



American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Expense Tracker

A Software Quality and Testing Project Submitted
By

Semester: Fall_23_24			Section: C	Group No: 04
SN	Student Name	Student ID	Individual Contribution (in %)	Total Marks: 50
				Earned Marks:
07	MD SAEEDULLAH AZIM	19-41351-3	25%	
12	MD NAFIJUL HOQ	20-42366-1	25%	
26	TASLIMA AKTHER TULI	20-43174-1	25%	
33	KONAK MOZUMDER	20-43749-2	25%	

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	Total Marks (50)	
Revision History, Test Plan Identifier, Reference Materials, Problem Background, Solutions	[10 Marks]	
Requirements Specification (System feature, Quality Attributes, System Interface, Project Requirements)	[10 Marks]	
Item Not to be tested, Testing approach (Testing levels, tools, meetings), Test cases	[10 Marks]	
Item pass/fail criteria, Test deliverables, Staffing and Training, Responsibilities, Scheduling, Risk	[10 Marks]	
Approval, Format, Submission, and Defense	[10 Marks]	

Software Test Plan

for

<Expense Tracker>

Version 1.0 approved

Prepared by < MD SAEEDULLAH AZIM, MD NAFIJUL HOQ, TASLIMA
AKTHER TULI, KONAK MOZUMDER >

< American International University-Bangladesh (AIUB)>

<24 November, 2023>

Table of Contents

Revision History	Error! Bookmark not defined.
1. TEST PLAN IDENTIFIER: EXPENSE TRACKER-AT-TP01.3	4
2. REFERENCE MATERIALS.....	4
3. INTRODUCTION.....	4
3.1 Background to the Problem	4
3.2 Solution to the Problem	4
4. REQUEIREMNT SPECIFICATION	5
4.1 System Features.....	5
4.2 System Quality Attributes	8
4.3 System Interface	10
4.4 Project Requirements.....	14
5. FEATURES NOT TO BE TESTED.....	15
6. TESTING APPROACH	16
6.1 Testing Level	16
6.2 Testing Tools	17
6.3 Meetings	17
7. TEST CASES/TEST ITEMS	17
8. ITEM PASS/FAIL CRITERIA	34
9. TEST DELIVERABLES.....	35
10. STAFFING AND TRAINING NEEDS.....	35
10.1 Staffing	35
10.2 Training	36
11. RESPONSIBILITIES	38
12. TESTING SCHEDULE.....	38
13. PLANNING RISKS AND CONTINGENCIES	39
14. APROVALS.....	40

1. TEST PLAN IDENTIFIER: **EXPENSE TRACKER-AT-TP01.3**

2. REFERENCE MATERIALS

- [1] [https://ictd.portal.gov.bd/sites/default/files/files/ictd.portal.gov.bd/legislative_information/232669ed_e18c_4557_af65_acabff1e1e0e/Government of Bangladesh ISM compressed \(1\).pdf](https://ictd.portal.gov.bd/sites/default/files/files/ictd.portal.gov.bd/legislative_information/232669ed_e18c_4557_af65_acabff1e1e0e/Government%20of%20Bangladesh%20ISM%20compressed%20(1).pdf)
- [2] https://ijirt.org/master/publishedpaper/IJIRT150860_PAPER.pdf
- [3] <https://www.expensify.com/>
- [4] <https://mint.intuit.com/>

3. INTRODUCTION

3.1 Background to the Problem

In the contemporary landscape, the need for effective expense management is evident across personal and business spheres. Manual methods of tracking expenses are prone to errors, delays, and a lack of real-time insights, posing challenges in financial accuracy and transparency. The fundamental issue lies in the complexities of manual expense management, leading to potential discrepancies, compliance challenges, and an overall lack of visibility into spending patterns. Addressing this predicament is vital for streamlined financial processes, accurate reporting, and informed decision-making, underscoring the significance of the proposed expense tracker.

3.2 Solution to the Problem

To tackle the shortcomings of manual expense tracking, the envisaged solution is sophisticated expense tracker software. This solution utilizes automation and user-friendly interfaces to revolutionize the recording, categorization, and analysis of expenses. By providing real-time insights into financial activities, the software aims to offer a seamless and error-free method of managing expenses. The appropriateness of this solution is evident in its ability to eliminate manual errors, ensure timely reporting, and enhance overall financial visibility.

The feasibility of this solution aligns with the business objective of optimizing expense management processes. Through the automation of tracking and categorization, the software aims to save time, reduce errors, and elevate financial decision-making. The proposed solution is versatile, catering to the needs of both individual users and businesses, offering scalability and adaptability.

Software Description:

The expense tracker software is an intuitive application designed for simplified and enhanced expense management. It automates the recording and categorization of expenses, providing users

with real-time insights. The software's primary objective is to streamline the expense tracking process, minimizing errors, and empowering users to make informed financial decisions. Key benefits include enhanced accuracy, time efficiency, and comprehensive reporting.

Existing Studies in the Problem Area:

Current studies in expense management emphasize the transition to digital solutions to overcome manual tracking limitations. Various software solutions, from basic mobile apps to complex business-oriented platforms exist. However, many lack the comprehensive automation and user-friendly interfaces necessary for seamless expense tracking. The proposed expense tracker project aims to address this gap by delivering a robust, intuitive solution for individual and business users, ensuring a more accurate and efficient expense tracking experience.

