Test Cases - Uploaded File: axis hr whatsapp bot - feature list.xlsx

User Story 1

As an employee, I want to authenticate myself using an OTP sent to my official email, so that I can securely access HR-related information via WhatsApp.

Test Cases:

Test Case 1:

Positive Test Cases: **1.** **Authentication with OTP:** **Description:** Verify that the employee can authenticate using OTP sent to the official email. **Preconditions:** Employee has a registered official email. **Steps:** 1. Employee enters their official email. 2. Employee requests OTP. 3. Employee enters OTP received on the official email. **Expected Results:** Employee is successfully authenticated and can access HR-related information via WhatsApp. **Priority:** High ### Validation Test Cases: **1.** **Incorrect Email:** **Description:** Verify that authentication fails if an incorrect email is entered. **Preconditions:** Employee has a registered official email. **Steps:** 1. Employee enters an incorrect email. 2. Employee requests OTP. **Expected Results:** Authentication fails, and the employee is not able to proceed. **Priority:** Medium **2.** **Invalid OTP:** **Description:** Verify that authentication fails if an invalid OTP is entered. **Preconditions:** Employee has a registered official email. **Steps:** 1. Employee enters their official email. 2. Employee requests OTP. 3. Employee enters an invalid OTP. **Expected Results:** Authentication fails, and the employee is not able to proceed. **Priority:** Medium ### UI/UX Test Cases: **1.** **Email Field Validation:** **Description:** Verify that the email field has proper validation. **Preconditions:** Employee is on the authentication screen. **Steps:** 1. Employee leaves the email field blank and tries to proceed. **Expected Results:** An error message should be displayed prompting the user to enter a valid email address. **Priority:** Medium **2.** **OTP Input Field Design:** **Description:** Verify that the OTP input field is user-friendly. **Preconditions:** Employee receives the OTP on the official email. **Steps:** 1. Employee enters the OTP in the input field. **Expected Results:** The OTP input field should be easy to use and clearly visible. **Priority:** Low ### Performance Test Cases: **1.** **Response Time:** **Description:** Measure the time taken for OTP delivery and authentication. **Preconditions:** Employee requests OTP for authentication. **Steps:** 1. Measure the time taken for OTP delivery. 2. Measure the time taken for OTP authentication. **Expected Results:** OTP delivery and authentication should be within acceptable time limits. **Priority:** High ### Security Test Cases: **1.** **OTP Expiry Time:** **Description:** Verify that the OTP expires after a certain time period. **Preconditions:** Employee requests OTP for authentication. **Steps:** 1. Wait for the OTP to expire. 2. Try to use the expired OTP for authentication. **Expected Results:** The expired OTP should not be accepted for authentication. **Priority:** High ### Combination Test Cases: **1.** **Incorrect Email and Invalid OTP:** **Description:** Verify the scenario when both incorrect email and invalid OTP are entered. **Preconditions:** Employee has a registered official email. **Steps:** 1. Employee enters an incorrect email. 2. Employee requests OTP. 3. Employee enters an invalid OTP. **Expected Results:** Authentication fails due to both incorrect email and invalid OTP. **Priority:** Medium

User Story 2

As an employee, I want to navigate through menu-based selection system in WhatsApp, so that I can easily find answers to HR-related queries without typing free-text inputs.

Test Cases:

Test Case 1:

