SettleUp: An Expense Splitting Web Application

[1]Faiza Sayed, [2]Sthairya Shetty, [3]Hasan Phudinawala
[1]Student, [2]Student, [3]Assistant Professor
[1,2,3]Department of Computer Science
[1,2,3]Royal College of Arts, Science and Commerce (Autonomous), Mira Road (East), India
[1]sayedfaiza2411@gmail.com, [2]sujayashetty236@gmail.com, [3]hasanphudinawal@gmail.com

Abstract—Group expense management is a frequent activity in social and domestic settings but is often handled through manual methods such as notes, spreadsheets, or group chats, leading to confusion, miscalculations, and disputes. Existing applications (e.g., Splitwise, Tricount) reduce manual effort but exhibit limitations in handling unequal splits, multiple payers, and settlement optimization. On the other hand, graph-theoretical models in financial research (e.g., DebtG, portfolio compression) provide robust approaches for large-scale debt clearing but are impractical for small group usage due to their computational complexity. This study introduces SettleUp, a lightweight web-based expense splitting system built with Django and SQLite. The system provides automated expense tracking, flexible equal and unequal splits, a transparent "Who Owes Whom" ledger, and a dummy payment simulation feature for settlement tracking. By integrating simplified graph-based approaches into an intuitive interface, the system enhances transparency, fairness, and financial accountability in small groups.

Index Terms—Expense splitting, expense management, debt settlement, financial transparency, Django, SQLite, graph algorithms, settlement optimization.

I. INTRODUCTION

Imagine a group of friends visiting a café together. One person orders from the vegetarian menu, another picks a non-vegetarian dish, a third adds a dessert, and one pays the full bill to save time. Later, when it's time to settle, the group struggles to figure out who owes what. Now think about flat mates sharing an apartment. Every month they divide rent, groceries, Wi-Fi, and electricity bills. Sometimes, one person ends up covering multiple expenses on behalf of everyone else, while others contribute in smaller ways. Or consider friends taking a cab ride together, where not everyone travels the same distance which makes cost-sharing even more complicated.

In all these everyday situations, expenses are usually managed manually through group chats, notes, or spreadsheets. While these methods may seem convenient at first, they often lead to confusion, miscalculations, delays in repayments, and unresolved debts. The lack of a clear and reliable system creates unnecessary stress and can even affect relationships.

To address these challenges, we propose an Expense Splitting Web Application which is a simple yet powerful solution that makes group expense management effortless. Instead of depending on manual calculations, the platform offers a transparent way of recording, tracking, and settling shared costs. It ensures that everyone knows their fair share, promotes accountability, and reduces the chances of disputes. By simplifying the process of splitting expenses, the application transforms something that usually causes frustration into a smooth and stress-free experience. Whether it's a casual outing with friends, managing household bills, or traveling as a group, this solution brings clarity, fairness, and peace of mind to shared financial responsibilities.

II. LITERATURE REVIEW

Graph-theoretical models have been extensively applied in financial networks to address settlement and optimization problems. Cui introduced DebtG, a weighted directed multi-arc graph to model debt relationships, successfully applied in real-world datasets [2]. Froese, Hoefer, and Wilhelmi studied the computational complexity of debt-swapping operations, demonstrating their NP-hard and PSPACE-complete nature [1]. Gavrila and Popa proposed algorithms for clearing inter-company debts, later implemented at a national scale in Romania [3]. Hanics developed portfolio compression models incorporating participant preferences to reduce redundant obligations [4]. Gazda and Vanko applied graph algorithms for mutual debt compensation, proving the efficiency of cycle cancellation in minimizing claims exposure [10].

At the consumer level, expense-splitting applications such as Splitwise and Tricount rely on greedy algorithms to minimize transactions [13]. While suitable for basic use, these systems struggle with customized splits, multi-payer cases, and transparent ledger tracking. Research on dynamic debt clearing [6] and multi-threaded cycle detection [7] provides scalable methods for updating debt graphs in real time, but these have not been widely integrated into consumer-facing applications.

Recent advancements in privacy-preserving payment splitting systems have introduced cryptographic masking and secure computation techniques to protect user identities and transaction details [9]. These systems ensure debtor and server privacy using AES encryption and arithmetic over masked vectors, bridging the gap between financial security and usability. However, their reliance on central servers and fixed transaction rounds limits scalability and responsiveness in dynamic environments.