4. REQUIREMENT SPECIFICATION

4.1 System Features

4.1.1 Sign Up

Functional Requirements:

4.1.1.1 The software shall provide a user registration form with fields for username, email and password.

4.1.1.2 Upon successful registration, the system shall send a confirmation email to the user.

Priority Level: High

Precondition: None

Cross Reference: N/A

4.1.2 Sign Out

Functional Requirements:

4.1.2.1 The system shall include a "Sign Out" button on the user interface.

4.1.2.2 Upon signing out, the user shall be redirected to the login page.

Priority Level: Medium

Precondition: User is logged in

Cross Reference: N/A

4.1.3 Dashboard

Functional Requirements:

4.1.3.1 The dashboard shall display an overview of total expenses, income, and balance.

4.1.3.2 Graphical representations shall illustrate spending patterns over time.

Priority Level: High

Precondition: User is logged in

Cross Reference: 4.1.1

4.1.4 Personalization and Settings

Functional Requirements:

4.1.4.1 Users shall have the option to customize the theme and language preferences.

4.1.4.2 Account settings shall allow users to change passwords and update profile information.

Priority Level: Medium

Precondition: User is logged in

Cross Reference: 4.1.1, 4.1.3

4.1.5 Reminder and Notification

Functional Requirements:

4.1.5.1 Users can set reminders for recurring expenses or upcoming bills.

4.1.5.2 The system shall send notifications for important financial events.

Priority Level: High

Precondition: User is logged in

Cross Reference: 4.1.1, 4.1.3, 4.1.4

4.1.6 Expense Entry

Functional Requirements:

4.1.6.1 Users can add new expenses with details such as date, category, amount, and description.

4.1.6.2 An option to upload receipts or attach documents shall be provided.

Priority Level: High

Precondition: User is logged in

Cross Reference: 4.1.1

4.1.7 Income Tracking

Functional Requirements:

4.1.7.1 Users can record income sources with details like date, source, and amount.

Priority Level: Medium

Precondition: User is logged in

Cross Reference: 4.1.1

4.1.8 Categories and Tags

Functional Requirements:

4.1.8.1 The system shall support categorization of expenses and income.

4.1.8.2 Users can add custom categories and tags.

Priority Level: High

Precondition: User is logged in

Cross Reference: 4.1.1, 4.1.6, 4.1.7

4.1.9 Budget Management

Functional Requirements:

4.1.9.1 Users can set budget limits for different categories.

4.1.9.2 Notifications shall be sent when approaching or exceeding budget limits.

Priority Level: High

Precondition: User is logged in

Cross Reference: 4.1.1, 4.1.5

4.1.10 Search and Filtering

Functional Requirements:

4.1.10.1 The system shall provide a search functionality to find specific transactions.

4.1.10.2 Filters for time periods, categories, and tags shall be available.

Priority Level: Medium

Precondition: User is logged in

Cross Reference: 4.1.1, 4.1.8

4.1.11 Reports and Analytics

Functional Requirements:

4.1.11.1 Users can generate reports and analytics to visualize spending habits.

4.1.11.2 Trends and insights based on historical data shall be provided.

Priority Level: High
Precondition: User is logged in
Cross Reference: 4.1.1, 4.1.6, 4.1.9

4.1.12 Currency Conversion

Functional Requirements:

4.1.12.1 The system shall support multiple currencies if users deal with expenses in different currencies.

Priority Level: Medium
Precondition: User is logged in
Cross Reference: N/A

4.1.13 Collaboration

Functional Requirements:

4.1.13.1 Users can share expenses and budgets with family or teammates.

4.1.13.2 Collaborative features for group financial planning shall be available.

Priority Level: Low
Precondition: User is logged in
Cross Reference: 4.1.6, 4.1.9

4.1.14 Backup and Restore

Functional Requirements:

4.1.14.1 The system shall regularly back up user data to prevent loss.

4.1.14.2 Users can restore their data in case of accidental deletion.

Priority Level: High
Precondition: User is logged in
Cross Reference: N/A

4.2 System Quality Attributes

4.2.1 Usability:

In the expense tracker new users should be able to create and manage expenses within 5 minutes of using the system.

Priority Level: High
Precondition: N/A

Cross Reference: N/A

4.2.2 Availability:

The expense tracker should have a system uptime of at least 99%, ensuring users can access their expense information.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.3 Scalability:

The expense tracker should handle a 50% increase in users and data volume without significant performance degradation.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.4 Security:

Sensitive financial information should be protected using industry-standard encryption. To access the account, users are required to input their email address and password, safeguarding against unauthorized access.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.5 Maintainability:

Updates or modifications to the expense tracker shall be completed within 48 hours to minimize downtime.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.6 Compatibility:

The expense tracker must be compatible with all major web browsers (e.g., Chrome, Firefox, Safari) and mobile platforms (e.g., iOS, Android) for consistent user experience.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.7 Efficiency:

The expense tracker application must have a minimum of 20 percent of available processor capacity and RAM unutilized under the anticipated peak load conditions.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.2.8 Accuracy:

Expense calculations, reports, and data integrity should be 100% accurate.

