What's Cookin' (Kitchen Assistant 2.0)

A Personalized Grocery Management and Recipe Recommendation Application

Abhishek Pandey
North Carolina State University
aapandey@ncsu.edu

Chengyuan Liu
North Carolina State University
cliu32@ncsu.edu

Piyush Mishra
North Carolina State University
pmishra@ncsu.edu

Xiaorui Tang
North Carolina State University
xtang9@ncsu.edu

ABSTRACT

Home cooked food is a luxury that most of us crave for yet the effort required both in time and money to manage your kitchen and grocery supplies puts us off from cooking most days. A significant amount of time goes into planning the dishes you choose to cook, based on factors like taste and cuisine preferences, time, money, nutritional value, and diet plans. People buy groceries either in brick and mortar stores or through an Internet based application, and cook a variety of dishes using these ingredients. Previous group presented an application for people to built and maintain an inventory of groceries with minimal manual intervention, on their phone, be able to view and edit it as they wish, and get recipe suggestions based on these groceries and user preferences to aid in cooking. Based on the results of the previous group, we have added a new section for users to actively search the recipes. In addition to, recommending recipes to users by scanning items on bill, we added a search feature, which users can actively use to search for their preferred recipes. They can determine whether the ingredients in the fridge/inventory are sufficient or do they have to go to the market to purchase depending on the required ingredients as well as the quantity. Furthermore, We used machine learning to predict the likability of recommended recipes. We used Neural Network to achieve the goal with a high accuracy of 89.66

Keywords – kitchen, ,android application, recipes, recommendations, ingredients, search, share

1. INTRODUCTION

There are a lot of students studying abroad just like us, we live on our own, everyday we have to buy many kinds of food and commodities, but due to the heavy academic load, we rarely care much about ourselves, in most cases, even if we buy lots of stuff, but after sometime we may forget about it, and this leads to the unnecessary wastage of food and money. So generally speaking, we cannot get a clear idea about how to use those ingredients in an optimized way. We have little to no idea on how to cook the dishes out of the routine with the ingredients we bought. Also it will take a lot of time to find the optimized use of the existing ingredients, besides it will be less efficient and eventually lead to waste of time and effort.

Everytime after buying groceries, we never care about the bill, or even throw it in a trash can as we go out, but if we make the best use of the bill, it will be better for us to cook in the kitchen, otherwise it will be tough to follow the items we bought, every time we may realize the food inventory needs to be restocked only when we open the fridge to cook something.

Even if we already got enough web resources right now, which allows us to search the recipes based on the ingredients. But we have less time to spend on going through the recipes, or to be familiar with the frequency of the stuff we need to buy.

Now there is no one-shoe-fits-all solution to cover all these troubles. We all can purchase the groceries, but what we need is an application to tell us what we can cook by using these ingredients. We may need other applications to check out the exact recipe, at the same time there will also be another one to maintain the list of food, this is really a boring and cumbersome process, and never allows us to try different recipes.

The previous team has already finish the 'Suggest Recipe' part with scanning the bill and lay out the recommended recipe depends on the items, our idea is to develop a new part on 'Searching Recipe', for those people who already got clear idea about what they want to eat but have no idea about how to cook or what are the required items. Moreover, we can check whether the we have the ingredients in sufficient quantity in our inventory, and then prepare a shopping list for missing items. Furthermore, the fridge management can give us a clear information about what is stored in our fridge, this is important as many people often forget what they bought, which leads to spoiled food. Also, after we cook or shop the fridge management will automatically update the inventory without manual addition or deletion.

2. LITERATURE REVIEW

The research on eating habits in America has revealed that people lack cooking skills and tend to eat outside[2], which negatively impacts the overall health. Unhealthy eating habits outside the home may be attributed to lack of time to plan and prepare meals at home[3]. In fact, lack of time has been cited as the major cause for people not cooking their meals at home[4]. The amount spent on food outside the home now is higher as compared to what it was 30 years ago[5]. Research has also shown that social eating alters our diet and we tend to eat more when we eat outside with people around[6]. This has become a large cause of obesity, an upcoming issue in the health domain. Furthermore, one recent paper showed that for children, half of all energy from fast food is consumed at home, demonstrating that even foods consumed within the home are not necessarily home-cooked[7]. Fastfood consumption has also been linked with the presence of potentially harmful chemicals, such as phthalates, that can contribute to severe adverse health effects[8].

The use of technology to encourage people to prepare and consume home cooked healthy meals will be extremely beneficial. A few applications on Google Play Store have tried to achieve the same. BigOven is one such application, that has a smart grocery list function and the ability to search for, upload, and bookmark recipes. However, the grocery items have to be entered manually. This tedious manual effort can discourage people from using the application. Also, there are no recommendations based on personal choices and attributes such as taste, cuisine choices, location, etc. There are also tons of websites that show the recipe for a particular dish. But searching and then filtering based on users choices can be tedious. Users then tend to go for the easy option of buying food outside. One popular mobile application is Kitchen Stories, which has a large number of filters to search for recipes based on cooking time, meat choices, etc. Once recipes are selected, it also auto-generates a shopping list. However, we see that these applications automate the steps once you select your recipe. We would like to take an inside-out approach to cooking management. Our aim is to add to this an important feature to auto-generate recipes for users once they purchase the daily groceries, and then recommend recipes based on these groceries.

3. PREVIOUS WORK

Based on the consideration of user experience and technology stack, the previous team have developed an Android mobile application that achieves most of their plan. The application adopted the MVC architecture and they involved Node.js to take charge of the server end. Also, they used RESTful APIs for client server communication which helps in maintaining a good Separation of Concerns between the client and server, making the application easy to scale. They deployed their server on Amazon Web Services Platform running Linux, with platform support for Node.js.

