**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document.**

* **Use Case Diagram**
* A use case diagram visually represents interactions between users (actors) and systems.
* It helps in understanding how the users will interact with the system to achieve specific goals which are represented as use cases.
* Use case diagrams are key components of UML (Unified Modelling Language) and are used in the early stage of software design to capture system functionality.

 

* **Activity Diagram:**
* Activity diagrams visually represent the workflow of specific activities that users perform on the platform
* It represents the workflow of stepwise activities and actions within the system
* Activity diagrams provide visual clarity of the flow of control and iterations
* It also helps in identifying the bottleneck in the processes

****

* **Use Case Specification Document:**
* Use cases are tools for capturing the functional requirements of a system.
* For this project, use case specifications will describe how farmers will interact with the system to perform tasks like searching the product, placing orders and making payments.
* By defining use cases, user roles and actions can be documented ensuring the user’s experience is aligned with real-world needs and expectations.

|  |  |
| --- | --- |
| **Use Case ID** | UC001 |
| **Use Case Name** | User Registration for Docubuilder App |
| **Use Case Description** | This use case describes the process of a new user registering for an account on the Docubuilder app to access features related to mutual fund document creation. |
| **Actors (Primary Actors and Secondary actors)** | **Primary Actors: New UserSecondary Actors:Docubuilder App SystemDatabase System** |
| **Basic Flow** | **Use Case Begins -**1. User opens the Docubuilder app
2. The system **displays the registration form** with fields for the user to enter required information, such as: Name, email,phone number,username,password,role
3. The **New User enters all required information** into the registration form.
4. The **Docubuilder App System** validates the entered data
5. If validation is successful, the system:Creates a new user profile in the **Database System.**
6. The user receives the **confirmation email** and clicks on the link to **verify their email address**.
7. Upon successful email verification, the system **activates the user account**.
8. The user is then redirected to the **login page** to sign in with their newly registered credentials.
9. **User logs into the Docubuilder app** using the registered username and password to start creating mutual fund documents.

**Use Case Ends** |
| **ALTERNATE FLOW** | **1: Invalid Email or UsernameI**f the user enters an email address or username that is already registered, the system displays an error message:"The email or username is already in use. Please choose a different one."**2: Weak Password**If the user enters a password that does not meet the minimum security requirements (e.g., length or special characters), the system displays an error message:"Password does not meet strength requirements. Please use at least 8 characters, including a number and a special character."**3: Email Already Verified**If the user clicks on the verification link in the email but their account has already been verified, the system displays a message:"Your email address has already been verified." |
| **Exceptional flows** | **1: Network Error During Registration**If there is a network failure or timeout while submitting the registration form, the system displays an error message:"Network error. Please check your connection and try again."**2: System Error During Profile Creation**If the system encounters an error while saving user details to the database (e.g., database connection failure), the user is shown an error message:"An error occurred while creating your account. Please try again later." |
| **Pre- Conditions** | * The user has internet access and a device to access the Docubuilder app.
* The system is up and running, and the user registration page is accessible.
* The user has not yet created an account with the Docubuilder app.
 |
| **post-conditions** | * The user has a valid account on the Docubuilder app.
* A verification email has been sent to the user’s email address.
* The user’s profile information is stored securely in the app’s database.
* The user can log in to the Docubuilder app once they verify their email.
 |
| **Assumptions** | * The user understands how to complete the registration form
* The user will receive the email verification link and be able to click it to verify their email address
 |
| **Constraints** | * The system must handle multiple user registrations concurrently and ensure the user data is stored securely.
* There may be limitations on the frequency of registration attempts to prevent spam
 |
| **Dependencies** | * The registration process depends on the availability and performance of the **Authentication Service** for creating and verifying user accounts.
* The **Email Service** is required for sending the verification email and notifications.
* The **Database System** is required to store the user’s profile and registration information.
 |
| **Inputs and Outputs** | **Inputs:*** Full Name
* Email Address
* Phone Number (optional)
* Username
* Password
* Confirm Password
* Role (optional)

