American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science &Technology (FST)

Summer 22 23

Section: F

Software Quality Assurance and Testing

FundFlow Navigator

A Report submitted
By

SN	Student Name	Student ID
1	Md. Mintajur Rahman Emon	20-43870-2
2	Sadman Sanid Tanim	20-43866-2
3	Md. Tanvir Hossen	20-42488-1
4	Luisa Khaleque Trisha	20-42548-1

Software Test Plan

for

FundFlow Navigator

Version 1.0 approved

Prepared by Md. Mintajur Rahman Emon, Sadman Sanid Tanim, Md. Tanvir Hossen, Luisa Khaleque Trisha

American International University-Bangladesh

25-08-2023

Table of Contents

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.03.04	Md. Mintajur Rahman Emon	First Draft
0.2	2023.03.20	Sadman Sanid Tanim	Front-End UI/UX is completed
0.3	2023.04.30	Md. Mintajur Rahman Emon	Back-end development is completed with unit testing
0.4	2023.05.08	Luisa Khaleque Trisha	Data Processing and Dependencies are checked
0.5	2023.05.20	Md. Mintajur Rahman Emon	First Integration of the modules are done
0.6	2023.07.29	Md. Tanvir Hossen	Integration testing for all modules is executed
0.7	2023.05.20	Md. Mintajur Rahman Emon	Second Draft

1. TEST PLAN IDENTIFIER: TP_FundFlowNavigator_1.03

2. REFERENCES

- https://www.gcreddy.com/2021/07/test-plan-documentation.html

3. INTRODUCTION

Background to the Problem

In today's fast-paced and ever-changing financial landscape, achieving and maintaining a healthy financial status has become increasingly complex. Navigating through various investment options, keeping track of expenses, and making informed decisions that align with personal financial goals can be daunting tasks. Recognizing this challenge, our team has developed the "FundFlow Navigator" mobile application software. This innovative solution is designed to revolutionize the way individuals manage and improve their financial health.

In a world where financial choices are abundant and intricate, people often struggle to make optimal decisions for their financial well-being. The lack of real-time insights and personalized guidance leaves many individuals unsure about the right path to take. Existing financial management tools might provide basic tracking features, but they fall short in offering intelligent, data-driven recommendations based on both personal financial situations and market dynamics. There is a clear need for a comprehensive and proactive solution that empowers users to take control of their financial journey.

The FundFlow Navigator project stands at the intersection of cutting-edge technology and the pressing need for comprehensive financial guidance. By offering real-time insights, personalized recommendations, and a user-friendly interface, this mobile application promises to be a game-changer in the realm of personal finance management. It aims to empower users to navigate their financial journeys with confidence, making informed decisions that pave the way for a secure and prosperous future.

The root cause of the problem described is the increasing complexity and dynamism of the modern financial landscape, coupled with the lack of personalized guidance and real-time insights. Several contributing factors can be identified:

Complex Financial Options: The proliferation of various investment vehicles, savings accounts, loans, and financial products has created a complex ecosystem that individuals must navigate. The

multitude of choices available makes it challenging for individuals to determine the most suitable options for their unique financial situations.

Market Volatility: Financial markets are subject to constant fluctuations and unpredictable shifts. These dynamics impact investment values, interest rates, and overall financial performance. Individuals without access to up-to-date information and expert insights may struggle to make decisions that yield the best outcomes.

Personalization Gap: Generic financial management tools often fail to account for individual circumstances, goals, and risk tolerances. Without tailored guidance, users might make decisions that do not align with their specific needs, leading to suboptimal results.

Lack of Financial Literacy: Many people have limited understanding of financial concepts and strategies. This lack of financial literacy can hinder effective decision-making and cause individuals to overlook important opportunities or fall into financial pitfalls.

The described problems hold significant importance due to their far-reaching impact on individuals' financial well-being and overall quality of life. So, our project “FundFlow Navigator will help in:

Financial Security: Sound financial decisions are essential for personal and family security. Making suboptimal choices can lead to debt accumulation, insufficient savings, and even financial crises that have a profound impact on individuals' and families' lives.

Wealth Accumulation: Effective financial management is key to accumulating wealth and achieving long-term financial goals, such as homeownership, education, retirement, and entrepreneurship. Without proper guidance, these goals may remain out of reach.

Reduced Stress: Financial instability and uncertainty contribute to stress and anxiety. Providing individuals with tools and insights to manage their finances can alleviate this stress, leading to improved mental health.

Economic Growth: On a broader scale, a population with better financial management skills can contribute to overall economic growth. Individuals who make informed decisions are more likely to invest, save, and contribute positively to the economy.

Empowerment: Equipping individuals with real-time insights and personalized recommendations empowers them to take control of their financial destinies. This empowerment fosters a sense of agency and responsibility over one's financial future.

Bridge the Gap: The described problem bridges the gap between advanced financial knowledge and everyday individuals. It addresses the need for democratizing access to sophisticated financial insights that were traditionally available only to financial experts.

Lastly, the root cause of the problem lies in the complexity of the financial landscape and the lack of personalized guidance. This problem is crucial to consider because it affects individuals' financial security, wealth accumulation, stress levels, overall well-being, and the broader economy. Addressing this problem has the potential to transform how people approach their financial decisions and lead to more secure and prosperous futures.

Solution to the Problem

The Solution: FundFlow Navigator

“FundFlow Navigator” is a cutting-edge mobile application that leverages the power of Artificial Intelligence to address these challenges. By combining advanced AI algorithms with real-time market data, the application offers users a holistic view of their financial landscape. From tracking income and expenses to assessing investments and debts, the platform provides a 360-degree snapshot of the user's financial situation.

To address the challenges outlined in the problem description, we propose the following solutions within the context of the "FundFlow Navigator" project:

AI-Powered Financial Insights:

Develop an advanced AI engine that utilizes machine learning algorithms to analyze users' financial data and market trends. This engine will generate real-time insights and personalized recommendations, helping users make informed decisions aligned with their financial goals.

Goal-Oriented Planning:

Implement a feature that allows users to set and prioritize their financial goals, such as saving for a house, paying off debt, or building an emergency fund. The application should break down these goals into achievable milestones, offering a clear roadmap for users to follow.

Real-Time Monitoring and Alerts:

Enable users to monitor their financial progress on a daily basis. Implement alerts and notifications that inform users about deviations from their financial plans, market shifts, or missed milestones.

Educational Resources:

Create an educational hub within the application that offers easy-to-understand resources on financial concepts, investment strategies, and risk management. Empowering users with financial literacy will enhance their decision-making capabilities.

Customized Investment Recommendations:

Incorporate AI-driven investment recommendations that consider users' risk tolerance, financial goals, and market conditions. The system should suggest diversified investment portfolios to help users maximize returns while managing risks.

Expense Tracking and Budgeting:

Develop robust expense tracking and budgeting tools that allow users to categorize and monitor their spending habits. The application should provide insights into areas where users can cut back to save more.

Appropriateness of the Solution:

The proposed solutions are particularly appropriate for several reasons-

Personalization: The solutions prioritize personalization, catering to individual financial situations, goals, and risk appetites. This is vital to ensure that users receive recommendations that align with their unique circumstances.

Data-Driven Insights: By leveraging AI and real-time market data, the solutions provide intelligent, data-driven recommendations that go beyond basic tracking features. This addresses the root problem of lack of timely insights.

Empowerment and Education: The solutions not only offer recommendations but also educate users on financial concepts and strategies. This bridges the knowledge gap and empowers users to make informed decisions independently.

Holistic Approach: The combination of expense tracking, goal setting, investment recommendations, and education offers a holistic approach to financial management. Users are guided through every aspect of their financial journey.

User Engagement: The real-time monitoring, alerts, and educational resources enhance user engagement, encouraging consistent and active usage of the application.

Feasibility and Business Objectives:

The proposed solutions are feasible given the advancements in AI technology, mobile app development, and financial data integration. The success of similar AI-driven financial platforms in the market demonstrates the viability of these solutions.

The solutions are aligned with the business objective of the "FundFlow Navigator" project, which is to provide users with a comprehensive and proactive tool to navigate their financial journey. The solutions address the core challenges highlighted in the problem description and offer a user-friendly, AI-powered platform that empowers users to make sound financial decisions.

By providing users with real-time insights, personalized guidance, and educational resources, the proposed solutions not only meet the business objective but also have the potential to transform how individuals approach their finances, leading to greater financial security and success.

FundFlow Navigator is an innovative mobile application designed to revolutionize the way individuals manage their finances and achieve their financial goals. With the ever-increasing complexity of the financial world, our solution provides users with the tools, insights, and guidance needed to navigate this landscape with confidence and success.

Purpose:

The primary purpose of FundFlow Navigator is to empower users to achieve and maintain a healthy financial status by providing real-time insights, personalized recommendations, and educational resources. The application serves as a comprehensive financial companion, assisting users in making informed decisions that align with their unique financial situations and goals.

Benefits:

Intelligent Insights: FundFlow Navigator utilizes advanced AI algorithms to analyze users' financial data and current market trends. This results in tailored insights and recommendations that empower users to make optimal decisions about investments, expenses, and savings.

Personalized Guidance: The application understands the individual goals, risk tolerance, and financial preferences of each user. This personalized approach ensures that the recommendations provided are relevant and aligned with the user's aspirations.

Real-Time Monitoring: Users can monitor their financial progress in real time, tracking how their decisions impact their overall financial health. This dynamic tracking fosters accountability and allows users to adjust strategies as needed.

Educational Resources: FundFlow Navigator is not just a tracking tool but an educational hub, providing users with accessible explanations of financial concepts, strategies, and best practices. This enhances users' financial literacy and empowers them to make more informed choices.

Objectives and Goals:

Comprehensive Financial Management: FundFlow Navigator aims to offer a holistic solution that covers all aspects of financial management, from tracking expenses to making investment decisions.

Empowerment: The application seeks to empower users by providing them with the information and tools necessary to make confident and strategic financial decisions.

Financial Literacy: FundFlow Navigator aims to bridge the gap in financial knowledge by offering educational resources that empower users to better understand and navigate the world of finance.

Goal Achievement: The application's goal-setting features help users define their financial objectives and provide step-by-step guidance to achieve them.

Enhanced User Experience: The user-friendly interface and intuitive design ensure that users of all backgrounds can easily navigate the application and utilize its features effectively.

FundFlow Navigator is a powerful tool that combines AI-driven insights, personalized guidance, and educational resources to empower users in their financial journey. By addressing the complexities of modern finance and providing actionable recommendations, the application helps users achieve financial security and success.

As per the recent study there are several financial tracking apps and software present. But none of them has the ability to do computation on real time up to date data and generate a decision for investing in particular sector. There are no AI-powered apps, software or, tools available right now in the market of software products. Though, there are somewhat relatable software and apps present-

Budgeting Apps:

There are various budgeting apps available that allow users to track their income and expenses. These apps provide insights into spending patterns and help users create budgets. However, they often lack the sophistication of AI-driven insights and personalized recommendations.

Personal Finance Management Software:

Software like Quicken and Mint offer comprehensive personal finance management features, including expense tracking, investment tracking, and budgeting. However, their recommendations might not be as advanced as those offered by AI-driven solutions.

Robo-Advisors:

Robo-advisors are platforms that use algorithms to create and manage investment portfolios for users. They consider factors like risk tolerance and financial goals to suggest investment strategies. While they provide investment advice, they might not cover the broader financial management aspects.

Financial Planning Tools:

Financial planning software helps users create long-term financial plans, considering factors like retirement, education funding, and estate planning. These tools might not offer real-time insights or address day-to-day financial decisions.

Investment Research Platforms:

Some platforms provide market research and insights for users to make informed investment decisions. However, they might lack the integration of personal financial data and overall financial health assessment.

Educational Websites and Blogs:

Various websites and blogs offer financial education and guidance. While these resources are valuable for improving financial literacy, they might not provide the personalized insights needed for individual financial situations.

Banking and Financial Institution Apps:

Many banks and financial institutions offer their own apps with basic tracking features, but they might focus primarily on the services provided by the institution rather than holistic financial management.

It's worth noting that while these solutions provide value in their respective areas, the problem described in the project background calls for a more comprehensive and proactive approach. The need for real-time insights, personalized guidance, and AI-driven recommendations based on both personal financial situations and market dynamics is what sets FundFlow Navigator apart as a potential solution. It aims to bridge the gap between basic tracking tools and more advanced financial management by leveraging the power of AI to empower users in their financial journey.

4. REQUEIREMNT SPECIFICATION

4.1 System Features

1. User Registration and Authentication:

Functional Requirements

1.1 The software shall allow users to create new accounts by providing a unique username, password, and valid email address.

1.2 Users shall receive a verification email upon registration, and their account shall be activated upon clicking the verification link.

1.3 The system shall provide a "Forgot Password" functionality to allow users to reset their passwords via email.

1.4 The application shall enforce password complexity requirements (e.g., minimum length, combination of characters).

1.5 In case of multiple unsuccessful login attempts, the system shall display a CAPTCHA to prevent brute-force attacks.

Priority Level: High

Precondition: User has a valid email address for registration.

2. Personal Financial Dashboard:

Functional Requirements

2.1 The application shall display an interactive dashboard upon user login.

2.2 Users shall be able to view their current balance, recent transactions, and an overview of financial goals on the dashboard.

2.3 The system shall update the dashboard in real-time to reflect the most current financial information.

2.4 The dashboard shall provide a summary of income sources and expense categories.

Priority Level: High

Precondition: User is logged in.

3. Expense Tracking and Categorization:

Functional Requirements

3.1 Users shall be able to manually enter expenses, specifying the amount, date, and category.

3.2 The application shall provide predefined expense categories and allow users to create custom categories.

3.3 Users shall have the ability to edit or delete previously entered expenses.

3.4 The system shall allow users to upload receipts or images related to expenses.

Priority Level: High

Precondition: User is logged in.

4. Investment Portfolio Management:

Functional Requirements

4.1 Users shall be able to add, edit, or delete investment holdings, including type, quantity, and purchase price.

4.2 The system shall retrieve real-time market data to calculate the current value of each investment.

4.3 Users shall receive notifications for significant changes in the value of their investments.

4.4 The application shall provide a historical performance chart for each investment.

Priority Level: High

Precondition: User is logged in.

