification comments, status, and user comments adhere to the specified maximum lengths.  
- The tele-verification history can be reviewed and audited as needed.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include specific SQL queries or table references.

# View Scrutiny Status Details

Type: BLK\_SCRUTINY\_STATUS

Detailed description: As a user, I want to view the scrutiny status details, including the reason, user ID, and comment date, so that I can understand the context and history of the scrutiny process.  
  
Acceptance criteria:  
1. The scrutiny status details should display the following fields:  
 - Reason: A text field that shows the reason for the scrutiny.  
 - User ID: A text field that shows the ID of the user who made the comment.  
 - Comment Date: A date-time field that shows when the comment was made.  
2. The fields should be read-only and not allow any modifications.  
3. There should be an "Exit" button that allows the user to close the scrutiny status view.  
  
Definition of Done:  
- The scrutiny status details are displayed correctly with the specified fields.  
- The fields are read-only and cannot be edited by the user.  
- The "Exit" button functions correctly and closes the scrutiny status view when clicked.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific CRUD operations or SQL queries.

# Exit Button Functionality

Type: BLK\_SCRUTINY\_STATUS

Title: Exit Button Functionality  
  
Acceptance Criteria:  
1. When the exit button is pressed, the current view should be hidden.  
2. The system should then navigate to the status tracking screen.  
  
Definition of Done:  
- The exit button is functional and correctly hides the current view.  
- The system successfully navigates to the status tracking screen upon pressing the exit button.  
- The feature is tested and verified to work as expected.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database CRUD operations.

# Manage Reason Comments, User IDs, and Comment Dates

Type: REASON\_COMMENTS

Detailed description: As a user, I want to be able to input and manage reason comments, user IDs, and comment dates for specific records, so that I can keep track of the reasons and comments associated with each record.  
  
Acceptance criteria:  
1. The system should allow the user to input a reason with a maximum length of 100 characters.  
2. The system should allow the user to input comments with a maximum length of 500 characters.  
3. The system should allow the user to input a user ID with a maximum length of 50 characters.  
4. The system should allow the user to input a comment date in the format "DD/MM/YY HH:MI AM".  
5. The system should provide a button to exit the form.  
6. The system should provide a button to view data enrichment details.  
  
Definition of Done:  
1. The user can successfully input and save reason comments, user IDs, and comment dates.  
2. The input fields for reason, comments, user ID, and comment date should be validated for their respective maximum lengths and format.  
3. The exit button should close the form without saving any changes.  
4. The "View Data Enrichment" button should navigate the user to the data enrichment details section.  
5. All functionalities should be tested and verified to work as expected.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include specific SQL queries or table references.

# Exit Button Functionality in Reason Comments Section

Type: REASON\_COMMENTS

Title: Exit Button Functionality in Reason Comments Section  
  
Acceptance Criteria:  
1. When the "Exit" button is pressed, the "Reason Comments" section should be hidden.  
2. The focus should then move to the "Received Search" control.  
  
Definition of Done:  
- The "Exit" button successfully hides the "Reason Comments" section.  
- The focus is correctly set to the "Received Search" control after the "Reason Comments" section is hidden.  
- The functionality is tested and verified to work as expected.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database CRUD operations.

# View Data Enrichment Details

Type: REASON\_COMMENTS

Title: View Data Enrichment Details  
  
Acceptance Criteria:  
1. When the user presses the "View Data Enrichment" button, the system should check if a parameter list named 'params' already exists.  
2. If the parameter list exists, it should be destroyed.  
3. A new parameter list named 'params' should be created.  
4. The parameter list should include the 'APPLICATION\_NO' from the current context and a 'call\_form\_name' with the value 'STATUS\_QUERY\_SCREEN'.  
5. The system should then open a new form named 'azbj\_data\_enrich\_form' with the provided parameters.  
  
Definition of Done:  
- The "View Data Enrichment" button is functional and triggers the described actions.  
- The parameter list is correctly managed (created, destroyed, and populated).  
- The new form 'azbj\_data\_enrich\_form' opens with the correct parameters when the button is pressed.  
- The feature is tested and verified to work as expected in the user interface.

# Manage Event Description Visibility and Content Based on Proposal Status

Type: AZBJ\_PHUB\_TRACKER

Title: Manage Event Description Visibility and Content Based on Proposal Status  
  
Acceptance Criteria:  
1. When the status of the proposal is 'PROPOSAL UPDATED':  
 - Increment a global counter by 1.  
 - If the event description field is not visible, make it visible.  
 - Populate the event description field with the description of the activity based on the event number, proposal number, and truncated date.  
2. When the status of the proposal is not 'PROPOSAL UPDATED':  
 - If the event description field is visible, make it invisible.  
 - Decrement the global counter by 1.  
 - Ensure the global counter does not go below 0.  
  
Definition of Done:  
- The event description field visibility and content are correctly managed based on the proposal status.  
- The global counter is accurately incremented or decremented as per the conditions.  
- The system handles exceptions gracefully without crashing.  
  
DB queries for Table reference CRUD operations only (With Usage):  
- Not applicable as the provided logic does not include direct CRUD operations on the database.

# Display EIA Process Status and Details

Type: EIA

Title: Display EIA Process Status and Details  
  
Acceptance Criteria:  
1. The system should display the status, description, user, and date for each EIA process step.  
2. The information should be read-only and not allow any modifications.  
3. The status and details should be visually distinct and easy to read.  
  
Definition of Done:  
1. The user interface displays the status, description, user, and date for each EIA process step.  
2. The information is presented in a read-only format.  
3. The visual attributes (such as font, color, and spacing) are consistent and enhance readability.  
4. The "Exit" button is functional and allows the user to exit the interface.  
  
DB queries for Table reference CRUD operations only (With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.

# Exit Button Functionality

Type: EIA

Title: Exit Button Functionality  
  
Acceptance Criteria:  
- When the "Exit" button is pressed, the current view should be hidden.  
- The focus should then shift to the search control screen.  
  
Definition of Done:  
- The "Exit" button is functional and performs the required actions.  
- The current view is hidden upon pressing the "Exit" button.  
- The search control screen is displayed and focused after the "Exit" button is pressed.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any database CRUD operations.

# View Email Log History

Type: EMAIL\_LOG\_HISTORY

Detailed description: As a user, I want to view the history of email logs related to policy bonds or EIA letters, so that I can track the status and details of sent emails.  
  
Acceptance criteria:  
1. The system should display the following details for each email log entry:  
 - E-Mail Status  
 - Date and Time of E-Mail Sent  
 - Registered E-Mail ID  
 - Type of Letter (Policy Bond or EIA Letter)  
 - Number of Successful Mails  
  
2. The email log entries should be displayed in a tabular format with a maximum of 7 records visible at a time.  
  
3. The fields should be read-only except for the E-Mail Status, which should be editable.  
  
Definition of Done:  
- The email log history is displayed with all the required fields.  
- The E-Mail Status field is editable, while other fields are read-only.  
- The table displays up to 7 records at a time.  
- The user interface is intuitive and matches the design specifications.  
- The feature is tested and verified to ensure it meets the acceptance criteria.  
  
DB queries for Table reference CRUD operations only(With Usage):  
- Not applicable as the provided XML content does not include any specific database queries.