

# Expense Tracker Project

Nupoor Korde,<sup>2</sup>Angel Fernandes,<sup>2</sup>Osborne Lopes

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

<sup>2</sup>Software Engineering Systems, Northeastern University

{korde.n, fernandes.ang, lopes.o }@northeastern.edu

**Abstract** - The Expense Tracker project is a comprehensive JavaFX application designed within the MVC framework, focused on efficient management of personal finances. The system includes user registration and login functionalities, ensuring a secure and personalized experience for each user. The core functionalities include seamless addition and automatic categorization of both incomes and expenses. Through intuitive user interfaces, users can effortlessly input financial transactions, leveraging the MVC architecture to maintain a clear separation between data, user interface, and business logic. The Model component handles data storage and manipulation, the View ensures a responsive and user-friendly interface, while the Controller orchestrates user inputs and updates the underlying data model. One notable feature is the automatic recognition of income and expense categories, enhancing user convenience by eliminating the need for manual categorization. Leveraging Java's object-oriented principles, the project employs polymorphism and encapsulation to create a flexible and extensible system capable of accommodating future enhancements. The project culminates in the creation of a visually informative pie chart. This chart dynamically represents the user's financial landscape by visually segregating incomes and expenses. The integration of JavaFX's charting capabilities enhances the project's visual appeal and provides users with a quick, at-a-glance understanding of their financial distribution.

**Keywords** – *Expense Tracker, Income, Expense, JavaFX, Java, User Registration, Automatic categorization, Pie-chart*

## I. PROBLEM DESCRIPTION

In the realm of personal finance management, the ever-growing complexity of financial transactions necessitates an efficient and intuitive solution. This project addresses the challenges faced by individuals in tracking and analyzing their incomes and expenses. The absence of a consolidated platform often leads to cumbersome manual record-keeping, making it difficult for users to gain comprehensive insights into their financial health.

The scope of this work extends to the development of an Expense Tracker using JavaFX and Java, orchestrated within the Model-View-Controller (MVC) framework. The primary goal is to create a user-friendly application that facilitates seamless user registration and login processes, ensuring data security and personalized financial tracking.

The project focuses on automating the recording and categorization of both incomes and expenses, alleviating the burden of manual data entry. Leveraging object-oriented design principles, the implementation incorporates features such as

polymorphism and encapsulation to enhance flexibility and accommodate potential future expansions.

Furthermore, the project aims to provide users with a visual representation of their financial data through the creation of a pie chart. This dynamic chart serves as an intuitive tool for users to grasp the distribution of their incomes and expenses at a glance, promoting a deeper understanding of their financial landscape.

In essence, the Expense Tracker project emerges from the need to simplify and streamline personal financial management. By combining robust technological frameworks, user-centric design, and intelligent automation, the project aspires to empower individuals with a comprehensive and accessible tool for effective tracking and analysis of their financial resources.

## II. ANALYSIS (RELATED WORK)

The primary goal of the project is to address challenges related to personal finance management. This involves creating a solution that is user-friendly, automated, and comprehensive. By adopting the Model-View-Controller (MVC) structure, the project aims to provide a well-organized and modular architecture for efficient tracking and analysis of incomes and expenses.

Users are required to log in to their accounts or register if they are new to the service. This ensures a personalized experience and enables the application to securely store and retrieve individual user data.

Once logged in, users can enter details of their financial transactions. This includes providing a description for the transaction, specifying the amount, selecting a relevant category (e.g., groceries, utilities, salary), and choosing the date of the transaction. This user-friendly interface simplifies the process of inputting financial data.

The application automates the categorization of transactions as either income or expense. This automation helps users save time and ensures accuracy in tracking their financial activities. In the user interface, incomes are displayed in green, and expenses are highlighted in red, making it easy for users to distinguish between the two.

The application calculates and displays the overall balance resulting from all recorded transactions. The balance is

presented prominently, allowing users to quickly assess their financial standing. A negative balance is indicated for cases where expenses exceed incomes, providing a clear visual representation of financial health.

To enhance user experience, the application includes an illustrative pie chart. This chart visually represents the distribution of incomes and expenses across different categories. Visualizations such as pie charts can offer users a quick and intuitive understanding of their spending patterns.

