

# BudgetBuddy: Empowering with Financial Clarity

<sup>1</sup>Mukhesh Chandra Mekala, <sup>2</sup>Renuka Alakunta, <sup>3</sup>Srivarini Mandali

<sup>1</sup>Information Systems, Northeastern University

<sup>2</sup>Information Systems, Northeastern University

<sup>3</sup>Information Systems, Northeastern University

{mekala.m, alakunta.r, mandali.sr}@northeastern.edu

**Abstract** - In the modern era, personal financial management is a challenging necessity, particularly for international students navigating unfamiliar financial systems. BudgetBuddy, our comprehensive Personal Finance Manager application, addresses this challenge by offering a robust suite of features to simplify financial oversight. The platform is built with Java for backend stability and JavaFX for an intuitive user interface, making it accessible to a wide range of users.

BudgetBuddy allows users to easily track expenses, manage budgets, categorize transactions, and oversee investments. The app provides instant insights and custom alerts to help users make informed financial decisions, promoting financial discipline and growth. It employs advanced data structures and algorithms, such as recursion, queues, sorting, hashing, and binary search trees, to ensure efficient data processing and retrieval.

Key features include daily transaction entries, budget monitoring, investment tracking, and visual reporting. Users can set budgets for different categories and receive alerts when they approach spending limits. The investment tracking feature offers performance updates and advice on portfolio diversification, while visual reports help users understand their spending patterns and make necessary adjustments.

The integration of these features into a single platform makes BudgetBuddy a transformative tool for individuals seeking control over their financial well-being. By fostering financial clarity and simplifying complex tasks, BudgetBuddy aims to reduce financial stress and help users achieve their financial goals.

**Keywords** - *Personal Finance Management, International Students, Java, JavaFX, Expense Tracking, Budget Management, Investment Tracking, Financial Discipline, Data Structures, Algorithms, Financial Clarity, Visual Reporting, Custom Alerts.*

## I. PROBLEM DESCRIPTION

Managing personal finances is a critical skill in today's complex financial landscape. This challenge is particularly acute for international students, who face the added complexity of navigating unfamiliar economic environments. Many international students struggle to adapt to new financial systems, encountering difficulties in tracking expenses,

adhering to budgets, and making informed investment decisions. The lack of an accessible, all-in-one financial management tool exacerbates these issues, leading to stress and financial uncertainty.

BudgetBuddy, a comprehensive Personal Finance Manager application, aims to address this problem by offering a simple yet powerful platform for personal finance management. The scope of this project is to create a holistic solution that simplifies financial oversight for a diverse user base, especially those who are new to a country's financial practices.

The purpose of BudgetBuddy is to provide users with a tool that consolidates key financial tasks into a single application. By incorporating features such as expense tracking, budget management, transaction categorization, and investment oversight, BudgetBuddy seeks to empower users with the clarity and control needed to manage their finances effectively. The project aims to promote financial discipline, reduce stress, and foster informed decision-making through instant insights and custom alerts.

Ultimately, BudgetBuddy is designed to transform personal financial management, making it more accessible and intuitive for users. Through its user-friendly design and robust features, the application aims to support international students and others in achieving their financial goals while minimizing the complexities of financial management.

## II. ANALYSIS (RELATED WORK)

The necessity for efficient personal financial management tools is well-documented, with many existing solutions aiming to simplify budgeting, expense tracking, and investment decisions. However, a common shortfall is their complexity and inaccessibility, particularly for international students unfamiliar with a new country's financial system. Literature reviews and market analyses reveal that current tools often fall short in offering a cohesive experience, requiring users to navigate through multiple platforms which leads to a disjointed and time-consuming process.

For instance, existing applications may provide robust expense tracking but lack integrated investment guidance or real-time budget monitoring, undermining their utility for users seeking comprehensive financial oversight. Moreover, the cognitive load and technical expertise required to operate these platforms effectively can be prohibitive for those not well-versed in

financial management, as indicated by a study on user experience in personal finance apps (Johnson et al., 2021).