The app works as a kitchen assistant and helps the users manage their ingredients, kitchen supplies and feeds them recipes based on their inventory. The following are the 2 main features that they have incorporated into the application:

1) Inventory Management
2) Recipe Recommendation

3.1 Inventory Management

There are three ways to keep the inventory updated:

A. Manually add/delete/edit items and quantity
B. Using the phones camera to take a scan the paper receipt and convert them to electronic inventory.
C. Deleting ingredients automatically based on item cooked.

Options B and C enable quick and convenient inventory management. They have used image analysis and character recognition to convert the receipt users upload of their grocery purchases through the application. They have utilized Google Vision API (Image Content analysis) to extract the food items and their quantity from the receipt. Google Vision API (Image Content Analysis) was their choice for conversion of hard copies. Even they handled the case that the receipt is obtained as the electronic version of the text when groceries are bought online. They send the electronic version of the receipt as an image/pdf to Google Vision API to convert the receipt to text form which is then used to build up the inventory. They have stored keywords (food items and quantity) on their end and perform a loose string match on the results returned by the API to compare and update the quantities of ingredients. The user is then navigated to an editable list of groceries extracted from the receipt. Once the user is satisfied, the user clicks on submit and the inventory is saved successfully on the server. The user can now log in using the same credentials on any android device to view and edit the saved inventory. The inventory is also updated when the user cooks a particular suggested item. The user only has to press a button for the same, and the ingredients required to cook the item are deducted from the inventory appropriately.

3.2 Recipe Recommendation

Now that they have the users inventory and a large database of recipes that could be sent to the user, they will filter all the recipes that the users can cook. The application queries the database and searches for recipes using the users current inventory. If all the required ingredients are present in the users inventory, the item is shown to the user along with the recipe including instructions to cook the recipe. The application does not show the items that the user cannot cook, and thus avoids any confusion.

4. DATA COLLECTION

User Survey is an important part of planning and developing an application. We have conducted an on line survey of six questions and gathered response from various user bases like students and working professionals of varied age-groups.

This user study was conducted to judge the feasibility and adaptability of kitchen assistant different from its previous implementation.

Question 1: How often do you search for a recipe?

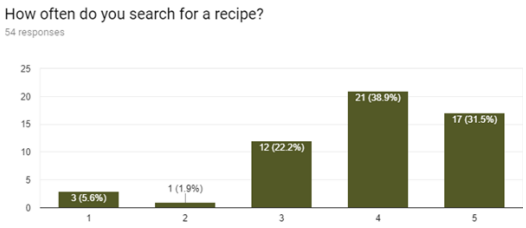


Figure 1: 1 - Never search online, 5 - Quite a lot

The primary purpose of this question is to check the functionality that we can add to the kitchen assistant based on the user request for an online search of a recipe. The chart shows that around 70% of the user group searched for recipe online. The application would be targeted for these users which brings ingredients, recipes and inventory management together in a single application. The user would receive recipe suggestions based on the preferences and ingredients the user has.

Question 2: How comfortable are you in cooking without a recipe?

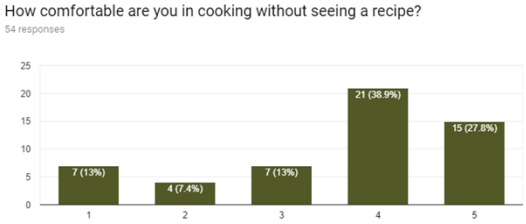


Figure 2: 1 - Very comfortable, 5 - Check the recipe a lot

This question further advances based on the input given by user in question 1. The main purpose of this question is to check if the user regularly checks for the recipe that he/she has prepared dishes on. This would help us in determining is the feature to integrate video/text recipe would be a good functionality in the kitchen assistant. Chart below shows that 66.7% of the users feel that they check the recipe online. It can be inferred that majority of the people who check for recipes online also constantly re-assert themselves on the recipes they are trying out. It has helped us to understand that recipe should be available to the user in a very user friendly way and that it should also be the primary highlight of kitchen assistant where user can easily navigate to the desired recipe.

Question 3: How often do you forget what you stored in the fridge?

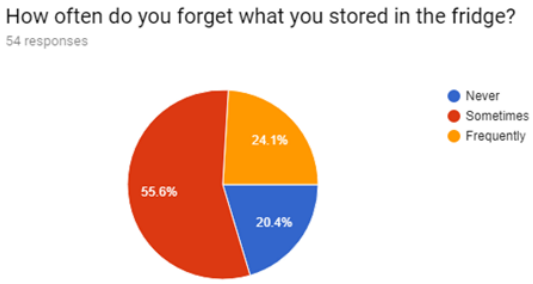


Figure 3: How often do you forget what you stored in the fridge?

We asked our users in the survey how frequently do they forget the things they kept in the fridge. Majority of the user base i.e. 55% of the users felt that they sometimes did forget the ingredients/food they stored in the fridge. 24% of the users felt they constantly forget what they keep in their fridge. Kitchen assistant can help the users to track their inventory of food items, ingredients they have already purchased so that they can make better judgment on what other things to buy based on some recipe and also to help them in utilizing their inventory before it reaches their expiry date. Various ingredients have a varied range of sustainability. This can be an important aspect of our application.

Question 4: How often do you start cooking and realize some ingredient is missing?

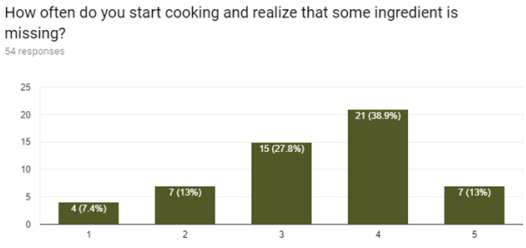


Figure 4: 1 - Never, 5 - Quite often

The chart shows that more than 50% of the users felt that they missed out on ingredients when they started cooking a recipe. This can be frustrating for users to realize after starting on preparing an item to miss out on ingredients due to lack of update or knowledge on the ingredients at home. This has given an insight of the core functionality for our application.

Question 5: Would you like to use a tool to manage the things you bought?

