Group Name: OfferRain
Team Member: Yitian Hu, Zeyu Wang, Ziyi Chen, Haozhe Wu

Web App That Collects Grocery Product Information

Project Summary

Whenever it is the start of the school year, new students from all over the world arrive and move to Champaign almost at the same time. Many popular daily goods in the local markets are usually sold out extremely fast, and the restock process often takes a long time. However, the demand remains high until students purchase the products they want. So, many students will have to visit multiple markets in Champaign everyday to check if products are restocked. Such a process costs students unnecessary efforts and time, and it also results in crowded shopping space and poor shopping experiences. Therefore, the team decided to design a web app where customers can provide updates on the stocking state of products among several major markets in Champaign.

The purpose of this app is to allow for real time updates on the state of the products by customers themselves and a quick tip to them about where they can purchase their needs with the least amount of effort. In the meantime, customers can do price comparison among multiple stores. The team believe such a project can be a good practice of skills in database systems as well as full-stack software development skills.

Data Storage

- market table: marketId, market name, market address
- item table: itemId, item name, price, brandId(foreign_key)
- market_item:market_item_Id(Unique), marketId(foreign_key), ItemId(foreign_key), price
- brand table: brandId, brandname
- user table: userId, username, email, pwd
- comment table: commentId, content, date, userId(foreign_key)
- comment_to_item: commentId(foreign_key), market_item_id(foreign_key)

Basic Functions

- In real-time, the user could upload information about whether certain products are in short supply or being sold in the supermarkets around UIUC. Other users could see the application and decide when or where to buy the needed product.
- Generally, there are two functions to help the user see the status of the products. First, the user could see every piece of information uploaded by other users sorted by the time. Second, the user could group by the information of a product in a given period (e.g. last 24 hours, last 5 days, etc.)
- The user could also get where a certain product is sold by grouping the data by supermarkets.
- The price could also be uploaded and updated to help users to compare the supermarkets.
- If there is sufficient time, we plan to add a Rewards Points System to encourage the users to upload status about the products.
- If there is sufficient time, we plan to visualize and analyze the historical and seasonal patterns of the products and markets to help the users and the market managers to prepare in advance.

Creative Components

- We plan to deploy the whole service to the cloud which makes our service more accessible and reliable. We plan to use the server provided by this course to install, develop and deploy our application
- We plan to utilize Docker to manage, develop and deploy our environment, which can make our system more maintainable. Meanwhile, we will save time on issues related to development environments. Generally, each part of the system (front-end, back-end, database) will be separated into different dockers and they could use the network to communicate with each other.
- If there is sufficient time, we plan to use React combing with Apache ECharts or AntV to visualize the historical and seasonal patterns of the demands and supplies of products in markets (i.e. To see if the products in the target will short sale before a new semester starts).
- If there is sufficient time, we plan to use Kubernetes to manage all of our containers and environments, making the whole application easy to scale up.
- If there is sufficient time, we plan to use Redis to cache the high concurrent information (i.e. the status of certain products) sent and retrieved by the users. We plan to cache all the data related to the status of the products in a Redis cache and update these data into MySQL periodically.

Description and Usefulness

Most of the supermarkets have their own apps where the availability of various products are listed. However, the updates are solely made by the market itself where it can be delayed or inaccurate during rush hours. Moreover, there isn't a software where customers can search for a product among all markets nearby. On the other hand, our app collects all the product data from the nearby stores and allows users to provide updates on the availability of the products. Then, users can make a judgment on the availability of the certain product by checking other users' updates in the last few hours. In this way, customers no longer need to check the availability by visiting multiple stores.

Users can also use our app to do price comparisons among stores or check if there's a sale going on in certain stores. We believe there is huge potential for this application, and more usefulness can be explored in the future.

Realness

- The data in the market table comes from manual insert. We only consider markets near the UIUC campus
- The items in the item table come from the use of a web crawler. We will strictly follow the instructions on robots.txt. We expect the data size to be over 1 thousand
- The items in the market should be stored in the market_item table, coming from a web crawler. We will strictly follow the instructions on robots.txt. We expect the data size to be over 1 thousand
- The brands in the brand table come from a web crawler. We will strictly follow the instructions on robots.txt.
- The user information in the user table comes from user sign up. We may automatically generate a few records
- The comments in the comment table mostly come from user comments. But we may automatically generate a few records
- The comment_to_item is a relation table, it will be automatically generated using data from the item table and comment table

