TikTok Group Project Report
Group 15

Clement Loh | Qiu Weihong | Wee Chun Hui | Ignatius Ng | Brenda Lim | Stephanie See

Introduction

Physical shopping lists are a hassle to manage and can be easily lost while their respective warranties are forgotten. Paper receipts, which are still prevalent as proofs of transactions, often use thermal paper which fades after a period of time as well. To enable convenient expenditure and budget tracking from the time of transaction itself, our group decided to create a digital shopping list application, known as Spending, which adds on to the existing transaction procedures and can be installed onto any Android device.

Our Product

Spending is a Java application with four main features that can be accessed using a navigation bar at the bottom of the screen. The pages that house the features are the homepage, Upload page, Expenditure page and Login page.

Some of the features implemented within the application are:

1. Monthly expenditure
2. Remaining monthly budget
3. Database of photo taken receipts for easy retrieval when claiming warranty/exchanges
4. Smart API recognition of goods purchased
5. Ability to edit anytime and anywhere

The homepage (Figure 1) is where the user can quickly access their current shopping list and edit its contents. The text box located at the top of the screen takes in the name of the object to be purchased, with a drop-down list of categories for that specific object. The add item button serves as a confirmation for both of the above user inputs to be listed as a shopping list item. The latest shopping list is fetched from the database and displayed below the input fields.

The Upload page (Figure 2) records the items that the user bought when the user uploads a photo of the receipt (Figure 3) on the Upload page (Figures 4 and 5). This changes the information found on the Expenditure page (Figure 6), which provides a budget breakdown based on the uploaded receipts (Figure 7) and shows the remaining budget based on user-provided monthly budget.

The login page (Figure 8) ensures that users can only view and modify their own budget and shopping list, which is done using Firebase authentication.