A comprehensive review of existing literature and prior works in the domain of personal finance management reveals significant insights and various methodologies employed to address similar challenges.

Chandini et. al.'s work [1], developed an all-purpose PHP expenses tracker eliminating the need for maintaining Excel sheets and aims to enhance efficiency by including setting daily expense limits, tracking payment methods, receiving email notifications for budget exceedance, and sending monthly expense reports to user emails. The study concludes that the developed system is more efficient than traditional methods, providing an attractive interface for users.

Jhadav et al. [2] designed an expense tracker application as a solution to monitor and reduce spending, allowing users to categorize expenses, track daily costs, and view spending details through graphs and figures. The system divides daily expenses based on income and provides features like transaction receipts, statistics, and the ability to set alerts. The results showcase pictorial representations of transactions and an exported CSV file. The conclusion highlights the software's development stages, addressing issues associated with manual approaches and the potential impact on societal well-being.

Thanpal et al.'s work [3], aims to provide a reminder and a platform for users to input information about their income sources and expenses for a specific date or month. The proposed system includes features such as adding income and expense details, categorizing expenses, exporting income and expenses, removing export files, and viewing income and expenses.

### III. SYSTEM DESIGN

The system is meticulously crafted to emulate the look and feel of a personalized finance application, ensuring a seamless and familiar user experience. To manage user-related functionalities, the system employs the User class, userService, userController, and userRepository. These components collectively store and handle user details, including automatically generated IDs, usernames, and passwords, providing a secure and structured foundation for user interactions.

For transaction-related operations, the Transaction class, along with transactionService, transactionController, and transactionRepository, is implemented. This set of classes

manages the storage and retrieval of transaction details such as description, amount, category, and creation date, facilitating the core functionality of the expense tracker.

To accommodate user-initiated updates, the system includes the editTransaction class and editTransactionController, responsible for modifying transaction details as prompted by the user. This feature enhances the system's flexibility, allowing users to edit and refine their financial records.

The PieChartController takes charge of acquiring and presenting data in the form of two pie charts, distinctly visualizing income and expenses. This component enhances the user's analytical capabilities, providing a graphical representation of their financial distribution.

The system unfolds across five UI pages, each meticulously designed to correspond with specific functionalities. Each UI page is complemented by its corresponding controller class, ensuring a modular and organized structure. This design choice promotes code maintainability and readability, enabling efficient development and future enhancements.