Detailed Functionalities

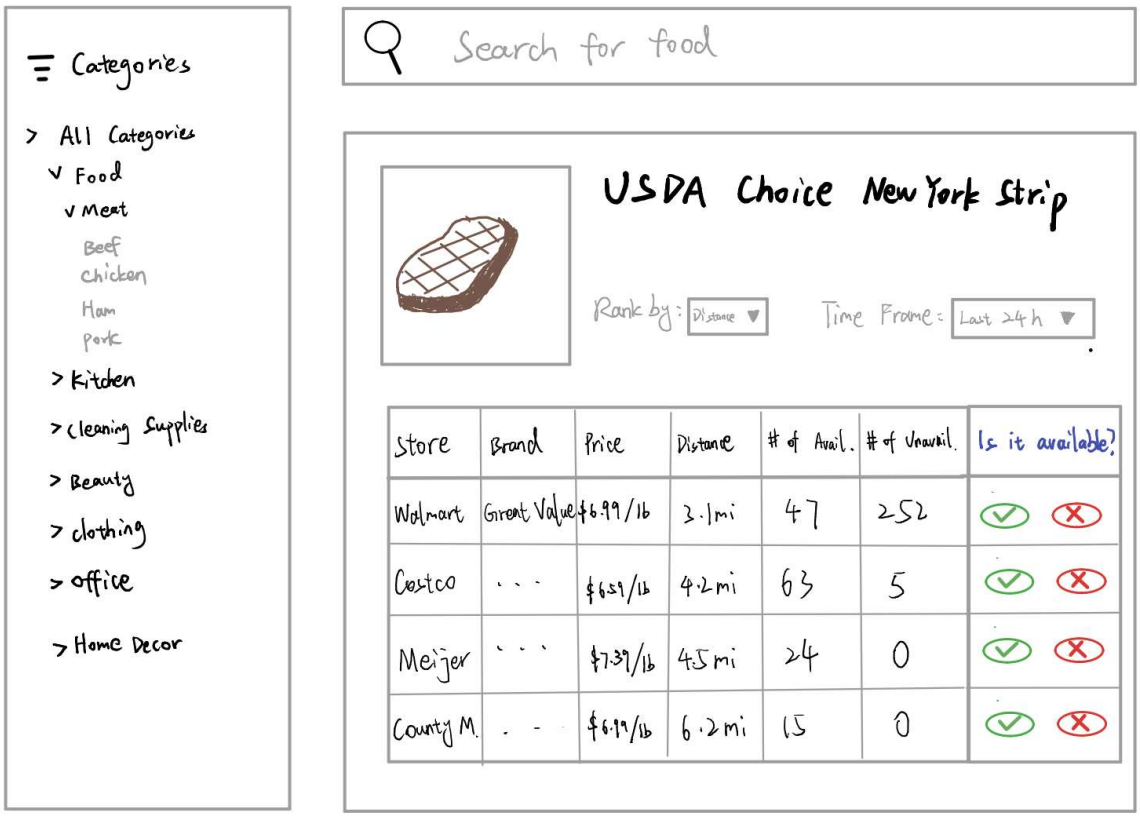
The functionality of our website includes but not limited to:

- It provides the complete CRUD functionality for users, who can always create, read, update and delete the information they provide of certain products, and

also administrators who can perform similar operations to markets, brands and products.

- It provides the filter functionality for users, who can easily select the data records they want to see by specifying constraints on product type or time on our website. We will also focus on code reusing by making good use of stored procedures.
- If time is sufficient, the user will receive reward points after they contribute a certain amount of information related to one product. If one user has enough reward points, he/she can redeem a coupon on some products. This feature will be implemented by using SQL Trigger.
- If time is sufficient, the user can search the content in our website by using the search engine we integrate into our application.
- If time is sufficient, a simple recommender system will be built in the back-end which will ensure the products displayed to users are ranked according to their preferences.

UI Mock-Up



Responsibilities

We divided the development process into several sections: UI Design, Front-end development, Back-end development, Database development, and DevOps. The corresponding group members of which are in charge:

- UI Design: Mike Wu
    - Mike will be in charge of planning and designing the layout of the user interface.
    - Mike also holds the responsibility of planning the functions of the app and their renderings.
- Front-end: Ziyi Chen, Yitian Hu
    - Ziyi Chen will be in charge of designing the framework of the front-end
    - Yitian Hu will be responsible for connecting the front-end to the back-end server.
- Back-end: Zeyu Wang, Mike Wu
    - They will be in charge of the development of the back-end functions and connecting them to the front-end and the database system.
- Database: Mike Wu, Yitian Hu, Ziyi Chen, Zeyu Wang
    - Mike Wu and Ziyi Chen will focus more on analyzing application requirements
    - Mike Wu and Ziyi Chen will be responsible for connecting the database to the back-end server.
    - Yitian Hu and Zeyu Wang will focus on the design and implementation of the whole database system
    - Yitian Hu and Zeyu Wang will be in charge of the data collection and cleaning.
- DevOps: Zeyu Wang
    - Zeyu Wang will be responsible for creating, developing and deploying all the related environments for the proposed application.
    - Zeyu Wang will be in charge of the management of the GCP server.
- Documentation: Mike Wu
    - Mike will put more focus on the documentation, such as report layout and grammar checks.