MUTUAL FUND APP

**Business Requirement Document (BRD) for Mutual Fund App**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to outline the business requirements for the development of a Mutual Fund App. This app will enable users to invest in mutual funds, track their investments, and access financial advice.

**1.2 Scope**

This document covers the functional and non-functional requirements of the Mutual Fund App, including user interfaces, integration points, and performance criteria.

**1.3 Definitions, Acronyms, and Abbreviations**

* **NAV**: Net Asset Value
* **SIP**: Systematic Investment Plan
* **AMC**: Asset Management Company

**2. Business Objectives**

* Provide a user-friendly platform for investing in mutual funds.
* Allow users to manage and track their investments.
* Offer educational resources and financial advice.
* Ensure secure and compliant transactions.

**3. Stakeholders**

* **Users**: Individual investors, Financial Advisors
* **Business Owners**: Mutual Fund Companies, Financial Institutions
* **Development Team**: Project Managers, Developers, QA Analysts
* **Regulatory Bodies**: SEBI (Securities and Exchange Board of India), AMFI (Association of Mutual Funds in India)

**4. Functional Requirements**

**4.1 User Registration and Login**

* **FR1.1**: Users should be able to register using email, phone number, or social media accounts.
* **FR1.2**: Users should be able to log in using their registered credentials.

**4.2 User Dashboard**

* **FR2.1**: Display an overview of the user’s portfolio including total investment, current value, and returns.
* **FR2.2**: Show recent transactions and notifications.

**4.3 Mutual Fund Search and Filter**

* **FR3.1**: Users should be able to search for mutual funds by name, AMC, or fund manager.
* **FR3.2**: Users should be able to filter mutual funds based on categories like equity, debt, hybrid, etc.

**4.4 Investment Options**

* **FR4.1**: Users should be able to invest in mutual funds via lump sum or SIP.
* **FR4.2**: Provide a comparison tool to compare different mutual funds.

**4.5 Transaction Management**

* **FR5.1**: Support secure payment options for transactions.
* **FR5.2**: Users should be able to view transaction history.

**4.6 Notifications and Alerts**

* **FR6.1**: Send notifications for NAV updates, dividends, and important announcements.
* **FR6.2**: Allow users to set custom alerts for fund performance thresholds.

**4.7 Reports and Statements**

* **FR7.1**: Provide detailed investment statements and reports.
* **FR7.2**: Allow users to download or email reports.

**4.8 Customer Support**

* **FR8.1**: Integrate a chatbot for basic queries.
* **FR8.2**: Provide contact options for customer support.

**5. Non-Functional Requirements**

**5.1 Security**

* **NFR1.1**: Ensure data encryption during storage and transmission.
* **NFR1.2**: Implement multi-factor authentication (MFA) for user accounts.

**5.2 Performance**

* **NFR2.1**: The app should load the user dashboard within 3 seconds.
* **NFR2.2**: Transactions should be processed within 5 seconds.

**5.3 Usability**

* **NFR3.1**: The app should have an intuitive user interface.
* **NFR3.2**: Provide help and support documentation within the app.

**5.4 Compliance**

* **NFR4.1**: Ensure compliance with SEBI and AMFI regulations.
* **NFR4.2**: Adhere to GDPR for data privacy and protection.

**6. Assumptions and Constraints**

**6.1 Assumptions**

* Users have access to the internet.
* Users possess basic knowledge of mutual fund investments.

**6.2 Constraints**

* Integration with external payment gateways.
* Dependency on AMC data feeds for real-time updates.

**7. Acceptance Criteria**

* The app should pass all functional and non-functional tests.
* Users should be able to complete transactions without errors.
* Compliance with regulatory requirements must be verified.

**8. Appendices**

**8.1 Glossary**

* **SIP**: Systematic Investment Plan
* **AMC**: Asset Management Company

**8.2 References**

* SEBI Guidelines for Mutual Funds
* AMFI Best Practices

This BRD provides a comprehensive overview of the requirements for developing a Mutual Fund App. It is essential to gather detailed inputs from stakeholders and refine these requirements as needed during the development process.