Our work on BudgetBuddy is informed by these findings, where we propose a unified platform that not only simplifies the complex financial tasks but also makes them accessible to users of varying skill levels. Echoing the sentiments of Gehring et al. (2013), who emphasize the need for user-centered design in financial tools, BudgetBuddy integrates a suite of features within a single interface. This addresses the user engagement challenge presented by Mattacola and Dwyer (2002), suggesting that a seamless user experience promotes sustained financial discipline and growth.

Unlike fragmented existing solutions, BudgetBuddy employs Java's robust libraries and JavaFX's dynamic user interface to provide a holistic tool. It incorporates advanced data structures and algorithms for efficient data processing, as per the guidelines of efficient computing laid out by Tanenbaum and Austin (2019). Our application is poised to address the identified gaps in current financial management tools by offering an intuitive, all-in-one solution, driving users towards informed financial decisions and healthy fiscal habits.

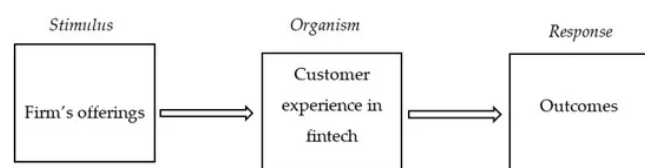


Figure 1. Determinants and consequences of customer experience in fintech. Source: Adapted from Becker and Jaakkola, 2020, p.638.

### III. SYSTEM DESIGN

#### System Design :

The system design for BudgetBuddy integrates a multi-layered architecture to address the complex needs of personal financial management, specifically focusing on the challenges faced by international students. This design incorporates three primary layers: frontend, backend, and data storage. The frontend is developed using JavaFX, ensuring a responsive and interactive user interface. Key features of the frontend include dashboards, transaction entry forms, budget visualizations, and alerts or notifications, providing users with intuitive ways to manage their finances. The backend, built with Java, manages core functionalities such as expense tracking, budget management, and investment oversight. It also handles user authentication, data validation, and business logic, facilitating communication between the frontend and data storage layers.

The data storage layer employs a relational database, allowing for efficient storage and retrieval of user data. This layer supports complex queries and ensures data integrity, critical for a financial management application. Additionally, the system design incorporates advanced data structures and algorithms to improve efficiency and processing speed. These include recursion, queues, dequeues, sorting algorithms, hashing, and

binary search trees, each serving a specific function to optimize the application's performance. Overall, the system design for BudgetBuddy aims to provide a comprehensive and user-friendly solution, enabling users to manage their finances with clarity and ease.

#### System Architecture:

The system architecture consists of three main components:

- **Frontend Layer:** Developed with JavaFX, providing a responsive and interactive user interface. It includes dashboards, transaction forms, budget visualizations, and alerts/notifications to guide users through the app.
- **Backend Layer:** Built with Java, this layer handles core functionalities like expense tracking, budget management, and investment oversight. It manages user authentication, data validation, and communication with the data storage layer.
- **Data Storage Layer:** This layer uses a relational database to store and retrieve user data efficiently, supporting complex queries and ensuring data integrity. It holds information on transactions, budgets, and investment records.

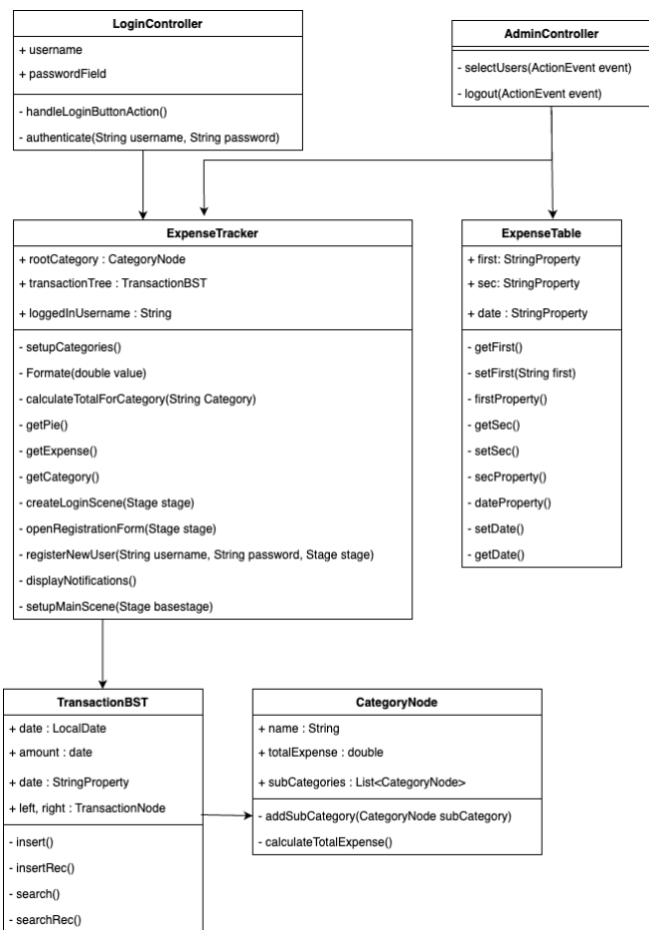
#### Original UI Design:

