

OOO MINI PROJECT REPORT

Problem Statement:

Tracking and settling shared expenses, ensuring fairness and transparency among users.

Team Members:

Archit Kankaria(07):1032231900

Akshay Oswal(13):1032231923

Shayan Athma(21):1032231966

Vedant Kaskar(25):1032231979

Anshuman Patil(26):1032231980

Bill Buddy: The Splitting System

The aim of this project is to create a C++ application that enables users to efficiently manage and split shared expenses among a group. Bill Buddy assists users by categorising expenses (e.g., food, travel, utilities), managing individual contributions, and automatically calculating the amount each person owes or is owed. The application leverages C++ OOP features such as inheritance, polymorphism, exception handling, and file I/O for functionality and usability. By facilitating fair expense sharing, Bill Buddy promotes transparency and ease in group financial management.

C++ Libraries Used:

- `<iostream>`: For input and output operations.
 - `<vector>`: To store dynamic arrays of user and expense data.
 - `<string>`: For string manipulation and handling user data.
 - `<fstream>`: For file input and output operations, enabling saving and viewing of expenses.
-

Tasks Performed / Functionalities:

- **User Registration and Login:** Users can create unique accounts with a username and password. Upon registration, their details are stored in a secure manner, and they can log in to access group functionalities.
- **Group Creation and Management:** Users can create multiple groups, add group members, and switch between groups for streamlined expense tracking across different contexts.
- **Expense Addition:** Within each group, users can add expenses by specifying details like description, amount, payer, and participants. Expenses are saved and displayed, allowing for transparency in shared spending.

- **Expense Viewing and Management:** Group expenses can be viewed in detail, with each expense indicating the description, amount, and users involved, allowing for easy tracking and review.
 - **Debt Settlement:** Users can settle debts with other members within a group by specifying the payer and payee along with the amount settled, ensuring clear and fair accounting.
 - **Amount Owed Display:** The application calculates and displays a summary of amounts owed between participants in a group, allowing for easy tracking of balances.
-

- **OOO Concepts Used:**
- **Classes and Objects:** Defined `UserAccount` and `ExpenseManager` classes to encapsulate user and group data, and operations.
- **Encapsulation:** Restricted access to sensitive data (like passwords) within classes, ensuring secure handling of information.
- **Exception Handling:** Implemented error handling for user inputs and file operations to ensure robust application performance.
- **File I/O Operations:** Used file handling to store and retrieve expenses, making data persistence possible for group and expense details.

Output Screenshots:

1. **Login Functionality:** Displays the login interface for user authentication.
2. **Group Creation:** Shows the interface for creating and managing groups.
3. **Expense Addition:** Interface where users add shared expenses, detailing amount, description, and involved members.
4. **View Expenses:** Allows users to view all expenses added within a group.
5. **Debt Settlement:** Interface where users settle debts with group members.

```
=====
                                Bill Buddy
=====
1. Create Account      2. Login
3. Create Group       4. Switch Group
5. Add Expense        6. View Group Expenses
7. Settle Expenses    8. Display Total Owed
0. Exit

Enter your choice: 1

Enter your name: shayan
Enter your email: sha.com
Enter a password: abc123
Account created! User ID: 1

=====
                                Bill Buddy
=====
1. Create Account      2. Login
3. Create Group       4. Switch Group
5. Add Expense        6. View Group Expenses
7. Settle Expenses    8. Display Total Owed
0. Exit

Enter your choice: 1

Enter your name: archit
Enter your email: arch.com
Enter a password: abc1234
Account created! User ID: 2

=====
Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
=====
                                Bill Buddy
=====
1. Create Account      2. Login
3. Create Group       4. Switch Group
5. Add Expense        6. View Group Expenses
7. Settle Expenses    8. Display Total Owed
0. Exit

Enter your choice: 1

Enter your name: akshay
Enter your email: ak.com
Enter a password: abc12345
Account created! User ID: 3

=====
                                Bill Buddy
=====
1. Create Account      2. Login
3. Create Group       4. Switch Group
5. Add Expense        6. View Group Expenses
7. Settle Expenses    8. Display Total Owed
0. Exit

Enter your choice: 2

Enter your email to login: sha.com
Enter your password: ab
Login failed. Incorrect email or password.

=====
                                Bill Buddy
=====
Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
=====
1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 2

Enter your email to login: sha.com
Enter your password: abc123
Welcome, shayan!

=====
Bill Buddy
=====
1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 3

Enter group name: beach_trip
Group 'beach_trip' created and activated.

=====
Bill Buddy
=====
1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
Enter your choice: 5

Enter expense description: dinner
Enter amount: 6789
Enter number of participants: 3
Enter participant User ID: 1
Enter participant User ID: 2
Enter participant User ID: 3
Enter the User ID of the payer: 2
Expense added: dinner of amount 6789

=====
Bill Buddy
=====
1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 5

Enter expense description: hotel
Enter amount: 5000
Enter number of participants: 3
Enter participant User ID: 1
Enter participant User ID: 2
Enter participant User ID: 3
Enter the User ID of the payer: 1
Expense added: hotel of amount 5000

=====
Bill Buddy
=====
1. Create Account      2. Login

Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
5. Add Expense          6. View Group Expenses
7. Settle Expenses      8. Display Total Owed
0. Exit

Enter your choice: 5

Enter expense description: travel
Enter amount: 4875
Enter number of participants: 3
Enter participant User ID: 1
Enter participant User ID: 2
Enter participant User ID: 3
Enter the User ID of the payer: 3
Expense added: travel of amount 4875

=====
                        Bill Buddy
=====

1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 6

Expenses for group: beach_trip
Description: dinner, Amount: 6789, Payer ID: 2
Participants: 1 2 3

Description: hotel, Amount: 5000, Payer ID: 1
Participants: 1 2 3

Description: travel, Amount: 4875, Payer ID: 3
Participants: 1 2 3

Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
=====
                        Bill Buddy
=====

1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 8

Total amounts owed between participants in group: beach_trip
shayan owes 2263 to archit
shayan owes 1625 to akshay
archit owes 1666.67 to shayan
archit owes 1625 to akshay
akshay owes 1666.67 to shayan
akshay owes 2263 to archit

=====
                        Bill Buddy
=====

1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 7

Enter the User ID of the payer: 2
Enter the User ID of the payee: 1
Enter the amount to settle: 2263
shayan has settled 2263 with archit

Ln 59, Col 43  Spaces: 4  UTF-8  LF  {} C++  Mac  Q
```

```
=====
                        Bill Buddy
=====

1. Create Account      2. Login
3. Create Group        4. Switch Group
5. Add Expense         6. View Group Expenses
7. Settle Expenses     8. Display Total Owed
0. Exit

Enter your choice: 0

Thank you for using Bill Buddy. Goodbye!
```


References:

- <https://www.geeksforgeeks.org/c-classes-and-objects/>
 - https://www.w3schools.com/cpp/cpp_files.asp
 - <https://www.geeksforgeeks.org/exception-handling-c/>
-

Conclusion:

The Bill Buddy project demonstrates how C++ and OOP concepts can be used to create a practical, real-world application for group expense tracking and management. The application provides a secure platform for users to manage shared expenses, track amounts owed, and settle debts within groups. By leveraging file handling, the application ensures data persistence, allowing users to store and retrieve group expenses effectively. This project highlights the value of object-oriented design in building a structured and maintainable application, demonstrating the strengths of C++ for building reliable financial tools.