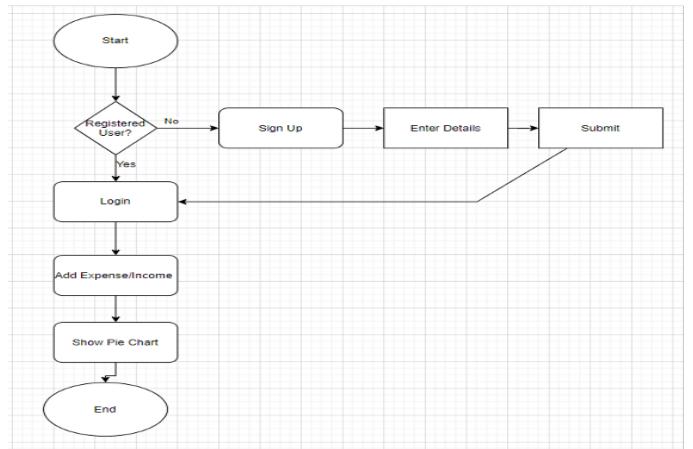


Figure 1. Draft UI Diagram

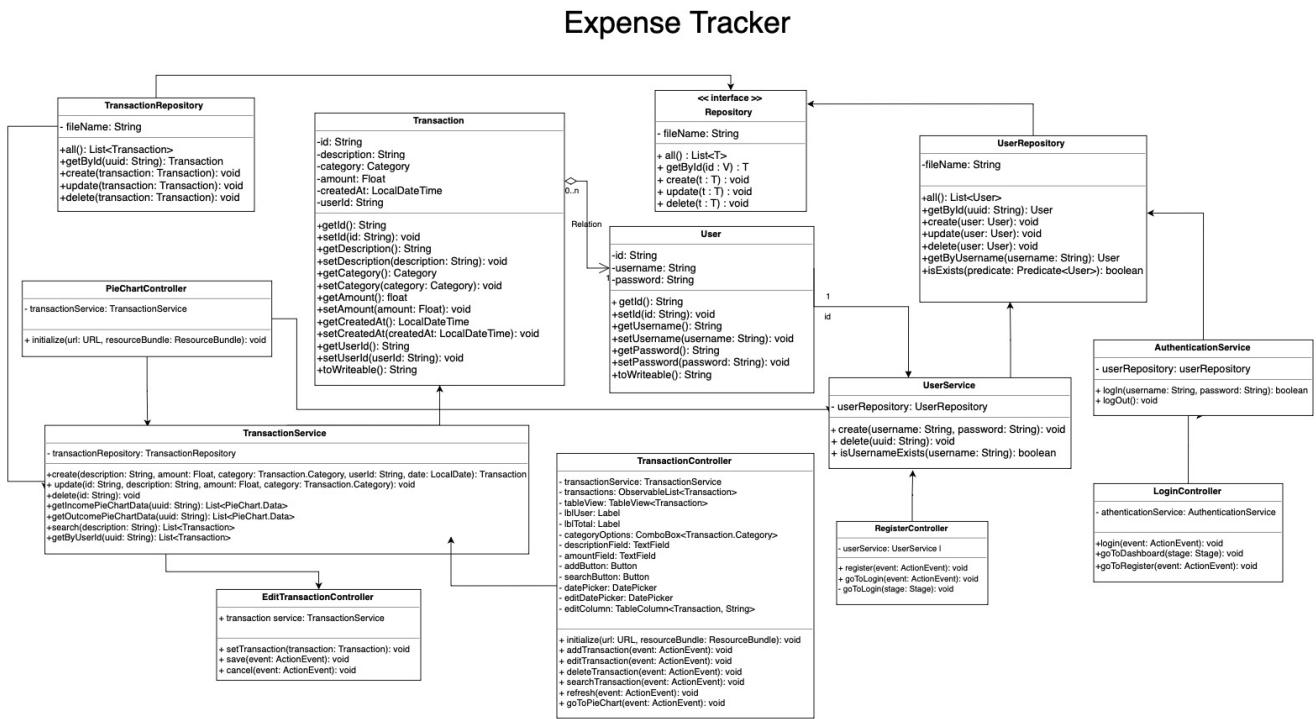


Figure 2. UI Diagram

#### **IV. IMPLEMENTATION**

#### A. *Class Definition*

All classes in our project make use of class definitions to encapsulate methods for showing JavaFX alerts, formatting and parsing LocalDateTime objects.

### B. *Inheritance / Polymorphism*

The class **EditButtonTableCell** extends `TableCell<S, T>`.  
the class **TransactionRepository** implements the  
`Repository<Transaction, String>` interface involving  
inheritance and supporting polymorphism.

### C. Abstract Classes / Interfaces

The class **EditButtonTableCell** is an abstract class that extends `TableCell<S, T>`, and it implements functionality specific to table cells.

The class **PieChartController** and **TransactionController** utilizes the **Initializable** interface, which requires the implementation of the **initialize** method.

The class **Repository** is an interface.

The **TransactionRepository** and **UserRepository** class implement the **Repository** interface.

#### D. Iterators

The **TransactionRepository** and **UserRepository** class use iterators in the update and delete methods, where a while loop is used with an Iterator to iterate through a collection of transactions.

### E. *Lists*

The class **EditTransactionController** utilizes a ComboBox (categoryOptions) to present a list of Transaction.Category values.

The class **PieChartController** and **TransactionController** uses ObservableList for managing PieChart data (incomePieChartData and expensePieChartData).

The class **Repository** uses `List<T>` in the method signatures. The class **TransactionRepository** and **UserRepository** use `List<Transaction>` and `List<User>` resp. to store and manipulate collections of transactions and users resp. in the all update, and delete methods.

The **TransactionService** class uses `List<Data>` to store and manipulate lists of data for pie charts in the `getIncomePieChartData` and `getExpensePieChartData` methods.

Upon launching the application, the user is immediately presented with the login screen, serving as the initial action. This intentional design choice prioritizes user authentication and interaction.

The user is given the choice to either log in using existing credentials or to register by providing new authentication details. These credentials, such as usernames and passwords, are stored as key-value pairs in a text file, creating a simple and accessible storage mechanism for user information.

