Splitsy: Implementation Report

Let’s go Splitsies! - Shared Expense Management

Aarushi Jain
University of Florida
Gainesville, FL
aarushijain@ufl.edu

Shreya Shenoy
University of Florida
Gainesville, FL
shreya.shenoy@ufl.edu

Jett Nguyen
University of Florida
Gainesville, FL
jett.nguyen@ufl.edu

Krithika Kondapalli
University of Florida
Gainesville, FL
kkondapalli@ufl.edu

Kriti Shah
University of Florida
Gainesville, FL
kriti.shah@ufl.edu

1 INTRODUCTION

Managing expenses among friends, roommates, or travel companions is a common challenge. While it’s often convenient for one person to cover costs upfront such as groceries or utilities, this convenience disappears when it comes time to settle. According to conducted interviews, some forget who owes who, lose track of exact amounts, or feel uncomfortable constantly reminding their debtor. This is corroborated by a survey done by Zelle in 2018, where about half of the general population (54%) indicated that they would follow up with friends until they received their money back for a group gift [1]. Lewis and Perry [2019] state that money management is as much about navigating relationships as it is about balancing accounts, where debts between friends are often emotionally charged interactions. For example, loans are usually paid back quickly to alleviate the social obligation that they carry [2]. Without a reliable system, these situations often lead to confusion, errors, and even strain personal relationships. Furthermore, trust and interface quality significantly influence whether users adopt and continue to use mobile payment services, demonstrating the importance of a clear and secure design in a financial application solution [3].

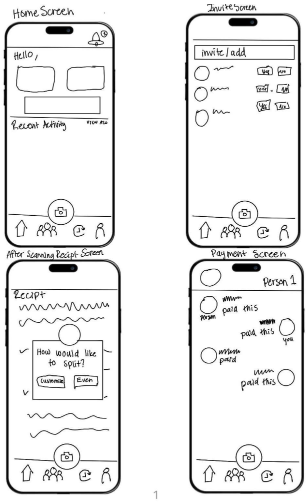
In many circumstances, people who face these challenges are college students, travel groups, and friend circles. They share costs in their everyday lives and need a method to manage splitting more efficiently. With how busy college students are and how fast-paced life can be, everyone deserves a streamlined process to track running totals, whether individual or over an extended period.

Splitsy addresses this problem by providing a clear and dependable system for tracking and settling shared expenses. Instead of relying on scattered messages or manual tracking, the app keeps balances transparent and simple to manage, which helps reduce cognitive load. Through a human-centered interface, our application emphasizes ease of use and fairness to help users maintain trust and avoid the stress that comes with managing money among friends.

2 VISION OF THE SOLUTION

Users of existing shared expense management systems often struggle with organization, ease of adding expenses, and maintaining accountability among peers for repayments, which can lead to awkward social dynamics. Quick expense addition, mobile application usage, and the ability to create groups are all features we aim to retain due to their convenience. However, Splitsy simplifies and organizes the expense management process by providing a platform for people to enter varied types of split expenses and to remind others about paying their debts. The aim is to reduce the cognitive load of calculating the splitting of items and give real-time feedback to the user as to where they stand financially by rapidly updating balances between users. The results Splitsy aims to achieve are quicker expense addition times, whether it be itemized splitting of a restaurant check or equal splitting of a monthly utilities bill. This will be achieved by maintaining consistency with familiar button layouts, applying affordance principles to guide user actions, and bridging the gulf of execution and evaluation through clear mappings and informative error messages. Users’ needs for organization, efficiency, and peer accountability will be prioritized in this vision.

3 INITIAL DESIGN



1

The initial design of Splitsy took inspiration from existing solutions like Chipp and Splitwise due to their applicable features such as expense addition and group creation. Their frontend design also allowed insight into what currently works and doesn’t work. A bottom navigation bar was the foundation of the app's features, which included a home page, groups page, a floating receipt scanning page, notifications page, and profile page. This layout was heavily influenced by Chipp's, by having a central scanning button that is elevated above the rest. Having five options was also critical to the symmetry of the app, so that the primary feature could stand out to users.