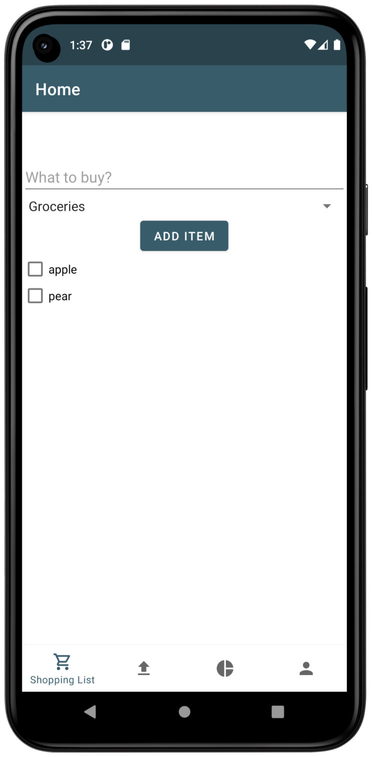


Figure 1: Homepage

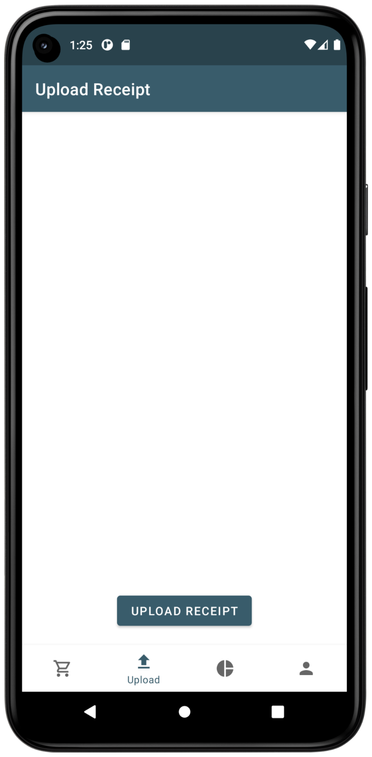
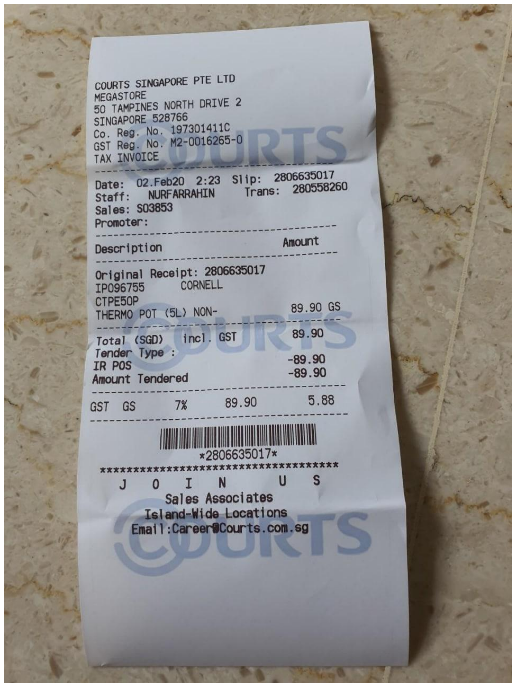
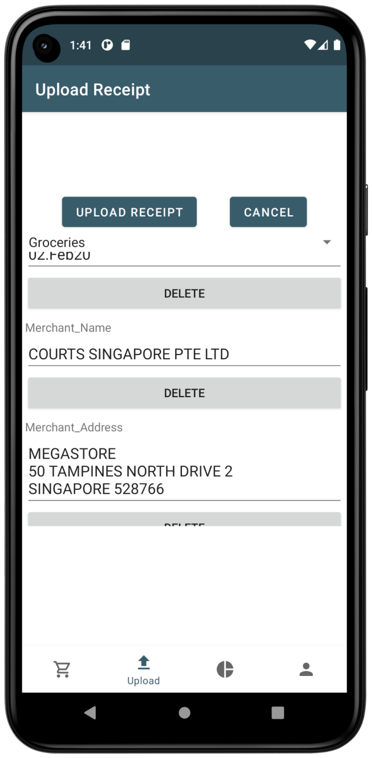


Figure 2: Upload page



```
COURTS SINGAPORE PTE LTD
MEGASTORE
50 TAMPINES NORTH DRIVE 2
SINGAPORE 528766
Co. Reg. No. 197301411C
GST Reg. No. M2-0016265-0
TAX INVOICE
-----------------------------------------
Date:  02.Feb20  2:23  Slip:  2806635017
Staff:   NURFARRAHIN     Trans:  280558260
Sales: S03853
Promoter:
-----------------------------------------
Description                       Amount
-----------------------------------------
Original Receipt: 2806635017
IP096755      CORNELL
CTPE50P
THERMO POT (5L) NON-            89.90 GS
-----------------------------------------
Total (SGD)   incl. GST         89.90
Tender Type :
IR POS                          -89.90
Amount Tendered                 -89.90
-----------------------------------------
GST   GS      7%      89.90       5.88
-----------------------------------------
              *2806635017*
*************************************
  J    O    I    N        U    S
          Sales Associates
        Island-Wide Locations
      Email:Career@Courts.com.sg
```

