THE ANDROID PROJECT

SHARED EXPENSES MOBILE APPLICATION

Teacher:
Mahyar T. Moghaddam
(*mtmo@mmmi.sdu.dk*)

Instructors:
Joakim Leed (leed@mmmi.sdu.dk)

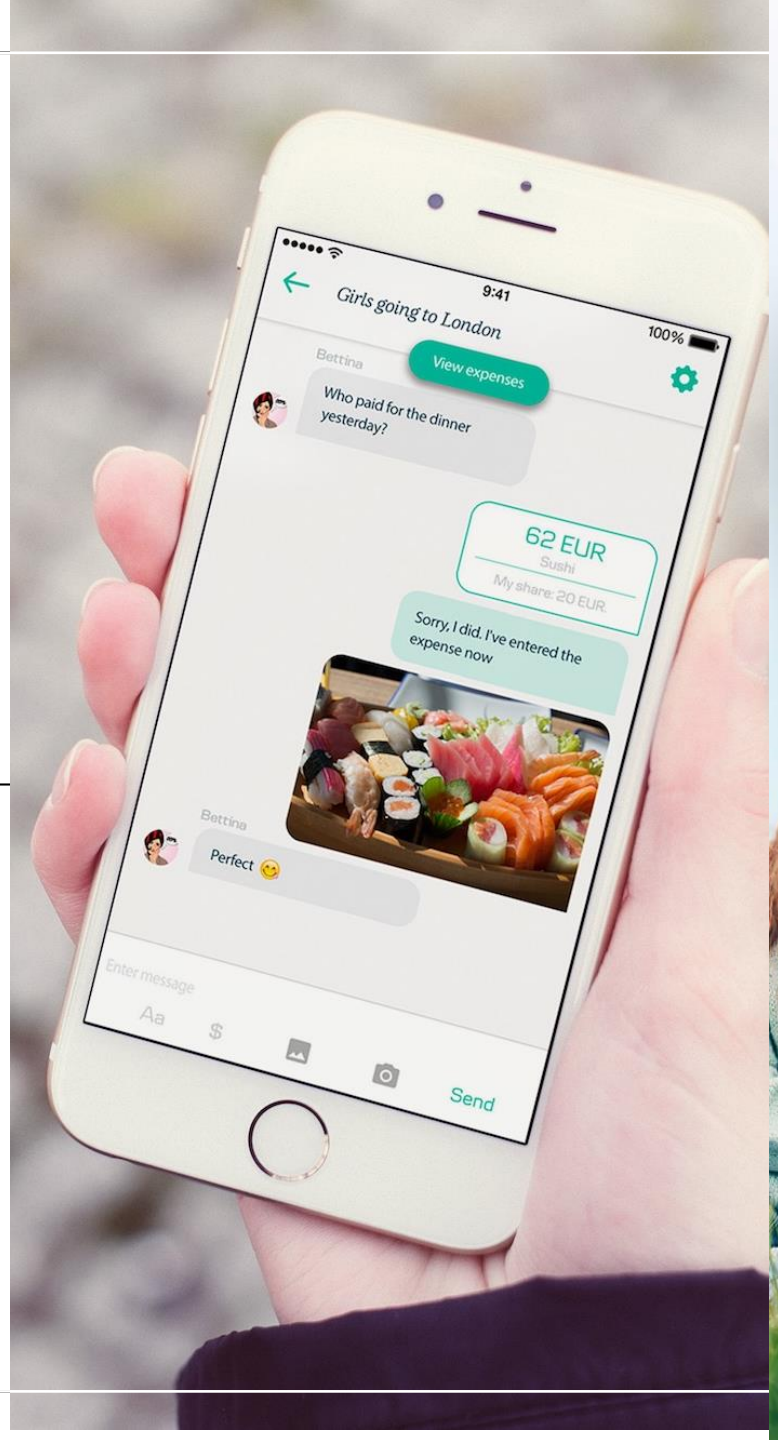
Jonas Larsen (jlars22@student.sdu.dk)

android

Objective

- Develop an Android mobile application for tracking and managing shared expenses
- Ensure user-friendly interface and clarity of who and what each person paid for
- Use cases could be
  - Meals
  - Gifts
  - Travel Costs
- It should calculate and notify members of their respective dues.

Examples


9:41
Girls going to London
100%
Bettina
View expenses
Who paid for the dinner yesterday?
62 EUR
Sushi
My share: 20 EUR
Sorry, I did. I've entered the expense now
Bettina
Perfect
Enter message
Aa
$
Send


9:41
Anders Andersen
Last round
New activity
125 kr.
Du skal modtage
443 kr.
Alle udgifter
Farewell
125 kr.
Du skal modtage
50 kr.
Alle udgifter

Functional Requirements

1: 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 Autho or by using dummy data and mocking interactions.

(Database and security is not required, but is recommended)

Functional Requirements

2: Group Management

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

3: Expense Entry & Management

- Enable users to input expenses, specifying who has paid and the amount.
- Facilitate the splitting of expenses among all or selected group members.
- [Optional] Allow attachment of receipts or invoices.

Functional Requirements

4: 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.

5: Notifications & Communications

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

Non-functional requirements

Focus on Reliability: Testing of atleast one

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

Deliverables

Design Document: 3 pages max

- Short and concise, just the basics.
  - FR/NFR.
  - Navigation model.
  - Low-Fi wireframe.

Implementation Document: 4 pages max

- Discussion of the implementation of the application.
- Diagram of the architecture of your project.
- A section detailing how you handle the non-functional requirements of reliability.
- A short video demonstrating the application.
- Link to the github repository.

Video Showcase of Joakims App last year

Submission Deadline: Nov 24th
Presentations: Nov 27th