**Outputs:*** Success: Confirmation email sent to the user’s email address.
* Failure: Error messages for invalid inputs (e.g., username/email already taken, weak password).
* Success after email verification: User profile is activated, and the user can log in.
 |
| **Business Rules** | * **Email Uniqueness Rule**: Each email address can only be associated with one account. If the email is already registered, the system must reject the registration attempt.
* **Password Strength Rule**: Passwords must meet a minimum strength (e.g., length of 8 characters, a combination of uppercase, lowercase, numbers, and special characters).
* **Email Verification Rule**: Users must verify their email address before they can log in and use the application.
 |
| **Miscellaneous Information** | * The registration page should be user-friendly and include appropriate validation messages for each field.
* The system might include CAPTCHA or other anti-bot mechanisms to prevent automated sign-ups.
* The email verification link should expire after a certain time period (e.g., 24 hours), requiring the user to request a new verification email if not verified in time.
 |

|  |  |
| --- | --- |
| **Use Case ID** | UC002 |
| **Use Case Name** | Log into Docubuilder App |
| **Use Case Description** | This use case describes the process of a user logging into the Docubuilder app to access the features required to create mutual fund documents. |
| **Actors (Primary Actors and Secondary actors)** | **Primary Actors: New UserSecondary Actors:Docubuilder App SystemDatabase System** |
| **Basic Flow** | **Use Case Begins -*** User opens the Docubuilder app
* The app **displays the login screen**, prompting the user to enter their username and password.
* The **User enters their username and password.**
* If the credentials are valid, the **user is logged in successfully** and can now proceed with document creation.
* If the credentials are invalid, The system displays an error message, "Invalid username or password. Please try again."
* The user can either retry the login process or request a **password reset** if they have forgotten their credentials.

**Use Case Ends** |
| **ALTERNATE FLOW** | **1: Forgotten Password**If the user selects the **"Forgot Password"** option, the system sends a **password reset link** to the user’s email.The user can follow the link to reset the password and log in.**2: Account Locked Due to Multiple Failed Attempts**If the user exceeds the allowed number of **failed login attempts**, the system temporarily locks the account. |
| **Exceptional flows** | **1: Network Error**If the user’s internet connection is lost during the login process, the system shows an error message:"Network error. Please check your internet connection and try again."**2: System Error** If the authentication service is down or unavailable, the system displays a message:"Unable to log in at the moment. Please try again later." |
| **Pre- Conditions** | * The user must have an existing account with valid credentials (username and password).
* The user has a working internet connection to access the app and authenticate their credentials.
 |
| **post-conditions** | * The user is either successfully logged in and granted access to the app or is shown an error message and is unable to log in.
* If the login is successful, the user has access to the mutual fund document creation functionality.
 |
| **Assumptions** | * The user knows their username and password.
* The app supports multiple login attempts before the account is locked for security reasons.
 |
| **Constraints** | * The system may impose a limit on the number of login attempts to prevent brute-force attacks.
 |
| **Dependencies** | * The user’s credentials must be stored securely in a user authentication database.
* The login process depends on a stable internet connection for both the user and the app to communicate with the authentication service.
 |
| **Inputs and Outputs** | **Inputs:*** Username
* Password

**Outputs:*** Success: Access to the app and dashboard or mutual fund document creation page.
* Failure: Error message if login credentials are invalid, account is locked, or system experiences issues.
 |
| **Business Rules** | * **Security Rule**: All user passwords must be stored securely, using proper encryption techniques.
* **Authentication Rule**: Users must input their credentials correctly to access the system. If they fail to do so multiple times, their account will be locked temporarily.
* **Password Recovery Rule**: If a user forgets their password, they must be able to request a password reset and securely recover their account.
 |
| **Miscellaneous Information** | * The system may offer two-factor authentication (2FA) for additional security, depending on the user’s role and the organization's security policy.
* The system might display a **"Remember Me"** option for users who wish to stay logged in between sessions, reducing the need to re-enter credentials each time.
 |