Figure 3: Sample receipt image



02.Feb20

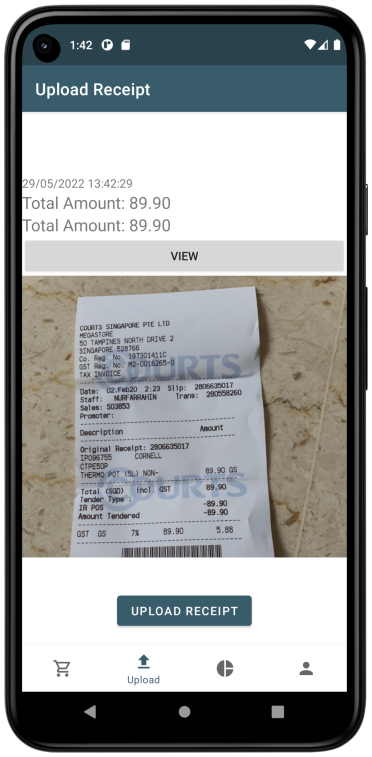
Merchant_Name

COURTS SINGAPORE PTE LTD

Merchant_Address

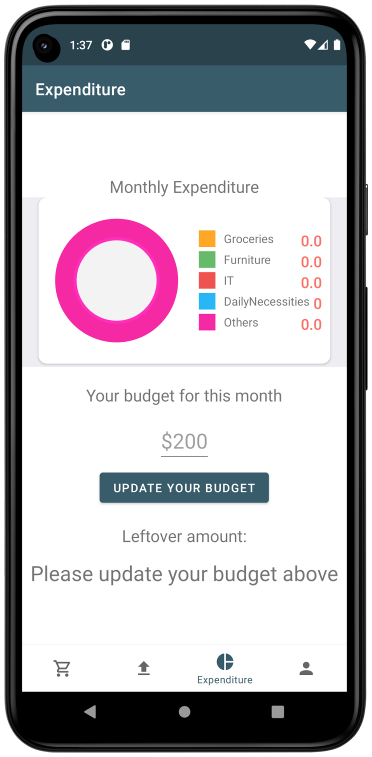
MEGASTORE
50 TAMPINES NORTH DRIVE 2
SINGAPORE 528766

Figure 4: Data display after OCR



29/05/2022 13:42:29
Total Amount: 89.90
Total Amount: 89.90

Figure 5: Uploaded receipt



Monthly Expenditure

🟧	Groceries	0.0
🟩	Furniture	0.0
🟥	IT	0.0
🟦	DailyNecessities	0
🟪	Others	0.0

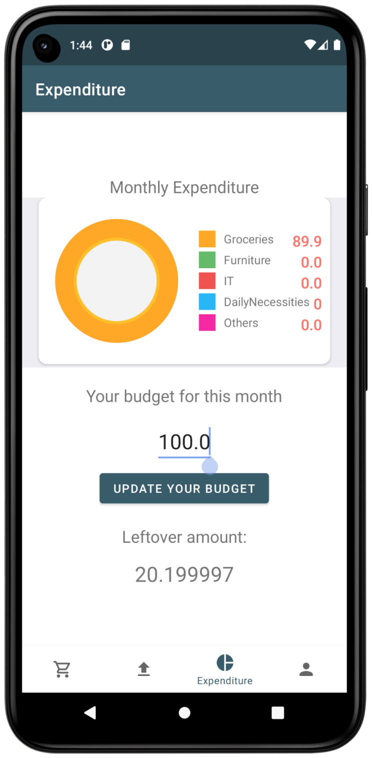
Your budget for this month

$200

Leftover amount:

Please update your budget above

Figure 6: Initial expenditure page



Monthly Expenditure

🟧	Groceries	89.9
🟩	Furniture	0.0
🟥	IT	0.0
🟦	DailyNecessities	0
🟪	Others	0.0

Your budget for this month

100.0

Leftover amount:

20.199997

Figure 7: Expenditure page post receipt

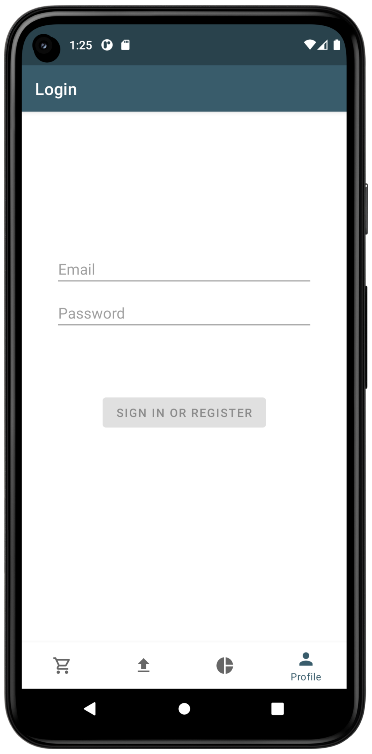


Figure 8: Login page (pre-login)