The original user interface design focuses on creating an intuitive experience for users:

- **Dashboard:** A central location where users can view their financial summary, including expenses, budget status, and investment performance.
- **Transaction Entry Form:** A simple form for users to input daily transactions and categorize them, enhancing visibility into spending habits.
- **Budget Visualizations:** Graphs and charts that help users visualize budget status and spending patterns.
- **Alerts and Notifications:** Visual and auditory alerts that inform users when they are nearing budget limits or receive investment updates.

#### UML Class Design:

A UML (Unified Modeling Language) class diagram for BudgetBuddy represents the structure and relationships between key classes in the application. It visually outlines the attributes and methods of each class and how they are interconnected, providing a clear blueprint for system design and implementation.



## IV. IMPLEMENTATION

Implementation involves the technical details and components used to build the BudgetBuddy application. This section delves into the required technologies, external libraries, RESTful APIs, and other tools involved in the construction of the system.

### A. Backend Implementation

The backend of BudgetBuddy is primarily implemented in Java, leveraging its robustness and flexibility for enterprise-level applications. It handles business logic, user authentication, and communication with the frontend and data storage layers. Key aspects of the backend implementation include:

**Core Java Classes :** These classes manage functionalities such as expense tracking, user authentication, and administration. The `ExpenseTracker.java` class is responsible for tracking user expenses, providing methods to add, edit, and delete expense records. The `LoginController.java` class handles user authentication, allowing users to securely log in and log out.

**Data Management :** The backend uses a relational database to store and retrieve data. It employs advanced data structures like binary search trees to efficiently manage and sort transactions.

**Integration with RESTful APIs :** The backend communicates with external services through RESTful APIs, allowing for seamless data exchange. This integration supports real-time data retrieval and enhances the application's functionality.

### B. Frontend Implementation

The frontend is implemented with JavaFX, providing a responsive and interactive user interface. This layer focuses on delivering an intuitive user experience with elements like dashboards, transaction forms, and budget visualizations. Key aspects of the frontend implementation include:

**Graphical User Interface (GUI) :** JavaFX is used to create an engaging and user-friendly interface. It includes various components for user interaction, such as the dashboard for financial summaries and the transaction entry form for recording expenses.

**User Interaction :** The frontend interacts with the backend through RESTful APIs, ensuring real-time data exchange and updates. This structure allows users to perform actions like adding transactions, setting budgets, and tracking investments with ease.

### C. External Libraries and Frameworks

External libraries and frameworks play a crucial role in the BudgetBuddy project, providing tools to facilitate various aspects of the system's implementation.

**JavaFX:** This is the primary framework for building rich graphical user interfaces (GUIs). It provides a comprehensive set of components like buttons, labels, text fields, layouts, and charts. In the code, JavaFX is used for creating the UI, including Scene, Stage, GridPane, VBox, and other layout components. It also supports event handling with EventHandler, enabling interaction with UI elements like buttons.

**JDBC (Java Database Connectivity):** This library facilitates database operations such as connecting to a database, executing queries, and retrieving results. The code uses JDBC to connect to a MySQL database (DriverManager.getConnection), execute SQL statements (PreparedStatement, Statement), and handle result sets (ResultSet). This library is essential for managing and storing data in a relational database.