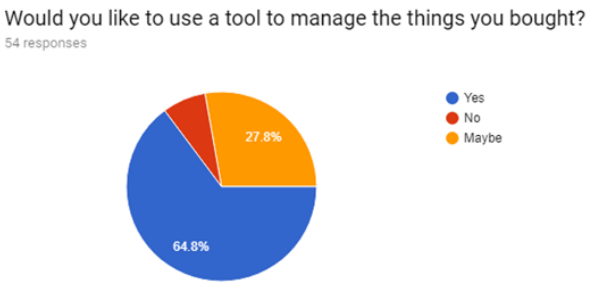


Figure 5: Would you like to use a tool to manage the things you bought?

This question helps us in determining the feasibility of our application. Almost 65% of the users responded they would really like an inventory management tool of some kind. The responses have helped us determine the functionality of our application.

Question 6: Open Ended Question

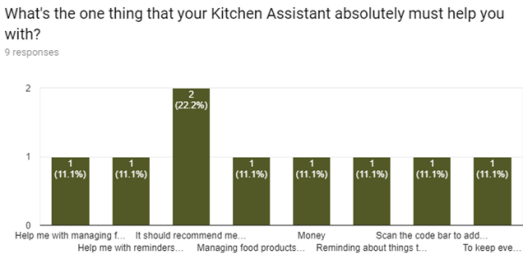


Figure 6: What's the one thing your Kitchen Assistant must absolutely help you with?

At the end of the survey we asked our surveyors as to what they would really like from a kitchen assistant they would use. We got quite a varied number of responses. A few responses were based on inventory management and notifications regarding the same. A few responses also suggested to scan either the bar code or image of things bought so as to elevate the burden on user to input each item. This question has helped us in determining new ways to improve the existing application.

5. EXPERIMENT

The first thing we needed to decide on was the platform to build this application. Given the restricted timeline and the amount of work done by the previous team we decided to go ahead with Android as our development platform. We needed to learn a bit of Android development in the beginning but since we were familiar with JAVA it wasn't an arduous task.

The next thing we thought of experimenting with was the Google Cloud Vision API. We found few other Image Extraction Tools online but none were as efficient and fast as the already implemented Vision API. So we decided to move on with it. Also, we used several machine learning techniques to use the extracted data and generalize the whole extraction process for receipts from any store. Unfortunately, we weren't able to get good enough results and accuracy to include our model in the application.

Moreover, the previous team used Node.js, AWS and noSQL database for their application. But we thought this would make the application really heavy and also it would take us sometime to get accustomed to these technologies because of their steep learning curve. So we decided to use Google Firebase to solve all our problems. We used Firebase for user Login/Authentication and storing the required data on the cloud securely.

We then searched online in order to get large and good enough dataset to use in our application. We needed a huge and rich recipe dataset to get few possible recommendations based on inventory list and also allow users to search their desired recipes. It is then that we stumbled upon this really amazing Spoonacular API which did it all for us and that too with good accuracy.

6. TESTING

6.1 Unit Test

- User Sign up : User should the first page as the signup activity and should be able to register as a new user if he/she hasn't signed up yet.
- User Sign in : User should be able to enter a valid and registered email-id and password, hit the login button and should be transferred to the main homepage.
- Add Item in inventory : User should be able to manually add an item into his inventory using the input text section.
- Delete Item in inventory : User should be able to manually delete an item from his inventory.
- Update item in inventory : User should be able to manually update the quantity of an item in his inventory.
- Add item using camera : User should be able to scan a bill and automatically update his inventory.
- Selecting recipe to cook : User should be able to view all recipes possible for him to cook according to his inventory.
- Finish cooking recipe : After the user hits 'I cooked this' button, the quantity of the ingredients used should get updated automatically.
- Searching for a recipe : User should be able to search for any recipe that he would like to cook or view from the online database.
- Recommend recipe : The recipe recommended to the user should match with the user's preferences and his past cooking selections.

6.2 Integration Test

- Inventory retrieval API call from android application to Firebase server must be completed successfully.
- Inventory update API call from android application to Firebase server must be completed and data must be reflected in the Firebase database.
- Inventory delete API call from android application to Firebase server must be completed successfully and data must be deleted from the corresponding tables on the server.
- Authorization of the user session token must be completed with calls to facebook API to validate the users facebook login.
- Authorization of the user session token must be completed with calls to Firebase to validate the registered users login.
- Android application must use the device camera to click photo of the paper receipts that has to be sent to the Google Vision API that must return the text form of the paper receipt.
- User search request should result in API call to Spoonacular API which would give the data back to the application.
- User's past searches, preferences and inventory should help the Machine Learning algorithm to predict and recommend recipes to user. This should result in automatic database update.

7. SOFTWARE DEVELOPMENT STRATEGY, BUGS AND ISSUES

7.1 Software Life cycle

1. Agile methodology: We have used Kan ban style of project management with daily stand up meetings which drove the entire development cycle.
2. Test-driven development: We started with TDD, but in later parts, found it hard to keep up to it. We then developed first before writing tests.
3. Continuous Integration Continuous Deployment: Although manually initiated, we tried to follow CI-CD so that we do not face deployment issues later.
4. Fail fast: We discussed the idea of the application with the previous team who developed it in the previous iteration to get an understanding about the project and decide features to develop accordingly.

7.2 Challenges

- Users need to be recommended recipes based on their ingredients and preferences. Currently this issue needs to be checked with an open issue involving Inventory database.
- OCR should recognize the ingredient already present in the inventory and update the quantity on client side rather than adding a new entry.
- Interaction with the cloud vision API, the secondary task includes enhancing the inventory database. Getting the response and showing it to the user.
- Home page of the android application needs to be enhanced for better user experience. Currently, the buttons need to implement the functionality and be aesthetic.