Positive Test Cases: **1.** **Test Case ID:** POS_TC_001 **Description:** Verify employee can navigate through menu-based selection system in WhatsApp. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate to the HR-related menu. 2. Select a specific category (e.g., Leave, Payroll, Benefits). 3. Choose a sub-category within the selected category. 4. Navigate back to the main menu. **Expected Result:** The employee should be able to easily navigate through the menu system and find answers to HR-related queries. **2.** **Test Case ID:** POS_TC_002 **Description:** Verify employee can view detailed information for each menu option. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate to the HR-related menu. 2. Select a specific category (e.g., Leave). 3. View detailed information provided for each sub-category (e.g., Annual Leave, Sick Leave). **Expected Result:** The employee should be able to view detailed information for each menu option within the selected category. ### Validation Test Cases: **1.** **Test Case ID:** VAL_TC_001 **Description:** Verify error message is displayed when employee selects an invalid menu option. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate to the HR-related menu. 2. Select an invalid menu option. **Expected Result:** An error message should be displayed indicating the selected option is invalid. **2.** **Test Case ID:** VAL_TC_002 **Description:** Verify error message is displayed when employee tries to navigate without selecting a menu option. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate to the HR-related menu. 2. Try to navigate without selecting any menu option. **Expected Result:** An error message should be displayed prompting the employee to select a menu option. ### UI/UX Test Cases: **1.** **Test Case ID:** UI_TC_001 **Description:** Verify menu options are clearly labeled and easy to navigate. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and observe the HR-related menu options. **Expected Result:** Menu options should be clearly labeled and easy to understand for the employee. **2.** **Test Case ID:** UI_TC_002 **Description:** Verify consistency in design and layout of the menu system. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Navigate through different categories and sub-categories in the menu system. **Expected Result:** The design and layout of the menu system should be consistent throughout. ### Performance Test Cases: **1.** **Test Case ID:** PER_TC_001 **Description:** Verify the response time when navigating through the menu system. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate through different menu options. **Expected Result:** The response time for navigating through menu options should be fast and efficient. ### Security Test Cases: **1.** **Test Case ID:** SEC_TC_001 **Description:** Verify menu options are accessible only to authorized employees. **Preconditions:** Employee is logged into WhatsApp. **Steps:** 1. Check if unauthorized employees can access the HR-related menu. **Expected Result:** Menu options should be accessible only to authorized employees with proper permissions. ### Combination Test Cases: **1.** **Test Case ID:** COMB_TC_001 **Description:** Verify employee can navigate through multiple levels of menu options. **Preconditions:** Employee is logged into WhatsApp and has access to the HR-related menu system. **Steps:** 1. Open WhatsApp and navigate to a category. 2. Select a sub-category within the category. 3. Navigate back to the main menu. 4. Select a different category and repeat the process. **Expected

Result:** The employee should be able to navigate through multiple levels of menu options without any issues.

User Story 3

As a vendor (Focaloid), I want to add predefined HR-related queries and options provided by Axis, so that the employees can navigate through structured menu selections in the WhatsApp bot.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** 001 **Description:** Add predefined HR-related query from Axis **Preconditions:** Vendor is logged in, predefined HR-related query is available **Steps:** 1. Vendor selects the option to add a predefined HR-related query 2. Vendor selects the query provided by Axis 3. Save the query **Expected Results:** The query is successfully added for employees to navigate through in the WhatsApp bot **Priority:** High **Validation (Negative/Edge Case) Test Cases:** **Test Case ID:** 002 **Description:** Add predefined HR-related query with missing information **Preconditions:** Vendor is logged in **Steps:** 1. Vendor selects the option to add a predefined HR-related query 2. Leave some required fields blank 3. Save the query **Expected Results:** System should prompt the vendor to fill in all required information **Priority:** Medium **UI/UX Test Cases:** **Test Case ID:** 003 **Description:** Check the appearance of added query in the WhatsApp bot menu **Preconditions:** Vendor is logged in, predefined HR-related query is added **Steps:** 1. Access the WhatsApp bot menu 2. Navigate to the HR-related queries section 3. Verify the appearance and formatting of the added query **Expected Results:** The query is displayed in a structured and user-friendly manner **Priority:** High **Performance Test Cases:** **Test Case ID:** 004 **Description:** Test the response time of navigating through the HR-related queries **Preconditions:** Vendor is logged in, multiple queries are added **Steps:** 1. Access the HR-related queries section in the WhatsApp bot 2. Time how long it takes to navigate through the menu **Expected Results:** Navigating through the queries should be fast and responsive **Priority:** Medium **Security Test Cases:** **Test Case ID:** 005 **Description:** Verify access control for adding queries **Preconditions:** Different vendor roles (admin, regular vendor) are set up **Steps:** 1. Attempt to add a query with a regular vendor account 2. Attempt to add a query with an admin vendor account **Expected Results:** Regular vendors should not have permission to add queries, while admin vendors should have the access **Priority:** High **Combination Test Cases:** **Test Case ID:** 006 **Description:** Add multiple predefined HR-related queries from Axis **Preconditions:** Vendor is logged in, multiple queries are available **Steps:** 1. Vendor selects multiple queries provided by Axis to add 2. Save the queries **Expected Results:** All selected queries are successfully added for employees to navigate through in the WhatsApp bot **Priority:** High

User Story 4

As an vendor (Focaloid), I want the bot to validate employees using a backend database or an API provided by Axis MF, so that only authorized users can access HR services.

