

A Personal Financial Planner Application

Team 8 | Intro. to Software Engineering - CS2043 | FCS, UNB

Roles

Project Manager- Amarjeet Singh Minhas

Progress Facilitator - Utsav Upadhyay

Documentation Lead - Hannah Chase

Technical Lead - Anush Matevosyan

Introduction

Managing personal finances is crucial, yet often overlooked. Our Java-based app simplifies financial tracking. It:

- Helps users stay on top of savings and expenses.
- Visual tools and alerts keep users informed.
- Built with simplicity, functionality, and user experience in mind.

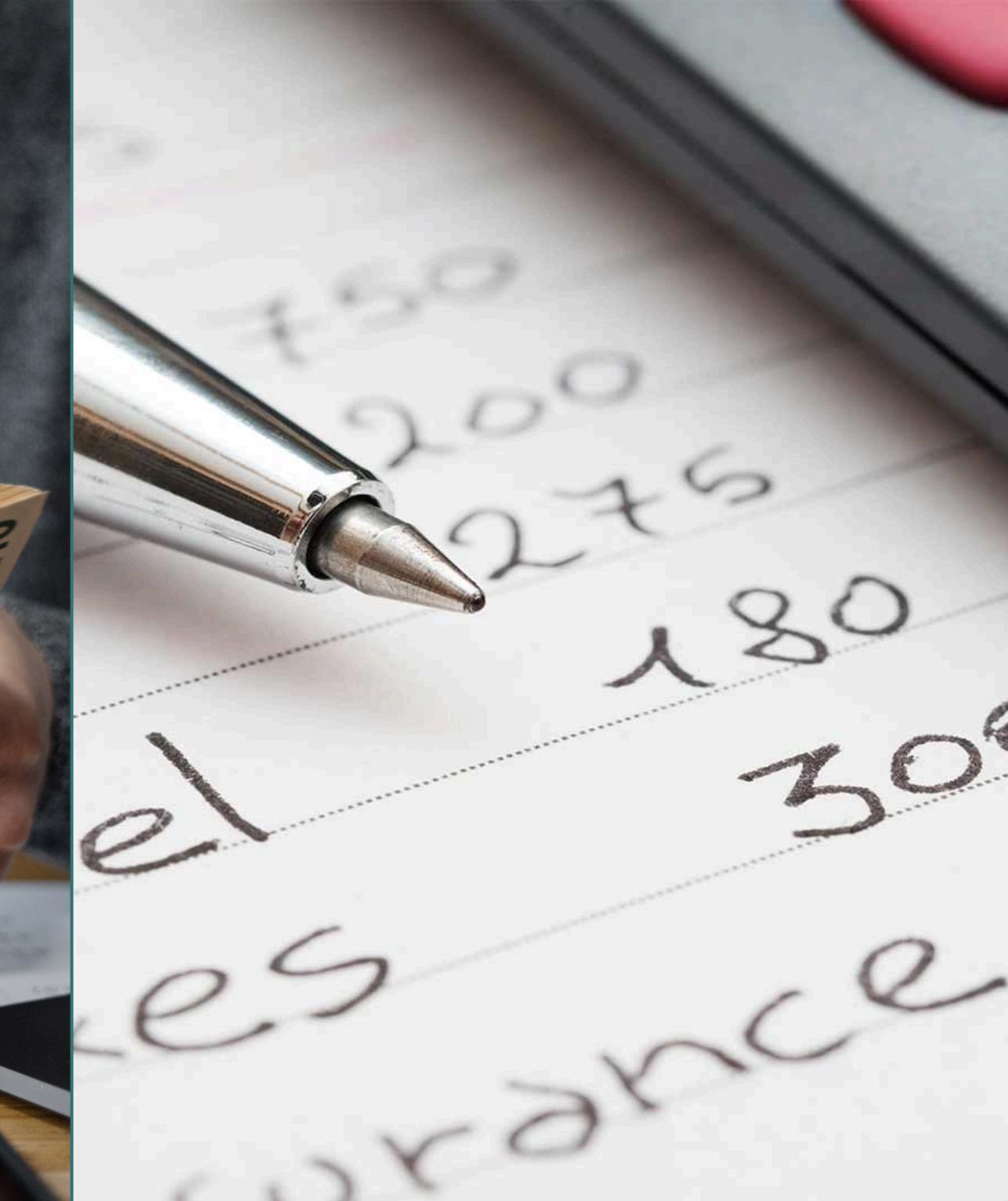


Key features



Savings Goal Setter

Set and edit your savings goals.



