Ideation Report

Tahu Tech: Bill Samurai

January 2023

Ideation

Diners' dilemma is a situation where diners agree to divide the bill evenly before ordering their meals, therefore resulting in overall higher costs as each of them would order more expensive meals. The process is financially unfair, especially for the person ordering less expensive meals. [Investopedia, 2021]

This dilemma is often avoided by agreeing to pay only for the items that each respective diner's orders. However, if each person gets billed individually, it will be inconvenient for the waitress. Therefore, most groups would choose a person to pay in advance, and the rest of the group would then pay back the original payer. This process is called bill-splitting.

If done manually, bill-splitting is painstaking as the person needs to identify each diner's items, sum the amount, calculate the GST and service charges, and keep track whether each diner has transferred the correct amount to the person. Moreover, bill-splitting is very slow, especially if the group is large, because the process has to be done repeatedly for each person's order. Therefore, our motivation for this project is to **make bill-splitting effortless and quick.**

One way to make bill-splitting effortless and quick is through the use of a bill-splitting app, which is a software application that helps people track and split expenses with friends or family members. These apps are designed to make it easy for users to input the details of their expenses, including the amount, the date, and the payer, and to calculate the appropriate amount that each person should pay. Some examples of commonly-used bill splitting apps are Splitwise, SettleUp, and Line Split Bill. [The Balance, 2022]

However, there are still some issues with these apps. Firstly, some bill-splitting apps may not offer the ability to scan receipts as a way for users to input the details of their expenses into the app. This is tedious as users need to type in the items in the receipt one by one, which may cause bill splitting with the app as painstaking and slow as without the app. Secondly, some bill-splitting apps like Line Split Bill may not offer the ability to connect to users' bank accounts or payment systems, such as PayPal or PayLah!, to facilitate the transfer of funds between users. This is inconvenient and prone to errors as users need to key in the amount they need to transfer manually.

Therefore, our project idea is to improve existing bill-splitting apps by **streamlining all processes of bill-splitting,** ranging from inputting data, cost calculation, and payment transfers, into one seamless process. Using our split-bill app, the users will be able to:

- Take a picture of the physical receipt which will be processed by the app
- Key in the people who ordered certain items from our contact list
- Send payment requests to respective diners
- Transfer payments using payment app linked to the split-bill app
- Display status bars to keep track who has transferred their payments

One thing to note is that although the bill-splitting app was inspired by the diners' dilemma, the app can be extended to a variety of contexts outside of diners, such as supermarket receipts.

References:
Best Bill-Splitting Apps of 2023. (2022, June 24). The Balance.
https://www.thebalancemoney.com/best-bill-splitting-apps-4170968

What Is Diner's Dilemma? (2021, October 11). Investopedia.
https://www.investopedia.com/terms/d/diners-dilemma.asp

Tech Stack (Languages/Packages/Frameworks)

- React Native JS (frontend & backend)
- Python OpenCV, pytesseract, Imaging Library (Machine Learning)
- Paylah API (payment method)
- MongoDB (database)

Structure of Code/Frameworks

Frontend and backend of the mobile app will be built using React Native JS.
The following is the structure of the mobile app:

1. Obtain an image of the receipt using the phone's camera.
2. Clean the image to handle uneven boundary, wrinkles and folds, faded or scratched receipts, handwritten markings, and image taken at a weird angle
3. OCR the text and interpreting the OCR'ed text: itemised list, quantity, inline discounts, tax rate, subtotals, etc.
4. Store the texts in a MongoDB database for recording transaction history.
5. Connect to PayLah! (or other payment methods) to send payment requests and transfer money.
6. Accumulate past receipts and introduce some AI to train on the dataset collected. Use the AI to improve the interpretation of receipts.

Explanation of the use of Blockchain/ ML in the API

After taking a picture of the receipt, Machine Learning in the form of optical character recognition (OCR) will be used to scan the receipt, clean the image, do text recognition, and determine the shop's name, the items, the prices, quantities, and any charges. The text data will then be categorised to be inserted into the database.