4o

Give me the Use case specification document for this above given details

ChatGPT

Here is a Use Case Specification Document based on the details provided in the Business Requirement Document (BRD) for the Mutual Fund App.

**Use Case Specification Document for Mutual Fund App**

**1. Introduction**

**1.1 Purpose**

The purpose of this document is to detail the use cases for the Mutual Fund App, providing a clear understanding of the system's functionality from the user's perspective.

**1.2 Scope**

This document covers the primary use cases, actors, preconditions, postconditions, main flow, and alternative flows for the key functionalities of the Mutual Fund App.

**2. Actors**

* **User**: An individual investor using the app.
* **System**: The Mutual Fund App.
* **Payment Gateway**: External system for processing payments.
* **Customer Support**: Support personnel assisting users.

**3. Use Case Specifications**

**Use Case 1: User Registration and Login**

**3.1.1 Use Case Name: User Registration**

* **Primary Actor**: User
* **Preconditions**: User has the app installed.
* **Postconditions**: User is registered in the system.
* **Main Flow**:
  1. User opens the app and selects "Register".
  2. System displays registration form.
  3. User enters email, phone number, and creates a password.
  4. User submits the form.
  5. System validates the input and creates a new user account.
  6. System sends a confirmation email/SMS to the user.
  7. User confirms registration via email/SMS.
* **Alternative Flows**:
  1. A1: Invalid input data. System prompts the user to correct the input.

**3.1.2 Use Case Name: User Login**

* **Primary Actor**: User
* **Preconditions**: User is registered.
* **Postconditions**: User is logged into the system.
* **Main Flow**:
  1. User opens the app and selects "Login".
  2. System displays login form.
  3. User enters email/phone number and password.
  4. User submits the form.
  5. System validates the credentials.
  6. System logs the user in and redirects to the dashboard.
* **Alternative Flows**:
  1. A1: Invalid credentials. System displays an error message.

**Use Case 2: User Dashboard**

**3.2.1 Use Case Name: View Dashboard**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views the dashboard.
* **Main Flow**:
  1. User accesses the dashboard from the main menu.
  2. System retrieves user portfolio data.
  3. System displays total investment, current value, and returns.
  4. System shows recent transactions and notifications.

**Use Case 3: Mutual Fund Search and Filter**

**3.3.1 Use Case Name: Search Mutual Funds**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views search results.
* **Main Flow**:
  1. User selects "Search Mutual Funds".
  2. System displays the search bar.
  3. User enters search criteria (e.g., fund name, AMC).
  4. User submits the search.
  5. System retrieves and displays matching mutual funds.
* **Alternative Flows**:
  1. A1: No results found. System displays a "no results" message.

**3.3.2 Use Case Name: Filter Mutual Funds**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views filtered results.
* **Main Flow**:
  1. User selects "Filter" option in the mutual funds section.
  2. System displays filter options (e.g., equity, debt, hybrid).
  3. User selects filter criteria.
  4. User applies the filter.
  5. System retrieves and displays filtered mutual funds.

**Use Case 4: Investment Options**

**3.4.1 Use Case Name: Invest in Mutual Fund**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User completes the investment.
* **Main Flow**:
  1. User selects a mutual fund to invest in.
  2. System displays investment options (lump sum, SIP).
  3. User selects investment type.
  4. User enters investment amount and payment details.
  5. User submits the investment.
  6. System processes the payment through Payment Gateway.
  7. Payment Gateway confirms the transaction.
  8. System updates user's portfolio.
  9. System sends confirmation notification to the user.
* **Alternative Flows**:
  1. A1: Payment failure. System displays an error message and allows retry.

**3.4.2 Use Case Name: Compare Mutual Funds**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views comparison results.
* **Main Flow**:
  1. User selects "Compare Mutual Funds".
  2. System displays mutual funds comparison interface.
  3. User selects multiple mutual funds to compare.
  4. User initiates the comparison.
  5. System retrieves and displays comparison data (e.g., NAV, returns, risk level).