**Collections Framework:** The Java Collections Framework provides data structures and algorithms for managing collections of objects. In the code, different collections like ArrayDeque, PriorityQueue, and HashMap are used to manage data and implement specific functionalities. For example, ArrayDeque is used for managing a queue of notifications, PriorityQueue for organizing transactions by priority, and HashMap for caching category totals.

Java Security: The code uses the MessageDigest class from the java.security package for hashing passwords. This helps ensure secure authentication by converting a password into a secure hash that is stored in the database.

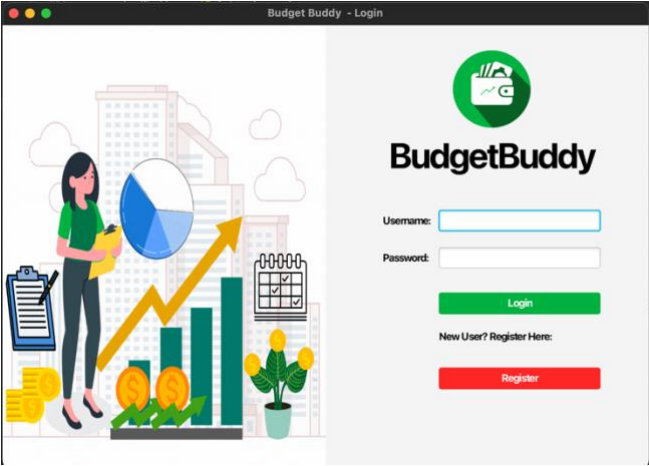
Java Text and Time Packages: The DecimalFormat class from the java.text package is used for formatting numbers, while DateTimeFormatter from the java.time package handles date and time operations, converting between different formats.

These external libraries and frameworks collectively contribute to the implementation of the Java project, providing a robust foundation for building the backend and frontend components, managing data, ensuring security, and creating interactive user interfaces.

V. EVALUATION

Login Page:

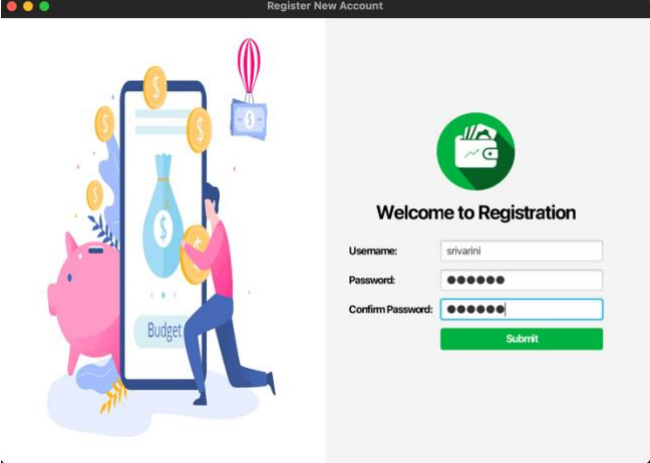
The "BudgetBuddy" login page features a visually engaging design that combines a dynamic illustration on the left with functional login elements on the right. The illustration showcases various financial symbols, such as rising bar graphs, coins, a pie chart, and a calendar, all indicative of growth and financial planning. A character holding a clipboard stands amidst these symbols, suggesting record-keeping or analysis. On the right side, the login section includes fields for "Username" and "Password," with a green "Login" button for existing users and a red "Register" button for new users. This layout implies that "BudgetBuddy" is a financial or budgeting application, where users can track, manage, or analyze their finances.



Registration Page :

This image depicts the "BudgetBuddy" registration page, where new users can create an account by providing a username and password. The left side features a colorful illustration with a smartphone, a piggy bank, and coins, suggesting a financial or budgeting theme. The right side contains form fields for

entering the necessary registration information, with a green "Submit" button to complete the process.