5. Goal Setting and Tracking:

Functional Requirements

5.1 Users shall have the ability to set financial goals with specific target amounts and deadlines.

5.2 The system shall break down larger goals into achievable milestones.

5.3 Users shall receive progress updates and notifications when milestones are reached.

5.4 The application shall provide a visual representation of goal progress.

Priority Level: Medium

Precondition: User is logged in.

6. AI-Generated Insights and Recommendations:

Functional Requirements

6.1 The application shall analyze user's financial data and market trends to generate personalized insights.

6.2 Users shall receive actionable recommendations for optimizing expenses and investments.

6.3 The system shall take into account user preferences and risk tolerance when generating recommendations.

6.4 Recommendations shall be presented in a clear and understandable format.

Priority Level: High

Precondition: User is logged in and has provided financial data.

7. Financial Education Hub:

Functional Requirements

7.1 The application shall provide educational resources and articles on financial concepts and strategies.

7.2 Users shall have access to tutorials, guides, and videos to improve their financial literacy.

Priority Level: Low

Precondition: User is logged in.

8. Real-Time Monitoring and Alerts:

Functional Requirements

8.1 Users shall receive real-time alerts about significant changes in their financial situation.

8.2 Alerts may include overspending, missed goals, or investment opportunities.

8.3 Users shall have the option to customize alert preferences.

Priority Level: High

Precondition: User is logged in and enabled the alert system.

9. Data Security and Privacy:

Functional Requirements

9.1 The application shall ensure user data encryption during transmission and storage.

9.2 Users shall have the ability to control their data sharing preferences.

9.3 The system shall comply with relevant data protection regulations.

Priority Level: High

Precondition: User is logged with personalized valid email.

10. User Support:

Functional Requirements

10.1 Users shall have access to a support center to address technical issues and inquiries.

10.2 The application shall provide a comprehensive FAQ section and contact options for customer support.

Priority Level: High

Precondition: User is logged in and agreed with the app's terms and conditions.

4.2 System Quality Attributes

Usability: An average user should be able to navigate and utilize the main features of the FundFlow Navigator app within 5 minutes of their first interaction, without requiring extensive training.

Accuracy: The AI algorithms used in FundFlow Navigator should consistently provide accurate financial insights, recommendations, and market predictions with a margin of error of less than 5%.

Real-time Data: The application should ensure that all financial data, including market information and user transactions, is updated in real-time, minimizing data latency to less than 1 minute.

Personalization: The AI should tailor its recommendations based on individual user preferences, financial goals, risk tolerance, and current financial situation, ensuring that the advice is relevant and aligned with each user's unique circumstances.

Security: The app should employ strong encryption and secure authentication methods to safeguard user financial data, preventing unauthorized access or data breaches.

Reliability: The FundFlow Navigator app should have a system uptime of at least 99.9%, ensuring that users can access their financial information and recommendations whenever they need.

Scalability: The system should be able to handle a growing user base and increasing data volume without compromising performance, ensuring that response times remain consistent even as the user load increases.

Intuitiveness: The user interface should be intuitive and user-friendly, presenting financial information in a clear and easy-to-understand manner, thus enabling users to make informed decisions effortlessly.

Adaptability: The app should adapt to changing market conditions, updating its recommendations and predictions in response to economic fluctuations or changes in user behavior.

Integration: The system should be able to integrate with a wide range of financial institutions, allowing users to import and consolidate financial data from various sources such as banks, investment platforms, and credit card companies.

Mobile Compatibility: The mobile application should function seamlessly across different mobile devices and operating systems, ensuring a consistent experience for users regardless of their chosen device.

Data Privacy: The application should adhere to strict data privacy regulations, giving users full control over their data and providing options for data anonymization or deletion.

Educational Support: The app should offer educational resources and explanations for financial concepts, helping users better understand the implications of different financial decisions and enabling them to improve their financial literacy.

Feedback Mechanism: The app should provide users with clear feedback on the impact of following its recommendations, allowing users to evaluate the effectiveness of the advice given.

Proactive Alerts: The application should notify users of important financial events, such as upcoming bill payments, investment maturity dates, or significant market fluctuations, helping users stay on top of their financial responsibilities.

Customizability: Users should have the option to customize the types of alerts, insights, and notifications they receive, tailoring the app's features to their specific preferences and needs.

Customer Support: The app should offer responsive and knowledgeable customer support, available through multiple channels, to assist users with any issues, questions, or concerns they might have.

Continuous Improvement: The development team should regularly update the app to enhance its features, improve its AI algorithms, and incorporate user feedback to ensure that FundFlow Navigator remains a relevant and valuable tool for users' financial well-being.

These are the system attributes contained by the FundFlow Navigator mobile application.

4.3 System Interface

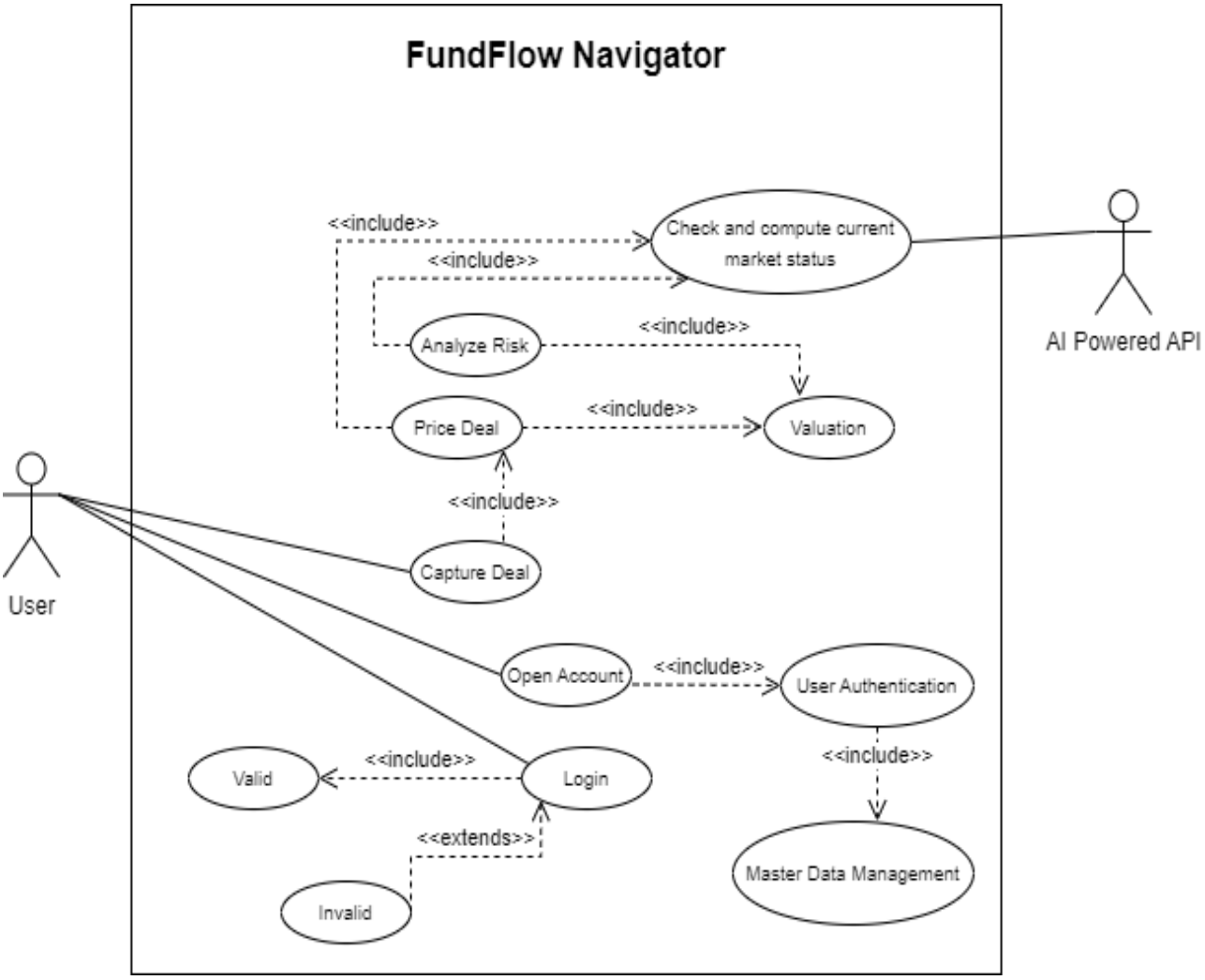


Figure 1: Use case Diagram.

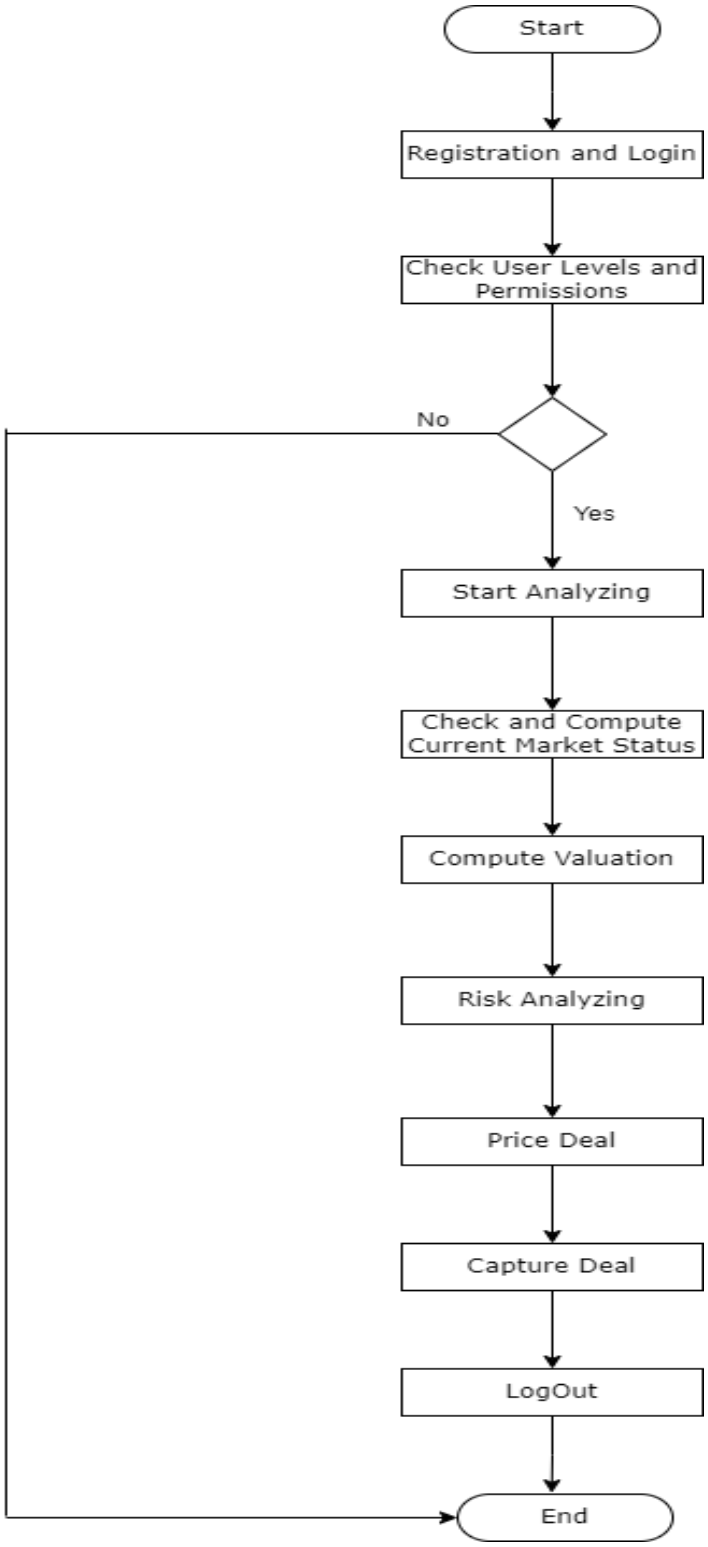


Figure 2: Activity Diagram.

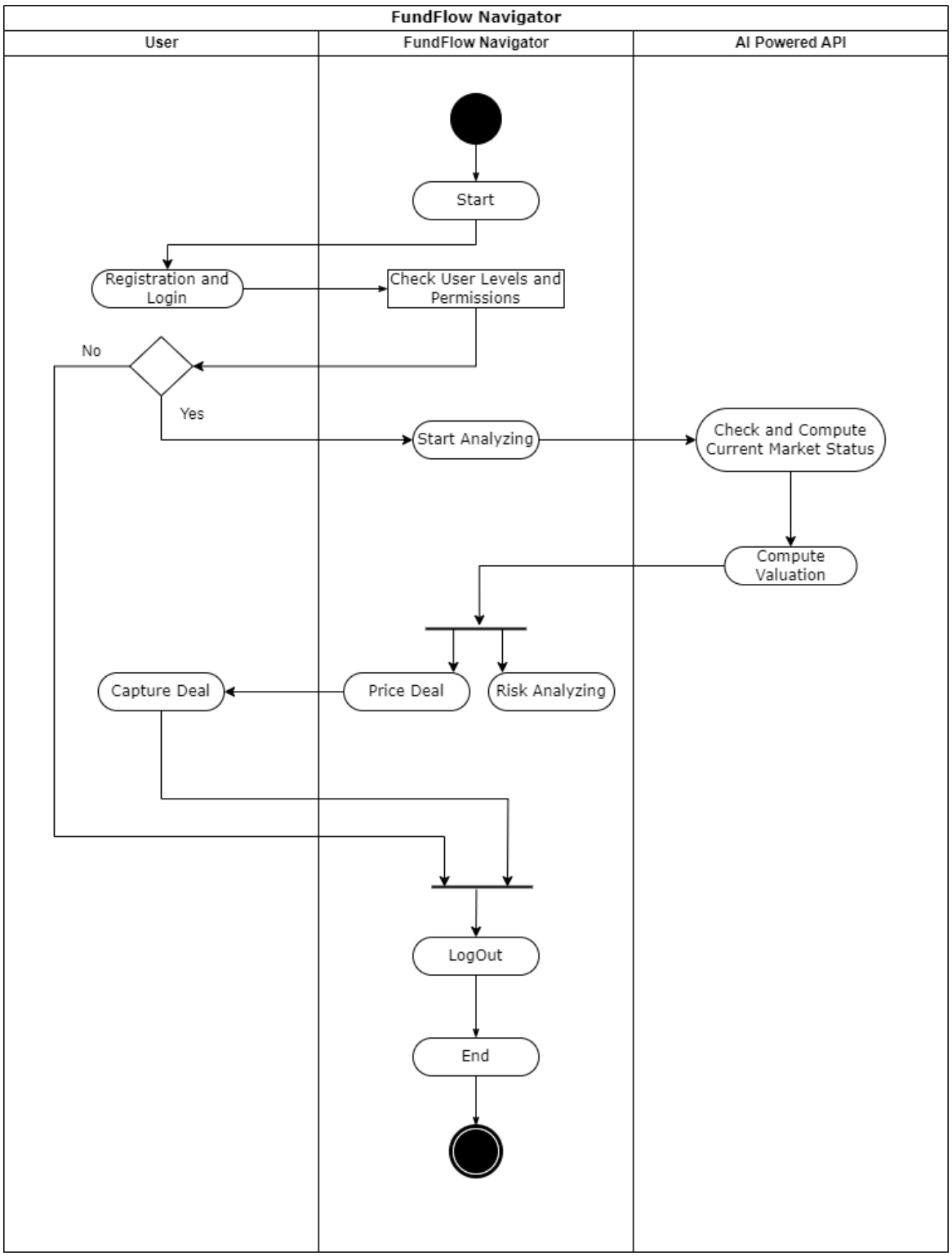


Figure 3: Swimlane Diagram.