**Use Case 5: Transaction Management**

**3.5.1 Use Case Name: View Transaction History**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views transaction history.
* **Main Flow**:
  1. User selects "Transaction History".
  2. System retrieves transaction data.
  3. System displays the transaction history.

**Use Case 6: Notifications and Alerts**

**3.6.1 Use Case Name: View Notifications**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views notifications.
* **Main Flow**:
  1. User selects "Notifications".
  2. System retrieves notification data.
  3. System displays notifications (e.g., NAV updates, dividends).

**3.6.2 Use Case Name: Set Custom Alerts**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User sets custom alerts.
* **Main Flow**:
  1. User selects "Set Alerts".
  2. System displays alert options.
  3. User configures alert criteria (e.g., NAV thresholds).
  4. User saves the alert settings.
  5. System stores alert configurations and monitors criteria.

**Use Case 7: Reports and Statements**

**3.7.1 Use Case Name: View Reports**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User views investment reports.
* **Main Flow**:
  1. User selects "Reports".
  2. System retrieves report data.
  3. System displays investment reports (e.g., portfolio performance).

**3.7.2 Use Case Name: Download Statements**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User downloads investment statements.
* **Main Flow**:
  1. User selects "Download Statements".
  2. System generates statement PDF.
  3. System provides download link.
  4. User downloads the statement.

**Use Case 8: Customer Support**

**3.8.1 Use Case Name: Contact Customer Support**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User contacts customer support.
* **Main Flow**:
  1. User selects "Contact Support".
  2. System displays support options (e.g., chat, call, email).
  3. User chooses a support option.
  4. System initiates the chosen support method.

**3.8.2 Use Case Name: Use Chatbot**

* **Primary Actor**: User
* **Preconditions**: User is logged in.
* **Postconditions**: User interacts with chatbot.
* **Main Flow**:
  1. User selects "Chatbot".
  2. System displays chatbot interface.
  3. User enters query.
  4. System provides automated response or directs to human support if needed.

**4. Appendix**

**4.1 Glossary**

* **NAV**: Net Asset Value
* **SIP**: Systematic Investment Plan
* **AMC**: Asset Management Company

**4.2 References**

* SEBI Guidelines for Mutual Funds
* AMFI Best Practices
* **Requirements Traceability Matrix (RTM) for Mutual Fund App**