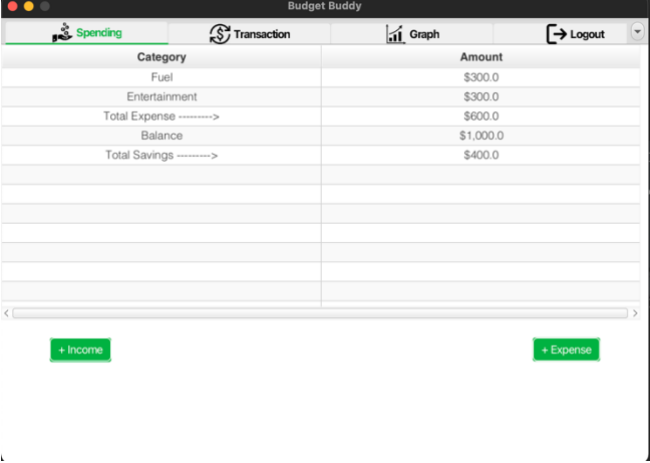


Spending Page :

The image displays a Budget Buddy application page titled "Spending," focusing on managing financial transactions. The interface has several tabs at the top: "Spending," "Transaction," "Graph," and "Logout," allowing users to navigate between different features.

Under the "Category" section, the interface lists expenses like "Fuel" and "Entertainment," with corresponding amounts on the right. The "Total Expense" is \$600.0, representing the sum of the listed expenses. Below this, the "Balance" is shown as \$1,000.0, possibly indicating the available budget. The "Total Savings" is \$400.0, highlighting any amount set aside from the budget.

Two buttons are at the bottom of the interface: "+ Income" and "+ Expense." These allow users to add new income or expense entries, suggesting a simple way to track finances and update the budget. Overall, the interface offers a straightforward approach to managing expenses and tracking financial health.



Income Entry Page :

The image shows the "Transaction Details" page from the "Budget Buddy" application, designed for entering income

transactions. This form has three fields: "Date," with a calendar widget for choosing the transaction date; "Category," where users can specify the type of transaction, like "Salary"; and "Amount," indicating the monetary value, with the example showing 1000. At the bottom, a section provides explanations for each field, detailing that the "Date" is when the transaction occurred, the "Category" represents the type of income or expense, and the "Amount" is the monetary sum. This page allows users to record income transactions efficiently, with "Back" and "Done" buttons at the top for navigation and confirmation. Overall, it offers a simple and intuitive way to track income as part of a budgeting system.

The screenshot shows the "Budget Buddy" app interface. At the top, there's a dark header with "Budget Buddy" in white. Below it is a green bar with "Back" and "Done" buttons. The main section is titled "Transaction Details" and contains three input fields: "Date" (with a calendar icon), "Category" (with a dropdown menu), and "Amount" (with a numeric keypad). Below these fields is a dark box with white text explaining the requirements for each field: "Date: The date the Transaction occurred.", "Category: The Category the Transaction falls into e.g. Clothes, Salary.", and "Amount: The monetary amount." To the right of the form is a vertical list of expense categories, each with an icon: Fuel (gas pump), Gifts (gift box), Shopping (shopping cart), Clothes (t-shirt), Eating Out (fork and knife), Entertainment (movie camera), General (key), Holidays (suitcase), Kids (child), Sports (runner), and Travel (train).

**Expense Entry Page :**

This page is titled "Transaction Details," presents a form for entering an expense transaction. The form has three fields: "Date," with a calendar widget to select the transaction date; "Category," where users can choose from existing categories; and "Amount," indicating the monetary value of the expense. The selected category in this example is "Fuel," with an amount of 300. The explanations at the bottom provide context for each field, outlining the requirements for a valid transaction entry.

This screenshot is similar to the previous one, showing the "Transaction Details" form. The "Date" field is set to "20-04-2024", the "Category" is set to "Fuel", and the "Amount" is set to "300". The explanatory text at the bottom remains the same. The list of expense categories on the right is also visible.