8. SOFTWARE REQUIREMENT SPECIFICATION

8.1 Functional and Non-functional requirements

1. **Functional Requirements**
   - **User Register/Login:** The user should be able to register and login into the application, even via login with facebook button.
   - **Firebase Connectivity:** The users should be authenticated using Firebase and the Firebase database should get populated/updated timely.
   - **Scan using Vision API:** The user should be able to use camera to scan receipt and the API should extract food items from it.
   - **Spoonacular API:** The Spoonacular API should be able to recommend recipes as per the Inventory List and also allow the user to search online available recipes.

1. **Non-Functional Requirements**
   - **Performance:** The application should be able to perform in real-time without lag and should update the database in realtime as well.
   - **Reliability:** The recipes suggested by the API should pertain to inventory list and the results should be consistent and reproducible.
   - **Scalability:** The application should be able to handle a large number of users and simultaneous traffic.
   - **Security:** The application should allow and authenticate users to securely login and keep their data safe in the database.

8.2 Use Cases

Our application can have following use cases:

1. **Register:** From the landing page of our application the user, if new to our application, can click on the "Register Now" button to go to the Registration Activity. Here the user is prompted to enter an email address, choose and confirm a password, and choose from a list of food preferences. Finally they can click on the Register button to register as a user into the application.

2. **Login:** The already registered user can enter their email and chosen password to login and go the Homepage of the application. Alternatively, the user can click on the "Login with Facebook" button to log into the application using their existing Facebook account.
3. **Scan the Receipt:** The user can click on the camera button on the homepage or can select the Scan option in the Menu drawer to open their device camera using which they can click the image of their receipt for text extraction. The application then automatically separates food items from the list and update the Inventory list database.
4. **Recommend:** From the Menu Drawer the user can click on the Recommend option which would then run the Spoonacular API in the background to search for the relevant recipes depending on the available ingredients in the inventory and user's food preference to suggest few top matches from all the applicable recipes.
5. **Search:** This is a straightforward use case, where the user can type the name of the recipe they want to search in the text box below the camera button on the homepage or can go to this option using the Search option from the Menu and then hit on the search button. Our Spoonacular API then performs an extensive search in the background to provide an exact match or the nearest possible match to the user with required ingredients and steps to cook.

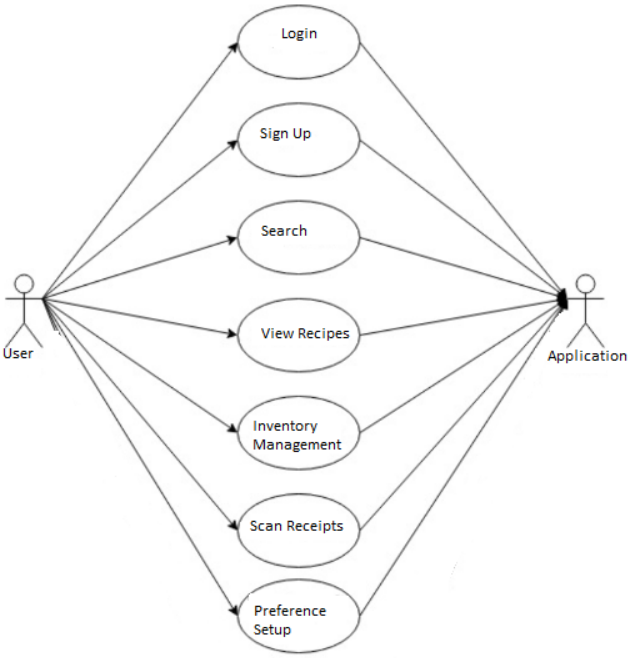
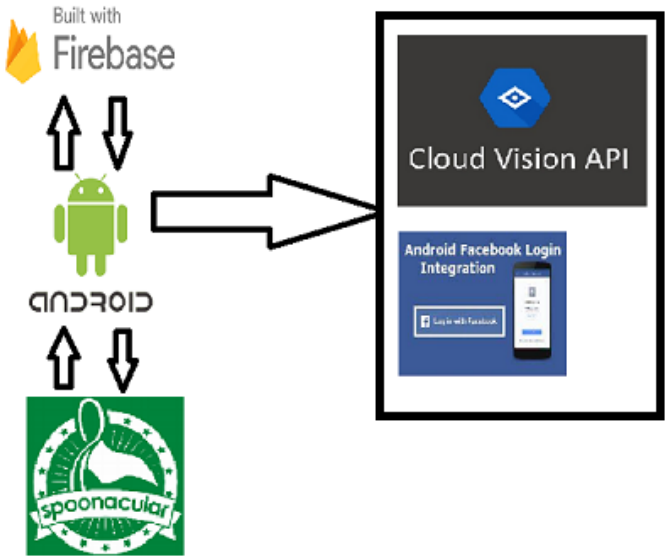


Figure 7: Use Case Diagram

9. ARCHITECTURE



9.1 Android Platform:

We decide to carry forward with the Android architecture, used by the previous team, to develop our application. This offered us some leverage for the time overhead. Also, it was a good decision made by previous team as Android phones are ubiquitous and hence are easily accessible making our application more feasible to use. Moreover, Android is JAVA and XML based which matched with our team skills rather than the iOS technologies like Objective-C or SWIFT. It is also much easier, quicker and cheaper to publish an application onto the Android Play Store than the Apple App store. Android Studio has a shallow learning curve,i.e., it is simple and easy to understand and has no license fee associated.

9.2 Firebase:

Unlike the previous team, who used Node.js and AWS, we preferred to use Firebase. Along with the fact that our team wasn't familiar with Node.js and noSQL database, Firebase provides us a more general platform for all our requirements. Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

9.2.1 Authentication:

We integrated our Android application with Firebase Authentication. We decided to use this as there are various tutorials available online and the Firebase documentation was easy to understand and follow. We followed this documentation to integrate the Email and Password based authentication. Firebase eases the task of maintaining authenticating the user and is secure at the same time. It also assigned a unique ID to all the user entries just like a Primary Key and made it easy for us to retrieve the information for the logged in user.