|  |  |
| --- | --- |
| **Use Case ID** | UC003 |
| **Use Case Name** | Mutual Fund Document Creation |
| **Use Case Description** | This use case describes the process of creating a mutual fund document using the Docubuilder app.The application enables users to input necessary details and generate compliant mutual fund documents. |
| **Actors (Primary Actors and Secondary actors)** | **Primary Actors: UserSecondary Actors: Docubuilder App SystemDatabase System** |
| **Basic Flow** | **Use Case Begins –*** **User logs in** to the Docubuilder app with valid credentials.
* The user selects the **“Create Mutual Fund Document”** option from the dashboard.
* The app prompts the user to **input or select the mutual fund** for which the document is being created.
* The user inputs required data such as fund name, strategy, performance data, risk assessment, and regulatory disclosures.
* The app **generates a preview** of the mutual fund document.
* The user **reviews the preview** and makes any necessary edits.
* Once satisfied with the preview, the user **confirms** and the app generates the final mutual fund document.
* The user is prompted to **download** the document in various formats (e.g., PDF, Word, Excel).
* The document is **saved** to the user’s account, and a confirmation message is displayed.

**Use Case Ends** |
| **ALTERNATE FLOW** | **1: Missing Information**If the user does not input required fields (e.g., performance data, risk assessment), the app prompts the user to **complete the missing fields** before continuing.**2: Data Validation Failure**If the input data does not meet regulatory standards, the system displays an error message, requesting the user to **correct the input data.** |
| **Exceptional flows** | **1: Failed Document Generation**If the document generation fails due to a system error or data issue, the app displays an error message:"Document generation failed. Please try again later."**2: Login Failure**If the user’s login credentials are invalid, the system displays an error message:"Invalid username or password. Please try again." |
| **Pre- Conditions** | 1. The user must have a valid account with the Docubuilder app.
2. The user must be authorized to create mutual fund documents.
3. The mutual fund data must be up to date in the database
 |
| **post-conditions** | 1. A valid mutual fund document is generated and saved.
2. The user is able to download or share the document as needed.
3. The generated document is archived in the user's profile for future reference.
 |
| **Assumptions** | 1. The user is familiar with mutual fund terms and regulatory requirements.
2. The Docubuilder app has access to real-time financial data and regulatory guidelines.
3. The app’s data validation rules and document templates are up to date and aligned with current regulations.
 |
| **Constraints** | 1. The system may be subject to performance issues with large datasets or high user load.
2. There may be restrictions on certain types of document formats
3. Internet connectivity may affect the user experience, especially during document download.
 |
| **Dependencies** | 1. The system depends on real-time data from external financial services for mutual fund performance and regulatory updates.
2. The app depends on accurate user input for generating correct documents.
3. The document creation depends on the availability and integrity of database systems and document templates.
 |
| **Inputs and Outputs** | **Inputs:*** Fund name
* Fund strategy/description
* Performance data (historical returns)
* Risk data (e.g., volatility, risk rating)
* Compliance and regulatory disclosures
* Contact and marketing details (if applicable)

**Outputs:*** Generated mutual fund document (PDF, Word, Excel formats)
* Document preview for review before finalization
* Confirmation messages and error notifications
 |
| **Business Rules** | * **Compliance Rule**: All mutual fund documents must meet regulatory requirements, including performance data disclosures, risk assessment, and legal disclaimers.
* **Data Integrity Rule**: The mutual fund’s financial data must be accurate and sourced from a trusted financial data provider.
* **Formatting Rule**: The generated document must adhere to predefined templates set by the Docubuilder app, ensuring consistency in document structure.
 |
| **Miscellaneous Information** | * The mutual fund document templates are customizable based on the user’s profile or the specific fund being documented.
* The app includes built-in version control, allowing users to track changes and updates to the document over time.
* The app supports multilingual document generation for international funds.
 |