The greeting home screen was designed to provide immediate contextual awareness of what they owe and what they are owed. It included three cards within close proximity: "You're Owed," "You Owe," and "Net Balance," along with their most recent activity below it.

The groups page lists all the groups a user has made, whether that be for a family trip, grocery runs, or splitting utility bills with roommates. It allows for at-a-glance viewing of specific groups' expenses. Users are be able to invite others and share details, so that transparency and user control are the priorities.

Receipt scanning was accessible via the floating action button. When pressed, it would activate the mobile camera to scan a receipt using Tesseract OCR. Users would then be prompted to select a splitting method (even or by item) and confirm the details.

The notifications page's purpose was to provide reminders for getting paid back or paying someone back. This continuous feature would help ease the friction between relationships by automatically sending reminders.

Lastly, the profile page allows users to edit their profile, set a color theme, add payment methods, and set notification preferences. App configuration is very important to users so they can own a product that feels tailored specifically for them.

Overall, the initial design decisions attempted to apply the Gestalt principles with similar buttons for similar actions, and other integrations mentioned above. However, a design is never perfect; only near perfect, and we needed feedback from potential users to get closer to the goal.

4 PROTOTYPE EVALUATION

Our prototype evaluation underwent a structured usability testing method to ensure our design addresses real user needs. We conducted informal interviews with six of the previous interviewees from Part 1 using the think-aloud method, in which they provided feedback on the low-fidelity prototypes. Each participant was asked to describe how they would perform specific actions such as adding an expense, scanning a receipt, and reviewing payments while interacting with the prototype. We carefully observed and noted their responses as they verbalized their thought processes. Their feedback helped us identify major pain points related to usability and navigation of the application.

Low-fidelity prototypes were initially created through sketching, as seen in the initial designs. These were refined iteratively based on user feedback, and the resulting improvements were documented in the Appendix with before-and-after comparisons.

4.1 Home Screen Prototype

During evaluation, our goal was to understand what users wanted to see when they first opened the app. Most participants expressed that viewing how much they owed others and how much others owed them would be convenient. However, while the "Net Balance" card attempted to summarize this information, users struggled to interpret what it truly represented. To improve clarity, we experimented with color cues. We made it so that green represents the amount they were owed, and red indicates the amount they still need to pay back. Although this helped users quickly identify their financial standing, they still found the idea of a net balance confusing and unnecessary. Ultimately, we decided to remove the Net Balance card entirely to simplify the home screen and focus on what users found most relevant.

When asked how they would add a new expense, participants often selected the camera icon in the center of the navigation bar. However, when asked why they chose that button, most interviewees explained it was because of its central placement rather than an understanding of its purpose. The icon itself failed to communicate its function, creating a sense of hesitation. To resolve this, we analyzed navigation patterns from other expense-tracking applications. We replaced the camera icon with a universal "+" symbol, which users immediately recognized as the action for adding a new expense.

4.2 Receipt Scanning Screen Prototype

Upon describing their potential interaction with the "+" Add Expense button, users were directed to the receipt scanning low-fidelity prototype. The initial design of the receipt scanning feature had a very simple interface. Our goal was to make it easy for users to add an expense by allowing them to scan a receipt, automatically detect items, and split costs by item or percentage. However, when we presented this early design to interviewees, they mentioned that while they understood it was meant for scanning, they couldn't clearly grasp the steps that needed to be followed afterward. This lack of structure left users uncertain about how to proceed once the receipt was scanned.

Using this feedback, we redesigned the flow into a clear three-step process to provide users with a stronger sense of progress and control. The new version begins with a prominent option to either scan a receipt or manually add an expense. Next, the user selects which friends or group they want to split the expense with. Finally, the user can choose how to split the cost, either evenly or by specific items. At the end, the interface summarizes the totals and confirms how each amount is divided, ensuring clarity before submission.