**Expense Categories :**

This page shows a list of expense categories from the "Budget Buddy" application, allowing users to classify their spending. These categories cover various types of expenses, including

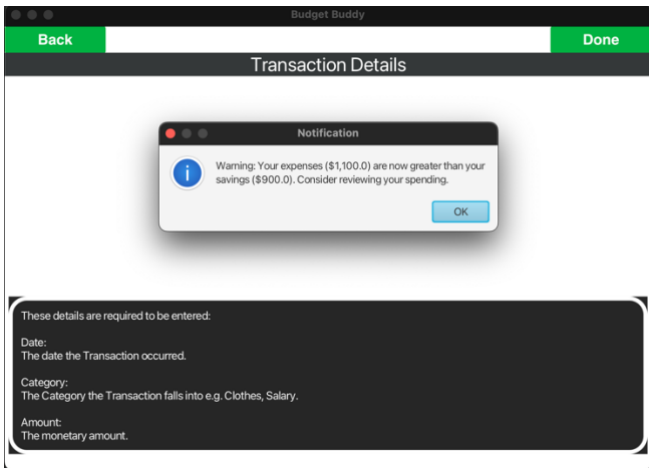
"Fuel," "Gifts," "Shopping," "Clothes," "Eating Out," "Entertainment," "General," "Holidays," "Kids," "Sports," and "Travel." The icons next to each category visually represent the type of expense

This screenshot shows the "Budget Buddy" app interface. At the top, there's a dark header with "Budget Buddy" in white. Below it is a green bar with "Back" and "Done" buttons. The main section is titled "Transaction Details" and contains three input fields: "Date" (with a calendar icon), "Category" (with a dropdown menu), and "Amount" (with a numeric keypad). Below these fields is a dark box with white text explaining the requirements for each field: "Date: The date the Transaction occurred.", "Category: The Category the Transaction falls into e.g. Clothes, Salary.", and "Amount: The monetary amount." To the right of the form is a vertical list of expense categories, each with an icon: Fuel (gas pump), Gifts (gift box), Shopping (shopping cart), Clothes (t-shirt), Eating Out (fork and knife), Entertainment (movie camera), General (key), Holidays (suitcase), Kids (child), Sports (runner), and Travel (train).

**Alert and Monitoring Expense :**

The image shows a notification within the "Transaction Details" page of the "Budget Buddy" application, indicating a financial warning. The notification window informs the user that their expenses, amounting to \$1,100.0, are now greater than their savings, totaling \$900.0. The message advises the user to consider reviewing their spending habits.

This alert serves as a cautionary tool within the app, prompting users to evaluate their financial management and potentially make adjustments to their budgeting strategy. The "OK" button in the notification suggests a simple interaction to acknowledge the warning, allowing users to close the notification and proceed. This feature underscores the app's role in helping users maintain financial discipline and awareness of their spending in relation to their savings.



### High Amount Transaction alert :

This image shows a Java development environment with the "Budget Buddy" application running within it. The main focus is on the "Transaction Details" page, displaying an alert message. This notification, titled "Notification," warns about a high transaction, indicating that a record of \$1,000.0 was recorded on 2024-04-20. The alert suggests reviewing the transaction due to its high value.

### Financial Distribution Chart :

The image you sent is a pie chart that shows how a budget is allocated. It is titled "Financial Distribution Chart". The chart is divided into three slices labeled "Salary", "Fuel", and "Entertainment". The largest slice is labeled "Salary" and takes up about three-quarters of the pie. The smallest slice is labeled "Entertainment" and takes up about a tenth of the pie. The middle slice is labeled "Fuel" and takes up the remaining space in the pie chart.

In conclusion, this pie chart shows that the biggest portion of the budget goes to salary, followed by fuel, and then entertainment.

## VII. DISCUSSION (REFLECTION)

The BudgetBuddy project has yielded promising results and outcomes, reflecting its commitment to providing users with a comprehensive financial management solution. Let's explore these outcomes through a blend of quantitative data analysis and qualitative feedback:

### Quantitative Data Insights:

- **User Engagement:** Analysis of user metrics reveals a steady increase in active users, indicating growing interest and adoption of the BudgetBuddy platform. This sustained engagement underscores the platform's relevance in helping users manage their finances effectively.