Test Cases:

Test Case 1:

Positive Test Cases: **1.** **Validation from Backend Database:** **Test Case ID:** 001 **Description:** Verify that the bot successfully validates employees using the backend database provided by Axis MF. **Preconditions:** Bot is integrated with the backend database. **Steps:** 1. Enter valid employee credentials in the bot. 2. Click on the submit button. **Expected Results:** Bot should authenticate the employee successfully. **2.** **Validation from API:** **Test Case ID:** 002 **Description:** Verify that the bot successfully validates employees using the API provided by Axis MF. **Preconditions:** Bot is integrated with the API. **Steps:** 1. Enter valid employee credentials in the bot. 2. Click on the submit button. **Expected Results:** Bot should authenticate the employee successfully. ### Validation Test Cases: **1.** **Invalid Credentials:** **Test Case ID:** 003 **Description:** Verify that the bot rejects invalid employee credentials. **Preconditions:** Bot is integrated with the backend database or API. **Steps:** 1. Enter invalid employee credentials in the bot. 2. Click on the submit button. **Expected Results:** Bot should display an error message indicating invalid credentials. ### UI/UX Test Cases: **1.** **User Interface Check:** **Test Case ID:** 004 **Description:** Verify that the bot interface displays the employee validation fields correctly. **Preconditions:** Bot interface is loaded. **Steps:** 1. Check if the employee ID and password fields are present. **Expected Results:** Employee validation fields should be displayed correctly. ### Performance Test Cases: **1.** **Load Testing:** **Test Case ID:** 005 **Description:** Verify the performance of the bot when multiple users try to validate simultaneously. **Preconditions:** Multiple users accessing the bot. **Steps:** 1. Simultaneously validate employees from multiple user accounts. **Expected Results:** Bot should handle simultaneous validation requests without performance degradation. ### Security Test Cases: **1.** **Data Encryption Check:** **Test Case ID:** 006 **Description:** Verify that the employee credentials are encrypted during the validation process. **Preconditions:** Bot is integrated with the backend database or API. **Steps:** 1. Validate an employee and check if the credentials are encrypted. **Expected Results:** Employee credentials should be encrypted for security. ### Combination Test Cases: **1.** **Backend Database and API Validation:** **Test Case ID:** 007 **Description:** Verify that the bot can validate employees using both the backend database and the API. **Preconditions:** Bot is integrated with both backend systems. **Steps:** 1. Validate an employee using the backend database. 2. Validate another employee using the API. **Expected Results:** Bot should authenticate employees using both systems successfully. **Priority:** High

User Story 5

As a system administrator, I want to ensure the bot follows compliance regulations, so that employee data is secure.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** Compliance_Bot_001 **Description:** Verify the bot can identify and flag any employee data that violates compliance regulations. **Preconditions:** Compliance regulations are set up in the system. **Steps:** 1. Input employee data that violates compliance regulations. 2. Trigger the bot to scan and detect compliance violations. 3. Check the report generated by the bot. **Expected Results:** The bot successfully identifies and flags the employee data that violates compliance regulations. **Priority:** High **Test Case ID:** Compliance_Bot_002 **Description:** Verify the bot can provide recommendations on how to rectify compliance violations. **Preconditions:** Compliance regulations are set up in the system. **Steps:** 1. Input employee data