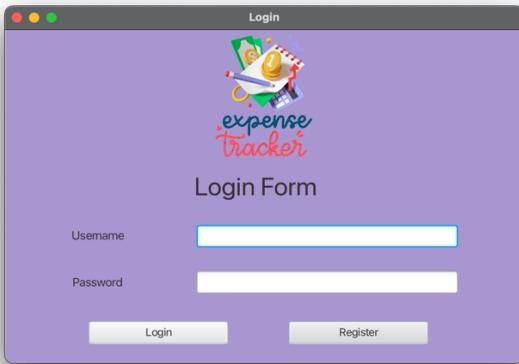


Figure 3. Login Screen

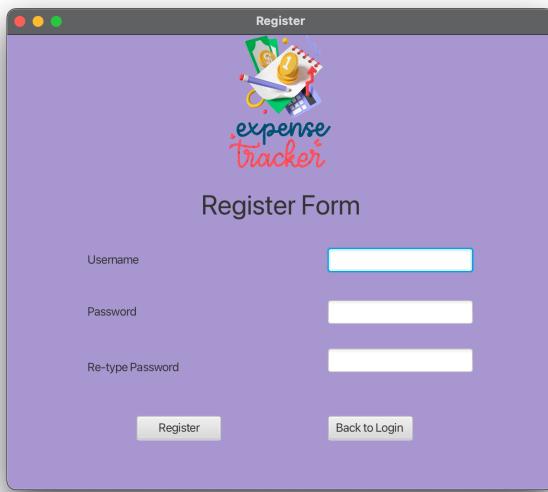


Figure 4. Registration Screen

To ensure data accuracy and completeness, the system validates the user's input. If the user fails to provide any of the required authentication information during the login or registration process, the system promptly triggers an error notification. This approach serves to guide users toward fulfilling the necessary criteria for authentication, enhancing the overall user experience by providing clear feedback and minimizing potential entry errors.

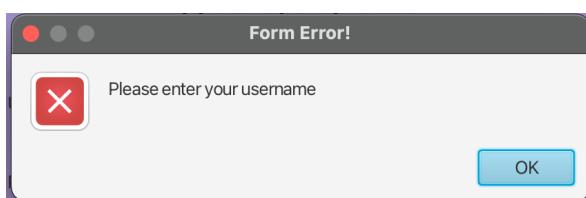


Figure 5. Error Screen

In essence, the login screen serves as the gateway to the application, emphasizing the importance of user authentication. The option to either log in or register, coupled with straightforward storage of credential data and error handling mechanisms, collectively contribute to a secure and user-friendly onboarding process within the application.

#### A. Displaying and Editing the Transactions

Following successful login, the user is seamlessly directed to the transaction page, where a clear and organized representation of their financial activities awaits. The user interface for this page is implemented using a TableView, providing a structured layout for transaction data display.

Within the transaction page, the user is empowered with versatile functionalities. They can effortlessly add new transactions using pre-specified categories, update existing ones, or remove transactions as needed. This interactive approach enhances the user's control over their financial records, contributing to a dynamic and user-centric experience.

Description	Amount	Created At	Edit
Salary	5000.0	2023-12-04T00:00	Edit
Rent	-2200.0	2023-12-04T00:00	Edit
Internet	-30.0	2023-12-04T00:00	Edit
Electricity	-50.0	2023-12-05T00:00	Edit
Groceries	-100.0	2023-12-05T00:00	Edit
Apple Music	-5.0	2023-12-05T00:00	Edit

Figure 6. Display Transactions Screen & Category Options

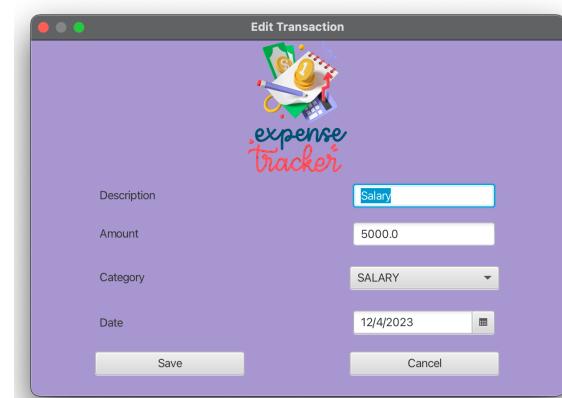


Figure 7. Edit Transactions Screen

The option to view pie charts directly from the transaction page provides users with a graphical overview of their income and expenses. This visual representation, coupled with transaction details, offers users an efficient means of understanding their financial distribution at a glance.

