EXPENSE TRACKER

HUSKIES_ASSEMBLE

Angel Natalina Fernandes
Osborne Victor Lopes
Nupoor Dilip Korde
CSYE 6200

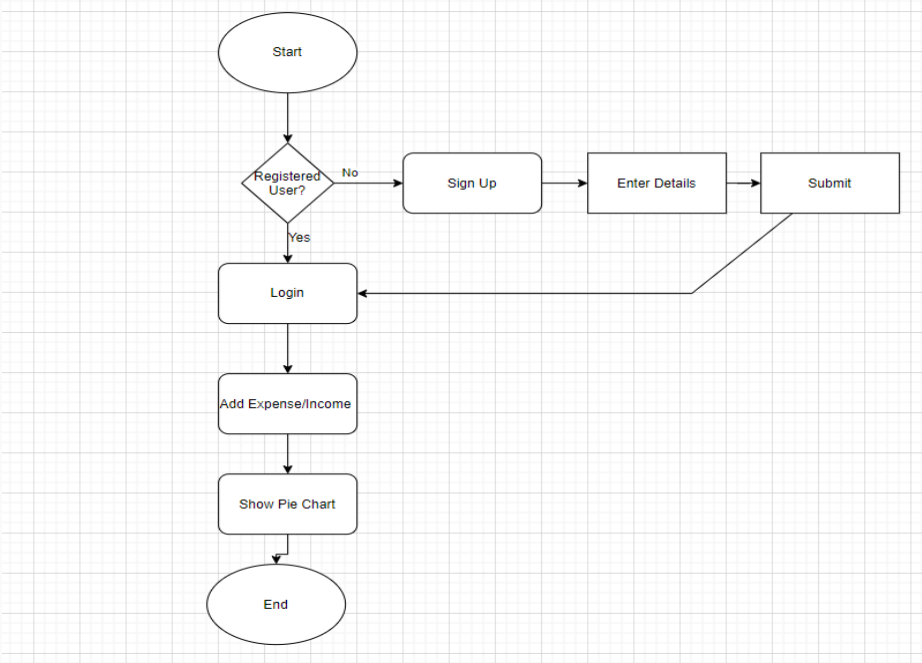
PROBLEM DESCRIPTION

- Managing personal finances is challenging without user-friendly expense tracking tools, leading to overspending and financial inefficiencies.
- Many struggle with money management due to a lack of easy tracking tools, hindering the development of good money habits.
- This not only affects individuals but also contributes to community-wide difficulties and growth limitations.
- To make it simpler, many people find managing their money a big challenge because there aren't any easy tools to keep track of their spending.
- It's tough for them to see where they spend money regularly, like on bills or other regular expenses. Because of this, it's hard for them to build good money habits and save money for things they need or want down the road.

ANALYSIS

- The aim of our project is to address challenges in personal finance management by providing a user-friendly, automated, and comprehensive solution for efficient tracking and analysis of incomes and expenses by using MVC structure.
- Users can login into their account or register themselves to utilize the service.
- The user can then enter in their income or expense description, amount and choose a category under which their description seems fit and pick the date of the transaction.
- The application will automatically categorize the transaction as income or expense and display incomes in green and expenses in red with a negative balance.
- For convenience, the balance of all transactions is displayed separately, accompanied by an illustrative pie chart for those who prefer visual representations.

