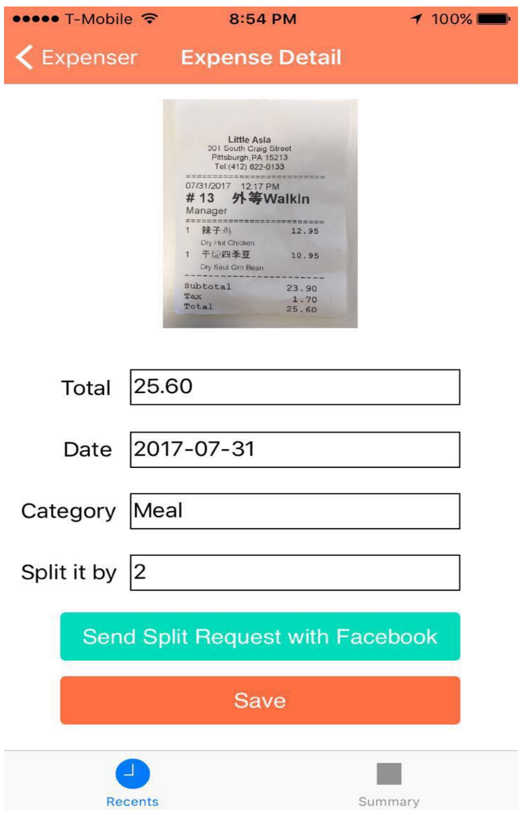
Final Report

Product Overview

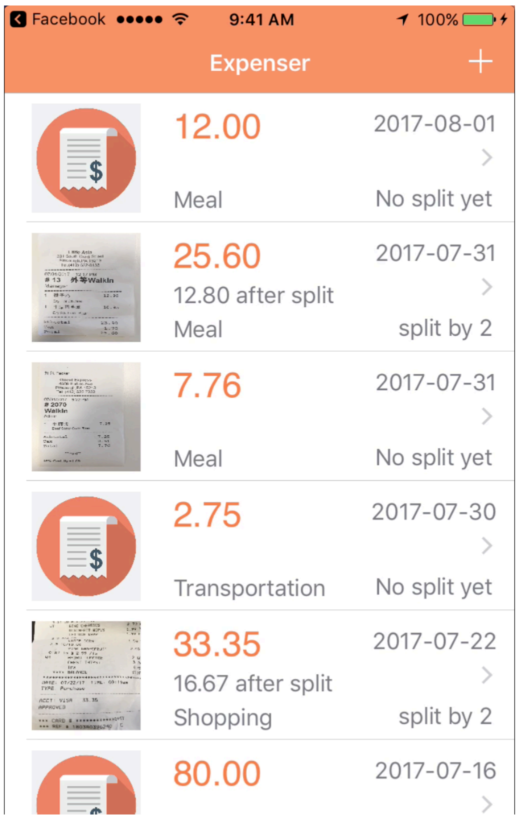
Our app name is called Expenser, it's a personal accounting assistant app, which provides three key features: (1) keep track of users' daily expenses; (2) split bills with Facebook friends; and (3) generate monthly summary using pie charts.

Use Case 1: Smart Scan
Expenser can scan receipts to record expense details, including total amount, transaction date, expense category, intelligently, Also, if we want to change the specific field, like expense category or total amount, we can change them manually, which is shown below.



Little Asia
301 South Craig Street
Pittsburgh, PA 15213
Tel:(412) 622-0133
07/31/2017 12:17 PM
13 外等Walkin
Manager
1 辣子鸡 12.95
Dry Hot Chicken
1 干煸四季豆 10.95
Dry Saut Grn Bean
Subtotal 23.90
Tax 1.70
Total 25.60

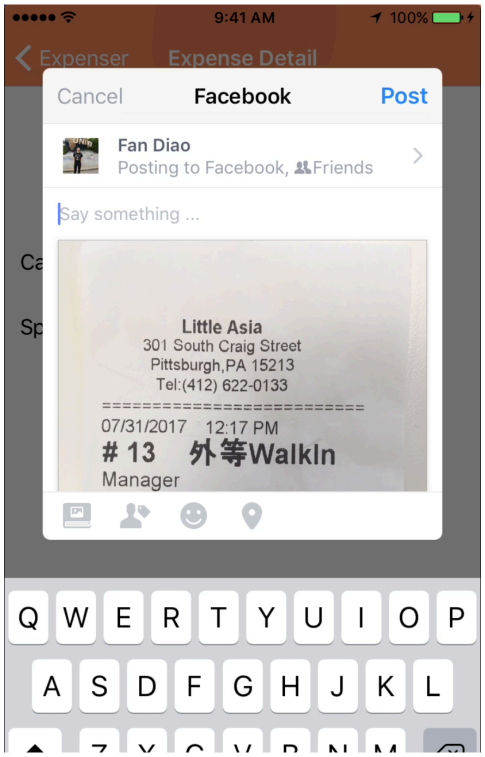
Total: 25.60
Date: 2017-07-31
Category: Meal
Split it by: 2



