CS7IS5-A-SEM202-201920: ADAPTIVE APPLICATION

# Adaptive Purchase Planner (APP) - Project Outline

Team James

---

## Contents

# 1. Introduction

You're running to the store after work but realize you left your list at home (ugh). Instead of standing in the dairy aisle wondering how much milk you have left, next time use your phone as your personal grocery shopping assistant. APP (Adaptive Purchase Planner) takes out all the work for you, whether you want to share a virtual list with your spouse and even find easy deals. To help you get your grocery shop on, we've rounded up the best list-managing to streamline your next shopping trip.

The application can keep track of products users bought and recommend them to use when they need them, suggest other products that might interest them, so users can find better and cheaper products more easily.

Our objective is to build an application that will track the users' buying habits through explicit data collection and implicit recognition of users' choices in the app to suggest purchases in a timely manner.

# 2. Proposed Architecture

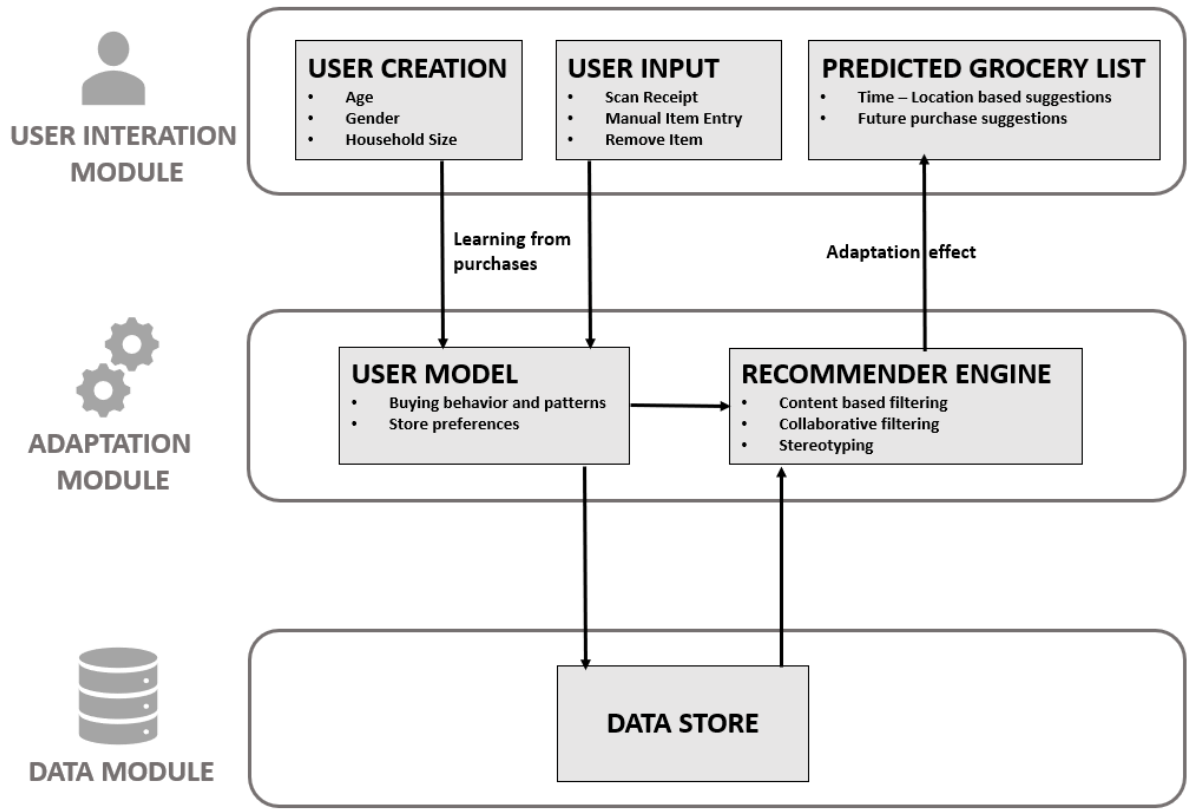


FIG 1. PROPOSED ARCHITECTURE

## A. Components

### 1. *User Interaction Module:*

- Users can add, edit or delete item details (e.g. Item, Quantity, Price, Location) manually, these activities will continuously update the user model.

- The system can recognize receipts uploaded by users, categorize them and feed results to the user model.
- The system can integrate with retailers in the future to simplify the process.

2. *Adaptation Module:*
   - The system can generate grocery lists that users might need to buy based on their user models.
   - The system can analyze different user models collaboratively and find the similarities among them and recommend items that might interest users.
   - Item-based, based on adjusted cosine:
     Pu,I = all similar items,
     N (si, N*Ru, N) all similar items,
     N (si, N), in which, si, N is the similarity between item i and item N, Ru, N is the score that the user u gave to item N and Pu, i is the prediction of item N that was never been graded by the certain user u before.
3. *Data Module:*
   - User models will be stored in a database.
   - User shopping history will be stored in a database.
   - Items details such as prices, quantities and locations will be stored in a database.