that violates compliance regulations. 2. Trigger the bot to scan and detect compliance violations. 3. Review the recommendations provided by the bot on how to rectify the violations. **Expected Results:** The bot offers clear recommendations on how to rectify compliance violations. **Priority:** Medium ### Validation Test Cases: **Test Case ID:** Compliance_Bot_003 **Description:** Verify the bot does not flag employee data that complies with regulations. **Preconditions:** Compliance regulations are set up in the system. **Steps:** 1. Input employee data that complies with compliance regulations. 2. Trigger the bot to scan and detect compliance violations. 3. Check the report generated by the bot. **Expected Results:** The bot does not flag any employee data that complies with regulations. **Priority:** High **Test Case ID:** Compliance_Bot_004 **Description:** Verify the bot handles exceptions and errors during compliance checks. **Preconditions:** Compliance regulations are set up in the system. **Steps:** 1. Input employee data that triggers an error during compliance checking. 2. Trigger the bot to scan and detect compliance violations. 3. Verify how the bot handles the error and if it provides appropriate feedback. **Expected Results:** The bot should handle errors gracefully and provide meaningful feedback. **Priority:** Medium ### UI/UX Test Cases: **Test Case ID:** Compliance_Bot_005 **Description:** Verify the user interface of the bot for setting up and managing compliance regulations. **Preconditions:** Compliance regulations module is accessible to the system administrator. **Steps:** 1. Navigate to the compliance regulations setup page. 2. Check the layout, ease of navigation, and clarity of options. 3. Attempt to add, edit, or remove compliance regulations. **Expected Results:** The user interface is intuitive, user-friendly, and allows easy management of compliance regulations. **Priority:** Medium ### Performance Test Cases: **Test Case ID:** Compliance_Bot_006 **Description:** Test the bot's performance in scanning and flagging compliance violations in a large dataset. **Preconditions:** Compliance regulations are set up in the system with a large dataset of employee information. **Steps:** 1. Input a large dataset of employee information for compliance checking. 2. Trigger the bot to scan and detect compliance violations. 3. Measure the time taken by the bot to complete the scan. **Expected Results:** The bot should scan and flag compliance violations in a large dataset within a reasonable timeframe. **Priority:** High ### Security Test Cases: **Test Case ID:** Compliance_Bot_007 **Description:** Verify the bot's access control and permission levels for compliance data. **Preconditions:** Different user roles with varying access levels are defined in the system. **Steps:** 1. Attempt to access compliance data with a non-administrator user account. 2. Check if the bot restricts access based on user roles. 3. Verify that only authorized personnel can view and manage compliance data. **Expected Results:** The bot enforces strict access control to ensure only authorized personnel can access compliance data. **Priority:** High ### Combination Test Cases: **Test Case ID:** Compliance_Bot_008 **Description:** Verify the bot's performance while scanning a large dataset for compliance violations under different user roles. **Preconditions:** Compliance regulations are set up in the system with a large dataset of employee information. **Steps:** 1. Login with different user roles (administrator, manager, regular user). 2. Input a large dataset of employee information for compliance checking. 3. Trigger the bot to scan and detect compliance violations. 4. Measure the time taken by the bot to complete the scan for each user role. **Expected Results:** The bot should perform efficiently in scanning compliance violations for different user roles in a large dataset. **Priority:** High

User Story 6

As an Vendor (Focaloid), I want the bot to support multiple employees simultaneously, so that all users receive timely responses.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** BOT001 **Description:** Test the bot's ability to support multiple employees simultaneously. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Send simultaneous queries from multiple employees to the bot. 2. Monitor bot's responses to each query. **Expected Results:** Bot should handle and respond to each query independently and in a timely manner. **Priority:** High **Validation Test Cases:** **Test Case ID:** BOT002 **Description:** Test the bot's limit of handling multiple employees. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Send queries from more employees than the bot can handle simultaneously. 2. Monitor bot's responses and behavior. **Expected Results:** Bot should either respond to as many queries as it can handle or provide an error message indicating the overload. **Priority:** Medium **UI/UX Test Cases:** **Test Case ID:** BOT003 **Description:** Test the user interface for interacting with the bot simultaneously. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Simultaneously interact with the bot from multiple user interfaces. 2. Check for any UI/UX issues such as overlapping responses or delays. **Expected Results:** Each user should have a seamless experience interacting with the bot without any UI/UX issues. **Priority:** Medium **Performance Test Cases:** **Test Case ID:** BOT004 **Description:** Test the bot's performance under heavy load. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Send a large number of simultaneous queries to the bot. 2. Measure the response time and system resource usage. **Expected Results:** Bot should respond within an acceptable time frame and should not crash or slow down significantly under heavy load. **Priority:** High **Security Test Cases:** **Test Case ID:** BOT005 **Description:** Test the bot's security measures for handling multiple employees. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Send queries with sensitive information simultaneously from multiple employees. 2. Monitor how the bot handles and protects the sensitive information. **Expected Results:** Bot should have measures in place to securely handle and protect sensitive information from multiple users. **Priority:** High **Combination Test Cases:** **Test Case ID:** BOT006 **Description:** Test a combination of simultaneous queries with heavy load. **Preconditions:** Bot is set up to handle multiple users. **Steps:** 1. Simultaneously send a large number of queries from multiple employees to the bot. 2. Monitor bot's responses and performance. **Expected Results:** Bot should handle all queries independently and respond in a timely manner even under heavy load. **Priority:** High