To facilitate ease of access to specific transactions, a search functionality is incorporated. Users can simply type the desired description, enabling quick retrieval of relevant transactions and streamlining the process of navigating through financial data.

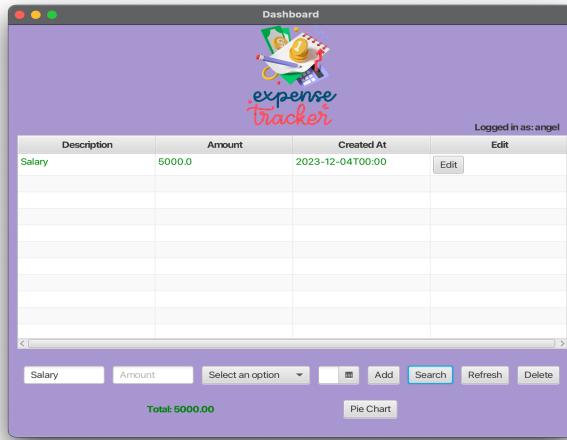


Figure 8. Searching the Transactions

On the transaction page, pertinent user information is thoughtfully displayed. The user's logged-in status is prominently showcased on the top right corner, ensuring constant awareness of their authentication status. Additionally, the total balance of all transactions is highlighted at the bottom, providing a consolidated snapshot of the user's financial position.

Behind the scenes, transaction data is intelligently stored as key-value pairs in a text file. This storage mechanism not only simplifies data retrieval and manipulation but also ensures a straightforward and portable structure for maintaining the user's financial history. In summary, the transaction page is a central hub that combines functionality, visual representation, and user-centric design, providing a comprehensive platform for efficient financial management.

#### B. Displaying the Pie Charts

Upon user input of transactions, the application dynamically generates informative visualizations, prominently featuring two pie charts that distinctly represent income and expenses. This visual breakdown provides users with an at-a-glance understanding of their financial activities. Each pie chart is meticulously designed, featuring labels and legends that reflect the amounts associated with different transaction categories. This visual representation enhances user comprehension,

allowing for a quick and intuitive assessment of their financial distribution.

To complement these visual aids, the application displays the current balance at the bottom of the page. This real-time calculation ensures users have immediate access to their overall financial standing, facilitating informed decision-making. The inclusion of this balance feature adds a practical dimension to the user interface, offering users a comprehensive snapshot of their financial position.

Ensuring the security and privacy of user information, relevant credential data is stored in a text (txt) file. This storage mechanism adheres to best practices in data management, providing a structured and secure format for handling user credentials. By combining transaction visualizations, real-time balance display, and secure data storage, the application creates a cohesive and user-centric financial tracking experience.

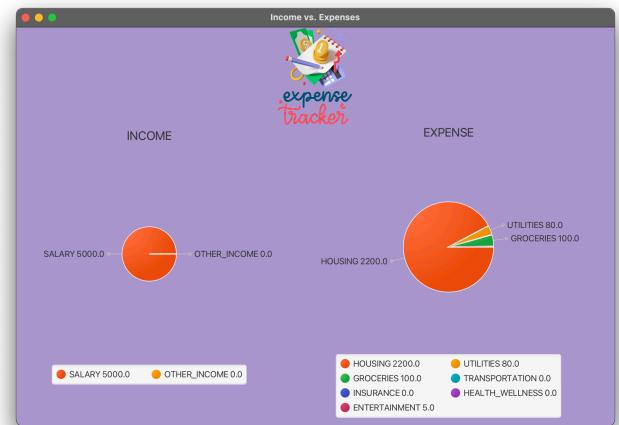


Figure 9. Displaying the Pie charts

## V. EVALUATION

This section includes screenshots of a sample run and a brief explanation of each screen.

The customer, a new user, is using our application for the first time and tries to enter their details.

Figure 10. Logging in as a new user

A Warning error pops up. This prompts them to register himself in the system.