|  |  |
| --- | --- |
| **Use Case ID** | UC004 |
| **Use Case Name** | Choose Template for Mutual Fund Document Creation |
| **Use Case Description** | This use case describes the process of selecting a document template in the **Docubuilder app** for creating a mutual fund document. |
| **Actors (Primary Actors and Secondary actors)** | **Primary Actor: User** **Secondary Actors:Docubuilder App SystemTemplate LibraryDatabase System** |
| **Basic Flow** | **Use Case Begins –**1. **User logs into the Docubuilder app.**
2. The user navigates to the **“Create Document”** section from the main dashboard.
3. The system **displays available templates** for mutual fund documents.
4. The user **browses through the template categories** and reviews available options.
5. The user **selects a template** that suits their needs.
6. The system loads the selected template and prompts the user to **input or select mutual fund data** to populate the document.
7. The system may allow the user to **customize the selected template**, such as modifying sections, adding custom text, or uploading images/logos.
8. The user begins the process of **creating and editing the mutual fund document** using the selected template.

**Use Case Ends** |
| **ALTERNATE FLOW** | **1: No Template Found for the Desired Document Type**If the user is unable to find a suitable template, the system may provide an option to **Request new templates** or **contact support** to create a custom template.**2: Template Preview Not Available**If the template preview cannot be loaded due to a technical issue (e.g., broken link or system error), the system displays a message:“Unable to load the template preview. Proceed with the selection?” |
| **Exceptional flows** | **1: System Error** If the system fails to retrieve the available templates due to a network or database issue, the system displays a message:“Unable to load templates. Please try again later.”**Exception Flow 2: Data Error** If the user selects a template but the mutual fund data does not match the required format or fields, the system prompts the user with:“Some fields are missing or incorrectly formatted. Please review the data before proceeding.” |
| **Pre- Conditions** | * The user must be logged in to the Docubuilder app.
* The user must have access to the mutual fund data to populate the selected template.
 |
| **post-conditions** | * The user has selected a valid template, and the template is now loaded for document creation.
* The system allows the user to input or edit mutual fund data to generate the document.
* The user can proceed with customization of the document as needed.
 |
| **Assumptions** | * The user is familiar with the types of documents and knows which template they need to choose.
* The templates are up to date and comply with the latest regulatory and branding guidelines for mutual fund documents.
* The Docubuilder app is integrated with the relevant mutual fund data sources to automatically populate the document once the template is selected.
 |
| **Constraints** | * The templates may have limitations on customization
* The user may require specific permissions to access premium or advanced templates.
 |
| **Dependencies** | * The template selection process depends on the availability and accuracy of the **Template Library** and **mutual fund data**.
* The **Database System** must be accessible to load templates and provide data for document generation.
 |
| **Inputs and Outputs** | **Inputs:*** Selection of the document type
* Choice of a specific template within the selected category.

**Outputs:*** A selected template is loaded and presented for further customization.
* A document preview, once the user starts filling in data, may be shown for review.
* A confirmation that the selected template is now ready for mutual fund data population.
 |
| **Business Rules** | * **Template Availability Rule**: The system must only display templates that are appropriate for the user’s role (e.g., Financial Advisor, Asset Manager).
* **Customization Rule**: While templates may be customizable, certain fields or sections (e.g., regulatory disclaimers, compliance information) should remain fixed to comply with legal and branding standards.
* **Preview Rule**: If a preview of the selected template is available, it must accurately reflect the structure and design of the document once populated with data.
 |
| **Miscellaneous Information** | * Users can save templates as drafts and continue editing later, without losing progress.
* If the user selects a template from a **third-party provider**, the app must integrate with external systems to ensure correct formatting and compatibility.
 |

