Faculty of Engineering and Information Technology
School of Computer Science

31927 - Applications Development with .NET
32998 - .NET Applications Development

SPRING 2023
ASSIGNMENT-2

Tut Group - Cmp1

Activity – 01

Application: Expense Tracker

Assignment Group Member:

- Name: Zhen Wang ID: 13333051
- Name: Montish Dudhatra ID: 25094804

Table of Content

Summary of project 3

Project Idea Description 3

Core Features 3

Expense Recording: 3

Expense Viewing and Searching: 3

Reports of Expense Data: 3

User Authentication: 3

Database Management: 3

Usage Instructions 4

Running App: 4

Logging Screen: 4

MainMenu Screen: 4

Development approach 5

Requirements Gathering and Analysis: 5

Database Design: 5

Back-end Development: 5

Front-end Development: 5

User Authentication: 5

Testing and Debugging: 5

Optimization: 5

Documentation and Deployment: 6

Tools and Libraries: 6

Flowchart 6

Role of team members 7

Summary of project

Project Idea Description

The Expense Tracker (Winform.NET 6) is a desktop application aimed at individuals who wish to have better control and understanding of their Daily Expense. The application provides a user-friendly interface for users to record and track their expenses over time.

Core Features

Expense Recording:

- Users can record individual expenses by entering essential details like the amount, category (e.g., groceries, entertainment, utilities), date, and optional notes.

Expense Viewing and Searching:

- Users can view a list or summary of their expenses.
- Users can search for specific expenses based on category.

Reports of Expense Data:

User can view a simple report it includes:

- Maximum cost of recording
- Minimum cost of recording
- The highest cost of category
- The lowest cost of category
- The cost of a specific category

User Authentication:

- Users can create personal accounts and password to securely manage their expenses.
- Admin account(s) for managing user accounts.
- Admin account can view the expense records of all users.

Database Management:

- A lightweight backend database using SQL Server Express to store and manage all user data.
- The database structure is designed to ensure data integrity and easy retrieval of information.
- Incorporates error handling to manage potential database errors, ensuring smooth user experience and data consistency.

Usage Instructions

Running App:

There are two ways to running the App. Firstly, User can go to the location **ExpenseTracker/ExpenseTrackerApp** Find the **ExpenseTrackerApp** in the folder where saved the project.

Optional, User can test the project in Visual Studio 2022.

Logging Screen:

User will see the login screen after loading the App successfully. User need enter the username and password to logging (user can simply press Enter key after input name and password/or using Cursor). User can also see the Admin button, is for Admin login to manage user account and expenses recording.
(for Testing purpose, here provide a User Account - UserName: **Zen** Password: **zpass**
The Admin Password is **adminpass**).

MainMenu Screen:

User will see the MainMenu screen after logging successfully. It will show the date and for buttons they are **Add Expense, View Expense, Reports and Log out**.

- Add Expense Screen:

  After clicking Add Expense, User can see a form for recording expenses. It needs to enter Expense name, category, and date, etc. Pressing Adding button after filled out form. User can press Back button to MainMenu, or simply click close will exit the application.

- View Expense Screen:

  After clicking View Expense, User can see a table of expense recording it will show Expense name, amount, date, etc. User can use filter to see a specific category expense, clear button can reset the table.

- Reports Screen:

  After clicking Reports, User can see the analytics of recording. Such as Max, Min, Avg and Total. It will show the Highest and Lowest cost by category. User can see the total expense of a specific category by use filter.

- Log out:

  After clicking Log out, it will back to login screen. User can login with other account or exit the application.

Development approach

Requirements Gathering and Analysis:

- Initial understanding and documentation of functional and non-functional requirements.
- Defining the scope and limitations of the application concerning user authentication, expense recording, viewing, and reporting.

Database Design:

- Utilized SQL Server Express for setting up a lightweight and efficient database.
- Designed a database schema to ensure data integrity and optimized for performance.