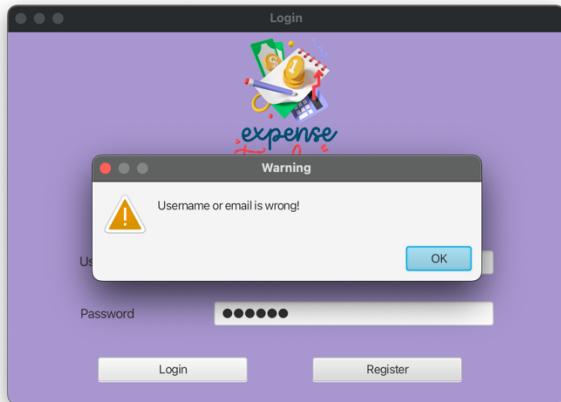


Figure 11. Warning Pop-up

The customer enters his username, password and confirmation password.

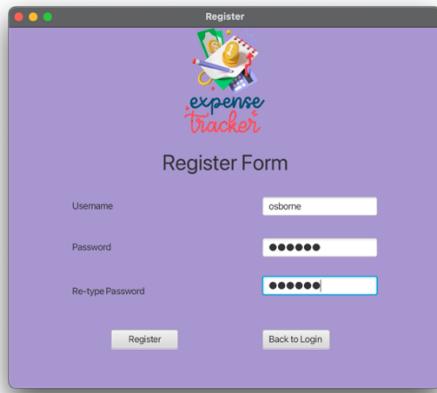


Figure 12. Registration Screen

They see the Success pop-up ensuring successful registration.

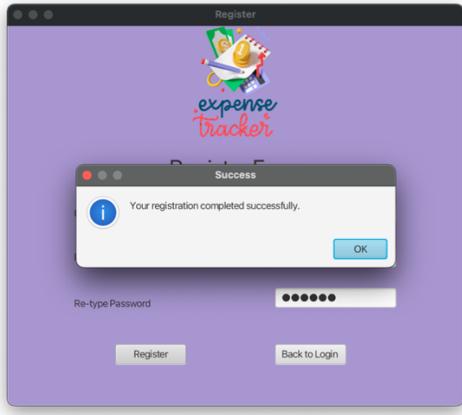


Figure 13. Successful Registration Pop-up

Following the login, the customer is able to insert the transactions. The system categorizes them as income and expenses. The balance is displayed in green reflecting positive balance.

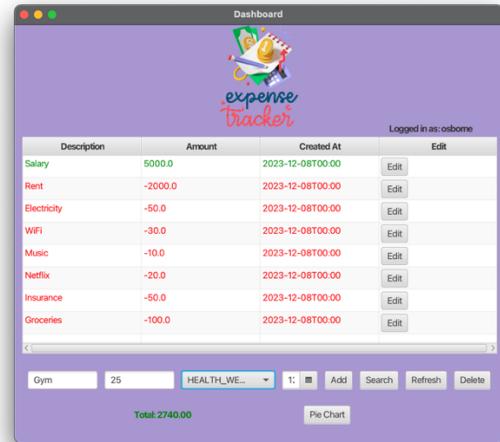


Figure 14. All Transactions

The customer wishes to search his 'Gym' expense, so he types Gym in the description and clicks 'Search'.

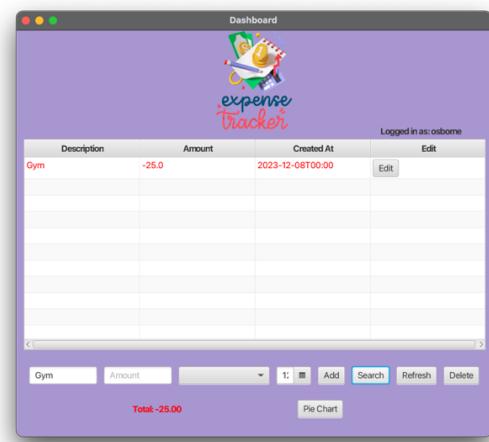


Figure 15. 'Gym' Transaction found

Now, the customer wants to change the amount of Gym from \$25 to \$50. So they change the amount and click 'Save'.