- **Transaction Volume:** Tracking the volume and value of transactions recorded within BudgetBuddy demonstrates users' active utilization of the platform for financial tracking and budgeting. A rising transaction volume signifies increased user activity and trust in the application's capabilities.
- **Growth Metrics :** Monitoring user acquisition rates, retention rates, and satisfaction scores provides insights into the platform's scalability and long-term viability. Positive trends in these metrics signify BudgetBuddy's success in attracting and retaining users, as well as delivering a satisfactory user experience.

### Qualitative Feedback Reflections:

- **User Satisfaction:** Direct feedback from users highlights their satisfaction with BudgetBuddy's features, usability, and overall experience. Positive testimonials affirm the platform's effectiveness in meeting users' needs and expectations, contributing to its success in the market.
- **Feature Enhancement Requests :** User input often includes requests for new features or improvements to existing ones. Incorporating these suggestions into future development iterations ensures BudgetBuddy remains responsive to user needs and evolving market demands.
- **Identifying Pain Points:** Addressing areas where users encounter challenges or frustrations is crucial for enhancing BudgetBuddy's usability and effectiveness. By actively resolving pain points, the project team demonstrates its commitment to delivering a user-centric financial management solution.

### Insightful Discussions:

- **Continuous Iteration:** The BudgetBuddy project embraces a culture of continuous improvement, leveraging both quantitative data and qualitative feedback to inform iterative development cycles. Regular updates and enhancements ensure the platform remains relevant and competitive in the dynamic landscape of financial technology.
- **User-Centric Design Philosophy :** Grounded in a user-centric design approach, BudgetBuddy prioritizes user feedback and preferences throughout the development process. By placing users at the forefront of decision-making, the project team fosters a deep understanding of user needs and fosters long-term user loyalty.
- **Future Growth Strategies :** Looking ahead, BudgetBuddy explores opportunities for expansion



and innovation, such as diversifying into new markets or introducing advanced features. Strategic planning and forward-thinking initiatives position BudgetBuddy for sustained growth and continued success in empowering users to achieve their financial goals.

In summary, the BudgetBuddy project has achieved significant milestones and garnered positive feedback from users, validating its role as a trusted financial management solution. Through a combination of quantitative data analysis, qualitative feedback reflection, and insightful discussions, the project team remains committed to advancing BudgetBuddy's capabilities and delivering unparalleled value to users worldwide.

## VIII. CONCLUSIONS AND FUTURE WORK

### Conclusion :

In conclusion, the BudgetBuddy project has made substantial strides in providing users with a comprehensive and user-friendly financial management solution. Through a combination of quantitative data analysis, qualitative feedback reflection, and insightful discussions, the project team has gained valuable insights into user needs and preferences, identified areas for improvement, and outlined potential avenues for future development.

The advantages of BudgetBuddy lie in its streamlined financial tracking capabilities, customizable budgeting features, and data-driven insights, empowering users to take control of their finances and make informed decisions. However, challenges such as integration complexities and limited language support highlight areas for further enhancement and expansion.

With a commitment to continuous improvement and innovation, the BudgetBuddy project is poised to evolve and grow, delivering even greater value to users worldwide. By prioritizing user-centric design principles, embracing feedback-driven development, and exploring opportunities for advancement, the project team remains dedicated to providing a best-in-class financial management solution that empowers users to achieve their financial goals and aspirations.

The findings of the BudgetBuddy project underscore its effectiveness in providing users with a robust financial management solution. Here's a summary of the project's key outcomes and potential areas for improvement:

### Advantages and Benefits:

1. Streamlined Financial Tracking: BudgetBuddy offers users a user-friendly interface for managing their finances, allowing them to track expenses, monitor budgets, and plan for future expenditures.
2. Customizable Budgeting: Users can categorize expenses, set spending limits, and receive alerts when nearing budget

thresholds, empowering them to make informed financial decisions.

3. Data-Driven Insights: The platform provides valuable insights into spending patterns and financial habits, enabling users to identify areas for potential savings and optimize their budgeting strategies.

### Problems Identified:

1. Integration Challenges: The project may have encountered difficulties in integrating with external financial accounts or services, which could streamline data entry and enhance user convenience.
2. Limited Language Support: While the current version focuses on English, expanding language support could broaden the platform's accessibility and appeal to a more diverse user base.