Technical Overview

The Model-View-ViewModel (MVVM) framework was used in the application. Each fragment is supported by a ViewModel that possibly obtains information from the Firestore database and Nanonets API depending on their functions, and returns the information to the fragment for display as per the XML files. All features are only accessible if the user is logged in and authenticated using Firebase authentication.

In terms of UI/UX, the whole application has a synced design theme for all xml files and the application logo for aesthetics. To display a pie chart for the expenditure page, two open-source libraries 'com.github.blackfizz:eazegraph:1.2.5l@aar' [1] and 'com.nineoldandroids:library:2.4.0' [2] were referenced. A navigation controller was added using NavController for the icons at the bottom of the page for easy access to all fragments. Hints and data validation are also available for all user input fields.

For backend work, the Nanonets API [3] was used to extract Total_Amount and the relevant data fields from the uploaded receipts. This API uses Optical Character Recognition to identify values that can fit into the various fields, which is then passed to the app’s expenditure page for further analysis. The shopping list and budget are all stored in Firestore and are accessed via API calls.

A unit test was conducted on the calculation of the remaining value, after deducting total_spent from budget_value. Additional informal UI tests were run where the flow of uploading receipts was modified several times in order to make the process intuitive for users.

Highlights

One of the most challenging parts of the application is to code all the features we had planned for our app before the deadline. We brainstormed about too many features in our app for a project with such a short timeframe such as displaying the price of each item, warranty, settings page, home page, notification page and a reminder function that alerts whenever an item’s warranty is ending. We hence had to cut down on some features while ensuring the essential functions are kept. As time was running out, we needed to ensure that all of the features we implemented are functional as well. It is important to prioritise the essential features due to the tight deadline provided.

In addition to this, another challenge was incorporating everyone’s code together. Each member was given different tasks to complete, we had to work on seperate parts of the code individually. As some of us had limited experience when it came to coding, using github to share and work together on the app was a first for some of us. Learning how to use github’s features such as merging different branches of code together as well as pulling updated code from others were definitely very useful to know. This would be especially beneficial for future projects with different groups.

Project Management

We defined the tasks mainly by the features within our app.

We have the following features:

- Retrieval of data from firebase firestore (Weihong, Clement)
- Uploading of receipts and permissions settings (Weihong)
- Use the OCR API to detect keywords and retrieve relevant data to be stored in the Firebase (Weihong, Chunhui)
- Calculate the expenditure, get user's set monthly budget from firebase and calculate the balance for the month (Clement, Weihong)
- Represent the monthly expenditure in the form of a pie chart at the expenditure page (Ignatius, Clement, Weihong)

Other tasks include:

- Designing and improving the user interface including the app logo (Brenda)
- Testing of the expenditure formulation for the remaining amount balance for the month under user testing (Stephanie)

At the end of the project, Weihong merged all our codes together while the rest of the team wrote the report.

References

[1] Blackfizz, "Blackfizz/EazeGraph: An Android chart and Graph Library," *GitHub*. [Online]. Available: https://github.com/blackfizz/EazeGraph. [Accessed: 29-May-2022].
[2] JakeWharton, "Jakewharton/Nineoldandroids: [deprecated] Android Library for using the honeycomb animation API on all versions of the platform back to 1.0!," *GitHub*. [Online]. Available: https://github.com/JakeWharton/NineOldAndroids. [Accessed: 29-May-2022].
[3] *NanoNets API Reference*. [Online]. Available: https://nanonets.com/documentation/. [Accessed: 29-May-2022].