Back-end Development:

- Created a **DatabaseHandler** class to manage database interactions using Microsoft's **System.Data.SqlClient**.
- Utilized LINQ and SQL commands for data retrieval and manipulation, ensuring a clean and efficient way to handle data operations.
- Incorporated error handling to manage potential database errors, ensuring a smooth user experience and data consistency.

Front-end Development:

- Developed using WinForms in .NET 6, ensuring a modern, user-friendly interface.
- Implemented various forms (**Reports**, **Expenses**, etc.) and controls to facilitate the core features of the application, like recording expenses, viewing expenses, and generating reports.

User Authentication:

- Designed a simple authentication system allowing users to create personal accounts and log in to manage their expenses.

Testing and Debugging:

- Conducted unit testing and debugging to ensure that all features were working as expected.
- Addressed any bugs or issues that arose during the testing phase.

Optimization:

- Optimized queries and code for better performance.

- Reviewed and refactored code to adhere to best practices and coding standards.

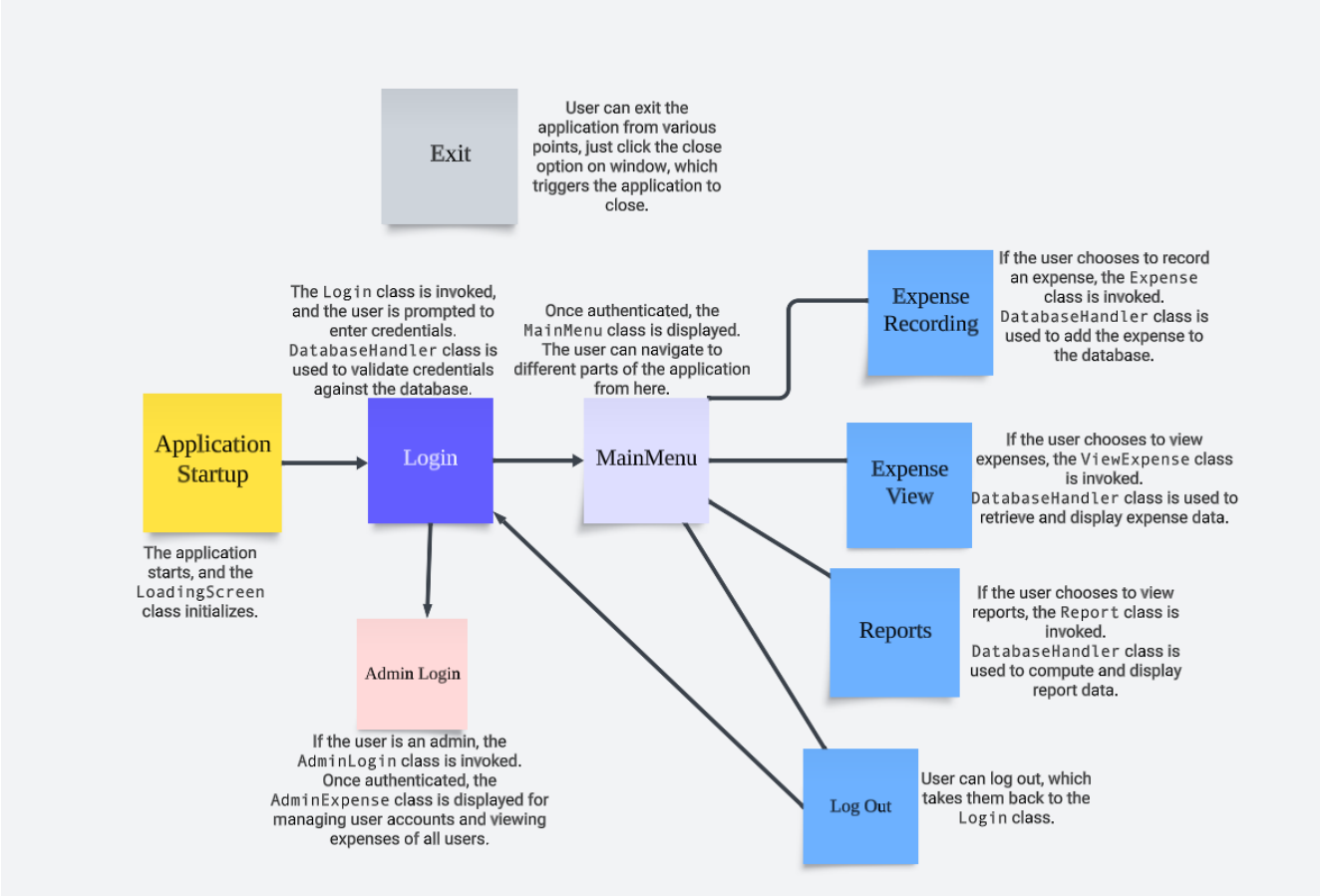
Documentation and Deployment:

- Documented code with meaningful comments explaining the purpose and functionality of classes and methods.
- Packaged the application for deployment using the Publish feature in Visual Studio 2022, ensuring it could be easily installed and run on other machines.

Tools and Libraries:

- **Visual Studio 2022:** Primary IDE for coding, testing, and debugging.
- **SQL Server Express:** Chosen for its lightweight nature and ease of setup for the backend database.
- **LINQ:** Utilized for its powerful and concise data querying capabilities.
- **.NET 6:** Modern framework offering a range of features and functionalities facilitating the development of a robust application.

Flowchart



Role of team members

Member - Montish (UI and UI Functions):

Montish holds the primary responsibility for crafting a user-friendly and visually appealing User Interface (UI) for the Expense Tracker application using WinForms in .NET 6. His role encompasses understanding user requirements and ensuring that the UI effectively meets these needs in a straightforward and engaging manner. Montish is engaged in designing the layout of various screens, creating graphical elements, and ensuring a logical flow between different segments of the application. Additionally, he manages the implementation of front-end functionalities such as form validations, user input handling, and displaying data fetched from the database. Montish also focuses on error handling at the UI level to provide users with appropriate feedback. Montish collaborates closely with Zhen to ensure seamless integration between the front-end and back-end components of the application.

Member - Zhen (Database and Function Interactions):

Zhen's role is crucial in establishing, managing, and ensuring the robustness of the database using SQL Server Express, which underpins the Expense Tracker application. Zhen is responsible for designing a database schema that not only satisfies data storage needs but also ensures data integrity and supports efficient data retrieval and manipulation using LINQ and SQL commands. This involves creating, optimizing, and managing SQL queries. Zhen also develops back-end functions that facilitate interactions between the database and the front-end. He ensures that data is accurately saved, retrieved, updated, and deleted as per the application requirements. Error handling at the database and function level is also within Zhen's domain to ensure data consistency and handle potential database errors. Collaborating with Montish, Zhen works towards ensuring that the UI components are well-integrated with the back-end services.

Role	Key Responsibilities	Components/Modules Handled	Key Deliverables
Montish	- UI Design and Implementation - Front-end Functionality - User Experience Optimization - Error Handling at UI Level - Collaboration with Zhen for Front-end and Back-end Integration	- Login Class - MainMenu Class - Expense Class - ViewExpense Class - Report Class - AdminLogin Class - AdminExpense Class	- Intuitive and visually appealing UI - Functional front-end components - Seamless navigation flow
Zhen	- Database Design and Management - Back-end Functionality - Data Integrity and Optimization - Error Handling at Database and Function Level - Collaboration with Montish for Front-end and Back-end Integration	- DatabaseHandler Class - SQL Server Express Database - LINQ Queries for Data Operations - Login Class - MainMenu Class - Expense Class - ViewExpense Class - Report Class - AdminLogin Class - AdminExpense Class	- Robust and optimized database schema - Efficient SQL queries and stored procedures - Reliable back-end functions - Effective error handling mechanisms