UML DIAGRAM

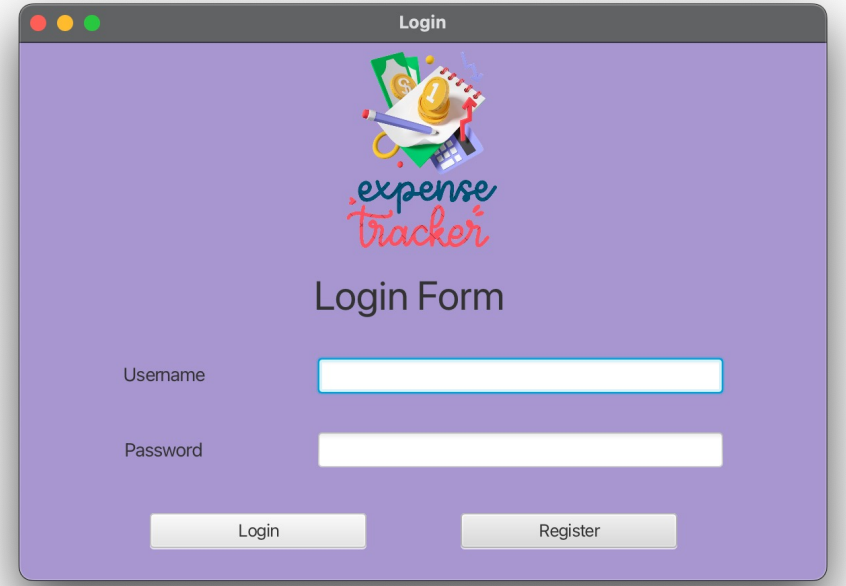

Start
Registered User?
No
Sign Up
Enter Details
Submit
Yes
Login
Add Expense/Income
Show Pie Chart
End

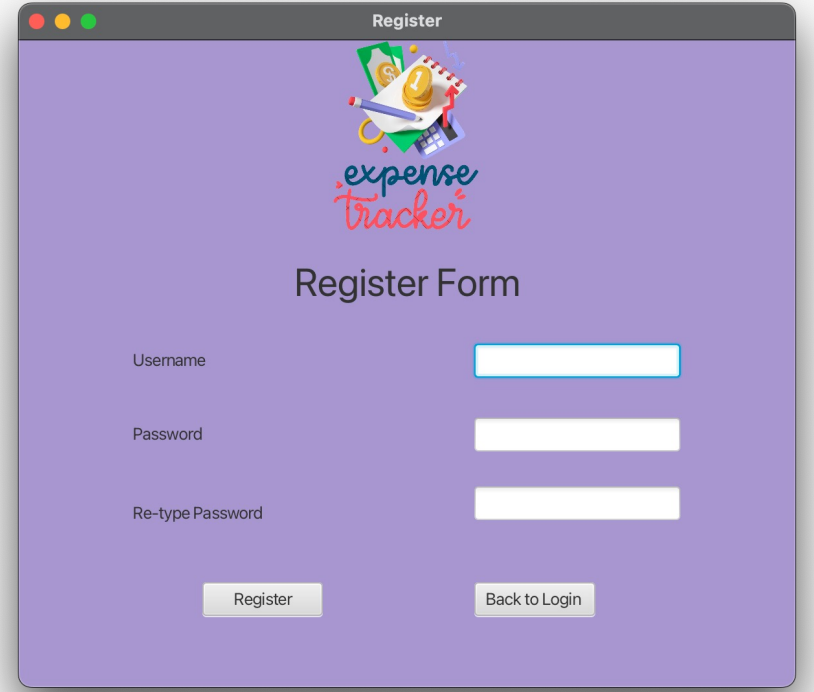
DESIGN

- The system is designed to match the feel the look and feel of a personal finance application.
- The User, userService, userController and userRepository class stores all current and new user details like automatically generated id, username, and password.
- The Transaction, transactionService, transactionController and transactionRepository class stores all the transaction details like description, amount, category and date created at.
- The editTransaction and editTransactionController updates the transactions if prompted by the user.
- The PieChartController is responsible for getting and displaying the data in the form of two pie charts, for income and expenses resp.
- A total of 5 UI pages are designed with corresponding controller classes.