In summary, while financial research provides robust graph-theoretical frameworks for large-scale systems, their adaptation to lightweight, user-friendly platforms for small groups remains limited. This study positions SettleUp as a bridge between these domains, combining practical usability with optimized settlement logic to enable secure, efficient, and transparent debt management for everyday users.

III. RESEARCH GAP

Although existing applications simplify group expense management, they often lack comprehensive support for multiple payer scenarios, advanced optimization of settlements, and transparent debt tracking. Advanced financial research, such as DebtG [2], portfolio compression models [4], and mutual debt compensation frameworks [10], address complex clearing problems in large financial systems using graph-based optimization. However, these solutions are computationally intensive and not designed for small-scale, everyday scenarios.

This reveals a gap between practical usability and theoretical optimization. A system that blends user-friendly interfaces with lightweight, optimized settlement algorithms remains underexplored. SettleUp addresses this gap by incorporating simplified graph-based methods into a web application tailored for small group expense management.

IV. RESEARCH OBJECTIVE

The study is guided by the following objectives:

- To develop a web-based expense management system that automates tracking and simplifies settlement in small groups.
- To implement equal and custom split mechanisms ensuring flexibility in expense distribution.
- To design a real-time ledger system that transparently records balances and dues.
- To introduce a dummy payment simulation feature for settlement verification without involving real financial transfers.
- To evaluate the system's effectiveness in terms of accuracy, usability, and efficiency compared with existing tools.
- To adapt graph-theoretical optimization techniques to improve settlement efficiency in small-scale expense sharing.

V. PROPOSED SYSTEM

System Architecture

The proposed Expense Splitting Web Application is designed using a three-tier architecture, consisting of the presentation layer, application layer, and data layer. The presentation layer serves as the frontend, developed using HTML, CSS, JavaScript, and Bootstrap, offering a clean and responsive interface where users can log in, create groups, add expenses, and view settlements through intuitive screens. The application layer, powered by Django, handles the business logic of the system, processes user requests, and manages communication between the frontend and database. It ensures accurate expense calculations, optimized settlements, and secure handling of transactions. The data layer, implemented using Django's models, defines tables and relationships for groups, members, expenses, shares, and payments. This layered separation enhances system organization, maintainability, and efficiency. Together, the architecture supports real-time updates, seamless interaction, and consistent performance across devices, allowing the system to handle multiple groups and transactions smoothly without errors or delays.

System Modules

The system is composed of several modules, each performing a distinct function while working cohesively to manage group expenses efficiently. The Models module forms the backbone of the application and defines the database schema using Django's ORM. The Group table stores essential group details such as name, description, and creation date. The Group Member table maintains records of all members associated with each group, linking user IDs to corresponding group IDs. The Expense table logs every transaction, capturing details such as payer, amount, date, category, and type of split (equal or unequal). The Expense Share table calculates and stores each member's share for accurate balance tracking, while the Payment table records settlements and dummy payments between users to maintain a clear and auditable transaction history.

The Forms module manages user input, offering structured and validated forms for creating groups, adding members, and logging expenses. This ensures that only accurate and necessary information is submitted to the backend, enhancing reliability and user experience.

The Views module handles business logic and acts as a bridge between the frontend and backend. It processes user requests, performs computations, updates the database, and sends dynamic responses. For instance, when a user adds an expense, the module automatically calculates each member's share, updates balances, and refreshes displayed data.

The Templates module consists of HTML files that render dynamic content using Django's template engine. It presents backend data in an organized, user-friendly manner, managing the layout of pages such as group dashboards, expense details, settlement views, and navigation components. Working closely with the views, it ensures real-time updates and an intuitive interface for users.

Overall, this modular and layered design ensures that the system is scalable, maintainable, and efficient, delivering an optimized expense management experience with high accuracy and usability.

VI. RESULT AND DISCUSSION

Prototype testing with a diverse group of users across various scenarios demonstrated:

- The application provides a responsive and user-friendly interface, enabling quick login, group creation, expense entry, and settlement viewing.
- It ensures accurate handling of both equal and unequal splits, with optimized settlements that reduce the number of transactions.
- The system shows stable performance across multiple groups, offering real-time updates without delays or errors.
- Built-in security features of Django safeguard data, while dummy payments allow safe settlement simulations.
- Compared to existing tools, it offers greater flexibility, supporting multi-payer scenarios and customized splits.
- The use of simplified graph-based optimization improves efficiency while keeping the system lightweight for small-scale use.
- A transparent "Who Owes Whom" ledger promotes accountability, reduces disputes, and enhances trust.
- The modular three-tier design ensures scalability and maintainability, supporting future enhancements like notifications, analytics, or payment gateway integration.