Priority Level: High

Precondition: N/A

Cross Reference: N/A

4.3 System Interface



Fig 1: App Loading page

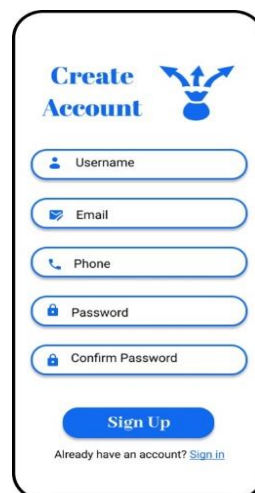


Fig 2: Sign Up

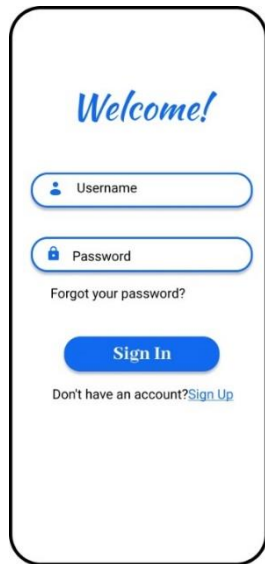


Fig 3: Sign in

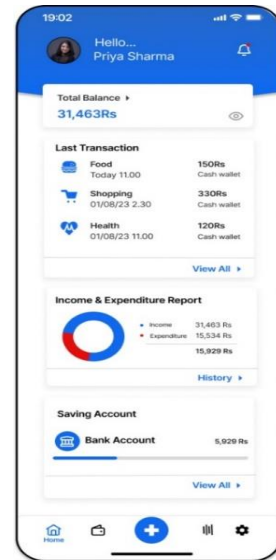


Fig 4: Dashboard and Budget Management

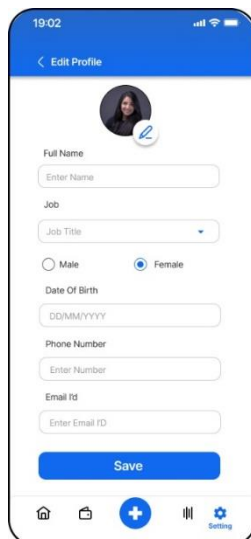


Fig 5: Personalization and Settings

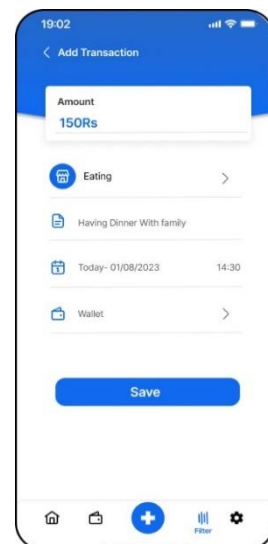


Fig 6: Expense Entry and Income Tracking

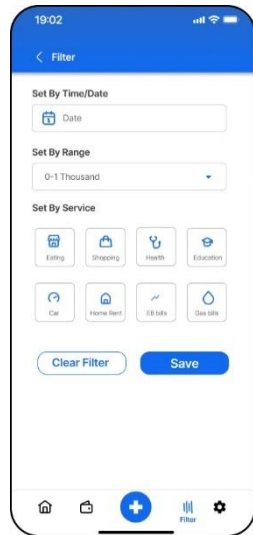


Fig 7: Search and Filtering

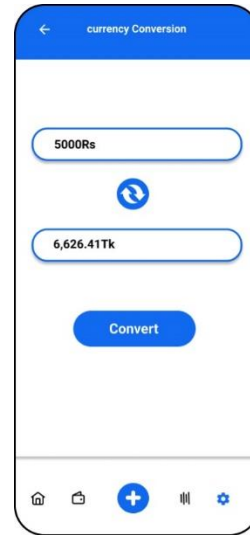


Fig 8: Currency Conversion

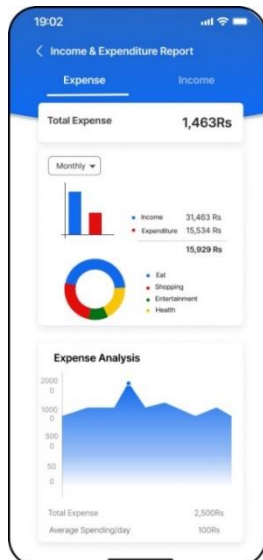


Fig 9: Reports and Analytics

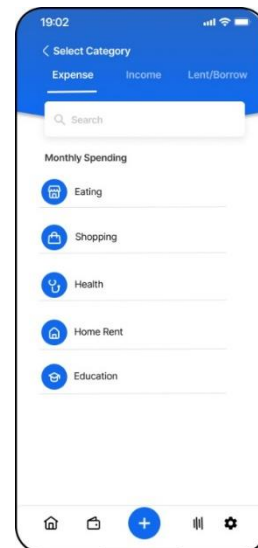


Fig 10: Categories and Tags

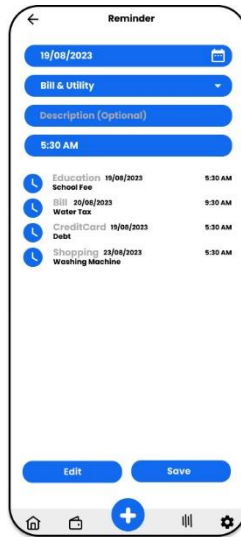


Fig 11: Reminder and Notifications

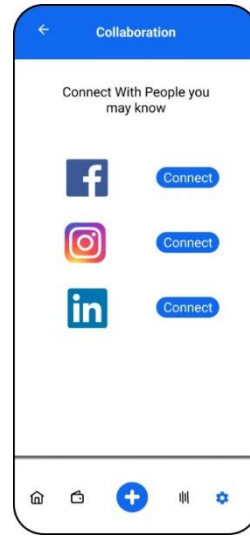


Fig 12: Collaboration

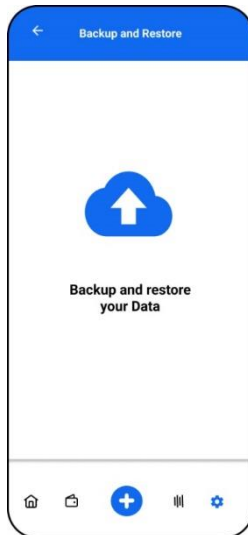


Fig 13: Backup and Restore

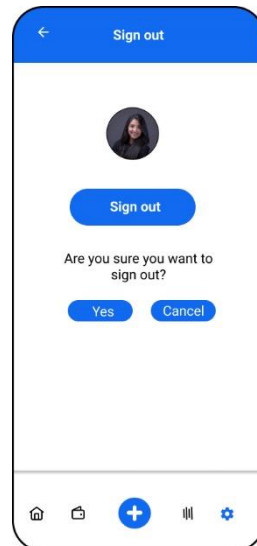


Fig 14: Sign out

4.4 Project Requirements

Our primary goal in project management is to help individuals or businesses monitor and manage their financial expenditures. However, various challenges such as time, budget, resources, and environment, etc. need to be managed effectively to achieve success. It is essential to complete the project within the deadline, on schedule, and within the allocated budget while ensuring the necessary functionality is added to the system. Proper management of resources is also crucial. Successfully addressing each constraint will lead to a favorable outcome.

Time: The project must be completed within the specified timeline and any delays should be minimized.

To develop a prototype of the software 120 hours of work are required, while for the development phase 480 hours are needed. Revision of the software will require 72 hours, and testing and debugging will take approximately 240 hours. In total 912 hours of work will be required. Assuming a daily work schedule of 10 hours, the project can be completed in approximately 91 days, which is equivalent to 3 months or 13 weeks.