This redesign significantly improved users' understanding of the process, reduced ambiguity, and made the experience feel more guided and transparent.

4.3 Activities Screen Prototype

One major change from the initial design we had implemented was redesigning the Notifications Screen to be an Activities Screen. The initial concept featured a chat-like interface that displayed each transaction as a separate message between users. Our intention was to create a more conversational and social experience, allowing users to see a record of their expense interactions. However, during testing, participants shared that while the chat format felt friendly, it would make it difficult to quickly understand their overall financial status. They would have to scroll through multiple threads to find specific payments, which would become inefficient and cluttered over time.

Based on this feedback, users emphasized the need for a more consolidated view that summarized all their transactions in one place. They wanted to see both incoming and outgoing payments together, with clear labeling of who paid whom and for what. In response, we redesigned the interface into a structured Activity Feed. Each entry now displays key details, such as the payer, participants, and total amount, in a concise card format, allowing for quicker scanning and comparison across transactions.

This change not only increased efficiency and clarity but also aligned with users' mental models of expense tracking, where data is expected to be organized chronologically and visually consistently. The new layout made it easier for users to verify transactions, track their balances, and gain confidence in how expenses were recorded and shared across groups.

4.4 Invite Screen Prototype

Initially, the invite screen was designed as a simple one-to-one "Add Friend" action, allowing users to connect only with individuals already registered on the app. This approach helped streamline the flow and maintain usability as we prioritized core functionality. However, through user interviews, we discovered that many participants envisioned using the app in more group settings, such as managing expenses among roommates, planning a vacation trip, or organizing shared events.

In response to this feedback, we reimagined the Invite Screen to support group creation. The updated design allows users to form groups within the app and have those groups saved for future use. This makes splitting expenses much faster and more intuitive, as users can easily access existing groups instead of manually adding participants each time. This redesign made the feature more adaptable to real-world scenarios while keeping the interface simple and consistent with the overall user experience.

4.5 Overall Evaluation

The prototype evaluation process allowed us to transform our initial ideas into a design that feels intuitive and natural to use. Each round of user feedback revealed small but meaningful ways to make the app clearer and more consistent, from simplifying navigation to refining visual cues and guiding users through each step. By focusing on how people actually think and interact with expense-tracking tools, we were able to create an experience that feels familiar yet efficient.

In the end, the final prototype balances simplicity with functionality. It supports users in managing expenses confidently while maintaining a clean, focused interface that reflects real-world use and genuine user needs.

5 IMPLEMENTATION[1]

The implementation of Splitsy transformed the prototype into a fully functional cross-platform mobile application developed with React Native and Expo. The system enables users to track shared expenses, scan receipts, and automatically split bills among friends or groups. The development focused on creating a responsive interface that maintained the design principles identified earlier in this paper.

The project repository is available at:

https://github.com/JettNguyen/Splitsy

The frontend was built using React Native with Expo, with styling and iconography handled by using Tailwind CSS and Expo Vector Icons. Both dark and light mode themes were implemented to enhance accessibility.

The backend was developed using Node.js and Express.js, connected to a MongoDB database for secure and scalable data storage. Authentication was implemented using JWT tokens, ensuring secure user activity and protected API access.

To support automatic expense recognition, a dedicated Python-based OCR (Optical Character Recognition) microservice was developed using Flask, Flask-CORS, Pillow, and PyTesseract. This service extracts textual data from uploaded receipts and sends processed information to the backend in a JSON format. The data within the JSON is then parsed through and connected with the MongoDB database to maintain the purchases between the users. The integration of this system allows users to capture receipt data efficiently, thereby preventing users from manually entering receipt information, making the workflow within the app efficient for users.

Implementation followed a development process using version control through Git and collaborative testing through Expo Go, allowing the team to deploy updates and verify functionality in real time. The system was tested across multiple devices to ensure consistent performance and responsiveness. The application demonstrates a visible and comprehensive realization of the design, integrating multiple technologies to create a cohesive product that aligns with the users.