4.4 Project Requirements

Here are the project constraints that shall need to be followed in the project management of the "FundFlow Navigator" system:

Total Budget: The project should be executed within a total budget of **$5,720**, which includes all development, testing, deployment, marketing, and operational costs. Budget tracking and management tools should be utilized to ensure adherence to the allocated budget.

Total Development Time: The development of the FundFlow Navigator app should be completed within a timeframe of 6 months from project initiation to product deployment. Clear project milestones and deadlines should be established to track progress and manage the development timeline effectively.

Resource Availability: The project team should manage and allocate human resources efficiently to ensure that all required skills, such as AI development, mobile app development, UI/UX design, and financial domain expertise, are available throughout the project.

Technology Stack: The project should adhere to the predefined technology stack and infrastructure, ensuring compatibility, scalability, and security of the application. Any changes to the technology stack should be evaluated against their impact on the project timeline and budget.

Data Privacy and Compliance: The system should adhere to relevant data privacy regulations and financial industry standards. This includes implementing strong data encryption, secure authentication, and user consent mechanisms to protect sensitive financial data.

User Testing and Feedback: The project plan should include provisions for user testing and feedback collection at various stages of development. Iterative testing and refinement based on user feedback should be integrated into the project timeline.

Integration with External APIs: The application's integration with external financial data sources and market data APIs should be seamless and accurate, ensuring real-time updates. The development team should follow best practices for integrating and maintaining these connections.

Scalability Planning: The architecture and design of the application should take scalability into account, allowing for future growth in terms of user base and data volume. The application should be able to handle increased load without compromising performance.

Quality Assurance and Testing: The project management plan should allocate sufficient time for comprehensive quality assurance and testing, including functional testing, security testing, and performance testing, to ensure a stable and reliable application.

User Training and Onboarding: The project should include a plan for user training and onboarding to ensure that users can effectively navigate and utilize the application's features. User-friendly tutorials and support materials should be provided.

Change Management: Any changes to project scope, requirements, or features should be properly evaluated, documented, and communicated to all relevant stakeholders. A change management process should be in place to manage scope creep and ensure alignment with project goals.

Backup and Disaster Recovery: The project should include strategies for regular data backup and disaster recovery planning to minimize data loss in case of system failures or unforeseen events.

Stakeholder Communication: Regular communication with stakeholders, including investors, management, and potential users, should be maintained throughout the project to provide updates on progress, challenges, and achievements.

Legal and Intellectual Property Considerations: The project management should address legal agreements, intellectual property rights, and any licensing requirements associated with the development of the AI algorithms, software, and user interface components.

User Accessibility: The application should adhere to accessibility guidelines and standards to ensure that users with disabilities can effectively use the app. This includes considerations for screen readers, keyboard navigation, and other assistive technologies.

Deployment Plan: The project should outline a clear deployment plan, including testing in a staging environment before releasing the application to production, to minimize disruptions to users during the rollout.

Continuous Improvement: The project management plan should include provisions for ongoing maintenance, updates, and improvements to the application after its initial release, in order to address user feedback, fix bugs, and adapt to changing market conditions.

Risk Management: The project should identify potential risks and mitigation strategies, such as technology limitations, regulatory changes, market volatility, and unexpected resource shortages, to minimize their impact on the project's success.

User Privacy and Consent: The application should include features that allow users to control their data and give explicit consent for data usage. Privacy policies and terms of use should be transparently communicated to users.

Cross-Platform Compatibility: The app should be developed and optimized to work across multiple mobile platforms (iOS, Android) to maximize its reach and user base.

These project constraints shall guide the project management team in successfully developing and delivering the FundFlow Navigator application within the defined budget, timeline, and quality standards.

5. FEATURES NOT TO BE TESTED

External Financial Data Sources: The accuracy and reliability of external financial data sources, such as market data providers or banking APIs, are not directly under the control of the FundFlow Navigator application. Testing of data accuracy and consistency will rely on the performance of these external sources.

User Behavior and Decision Making: The application cannot directly control or test how users interact with the financial recommendations and insights provided. User behavior, decision-making processes, and actions taken based on the app's suggestions will not be directly tested.

External Security Factors: While the application will implement strong security measures, factors outside its control, such as user device security, network vulnerabilities, and user behaviors, may impact the overall security of the user's financial data.

These areas will be indirectly influenced by the performance of the FundFlow Navigator application and its interactions with users and external services, but testing in these areas will be limited by factors outside the direct control of the application.

6. TESTING APPROACH

6.1 Testing Levels

Scope of Integration Testing

The scope of integration testing for the "FundFlow Navigator" project involves testing the interactions between various components, modules, and systems to ensure seamless functionality, accurate data flow, and proper communication within the application. Given the complexity of the project and the interdependencies among its features, thorough integration testing is crucial to guarantee a reliable and cohesive user experience. Here's the outline of the key areas to focus on:

1. AI-Powered Financial Insights:

- Integration between the AI engine and the user's financial data input.

- Testing the accuracy of AI-generated insights and recommendations based on different user scenarios and market data.

2. Goal-Oriented Planning:

- Integration of the goal-setting module with user profiles and financial data.

- Testing the alignment of goal breakdowns with financial data and milestones.

3. Real-Time Monitoring and Alerts:

- Integration between real-time data feeds, user profiles, and alert/notification systems.

- Testing the accuracy and timeliness of alerts related to financial plan deviations and market shifts.

4. Educational Resources:

- Integration of the educational hub with user profiles, financial data, and recommended strategies.

- Testing the accessibility of educational resources and their relevance to user goals and financial situations.

5. Customized Investment Recommendations:

- Integration between user profiles, risk assessment data, and AI-driven investment recommendations.

- Testing the suitability of investment recommendations based on user preferences, goals, and market trends.

6. Expense Tracking and Budgeting:

- Integration of expense tracking features with user profiles and financial data.

- Testing the accuracy of expense categorization and tracking, along with budget recommendations.

7. User Authentication and Security:

- Integration of user authentication systems with all modules and features.

- Testing the security of user data and access controls across the application.

8. Data Synchronization:

- Testing the synchronization of data across different devices and platforms where the application is accessible.

9. Third-Party Integrations:

- Integration with external APIs for real-time market data.

- Testing the reliability and accuracy of data retrieved from external sources.

10. User Interface and User Experience:

- Testing the overall usability, responsiveness, and consistency of the user interface across different modules.

11. Cross-Module Dependencies:

- Testing interactions and data flow between different modules to identify any data inconsistencies or conflicts.

12. Data Integrity and Storage:

- Testing the accuracy and integrity of data stored in the application's databases.

13. Performance and Load Testing:

- Testing the application's performance under different user loads and traffic conditions.

14. Error Handling and Failover:

- Testing the application's response to unexpected errors or failures, ensuring graceful degradation and user notifications.

15. Compatibility Testing:

- Testing the application on different devices, operating systems, and browsers to ensure consistent functionality.

16. API Testing:

- Testing the communication and data exchange between different APIs used in the application.

18. Integration with Backend Systems:

- Testing the integration between the front-end application and the backend systems responsible for data processing and storage.

By thoroughly testing the integration of these components, modules, and systems, the "FundFlow Navigator" project can ensure that the application functions seamlessly, providing users with accurate insights, personalized recommendations, and a reliable financial management experience.

We have set a structure for the integration levels of the project. This will help the project to be build in an effective and efficient manner.

The Integration Test Phases-

Component Integration:

Modules/Subsystems: Individual components that make up the core features (e.g., expense tracking, goal planning, AI engine, educational hub).

Building Process: Combine individual components to form higher-level subsystems.

Scheduling: Initiate this phase after each component has passed unit testing.

Subsystem Integration:

Modules/Subsystems: Subsystems formed in the previous phase (e.g., goal-oriented planning with expense tracking, real-time monitoring with AI engine).

Building Process: Combine related subsystems to create broader functional groups.

Scheduling: Start after component integration is completed and successful.

Functional Integration:

Modules/Subsystems: Functional groups from the subsystem integration phase (e.g., financial insights with goal planning and real-time monitoring).

Building Process: Combine functional groups to establish end-to-end functionality.

Scheduling: Begin after subsystem integration is validated.

System Integration:

Modules/Subsystems: The entire application, including all functional groups and subsystems.

Building Process: Assemble the complete application, including UI and backend systems.

Scheduling: Commence after functional integration is verified.

Modules or subsystems to be integrated in each phase of the integration test:

Component Integration:

- AI Engine
- Expense Tracking Module
- Goal Planning Module
- Educational Hub Module

Subsystem Integration:

- AI Engine + Real-Time Monitoring Module
- Expense Tracking Module + Goal Planning Module
- Educational Hub Module + Investment Recommendation Module

Functional Integration:

- AI Engine + Real-Time Monitoring Module + Expense Tracking Module + Goal Planning Module
- AI Engine + Real-Time Monitoring Module + Educational Hub Module + Investment Recommendation Module

System Integration:

- Complete Application: All modules and subsystems combined.

The building process and scheduling of each integration test phase has been given next.

Component Integration:

- ❖ Build individual components.
- ❖ Test each component in isolation using unit tests.
- ❖ Schedule based on completion of component development and unit testing.

Subsystem Integration:

- Build subsystems from integrated components.
- Test the interaction and communication between subsystems.
- Schedule after successful completion of component integration and unit testing.

Functional Integration:

- Build functional groups from integrated subsystems.
- Test end-to-end functionality of each functional group.
- Schedule after subsystem integration is validated.

System Integration:

- Assemble the complete application, including UI and backend.
- Conduct comprehensive end-to-end tests for the entire system.
- Schedule after successful functional integration.

Environment to Be Set Up and Resources Required in Each Phase:

Component Integration:

Environment: Development environments for individual component development.

Resources: Developers, version control, development tools.

Subsystem Integration:

Environment: Integration environment with tools for module interaction testing.

Resources: Integration testers, test data, integration testing tools.

Functional Integration:

Environment: Integration environment with tools for functional group interaction testing.

Resources: Integration testers, test data, integration testing tools.

System Integration:

Environment: Complete system integration environment, including UI and backend.

Resources: System testers, test data, testing tools, devices for UI testing.

By following this integration test structure, each phase of integration shall be methodically carried out, tested, and validated, leading to a robust and reliable "FundFlow Navigator" application that meets its objectives and provides users with the intended comprehensive and proactive financial management capabilities.

The entry criteria, exit criteria, integration techniques, and test configuration setups are given below for each integration test phase:

AI-Powered Financial Insights Integration:

Entry Criteria:

- AI engine and user financial data input modules are individually unit tested.
- AI engine generates insights and recommendations based on basic test scenarios.
- User profiles with simulated financial data are available.

Exit Criteria:

- AI engine successfully integrated with financial data input module.
- AI-generated insights and recommendations are consistently accurate for a range of user scenarios.
- Preliminary integration test cases have passed.

Integration Techniques:

- Top-down integration approach: Integrating higher-level modules first.

Test Configuration Set-up:

- Test environment with sample user profiles and financial data.
- Mocked real-time market data feeds for testing AI responses.

Goal-Oriented Planning Integration:

Entry Criteria:

- Goal-setting module is unit tested.
- User profiles and financial data inputs are validated.
- Basic goal-setting interactions work as expected.

Exit Criteria:

- Goal-setting module successfully integrated with user profiles and financial data.
- Goals are aligned with financial data and milestones accurately.
- Goal breakdowns are displayed correctly in user profiles.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with varied user profiles and financial data.
- Mocked data for different goal scenarios.

Real-Time Monitoring and Alerts Integration:

Entry Criteria:

- Real-time data feeds are tested for accuracy.
- User profiles are populated with financial data.
- Alert/notification systems are unit tested.

Exit Criteria:

- Real-time data feeds integrated with user profiles and alert/notification systems.
- Alerts for financial plan deviations and market shifts are accurate and timely.
- Alerts are correctly linked to the relevant user profiles.

Integration Techniques:

- Big Bang integration approach: Integrating all components at once due to interdependencies.

Test Configuration Set-up:

- Test environment with simulated real-time data streams.
- User profiles with various financial situations for alert testing.

Educational Resources Integration:

Entry Criteria:

- Educational hub and user profiles are unit tested.
- Financial data is available for user profiles.
- Recommended strategies and educational content are generated.

Exit Criteria:

- Educational hub successfully integrated with user profiles and financial data.

- Educational resources are accessible within user profiles.
- Recommended strategies are relevant to user goals and financial situations.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with different user profiles and corresponding educational content.
- Mocked data for recommended strategies and educational material.

Customized Investment Recommendations Integration:

Entry Criteria:

- User profiles and risk assessment data are validated.
- AI-driven investment recommendations module is unit tested.
- Recommended investment strategies are generated.