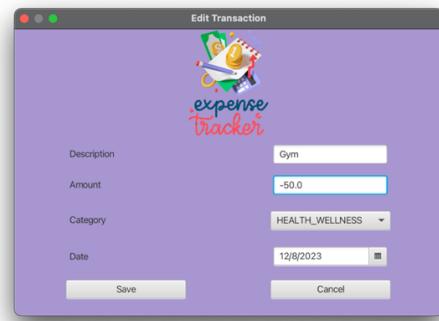


Figure 16. 'Gym' Transaction updated in Edit Transaction

Changes reflected in the table.

The screenshot shows the 'Expense Tracker' application's dashboard. At the top, there's a logo with a pencil and a calculator, and the text 'Logged in as: osbome'. Below the logo is a table with columns: Description, Amount, Created At, and Edit. The table lists various expenses: Rent (-2000.0), Electricity (-50.0), WiFi (-30.0), Music (-10.0), Netflix (-20.0), Insurance (-50.0), Groceries (-100.0), and Gym (-50.0). All entries have a 'Created At' date of 2023-12-08T00:00 and an 'Edit' button. At the bottom of the table is a search bar with fields for 'Description' and 'Amount', and buttons for 'Add', 'Search', 'Refresh', and 'Delete'. A red 'Total -2310.00' is displayed at the bottom left, and a 'Pie Chart' button is at the bottom right.

Figure 17. 'Gym' Transaction updated in Table

The customer clicks on the 'Pie Chart' button to see his incomes and expenses graphically.

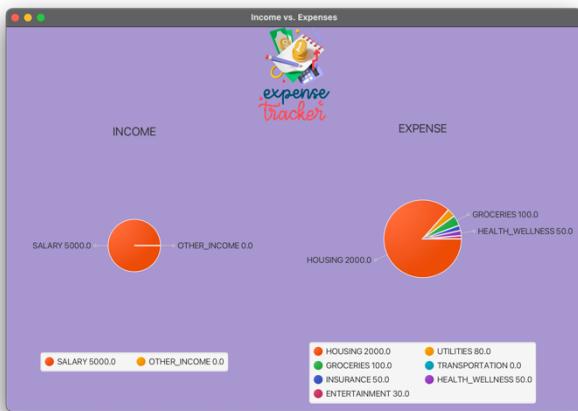


Figure 18. Income and Expense Pie Charts

## VII. DISCUSSION (REFLECTION)

This section critically examines key project outcomes, emphasizing user engagement, data accuracy, and the tool's impact on financial behavior. Security, scalability, educational contributions, and long-term sustainability are also discussed, providing a holistic view of the application's effectiveness and future prospects.

### A. User Engagement and Adoption

The success of the application hinges on user engagement. Regular use of the tool is vital for effective personal finance management. Monitoring user adoption rates and analyzing user feedback could provide insights into the application's perceived usefulness and areas for improvement.

### B. Financial Awareness and Behavior Change

The application's ability to display a visual representation of income and expenses can contribute to increased financial awareness. Tracking spending habits through pie charts may prompt users to reconsider their financial choices, fostering positive behavior change and improved financial discipline.

### C. Data Accuracy and Reliability

One key outcome to assess is the accuracy and reliability of the financial data recorded by the application. Any discrepancies or inaccuracies in transaction recording could undermine the trust users place in the tool. Regularly auditing the data against external records or user inputs can help maintain data integrity.

### D. Impact on Savings and Budgeting

The project's ultimate goal is to facilitate better money management. Monitoring the impact of the application on users' savings and budgeting practices provides valuable insights into its effectiveness. Analyzing changes in user spending patterns and the growth of savings over time can indicate whether the tool is helping users achieve their financial goals.

### E. Security and Privacy

Assessing the security measures implemented in the application is crucial. Monitoring any potential security breaches or vulnerabilities and promptly addressing them is essential. User trust is paramount in financial applications, and any compromise of sensitive data could lead to severe consequences.

### F. Scalability and Performance

As the user base grows, the application's ability to scale and handle increased loads becomes critical. Monitoring the performance of the application under different usage scenarios and identifying any bottlenecks or performance issues is essential for long-term success.

### G. Educational Impact

Assessing whether the application contributes to financial literacy and education is another important aspect. If the tool provides insights or tips on better money management, users may not only track their expenses but also gain knowledge that empowers them to make more informed financial decisions.