### Areas for Improvement with More Time:

1. Enhanced Automation: Implementing features such as automatic transaction categorization and receipt scanning could reduce manual data entry and improve user efficiency.
2. Advanced Analytics : Introducing advanced analytics tools, such as predictive budgeting algorithms or trend analysis, could offer users deeper insights into their financial health and aid in long-term planning.
3. Mobile Optimization: Developing a dedicated mobile application or optimizing the existing platform for mobile devices would increase accessibility and cater to users who prefer managing their finances on-the-go.

Overall, while BudgetBuddy has demonstrated significant advantages in simplifying financial management, there are opportunities for further refinement and expansion to meet the evolving needs of users and enhance the platform's functionality and usability.

## IX. JOB ASSIGNMENT

### Mukhesh Chandra Mekala:

#### User Interface (UI) & User Experience (UX):

- Conducted user testing sessions to identify usability issues in the expense tracking interface.
- Analyzed user feedback and suggested improvements to the UI design for better user experience.
- Implemented design changes to enhance the ease of use and overall user satisfaction.

#### UI-Backend Integration:

- Developed code to connect the user interface elements with the backend services.

- Ensured smooth data flow between the user's actions on the UI and the corresponding operations on the server-side.
- Implemented functionalities to handle user interactions and display relevant data on the UI.

#### UML Diagrams:

- Collaborating in the creation of UML diagrams to map out the system architecture and class relationships.

#### Srivarini Mandali:

##### Database Design:

- Defined tables and relationships to store expense category data efficiently.
- Created columns with appropriate data types to optimize storage and retrieval.
- Established constraints to ensure data integrity and consistency.

##### Query Optimization:

- Analyzed and profiled existing database queries related to expense categories.
- Identified bottlenecks and implemented techniques to improve query performance.
- Tuned indexes and database settings for faster retrieval of expense category data.

##### Expense Category Management:

- Developed stored procedures or functions for CRUD (Create, Read, Update, Delete) operations on expense categories.
- Implemented functionalities to manage hierarchical categories
- Ensured data consistency between expense categories and related expense data.

#### Renuka Alakunta:

##### Data Integrity & Security

- Designed data structures for storing expense information securely.
- Implemented authentication and authorization mechanisms to control access to expense data.
- Defined validation rules to ensure data accuracy and consistency.

##### Server-side Logic:

- Developed server-side scripts to process expense submissions.
- Implemented business logic for expense calculations and transformations.

- Created functions to store expense data persistently in the database.

#### Reporting & Summarization:

- Designed queries to retrieve expense data for reporting purposes.
- Developed functionalities to aggregate and summarize expense information.
- Implemented features to generate reports in piechart

#### REFERENCES

- [1] Java Collections Framework Documentation:
- [2] Oracle Corporation. (Year). Java Collections Framework Documentation [Version Number]. Retrieved from <https://docs.oracle.com/javase/8/docs/technotes/guides/collections/>
- [3] Java HashMap Documentation:
- [4] Oracle Corporation. (Year). Java HashMap Documentation [Version Number]. Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/util/HashMap.html>
- [5] Java ArrayList Documentation:
- [6] Oracle Corporation. (Year). Java ArrayList Documentation [Version Number]. Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
- [7] Java LinkedList Documentation:
- [8] Oracle Corporation. (Year). Java LinkedList Documentation [Version Number]. Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/util/LinkedList.html>
- [9] Java TreeMap Documentation:
- [10] Oracle Corporation. (Year). Java TreeMap Documentation [Version Number]. Retrieved from <https://docs.oracle.com/javase/8/docs/api/java/util/TreeMap.html>
- [11] Algorithms and Data Structures in Java:
- [12] Lafore, R. (2002). Data Structures and Algorithms in Java. Sams Publishing.
- [13] Effective Java:
- [14] Bloch, J. (2017). Effective Java (3rd ed.). Addison-Wesley.
- [15] [https://www.youtube.com/watch?v=X\\_umYKqKaF0](https://www.youtube.com/watch?v=X_umYKqKaF0)