Exit Criteria:

- Investment recommendations module integrated with user profiles and risk assessment data.
- AI-driven investment recommendations aligned with user preferences, goals, and market trends.
- Investment portfolios are diversified and appropriate.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with varied user profiles and risk assessment data.
- Mocked data for investment scenarios and market trends.

Expense Tracking and Budgeting Integration:

Entry Criteria:

- Expense tracking module is unit tested.
- User profiles and financial data inputs are validated.
- Expense categorization and budget recommendations are functioning.

Exit Criteria:

- Expense tracking module successfully integrated with user profiles and financial data.
- Expenses are accurately categorized and tracked within user profiles.
- Budget recommendations reflect user financial situations.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with diverse user profiles and financial data.
- Mocked data for different expense categories and budget scenarios.

User Authentication and Security Integration:

Entry Criteria:

- User authentication systems are unit tested.
- User profiles with financial data are available.

- Data security measures are in place.

Exit Criteria:

- User authentication systems integrated with all modules and features.
- User data is securely accessed based on authentication.
- Access controls are enforced consistently.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with user profiles and various authentication scenarios.
- Simulated data for security testing.

Data Synchronization Integration:

Entry Criteria:

- Data synchronization mechanisms are unit tested.
- User profiles are populated with financial data.
- Multiple devices are simulated for synchronization.

Exit Criteria:

- Data synchronization integrated across different devices and platforms.
- User financial data remains consistent across devices.
- Synchronization is triggered accurately and reliably.

Integration Techniques:

- Big Bang integration approach: Integrating all components at once due to interdependencies.

Test Configuration Set-up:

- Test environment with simulated devices and data synchronization scenarios.
- Mocked data for data synchronization testing.

Third-Party Integrations Integration:

Entry Criteria:

- External APIs for real-time market data are validated.
- User profiles are populated with financial data.
- Data retrieval from external sources is successful.

Exit Criteria:

- Third-party APIs successfully integrated with user profiles and market-related modules.
- Data retrieved from external sources (market data) is reliable and accurate.
- External API responses are seamlessly incorporated into the application.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with user profiles and simulated market data.
- Mocked responses from external APIs for testing.

User Interface and User Experience Integration:

Entry Criteria:

- User interface components are individually validated.
- The core functionality of each module is tested and functioning.

Exit Criteria:

- User interface elements integrated across all modules and features.
- UI elements provide consistent and intuitive user experience.
- The core functionality of each module works seamlessly through the UI.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with diverse user profiles and various usage scenarios.
- Mocked data for UI testing.

Cross-Module Dependencies Integration:

Entry Criteria:

- Individual modules and their core functionality are validated.
- Data interactions between modules are understood and documented.

Exit Criteria:

- Interactions between different modules are successfully integrated.
- Data inconsistencies or conflicts are identified and resolved.
- Cross-module data flows smoothly without errors.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with different user profiles and complex data flow scenarios.
- Mocked data for cross-module interaction testing.

Data Integrity and Storage Integration:

Entry Criteria:

- Database schema and data storage mechanisms are validated.
- User profiles with financial data are available.
- Data integrity constraints are in place.

Exit Criteria:

- Data integrity and storage mechanisms integrated across all modules.
- User data stored accurately and securely in the database.
- Data retrieval and storage are reliable without data corruption.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with diverse user profiles and various data storage scenarios.
- Mocked data for data integrity testing.

Performance and Load Testing Integration:

Entry Criteria:

- The core functionality of each module is validated.
- The test environment is set up with representative user profiles and data.

Exit Criteria:

- Application performance tested under different user loads and traffic conditions.
- Application response times meet acceptable standards.
- Performance degradation or bottlenecks are identified and addressed.

Integration Techniques:

- Top-down integration approach: Integrating higher-level modules first.

Test Configuration Set-up:

- Test environment with varying loads and traffic simulation tools.
- Simulated user profiles and data for load testing.

Error Handling and Failover Integration:

Entry Criteria:

- Individual modules are tested for error handling and failover scenarios.
- User profiles with financial data are available.
- Error handling mechanisms are defined.

Exit Criteria:

- Error handling and failover mechanisms integrated across all modules.
- Application responds gracefully to unexpected errors or failures.
- Users receive appropriate notifications for errors or failures.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with diverse user profiles and simulated error scenarios.
- Mocked data for error handling and failover testing.

Compatibility Testing Integration:

Entry Criteria:

- User interface components are validated individually.
- User profiles with financial data are available.
- Compatibility requirements are defined.

Exit Criteria:

- Application successfully tested on different devices, operating systems, and browsers.
- UI elements render correctly and consistently across various platforms.
- Core functionality works uniformly across different configurations.

Integration Techniques:

- Bottom-up integration approach: Integrating lower-level modules first.

Test Configuration Set-up:

- Test environment with various devices, operating systems, and browsers.
- Simulated data for compatibility testing.

API Testing Integration:

Entry Criteria:

- APIs used for communication between modules are validated.
- User profiles with financial data are available.
- API endpoints and data structures are defined.

Exit Criteria:

- APIs integrated across modules and features, enabling data exchange.
- Communication between modules through APIs is reliable and accurate.
- API responses conform to expected data structures.

Integration Techniques:

- Thread integration approach: Incrementally integrating components while focusing on specific features.

Test Configuration Set-up:

- Test environment with simulated API requests and responses.
- Mocked data for API communication testing.

Integration with Backend Systems Integration:

Entry Criteria:

- Front-end modules are validated and tested.
- Backend systems responsible for data processing and storage are functional.

Exit Criteria:

- Front-end modules successfully integrated with backend systems.
- Data communication between front-end and backend is accurate and reliable.

- Application functions as expected with real data from the backend.

Integration Techniques:

- Big Bang integration approach: Integrating all components at once due to interdependencies.

Test Configuration Set-up:

- Test environment with user profiles and data synchronized between front-end and backend.
- Mocked data for integration with backend systems.

By adhering to these criteria for each integration test phase, the "FundFlow Navigator" application shall thoroughly be tested, and each component, module, and feature shall work harmoniously together to provide users with a comprehensive and proactive financial management solution.

Scope of System Testing

System testing is a crucial phase in the software development life cycle where the entire application is tested as a whole to ensure that all components and functionalities work together as expected. Here's how the scope of system testing shall be for “FundFlow Navigator” project:

End-to-End Workflow Testing:

- Test the entire workflow of the application, from user registration to setting financial goals, receiving insights, making investments, and monitoring progress.
- Ensure that all steps in the user journey are seamless and free of errors.

User Authentication and Security Testing:

- Verify that user authentication and data security measures are effective in preventing unauthorized access and ensuring data protection.
- Test password management, account recovery, and secure storage of sensitive information.

Usability and User Experience Testing:

- Evaluate the overall user interface, navigation, responsiveness, and user-friendliness of the application.
- Ensure that users can easily navigate through different features and functionalities.

Performance Testing:

- Test application performance under different load conditions to ensure responsiveness and optimal resource utilization.
- Measure response times, loading times, and resource consumption.

Compatibility Testing:

- Test the application across various devices (desktop, mobile, tablet) and different web browsers to ensure consistent functionality and appearance.

Error Handling and Recovery Testing:

- Test how the application handles various types of errors, unexpected inputs, and failures.
- Verify that error messages are informative, and users are guided toward appropriate actions.

Data Integrity and Reliability Testing:

- Verify the accuracy, consistency, and integrity of user data stored within the application's databases.
- Test data retrieval, storage, and updates.

Accessibility Testing:

- Test the application's accessibility for users with disabilities, ensuring compliance with accessibility standards.

Security Testing:

- Conduct security testing to identify vulnerabilities, potential breaches, and ensure that sensitive data is properly protected.

Scalability Testing:

- Assess how well the application performs and scales when the user base and data volume increase.

Data Backup and Recovery Testing:

- Test data backup and recovery mechanisms to ensure that user data can be restored in case of data loss.

Cross-Platform Testing:

- Test the application on different operating systems to ensure cross-platform compatibility.

Performance Under Stress Testing:

- Simulate scenarios of high user loads and stress to assess how the application performs under extreme conditions.

Structure of System Testing Levels for FundFlow Navigator:

Here, we have provided a structure of the system testing levels for the "FundFlow Navigator" project:

Phase 1: Unit and Component Testing

Building Process:

- Develop and test individual units and components in isolation.
- Test the functionalities of each module independently to ensure they work as expected.

Scheduling:

- Occurs during the development phase.
- Parallel to the development of individual modules.

Environment and Resources:

- Development environment with access to code repositories, development tools, and testing frameworks.
- Test data sets for unit and component testing.
- Test automation tools for running automated unit tests.

Phase 2: Document Verification and Validation

Building Process:

- Review and validate all project documentation, including requirements, design specifications, test plans, and user documentation.
- Ensure that documentation accurately reflects the project's scope, features, and requirements.

Scheduling:

- Occurs throughout the project lifecycle, especially during the requirement and design phases.
- Regularly validate documentation against the evolving project.

Environment and Resources:

- Documentation review environment with access to all project documents.
- Collaboration tools for communication among stakeholders and review teams.

Phase 3: Compliance and Regulatory Testing

Building Process:

- Verify that the application complies with relevant industry regulations, standards, and legal requirements.
- Test functionalities related to data privacy, security, and financial regulations.

Scheduling:

- Conducted after successful completion of security and functional testing.
- Collaborate with legal and compliance teams to ensure alignment with regulations.

Environment and Resources:

- Regulatory testing environment with tools for compliance testing.
- Regulatory compliance checklists and guidelines.
- Collaboration tools for communication among legal, compliance, and testing teams.

Phase 4: Accessibility and Usability Compliance Testing

Building Process:

- Test the application for compliance with accessibility standards (such as WCAG) and usability guidelines.
- Ensure that the application is accessible to users with disabilities.

Scheduling:

- Conducted after successful completion of user acceptance testing (UAT) and functional testing.
- Collaborate with accessibility experts to identify and fix accessibility issues.

Environment and Resources:

- Accessibility testing environment with tools for assessing accessibility compliance.

- Accessibility guidelines and standards documentation.
- Collaboration tools for communication among accessibility experts and testing teams.

Phase 5: Performance Testing

Building Process:

- Test the application's performance under different loads and conditions.
- Measure response times, loading times, and resource consumption.

Scheduling:

- Parallel to functional testing, focusing on the application's responsiveness and resource usage.

Environment and Resources:

- Performance testing environment with simulated user loads and monitoring tools.
- Test data sets for different load scenarios.
- Performance testing tools for measuring application response times and resource usage.

Phase 6: Security and Vulnerability Testing

Building Process:

- Identify and test potential security vulnerabilities, breaches, and data protection measures.
- Ensure that user data is securely stored and transmitted.

Scheduling:

- Parallel to the later stages of functional and performance testing.
- Conducted alongside other testing phases to ensure security measures are integrated properly.

Environment and Resources:

- Security testing environment with tools for vulnerability assessment and penetration testing.
- Test data sets for security testing scenarios.
- Security testing tools for identifying vulnerabilities and security risks.

Phase 7: Load Testing

Building Process:

- Simulate various user loads to test the application's performance under expected and peak usage scenarios.
- Measure response times, resource utilization, and system behavior under different loads.

Scheduling:

- Conducted after successful completion of functional testing and performance testing.
- Can be incremental, testing increasing loads to identify breaking points.
- Environment and Resources:
- Load testing environment with tools for simulating user loads.
- Load test scenarios and test data sets.
- Load testing tools for measuring application performance under different loads.

Environment and Resources:

- Load testing cloud-based environment with tools and scalable resources for simulating extreme conditions.
- Load test simulation scenarios where there is a sudden surge in user activity, mimicking events like stock market fluctuations or special promotions.

Phase 8: Stress Testing

Building Process:

- Test the application's performance under extreme conditions beyond its expected capacity.

- Identify the application's breaking point and its ability to recover gracefully.

Scheduling:

- Conducted after successful completion of load testing and performance testing.
- Simulated stress conditions to observe the application's behavior.

Environment and Resources:

- Stress testing environment with tools for simulating extreme conditions.
- Stress test scenarios that push the application's limits.
- Stress testing tools for measuring application behavior under extreme loads.

Phase 9: Volume Testing

Building Process:

- Test the application's performance by subjecting it to a large volume of data.
- Assess the application's ability to handle large data sets and maintain its performance.

Scheduling:

- Conducted after successful completion of load testing and stress testing.
- Simulate different scenarios with varying data volumes.

Environment and Resources:

- Volume testing environment with tools for generating large data sets.
- Volume test scenarios and data sets of varying sizes.
- Volume testing tools for measuring application performance with large data volumes.

The structure of system testing levels is designed to progressively ensure the quality and functionality of the "FundFlow Navigator" application. Each phase builds upon the previous one,

ultimately leading to a thoroughly tested and reliable application that empowers users in their financial management journey.

Here are the criteria for each of the system test phases:

End-to-End Workflow Testing:

Entry Criteria:

- Completion of integration testing to ensure individual modules work together.
- Availability of test environments and test data.
- Approval of test scenarios and test cases.

Exit Criteria:

- Successful execution of end-to-end test scenarios.
- No critical defects impacting the workflow.
- All user journeys tested and documented.

System Testing Techniques:

- Scenario-based testing to cover user journeys.
- Positive and negative testing to verify expected and unexpected scenarios.

Test Configuration Set-Up:

- Configured test environment that replicates production settings.
- Test data reflecting real user scenarios.

User Authentication and Security Testing:

Entry Criteria:

- Availability of security requirements and authentication specifications.
- Completion of unit and integration testing for authentication components.

Exit Criteria:

- No security vulnerabilities identified.
- Successful completion of security-related test cases.
- User data remains secure throughout testing.

System Testing Techniques:

- Security testing to identify vulnerabilities.
- Penetration testing to simulate attacks and verify defenses.

Test Configuration Set-Up:

- Configured security testing environment.
- Tools for identifying vulnerabilities and attacks.

Usability and User Experience Testing:

Entry Criteria:

- Availability of user interface (UI) designs and wireframes.
- Completion of UI development.

Exit Criteria:

- Positive user feedback on usability.
- No major usability issues reported.
- UI meets design specifications.

System Testing Techniques:

- Usability testing with real users.
- User interface testing to verify UI elements.

Test Configuration Set-Up:

- Various devices and browsers for cross-platform testing.
- Tools for capturing user feedback.