12.00 — 2017-08-01 — Meal — No split yet
25.60 — 2017-07-31 — 12.80 after split — Meal — split by 2
7.76 — 2017-07-31 — Meal — No split yet
2.75 — 2017-07-30 — Transportation — No split yet
33.35 — 2017-07-22 — 16.67 after split — Shopping — split by 2
80.00 — 2017-07-16

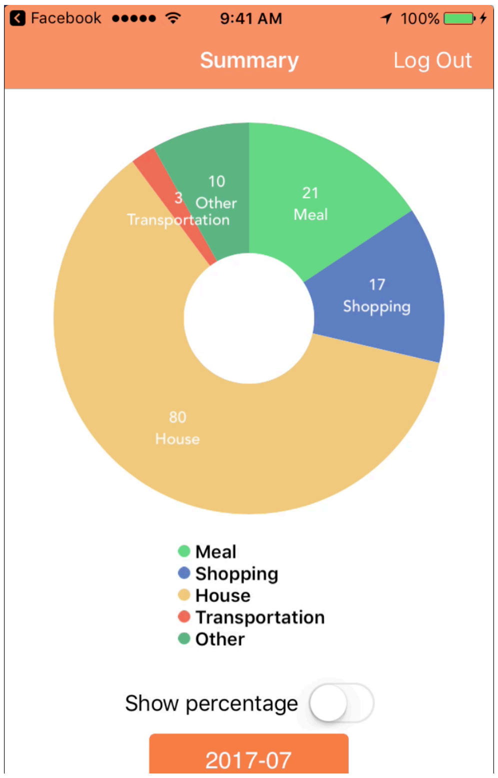
Once you have clicked the save button, the receipt details would be saved into the database, and a small summary of this receipt would be appeared on the main screen, which is shown above.

Use Case 2: Accurate Split
After scanning the receipt and input the number of people we want to split, the user can split bill with Facebook friends and send them notifications on their Facebook Timeline. Specifically, we can click “Send Split Request with Facebooks” button and choose the friends we want to notify in the below popup window. Apart from that, we can add some words and the amount of money we want to request.



Little Asia
301 South Craig Street
Pittsburgh,PA 15213
Tel:(412) 622-0133
==========================
07/31/2017 12:17 PM
13 外等WalkIn
Manager

Use Case 3: Summary
The third user case is that Expenser can show expense summary and details per category by month. Here, we try to display the expense information using pie chart where Expenser compares the user’s expenses and their proportion in the given month according to their specific category. In addition, we can move on to other month to show the corresponding expense information.



Solution Overview

Technical Challenges (API and 3rd party web service usage):

Google Vision API

To support the scanning and resolving of receipts, we use Google Vision API which is a third-party OCR API. Google Vision API enables us to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API[1] .

In terms of data exchange type, the Cloud Vision API is a REST API that uses HTTP POST operations to perform data analysis on images[2]. The API uses JSON for both requests and responses. In addition, when uploading image, Expenser would resize the image if it exceeds the 2MB API limit and performs base64 encoding.

For this part, we need also to consider image processing before uploading images. We need to resize an image if it exceeds certain size (such as 2M). And we need to do base64 encoding so that we can serialize it later in JSON format.