Budget: The project must be completed within the allocated budget and any additional costs should be kept to a minimum.

Total budget 2,350,000 BDT.

Effort Estimation:

Let's, assume our project is an organic project, where $a=2.4$, $b=1.05$, $c=2.5$, $d=0.38$ according to the COCOMO model

The estimated size of the software product in Kilo Lines of Code is 10

Estimation of Development Effort, $E = a \times (KLOC)^b$

$$= 2.4 \times (10)^{1.05}$$

$$= 26.928 \text{ Person-Hours}$$

$$= 3500.64 \text{ Person-Hours}$$

Estimation of Development Time, $D = c \times (E)^d$

$$= 2.5 \times (26.928)^{0.38}$$

$$= 8.8 \text{ Months}$$

Required number of people = Effort/Time

$$= 26.928/8.8 = 3.06$$

Let's assume average salary of each employee in our company is TK 20,000 and they work 10 hours each day

Therefore, Rate of per hour = $20,000 \div (3.06 \times 10)$

$$= \text{TK } 653.60 \text{ per hour}$$

Cost = Effort \times Rate = 3500.64 Person-Hours \times TK 653.60 per hour

$$= 2,288,018.304 \text{ Taka}$$

Suppose other necessary cost will be 62,000 Taka

Total cost = $(2,288,018.304 + 62,000)$ taka

$$= 2,350,018.304 \text{ taka.}$$

5. FEATURES NOT TO BE TESTED

Here is the list of features not to be tested:

- **Collaboration:** The collaboration feature has been excluded from testing due to its non-mandatory status, resource constraints, and minimal impact on core functionality. User feedback did not emphasize its importance, and a risk assessment determined its exclusion poses minimal risk to the Expense Tracker's overall performance and security. This decision optimizes testing resources for more impactful areas. If prioritized in future updates, collaboration testing can be revisited.

We have only one feature which will not be tested. Rest of the features must be tested among all the features we have included for this project.

6. TESTING APPROACH

6.1 Testing Level

6.1.1 Unit Testing

In the unit testing phase of our Expense Tracker project, we embraced the Agile model to foster a collaborative and iterative testing approach. Following Agile principles, our development and testing teams worked in tandem throughout short development cycles or sprints. This facilitated continuous feedback, allowing us to identify and rectify defects early in the process. The Agile model's flexibility and adaptability were instrumental in ensuring that each unit of code underwent rigorous testing against predefined requirements, thereby enhancing the overall quality and maintainability of our software.

6.1.2 Integration Testing

For integration testing, we adopted the V-Model, a systematic and step-by-step approach that aligns testing activities with development phases. This model enabled us to detect and address interface issues between integrated components by emphasizing a parallel testing process. As we progressed through the development stages, we concurrently conducted integration tests to validate the interactions and dependencies between different modules. The V-Model's structured framework allowed us to identify and resolve integration challenges efficiently, ensuring the seamless functioning of our Expense Tracker application.

6.1.3 System Testing

In the system testing phase, we employed the Waterfall model to thoroughly evaluate the entire Expense Tracker system. This model follows a linear and sequential approach, where each phase must be completed before moving on to the next. This systematic progression ensured a comprehensive assessment of our software's functionality, performance, and security. By breaking down the testing process into distinct stages, we were able to identify and address issues systematically, resulting in a more robust and reliable Expense Tracker system.

6.1.4 Acceptance Testing

During acceptance testing, we implemented the Prototype model to gather early feedback from stakeholders and end-users. This model involves creating a working model of the software to demonstrate its features and functionality. By providing stakeholders with a tangible representation of the Expense Tracker, we could incorporate their insights and preferences early

in the development process. The Prototype model facilitated effective communication and collaboration, ensuring that the final product met user expectations and requirements.

6.2 Testing Tools

- JUnit
- Postman
- Selenium WebDriver
- JMeter
- Cucumber
- OWASP ZAP (Zed Attack Proxy)
- Figma

6.3 Meetings

The testing teams meet twice in a every week and identify the errors and solve the errors. The testing team manager meets once a month with the developer team manager. More meetings will be conducted if any major issue arises.

7. TEST CASES/TEST ITEMS

Test Case 1: Sign up

Project Name: Expense Tracker	Test Designed by: konak
Test Case ID: ET_1	Test Designed date: 25-11-23
Test Priority (Low, Medium, High): High	Test Executed by:
Module Name: User Signup	Test Execution date:
Test Title: Signup with the valid information	
Description: Check User signup works perfectly with the valid information.	

Precondition (If any): User must have valid user id and password				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the apps 2. Then Click user signup 3. Enter valid information 4. Click submit	Username: konak Password: @1234Hello	Signup successfully done		
Post Condition: User signup successfully with the valid information.				

Table 1 Test Case for Sign up

Test Case 2: Sign out

Project Name: Expense Tracker			Test Designed by: konak	
Test Case ID: ET_2			Test Designed date: 25-11-23	
Test Priority (Low, Medium, High): Low			Test Executed by:	
Module Name: User Sign out			Test Execution date:	
Test Title: Sign out with the valid information				
Description: Check User sign out works perfectly with the valid information.				
Precondition (If any): User must have valid user id and password				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the apps 2. Then Click user Sign out 3. Click Yes	N/A	Sign out successfully		
Post Condition: Users sign out successfully				

Table 2 Test Case for Sign out

Test Case 3: Dashboard

Project Name: Expense Tracker		Test Designed by: konak		
Test Case ID: ET_3		Test Designed date: 25-11-23		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Edit Profile		Test Execution date:		

Test Title: User can see the dashboard expenses, income, and balance				
Description: Graphical representations shall illustrate spending patterns over time				
Precondition (If any): The user must log into their account				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the application 2. Enter username 3. Enter password 4. Click submit	N/A	Profile edited successful		
Post Condition: User is validated with database and successfully login to account. Then user can update his/her profile.				