Performance Testing:

Entry Criteria:

- Completion of functional testing.
- Defined performance requirements and metrics.

Exit Criteria:

- Performance metrics meet predefined thresholds.
- Application responds within acceptable time frames.
- No major performance bottlenecks identified.

System Testing Techniques:

- Load testing to simulate user loads.
- Stress testing to assess breaking points.

Test Configuration Set-Up:

- Configured performance testing environment.
- Load testing tools for simulating user loads.

Compatibility Testing:

Entry Criteria:

- Completion of functional testing.
- Availability of a list of compatible devices and browsers.

Exit Criteria:

- Application functions consistently across all listed devices and browsers.
- No major compatibility issues identified.

System Testing Techniques:

- Cross-browser testing.
- Device-specific testing.

Test Configuration Set-Up:

- Different devices and browsers for testing.
- Tools for checking cross-browser compatibility.

Error Handling and Recovery Testing:

Entry Criteria:

- Completion of functional testing.
- Defined error scenarios and recovery procedures.

Exit Criteria:

- Application handles errors gracefully and guides users toward recovery.
- No critical errors leading to application crashes.

- System Testing Techniques:
- Negative testing to simulate error scenarios.
- User journey testing with deliberate errors.

Test Configuration Set-Up:

- Test environment with tools to simulate errors.

Data Integrity and Reliability Testing:

Entry Criteria:

- Completion of integration testing.
- Availability of test data reflecting real user scenarios.

Exit Criteria:

- User data remains consistent and accurate throughout testing.
- No data corruption or loss during testing.

System Testing Techniques:

- Data validation testing.
- Data retrieval and storage testing.

Test Configuration Set-Up:

- Test environment with configured databases.

Accessibility Testing:

Entry Criteria:

- Availability of accessibility guidelines and standards.
- Completion of user interface (UI) development.

Exit Criteria:

- Application complies with accessibility standards.
- Usable by users with disabilities.

System Testing Techniques:

- Accessibility testing with assistive technologies.
- Manual testing by users with disabilities.

Test Configuration Set-Up:

- Assistive technologies for testing.

Security Testing:

Entry Criteria:

- Completion of functional testing.
- Availability of security requirements and specifications.

Exit Criteria:

- No security vulnerabilities identified.
- Application data remains secure during testing.

System Testing Techniques:

- Penetration testing.
- Vulnerability scanning.

Test Configuration Set-Up:

- Security testing environment.
- Tools for identifying vulnerabilities.

Scalability Testing:

Entry Criteria:

- Completion of performance testing.
- Defined scalability metrics and thresholds.

Exit Criteria:

- Application scales well with increasing user loads.
- No significant performance degradation with increased users.

System Testing Techniques:

- Load testing with incremental user loads.
- Monitoring resource utilization.

Test Configuration Set-Up:

- Load testing environment.
- Tools for measuring resource utilization.

Data Backup and Recovery Testing:

Entry Criteria:

- Completion of integration testing.
- Availability of backup and recovery mechanisms.

Exit Criteria:

- Successful data recovery from backups.
- User data can be restored in case of data loss.

System Testing Techniques:

- Data recovery testing with different scenarios.
- Backup and recovery procedure validation.

Test Configuration Set-Up:

- Test environment with backup and recovery tools.

Cross-Platform Testing:

Entry Criteria:

- Completion of functional testing.
- Availability of different operating systems and platforms.

Exit Criteria:

- Application functions consistently across different operating systems.
- No major platform-specific issues identified.

System Testing Techniques:

- Testing on different operating systems and platforms.

Test Configuration Set-Up:

- Different operating systems for testing.

Performance Under Stress Testing:

Entry Criteria:

- Completion of load and stress testing.
- Availability of high-stress scenarios.

Exit Criteria:

- Application remains functional under extreme conditions.
- No critical failures or data loss.

System Testing Techniques:

- High-stress load testing.
- Measuring application behavior under extreme conditions.

Test Configuration Set-Up:

- High-stress testing environment.
- Tools for simulating extreme conditions.

The criteria for each system test phase ensure that the "FundFlow Navigator" application is thoroughly tested from various aspects, including functionality, usability, security, performance, and reliability.

6.2 Test Tools

Test Automation Tools:

Selenium: Selenium will be used to automate the testing of user interfaces and user interactions within the FundFlow Navigator application. It will simulate user actions, interactions, and verifications, ensuring consistent behavior across different scenarios.

Load Testing Tools:

JMeter: JMeter will be utilized to conduct load testing on the application. It will simulate a large number of concurrent users to assess the application's performance, responsiveness, and resource utilization under different load conditions.

Security Testing Tools:

OWASP Zap: OWASP Zap will be employed for security testing. It will scan the application for potential vulnerabilities, such as injection attacks, cross-site scripting, and other security flaws, ensuring that the application is secure.

Performance Monitoring Tools:

New Relic: New Relic will be integrated to monitor the application's performance in real-time. It will provide insights into response times, resource consumption, and other performance metrics, allowing quick identification and resolution of performance issues.

Test Management Tools:

TestRail: TestRail will be used to manage test cases, test plans, and test execution. It will help organize test scenarios, track test progress, and generate comprehensive test reports.

Issue Tracking Tools:

JIRA: JIRA will be employed to track and manage issues identified during testing. It will facilitate collaboration among team members, assigning and monitoring issue resolution tasks.

Continuous Integration Tools:

Jenkins: Jenkins will be used for continuous integration and automated builds. It will ensure that code changes are integrated smoothly, tested, and deployed to the testing environment.

Code Review and Collaboration Tools:

GitHub/GitLab/Bitbucket: These version control platforms will be used for collaborative code development and code review. They will facilitate version control, code collaboration, and peer review processes.

Network Virtualization Tools:

Wireshark: Wireshark will be utilized for network virtualization and analysis. It will help simulate and analyze different network conditions to assess application performance and behavior.

Database Testing Tools:

DBUnit: DBUnit will be used to test database interactions. It will facilitate the setup of test data, perform database comparisons, and validate data integrity.

These tools shall play essential roles in performing integration and system tests for the FundFlow Navigator application. These tools shall enhance efficiency, accuracy, and coverage of testing activities, ensuring the application's reliability, performance, and security.

6.3 Meetings

Here's a schedule of meetings to complete the entire FundFlow Navigator project:

Project Kick-off Meeting:

Attendees: Project Manager, Development Team, Testing Team, UI/UX Designers.

Purpose: Introduce the project, discuss objectives, roles, responsibilities, and overall project plan.

Requirements Review Meeting:

Attendees: Project Manager, Business Analysts, Development Team, Testing Team.

Purpose: Review and clarify project requirements, ensuring a shared understanding among all stakeholders.

Design Review Meeting:

Attendees: UI/UX Designers, Development Team, Testing Team.

Purpose: Review and discuss the application's design, user interface, and user experience.

Test Planning Meeting:

Attendees: Testing Team, Project Manager.

Purpose: Define the test strategy, scope, objectives, and allocate resources for testing activities.

Development Progress Review Meeting:

Attendees: Project Manager, Development Team, Testing Team.

Purpose: Review the progress of development work, address any challenges, and ensure alignment with project timelines.

Test Case Review Meeting:

Attendees: Testing Team, Project Manager, Development Team.

Purpose: Review and finalize the test cases, ensuring coverage of all requirements and scenarios.

Integration Testing Kick-off Meeting:

Attendees: Testing Team, Development Team, Project Manager.

Purpose: Discuss the scope and objectives of the integration testing phase, allocate test environments, and resources.

System Testing Kick-off Meeting:

Attendees: Testing Team, Development Team, Project Manager.

Purpose: Discuss the scope and objectives of the system testing phase, allocate test environments, and resources.

Progress Update Meeting (Weekly):

Attendees: Project Manager, Development Team, Testing Team.

Purpose: Review the overall project progress, discuss challenges, and assess if milestones are being met.

Defect Triage Meeting (Bi-weekly):

Attendees: Testing Team, Development Team, Project Manager.

Purpose: Review reported defects, prioritize, assign, and plan for resolution.

UAT Kick-off Meeting:

Attendees: Testing Team, Business Stakeholders, Project Manager.

Purpose: Introduce the User Acceptance Testing (UAT) phase, discuss expectations, and finalize test criteria.

UAT Progress Review Meeting (Weekly):

Attendees: Project Manager, Business Stakeholders, Testing Team.

Purpose: Review UAT progress, address any issues, and ensure alignment with project goals.

Project Status Review Meeting:

Attendees: Project Manager, Development Team, Testing Team, Stakeholders.

Purpose: Review overall project status, deliverables, and discuss any potential adjustments to the project plan.

Release Readiness Meeting:

Attendees: Project Manager, Development Team, Testing Team.

Purpose: Review the application's readiness for release, including testing outcomes and resolution of issues.

Project Closure Meeting:

Attendees: Project Manager, Development Team, Testing Team, Stakeholders.

Purpose: Review the project's success, lessons learned, and discuss future enhancements.

Lessons Learned Meeting:

Attendees: Project Manager, Development Team, Testing Team.

Purpose: Reflect on the project, identify areas of improvement, and capture lessons learned for future projects.

Additional meetings shall be scheduled as needed to address specific challenges, emergencies, or updates throughout the FundFlow Navigator project.

7. TEST CASES/TEST ITEMS

Unit Testing:

Project Name: FundFlow Navigator			Test Designed by: Md. Mintajur Rahman Emon	
Test Case ID: FFN_log_1			Test Designed date: 06/06/2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: Md. Mintajur Rahman Emon	
Module Name: Login Session			Test Execution date: 06/06/2023	
Test Title: Verify login with valid username and password				
Description: Test website login page				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Application 2. Enter username 3. Enter password 4. Click submit	Username: Erling3 Password: $#300erling	User should login into the application	As expected	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

As part of the testing procedure, we have focused only on integration testing and system testing. Therefore, the “Test Plan” suggests that unit testing is not been rigidly visualized in this test plan. The “Integration Testing” and “System Testing” are given below respectively.

Integration Testing:

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-AI-01	Test Designed date: 10/06/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Tanvir Hossen
Module Name: AI Insights Integration	Test Execution date: 11/06/2023
Test Title: AI-Powered Financial Insights Integration	
Description: Test AI engine's generation of accurate insights and recommendations for user John Doe based on financial data and market trends	
Precondition (If any): The AI engine is integrated with the user's financial data input. User data and market data are available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Launch the FundFlow Navigator application. 2. Log in as John Doe. 3. Navigate to the "AI Insights" section. 4. Verify user's financial data display. 5. Observe AI-generated insights and recommendations. 6. Compare AI recommendations with user's situation and market trends. 7. Execute different test scenarios (e.g., modify monthly income, add expenses). 8. Verify AI's insights adapt to changes. 9. Capture screenshots of AI-generated insights. 10. Analyze insights for accuracy and alignment.	Username: John Doe Password: [John123] Age: 30 years Monthly Income: $5000 Expenses: $1200 Interest in business: Stock market	Application should launch successfully. Application should successfully log in as John Doe. The application should take User to the "AI Insights" section. The application should display User's financial data accurately. The AI should generate insights and recommendations. The AI's recommendations should be aligned with user's financial	As expected	Pass

	Interest level: High	situation and market trends. The AI's insights should adapt to the modified data. The AI's insights should update according to modified data. The AI-generated insights should be accurate and aligned with user's data and market trends.		
Post Condition: App is operational, user session is maintained, accurate AI insights shown. Modified financial data stored, screenshots documented. Issues noted for resolution. AI-engine integrated, app stable for further testing or use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Goal-01	Test Designed date: 12/06/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Tanvir Hossen
Module Name: Goal-Oriented Planning	Test Execution date: 13/06/2023
Test Title: Goal-Oriented Planning Integration	
Description: Test integration of goal-setting module with user profiles and financial data.	
Precondition (If any): Goal-setting module integrated with user profiles and financial data. User data and goal data available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Jane Smith in the application. 2. Access the goal-setting section. 3. Input the car savings goal and required timeframe. 4. Verify goal breakdown aligns with Jane's financial data. 5. Adjust financial data, verify goal breakdown updates accordingly. 6. Capture screenshots of goal breakdown and progress. 7. Analyze goal breakdown for accuracy and financial feasibility.	Username: Jane Smith Password: Jane123 Age: 28 years Goal: Savings for a car Timeframe: 10 months	The User should successfully log in as Jane Smith. The user should be navigated to the goal-setting section. The user should give input of Goal and timeframe successfully. The Goal breakdown should match Jane's financial data and milestones. The Goal breakdown should update it. The Goal breakdown should be accurate and financially feasible.	As expected	Pass

Post Condition: App remains operational with user logged in; goal-setting module integrated. User data intact, goal breakdown aligned with finances. Adjustments and screenshots saved; any issues documented. App stable, ready for use and future testing.

