Bill Buddy

By: Bug Busters

Team members : Lei Zhu, Boyuan Ge, Yuheng Xia

CONTENTS

Problem Domain

Lack of Transparency
Calculation Complexity
Limited Accessibility

Solutions

Automated Expense Sharing
Real-time balance updates
Transparent transaction history

Plans

Research and Planning
Framework Setup
Code Development
Testing
Feedback and Deployment

Team Intro

Lei Zhu
Yuheng Xia
Boyuan Ge

01 Problem Domain

- ✓ Requirements analysis
- ✓ Lack of Transparency
- ✓ Calculation Complexity
- ✓ Limited Accessibility

Requirements analysis

Managing shared expenses among groups can be frustrating and time-consuming. Whether it´s a trip, apartment rent, or a social gathering, keeping track of who paid for what and how much each person owes can lead to misunderstandings and disputes. Manual tracking (e.g., sheets or chat messages) is error-prone and inconvenient.

Lack of Transparency

People struggle to remember and fairly distribute expenses.

Calculation Complexity

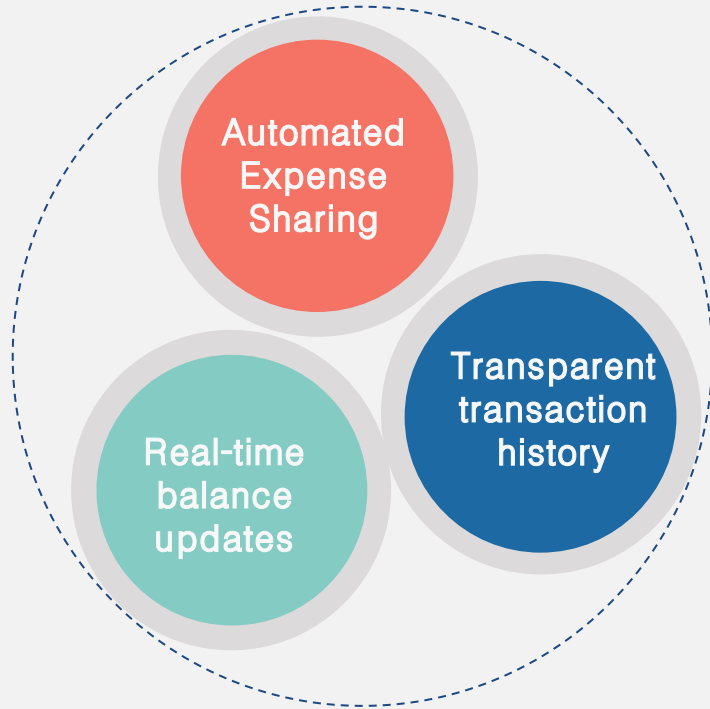Manually splitting bills for multiple transactions is tedious.

Limited Accessibility

Existing solutions often require manual entry and lack automation.

02 Solutions

✓ Automated Expense Sharing

✓ Real-time balance updates

✓ Transparent transaction history

Automated Expense Sharing

Users can record shared expenses, select participants, and have the system automatically calculate everyone's share of the expense. This eliminates manual calculations and ensures fairness.