## B. Design

1. *Proposed Technologies – The goal for our application is to make a clean and user-friendly design. We want users to have easy access to their generated shopping list with our suggestions, while giving them a significant amount of control to edit the list in real-time, while the system adapts to what the users choose to filter out. Also, crucially the application will always be keeping track of the total cost of all items on the list and will help users keep track of their spending habits over time.*

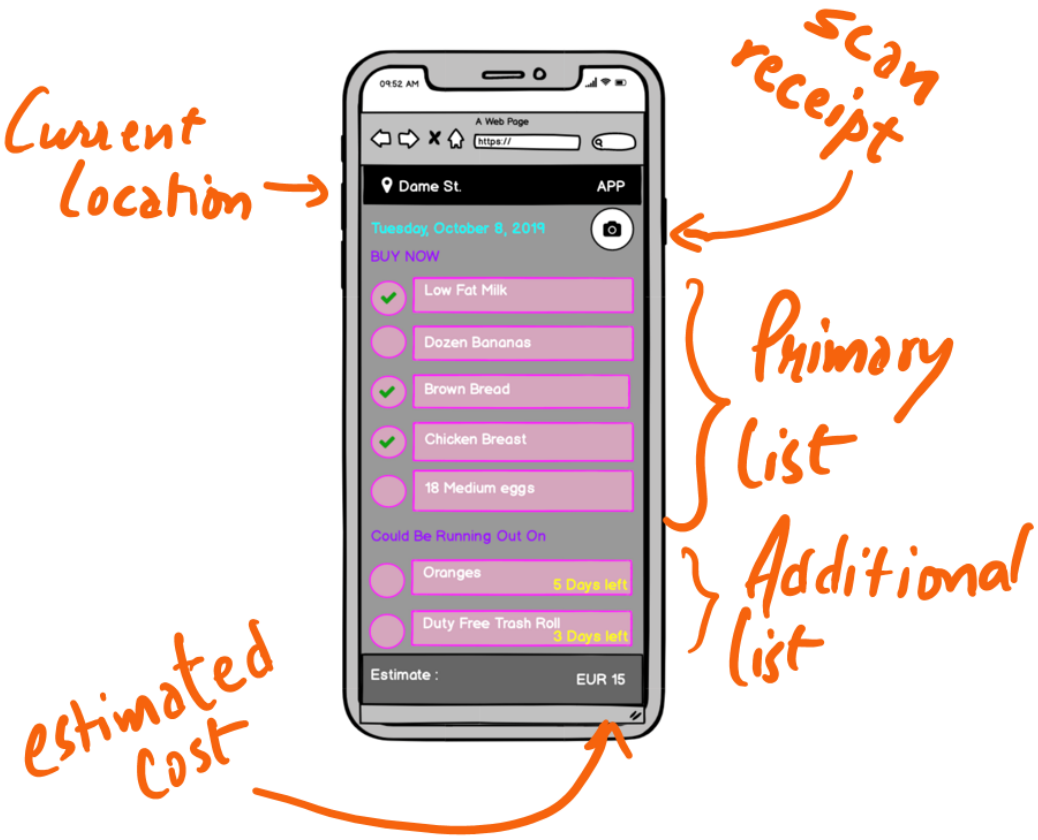


FIG 2. UI MOCKUP

Following technologies will be used to implement the above architecture in APP.

- **Backend Technology:** Python (Django)
- **Frontend Technology:** Angular JS
- **Database**: MySQL
- **Receipt Scanning:** Google Cloud Vision API
- **Machine Learning:** Natural Language Processing (NLP) to classify grocery items

2. *Recommender Engine – The underlying recommender system will be composed of three major filtering components:*
   a. **Content-based filtering –** Over time, the application will adapt to user preferences and recommend purchases based on a user's buying history, However, other factors will also be involved in building the model. When the application starts, the model will be built on User data which is collected explicitly, and Location details collected implicitly. A user also interacts with the system in different ways, like suppose user bought milk, and is recommended milk next month. And the user

buys it, it is considered as confirmation that user needs milk. This is where we will used Content-based filtering, to predict based on User specific traits.

b. **Collaborative filtering –** The highlight of the application is to best recommend most probable items to be purchased on that day. This depends mainly on the user's historic data and buying behaviour. The system understands the user's behaviour over time and predicts future needs. The basic idea is, "users who have agreed in the past tend to also agree in the future". For this use case, the best method is Collaborative filtering which works with a similar notion.
c. **Stereotyping –** As the User Model will require regular input from the user to stay accurate, we hope to use some aspects of Stereotype user modelling in the early stages when we lack data from the user, or to help plug gaps where we lack knowledge of the user's activities. The basic information we get from the user, such as their age, gender, location and household size will enable us to group them into a stereotype to be used as a starting point. As the user continues to interact with our system, the need for us to use this modelling technique will lessen, however, we can fall back on it, if a user decides to cease using the app for a long time before returning.

# 3. Management Approach

We intend to use an agile management approach for the development of the application. Each week, we will hold a minimum of 1 scrum meeting, during which we will recap the progress each person has made on their assigned task and if necessary, assign them new ones, as well as keeping an eye on the overall scope and progress of the project. We will also organize the goals to achieve for the next meeting.
All tasks will be broken down into their component pieces and assigned throughout the team. We intend to pair program so group members do not need to make design decisions in isolation.

## A. Role Assigments

Design data structure and implement data CRUD (Create, Read, Update and Delete) API.

I. Design the user model and implement create and update logic for the user model.
II. Identify the similarity between and generate recommendations based on similarity and user model.
III. Recognize items from receipts and classifies them into predefined categories.
IV. Implement HTTP API, including add and delete items, upload receipt photo, and show grocery list that system recommended.
V. Design UI and implement our web user interface.

**Note**: Role assignments are ephemeral and will be changed as per the requirements of the project.

## B. Timeline


# 4. Resources


- https://cloud.google.com/vision/overview/docs/get-started
- https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26
- https://media.neliti.com/media/publications/263309-comparing-content-based-and-collaborativ-eb8f08d0.pdf
- https://medium.com/@rabinpoudyal1995/content-based-filtering-in-recommendation-systems-8397a52025f0
- https://blog.csdn.net/yimingsilence/article/details/54934302