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Educational-01	Test Designed date: 13/06/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Tanvir Hossen
Module Name: Educational Resources	Test Execution date: 20/06/2023
Test Title: Educational Resources Integration	
Description: Test integration of educational resources within the application to ensure accessibility and relevance to users' varying financial knowledge levels and goals.	
Precondition (If any): Educational resources integrated within the application. User profiles with varying financial knowledge available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users with varying financial knowledge. 2. Access the educational resources section of the application. 3. Verify that educational content is categorized and easily navigable. 4. Evaluate the relevance of content to users' financial goals and situations. 5. Interact with educational content (read articles, watch videos). 6. Capture screenshots of educational content and user interactions.	User: David Snackle Financial knowledge: Accounting User: Steve Herby Financial knowledge: Economics	The users should successfully log in as different users. The users should be navigated to the educational resources section. The content should be relevant to users' financial goals and situations. The users should be able to interact with educational content successfully. The Educational resources should be accessible and relevant.	As expected	Pass

7. Analyze the accessibility and relevance of educational resources.				
Post Condition: App operational, user profiles exist, educational resources integrated. Interactions and issues documented; resources categorized. App ready for further testing, usage, catering to diverse financial knowledge.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Investment-01	Test Designed date: 21/06/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Tanvir Hossen
Module Name: Investment Recommendations	Test Execution date: 28/06/2023
Test Title: Customized Investment Recommendations Integration	
Description: Test integration of investment recommendations module within the application to ensure personalized recommendations matching user's risk tolerance, financial goals, and market conditions.	
Precondition (If any): Investment recommendations module integrated with user profiles and risk assessment data. User data and market data available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Robert Williams. 2. Access the "Investment Recommendations" section. 3. Verify user's risk tolerance and financial goals are accurately displayed. 4. Observe AI-generated investment recommendations 5. Compare recommendations with Robert's preferences, goals, and market trends. 6. Modify user's risk tolerance and goals, verify changes in recommendations.	Username: Robert Williams Password: Robert123 Age: 40 years Risk tolerance: Moderate Financial Goals: Generate $1000/Month	The user should successfully log in as Robert Williams. The user should be navigated to the "Investment Recommendations" section. The user's risk tolerance and goals should be displayed accurately. The AI-generated investment recommendations should be displayed. Recommendations should align with Robert's preferences, goals, and market trends.	As expected	Pass

7. Capture screenshots of investment recommendations. 8. Analyze recommendations for suitability and alignment.	Market Data: Forex trading	Recommendations should be updated based on modified data. Recommendations should be suitable and aligned with user's data and market trends.		
Post Condition: App operational, user's profile maintained, investment module integrated. User interactions documented, app ready for use and further testing. Investment recommendations align with user's preferences and market trends.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Alert-01	Test Designed date: 29/06/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Tanvir Hossen
Module Name: Real-Time Monitoring and Alerts	Test Execution date: 04/07/2023
Test Title: Real-Time Monitoring and Alerts Integration	
Description: Test integration of real-time data feeds with user profiles and alert systems to ensure timely alerts for financial plan deviations and market shifts.	

Precondition (If any): Real-time data feeds integrated with user profiles and alert systems. User data and market data available.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Mary Johnson. 2. Navigate to the "Alerts" section. 3. Verify user's financial data is correctly displayed. 4. Simulate a deviation from the financial plan, verify alert is generated. 5. Introduce changes in the market data, verify corresponding market shift alert. 6. Capture screenshots of alerts and notification triggers. 7. Analyze alerts for accuracy, timing, and	Username: Mary Johnson Password: Mary 123 Age: 30 years Income: $7000/Month Expense: $1500/Month Real-time Data: stock market simulated data	The user should successfully log in as Mary Johnson. The user should be navigated to the "Alerts" section. The user's financial data should be displayed accurately. The alert should be generated for the financial plan deviation. The alert should be generated for the corresponding market shift.	As expected	Pass

alignment with user's situation.		The alerts should be accurate, timely functioned, and aligned with user's data.		
Post Condition: App operational, user's profile maintained, real-time alerts integrated. User interactions documented, app ready for further use and testing. Timely alerts generated for financial plan deviations and market shifts, enhancing monitoring experience.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Expense-01	Test Designed date: 05/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Tanvir Hossen
Module Name: Expense Tracking and Budgeting	Test Execution date: 11/07/2023
Test Title: Expense Tracking and Budgeting Integration	
Description: Test integration of expense tracking module within the application to ensure accurate categorization, tracking, and alignment of user's expenses and budget recommendations.	
Precondition (If any): Expense tracking module integrated with user profiles and financial data. User data and budget data available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as Emily Brown. 2. Navigate to the "Expense Tracking" section. 3. Verify user's expenses and budget are displayed accurately. 4. Record various expenses, ensure accurate categorization and tracking. 5. Compare actual expenses to budget recommendations. 6. Modify user's budget, verify changes affect recommendations 7. Capture screenshots of expense tracking and budget recommendations.	Username: Emily Brown Password: Emily123 Age: 28 years Expense: $2000/Month Budget for Investment: $20,000	The user should successfully log in as Emily Brown. The user should be navigated to the "Expense Tracking" section. The user's expenses and budget should be displayed accurately. The expenses should be accurately categorized and tracked. The actual expenses should match budget recommendations. The budget recommendations	As expected	Pass

8. Analyze accuracy of tracking and alignment with financial data.		should update based on modified data. The expense tracking and budgeting should be accurate and aligned.		
Post Condition: App operational, user's profile maintained, expense module integrated. User interactions documented, app ready for use and testing. Expense tracking and budget recommendations align with user's financial data, facilitating effective budget management.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Security-01	Test Designed date: 12/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Tanvir Hossen
Module Name: User Authentication and Security	Test Execution date: 14/07/2023
Test Title: User Authentication and Security Integration	
Description: Test integration of user authentication systems within the application to ensure secure and protected user data access.	
Precondition (If any): User authentication systems integrated with all modules and features. User profiles and security measures available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Attempt to log in with valid credentials. 2. Attempt to log in with incorrect username and/or password. 3. Verify account lockout after multiple failed login attempts 4. Log in with correct credentials, ensure successful access. 5. Verify access to various modules based on user's role. 6. Attempt unauthorized access to other user's profiles. 7. Capture screenshots of successful and failed login attempts.	Username: Michele Password: Michele123	The user should successfully log in with valid credentials. The login attempt should fail due to incorrect credentials. The Account should be locked after multiple consecutive failed login attempts. The user should successfully log in with correct credentials. Access should be granted to modules based on the user's role. Unauthorized access to other	As expected	Pass

8. Analyze access control and security measures.		user's profiles should be denied.		
Post Condition: App operational, security systems intact. User profiles and data remain secure. Interactions documented, app ready for further use and testing. Unauthorized access issues documented for resolution, ensuring user data protection.				

Project Name: FundFlow Navigator	Test Designed by: Md. Tanvir Hossen
Test Case ID: IT-Sync-01	Test Designed date: 14/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Tanvir Hossen
Module Name: Data Synchronization	Test Execution date: 15/07/2023
Test Title: Data Synchronization Integration	
Description: Test integration of data synchronization mechanisms across different platforms and devices to ensure accurate and consistent user data across all platforms.	
Precondition (If any): Data synchronization mechanisms integrated across platforms and devices. User data synchronized and consistent across all platforms.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users on different devices. 2. Update user data on one platform, verify changes reflect on other platforms. 3. Simulate scenarios where data synchronization may be disrupted. 4. Capture screenshots of synchronized data across platforms. 5. Analyze data consistency and synchronization accuracy.	User Profile 1: Businessman Financial Data: Net Profit is $500,000 User Profile 2: Student Financial Data: Monthly Income is $3,000	The user should successfully logg in as different users on different devices. The changes made on one platform should be reflected accurately on other platforms. The data synchronization should be continued smoothly despite simulated disruptions. The data should remain consistent and accurately synchronized across platforms.	As expected	Pass

	User Profile 3: Government Employee Financial Data: Monthly Income is $10,000			
Post Condition: App operational. User profiles on devices maintained. Data sync across platforms active. User actions saved. Discrepancies documented for resolution. Data remains accurate and consistent. App stable for testing or use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-ThirdParty-01	Test Designed date: 15/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Third-Party Integrations	Test Execution date: 18/07/2023
Test Title: Third-Party Integrations	
Description: Test integration of external APIs within the application to ensure accurate retrieval and display of real-time market data from external sources.	
Precondition (If any): External APIs integrated to provide real-time market data. Data retrieved from external sources and integrated into the application.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Access sections of the application that display real-time market data. 2. Verify that the data retrieved matches the external source's data. 3. Simulate scenarios where external source data changes rapidly. 4. Capture screenshots of real-time market data in the application. 5. Analyze the accuracy and timeliness of the displayed data.	Real-time Market Data: Customized financial data chart inputted (simulated)	The user should be navigated to sections displaying real-time market data. The application should display accurate real-time market data matching the external source's data. The application should update displayed data rapidly to match changes in the external source.	As expected	Pass

		The displaying visualization should be accurate and updates in a timely manner.		
Post Condition: App remains operational. External APIs for real-time market data stay integrated. App continues to display accurate real-time market data. User interactions and retrieval instances are stored. Discrepancies are documented for resolution. App is stable for testing or regular use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-UIUX-01	Test Designed date: 18/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: User Interface and User Experience	Test Execution date: 19/07/2023
Test Title: User Interface and User Experience Integration	
Description:Test integration of user interface components across different platforms and devices to ensure consistency, responsiveness, and user-friendliness.	
Precondition (If any): User interface components integrated across platforms and devices. User profiles with different devices available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users on different devices. 2. Navigate through various sections of the application. 3. Verify that UI components adapt to different screen sizes. 4. Check responsiveness of UI elements. 5. Capture screenshots of UI on different devices. 6. Analyze user experience and interface consistency.	User profiles with different devices: User 1- Danilo Brown Device: I phone 14 Pro Max User 2- Anderson Ryllee Device: Samsung Glaxy S21 Ultra	The users should successfully log in as different users on different devices. The users should be successfully navigated through different sections of the application. The UI components should adapt to screen sizes of different devices. The UI elements (buttons, forms, graphs) should be	As expected	Pass

	User 3- Allen Bannet Device: Xiaomi Redmi Note 11	responsive and functional. The User experience should be smooth, and interface remain consistent across devices.		
Post Condition: App remains operational. User profiles on different devices persist. UI components stay integrated, ensuring consistent user experience. User interactions and UI views are saved. Discrepancies are documented for resolution. UI remains consistent, responsive, and user-friendly across devices. App is stable for further testing, engagement, or use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-Dependencies-01	Test Designed date: 19/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Cross-Module Dependencies	Test Execution date: 21/06/2023
Test Title: Cross-Module Dependencies Integration	
Description: Test integration of cross-module dependencies to ensure data consistency and proper interactions between interdependent modules.	

Precondition (If any): Cross-module dependencies integrated across different features. User profiles with varying financial data available.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users with distinct financial data. 2. Access AI insights, investment recommendations, and expense tracking. 3. Verify that data shared between modules remains consistent. 4. Perform actions in one module and observe impacts in other modules. 5. Capture screenshots of interactions between modules. 6. Analyze data consistency and cross-module behavior.	Username: George Furry Password: George123 Age: 45 years Monthly Income: $15000 Expenses: $4000 Interest in business: Stock market, real state	The user should successfully log in as different users. The accessed AI should give insights, investment recommendations, and expense calculation. The data shared between modules should remain consistent and accurately reflect user's financial data.	As expected	Pass

		The actions in one module should affect data and recommendations in other modules accurately.		
Post Condition: App operational & available. User profiles across modules retained. Cross-module dependencies integrated, ensuring consistent data interactions. User interactions & screenshots documented. Discrepancies noted & reported. App stable for testing or regular use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-DataIntegrity-01	Test Designed date: 21/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Data Integrity and Storage	Test Execution date: 23/07/2023
Test Title: Data Integrity and Storage Integration	
Description: Test integration of data storage mechanisms across modules and features to ensure accurate, consistent, and secure storage of user data in the application's databases.	
Precondition (If any): Data storage mechanisms integrated across modules and features. User profiles with varying financial data available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users with varying financial data. 2. Perform actions across modules that involve data storage. 3. Verify that data is accurately stored in the application's databases. 4. Modify user data, check if changes are reflected in relevant modules. 5. Capture screenshots of data storage mechanisms. 6. Analyze data accuracy, consistency, and security measures.	Username: Herby Jefferson Password: Herby156 Age: 39 years Monthly Income: $7500 Expenses: $2000 Interest in business: Beauty Shop, Food cart business	The user should successfully log in. Actions involving data storage should be completed successfully. The data should be accurately stored in the application's databases. The changes should be accurately reflected in relevant modules after data modification.	As expected	Pass

	Cloud Server Response: Save user data	The Cloud Server should respond and successfully load and save the input data.		
Post Condition: User profiles intact. Data storage maintains accuracy, consistency, and security. User interactions and screenshots recorded. Inconsistencies or vulnerabilities documented. App stable for future tests, usage, and maintenance.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-Performance-01	Test Designed date: 23/06/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Performance	Test Execution date: 24/07/2023
Test Title: Performance Testing Integration	
Description: Test integration of performance and load testing mechanisms to assess the application's optimal performance under varying user load conditions.	
Precondition (If any): Performance and load testing mechanisms integrated to assess system performance. Application running under different user load conditions.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Simulate low, moderate, and high user loads on the application. 2. Measure application response times, loading times, and resource utilization. 3. Verify that the application remains responsive and functional. 4. Capture performance metrics and analyze results. 5. Assess if the application meets performance expectations.	User 1: Hery Kates Password: Herby#5283 User 2: Robert Williams Password: bbb3441 User 3: Petric Muller Password: FootballCR7 User 4: Shaksi Sinha	The simulated different user conditions on the application should be performed successfully. The measured application response times, loading times, and resource utilization for each condition should be accurate. The application should remain responsive and functional under different user conditions.	As expected	Pass

	Password: decode37999@# User 5: Adward Wellington Password: 12345a User 6: Himu Islam Password: HimuIslam9	The performance metrics should be captured, and results should be analyzed for each condition.		
Post Condition: App remains operational and accessible. Performance testing stays integrated for ongoing evaluation. App consistently performs well under varied user loads. Captured performance metrics stored for future analysis. Any identified issues documented for resolution. App is stable, ready for testing or regular use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-ErrorHandling-01	Test Designed date: 24/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Error Handling and Failover	Test Execution date: 26/07/2023
Test Title: Error Handling and Failover Integration	
Description: Test integration of error handling and failover mechanisms to ensure the application gracefully handles errors and recovers from failures, minimizing disruption for users.	
Precondition (If any): Error handling and failover mechanisms integrated to manage unexpected situations. Application running with simulated user actions that may cause errors.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Perform actions that may trigger errors (invalid inputs, network issues). 2. Observe how the application responds to these scenarios. 3. Verify that errors are handled gracefully with informative error messages. 4. Simulate system failures and verify if failover mechanisms kick in. 5. Capture screenshots of error messages and failover processes. 6. Analyze how the application handles errors and recovers from failures.	Error Data: Email address- w 5@gmail.com Name- ju*lly Phone Number- +01773548756 Net Income- $0000	Execute actions that should trigger errors. The application should respond to different error scenarios. The application should display informative error messages and guide users on how to proceed. The failover mechanisms should initiated and handle	As expected	Pass

		system failures effectively. The application should handle errors gracefully and recovers from failures as expected.		
Post Condition: App operational, error handling integrated, gracefully manages errors, documented issues resolved. Failover mechanisms operational for seamless user experience. Screenshots and findings stored. App stable for further testing or user interaction.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-Compatibility-01	Test Designed date: 26/072023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Compatibility Testing	Test Execution date: 27/07/2023
Test Title: Compatibility Testing Integration	
Description: Test integration of compatibility testing to ensure the application functions consistently and correctly across different devices and web browsers.	
Precondition (If any): Application accessible through different devices and browsers. User profiles with different devices and browsers available.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in as different users on different devices and browsers. 2. Access different modules and features of the application. 3. Verify that UI components and functionalities work as intended. 4. Perform actions across different browsers, monitor compatibility. 5. Capture screenshots of application on various devices and browsers. 6. Analyze consistency of UI and functionality.	User profiles with different devices and browsers: User 1- Danilo Brown Device: I phone 14 Pro Max Browser: Chrome User 2- Anderson Ryllee Device: Samsung Glaxy S21 Ultra Browser: Firefox	The users should successfully log in as different users on various devices and browsers. The user should accessed various modules and features on different devices and browsers. The UI components and functionalities should work consistently across different devices and browsers.	As expected	Pass