Real-time balance updates

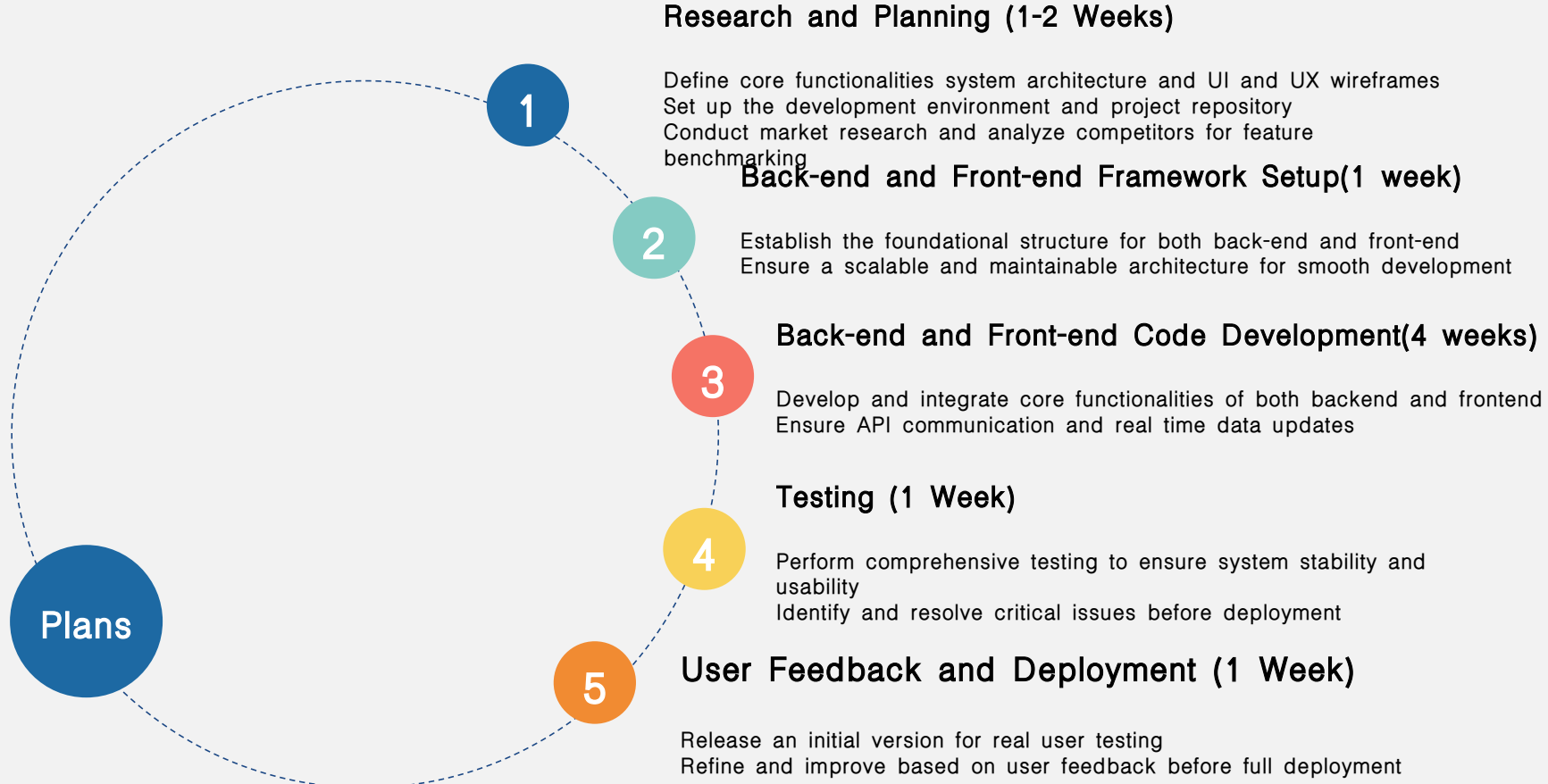Each team member can instantly see their contributions, pending payments and settlements, reducing confusion and disputes.

Transparent transaction history:

A clear record of all expenses ensures accountability and prevents misunderstandings.

03 Plans

- ✓ Research and Planning
- ✓ Back-end and Front-end Framework Setup
- ✓ Back-end and Front-end Code Development
- ✓ Testing
- ✓ User Feedback and Deployment

Plans

Research and Planning (1-2 Weeks)

Define core functionalities system architecture and UI and UX wireframes
Set up the development environment and project repository
Conduct market research and analyze competitors for feature benchmarking

Back-end and Front-end Framework Setup(1 week)

Establish the foundational structure for both back-end and front-end
Ensure a scalable and maintainable architecture for smooth development

Back-end and Front-end Code Development(4 weeks)

Develop and integrate core functionalities of both backend and frontend
Ensure API communication and real time data updates

Testing (1 Week)

Perform comprehensive testing to ensure system stability and usability
Identify and resolve critical issues before deployment

User Feedback and Deployment (1 Week)

Release an initial version for real user testing
Refine and improve based on user feedback before full deployment

04 Team Intro

Yuheng Xia

Implement key pages such as registration/login interface, team management panel, bill entry form, etc.

Develop real-time balance display components and transaction history visualization modules

Integrate back-end API to achieve dynamic data interaction (Axios/Fetch)

Lei Zhu

Design database relational model (ER diagram) to implement multi-user accounting algorithm

Develop intelligent accounting engine (supports even distribution/proportional distribution/custom rules)

Implement WebSocket real-time notification system (balance change reminder)

Boyuan Ge

Build an automated testing framework (Jest front-end testing/Mocha back-end testing)

Implement security audit (OWASP Top 10 protection, SQL injection protection)

Coordinate joint debugging of front-end and back-end interfaces to solve cross-domain issues