| **Requirement ID** | **Requirement Description** | **Use Case ID** | **Use Case Description** | **Test Case ID** | **Test Case Description** |
| --- | --- | --- | --- | --- | --- |
| FR1.1 | Users should be able to register using email, phone number, or social media accounts. | UC1.1 | User Registration | TC1.1.1 | Verify registration with email. |
|  |  |  |  | TC1.1.2 | Verify registration with phone number. |
|  |  |  |  | TC1.1.3 | Verify registration with social media. |
| FR1.2 | Users should be able to log in using their registered credentials. | UC1.2 | User Login | TC1.2.1 | Verify login with valid credentials. |
|  |  |  |  | TC1.2.2 | Verify login with invalid credentials. |
| FR2.1 | Display an overview of the user’s portfolio including total investment, current value, and returns. | UC2.1 | View Dashboard | TC2.1.1 | Verify dashboard displays total investment. |
|  |  |  |  | TC2.1.2 | Verify dashboard displays current value. |
|  |  |  |  | TC2.1.3 | Verify dashboard displays returns. |
| FR2.2 | Show recent transactions and notifications. | UC2.1 | View Dashboard | TC2.1.4 | Verify dashboard displays recent transactions. |
|  |  |  |  | TC2.1.5 | Verify dashboard displays notifications. |
| FR3.1 | Users should be able to search for mutual funds by name, AMC, or fund manager. | UC3.1 | Search Mutual Funds | TC3.1.1 | Verify search by fund name. |
|  |  |  |  | TC3.1.2 | Verify search by AMC. |
|  |  |  |  | TC3.1.3 | Verify search by fund manager. |
| FR3.2 | Users should be able to filter mutual funds based on categories like equity, debt, hybrid, etc. | UC3.2 | Filter Mutual Funds | TC3.2.1 | Verify filter by equity. |
|  |  |  |  | TC3.2.2 | Verify filter by debt. |
|  |  |  |  | TC3.2.3 | Verify filter by hybrid. |
| FR4.1 | Users should be able to invest in mutual funds via lump sum or SIP. | UC4.1 | Invest in Mutual Fund | TC4.1.1 | Verify lump sum investment. |
|  |  |  |  | TC4.1.2 | Verify SIP investment. |
| FR4.2 | Provide a comparison tool to compare different mutual funds. | UC4.2 | Compare Mutual Funds | TC4.2.1 | Verify comparison of multiple funds. |
| FR5.1 | Support secure payment options for transactions. | UC4.1 | Invest in Mutual Fund | TC5.1.1 | Verify secure payment processing. |
| FR5.2 | Users should be able to view transaction history. | UC5.1 | View Transaction History | TC5.2.1 | Verify transaction history display. |
| FR6.1 | Send notifications for NAV updates, dividends, and important announcements. | UC6.1 | View Notifications | TC6.1.1 | Verify NAV update notifications. |
|  |  |  |  | TC6.1.2 | Verify dividend notifications. |
|  |  |  |  | TC6.1.3 | Verify important announcement notifications. |
| FR6.2 | Allow users to set custom alerts for fund performance thresholds. | UC6.2 | Set Custom Alerts | TC6.2.1 | Verify setting custom alerts. |
|  |  |  |  | TC6.2.2 | Verify alert notifications based on set criteria. |
| FR7.1 | Provide detailed investment statements and reports. | UC7.1 | View Reports | TC7.1.1 | Verify display of investment reports. |
| FR7.2 | Allow users to download or email reports. | UC7.2 | Download Statements | TC7.2.1 | Verify downloading of investment statements. |
| FR8.1 | Integrate a chatbot for basic queries. | UC8.2 | Use Chatbot | TC8.2.1 | Verify chatbot response to basic queries. |
| FR8.2 | Provide contact options for customer support. | UC8.1 | Contact Customer Support | TC8.1.1 | Verify contact options for customer support. |
| NFR1.1 | Ensure data encryption during storage and transmission. | UC4.1 | Invest in Mutual Fund | TC1.1.1 | Verify data encryption during payment. |
| NFR1.2 | Implement multi-factor authentication (MFA) for user accounts. | UC1.2 | User Login | TC1.2.3 | Verify multi-factor authentication. |
| NFR2.1 | The app should load the user dashboard within 3 seconds. | UC2.1 | View Dashboard | TC2.1.6 | Verify dashboard load time. |
| NFR2.2 | Transactions should be processed within 5 seconds. | UC4.1 | Invest in Mutual Fund | TC4.1.3 | Verify transaction processing time. |
| NFR3.1 | The app should have an intuitive user interface. | All | All | TC3.1.4 | Verify the intuitiveness of the user interface. |
| NFR3.2 | Provide help and support documentation within the app. | All | All | TC8.1.2 | Verify availability of help and support documentation. |
| NFR4.1 | Ensure compliance with SEBI and AMFI regulations. | All | All | TC4.1.4 | Verify compliance with SEBI and AMFI regulations. |
| NFR4.2 | Adhere to GDPR for data privacy and protection. | All | All | TC4.1.5 | Verify GDPR compliance. |

**Process – Flow Diagram:**

**User Registration and Login**:

* User opens the app and selects to register or log in.
* The system displays the registration or login form.
* User enters the details, and the system validates them.
* If the details are valid, the system creates an account or logs in the user and sends a confirmation notification. If not, it shows an error message.

**User Dashboard**:

* User accesses the dashboard.
* The system displays the portfolio overview and recent transactions and notifications.