IMPLEMENTATION

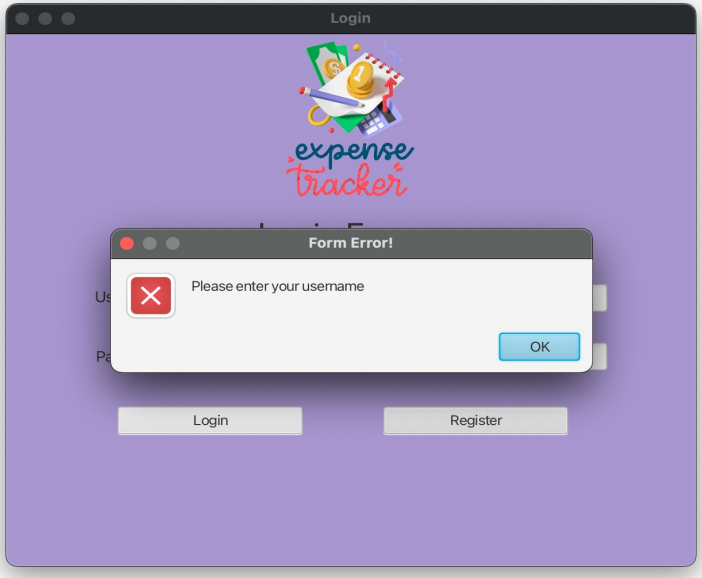
- The login screen loads as the application's first action.
- The user can either login or register with his/her credentials.

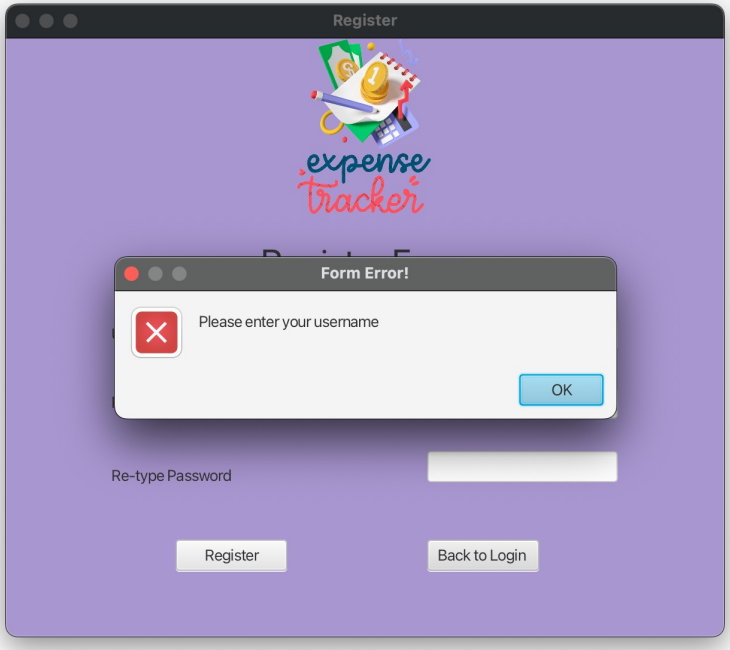




IMPLEMENTATION

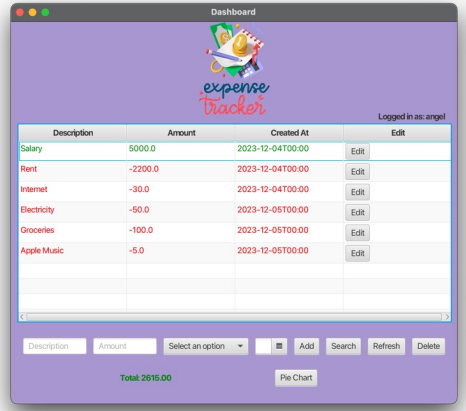
- Relevant credential data is stored as a key-value pairs in a txt file.
- If the user fails to enter any of the required authentication information, an error pops up.