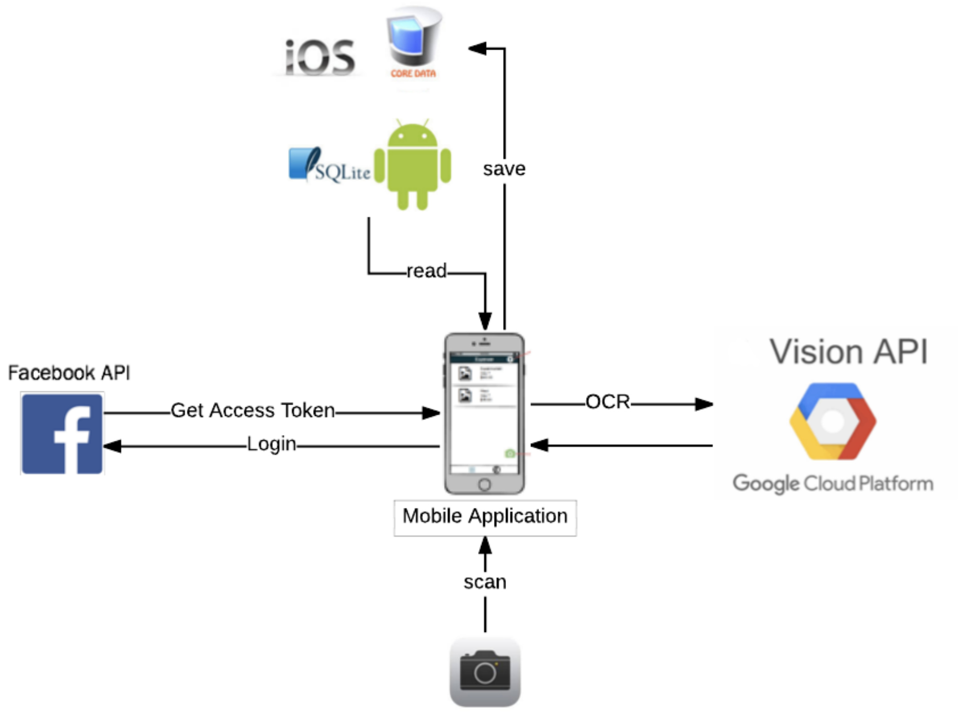
Facebook Integration

Our app integrates with Facebook via Facebook SDK for iOS and Android which supports Facebook login and sending messages to Facebook friends. The problem here is how to send private messages to Facebook friends directly from a third-party app?

We tried the following several approaches. First, can we send split requests via emails to friends? The answer is 'No', recent versions of Facebook API does not allow getting emails of Facebook friends. Secondly, can we send messages from Facebook share dialog? The answer is 'Yes', we can call Facebook API to show its share dialog and post to friends' timeline. But probably our friends do not want their bills to be public. Thirdly, can we send messages privately? The answer is 'Yes', we can send via Messenger. But what if the user hasn't installed Messenger? In this case, the app will detect that and let the user can send via Facebook share dialog.

Drawing Pie Chart for "Monthly Summary"
In order to fully display the comparison of expense information among different categories, we use Pie Charts for Monthly Summary. For iOS, we import *'PNChart'[4]* github library to draw Pie Chart, while for Android, we employ the *'MPAndroidChart'[5].*

Architecture Overview(Sensors, Social media, 3rd party web service APIs)



Lessons earned

Lessons Learned: iOS vs. Android

iOS:

(1) It would be easier to use Pod for importing third party libraries. For example, when integrating with 'PNChart' library, initially we tried to import it by manually copying all the necessary folders into our project, which was quite time consuming and error-prone. After we tried to use Pod, things became much easier.

(2) It would be good if we develop and test each individual component/function before we add it to our project. If so, we can debug more easily if there is something wrong.

Android:

(1) It is always a good practice to finalize MVC model and then start coding. Otherwise, it will take a significant amount of time to recode along with the design changes.

(2) A plenty amount of feature requirements is already implemented and exposed as web service API. It is a good practice to integrate existing mature API into our app development, instead of rebuilding things from scratch.

Comparison between iOS and Android in your application development

(1) UI design for iOS is easier and more direct than that for Android. For iOS, we can drag and drop and components in the storyboard, while for Android, we need to write xml file which is not so easy for new learners.

(2) For iOS, it's critical to be familiar with the differences among different view controllers and relations among them, such as 'UIViewController', 'UINavigationController', 'UITabBarController', etc. While for Android, we need to be familiar with the differences and relations among Activity, Fragment, Content Provider, Intent, etc.

Reference:
[1] https://cloud.google.com/functions/docs/tutorials/ocr
[2] https://cloud.google.com/vision/docs/request
[3] https://www.programmableweb.com/api/facebook
[4] https://github.com/kevinzhow/PNChart
[5] https://github.com/PhilJay/MPAndroidChart