	User 3- Allen Bannet Device: Xiaomi Redmi Note 11 Browser: Safari	The application should function as intended across different browsers. The UI and functionality should remain consistent across devices and browsers.		
Post Condition: Various user profiles on different devices and browsers exist. Compatibility testing mechanisms persist, ensuring consistent functionality. User interactions and compatibility data, along with screenshots, are stored. Any identified UI or functionality issues are documented for resolution. App is stable for further testing, user engagement, or regular use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-API-01	Test Designed date: 27/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: API	Test Execution date: 28/07/2023
Test Title: API Testing Integration	
Description: Test integration of APIs used for real-time market data and third-party services to ensure correct communication, accurate data exchange, and timely responses.	
Precondition (If any): APIs integrated with the application for data exchange. Data retrieved from external sources through APIs.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Access sections of the application that use external APIs. 2. Perform actions that trigger API calls and data retrieval. 3. Verify that data retrieved matches the expected data from external sources. 4. Simulate scenarios where API data changes rapidly. 5. Capture screenshots of API data displayed in the application. 6. Analyze the accuracy and timeliness of the data retrieved from APIs.	Simulated changes in API data: Customized user finance data chart	The user should successfully access sections using external APIs. The actions should trigger API calls and data retrieval. The data should be retrieved through APIs to match the expected data. The API data should reflect changes as expected.	As expected,	Pass

		The data should be retrieved accurately and in a timely manner from APIs.		
Post Condition: Integrated APIs are functional. Consistent, accurate API communication. User interactions and screenshots saved. Discrepancies documented for resolution. App is stable for testing or regular use.				

Project Name: FundFlow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: IT-Backend-01	Test Designed date: 28/08/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Backend Codes	Test Execution date: 29/07/2023
Test Title: Integration with Backend Systems	
Description: Test integration between the front-end application and backend systems for data processing, storage, and retrieval.	

Precondition (If any): Front-end application integrated with backend systems for data handling. Backend systems functioning and capable of handling user data.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Perform user actions across different modules that involve data processing. 2. Verify that data is correctly sent to the backend for processing and storage. 3. Perform actions that retrieve data from the backend, validate accuracy. 4. Simulate scenarios where backend systems experience high loads. 5. Capture screenshots of user actions and backend interactions. 6. Analyze data processing, storage, and retrieval accuracy.	User actions in Front End: Login- Username is Manny Morella Password is manny1111 Alert system-enable and disable Click on “ON” button to enable Click on “OFF” button to enable	The user actions involving data processing should be executed. The data should be sent to the backend accurately and successfully stored. The data should be retrieved from the backend to match the expected values. The backend systems should be simulated to experience high loads.	As expected	Pass

	Data Processing- Net Worth is $800,000 Monthly Income is $4000 Age is 36 years 2 properties including house and land Interested sector for investment is real-state, stock market, it service			
Post Condition: Front-end and back-end integration persists seamlessly. Backend systems process, store, and retrieve data efficiently. Screenshots of user interactions with backend are saved. Documented and resolved issues ensure accurate data handling. App is stable, ready for more testing or regular use.				

System Testing:

Project Name: Fund Flow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: E2EWT-001	Test Designed date: 29/07/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Mintajur Rahman Emon
Module Name: End-to-End User Workflow	Test Execution date: 30/07/2023
Test Title: Test Specification for End-to-End Workflow Testing (1)	
Description: Verifying the successful completion of tasks related to user onboarding, financial goal management, investment execution, and progress tracking.	
Precondition (If any): User must have valid username and password	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Log in to the application using valid credentials. 2. Complete user registration by providing necessary details. 3. Set financial goals and milestones for the user account. 4. Verify that insights are generated based on financial goals and market trends. 5. Make test investments according to the chosen preferences. 6. Navigate to the progress monitoring section. 7. Ensure that all steps are completed without encountering errors.	Registration Details: Username-Mona99 Password-$99mona Financial Goals: Generate $4000/Month Investment Choices with Preferences: Real State(1), Stock Market(2), Food Business(3), Shipment Business(4)	All steps in the user journey from registration to monitoring progress should be completed without errors. Financial goals should be set up correctly, and insights should be generated. Investments should be made successfully. User should view and monitor progress	As expected	Pass

8. Check that financial goals are correctly displayed, and insights are accurate. 9. Verify that the investments are reflected in the user's portfolio. 10. Ensure that progress monitoring provides up-to-date information				
Post Condition: 1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful. 2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.				

Project Name: Fund Flow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: UAST-001	Test Designed date: 30/07/2023
Test Priority (Low, Medium, High): High	Test Executed by: Md. Mintajur Rahman Emon
Module Name: User Authentication and Security	Test Execution date: 31/07/2023
Test Title: Test Specification for User Authentication and Security Testing (2)	
Description: The User Authentication and Security module encompasses all aspects related to user identity verification, access control, password management, and safeguarding sensitive user information	
Precondition (If any): User must have valid username and password.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Attempt to log in with valid credentials. 2. Attempt to log in with invalid credentials. 3. Initiate the password recovery process using recovery email or security questions. 4. Attempt to reset the password using valid recovery options. 5. Attempt to reset the password using invalid recovery options.	Username: kylirodes Password: k8790$ Recovery Email: kyli1999@gmail.com Security Questions: Birth date, Favorite animal, Grandmother name	The user authentication and security measures should effectively prevent unauthorized access. Password management and account recovery procedures should work as intended. Sensitive information should be securely stored.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Md. Mintajur Rahman Emon
Test Case ID: UUET-001	Test Designed date: 01/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Md. Mintajur Rahman Emon
Module Name: Usability and User Experience	Test Execution date: 01/08/2023
Test Title: Test Specification for Usability and User Experience Testing (3)	
Description: The Usability and User Experience module focuses on evaluating how users interact with the application's interface and overall experience	
Precondition (If any): User must have valid username and password	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate through the user registration process. 2. Set up financial goals and milestones. 3. Explore insights and investment options. 4. Monitor progress and investment performance. 5. Verify that navigation is consistent and intuitive. 6. Ensure that UI elements are responsive on different devices. 7. Test the user interface against predefined design specifications. 8. Complete tasks using different user profiles and preferences	User Preference: Admin of the Application (1) Development and Maintenance Team (2) User (3) Designs and Wireframes: Visual go through	Users should find the application interface intuitive and easy to use. Navigation should be smooth and responsive. Users should complete tasks without confusion or errors.	As expected	Pass

Post Condition:
1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful. 2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Sadman Sanid Tanim
Test Case ID: PT-001	Test Designed date: 01/08/2023
Test Priority (Low, Medium, High): High	Test Executed by: Sadman Sanid Tanim
Module Name: Performance Testing	Test Execution date: 02/08/2023
Test Title: Test Specification for Performance Testing. (4)	
Description: The Performance Testing module focuses on evaluating how the application performs under different levels of user loads	

Precondition (If any): Verify login with valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Conduct performance testing with low user loads. 2. Conduct performance testing with moderate user loads. 3. Conduct performance testing with high user loads. 4. Measure response times and loading times for different scenarios. 5. Monitor resource utilization (CPU, memory) during testing.	Username: june Password: j9897 Login: 100 times Submit Button/ Alert Button Press: 100 times Navigation bar Press: 100 times	Application should respond within acceptable time frames for different user loads. Loading times and response times should meet predefined thresholds. Resource consumption should remain within acceptable limits.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Sadman Sanid Tanim
Test Case ID: CT-001	Test Designed date: 02/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadman Sanid Tanim
Module Name: Compatibility Testing	Test Execution date: 03/08/2023
Test Title: Test Specification for Compatibility Testing (5)	
Description: The Compatibility Testing module focuses on evaluating how the application functions across different devices and browsers	
Precondition (If any): Verify login with valid username and password	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Test the application on each device listed in the compatibility list. 2. Test the application on each browser listed in the compatibility list. 3. Complete tasks using different user profiles on different devices and browsers. 4. Verify that UI elements and functionalities are consistent.	User profiles with different devices and browsers: User 1- Danilo Brown Device: I phone 14 Pro Max Browser: Chrome User 2- Anderson Ryllee Device: Samsung Glaxy S21 Ultra Browser: Firefox	Application should function consistently across all listed devices and browsers. No major compatibility issues should be expected. .	As expected	Pass

	User 3- Allen Bannet Device: Xiaomi Redmi Note 11 Browser: Safari			

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Sadman Sanid Tanim
Test Case ID: EHRT-001	Test Designed date: 03/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadman Sanid Tanim
Module Name: Error Handling and Recovery	Test Execution date: 03/08/2023
Test Title: Test Specification for Error Handling and Recovery Testing. (6)	
Description: The Error Handling and Recovery module focuses on evaluating how the application manages and recovers from various error scenarios	

Precondition (If any): Verify login with valid username and password

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Input invalid data during user registration (e.g., invalid email format. 2. Attempt to submit incomplete financial goals. 3. Provide incorrect login credentials and verify error responses. 4. Test error handling during investment processes. 5. Trigger various errors intentionally and verify application response. 6. Verify that error messages provide clear instructions for recovery.	Valid Input: Username- David Password: 7d698 Invalid Input: Username- Da~vid Password: 7d6_98	Application should show error message due to invalid input Application should show successfully login message for valid input Application should suggest for accurate input	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Sadman Sanid Tanim
Test Case ID: DIRT-001	Test Designed date: 03/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadman Sanid Tanim
Module Name: Data Integrity and Reliability	Test Execution date: 03/08/ 2023
Test Title: Test Specification for Data Integrity and Reliability Testing. (7)	
Description: The Data Integrity and Reliability module focuses on evaluating the accuracy, consistency, and reliability of user data, financial goals, investments, and transactions within the application.	

Precondition (If any): User must have valid username and password

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Create user profiles with different financial goals. 2. Add test investments and transactions for user accounts. 3. Modify user data and verify changes are reflected correctly. 4. Test data retrieval for generating insights and progress monitoring. 5. Validate that user data integrity is maintained during updates.	User Data: Username- tanim Password- t8796 Financial Goals: $3000/Month Investment: $10000	The input data should remain accurate, consistent, and reliable through interaction with the application. Data retrieving, storing and updating should function as the threshold.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Sadman Sanid Tanim
Test Case ID: AT-001	Test Designed date: 03/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Sadman Sanid Tanim
Module Name: Accessibility Testing	Test Execution date: 04/08/2023
Test Title: Test Specification for Accessibility Testing (8)	
Description: The Accessibility Testing module focuses on evaluating the application's accessibility to users with disabilities, ensuring that it adheres to established accessibility guidelines and standards, such as WCAG.	

Precondition (If any): Verify login with valid username and password

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Test the application with screen readers to verify compatibility. 2. Navigate the application using keyboard-only inputs. 3. Test the application's compatibility with assistive technologies. 4. Verify text alternatives for non-text content and proper heading structure. 5. Ensure that color contrast meets accessibility standards	Username: hervy Password: 987hervy Accessibility Guidelines Matching: WCAG	Application should be accessible to users with disabilities.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Luisa Khaleque Trisha
Test Case ID: ST-001	Test Designed date: 04/08/2023
Test Priority (Low, Medium, High): High	Test Executed by: Luisa Khaleque Trisha
Module Name: Security Testing	Test Execution date: 05/08/2023
Test Title: Test Specification for Security Testing (9)	
Description: Application is secure from potential vulnerabilities, breaches, and attacks sensitive user data is protected.	

Precondition (If any): User is registered and logged into the application.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Conduct penetration testing to identify potential vulnerabilities. 2. Test for common security vulnerabilities (e.g., SQL injection, cross-site scripting). 3. Verify that user data is encrypted during transmission. 4. Test user authentication against various attack scenarios. 5. Verify that sensitive data is properly stored and protected	Ways used: Ethical hacking and penetration scanning	Application should be secure from potential vulnerabilities, breaches, and attacks. Sensitive user data should not be accessed.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Luisa Khaleque Trisha
Test Case ID: SCLT-001	Test Designed date: 05/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Luisa Khaleque Trisha
Module Name: Scalability Testing	Test Execution date: 06/08/2023
Test Title: Test Specification for Scalability Testing (10)	
Description: Different user load scenarios will be created to test performance metrics for scalability.	
Precondition (If any): Application is ready for scalability testing.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1.Test application performance with low user loads. 2.Test application performance with moderate loads. 3.Test application performance with high loads. 4.Measure response times, loading times, and resource utilization.	User 1: Hery Kates Password: Herby#5283 User 2: Robert Williams Password: bbb3441 User 3: Petric Muller Password: FootballCR7 User 4: Shaksi Sinha Password: decode37999@#	The simulated different user conditions on the application should be performed successfully. The measured application response times, loading times, and resource utilization for each condition should be accurate. The application should remain responsive and functional under different user conditions.	As expected	Pass

	User 5: Adward Wellington Password: 12345a User 6: Himu Islam Password: HimuIslam9	The performance metrics should be captured, and results should be analyzed for each condition.		