**Mutual Fund Search and Filter**:

* User searches for mutual funds by entering search criteria.
* The system retrieves and displays the search results.
* User applies filters, and the system retrieves and displays the filtered results.

**Investment Options**:

* User selects a mutual fund and chooses the investment type.
* User enters the investment amount and payment details.
* The system processes the payment, updates the portfolio, and sends a confirmation notification.

**Transaction Management**:

* User views the transaction history.
* The system retrieves and displays the transaction history.

**Notifications and Alerts**:

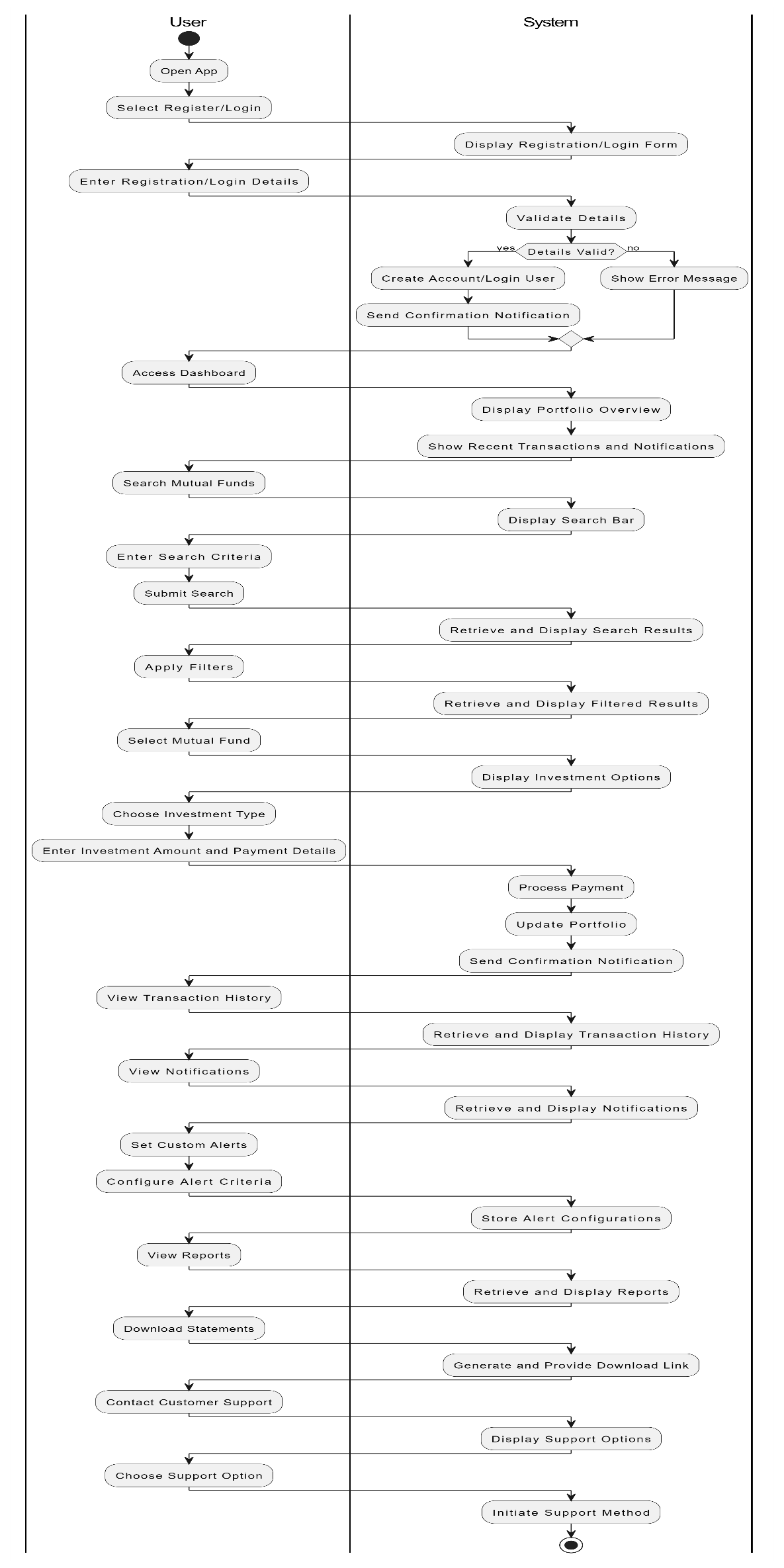
* User views notifications.
* The system retrieves and displays notifications.
* User sets custom alerts by configuring alert criteria.
* The system stores the alert configurations.