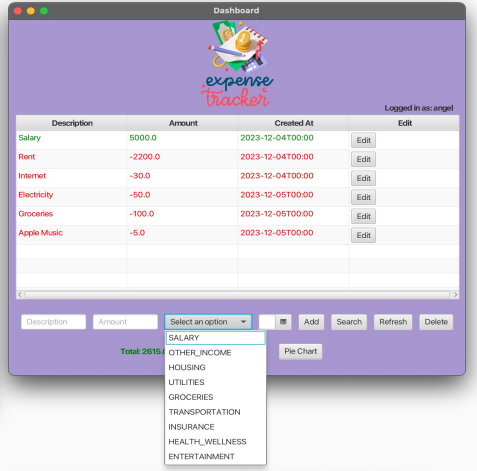


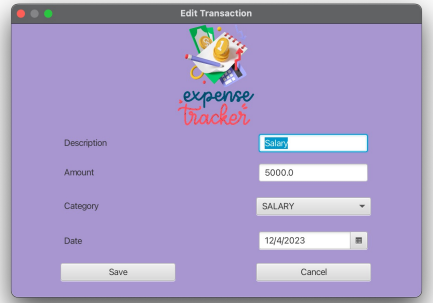


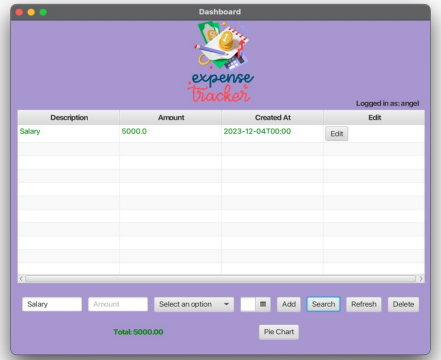
IMPLEMENTATION

- Once the user is logged in the user is directed to the transaction page where the user can see transactions.
- This UI is implemented using TableView.
- The user has the option to add, update or delete transactions.
- The user can also view the pie charts from this page.
- The user may search transactions by typing the desired description.
- User logged in is displayed on the top right and total balance is mentioned on the bottom.
- Transactions data is stored in a txt file.



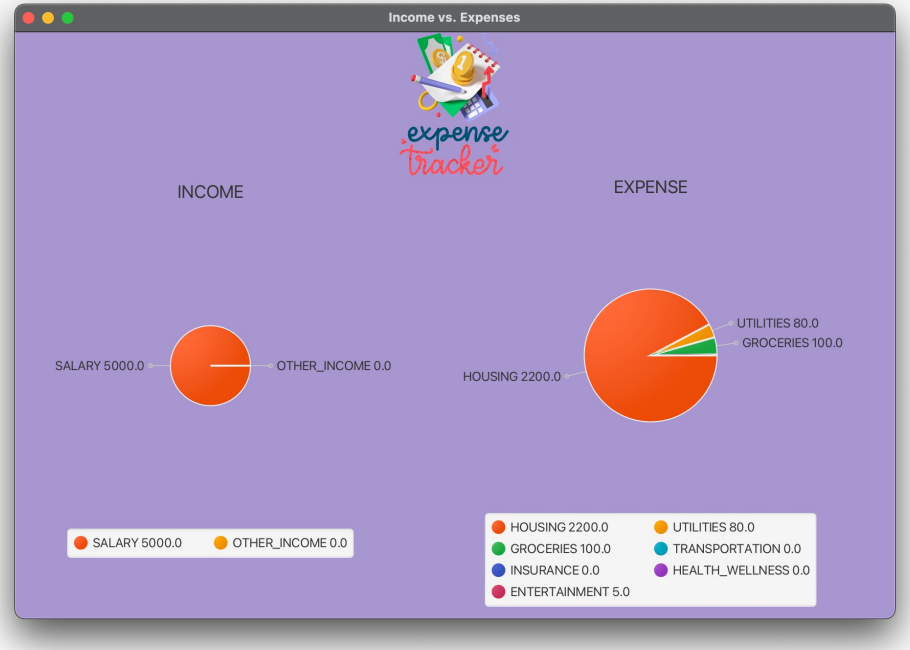






IMPLEMENTATION

- Once the user enters transactions, the pie charts can be generated, and the application displays the balance at the bottom of the page.
- 2 pie charts are displayed, for income and expenses respectively.
- The pie charts have labels and legends with amounts reflective of the transactions.
- Relevant credential data is stored in a txt file.