Table 3 Test Case for Dashboard

Test Case 4: Personalization and Settings

Project Name: Expense Tracker			Test Designed by: konak	
Test Case ID: ET_4			Test Designed date: 25-8-23	
Test Priority (Low, Medium, High): Medium			Test Executed by:	
Module Name: Personalization and Settings			Test Execution date:	
Test Title: User can update their previous profile				
Description: Update username, password, and photo				
Precondition (If any): The user must log into their account				
Test Steps	Test Data	Expected Results	Actual Results	Status
1.Go to the application 2.click settings 3.Give information 4.Click submit	Previous Information (Username: konak Password: @1234Hello)	Profile edited successful		
Post Condition: User is validated with database and successfully login to account. Then user can update his/her profile.				

Table 4 Test Case for Personalization and Settings

Test Case 5: Efficiency

Project Name: Expense Tracker	Test Designed by: konak
-------------------------------	-------------------------

Test Case ID: ET_5		Test Designed date: 25-8-23		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Efficiency		Test Execution date:		
Test Title: User can see the efficiency.				
Description: Efficiency refers to the ability of a system to perform its functions with minimal resource utilization, such as time, memory, and processing power.				
Precondition: The system has access to an adequate number of resources required for the operation				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Log in to the application 2. Use any feature	N/A	The applications efficiency is grate		
Post Condition: The application minimizes its overall resource footprint after the operation.				

Table 5 Test Case for Efficiency

Test Case 6: Accuracy

Project Name: Expense Tracker		Test Designed by: konak		
Test Case ID: ET_6		Test Designed date: 25-8-23		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Accuracy		Test Execution date:		
Test Title: user test the Accuracy				
Description: It ensures that the software produces correct and precise results in accordance with specified requirements				
Precondition: Valid and representative input data is provided to the app.				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the application 2. Login to the application	User Information (Username: konak Password: @1234Hello)	The Application accuracy is grate		
Post Condition: The app delivers correct and precise results consistent with specified requirements and user expectations.				

Table 6 Test Case for Accuracy

Test Case 7: Reminder and Notification

Project Name: Expense Tracker		Test Designed by: Taslima Akther Tuli		
Test Case ID: ET_7		Test Designed date: 18/12/2023		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Reminder and Notification		Test Execution date:		
Test Title: Verify setting remainder and receiving notification				
Description: Test the functionality of setting reminders for recurring expenses and receiving notifications for important financial events.				
Precondition (If any): User must be logged in				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the remainder and notification section. 2. Set a remainder. 3. Wait for notification. 4. Access the notification.	Date: 12/12/2023 Category: "Groceries" Description: “Bill Pay” Time: 05:30pm	User should be able to save remainder successfully and receive notification in expected time.		
Post Condition: The system successfully allows users to set reminders for recurring expenses, identifies upcoming bills, and sends notifications for important financial events. The user receives timely notifications with detailed information about the upcoming expense.				

*Table 7 Test Case for Reminder and Notification***Test Case 8: Expense Entry**

Project Name: Expense Tracker	Test Designed by: Taslima Akther Tuli
Test Case ID: ET_8	Test Designed date: 18/12/2023

Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Expense Entry			Test Execution date:	
Test Title: Verify adding new expenses with details and attaching documents.				
Description: Test the functionality of adding new expenses. Also, test the option to upload receipts or attach documents.				
Precondition (If any): User must be logged in				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the expense entry section. 2. Add new expense. 3. Attach document or receipts. 4. Click save button.	Date: 12/12/2023 Category: "Groceries" Amount: 30\$ Description: "Weekly grocery shopping" Document: [Receipt Image]	User should be able to add expense details and documents or receipts successfully.		
Post Condition: The system successfully allows users to add new expenses with details, and it allows to attach documents or receipts to the expense entry.				

Table 8 Test Case for Expense Entry

Test Case 9: Income Tracking

Project Name: Expense Tracker	Test Designed by: Taslima Akther Tuli
Test Case ID: ET_9	Test Designed date: 18/12/2023

Test Priority (Low, Medium, High): Medium			Test Executed by:		
Module Name: Income Tracking			Test Execution date:		
Test Title: Verify recording income sources with details.					
Description: Test the functionality of recording income sources with details like date, source, and amount.					
Precondition (If any): User must be logged in					
Test Steps		Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the income tracking section. 2. Click on the option to add new income. 3. Enter the income details (date, source, amount) 4. Click save button.		Date: 12/12/2023 Source: "Freelance Work" Amount: 150\$	User should be able to add income details successfully.		
Post Condition: The system should display the updated income details, and the changes should be saved successfully.					

Table 9 Test Case for Income Tracking

Test Case 10: Categories and Tags

Project Name: Expense Tracker	Test Designed by: Taslima Akther Tuli
Test Case ID: ET_10	Test Designed date: 18/12/2023
Test Priority (Low, Medium, High): High	Test Executed by:
Module Name: Categories and Tags	Test Execution date:
Test Title: Verify adding custom categories and tags for expenses and income.	

Description: Test the functionality of the system to support the categorization of expenses and income. Ensure users can add custom categories and tags.				
Precondition (If any): User must be logged in				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
<div>1. In expense/income entry, click "Select Category" dropdown.</div> <div>2. Click "+" button.</div> <div>3. Enter unique category name (e.g., "Gifts").</div> <div>4. Save category.</div> <div>5. Verify new category appears in dropdown and select it.</div> <div>6. Save expense/income entry.</div>	Category: “Entertainment”	<div>1. New category saved and displayed in dropdown.</div> <div>2. Selected category associated with saved entry.</div>		
Post Condition: The system should allow users to add custom categories and tags, and these should be successfully assigned to expenses and income entries.				

Table 10 Test Case for Categories and Tags

Test Case 11: Budget Management

Project Name: Expense Tracker		Test Designed by: Saeedullah Azim		
Test Case ID: ET_11		Test Designed date: 12.22.2023		
Test Priority: Medium		Test Executed by:		
Module Name: Budget Management		Test Execution date:		
Test Title: Verify the creation of a new budget				
Description: Test the functionality of creating a new budget in the Expense Tracker application.				
Precondition (If any): User must be logged into the application.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Budget Management section. 2. Click on the "Create New Budget" button. 3. Enter the budget details (e.g., name, amount, categories).	Budget Name: Monthly Budget Amount: \$2000 Categories: Groceries, Utilities, Entertainment	The new budget is successfully created and displayed in the Budget Management section.		