6 REFLECTION

Throughout the development of Splitsy, the team encountered several challenges that shaped both our design approach and collaborative workflow. One of the most significant challenges was accurately understanding user needs and distinguishing them

[1] AI was utilized in this project to assist in setting up the initial code environment and address specific backend implementation issues.

from our own assumptions as designers and developers. In the early stages, we often projected our personal preferences onto the interface and workflow, which led to certain design decisions that did not align with actual user expectations. Conducting interviews and usability testing helped uncover discrepancies between our assumptions and real user behaviors. This process showed the importance of human-centered design and listening to user feedback. Through this, we learned that simplicity and clarity often outweigh having an abundance of features when creating a product that users can easily understand. In the future, we also intend to continue this approach to ensure that all of our decision-making processes are data-driven.

On the technical side, another major challenge was resolving merging conflicts during collaborative development. As multiple contributors worked on separate components concurrently, maintaining a consistent code structure and avoiding conflicts during integration proved difficult. This experience highlighted the importance of organized version control, establishing structured branching workflows, and conducting frequent code reviews to ensure stability. To prevent similar issues in the future, the team plans to increase transparency in development progress and establish a clearer branch structure to streamline integration with the main branch. Our next steps include further iterations of our design driven by experimental studies. We aim to continue evaluating our design decisions to ensure they improve the usability of adding and managing expenses while minimizing the gulf of execution and evaluation.

Finally, looking back on this process has taught us the importance of learning to balance feature complexity with user comprehension. The team recognized that while additional features may enhance functionality, excessive complexity can detract from usability. Moving forward, we aim to preserve the product's simplicity and ensure that each new feature directly supports user needs rather than overwhelming them. We will also continue to greatly emphasize the importance of research, evaluation, and iteration.

REFERENCES

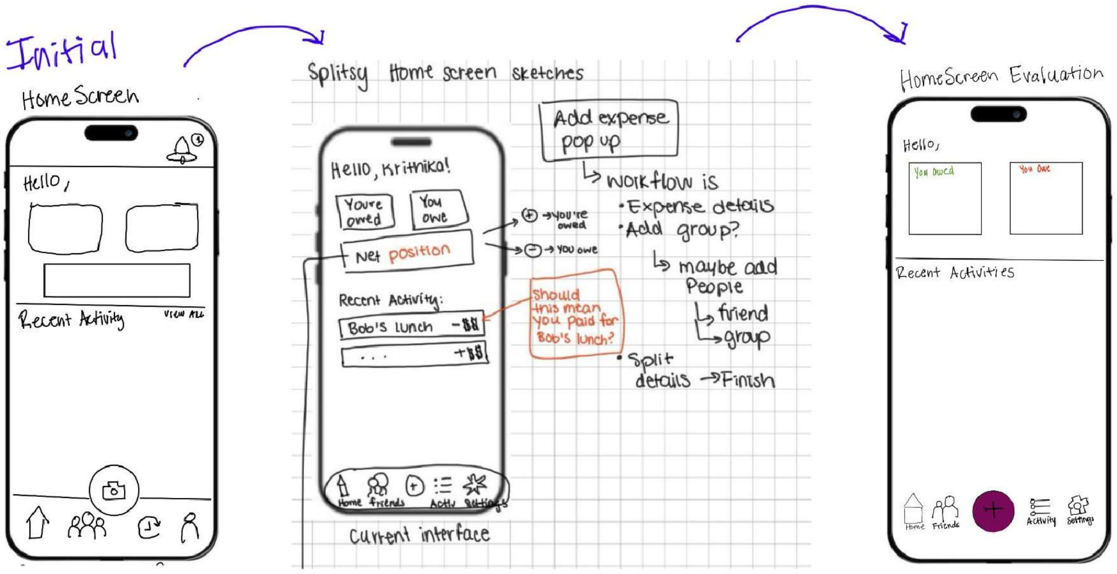
[1] Lingyuan Li, Guo Freeman, and Bart Knijnenburg. 2024. Beyond Just Money Transactions: How Digital P2P Payments (Re)shape Existing Offline Interpersonal Relationships. Proc. ACM Hum-Comput. Interact. 8, CSCW1, Article 24 (April 2024), 36 pages. https://doi.org/10.1145/3637301

