

## Objective

Develop a mobile application (Android) that helps groups of people to track and manage their shared expenses, ensuring everyone is aware of their financial obligations with a friendly and intuitive user interface. If you're familiar with the application WeShare, the concept is very much the same.

## Background

In various social and formal gatherings, expenses often occur that are typically shared among participants, such as meals, gifts, or travel costs. Tracking who paid what and calculating how much others owe can become a cumbersome task. A mobile application that alleviates this burden by calculating and notifying members of their respective dues can be incredibly beneficial.

## Functional Requirements

### User Registration & Profile Management:

- Allow users to create a profile, including essential information like name and contact details.
- Enable users to update their profiles and manage notification settings.
- You can manage users by creating a custom authentication system, using a third-party provider such as Auth0 or by using dummy data and mocking interactions. **(Database and security is not required, but is recommended)**

### Group Management:

- Allow the creation of groups, adding and removing participants.
- Permit the naming and description of the group.

### Expense Entry & Management:

- Enable users to input expenses, specifying the payer and the amount.
- Allow attachment of receipts or invoices (optional feature).
- Facilitate the splitting of expenses among all or selected group members.

### Debt Calculation:

- Implement algorithms to calculate how much each person owes or is owed in a group.
- Allow users to view a summary of all transactions, and their current balance (owed/owing) within a group.

### Notifications & Communications:

- Notify group members of new expenses and updates.
- Allow users to send reminder notifications to those who owe them money.

## Non-functional Requirements

### Reliability (Quality Attribute, QA):

- Provide contextual feedback on successful and failed transactions
- If a brief network outage of the device results in failed request, notify the user, and retry request in the background without affecting the user's continued usage of the app
- User gets notified when a request has finally been completed successfully in the background
- Time between retries in the background is carried out with decreasing frequency

## Deliverables

Design document (3 pages max) detailing:

- FR/NFR
- Navigation model
- Low-fi wireframe

Implementation document detailing:

- A 4-page max group report discussing the implementation of the application
  - Your report should contain a diagram of the architecture of your project targeting to deal with reliability.
  - Your report should also cover the testing of the reliability QA
- A short video demonstrating the application.
- Link to group GitHub repository

**Submission Deadline on Itslearning: 24 November**

Competition among three best groups: 27 November