4. Save the new budget.				
Post Condition: The new budget is stored in the database, and relevant details are updated.				

Test Case 12: Search and Filtering

Project Name: Expense Tracker			Test Designed by: Saeedullah Azim	
Test Case ID: ET_12			Test Designed date: 12.22.2023	
Test Priority: Medium			Test Executed by:	
Module Name: Search and Filtering			Test Execution date:	
Test Title: Verify the search functionality for expenses				
Description: Test the effectiveness of the search feature in locating specific expenses within the Expense Tracker application.				
Precondition (If any): User must be logged into the application, and there should be existing expenses in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Search and Filtering section. 2. Enter specific criteria in the search bar (e.g., date, category). 3. Click the "Search" button.	Search Criteria: Date range (01/01/2023 - 01/31/2023), Category: Utilities	The search results display expenses that match the specified criteria.		
Post Condition: The search query is logged, and the relevant expenses are displayed to the user.				

Table 12 Test Case for Search and Filtering

Test Case 13: Reports and Analytics

Project Name: Expense Tracker		Test Designed by: Saeedullah Azim		
Test Case ID: ET_13		Test Designed date: 12.22.2023		
Test Priority: Medium		Test Executed by:		
Module Name: Reports and Analytics		Test Execution date:		
Test Title: Verify the generation of monthly expense report				
Description: Test the generation of a comprehensive monthly expense report in the Expense Tracker application.				
Precondition (If any): User must be logged into the application, and there should be existing expenses in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Navigate to the Reports and Analytics section. 2. Select the option for generating a monthly expense report. 3. Choose the specific month for the report. 4. Click the "Generate Report" button.	Month: February 2023	A detailed expense report for the selected month is generated and displayed.		
Post Condition: The generated report is stored for future reference, and relevant details are updated in the database.				

Table 13 Test Case for Reports and Analytics

Test Case 14: Scalability

Project Name: Expense Tracker		Test Designed by: Saeedullah Azim		
Test Case ID: ET_14		Test Designed date: 12.22.2023		
Test Priority: High		Test Executed by:		
Module Name: Scalability Testing		Test Execution date:		
Test Title: Verify system scalability under increased data volume				
Description: Test the application's ability to handle a growing volume of data to ensure scalability and performance as the user and data load increases.				
Precondition (If any): The system is running, and there is existing data in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Start with a baseline of existing data in the database. 2. Gradually increase the volume of data by adding new expenses, income entries, and budgets.	Various data volumes, including a baseline and increased volumes (e.g., 10x, 50x, 100x).	The application scales effectively, maintaining consistent response times and system performance as the data volume and concurrent user load increase.		
Post Condition: Scalability test results and recommendations for optimizing scalability are documented for future reference.				

Table 14 Test Case for Scalability

Test Case 15: Security

Project Name: Expense Tracker		Test Designed by: Saeedullah Azim		
Test Case ID: ET_15		Test Designed date: 12.22.2023		
Test Priority: High		Test Executed by:		
Module Name: Security Testing		Test Execution date:		
Test Title: Verify protection against SQL injection attacks				
Description: Test the application's resistance to SQL injection attacks to ensure the security of user data stored in the database.				
Precondition (If any): The system is running, and there is existing user data in the database.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Attempt to inject malicious SQL queries into input fields (e.g., username, expense entry). 2. Monitor the application's response and verify that it prevents or mitigates SQL injection attempts.	Malicious SQL injection attempts (e.g., ' OR '1'='1', DROP TABLE Users, etc.).	The application detects and mitigates SQL injection attempts, preventing unauthorized access or manipulation of the database.		
Post Condition: Security test results are documented, and necessary measures are taken to address any identified vulnerabilities.				

Table 15 Test Case for Security Testing

Test Case 16: Currency Conversion

Project Name: Expense Tracker		Test Designed by: Nafijul		
Test Case ID: ET_16		Test Designed date: 19/12/23		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Currency Conversion.		Test Execution date:		
Test Title: Perform Currency Conversion.				
Description: Verify that the Expense Tracker application accurately performs currency conversion for expense transactions.				
Precondition (If any): 1. User must have a registered account. 2. User must be logged into the Expense Tracker application.				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Go to the apps 2. Click on “Currency Conversion” 3. Enter the necessary expense details: a. Expense Amount: [Enter an amount] b. Currency: [Select a source currency] Converted To: [Select a target currency] 4. Save the expense entry	Bdt 1,000 taka	Currency Conversion Successfully done	US \$109,870	
Post Condition: 1. Expense entries are saved accurately with converted amounts. 2. The application displays the correct converted amounts based on the selected currencies.				

Table 16 Test Case for Currency Conversion.

Test Case 17: Usability

Project Name: Expense Tracker		Test Designed by: Saeedullah Azim		
Test Case ID: ET_17		Test Designed date: 19/12/23		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Usability		Test Execution date:		
Test Title: Evaluate the usability of the Expense Tracker's user interface				
Description: Test the user interface (UI) of the Expense Tracker application to ensure that it is intuitive, user-friendly, and provides a positive user experience.				
Precondition (If any): The system is running, and the user is logged in.				
Test Steps	Test Data	Expected Results	Actual Results	Status
1. Navigate through the main dashboard, expense entry, budget management, and reports sections. 2. Attempt to perform common tasks such as adding a new expense, creating a budget, and generating a report. 3. Evaluate the clarity of labels, buttons, and navigation elements.	N/A	Users can easily navigate the application, perform tasks intuitively, and find necessary features without confusion. The UI is responsive and visually consistent.		
Post Condition: 1. Shared expenses and budgets are accurately reflected in both the sender's and recipient's accounts. 2. The application provides appropriate notifications or indicators for shared expenses.				