|  |  |
| --- | --- |
| **Use Case ID** | UC005 |
| **Use Case Name** | Fill Information in Chosen Template for Mutual Fund Document Creation |
| **Use Case Description** | This use case describes the process in which a user populates the selected template with specific mutual fund data to create a customized document. |
| **Actors (Primary Actors and Secondary actors)** | **Primary Actor: UserSecondary Actors:Docubuilder App SystemDatabase System** |
| **Basic Flow** | **Use Case Begins –**1. **User logs into the Docubuilder app** and navigates to the **“Create Document”** section.
2. The user **selects a template** for the mutual fund document (e.g., Fund Fact Sheet, Performance Report).
3. The system **loads the selected template** and presents it with placeholders for the relevant data (e.g., fund name, NAV, historical performance).
4. The user **fills in the fields** by: **Manually entering data or Selecting data from pre-integrated sources** populated by the system from connected databases or external APIs.
5. The system **displays real-time suggestions** or provides auto-fill options for certain fields
6. The user can **edit, add, or remove sections** in the template, depending on the flexibility of the selected template.
7. The user **reviews the document**, making sure all necessary sections are filled and properly formatted.
8. The system **validates the input data** for consistency and correctness (e.g., correct format for NAV, performance, and risk data).

**Use Case Ends** |
| **ALTERNATE FLOW** | **1: Missing Required Data**If the user leaves a mandatory field empty or inputs incorrect data the system displays an error message:“Missing required data: NAV is required for the selected template. Please fill in the data to proceed.”**2: External Data Source Unavailable**If the system fails to fetch data from external sources, the system displays a warning:“Unable to fetch real-time data for the selected fields. Would you like to continue manually entering the data?”**3: User Wants to Add Custom Sections**If the user wants to add a custom section the system allows them to select **"Add Section"** and provides a prompt for custom content. |
| **Exceptional flows** | **1: Data Format Error**If the user inputs incorrect data format,the system highlights the problematic field and displays a message:“Invalid format for field [Field Name]. Please enter a valid number/date.”**2: System Crash or Timeout During Data Entry**If the app crashes or the session times out while the user is filling in data, the system will prompt with:“Your session has expired or an error occurred. Would you like to continue from where you left off?” |
| **Pre- Conditions** | * The user must have logged into the Docubuilder app.
* The user must have selected a template for mutual fund document creation.
* The user must have access to the required mutual fund data either through manual input or data integration.
 |
| **post-conditions** | * The user has filled in all necessary fields within the selected template.
* The document is either saved as a draft or exported in the desired format (e.g., PDF, Word).
* The data entered by the user is validated and confirmed to be correct (if applicable).
* The system may store the completed document in the user’s account or allow sharing via email or other communication channels.
 |
| **Assumptions** | * The user has access to the necessary data (either manual or automated) to complete the document.
* The templates provided by Docubuilder are designed to handle typical mutual fund document sections and are flexible enough to allow for customization where needed.
* The user understands how to navigate the app to enter or select the necessary data fields for mutual fund documents.
 |
| **Constraints** | * The available templates may have predefined sections that cannot be modified, limiting the level of customization.
* The data fetched from external sources (e.g., APIs) may be subject to availability or delay.
* Data may need to be updated periodically to ensure that the most recent figures (e.g., NAV, performance) are available.
 |
| **Dependencies** | * The template system must be connected to a **data integration service** to pull mutual fund data (e.g., NAV, returns).
* The system depends on a **Database System** to store user data, document content, and templates.
* **External APIs** may be required to fetch real-time data for mutual fund metrics (performance, NAV, sector allocation, etc.).
 |
| **Inputs and Outputs** | **Inputs:*** User data entry (e.g., fund overview, market commentary, performance metrics).
* Data fetched from external sources (e.g., NAV, historical performance).
* User-selected sections to add or edit in the document.
* Customized content (e.g., images, custom commentary).

