

Technical Specification Document

Project: Group Expense Splitter Application (Splitwiser)
Version: 1.0
Date: June 2025
Prepared by: Development Team

1. Technology Stack

Layer	Technology
Frontend	React Native (JS/TS)
Backend	Node.js (Express) or Firebase (optional hybrid)
Auth	Firebase Authentication / JWT
Database	PostgreSQL (preferred) / Firebase Firestore (fallback)
Cloud Storage	Firebase Storage / AWS S3 (for attachments)
DevOps	GitHub Actions, Bitrise, Firebase Hosting or AWS
State Mgmt	Redux / Zustand
Realtime	WebSockets / Firebase Realtime DB

2. Services Overview

2.1Authentication Service

- **Tech:** Firebase Auth or custom JWT
- **Features:**
 - Email/password sign-up/login •
 - Token-based session (JWT)
 - Password reset
 - Google login (optional)

- **DB Table(s):**

- Users
-

2.2 User Service

- **Features:**

- Read/Update/Delete user preferences (currency, image, state)

- **DB Table(s):**

- Users
-

2.3 Group Service (with optional role-based access)

- **Features:**

- Create/Join/Leave Group
- Update Group settings
- Manage Invites

- **DB Table(s):**

- Groups
 - GroupMembers
-

2.4 Expense Management Service

- **Features:**

- CRUD expenses
 - Advanced splitting logic
 - Expense simplification algorithm (graph-based) •
- Maintain change history

- **DB Table(s):**

- Expenses
 - Settlements
 - ExpenseComments (optional)
 - ExpenseHistory (optional)
-

2.5 Settlement Service (Optional Phase)

- **Features:**

- Record and view payments •
- Settle debts partially or fully

- Reminders
- Notifications
- **DB Table(s):**
- Settlements

2.6 Analytics Service (Optional Phase)

- **Features:**
 - Visual financial summaries
 - Per-user and per-group analytics
 - Graphs and trends
- **DB Table(s):**
 - Derived or materialised views on core tables

3. Database Schema

3.1 Users

Column	Type	Description
id (PK)	UUID	Unique user identifier
name	TEXT	User's full name
email	TEXT	Unique email
password_hash	TEXT	Hashed password
image_url	TEXT	Profile picture URL
currency	TEXT	Preferred currency
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.2 Groups

Column	Type	Description
id (PK)	UUID	Group ID
name	TEXT	Group name
currency	TEXT	Group currency
image_url	TEXT	Group picture
created_by	UUID (FK)	Linked to Users(id)
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.3 GroupMembers

Column	Type	Description
group_id (FK)	UUID	Linked to Groups(id)
user_id (FK)	UUID	Linked to Users(id)
role	TEXT	Optional: admin/member
joined_at	TIMESTAMP	
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.4 Expenses

Column	Type	Description
id (PK)	UUID	Expense ID
group_id (FK)	UUID	Linked to Groups(id)
created_by	UUID	Linked to Users(id)
description	TEXT	Expense description
amount	NUMERIC	Total amount
metadata	JSONB	Splits, tags, receipts
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.5 Settlements

Column	Type	Description
id (PK)	UUID	Settlement ID
expense_id (FK)	UUID	Linked to Expenses(id)
payer_id (FK)	UUID	Who paid
payee_id (FK)	UUID	Who received
amount	NUMERIC	Settlement amount
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.6 Optional Tables

3.6.1 ExpenseComments

Column	Type	Description
id (PK)	UUID	Comment ID
expense_id(FK)	UUID	Linked to Expenses(id)
user_id (FK)	UUID	Author

Column	Type	Description
content	TEXT	Comment text
created_at	TIMESTAMP	
updated_at	TIMESTAMP	

3.6.2 ExpenseHistory

Column	Type	Description
id (PK)	UUID	Record ID
expense_id(FK)	UUID	Linked to Expenses(id)
user_id (FK)	UUID	Who edited
before_data	JSONB	Snapshot of previous data
edited_at	TIMESTAMP	

4. Notifications (Nice to Have)

- Service via Firebase Cloud Messaging (FCM) •

Triggered on:

- Expense added
- Comment added •
- Settlement made •
- Group invite

5. Future-Ready Features (Phase 2+)

- OCR for receipts
- Recurring expenses
- Multi-currency exchange APIs •

Group chat

- Payment integrations (UPI/PayPal)