*Table 17 Test Case for Usability***Test Case 18: Backup and Restore**

Project Name: Expense Tracker	Test Designed by: Nafijul
Test Case ID: ET_18	Test Designed date: 19/12/23

Test Priority (Low, Medium, High): High			Test Executed by:		
Module Name: Backup and Restore			Test Execution date:		
Test Title: Regular Backup and Data Restoration.					
Description: Verify that the Expense Tracker application can regularly back up user data and allow users to restore their data in case of accidental deletion.					
Precondition (If any): User must be logged into the Expense Tracker application.					
Test Steps		Test Data	Expected Results	Actual Results	Status
1. Click on to the 'Backup and Restore' section 2. Check the system's backup schedule and frequency settings. (e.g., daily, weekly). 3. Manually trigger a backup process. 4. Perform actions that lead to data changes (e.g., add, edit, delete expenses). 5. Initiate the data restoration process. 6. Confirm the data is restored successfully.		N/A	The system should consistently perform regular backups and provide users with an intuitive process to restore their data to any chosen backup point.		
Post Condition: 1. User data is regularly backed up according to the schedule. 2. Users can successfully restore their data to any chosen backup point.					

Table 18 Test Case for Backup and Restore

Test Case 19: Maintainability

Project Name: Expense Tracker	Test Designed by: Taslima Akther Tuli
Test Case ID: ET_18	Test Designed date: 18/12/2023
Test Priority (Low, Medium, High): High	Test Executed by:
Module Name: Maintainability	Test Execution date:
Test Title: Implement minor UI change to expense entry form and test maintainability.	

Description: Measure time to implement a minor UI update and assess maintainability.				
Precondition (If any): None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Assign task: Describe minor UI change to developer. 2. Record start time. 3. Implement and test update. 4. Record completion time. 5. Measure elapsed time.	Verify various expense entries across devices/browsers to confirm update stability.	Completion within 48 hours and no affected functionality or regressions.		
Post Condition: Updated UI maintains visual integrity and existing functionality after 48 hours.				

Table 19 Test Case for Maintainability

Test Case 20: Compatibility

Project Name: Expense Tracker		Test Designed by: Taslima Akther Tuli		
Test Case ID: ET_20		Test Designed date: 18/12/2023		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Compatibility		Test Execution date:		
Test Title: Verify core functionality across different browsers and platforms.				
Description: Test essential features and visual consistency on various browsers and devices.				
Precondition (If any): None				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

<ol style="list-style-type: none"> 1. Identify target browsers and platforms. 2. Access the application on each target. 3. Test core features (login, add/edit expenses, view reports, settings). 4. Check visual consistency (layout, fonts, colors, spacing, alignment). 5. Test responsiveness (resize windows, rotate devices). 6. Test interactions (clicks, text entry, selections, scrolling). 7. Check error handling (messages, warnings). 		Seamless and errorless functionality across all tested platforms.		
<p>Post Condition: After testing across platforms, the application should remain fully functional, preserve data integrity, and avoid errors or instability.</p>				

Table 20 Test Case for Compatibility

8. ITEM PASS/FAIL CRITERIA

- Tests must be completed within the specified amount of time.
- Functional requirements should work as expected.
- Tests must have 80% of success rate.
- Any bugs found during the test must be documented and addressed.
- Tests must not cause any system crashes or data loss.
- Documentation must be provided and accurate.
- User should not face difficulty while performing tasks.
- Error messages must be meaningful and self-explanatory.

9. TEST DELIVERABLES

The test deliverables typically include various documents and materials that help ensure the quality and reliability of the software.

Here is a list of the documents/materials we will deliver along with the testing process –

1. Test Plan	2. Test Data
3. Requirement Document	4. Design Document
5. Test Cases	6. Defect Reports
7. Acceptance Test Report	8. Security Test Reports
9. Partnership Materials	10. Test Summary Reports

10. STAFFING AND TRAINING NEEDS

Staffing and training are one of the crucial aspects of ensuring the effectiveness of the testing team. The composition of the testing team and the training needs will depend on various factors including project complexity, size, and the specific testing methodologies adopted.

10.1 Staffing

For recruiting staff, we will use both Horizontal and Vertical testing for this project.

- **Horizontal Testing is useful for:**
 - Flexibility and adaptability in employees.
 - Organization needs mix of skills and responsibilities.
 - Startups or small teams where individuals need to wear multiple hats.
- **Vertical Testing is useful for:**
 - Organization needs experts in specific domains or technologies.
 - Roles that require deep technical or domain-specific knowledge.

Balancing both approaches we will recruit staffs for this project because we need specialized candidates who possesses deep knowledge in specific domain such as performance and security.

10.2 Training

Training needs for recruited staffs can be divided into different parts.

Technical Training:

- **General Testing Concepts:**
 - Ensure all team members are familiar with testing fundamentals, testing types, and methodologies.
 - Training resources could include industry-standard testing certifications.
- **Testing Tools and Technologies:**
 - Provide training on testing tools used for functional testing, performance testing, and security testing.
 - For example, training on Selenium for automated testing, JMeter for performance testing and security testing tools.
- **Programming Languages:**
 - Familiarize team members with relevant programming languages used in testing such as Java, Python.
 - Enable them to understand and contribute to automated testing frameworks.
- **Database Testing:**
 - Train team members on database testing techniques.
 - Ensure they can verify data integrity, perform data migrations, and validate data transformations.

Domain-specific Training:

- **Financial Domain Knowledge:**
 - Provide training on financial concepts relevant to the expense tracker.
 - Ensure the team understands the significance of accurate financial data handling.
- **Regulatory Compliance:**
 - Train team members on relevant regulations impacting financial applications.
 - Ensure they understand the importance of compliance in the context of the expense tracker.

Soft Skills and Collaboration:

- **Communication Skills:**
 - Enhance communication skills to effectively report bugs, discuss issues and provide status updates.
- **Collaboration Training:**
 - Emphasize collaboration with other teams such as development and product management.
 - Encourage a culture of cross-functional teamwork.