VII. CONCLUSION AND FUTURE WORKS

The proposed Expense Splitting Web Application effectively tackles the challenges faced in managing shared expenses by automating the entire process from recording contributions to calculating individual shares. Through systematic and transparent computation, the system minimizes manual errors and misunderstandings that often arise during group transactions. Its intuitive interface and accurate computation engine make it suitable for various real-life situations such as trips, shared accommodations, and group projects, ensuring fairness and clarity among users.

The system's robust three-tier architecture enhances both functionality and reliability. The separation between presentation, business logic, and database layers ensures efficient performance and secure data handling. Compared to existing expense management tools, this model offers greater flexibility by supporting both equal and unequal expense splits, as well as single and multiple payer scenarios. Furthermore, the integrated ledger system promotes transparency and accountability by maintaining a detailed transaction history that users can easily verify at any time.

In the future, this application can be further expanded to include features such as AI-based expense prediction, smart reminders for pending settlements, and integration with digital payment gateways for instant transfers. Additional modules could also provide analytics and insights into spending patterns, helping users manage their budgets more effectively. By incorporating such enhancements, the system can evolve into a comprehensive financial management platform, transforming group expense handling into a seamless, intelligent, and trustworthy experience.

Future enhancements include:

- To automate recurring expenses for regular bills and subscriptions.
- To provide mobile support for adding and checking expenses on the go.
- To send notifications for pending payments and updates in shared expenses.
- To integrate payment gateways for direct settlements (UPI, PayPal, etc.).
- To offer analytics and reports showing spending trends, top categories, and balances.
- To improve group management for easier member handling and expense tracking.
- To enhance security with two-factor authentication and secure data storage.

REFERENCES

[1] H. Froese, M. Hoefer, and L. Wilhelmi, "The complexity of debt swapping," arXiv preprint arXiv:2302.01476, Feb. 2023.

[2] H. Cui, "DebtG: A graph model for debt relationship," Information, vol. 12, no. 9, art. 347, 2021.

[3] L.-I. Gavrila and A. Popa, "Algorithm for clearing financial obligations between companies," arXiv preprint arXiv:2012.07986, Dec. 2020.

[4] M. P. Hanics, "Graph theoretical models and algorithms of portfolio compression," arXiv preprint arXiv:2212.09301, Dec. 2022.

[5] C. Kertész, "The debts clearing problem – A new approach," arXiv preprint arXiv:1111.1534, Nov. 2011.

[6] C. Boja and M. Doinea, "A multi-threading algorithm to detect and remove cycles in vertex- and arc-weighted digraph," Informatica Economica, vol. 21, no. 4, pp. 46–55, 2017.

[7] P. Kumar, et al., "Cash settlement," International Journal for Modern Trends in Science and Technology, vol. 8, no. 12, pp. 1–6, 2022.

[8] S. Hu, D. Li, J. Zhang, and X. Shen, "The power of connection: Leveraging network analysis to advance receivable financing," arXiv preprint arXiv:2009.12147, 2020.

[9] R. Gazda and J. Vanko, "An application of graph theory in the process of mutual debt compensation," Scientific Proceedings of the Faculty of Economics, Technical University of Košice, vol. 21, no. 29, pp. 66–71, 2015.

[10] Y. Yao, Settling Debts Efficiently: Zero-Sum Set Packing, Senior Thesis, Harvard University, 2017.

[11] A. Fielbaum, R. Kucharski, O. Cats, and J. Alonso-Mora, "How to split the costs among travellers sharing a ride? — Aligning system's optimum with users' equilibrium," European Journal of Operational Research, vol. 301, no. 3, pp. 956–973, 2022.

[12] T. Verhoeff, "Settling Multiple Debts Efficiently: Problems," Technical Note, Eindhoven University of Technology, 2017.

[13] Splitwise, "Splitwise algorithm analyses and debt simplification methods," Technical Article, 2019.

[14] LeetCode Community, "Settling Debts Efficiently / Minimize Cash Flow — Optimal Account Balancing Discussions," 2020.

[15] C.-K. Chau and K. Elbassioni, "Quantifying Inefficiency of Fair Cost-Sharing Mechanisms for Sharing Economy," arXiv preprint arXiv:1511.05270, 2015.