9.2.2 Realtime Database:

We needed to maintain two different database tables for application. The first table was used to store the user in-

formation like email and food preferences. This table was populated once the user registered for the application. The second table that we used was for our inventory management. This table was specific to each user and was populated or updated every time the user checks-in a new receipt or updated the inventory manually. Firebase and Android provided a smooth handshake for us to put the required data on the cloud securely, saving us the pain of deployment on a server. All this data was saved on the Firebase in json format based on the respective modal classes that we made in the Android project.

9.3 Google Cloud Vision API:

Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy to use REST API. It quickly classifies images into thousands of categories (e.g., "sailboat", "lion", "Eiffel Tower"), detects individual objects and faces within images, and finds and reads printed words contained within images. You can build metadata on your image catalog, moderate offensive content, or enable new marketing scenarios through image sentiment analysis. Analyze images uploaded in the request or integrate with your image storage on Google Cloud Storage. We explore the Google Vision API (image Content Analysis), used by the previous team, and experimented with it. The results we found were too appealing to ignore. We were able to scan and store the relevant food items in our inventory with high accuracy. It took us some time to understand and implement this API in our application using the Device Camera but once done it worked seamlessly. We also thought it would be a good idea to go ahead with this as it offers a lot of other functionalities that could be of some future use for adding new features in our application.

9.4 Login With Facebook:

The previous team used just this feature to let users login and still it had some bugs. Although, we implemented a whole user registration/authentication system, we thought it would be a good idea to keep this functionality keeping in mind the current trends in Android Application. We were able to successfully implement this feature in our login page following the instructions provided by Facebook Developer Documentation. Nevertheless, this functionality is down and couldn't be tested in our application due to the ongoing cases and accusation on Facebook involving Cambridge Analytica. Due to this they are undergoing some policy change and didn't allow us to login user based on their facebook account.

9.5 Recipe API:

Recipes dataset is a big part of our application. But where we get such a dataset is a problem. Possible solutions could be searching an exist dataset or using Recipe API. The previous team imported a dataset manually which was found on the Internet, but we found it insufficient after we added the searching recipes feature. After doing some research, we decided to use an Recipe API which provides perfect data information for our application. The Spoonacular Nutrition, Recipe, and Food API allows us to access over 365,000 recipes and 86,000 food products. Its food ontology and semantic recipe search engine makes it possible to search for recipes using natural language queries, such as "gluten free brownies without sugar" or "low fat vegan cupcakes." With this powerful API, we can find recipes for what's in the fridge, even based on special diets, nutritional requirements, or favorite ingredients.

10. MACHINE LERANING

10.1 Why we use machine learning?

In this project, one of the section is aimed to give users a list of recommended recipes, based on user's feedback on history cooked recipes, then next time when users hope to cook, we can directly give him the best choice recommendation of recipe depends on his/her previous preference, consider there will be more users and more data after the application go live, we finally decides to apply some machine learning model to do some data mining, according to the regularity we explore from the features, we can predict the label and give the recommendation.

10.2 Description of the dataset

Because for now, we cannot get enough data from users, however, in order to get a better prediction result, we need at least thousands of instances for training, so we download the data from Kaggle of recipe rating, which has 61878 instances of recipe, with 93 ingredient label as the features, the prediction label is 5 levels of favorite rate(from star1 to star5), so this regulate the format of data we need to collect in the future, after the users import the ingredients as well as the amount of each, we can translate that to this proper format then use it as a new instance.

	almond	amaretto	anchovy	anise	anniversary	anthony bourdain	aperitif	appetizer	apple
0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0
3	1	0	0	1	6	1	5	0	0
4	0	0	0	0	0	0	0	0	0
5	2	1	0	0	7	0	0	0	0
6	2	0	0	0	0	0	0	2	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	4	0
9	0	0	0	0	0	0	1	0	0
10	0	1	1	2	0	0	2	1	0
11	0	1	2	1	0	0	0	0	0
12	1	0	1	0	0	0	0	1	0

10.3 Selection of training model

For the training model, We first thought of using logistic regression to solve the problem, but the result proves that it is not applicable. After we analyze the reason, we found that this is not a linearly separable problem. So we applied a more complex model—Neural Network, and finally the result got a significant improvement, the final prediction accuracy rate has reached to 89.66%.

10.3.1 Logistic regression

Logistic regression is one of the most widely used machine learning models. It used for these scenario of giving certain conditions, predicts the occurrence probability of an event, the final result is yes or no with a certain probability. The goal of logistic regression is to give definite inde-

pendent variable X, which contains multidimensional mutually independent features, ie descriptions of various aspects: x1, x2..., xn, and outputs the predicted probability of Y is true or not. Logistic regression training data contains many samples, whose result Y is already known: X{(X1,Y1),(X2,Y2),...,(Xm,Ym)}. This method is supervised learning because the label label Y has been already known. The basic idea of logistic regression is developed from Linear programming model:

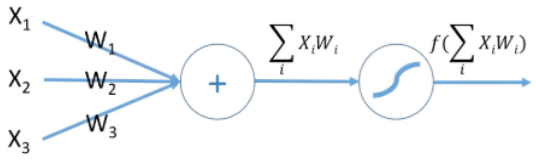
$$P[Y = True \mid X] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n$$

$\beta 0, \beta 1, ... \beta n$,is the parameters, we can also regard them as correlation coefficient, $\beta 0$ is intercept.
However, the range of linear programming can not always be guaranteed between 0 and 1, so for logistic regression, we calculate the Log Odds = log(P/(1-P)), it is proved that Log(Odds) has a better results in practical applications. So the equation of logistic regression is:

$$P[Y = True \mid X] = \frac{1}{1 + e^{-(\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_n x_n)}}$$

10.3.2 Neural Network

1957, Frank Rosenblatt came up with the idea of perception, the idea of perception is that by simulating the human brain, the computer model can achieve similar computational capabilities as the human brain. The components of neural network model are simple and convenient to optimize, which is also the cornerstone of deep learning. In contrast, the logical regression can only used for classification, however, neural networks can also used for image recognition, speech recognition, dialogue robots, neural networks composed of several perceptron, arranged in a layer, sharing the output of the previous layer, multi-layered perception combined together form a Neural Network, this is almost same as the relation of dots, lines and polygon in geometry.
Perceptron is the basic unit for processing information. The input includes multiple features, similar to the nerve endings in the brain. The output is a number, similar to the neurotransmitter substance.
For the calculations, first step is to input value with corresponding weights, calculate the weighted sum, then apply an activation function (generally a sigmoid function or other non-linear function such as tanh), and finally output a number.