Expense Tracker

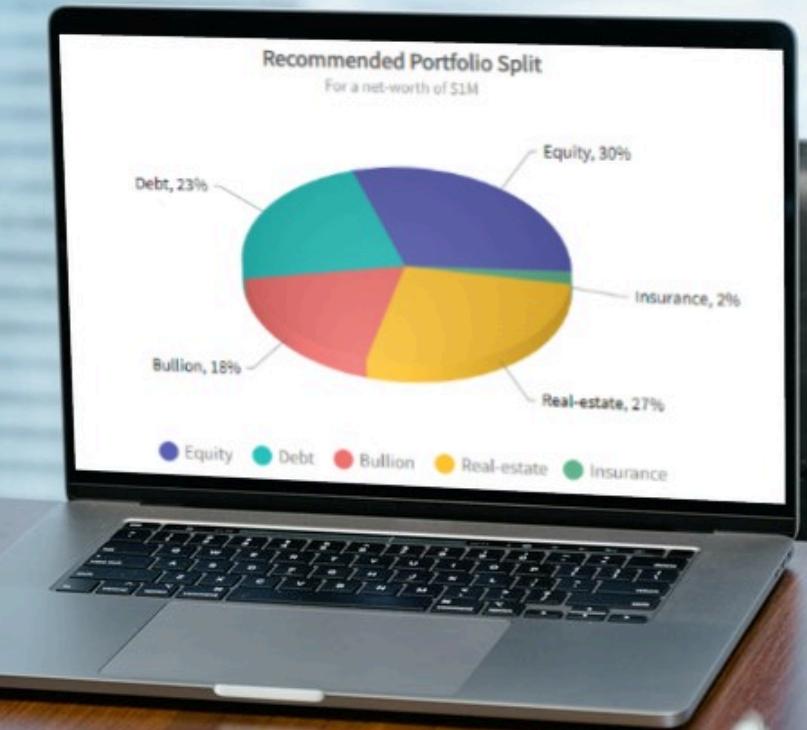
Add and categorize daily expenses.

Key features



Alert System

Notify users when they exceed their savings target.



Data Visualization

Pie chart displaying spending by category.



User Interface

Simple and interactive GUI built using JavaFX

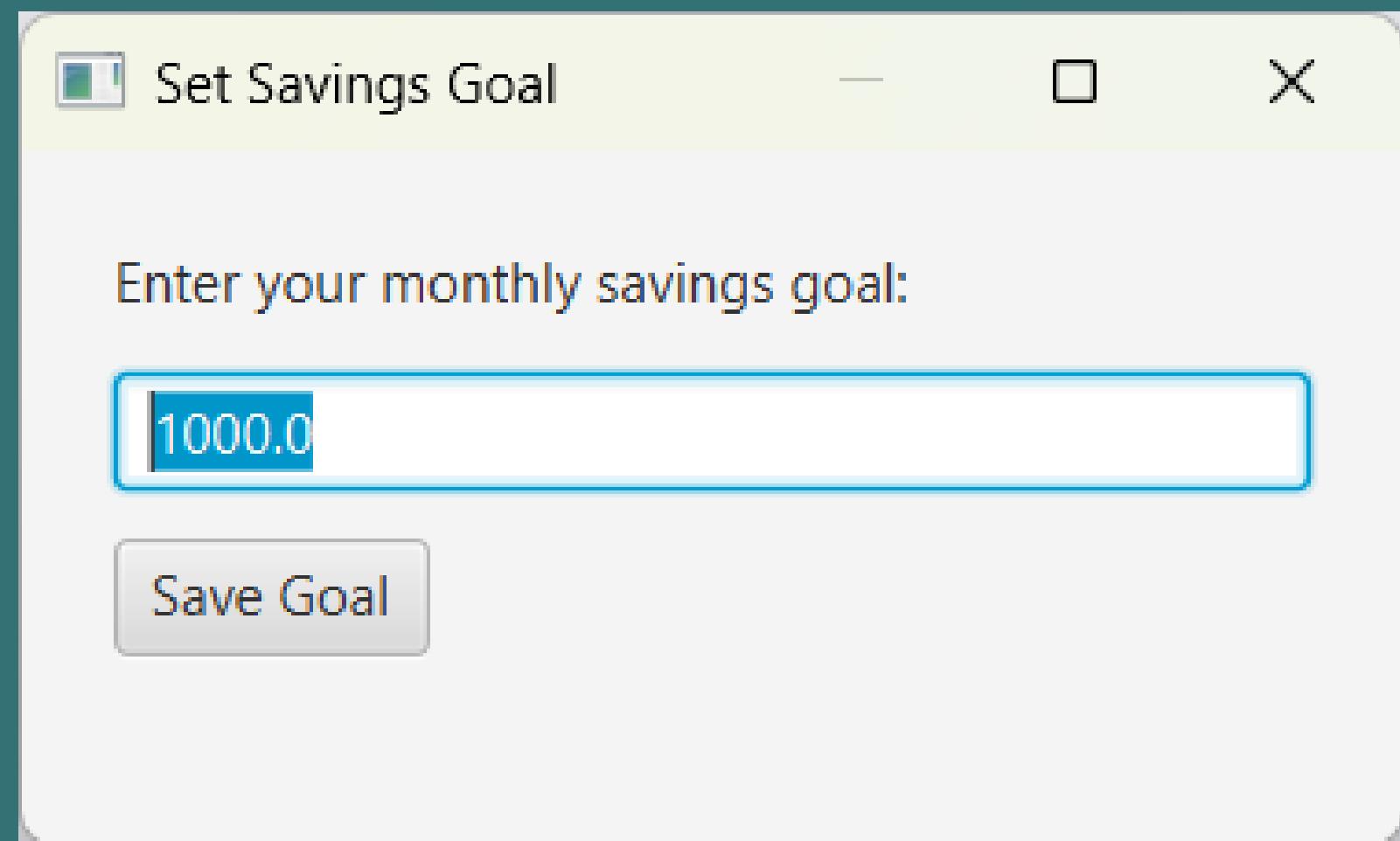
The image shows a JavaFX application interface with three main components:

- Add Entry Window:** A modal window titled "Add Entry" containing fields for "Select Date" (with a date picker), "Select Category" (with a dropdown menu), "Enter Amount" (with a text input), "Enter amount" (with a text input), "Additional Notes" (with a text input), and "Enter additional notes..." (with a text input). A "Add" button is at the bottom.
- Personal Financial Manager Window:** The main application window titled "Personal Financial Ma...". It displays a welcome message "Welcome to Your Personal Finance Manager!" and a vertical list of buttons: "Add Entry", "Set Savings Goal", "Transaction History", "Financial Report", and "My Categories".
- Pie Chart View:** A pie chart showing spending categories. The chart is divided into three segments: Rent (green), Food (orange), and Electricity (red). Below the chart is a legend with colored circles corresponding to the categories. To the right of the chart, there is text providing financial goals and current spending information.

Category	Value
Your Savings Goal:	1000.0
The Amount you Spend:	710.0
Money saved:	290.0

How it works?

Step 1: User sets a savings goal.



How it works?

Step 2: User logs expenses with category and amount.

The screenshot shows a window titled "Add Entry" with the following fields:

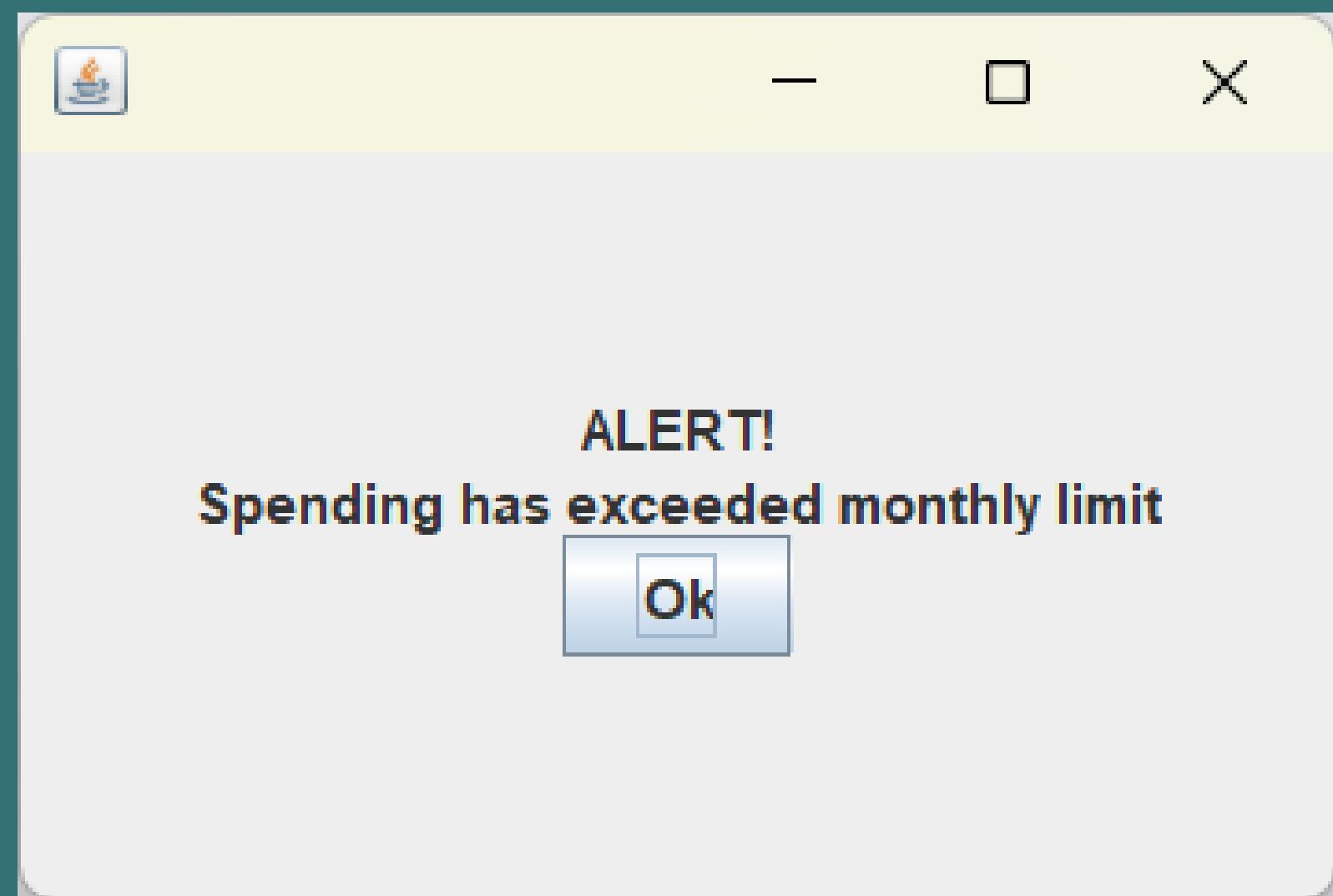
- Select Date: 2025-03-25
- Select Category: Rent
- Enter Amount: 600
- Additional Notes: Paid Rent

An "Add" button is at the bottom right.

How it works?

Step 3: System compares total expenses with limit.

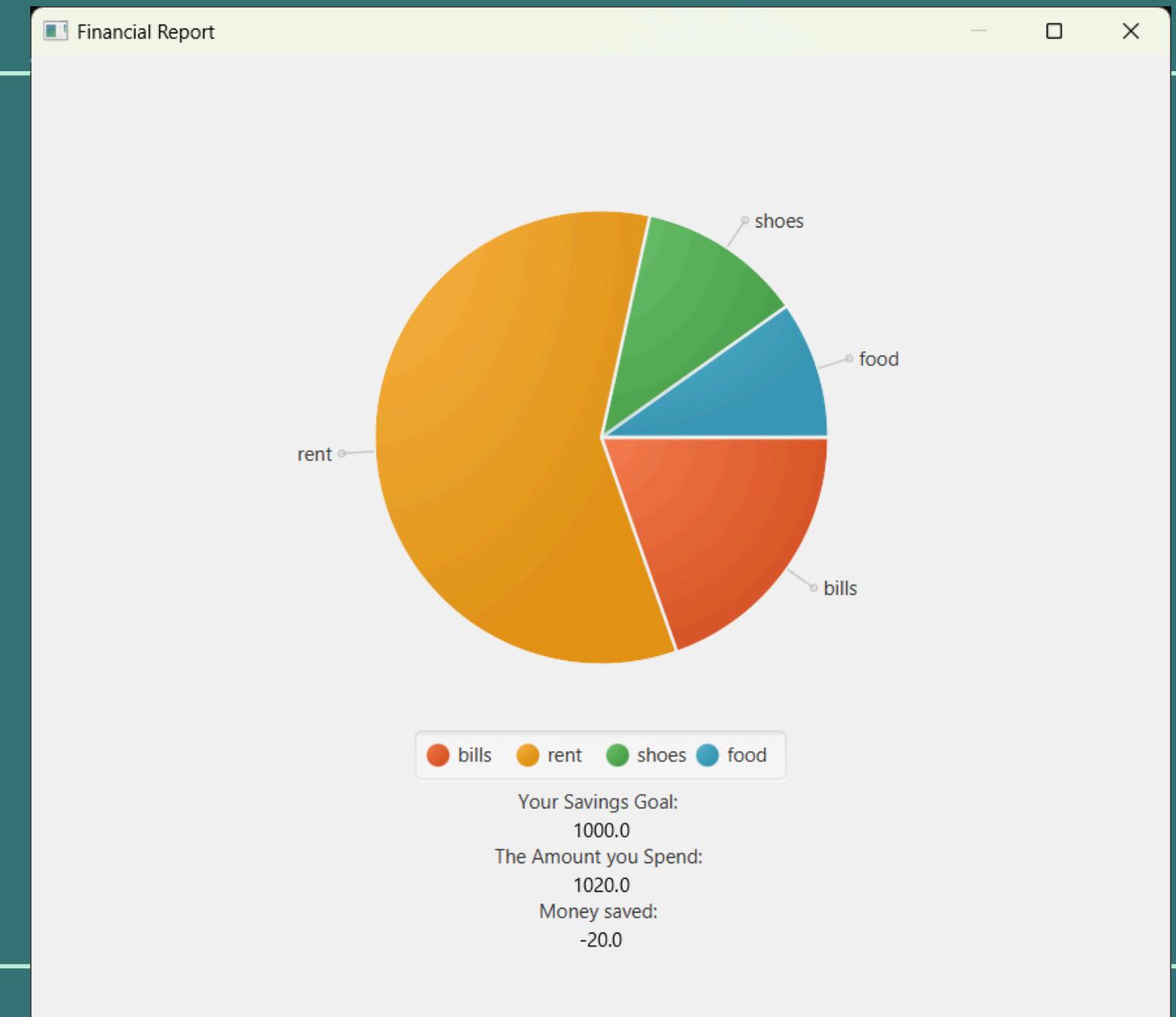
If expenses exceed savings goal → Alert is triggered.