[2] Makayla Lewis & Mark Perry. 2019. Follow the Money: Managing Personal Finance. In 2019 CHI Conference on Human Factors in Computing Systems Proceedings (CHI 2019), May 4-9, 2019, Glasgow, Scotland, UK. ACM, New York, NY, USA. 14 pages. https://doi.org/10.1145/3290605.3300620

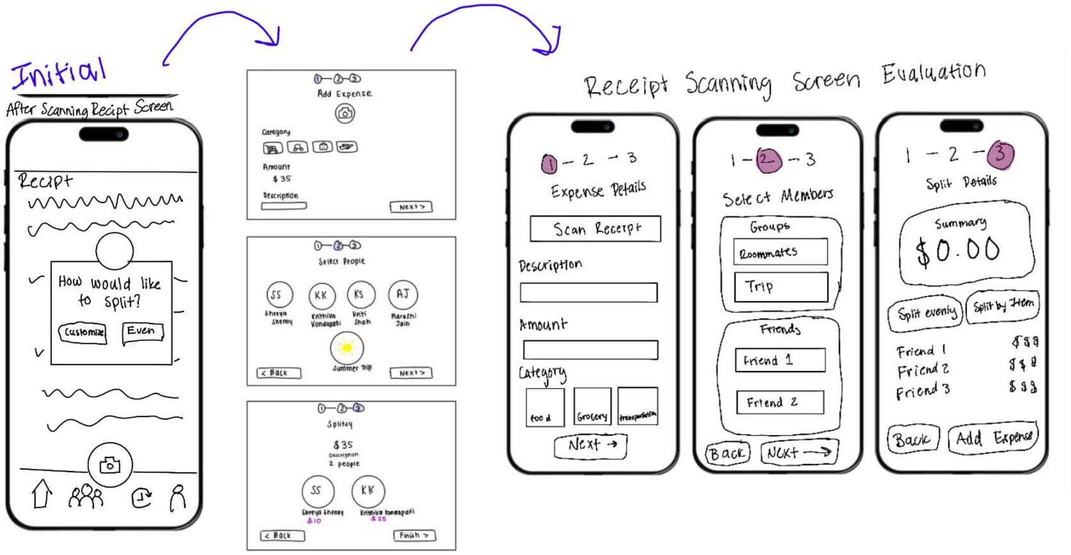
[3] Laddawan Kaewkitipong, Charlie Chen, Jiangxue Han, and Peter Ractham. 2022. Human-computer interaction (HCI) and trust factors for the continuance intention of mobile payment services. *Sustainability* 14, 21 (November 2022), 14546. http://dx.doi.org/10.3390/su142114546