**Outputs:*** A completed mutual fund document, either saved as a draft or exported.
* Error or validation messages for missing or incorrect data.
* A document preview (in case of preview mode) showing how the populated template will look.
 |
| **Business Rules** | * **Data Validation Rule**: All data entered must conform to the expected format (e.g., numbers, dates) and be validated before finalizing the document.
* **Customization Rule**: Users may customize certain sections, but any legal or compliance-related fields (e.g., disclaimers) cannot be altered.
 |
| **Miscellaneous Information** | * The system may include a **"Save Draft"** feature to allow users to save their progress without completing the document.
* Users may have the ability to **share documents** via email or other means directly from the app.
 |

 **Document 7- Screens and pages**

* 1. **Login**

****

* 1. **User Registration**

 ****

* 1. **Create Mutual Fund Document**

 ****

**Document 8- Tools-Visio and Axure**

**Write a paragraph on your experience using Visio and Axure for the project.**

* In my experience using Visio and Axure for project design and documentation, each tool has its own strengths to different aspects of the process.
* In simple terms, Visio helped me draw clear, easy-to-understand diagrams that showed how different people—like investors or financial advisors—would interact with the DocuBuilder app when creating or managing mutual fund documents.
* It was great for laying out the overall picture of how everything works.
* Then, I used balsamic/Axure to create a clickable prototype of the app. This allowed me to show how the app would actually work in real life, like how users would click through different screens to create and review documents.
* So, while Visio helped make the process easy to understand with simple diagrams, balsamic helped bring that process to life by showing how users would actually interact with the app.

**Document 9- BA experience**

**My experience as BA in following phases:**

1. **Requirement Gathering:**
* I employed the MOSCOW technique to gather requirements effectively.
* Due to the client's unavailability for certain periods, I had to identify alternate points of contact from their side to ensure timely information flow.
* I used the FURPS technique to validate the requirements, ensuring they were complete and feasible.
* I identified and removed duplicated or repeated requirements immediately to avoid confusion.
* Prototyping was utilised to clarify and gather more specific requirements.
1. **Requirement Analysis:**
* I created UML diagrams to visually represent the requirements, ensuring clarity and understanding across the team.
* Activity diagrams were also used to depict the process flow in a more detailed manner.
* After sharing these diagrams with the team, I conducted discussions and incorporated feedback, making necessary changes where required.
* I prepared the Business Requirement Specification (BRS) and Software Requirement Specification (SRS) documents to formally capture and communicate requirements.
1. **Design:**
* From the use case diagrams, I developed test cases to ensure proper coverage of requirements.
* I collaborated closely with the client to review design and solution documents.
* In addition to positive test cases, I also wrote negative test cases to cover edge cases and potential issues.
* I made sure not to overlook any test case, as it could have significant impacts during later stages of development.
* I prepared test data required for the testing phase.
* The Requirements Traceability Matrix (RTM) was regularly updated to ensure that all requirements were addressed.
1. **Development:**
* I organized Joint Application Development (JAD) sessions to gather input and clarify requirements with the team.
* I served as the point of contact for any queries raised by the tech team during the coding process.
* When disagreements or lack of cooperation occurred within the team during JAD sessions, I handled the situation diplomatically, often having one-on-one discussions to explain the impact of their actions on the project and fostering a cooperative team environment.
* I referred to the diagrams to support the development of the Unit.
* I conducted regular meetings with both the technical team and the client, understanding the challenge of availability.
* If someone missed a meeting, I recorded the session and arranged one-on-one discussions to keep things aligned.
1. **Testing:**
* I prepared detailed test cases derived from the use cases to ensure comprehensive testing.
* I performed high-level testing to validate that the system met business requirements.
* I worked with the client to request and gather the necessary test data.
* I kept the RTM up-to-date to track testing progress and coverage.
* After thorough testing, I took formal signoff from the client.
* I also prepared the client for User Acceptance Testing (UAT), ensuring they were ready to conduct final validation.
1. **Deployment:**
* I forwarded the final RTM to the client as part of the project closure documentation.
* I coordinated the completion and sharing of end-user manuals to ensure smooth adoption.
* I organized and planned training sessions for end users, ensuring all relevant candidates attended and were well-prepared to use the system post-deployment.