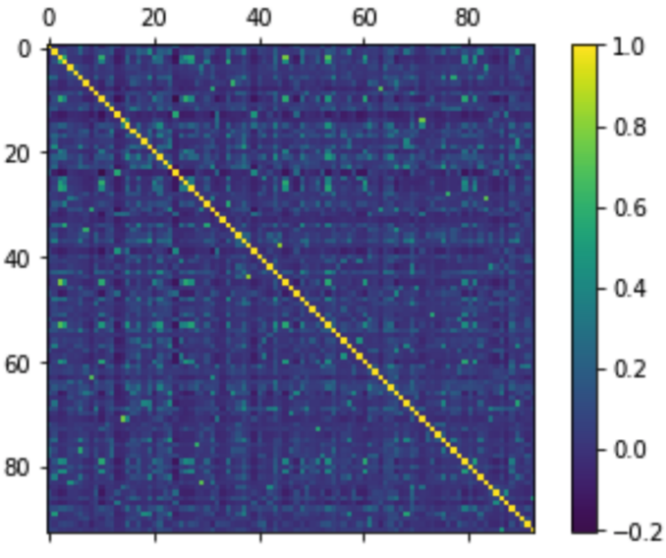


As we know, if we use sigmoid function as the activation function, the single layer perception model is totally same as logistic regression, that is because the weighted sum can only be valid for linearly separable data, but just like the famous XOR problem, the feature here is different ingredient as well as the quantity. Although they are mutually independent, they are not linearly separable, such problems are what multi-layer neural networks are good at. Classical neural networks are usually composed of input, output and hidden layers. Layers are independent of each other. According to the theoretical proof, if there is only one hidden layer in neural networks, any function(now matter it is linear or non-linear) can be implemented. The main reason is that the non-linear activation function makes the classification boundary can be nonlinear.

```
In [104]: metrics.accuracy_score(pred, Y_test_split)
Out[104]: 0.8965740142210731

In [109]: conf = metrics.confusion_matrix(pred, Y_test_split)
          [[ 171    2    2   35   86]
           [  10 4593  321  136   44]
           [   1   79  756   23    5]
           [  47   57   24 3094  135]
           [ 141   20    5  107 2482]]

In [119]: print metrics.classification_report(pred, Y_test_split)
                        precision    recall  f1-score   support

             1_star       0.46      0.58      0.51       296
             2_star       0.97      0.90      0.93      5104
             3_star       0.68      0.88      0.77       864
             4_star       0.91      0.92      0.92      3357
             5_star       0.90      0.90      0.90      2755

          avg / total       0.91      0.90      0.90     12376
```

10.4 Machine Learning Implementation:

First we use pandas to read the .csv data, extract the columns except for the label as the training set, then use numpy to change the label data to a list, because we hope to make the prediction of rating from star1 to star5, so we have to classify the original label 'Class_1' to 'Class_9' to 5 levels, then in order to make sure the independence between different features, we need to make a matrix to see the correlation between them.
Here is the result:



Then we apply the neural network model by using MLPClassifier from sklearn package, using l-bfgs algorithm as

the optimized solver, giving two hidden layers of size 30 and 10, then fit the model and predict the probability.
The result shows as following: prediction accuracy is 89.66%

	1_star	2_star	3_star	4_star	5_star
0	2.462135e-06	9.986035e-01	5.718270e-04	1.400962e-05	8.082007e-04
1	5.163455e-09	1.614420e-07	5.710840e-09	9.999995e-01	2.928715e-07
2	1.862937e-25	1.965242e-17	9.339008e-29	1.000000e+00	8.261362e-13
3	1.940393e-02	8.101533e-04	5.806284e-05	9.687620e-01	1.096583e-02
4	1.152660e-08	5.891328e-16	7.592577e-19	9.999999e-01	6.639831e-08
5	3.964182e-03	6.393486e-02	1.600755e-01	7.705898e-01	1.435656e-03
6	5.553133e-04	1.825502e-05	9.580262e-06	1.463421e-02	9.847826e-01
7	1.963124e-02	9.667525e-05	2.266653e-05	5.615588e-03	9.746338e-01
8	9.973647e-16	9.918587e-01	8.141317e-03	7.385135e-15	3.210888e-13
9	5.768602e-10	1.406031e-08	2.294906e-13	1.000000e+00	2.150869e-09
10	5.562944e-07	3.698698e-10	2.437355e-09	9.999498e-01	4.964349e-05
11	1.021288e-11	9.975597e-01	2.440208e-03	1.159072e-07	6.579960e-13
12	1.108138e-21	6.098562e-05	9.999390e-01	2.051321e-09	1.069923e-15
13	8.840086e-04	1.146794e-07	1.013295e-05	9.984594e-01	6.463708e-04
14	1.024048e-04	9.881076e-01	1.065141e-02	8.389350e-05	1.054737e-03
15	8.375964e-15	9.763805e-10	1.730619e-12	9.999982e-01	1.788073e-06
16	8.636982e-03	5.266465e-01	4.586884e-01	3.966359e-03	2.061825e-03
17	1.136029e-09	9.999808e-01	1.477923e-05	1.085718e-07	4.302537e-06

11. FEATURES:

11.1 Scanning Receipts:

Right now the application uses Google Cloud Vision API to scan the receipts and extract relevant information from it. We earlier planned to search for some substitutes for it like Project Oxford, Vize.ai, CloudSight, etc. and compare their performance to deliver a better user experience. On further experimentation, we found out that the Vision API performed far better than other possible substitutes. Still, the biggest roadblock is that this whole system isn't generalised and works on the receipts from a particular store. We need more time to use some algorithms and methods to overcome this problem and make our system independent of this impediment. We also propose on looking for some machine learning techniques which would help us in accomplishing this task. There could be multiple possible ways to achieve this and we need to check feasibility of each way individually.

