

CS 4720 - F17 - Final Project Document

Link to Github: <https://github.com/UVA-CS4720-F17/final-project-final-sudowoodo>

Device Name: Sudowoodo

Platform: Android

Name: Joe Chey

Computing ID: jc7yu

Name: Katherine Qian

Computing ID: kq3zm

App Name: Money Tracker

Project Description:

Money Tracker is an application that will serve consumers who wish to keep better track of their monthly expenditures. Budget management is an important aspect in life, and we believe that it would be useful to provide an efficient way to store and organize expenditure information. Through Money Tracker, users will be able to record and organize a variety of data regarding their expenditure, including name of expenditure, category, date/time, and description of the item, and photos.

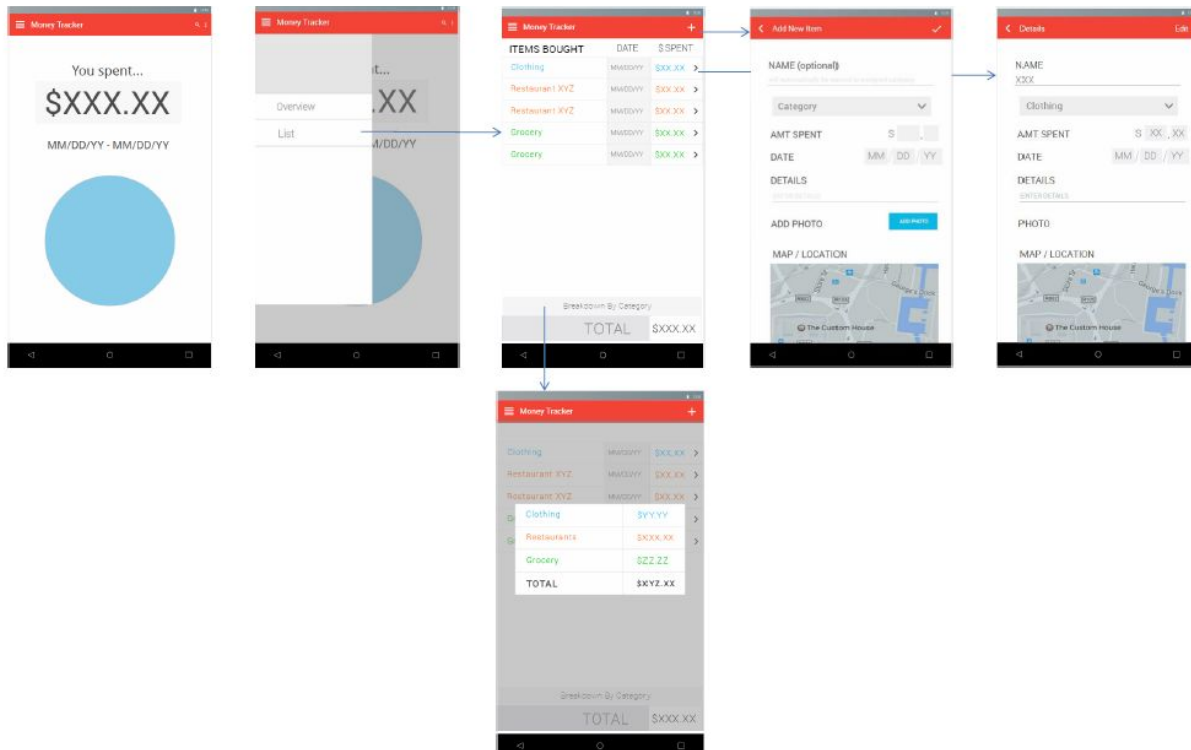
What we propose to do is create an app that will do the following:

- The system will allow users to list their purchases/spending under different categories
- The system will allow users to make an expenditure entry that includes date, location, category, and description of the expenditure.
- The system will allow users to expand each entry to include pictures of the receipt or the item that they bought.
- The system will allow users to save their images that they took through their app
- The system will allow users to share their expenditure log with others through email.