### H. Long-Term Sustainability

The sustainability of the application over the long term is a critical consideration. Regular updates, maintenance, and adaptation to changing technologies and user needs are necessary for the tool to remain relevant and effective.

## VIII. CONCLUSIONS AND FUTURE WORK

### What are the advantages or benefits of using your solution?

#### A. Efficiency and Automation

The solution automates the recording and categorization of incomes and expenses, significantly reducing the burden of manual data entry. This automation ensures a more efficient and convenient user experience.

#### B. Comprehensive Financial Insights

Users gain comprehensive insights into their financial health by consolidating all transactions within a single platform. This promotes a deeper understanding of their financial landscape and facilitates more informed decision-making.

#### C. User-Friendly Interface

The application prioritizes a user-friendly interface, simplifying user registration and login processes. This enhances the overall user experience, making financial tracking more accessible to a broader audience.

#### D. Personalized Financial Tracking

The application enables users to personalize their financial tracking based on individual needs and preferences. This customization ensures a tailored experience that aligns with users' unique financial goals.

#### E. Visual Representation with Pie Chart

A dynamic pie chart provides users with an intuitive visual representation of their financial data. This feature enhances accessibility and promotes a quick understanding of income and expense distribution.

#### F. Scalability and Flexibility

Leveraging object-oriented design principles, the solution is designed with scalability and flexibility in mind. Polymorphism and encapsulation enhance adaptability to evolving user needs and potential future expansions.

#### G. Empowerment through Technology

By combining robust technological frameworks, user-centric design, and intelligent automation, the Expense Tracker empowers individuals with a comprehensive and accessible tool for effective tracking and analysis of their financial resources.

### What are the problems found during the development but not yet explored in this project?

#### A. Security Enhancement

The project initially stores user credentials as key-value pairs in a text file. To enhance security, we are exploring the

implementation of encryption or more secure methods for storing and managing user credentials, preventing potential vulnerabilities associated with plain text storage.

#### B. Data Persistence and Scalability

While transaction data is currently stored in a text file, we are considering a shift to a more robust solution, such as a database, to improve data management, retrieval, and scalability. This change will contribute to a more sustainable and secure data storage approach.

#### C. Enhanced Error Handling

A comprehensive error handling mechanism is under development to provide users with meaningful error messages in the event of errors during transactions or file operations. This enhancement aims to improve the overall user experience by offering clear and informative feedback.

#### D. Multi-Factor Authentication

As part of our commitment to security, exploring the implementation of multi-factor authentication mechanisms will be helpful. This additional layer of security will further safeguard user accounts from unauthorized access.

#### E. Transaction History and Filters

Enhancing the application to provide users with a comprehensive transaction history. Also, users can filter transactions based on specific criteria, such as date range and category, to facilitate easier tracking and analysis.

#### F. Backup and Restore Feature

To prevent data loss, a backup and restore feature can be developed. Users will have the option to export their data, providing a safety net for unforeseen circumstances, and easily import it back into the application.

### If your team has more time, what do you want to improve?

Enhancing the application's scalability holds promising potential. Introducing currency conversions would alleviate challenges for frequent travelers managing expenses in various currencies. Furthermore, envisioning creative notifications, such as playful alerts signaling over-spending like, "Oh no! Your spending bucket is nearly full – time for a financial check-in!" adds a personalized touch to user engagement.

Looking forward, we contemplate expanding the project's scope by incorporating advanced features. Introducing comprehensive budget planning capabilities would empower users to plan their entire holidays effortlessly. Our software aims to provide users with accurate expense approximations, offering a seamless and proactive approach to financial management.

## **IX. JOB ASSIGNMENT**

Angel Natalina Fernandes – Login, Login Controller, Users, Register, Register Controller & its UI

Osborne Victor Lopes – Main, Transaction, Transaction Controller, Edit Transaction & its UI

Nupoor Dilip Korde – Pie Chart, Pie Chart Controller & its UI

## **REFERENCES**

- [1] Chandini, S., et al. "Online Income and Expense Tracker." International Research Journal of Engineering and Technology (IRJET) 6.3 (2019): 2395-0056.
- [2] Jadhav, Nidhi Jitendra, et al. "EXPENSE TRACKER."
- [3] Thanapal, P., et al. "Income and expense tracker." Indian Journal of Science and Technology 8.S2 (2015): 118-122.