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Luisa Khaleque Trisha
Test Case ID: DBRT-001	Test Designed date: 06/08/2023
Test Priority (Low, Medium, High): High	Test Executed by: Luisa Khaleque Trisha
Module Name: Data backup and Recovery Testing	Test Execution date: 06/08/2023
Test Title: Test Specification for Data Backup and Recovery Testing. (11)	
Description: Backup and recovery procedures and simulated data loss scenarios are created.	
Precondition (If any): Backup and recovery mechanisms are in place	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Simulate data loss scenarios. 2. Initiate data recovery procedures 3. Verify user data is restored successfully	Data Loss Scenarios: Delete the user credentials Simulated virus attack (eg. Manually remove and contaminate data)	Data recovery procedures should be executed. User data should be restored to previous state. Backup cloud server should store the previous data before virus attack.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Luisa Khaleque Trisha
Test Case ID: CPT-001	Test Designed date: 07/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Luisa Khaleque Trisha
Module Name: Cross Platform Testing	Test Execution date: 07/08/2023
Test Title: Test Specification for Cross-Platform Testing. (12)	
Description: List of supported operating systems and list of supported browsers is used.	
Precondition (If any): Application is ready for cross-platform testing.	

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Test application on each listed operating system 2. Test application on each listed browser 3. Complete tasks with different user profiles on various platforms 4. Verify consistent UI and functionalities	Testing on Operating System: Android OS, Apple iOS, BlackBerry OS, Palm OS, Bada, Windows Mobile OS, Symbian OS, Web OS	Consistent functionality tasks should be completed without issues. Consistency should be maintained across different android operating systems.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.

2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Project Name: Fund Flow Navigator	Test Designed by: Luisa Khaleque Trisha
Test Case ID: PUST-001	Test Designed date: 07/08/2023
Test Priority (Low, Medium, High): Medium	Test Executed by: Luisa Khaleque Trisha
Module Name: Performance Under Stress Testing	Test Execution date:08/08/2023
Test Title: Test Specification for Performance Under Stress Testing. (13)	
Description: Stress testing scenarios are created with high user loads to defined performance metrics and thresholds.	

Precondition (If any): Application is ready for stress testing.

Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Simulate scenarios of extremely high user loads. 2. Measure response times, loading times, and resource utilization. 3. Verify application maintains acceptable performance	User login at a time using automation test: 500 users/second	Application should maintain acceptable response times and performance during a lot of logins. Resource consumption should remain within acceptable limits.	As expected	Pass

Post Condition:

1. If the application maintains acceptable performance, responsiveness, and resource utilization under extreme user loads, the test is successful.
2. Any performance degradation, resource overload, or unacceptably slow response times should be documented as defects.

Acceptance Testing:

Project Name: FundFlow Navigator			Test Designed by: Md. Mintajur Rahman Emon	
Test Case ID: FFN_BAT_1			Test Designed date: 24/08/2023	
Test Priority (Low, Medium, High): Medium			Test Executed by: Md. Mintajur Rahman Emon	
Module Name: Login Session			Test Execution date: 25/08/2023	
Test Title: Verify login with valid username and password				
Description: Test Application login panel				
Precondition (If any): User must have valid username and password				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to the Application 2. Enter username 3. Enter password 4. Click submit	User login Numbers: 1000 Login Credentials: Random individual selection credentials	Users should login into the application. Users should accept this login module as interactive and easy to use.	As expected	Pass
Post Condition: User is validated with database and successfully login to account. The account session details are logged in the database.				

As part of the testing procedure, we have focused only on integration testing and system testing. Therefore, the “Test Plan” suggests that acceptance testing is not been rigidly visualized in this test plan.

8. ITEM PASS/FAIL CRITERIA

Here are the pass/fail criteria for the test phases in the FundFlow Navigator project:

Unit Testing:

Pass Criteria: Each unit test case must pass without any critical defects.

Fail Criteria: If critical defects are identified in the unit testing phase.

Component Integration Testing:

Pass Criteria: All components must integrate successfully without critical defects.

Fail Criteria: If critical defects prevent the proper integration of components.

System Integration Testing:

Pass Criteria: All integrated components and modules must work cohesively, and core functionalities should be operational.

Fail Criteria: If core functionalities fail to operate as expected or significant integration issues arise.

End-to-End Workflow Testing:

Pass Criteria: The end-to-end workflow must be completed successfully without any show-stopping defects.

Fail Criteria: If show-stopping defects prevent the completion of the end-to-end workflow.

User Authentication and Security Testing:

Pass Criteria: User authentication and data security measures should prevent unauthorized access and protect sensitive data.

Fail Criteria: If there are vulnerabilities or unauthorized access is possible.

Usability and User Experience Testing:

Pass Criteria: The application's user interface and navigation should be intuitive and user-friendly.

Fail Criteria: If usability issues make navigation difficult or confusing for users.

Performance Testing:

Pass Criteria: The application should meet predefined performance metrics under varying load conditions.

Fail Criteria: If the application's performance degrades significantly or breaches defined thresholds.

Compatibility Testing:

Pass Criteria: The application should function consistently across various devices and browsers.

Fail Criteria: If the application exhibits functionality or appearance discrepancies on different platforms.

Error Handling and Recovery Testing:

Pass Criteria: The application should handle errors gracefully, guiding users toward appropriate actions.

Fail Criteria: If error messages are unclear or the application fails to recover from errors.

Data Integrity and Reliability Testing:

Pass Criteria: User data should be accurately stored, retrieved, and updated within the application's databases.

Fail Criteria: If data inconsistencies or inaccuracies are identified.

Accessibility Testing:

Pass Criteria: The application should adhere to accessibility standards, ensuring usability for users with disabilities.

Fail Criteria: If accessibility barriers are identified that prevent users with disabilities from using the application effectively.

Security Testing:

Pass Criteria: The application should be free from security vulnerabilities and breaches.

Fail Criteria: If security vulnerabilities are identified that could compromise user data.

Scalability Testing:

Pass Criteria: The application's performance and functionality should scale proportionally with increased user base and data volume.

Fail Criteria: If the application's performance degrades significantly under higher loads.

Data Backup and Recovery Testing:

Pass Criteria: Data backup and recovery mechanisms should successfully restore user data in case of data loss.

Fail Criteria: If data restoration fails or is incomplete during recovery testing.

Cross-Platform Testing:

Pass Criteria: The application should work seamlessly on different operating systems.

Fail Criteria: If the application exhibits functionality discrepancies on different operating systems.

Performance Under Stress Testing:

Pass Criteria: The application should exhibit stable behavior and performance under extreme load conditions.

Fail Criteria: If the application crashes, hangs, or exhibits severe performance degradation under stress.

Meeting these pass criteria will signify the successful completion of each test phase, while failing to meet them will result in the identification of defects and issues that need to be addressed before moving forward.

9. TEST DELIVERABLES

These are the test deliverables for the FundFlow Navigator project:

- Integration and System Test Plan
- Unit Test Plans
- Acceptance Test Plan
- Test Cases and Test Scripts
- Test Data and Test Datasets
- Test Environments Setup and Configuration
- Test Execution Logs and Reports
- Defect/Incident Summaries
- Test Completion Reports
- UAT Execution Logs and Reports
- Test Summary and Defect Metrics
- Performance Test Execution Logs
- Security Test Execution Logs
- Accessibility Test Execution Logs
- Compatibility Test Execution Logs
- Test Sign-Off and Approval Document
- Training Materials for Testing Team
- Documentation on Testing Tools and Processes
- Testing Schedule
- Risk Analysis and Contingencies
- Responsibility Chart

These deliverables will ensure a comprehensive testing process and provide documentation of the testing activities, results, and outcomes for the FundFlow Navigator project.

10. STAFFING AND TRAINING NEEDS

For the FundFlow Navigator Project, staffing and training needs are essential to ensure a smooth testing process and successful implementation. The following staffing and training requirements are necessary to perform:

Staffing Needs:

Test Manager: Responsible for overall test planning, coordination, and management of testing activities. Ensures that testing aligns with project goals and timelines.

Testers: At least one full-time tester should be assigned for system/integration testing and user acceptance testing (UAT) phases. This person will work closely with developers, business analysts, and end-users to identify, document, and verify defects and requirements.

Test Automation Engineer (Optional): If the project involves test automation, a dedicated test automation engineer should be assigned to develop and maintain automated test scripts and frameworks.

Training Needs:

EDI Interface Training:

Developers and testers involved in the project need to be trained on the basic operations of the EDI (Electronic Data Interchange) interface. This includes understanding the data exchange process, message formats, data validation, and error handling.

Operations Staff Training:

The operations staff responsible for managing EDI communications should receive comprehensive training on the EDI communication process. This includes setting up connections, monitoring data transfers, resolving communication issues, and handling exceptions.

Sales Administration Staff Training:

The sales administration staff should undergo training on the new screens, reports, and functionalities introduced by the FundFlow Navigator application. This training will ensure they

can effectively use the application to manage financial data, monitor progress, and make informed decisions.

Testing Process Training:

All team members involved in testing, including testers, developers, and project managers, should receive training on the testing processes, methodologies, and tools being used in the project. This will ensure a common understanding of testing procedures and expectations.

User Training (UAT Phase):

During the User Acceptance Testing (UAT) phase, end-users and stakeholders should be trained on how to use the application, provide feedback, and verify its functionality against their requirements.

Test Automation Training (If Applicable):

If test automation is being implemented, the designated test automation engineer should receive training on the selected automation tools, frameworks, scripting languages, and best practices.

Security and Privacy Training:

All team members handling sensitive financial and personal data should undergo security and privacy training to ensure compliance with data protection regulations and secure handling of user information.

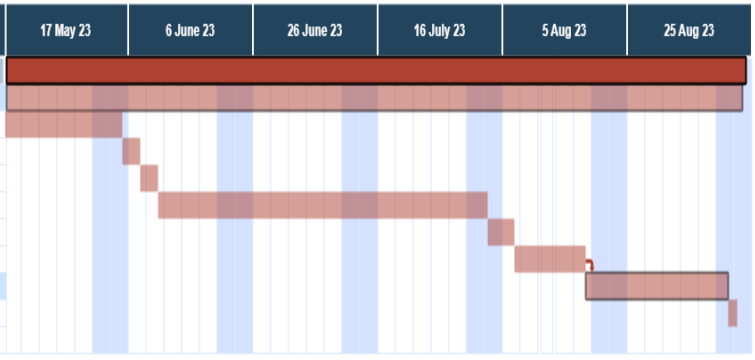
By addressing these staffing and training needs, the FundFlow Navigator Project will ensure that the testing team is well-equipped to perform thorough testing and that all stakeholders have the necessary skills to utilize the application effectively.

11. RESPONSIBILITIES

Project Responsibilities	TM	PM	DEV Team	Test Team	Client
Acceptance test Documentation & Execution	X	X		X	X
System/Integration test Documentation & Execution	X		X	X	
Unit test Documentation & Execution	X		X	X	
System Design Reviews	X	X	X	X	X
Detail Design Reviews	X	X	X	X	
Test procedures and rules	X	X	X	X	
Screen & Report prototype reviews			X	X	X
Change Control and Regression testing	X	X	X	X	X

12. TESTING SCHEDULE

	Task Name	Duration	Start	ETA
1	Complete project execution	100 days	17.05.23	25.08.23
2	Engineering	100 days	17.05.23	24.08.23
3	Requirement Specification & Analysis	10 days	17.05.23	26.05.23
4	Design	7 days	27.05.23	02.06.23
5	Test Plan	7 days	03.06.23	09.06.23
6	Integration Testing	50 days	10.06.23	29.07.23
7	System Testing	10 days	30.07.23	08.08.23
8	Documentation	7 days	09.08.23	15.08.23
9	Project Completion	7 days	16.08.23	22.08.23
10	Feedback	2 days	23.08.23	24.08.23
11				



13. PLANNING RISKS AND CONTINGENCIES

Here, we have provided the Risks and Contingencies plan for the FundFlow Navigator Mobile App:

1. Limited Availability of Financial Experts:

There is a potential risk related to the availability of financial experts who are essential for validating and providing insights into the AI-generated recommendations. In the event of key financial experts being unavailable due to unforeseen circumstances, there might be delays in generating accurate and meaningful insights. To address this, the project team will establish a contingency plan by cross-training additional team members or collaborating with external consultants to ensure timely reviews and validations.

2. Technical Dependencies on Third-Party APIs:

The FundFlow Navigator relies on real-time market data from external APIs for accurate investment recommendations. However, there's a risk of interruptions or changes in the third-party APIs, which could impact the quality and reliability of the data. To mitigate this risk, the project team will monitor the API providers' status and maintain a backup source for market data. Additionally, a contingency plan will be in place to switch to an alternative API provider if necessary.

3. User Engagement and Adoption:

Ensuring user engagement and adoption of the FundFlow Navigator application is crucial for its success. If users do not actively engage with the app or find value in its features, it could impact the project's overall objectives. To address this, the project team will develop a comprehensive user engagement strategy that includes regular communication, user feedback channels, and continuous improvement based on user input. In the event of lower than expected user engagement, the team will implement targeted marketing efforts to increase awareness and usage.

4. Data Privacy and Security Concerns:

Given the sensitive financial and personal data involved, there's a risk of data privacy breaches and security vulnerabilities. To mitigate this risk, the project team will adhere to best practices in data encryption, secure authentication methods, and compliance with relevant data protection

regulations. In the event of a security breach, a well-defined incident response plan will be activated to swiftly address the issue, notify affected users, and implement corrective measures.

By identifying these potential risks and implementing corresponding contingency plans, the FundFlow Navigator project aims to ardently address challenges and ensure a successful and resilient launch of the mobile app.

14. APROVALS

Roles	Approvement Status
Project Sponsor – Md. Mintajur Rahman Emon	Approved
Development Management – Sadman Sanid Tanim	Approved
Business Analyst - Md. Tanvir Hossen	Approved
EDI Project Manager – Sadman Sanid Tanim	Approved
RS Test Manager – Md. Mintajur Rahman Emon	Approved
RS Development Team Manager – Md. Mintajur Rahman Emon	Approved
SQA Team Lead – Md. Tanvir Hossen	Approved
AI EDI Team Manager – Luisa Khaleque Trisha	Approved
Data Management Lead – Luisa Khaleque Trisha	Approved

15. ACKNOWLEDGEMENT

We would like to acknowledge and give our warmest thanks to our course teacher Md. Moman Ul Haque Khan for his guidance throughout our course and for the project. His guidance helped us in making our “Test Plan” accurate.

We would also like to acknowledge the website that we have referenced which helped us gather knowledge based on current world finance. This, we came up with the solution of this AI-Powered application named “Fund Flow Navigator”.