User Story 7

As a vendor (Focaloid), I want the bot to have a scalable architecture, so that future integrations (like live chat and third-party API connections) are possible.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** 001 **Description:** Verify that the bot architecture supports easy integration of live chat feature **Preconditions:** Bot architecture is in place **Steps:** 1. Attempt to integrate a live chat feature 2. Check for any errors or complications during the integration process **Expected Results:** Live chat feature integrates seamlessly without any issues **Priority:** High **Test Case ID:** 002 **Description:** Verify that the bot architecture supports integration with third-party APIs **Preconditions:** Bot architecture is in place **Steps:** 1. Attempt to connect the bot with a third-party API 2. Check for any errors or complications during the API connection process **Expected Results:** Third-party API connects successfully without any issues **Priority:** High **Validation Test Cases:** **Test Case ID:** 003 **Description:** Attempt to integrate a feature that is not supported by the bot architecture **Preconditions:** Bot architecture is in place **Steps:** 1. Try to integrate a feature that is not compatible with the bot architecture 2. Check for error messages or

issues encountered during the integration process **Expected Results:** Error message indicating that the feature is not supported by the current architecture **Priority:** Medium **UI/UX Test Cases:** **Test Case ID:** 004 **Description:** Verify that the integration process for new features is user-friendly **Preconditions:** Bot architecture is in place **Steps:** 1. Attempt to integrate a new feature using the provided tools and interfaces 2. Evaluate the ease of use and clarity of instructions during the integration process **Expected Results:** Integration process is intuitive and easy to follow **Priority:** High **Performance Test Cases:** **Test Case ID:** 005 **Description:** Test the response time of the bot after integrating new features **Preconditions:** Bot architecture is in place **Steps:** 1. Integrate a new feature and interact with the bot to test response time 2. Measure the time taken for the bot to respond to user queries **Expected Results:** Bot responds promptly without any delays **Priority:** High **Security Test Cases:** **Test Case ID:** 006 **Description:** Ensure that integrating new features does not compromise the security of the bot **Preconditions:** Bot architecture is in place **Steps:** 1. Integrate a new feature and conduct security testing to identify vulnerabilities 2. Check for any security risks or breaches after the integration **Expected Results:** Bot remains secure and there are no vulnerabilities introduced by the new feature integration **Priority:** High **Combination Test Cases:** **Test Case ID:** 007 **Description:** Integrate live chat feature and connect with a third-party API simultaneously **Preconditions:** Bot architecture is in place **Steps:** 1. Integrate live chat feature 2. Connect with a third-party API 3. Check for any conflicts or issues that arise from integrating both features at the same time **Expected Results:** Both features integrate successfully without any conflicts **Priority:** High

User Story 8

As a vendor (Focaloid), I want to set up the server for hosting the application, so that the application is securely deployed, scalable, and accessible to end-users.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** 001 **Description:** Verify if the server setup process is successful. **Preconditions:** Vendor has access to the server setup tools. **Steps:** 1. Vendor initiates the server setup process. 2. Configure server settings, security measures, and scalability options. 3. Deploy the application on the server. **Expected Result:** Server setup is completed without errors, application is deployed successfully, and server is accessible. **Priority:** High **Validation Test Cases:** **Test Case ID:** 002 **Description:** Verify server setup failure due to incorrect configuration. **Preconditions:** Incorrect server settings or security measures. **Steps:** 1. Vendor initiates the server setup process. 2. Provide incorrect configuration details. 3. Deploy the application on the server. **Expected Result:** Server setup fails with error messages indicating incorrect configuration. **Priority:** High **UI/UX Test Cases:** **Test Case ID:** 003 **Description:** Verify the user-friendly interface of the server setup tool. **Preconditions:** Server setup tool is accessible to the vendor. **Steps:** 1. Vendor accesses the server setup tool. 2. Navigate through the setup process. 3. Check for intuitive design and clear instructions. **Expected Result:** Server setup tool has a user-friendly interface with clear instructions and easy navigation. **Priority:** Medium **Performance Test Cases:** **Test Case ID:** 004 **Description:** Verify server performance under load. **Preconditions:** Application deployed on the server. **Steps:** 1. Simulate high traffic to the application. 2. Monitor server response time and resource utilization. **Expected Result:** Server can handle high traffic load efficiently without significant performance degradation. **Priority:** High **Security Test Cases:** **Test Case ID:** 005 **Description:** Verify server security measures. **Preconditions:** Security measures configured during setup. **Steps:** 1. Perform security