**Reports and Statements**:

* User views reports.
* The system retrieves and displays the reports.
* User downloads statements.
* The system generates and provides a download link for the statements.

**Customer Support**:

* User contacts customer support.
* The system displays support options.
* User chooses a support option.
* The system initiates the chosen support method.



A RACI chart (Responsible, Accountable, Consulted, and Informed) helps clarify roles and responsibilities in a project. Below is the RACI chart for the Mutual Fund App project based on the provided details:

**Legend:**

* **R (Responsible)**: The person who does the work to complete the task.
* **A (Accountable)**: The person who is ultimately answerable for the correct and thorough completion of the deliverable or task.
* **C (Consulted)**: Those whose opinions are sought, typically subject matter experts; and with whom there is two-way communication.
* **I (Informed)**: Those who are kept up-to-date on progress, often only on completion of the task or deliverable; and with whom there is just one-way communication.

**Explanation:**

* The **Project Manager** is responsible for overseeing the entire project and ensuring tasks are completed on time.
* The **Product Owner** is accountable for the overall product and ensures the project meets the business requirements.
* The **Business Analyst** gathers requirements and ensures they are well-documented.
* The **Development Team** is responsible for coding and implementing the features.
* The **QA Team** is responsible for testing the product to ensure it meets quality standards.
* The **UI/UX Designer** is responsible for designing the user interface and user experience.
* The **Security Expert** ensures that all security measures are implemented and followed.
* The **Marketing Team** is responsible for promoting the app and ensuring it reaches the target audience.
* The **Support Team** handles post-deployment support and user training.

**RACI – Chart:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Project Manager** | **Product Owner** | **Business Analyst** | **Developer** | **Testing** | **UI/UX Designer** |
| **Requirements Gathering** | A | R | R | C | C | C |
| **User Registration and Login Design** | C | A | R | C | I | R |
| **User Dashboard Design** | C | A | R | C | I | R |
| **Development of User Dashboard** | I | I | I | R | C | C |
| **User Acceptance Testing** | I | R | C | I | R | I |
| **Deployment** | R | I | I | R | I | I |

# Software Requirements Specification (SRS) Document for Mutual Fund App

## Table of Contents

1. **Introduction**
   * 1.1 Purpose
   * 1.2 Scope
   * 1.3 Definitions, Acronyms, and Abbreviations
   * 1.4 References
   * 1.5 Overview
2. **Overall Description**
   * 2.1 Product Perspective
   * 2.2 Product Functions
   * 2.3 User Classes and Characteristics
   * 2.4 Operating Environment
   * 2.5 Design and Implementation Constraints
   * 2.6 Assumptions and Dependencies
3. **Specific Requirements**
   * 3.1 Functional Requirements
   * 3.2 Non-Functional Requirements
   * 3.3 External Interface Requirements
   * 3.4 System Features
4. **Appendices**

## 1. Introduction

### 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the requirements for the development of the Mutual Fund App. It aims to provide a detailed description of the app's functionalities, user interactions, system constraints, and external interfaces.

### 1.2 Scope

The Mutual Fund App is intended to provide users with a platform to manage their mutual fund investments efficiently. The app will facilitate user registration, account management, investment tracking, notifications, and support services related to mutual funds.