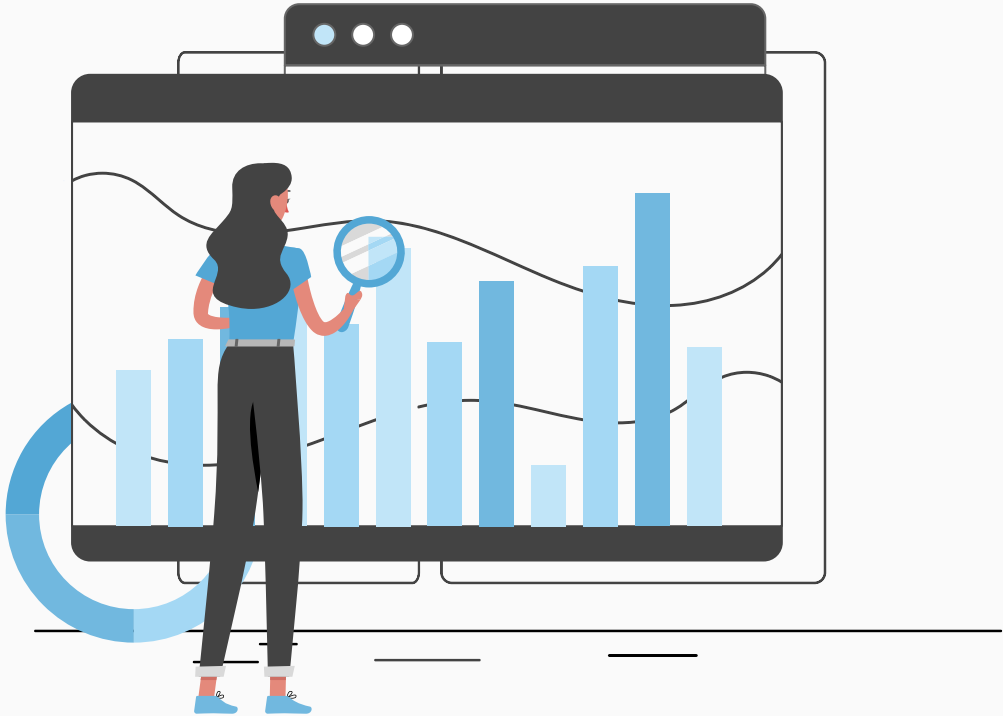
TOPICS COVERED

- Class Definition
- Inheritance/Polymorphism
- Abstract Classes/Interfaces
- Generics/Collections/Iterators
- Lists

TOOLS USED

- Eclipse
- Git repository
- Scene Builder
- Task Management Tools: Todoist, Microsoft Project, etc.
- Documentation Tools: MS Word, etc.

LIVE DEMO



FUTURE WORKS

- Currency Conversions
- Creative Notifications
- Budget Planning Features
- Collaborative Budgeting
- Gamification for Financial Goals

CONCLUSION

- An attempt at building a purposeful application to aid customers in their personal financial experience was made.
- Due to the hands-on nature of the project, we were able to learn the application of core-Java to build useful applications.
- As a team with little to some prior experience in Java programming, taking up the challenge of building a project helped in understanding the scope of Java and its reason behind a ubiquitous presence.
- Taking the object-oriented programming approach has allowed us to code in an efficient and intuitive manner, and also simplify the process of thoughtfully including different libraries in our project.

JOB ASSIGNMENT

- To ensure that everyone had the opportunity to learn while working on the project, we created the components and distributed the work as fairly as feasible.
- 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

THANK YOU