vulnerability testing on the server. 2. Check for encryption protocols, access controls, and secure connections. **Expected Result:** Server passes security testing with no vulnerabilities detected. **Priority:** High **Combination Test Cases:** **Test Case ID:** 006 **Description:** Verify server setup with both high traffic load and security testing. **Preconditions:** Server setup with security measures and application deployed. **Steps:** 1. Simulate high traffic to the application. 2. Perform security vulnerability testing simultaneously. **Expected Result:** Server maintains performance under load while passing security testing. **Priority:** High

User Story 9

As a vendor (Focaloid), I want to set up a new database that is identical to the one used for investor or distributor services, so that we can reuse it for the HR bot.

Test Cases:

Test Case 1:

1. **Positive Test Cases:** **1.1** **Test Case ID:** VENDOR_DB_001 **Description:** Verify that the vendor (Focaloid) can successfully set up a new database identical to the one used for investor or distributor services. **Preconditions:** Access to the database setup tool, permissions to create a new database. **Steps:** 1. Login to the database setup tool. 2. Select the option to create a new database. 3. Choose to replicate the structure of the database used for investor or distributor services. 4. Complete the setup process. **Expected Result:** New database created successfully with the same structure as the one used for investor or distributor services. 2. **Validation Test Cases:** **2.1** **Test Case ID:** VENDOR_DB_002 **Description:** Attempt to set up a new database without the necessary permissions. **Preconditions:** Lack of permissions to create a new database. **Steps:** 1. Try to access the database setup tool. 2. Select the option to create a new database. 3. Proceed with the setup process. **Expected Result:** Permission error message displayed, unable to create the new database. 3. **UI/UX Test Cases:** **3.1** **Test Case ID:** VENDOR_DB_003 **Description:** Check the user interface for setting up a new database. **Preconditions:** Access to the database setup tool. **Steps:** 1. Log in to the database setup tool. 2. Navigate to the option for creating a new database. 3. Observe the layout, ease of navigation, and clarity of instructions. **Expected Result:** Intuitive user interface with clear instructions for creating a new database. 4. **Performance Test Cases:** **4.1** **Test Case ID:** VENDOR_DB_004 **Description:** Test the time taken to set up a new database. **Preconditions:** Access to the database setup tool. **Steps:** 1. Start the process of creating a new database. 2. Measure the time taken to complete the setup. **Expected Result:** Database setup completed within a reasonable timeframe without significant delays. 5. **Security Test Cases:** **5.1** **Test Case ID:** VENDOR_DB_005 **Description:** Verify that only authorized users can access the database setup tool. **Preconditions:** Access to the database setup tool. **Steps:** 1. Attempt to access the database setup tool with unauthorized credentials. **Expected Result:** Unauthorized access denied, ensuring data security. 6. **Combination Test Cases:** **6.1** **Test Case ID:** VENDOR_DB_006 **Description:** Combine positive and security test cases by setting up a new database with authorized credentials. **Preconditions:** Access to the database setup tool, permissions to create a new database. **Steps:** 1. Login to the database setup tool with authorized credentials. 2. Select the option to create a new database. 3. Choose to replicate the structure of the database used for investor or distributor services. 4. Complete the setup process. **Expected Result:** New database created successfully with proper security measures in place.