APPENDIX

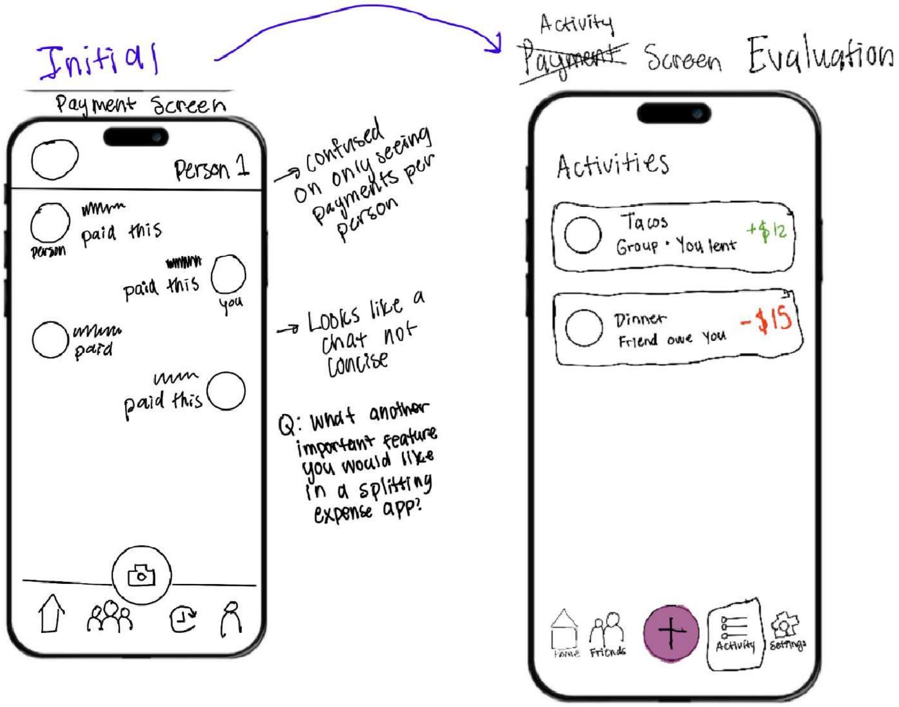
1) Home Screen Prototype Evaluation



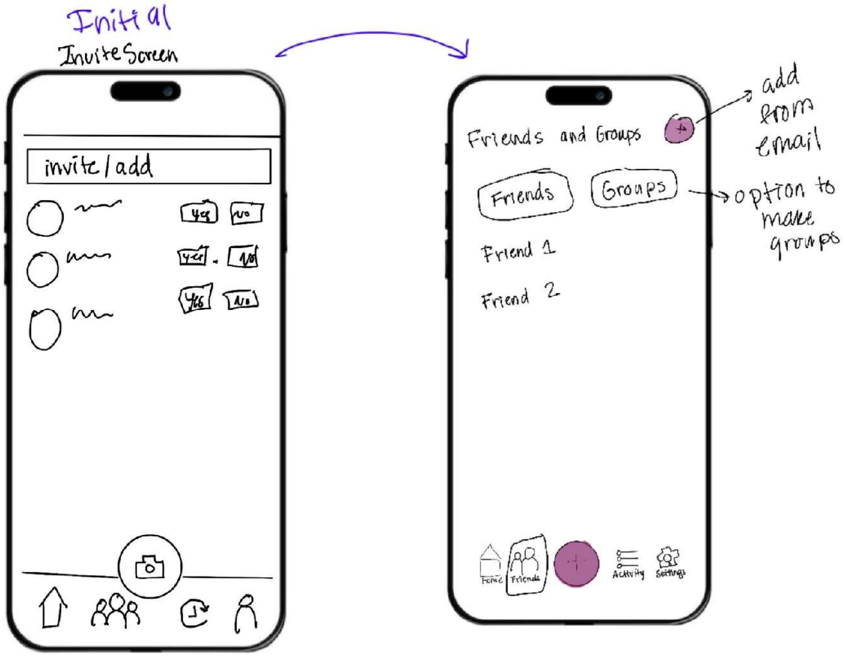
2) Receipt Scanning Screen



3) Activities Screen


Initial
Payment Screen
Person 1
person
paid this
paid this
you
paid
paid this
→ confused on only seeing payments per person
→ Looks like a chat not concise
Q: What another important feature you would like in a splitting expense app?
Activity
~~Payment~~ Screen Evaluation
Activities
Tacos
Group • You lent
+$12
Dinner
Friend owe You
-$15
Home
Friends
Activity
Settings

4) Invite Screen


Initial
InviteScreen
invite/add
yes
no
yes
no
yes
no
Friends and Groups
add from email
Friends
Groups
option to make groups
Friend 1
Friend 2
Home
Friends
Activity
Settings