### 1.3 Definitions, Acronyms, and Abbreviations

* **NAV**: Net Asset Value
* **SIP**: Systematic Investment Plan
* **AMC**: Asset Management Company
* **SEBI**: Securities and Exchange Board of India
* **AMFI**: Association of Mutual Funds in India
* **API**: Application Programming Interface
* **UI**: User Interface
* **UX**: User Experience

### 1.4 References

* SEBI Guidelines for Mutual Funds
* AMFI Best Practices
* GDPR Compliance Documentation

### 1.5 Overview

This SRS document provides an overview of the Mutual Fund App, including its functional and non-functional requirements. It outlines the system's user classes, operating environment, design constraints, assumptions, and dependencies.

## 2. Overall Description

### 2.1 Product Perspective

The Mutual Fund App is a standalone application that allows individual investors to manage their mutual fund investments. It interfaces with external systems for payment processing, mutual fund data retrieval, and regulatory compliance.

### 2.2 Product Functions

Key functions of the Mutual Fund App include:

* User registration and authentication
* Portfolio management
* Mutual fund search and filtering
* Investment options (lump sum and SIP)
* Transaction tracking and history
* Notifications and alerts
* Reporting and statements
* Customer support and assistance

### 2.3 User Classes and Characteristics

* **Individual Investors**: Users who manage their personal investments through the app.
  + Basic understanding of mutual funds and financial investments
  + Varied levels of experience in using digital financial services

### 2.4 Operating Environment

The Mutual Fund App will operate on:

* **Mobile Platforms**: Android and iOS
* **Web Browsers**: Latest versions of Chrome, Firefox, Safari, and Edge

### 2.5 Design and Implementation Constraints

* Compliance with SEBI regulations for mutual fund investments
* Integration with secure payment gateways for transactions
* Data encryption and protection in accordance with GDPR standards

### 2.6 Assumptions and Dependencies

* Users have access to stable internet connectivity for app usage.
* Third-party APIs for mutual fund data and payment processing are reliable and accessible.
* Compliance with regulatory changes from SEBI and other financial authorities.

## 3. Specific Requirements

### 3.1 Functional Requirements

#### 3.1.1 User Registration and Authentication

* **FR1.1**: The app shall allow users to register using email, phone number, or social media accounts.
* **FR1.2**: The app shall authenticate users securely using credentials or biometric authentication.

#### 3.1.2 Portfolio Management

* **FR2.1**: The app shall display a summary of the user's mutual fund portfolio, including current investments and returns.
* **FR2.2**: The app shall provide options to add, edit, and delete investments from the portfolio.

#### 3.1.3 Mutual Fund Search and Filtering

* **FR3.1**: The app shall enable users to search for mutual funds by name, category, AMC, or fund manager.
* **FR3.2**: The app shall allow users to filter mutual funds based on criteria such as risk profile, performance, and expense ratio.

#### 3.1.4 Investment Options

* **FR4.1**: The app shall support investments in mutual funds through lump sum and SIP methods.
* **FR4.2**: The app shall provide tools to compare different mutual funds based on historical performance and risk metrics.

#### 3.1.5 Transaction Tracking and History

* **FR5.1**: The app shall maintain a record of user transactions, including purchases, redemptions, and dividends.
* **FR5.2**: The app shall allow users to view detailed transaction history and download statements.

#### 3.1.6 Notifications and Alerts

* **FR6.1**: The app shall notify users of important updates such as NAV changes, dividend declarations, and regulatory announcements.
* **FR6.2**: The app shall allow users to set personalized alerts for specific mutual fund performance thresholds.

#### 3.1.7 Reporting and Statements

* **FR7.1**: The app shall generate periodic investment reports summarizing portfolio performance, asset allocation, and tax implications.
* **FR7.2**: The app shall provide downloadable statements for tax reporting and auditing purposes.

#### 3.1.8 Customer Support

* **FR8.1**: The app shall integrate a support center with FAQs, troubleshooting guides, and user forums.
* **FR8.2**: The app shall offer direct communication channels for users to contact customer support representatives.