How it works?

Additionally!

Pie chart dynamically updates with expense data.

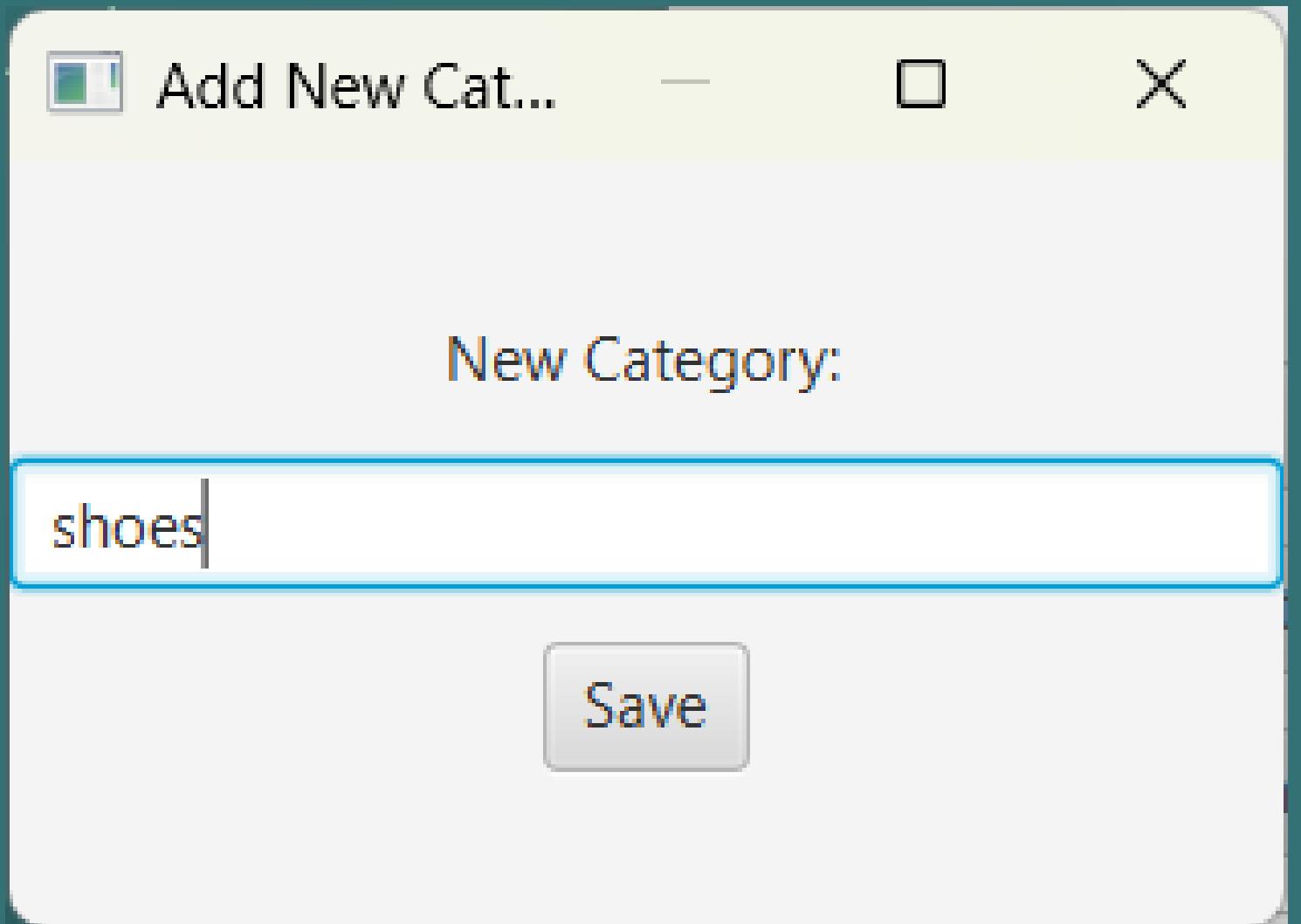


How it works?

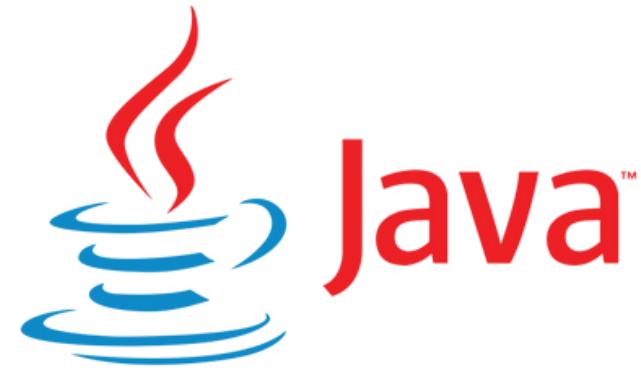
Additionally!

Users can create their own new expense categories.

That is: manage categories.



What's inside



Programming Language: Java

Core language used to build the structure of the application.



GUI Framework: JavaFX

Used to design a user-friendly graphical interface.



Data Storage: File Based

User data stored and retrieved using file I/O operations.



Charting Tool: JavaFX PieChart

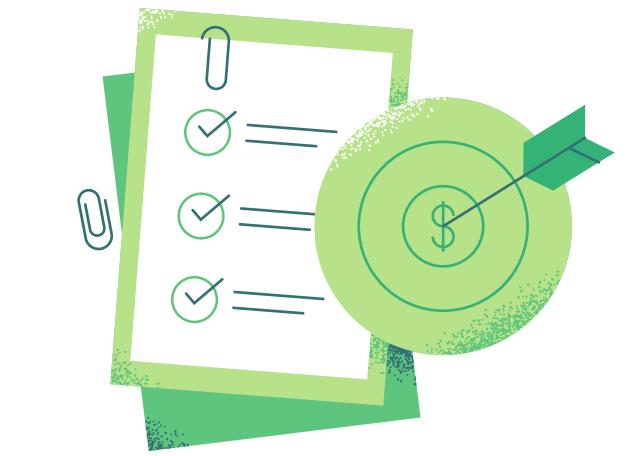
Enables dynamic visual representation of spending categories through pie charts.

Benefits to Users



Encourages savings discipline

Stay committed to your financial goals with built-in reminders and alerts.



Helps in budgeting and financial planning

Know exactly where your money goes to make smarter budget decisions.



Easy-to-use interface for all age groups

Clean, intuitive layout designed for both beginners and tech-savvy users.



Clear overview of financial habits

Visual charts and category-wise tracking reveal spending patterns at a glance.