Project-specific Training:

- **Expense Tracker Features:**
 - In-depth training on the functionality of the expense tracker.
 - Understanding user scenarios and business logic.
- **Testing Processes and Procedures:**
 - Detail the testing processes, methodologies and procedures specific to the project.
 - Ensure alignment with the overall development and release cycle.

Continuous Learning:

- Encourage continuous learning through regular knowledge-sharing sessions, workshops, and participation in relevant conferences or webinars.
- Keep the team updated on emerging testing trends and technologies.

By tailoring the staffing structure and training program based on our project's requirements we can build a testing team that is well-equipped to ensure the quality of this expense tracker application. Regular assessments and ongoing learning initiatives will contribute to the team's growth and effectiveness over time.

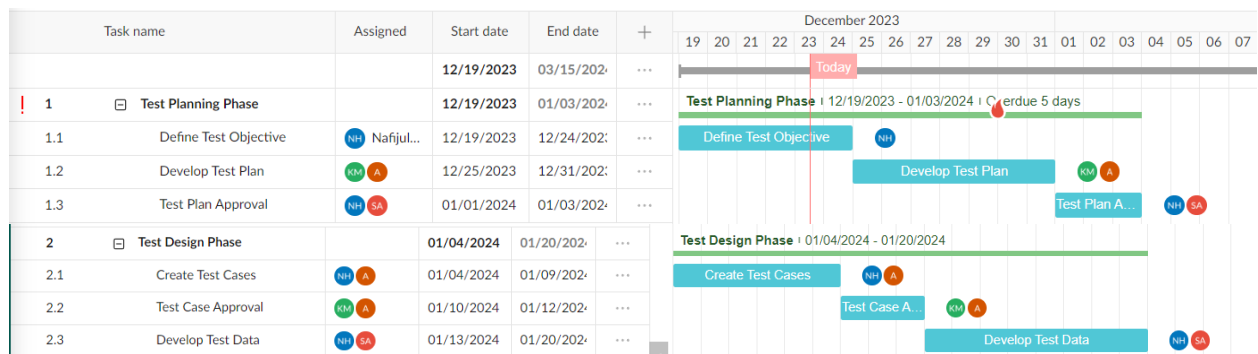
11. RESPONSIBILITIES

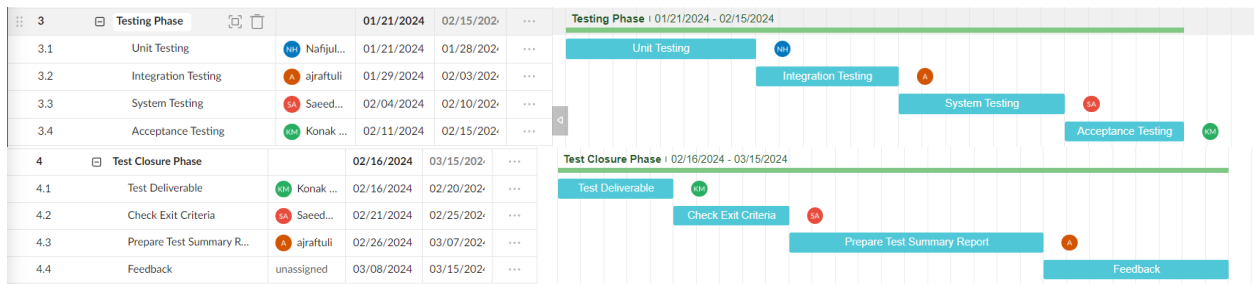
Roles and responsibilities of each stakeholder in the testing process:

Responsibilities of Stakeholder	Roles of Stakeholder							
	Project Manager	Product Manager	Test Manager	Test Lead	Testers	Performance Testers	Security Testers	Developers
System design reviews			X	X	X	X	X	
Detail design reviews			X	X	X	X	X	
Test approach and rules								X
Screen & report prototype reviews			X	X	X	X	X	
Unit test documentation & execution	X	X				X	X	X
Integration test documentation and execution	X	X				X	X	X
System test documentation & execution	X	X						X
Acceptance test documentation & execution	X	X						X
Change control				X	X	X	X	

12. TESTING SCHEDULE

A project management application tool named **GanttPro** has been used for scheduling -





Link: Using this link anyone can see our Project Scheduling Gantt Chart on GanttPro –

<https://app.ganttpro.com/shared/token/9c6a526274ccd6cd7adcc7eebff1eee81c0bf4390ef8ed6dde7842b19d15afd4/1310357>

13. PLANNING RISKS AND CONTINGENCIES

S/N	Risk Description	Probability	Impact	Mitigation Plan
1	Account security	10%	Moderate	Allow users to enter 5 times to enter incorrect password.
2	Deficient Requirements	40%	High	Collaborate closely with stakeholders to ensure clear and complete requirements. Establish a feedback loop for requirement clarification and updates.
3	Insufficient Test Data	30%	Medium	Create a comprehensive set of test data early in the project. Validate data for accuracy and completeness.
4	Resource Constraints	30%	High	Clearly identify resource requirements. Have contingency plans for additional resources. Communicate resource constraints to project stakeholders.
5	Communication Breakdown	40%	Medium	Establish clear communication channels. Schedule regular status meetings. Use collaboration tools for documentation and communication.

6	Exceeding budget	60%	High	Take some extra money from client for safety.
7	Unable to acquire required hardware for testing	5%	Low	Start testing after making sure all the required hardware's are available.

14. APROVALS

Name	Role	Responsibility	Date & Signature
KONAK MOZUMDER	Designer	Ensure that the design meets need for the solution	<i>Konak</i> 24/12/2023
MD SAEEDULLAH AZIM	Test Lead	A test lead oversees and coordinates the testing process, ensuring the successful execution of test plans and delivery of high-quality software.	<i>Azim</i> 24/12/2023
MD NAFIJUL HOQ	Project Manager	Planning Scheduling and Managing the delivery of software projects.	<i>Nafijul</i> 24/12/2023
TASLIMA AKTHER TULI	Business Analyst	Work with internal stakeholders, determine requirements and shine a light on any issues that may be affecting a business's bottom line.	<i>Tuli</i> 24/12/2023