### 3.2 Non-Functional Requirements

#### 3.2.1 Security

* **NFR1.1**: User data shall be encrypted during transmission and storage to prevent unauthorized access.
* **NFR1.2**: The app shall implement multi-factor authentication (MFA) for user accounts to enhance security.

#### 3.2.2 Performance

* **NFR2.1**: The app shall load the dashboard and transaction screens within 3 seconds on average internet connection speeds.
* **NFR2.2**: Transactions shall be processed and reflected in the user's portfolio in real-time with minimal latency.

#### 3.2.3 Usability

* **NFR3.1**: The app shall have an intuitive and responsive user interface across mobile and web platforms.
* **NFR3.2**: The app shall provide contextual help and tooltips to guide users through complex features and investment decisions.

#### 3.2.4 Compliance

* **NFR4.1**: The app shall comply with SEBI regulations and guidelines related to mutual fund investments and disclosures.
* **NFR4.2**: The app shall adhere to GDPR requirements for data privacy and protection of user information.

### 3.3 External Interface Requirements

#### 3.3.1 User Interfaces

* The app shall provide a consistent user interface design following UI/UX best practices.
* The app shall support responsive design principles for seamless user experience across devices.

#### 3.3.2 Hardware Interfaces

* The app shall utilize standard mobile device hardware features such as camera and GPS for user authentication and document upload.

#### 3.3.3 Software Interfaces

* The app shall integrate with third-party APIs for mutual fund data retrieval and payment processing.
* The app shall support integration with analytics tools for monitoring user behavior and app performance.

#### 3.3.4 Communication Interfaces

* The app shall use secure HTTPS protocols for communication between the client-side app and server-side API endpoints.
* The app shall implement push notifications for timely updates and alerts to users.

### 3.4 System Features

#### 3.4.1 User Registration and Authentication

* **Description**: Enables users to create accounts securely and authenticate themselves to access app features.
* **Priority**: High
* **Inputs**: User registration details, login credentials
* **Outputs**: User authentication token, access to user-specific data

#### 3.4.2 Portfolio Management

* **Description**: Allows users to manage their mutual fund investments, track performance, and make informed decisions.
* **Priority**: High
* **Inputs**: Investment details, transaction records
* **Outputs**: Portfolio summary, investment reports

#### 3.4.3 Mutual Fund Search and Filtering

* **Description**: Facilitates easy discovery and selection of mutual funds based on user-defined criteria.
* **Priority**: Medium
* **Inputs**: Search keywords, filtering options
* **Outputs**: List of matching mutual funds, filtered results

#### 3.4.4 Investment Options

* **Description**: Provides flexible investment methods (lump sum and SIP) tailored to user preferences and financial goals.
* **Priority**: High
* **Inputs**: Investment amount, duration, risk tolerance
* **Outputs**: Investment confirmation, periodic investment statements

#### 3.4.5 Transaction Tracking and History

* **Description**: Maintains a comprehensive record of user transactions for transparency and accountability.
* **Priority**: Medium
* **Inputs**: Transaction details, user actions
* **Outputs**: Transaction history, downloadable statements

#### 3.4.6 Notifications and Alerts

* **Description**: Notifies users of important updates and allows customization of alerts based on user preferences.
* **Priority**: Medium
* **Inputs**: Notification triggers, user-defined alert settings
* **Outputs**: Real-time notifications, alert messages

#### 3.4.7 Reporting and Statements

* **Description**: Generates detailed reports summarizing portfolio performance and tax-related information.
* **Priority**: Medium
* **Inputs**: Investment data, tax implications
* **Outputs**: Investment reports, tax statements

#### 3.4.8 Customer Support

* **Description**: Provides users with access to help resources, support tickets, and direct assistance channels.
* **Priority**: Medium
* **Inputs**: User queries, support requests
* **Outputs**: Help articles, support responses

## 4. Appendices

* Appendix A: Glossary of Terms
* Appendix B: User Interface Mockups
* Appendix C: Data Flow Diagrams
* Appendix D: API Specifications