International University of Business Agriculture and Technology

Project Name: MoneyExpense – Spending Tracker

Course Name : Java Programming LAB
Course Code : CSC-384
Section : A
Semester : Fall-2024

Group Name: Quad Coder

Sl No.	ID	Name	Phone	Email	Signature
01	**21303134**	**Md. Alamin**	**01822679672**	**alamin5g@yahoo.com**	
02	22203004	Tonmoy Sarker	01749240754	tonmoysarker7676@gmail.com	
03	22203032	Hasan Imam	01783174890	hasanimam72108@gmail.com	
04	23203053	Farzana Yasmin	01319961616	farzanayasmin199@gmail.com	
05	20203040	Al Amin Akash	01790770652	alaminakash550@gmail.com	
06	22203184	Mahabuba Akter Mima	01763496878	mahabubamima@gmail.com	

Overview of the Project: As a student we have to utilize (expense/cost) the money properly for the whole month. Sometimes, we are out of money at the end of last week. Because we don't know where we expense/invest our money. So, we thought we will develop a spending tracker application where we can observe in which particular categories we have invest/expense our money. Also, we can see the current uses of money for the current month. Seems it will help us to find the how much money I spent already and what were the categories.

Project Functionalities:

1. Add Category (Unique)
2. Delete Category
3. Add expense based on date, category.
4. Delete selected expense
5. Show(filter) expense based on categories
6. Show (filter) expense based on months (date wise)
7. Show current months expense (total)
8. Show all month's expense (total)

Abstract

Managing personal finances can be tricky, especially when you don't know where your money is going. As students, we've all been there—running out of cash by the end of the month without really knowing what happened. That's why we created **MoneyExpense – Spending Tracker**, a JavaFX-based app that helps track your expenses in a more organized way.

With this application, users can categorize their spending, keep an eye on monthly expenses, and visualize exactly where their money went. Features include adding categories, tracking expenses by date and category, and viewing a summary of expenses by month. The app uses the **MVC** architecture for clean and modular design and is built using **JavaFX** for an interactive interface and **MySQL** for persistent data storage, this app helps users make smarter financial decisions.

2. Table of Contents

1. Introduction

We've all faced the challenge of budgeting—especially when money seems to vanish without us even realizing where it went. As a group of students, we decided to build a simple tool that would let us track where our money is being spent. We wanted something that would show us not just the totals, but also break down expenses into categories so we could better understand our spending habits. And that's how the **MoneyExpense – Spending Tracker** was born.

The application is built with **JavaFX**, which is great for creating sleek desktop applications. The goal was to keep the UI simple, intuitive, and responsive. Users can easily add or remove expense categories, input their expenses, and view detailed reports of their spending over time.

2. Project Objective

The main goal of this project was to create a desktop application that helps users manage their personal finances by tracking monthly expenses. Some of the core functionalities include:

- Creating and deleting categories for different types of expenses.
- Adding and deleting expenses with details like date and category.
- Filtering expenses by category or by month to help users get a clear picture of their spending habits.
- Showing a summary of total expenses for the current month and over time.

This project was designed to make it easier for students—and anyone really—to stay on top of their finances and make smarter spending decisions.

3. System Design

For this project, we chose the **Model-View-Controller (MVC)** architecture to keep the code clean and maintainable:

- **Model**: Manages the data. This includes the expenses, categories, and related logic.
- **View**: The user interface, built using JavaFX's scene graph. This is what the user interacts with—buttons, tables, and input fields.
- **Controller**: Handles user actions (like adding an expense or deleting a category), updates the model, and refreshes the view accordingly.

We tried to kept the design simple:

1. **Category Management**: Users can add new expense categories or delete old ones.
2. **Expense Entry**: A form where users can enter expenses with a specific date and category.
3. **Expense View**: A table that lists all expenses and allows users to filter them by category or month.
4. **Expense Summary**: Displays the total expenses for the current month and aggregates totals for previous months.