We plan to incorporate the following features:

- Camera - Users can take pictures of the receipt or the item that they bought for future reference
- Data storage using SQLite: Store data of all the categories and expenditure entries in SQLite database
- Open shared activity / features: Users can share their expenditure log with others through email
- Data storage using file write: Users can download the image that they took through the app and save it in their gallery

Wireframe Description:



For the overview screen (the first screen), we originally planned to have a pie chart that displayed the distribution between the different categories of expenditure. However, we found it to be a little out of scope whether it was us manually creating the pie chart or bringing in Google charts (that seemed nice, but harder to understand tying it in with mobile development and dynamically generate based on the dataset). In addition, the dates were there originally so that the app could reflect payment cycles or specific time frames that the user wanted, but we weren't able to implement this feature, since we felt challenged enough in trying to manage the SQLite data structure.

The side drawable navigation menu is used to navigate between the list and overview screens. While we did follow the list screen in most of the information displayed and the color coding of items based on category, we didn't get to the implementation of the popup for breakdown by categories.

Lastly, the input layouts of some aspects such as amount spent and date do not completely follow the wireframe, which was in large part due to the constraints on the layout tools built in Android Studio. While it would not have been completely out of question to implement aesthetic design aspects using external libraries/resources, we prioritized the soundness of back-end logic and data management over the aesthetics. Maps were forgone, as we didn't see much use in them for the complexity.

Some of the unrealized features were replaced by those that were not included in this wireframe: shared activity (email) and data storage using file write.

Platform Justification - What are the benefits to the platform you chose?

Android is more widely used among the population around the world. In addition, neither of us owned Macbooks/Apple computers, which would have been a significant barrier in working on this project and made it more burdensome.

Major Features/Screens - Include short descriptions of each (at least 3 of these)

Overview Screen: The main activity screen that displays the total expenditure logged through this app.

List Screen: Displays a list of all expenditure entries. Each entry is colored based on the category of expenditure ("Grocery": green; "Restaurants": orange; "Clothing": blue; "Bills": red; "Other": purple). At the bottom of the screen, it displays the total amount spent. This screen is accessible through the drawable menu on the top left corner of the screen.

Add/Edit: Share the same layout with fields that allow the user to input entries. Features in the screen include name, category, amount spent, date, details, and picture. All fields need to be completed for a successful entry, with the exception of the picture. An unmodified date is automatically set to be the date of the day. Add item screen is accessed by clicking on the three small circles in the top right corner which would show the Add Item option upon click (this is possible on both overview and list screens). Edit Item screen is accessible by clicking on an item in the list.

Optional Features - Include specific directions on how to test/demo each feature and declare the exact set that adds up to ~60 pts

- (15 pts) Camera - Users can take pictures of the receipt or any other possible related picture to what they bought for future reference; this is accessed through "Add Item" and "Edit Item." Users have the option of using the camera to take a picture or upload a picture from gallery.
- (20 pts) Data storage using SQLite: All data regarding the expenditure entries are stored in an SQLite database for retention between each session. It is consisted of 10 columns that correspond to name, category, amount spent, date, year, month, day, details, blob (for picture byte array), and the row index of the entry in the table. When the application is opened for the first time right after installation, the list will be empty, so items need to be added in order for this functionality to be observed in action.
- (10 pts) Open shared activity / features: Users can share their expenditure log with others through email. This can be done in two ways: 1) clicking on the floating action button in the bottom right of the screen in overview, or 2) through the drawable side menu by clicking on send. Both open up an email app (gmail) and fills in the sender, subject, and body (with the information displayed for all entries in the list).
- (15 pts) Data storage using file write: Users can download the image that they save in the app (whether it is from camera or gallery) and save it in a newly created folder labelled "Money Tracker." This can be done in the add and edit screen through a button grouped with the camera entries.

Total: 60 points

Testing Methodologies - What did you do to test the app?

Tested among ourselves thoroughly as we were developing the app, with a concluding final test upon finishing. We tried to think of corner cases and random situations that could happen in the app. Occasionally as we were making the app, we let our friends play around with it, as well.

Usage - Include any special info we need to run the app (username/passwords, etc.)

There is no login for this application.

Lessons Learned - What did you learn about mobile development through this process?

Even seemingly simple features can be difficult to implement. In addition, certain features require different structures and methods for implementation that need to be explored early on. For example, having fragments, SQLite, and RecyclerViews/adapters is not as straightforward as how it was implemented in the Bucket List application. Implementation of some functionality did not necessarily mirror implementation in another class. It was sometimes helpful to take a step back and think about how these different things interacted with each other and how information can be passed.