11.2 Recipe Recommendation Engine:

Previously, the recipes database was been manually populated and contained only so many hand-picked recipes. We planned to build a "Pinterest" like database which would contain recipes from all around the world. Later on, due to lack of time and a rich database we had to stick with the Spoonacular API for this feature. We read the available ingredients from the Inventory table stored in Firebase and the user's food preference from the user info table. This is a new functionality that we implemented. Earlier, the preferences were not taken into consideration.

11.3 Recipe Search Engine:

We realized that it is not necessary that the user would want the application to suggest some recipe based on their inventory, sometimes they might want to look for a particular recipe that they want to cook. This gave rise to the new feature we planned on adding. This new search feature allows the user to search via the rich recipe database provided by the Spoonacular API for known recipes and get to know about required ingredients and the steps to cook a recipe. In the future it can be connected to the inventory database as well. This will let the user know about the missing/insufficient ingredients, if any. The user can then add those to the shopping cart from the same screen. This will help in preparing a shopping list of only the required products and the required quantity, which, in turn, will help avoid unnecessary shopping. This can help the user to save both their time and money.

12. EVALUATION

Once our mobile application was ready, we rolled it out to the users to gain feedback on the application and understand if they were satisfied with the application. We evaluated our Software based on following Metrics:

1. User Experience
2. Utility
3. Recommendation

The questions asked along with the results were as follows:

1. Were you able to register/sign up in the application?

   Users are unable to do anything unless they log in to the application using either their email address or via Facebook. Hence, this was an important question to check if users were able to use our application at all. Since Facebook has updated their policies recently, Facebook login is currently unavailable. All of the users were able to successfully sign-up and login into the app.

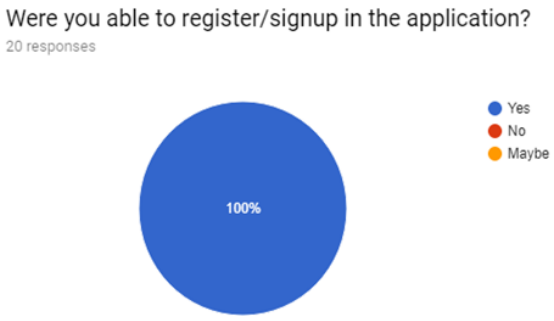


Figure 8: Login Success

2. Were you able to search recipes? This question is directly related to the core functionality of our app which we had proposed when the project initiated. Users should be able to search for any recipe they like. All of the users were able to search recipes through our application.

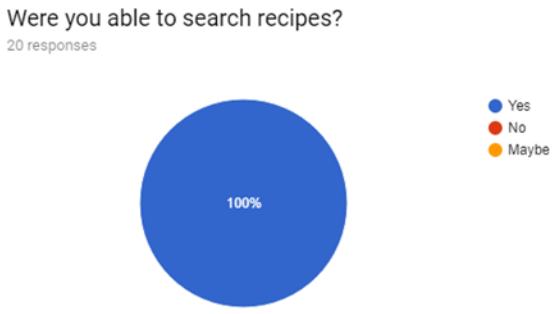


Figure 9: Searching Recipes

3. How would you like to rate the User Interface of the application? The user interface is the first impression of the application to the outside world, and it is extremely important to have a simple, easy-to-understand, and clean interface design. Over 90% of our users felt that the UI was impressive and clearly helped them navigate through the application.

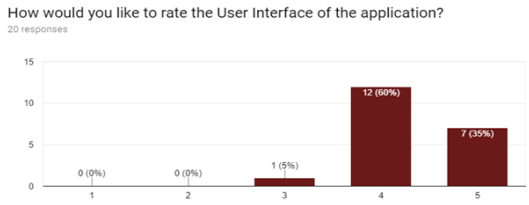


Figure 10: User Experience

4. How easily were you able to navigate through the application? Users tend to use the application lesser if they are unable to navigate easily. We wanted to understand if users had any difficulty in navigating through our application. Over 80% of our users were comfortable in navigating through our application. A few users found it difficult which is certainly an area to look into.

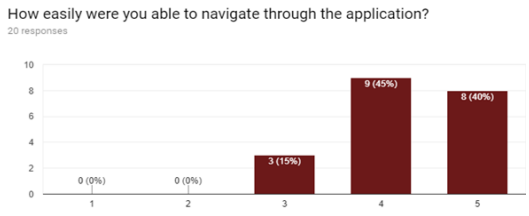


Figure 11: User Experience

5. Did you like the inventory manager that tracks your ingredients? We wanted to extend this part of the previous application as we found it to be an innovative and selling point of the cooking application. 75% of our users liked the inventory manager which helps in keeping track of the users' ingredients which previously user needed to keep a track of.

Figure 12: Inventory Management

6. Were the results of recommendations to your liking?

This is also a functionality of our application which would recommend the users with recipes based on their previous searches, preferences and ingredients list using machine learning approach. Majority of our users were satisfied with the functionality.

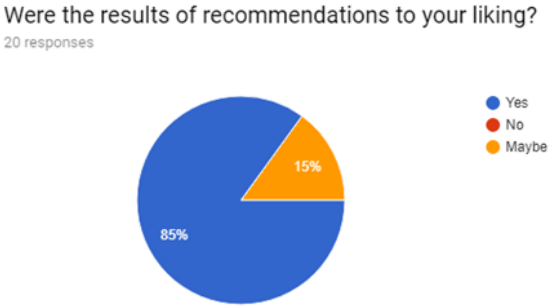


Figure 13: Recommendation Analysis

7. Did you select the suggested recipe? 95% of our users used the suggested recipes which were provided by our algorithm.