4. Database Schema

Table Name	Field	Type	Description
category_info	category	VARCHAR(100)	Stores unique categories.
spendings	sid	INT (AUTO-INCR)	Unique ID for each spending entry.
	sdate	DATE	The date of the spending.
	category	VARCHAR(100)	The category of the spending.
	amount	INT	Amount spent.

5. Technologies Used

- **JavaFX**: This was the framework of choice for creating the application's user interface. JavaFX makes it easy to create interactive and attractive desktop apps.
- **MySQL**: A simple database to store user data and expenses. We wanted the app to remember past expenses even after closing.
- **Apache NetBeans IDE**: This helped us design the user interface visually, making the whole process a lot easier than hand-coding all the UI components.

6. Development Process

Here's a quick rundown of how we approached this project:

1. **Planning**:
   First, we brainstormed the features we needed: categories for expenses, a way to track expenses by date, and the ability to see summaries. Once we had a clear idea of what the app should do, we started designing.

2. **Design**:
   We decided on a simple yet functional UI. Using Scene Builder, we laid out the basic components—buttons, tables.

3. **Implementation**:
   We broke down the features into manageable tasks. First, we implemented the category management functionality, then followed by expense entry and filtering options.

4. **Testing**:
   After getting everything up and running, we tested the application for any bugs or issues. We made sure the adding/deleting categories worked and that the filters were displaying accurate results.

7. Features with Screenshots

1. Add/Delete Categories

Description: Users can add unique categories or delete existing ones.
Screenshots: *(Placeholder for screenshots)*

2. Add Expense

Description: Log expenses with a date, category, and amount.
Screenshots: *(Placeholder for screenshots)*

3. Filter by Category/Date Range

Description: View specific expenses filtered by category or a chosen date range.
Screenshots: *(Placeholder for screenshots)*

4. Summarize Expenses

Description: Displays total spending for the current month and cumulative totals.
Screenshots: *(Placeholder for screenshots)*

8. Results and Testing

- **Testing Approach**:
  We did manual testing by entering different types of expenses and categories to ensure everything was working. We also tested the filtering feature—making sure the app correctly filtered by category and month.
- **Results**:
  All core features were successfully implemented:
  - Expense categories can be added and deleted.
  - Expenses could be logged and displayed in a table, with filters working correctly.
  - The summary page showed the total expenses accurately.

Feature	Test Case	Result
Add/Delete Categories	Add/Delete multiple categories.	Passed.
Add Expense	Log expense with valid inputs.	Passed.
Filter by Category	Select categories from the list.	Passed.
Summarize Expenses	Show accurate totals.	Passed.

- **User Feedback**:
  Early testers found the app easy to use, and they particularly liked the ability to filter by month and category. A few suggested adding pie charts for better visualization of spending, which could be a future improvement.

9. Challenges and Solutions

- **Challenge 1: Handling Expense Filters**
  At first, filtering expenses by category or month wasn’t as easy as we thought. We resolved this by structuring the data well and using JavaFX’s Table View components to dynamically update the list of expenses based on selected filters.
- **Challenge 2: Synching UI Updates with data Changes**

  Used JavaFX's observable collections and bindings to solve the issues on Spending Table (Table View)

10. Conclusion

In the end, the **MoneyExpense – Spending Tracker** application successfully meets its goal of helping users manage their expenses. It allows students and anyone who wants to track their spending to categorize expenses, filter by month, and get a clear view of their financial situation.

We've learned a lot throughout this project—both in terms of JavaFX development and how to structure an app with MVC. If we had more time, we'd add features like graphs for better data visualization and maybe even an export option for expense reports.

11. References

- Oracle JavaFX Documentation: https://docs.oracle.com/javase/8/javafx/api/
- MySQL Documentation: https://www.w3schools.com/mysql/default.asp

12. Appendices

Contribution Table

Team Member	**Technology/Feature**
Md. Alamin	JavaFX development, UI layout design.
Tonmoy Sarker	MySQL integration, database schema design.
Hasan Imam	Expense filtering.
Farzana Yasmin	Category management, error handling.
Al Amin Akash	Report generation,
Mahabuba Akter Mima	testing, and debugging.