Challenges

Easy

Creating predefined categories

Saving Data to file

Writing simple java code

Implementing a basic expense input form

Displaying a static pie chart

Difficult

Ensuring data loads correctly on restart

Allowing users to add and manage custom categories

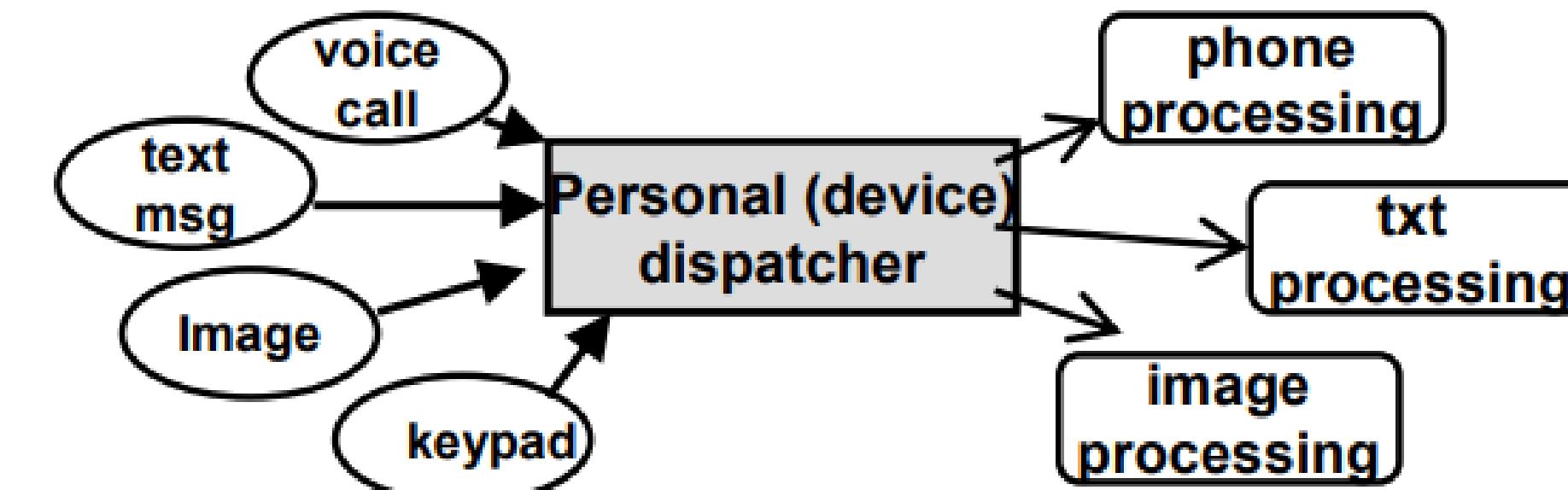
learning JavaFX

Updating the chart with every new entry

Validating user input and handling incorrect data entries

Architecture Design

Event-Driven (Real-time)