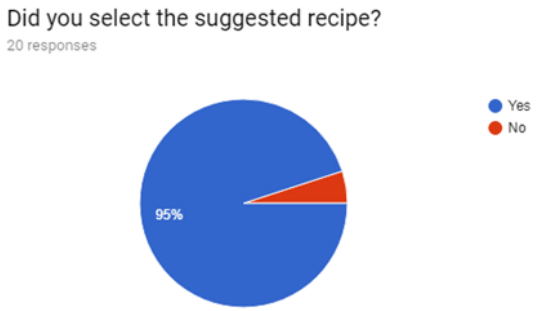


Figure 14: Did you select the suggested recipe?

8. Would you recommend this application to others? This question is directly related to the viability of the application as well as speaks of the acceptance of the application. For an application to be successful it needs a strong user base who are excited about the features of that application. 85% of the users felt that they could recommend our application to others.

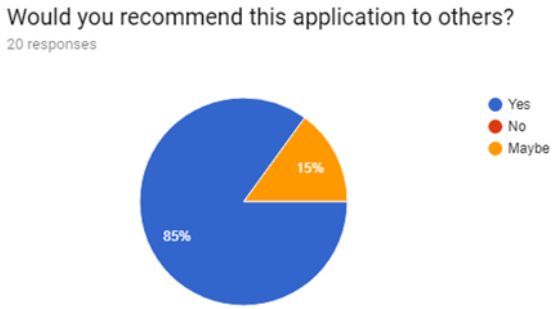


Figure 15: Feature Overview

9. What other features would you like in the application? We wanted to have an open-ended question to understand if users had any particularly useful features in mind that were missing from our application to determine the future scope of the application. The responses by users were extremely critical in analyzing and evaluating the work needed in future on the application.

13. CONCLUSION

previous team implemented several basic functionalities of their proposed application - kitchen assistant. However, given the limited period of time they got to work on their idea gave us a lot of room for improvements. From our proposed plan of actions we achieved the following:

- Improved the recommendation engine by including eating preferences, and even plan to take their health and bal- anced diet into consideration.
- Added a new feature to let the user search a recipe of their own choice, and plan to make a shopping list of the missing or insucient ingredients
- To adjust the quantity used and update the inventory for multiple servings of the same recipe.

We believe that given the time frame these functions were successfully imple- mented, the application will have a more complete workflow if few more modification are made in the future given surplus time. We covered most of the proposed basic use cases. If implemented to its full capacity and scaled appropriately, this application will be ready to be launched on Google Play Store.

14. FUTURE SCOPE

14.1 Allergy Considerations

Currently we are just taking eating preferences into consideration but not the allergies. We can also include detailed information of all the food products and ingredients so we can notify user if any particular recipe can cause some sort of allergic reaction upon consumption. This way the user can rely on our information and doesn't have to manually go to check all the contents of each product/ingredient.

14.2 Dietary Suggestions

We can add a new feature that could monitor the users dietary and eating habits. We can then use this information to suggest a healthy and balanced diets on a timely basis. To further build on this idea, we can ask user if they want their diet to be rich in a particular vitamin, protein, carbohydrates, etc.. or should not contain gluten, fat, etc.. I guess this could be really helpful for people who go to gym or aim at building muscles.

Moreover, we can maintain a list of favorite dishes which the user can access directly whenever they feel lazy or want to eat something special depending on the occasion.

14.3 Monthly expense budget

Now this is something that is different from the central idea of our application, as our application mainly focuses on optimal and quick use of resources to cook food and find recipes online. But we feel that this could be a good addition to our application, if we let our users monitor their monthly expenses and also see the types of products that were more expensive than others, then we can allow them to make an informed decision next time they go to buy groceries.

14.4 Integration with smart digital assistants

As suggested by the previous team, we definitely think that this would be a good feature to include in our application in the future. Google Assistant and Amazon Alexa are some interfaces that would make our application more convenient to use while cooking and also, would attract a larger user base.

15. REFERENCES

[1] Amazon Unveils Futuristic Plan: Delivery by Drone. CBS News. 1 December 2013. Retrieved 6 May 2014.

[2] Soliah LL, Walter JM, Jones SA. Benefits and barriers to healthful eating: what are the consequences of decreased food preparation ability?, Am J Lifestyle Med. 2012;6:152158.

[3]Jabs J, Devine CM. Time scarcity and food choices: an overview. Appetite., 2006;47:196204.

[4]R. Lappalainen, A. Saba, L. Holm, H. Mykkanen, M.J.Gibney, A. Moles. Difficulties in trying to eat healthier: descriptive analysis of perceived barriers for healthy eating, Eur J Clin Nutr, 51 (S2) (1997), pp. S36-S40

[5]Clausen A. Food CPI and Expenditures Briefing Room, Table 10. [Accessed 4-4-12];US Department of Agriculture, Economic Research Service, 2011

[6]Suzanne Higgs, Jason Thomas. Social influences on eating,Current Opinion in Behavioral Sciences,Volume 9, 2016, Pages 1- 6,ISSN 2352-1546,

[7]Poti JM, Popkin BM. Trends in Energy Intake among US Children by Eating Location and Food Source, 19772006. J Am Diet Assoc. 2011;111:11561164.

[8]Roberto A. Ferdman. Researchers have found a striking new side effect from eating fast food

[9]M. Pazzani, A Framework for Collaborative Content-Based and Demographic Filtering, Artificial Intelligence Rev., pp. 393-408, Dec. 1999.

[10]Xiaoyuan Su and Taghi M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques, Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009. doi:10.1155/2009/421425

[11]Jagithyala, Anirudh, Recommending recipes based on ingredients and user reviews, http://hdl.handle.net/2097/18154

16. CHITS

UAD, ACK, GLG, WSP, IPN, RJF, CPT, NIQ, HIP, RVB, EGN, XAD, GSP, LQA, OBK, FWH, NUU, DJC, WXB, XGW.