User Story 10

As a vendor (Focaloid), I want to authenticate Axis admin users to the backend admin panel, so that can manage and edit menus and messages on the admin panel.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** FT001 **Description:** Verify successful authentication of Axis admin users to the backend admin panel. **Preconditions:** User is an Axis admin user. **Steps:** 1. Enter valid credentials for an Axis admin user. 2. Click on the login button. **Expected Results:** User successfully logs in to the backend admin panel. **Priority:** High **Validation Test Cases:** **Test Case ID:** VT001 **Description:** Verify authentication failure for incorrect credentials. **Preconditions:** User is an Axis admin user. **Steps:** 1. Enter incorrect credentials for an Axis admin user. 2. Click on the login button. **Expected Results:** User should not be able to log in and receive an authentication error message. **Priority:** High **UI/UX Test Cases:** **Test Case ID:** UT001 **Description:** Verify the login page layout and design. **Preconditions:** User is on the login page. **Steps:** 1. Check the alignment of input fields and buttons. 2. Check if error messages are displayed correctly. **Expected Results:** The login page should have a clean and user-friendly layout. **Priority:** Medium **Performance Test Cases:** **Test Case ID:** PT001 **Description:** Verify the response time for user authentication. **Preconditions:** User is an Axis admin user. **Steps:** 1. Enter valid credentials for an Axis admin user. 2. Click on the login button. **Expected Results:** The authentication process should be quick and responsive. **Priority:** Medium **Security Test Cases:** **Test Case ID:** ST001 **Description:** Verify the handling of multiple failed login attempts. **Preconditions:** User is an Axis admin user. **Steps:** 1. Enter incorrect credentials multiple times. **Expected Results:** After a certain number of failed attempts, the account should be locked to prevent brute force attacks. **Priority:** High **Combination Test Cases:** **Test Case ID:** CT001 **Description:** Verify authentication with various combinations of valid and invalid credentials. **Preconditions:** User is an Axis admin user. **Steps:** 1. Enter valid username and invalid password. 2. Enter invalid username and valid password. 3. Enter both username and password incorrectly. **Expected Results:** System should handle each combination appropriately and provide relevant error messages. **Priority:** High

User Story 11

As a vendor (Focaloid), I want to clone the existing investor services codebase, so that I can make the necessary minor modifications for the HR Bot project.

Test Cases:

Test Case 1:

Positive Test Cases: **Test Case ID:** 001 **Description:** Cloning the existing investor services codebase for HR Bot project. **Preconditions:** Vendor (Focaloid) has access to the existing investor services codebase. **Steps:** 1. Log in to the version control system. 2. Clone the existing investor services codebase. 3. Make necessary minor modifications for the HR Bot project. **Expected Results:** The existing investor services codebase is successfully cloned and modifications for the HR Bot project are made. **Validation (Negative/Edge Case) Test Cases:** **Test Case ID:** 002

Description: Cloning the existing investor services codebase without access. **Preconditions:** Vendor (Focaloid) does not have access to the existing investor services codebase. **Steps:** 1. Try to log in to the version control system. 2. Attempt to clone the existing investor services codebase. **Expected Results:** Unable to clone the existing investor services codebase due to lack of access. **UI/UX Test Cases:** **Test Case ID:** 003 **Description:** Ensure the user interface is intuitive for cloning the codebase. **Preconditions:** Vendor (Focaloid) is logged in to the version control system. **Steps:** 1. Locate the option to clone the codebase. 2. Verify that the process is straightforward and user-friendly. **Expected Results:** The user interface for cloning the codebase is intuitive and easy to use. **Performance Test Cases:** **Test Case ID:** 004 **Description:** Test the time taken to clone the codebase. **Preconditions:** Vendor (Focaloid) has access to the existing investor services codebase. **Steps:** 1. Start the process of cloning the codebase. 2. Measure the time taken to complete the cloning process. **Expected Results:** The codebase is cloned within a reasonable time frame without significant delays. **Security Test Cases:** **Test Case ID:** 005 **Description:** Verify that only authorized users can clone the codebase. **Preconditions:** Vendor (Focaloid) has access to the version control system. **Steps:** 1. Attempt to clone the codebase with unauthorized credentials. **Expected Results:** Unauthorized users are unable to clone the codebase. **Combination Test Cases:** **Test Case ID:** 006 **Description:** Cloning the codebase with simultaneous modifications. **Preconditions:** Vendor (Focaloid) has access to the existing investor services codebase. **Steps:** 1. Clone the codebase. 2. Make modifications to the code while cloning is in progress. **Expected Results:** The codebase is successfully cloned, and modifications are applied simultaneously without conflicts.