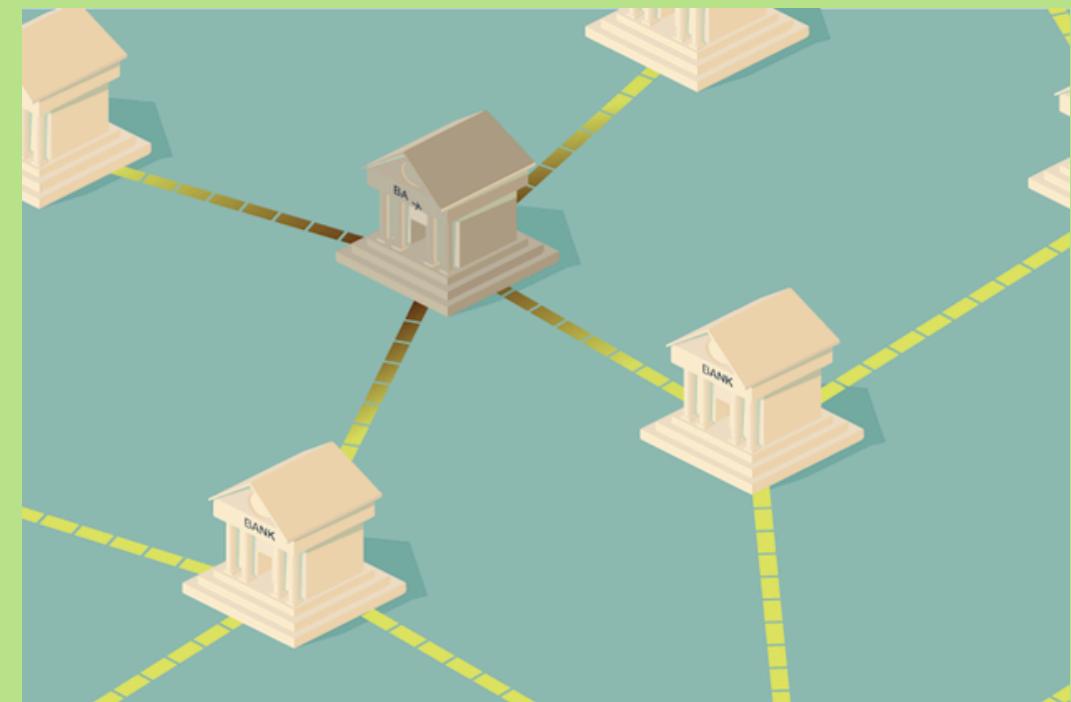
Future Enhancements



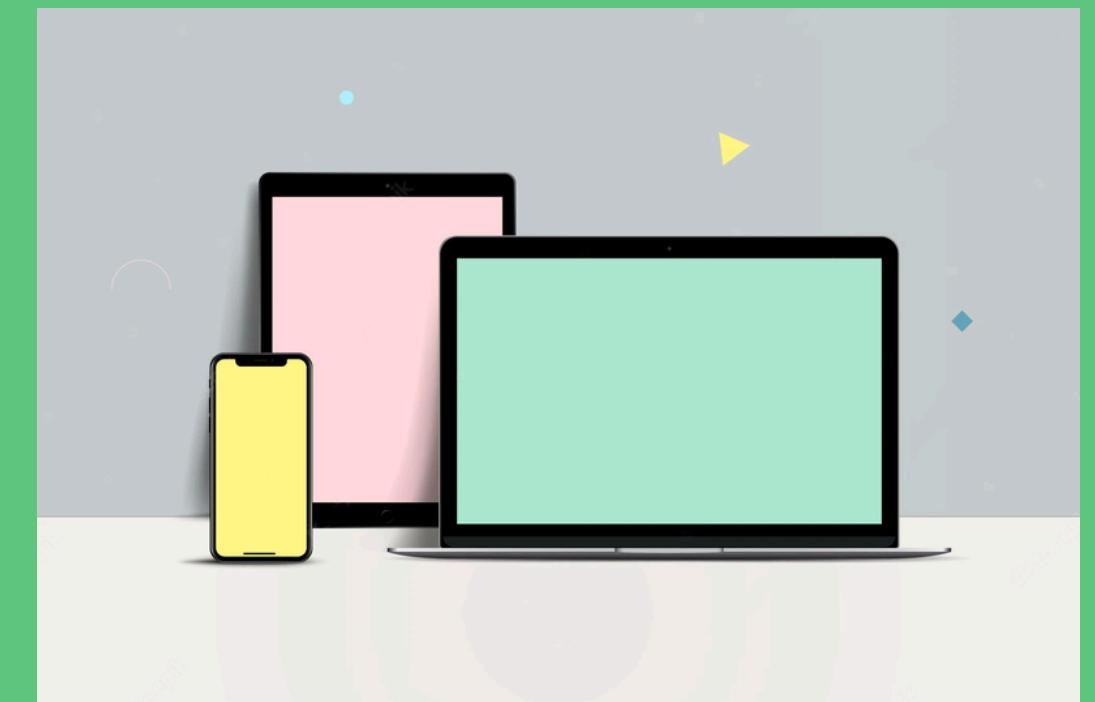
Add multi-user support with login



Integrate with bank APIs for auto-fetching expenses



Mobile version of the app



Demo

Any Questions?

Demo Video

The screenshot shows a Java IDE interface with a code editor containing Java code. The code is setting up the JavaFX module path and compiling and running a JavaFX application. The code completion feature is being demonstrated, with the word 'MainMenuItem' highlighted in red, indicating it is being completed or is part of a current selection.

```
C:\Windows\System32\cmd.exe - java --module-path "C:\Anush\jdk1.8.0_144\openjfx-17.0.8_windows-x64_bin-sdk\javafx-sdk-17.0...."
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Anush\UNB term 3\2043\team8\project>set PATH_TO_FX="C:\Anush\jdk1.8.0_144\openjfx-17.0.8_windows-x64_bin-sdk\javafx-sdk-17.0.8\lib"
C:\Anush\UNB term 3\2043\team8\project>
C:\Anush\UNB term 3\2043\team8\project>
C:\Anush\UNB term 3\2043\team8\project>javac --module-path %PATH_TO_FX% --add-modules javafx.controls *.java
C:\Anush\UNB term 3\2043\team8\project>
C:\Anush\UNB term 3\2043\team8\project>
C:\Anush\UNB term 3\2043\team8\project>
C:\Anush\UNB term 3\2043\team8\project>java --module-path %PATH_TO_FX% --add-modules javafx.controls MainMenuItem
```