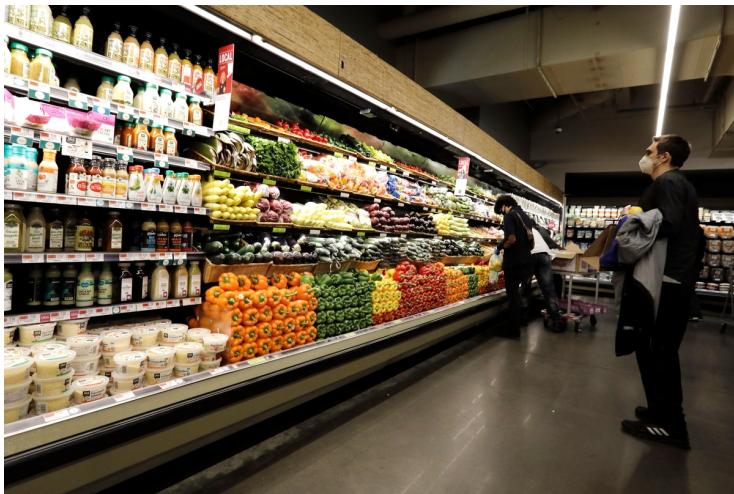


Smart Grocery Project Report



**Prepared by
Agnes Folga, Aditya Nadkarni, Gabriel Nicdao, & Megh Patel
for use in CS 440
at the
University of Illinois Chicago**

October 2022

Table of Contents

List of Figures.....	5
List of Tables.....	5
I Project Description	8-29
1 Project Overview.....	8
2 The Purpose of the Project	8-10
2a The User Business or Background of the Project Effort.....	8
2b Goals of the Project.....	9
2c Measurement.....	10
3 The Scope of the Work.....	10-14
3a The Current Situation.....	11
3b The Context of the Work.....	12
3c Work Partitioning.....	13
3d Competing Products.....	13-14
4 The Scope of the Product	15-18
4a Scenario Diagram	16
4b Product Scenario List	16
4c Individual Product Scenarios.....	17-18
5 Stakeholders	18-21
5a The Client.....	18
5b The Customer.....	18
5c Hands-On Users of the Product	18-19
5d Maintenance Users and Service Technicians.....	19
5e Other Stakeholders.....	20
5f User Participation.....	20
5g Priorities Assigned to Users.....	20
6 Mandated Constraints.....	21-25
6a Solution Constraints.....	21-22
6b Implementation Environment of the Current System	22-23

6c Partner or Collaborative Applications	23
6d Off-the-Shelf Software.....	24
6e Anticipated Workplace Environment	24
6f Schedule Constraints.....	24-25
6g Budget Constraints.....	25
7 Naming Conventions and Definitions.....	25- 27
7a Definitions of Key Terms	25
7b UML and Other Notation Used in This Document.....	26-27
7c Data Dictionary for Any Included Models	27
8 Relevant Facts and Assumptions.....	28
8a Facts.....	28
8b Assumptions.....	28
II Requirements.....	29
9 Product Use Cases.....	29-30
9a Use Case Diagrams.....	29
9b Product Use Case List	29
9c Individual Product Use Cases.....	30
10 Functional Requirements.....	31
11 Data Requirements	31-32
12 Performance Requirements	32-33
12a Speed and Latency Requirements.....	32
12b Precision or Accuracy Requirements.....	33
12c Capacity Requirements	33
13 Dependability Requirements.....	34
13a Reliability Requirements.....	34-35
13b Availability Requirements.....	34
13c Robustness or Fault-Tolerance Requirements.....	35
13d Safety-Critical Requirements.....	35
14 Maintainability and Supportability Requirements.....	35-38
14a Maintenance Requirements.....	35-36
14b Supportability Requirements.....	36

14c Adaptability Requirements.....	37
14d Scalability or Extensibility Requirements.....	37
14e Longevity Requirements.....	38
15 Security Requirements.....	36-
15a Access Requirements.....	38
15b Integrity Requirements.....	39
15c Privacy Requirements	40
15d Audit Requirements.....	40
15e Immunity Requirements.....	41
16 Usability and Humanity Requirements	41
16a Ease of Use Requirements.....	41-44
16b Personalization and Internationalization Requirements.....	42
16c Learning Requirements.....	42
16d Understandability and Politeness Requirements.....	43
16e Accessibility Requirements.....	43
16f User Documentation Requirements	43
16g Training Requirements.....	43-44
17 Look and Feel Requirements.....	44-45
17a Appearance Requirements.....	44
17b Style Requirements	45
18 Operational and Environmental Requirements	43-45
18a Expected Physical Environment.....	44
18b Requirements for Interfacing with Adjacent Systems.....	44
18c Productization Requirements	44
18d Release Requirements.....	45
19 Cultural and Political Requirements.....	45-46
19a Cultural Requirements.....	45
19b Political Requirements.....	46
20 Legal Requirements.....	46
20a Compliance Requirements.....	46
20b Standards Requirements.....	47

21 Requirements Acceptance Tests.....	47 -51
21a Requirements – Test Correspondence Summary	47
21b Acceptance Test Descriptions.....	47-51
III Design.....	52-64
22 Design Goals	52
23 Current System Design.....	52
24 Proposed System Design	52-54
24a Initial System Analysis and Class Identification.....	52
24b Dynamic Modelling of Use-Cases	53
24c Proposed System Architecture	53
24d Initial Subsystem Decomposition.....	54
25 Additional Design Considerations.....	55-57
25a Hardware / Software Mapping	55
25b Persistent Data Management.....	55
25c Access Control and Security	55
25d Global Software Control	55
25e Boundary Conditions.....	55
25f User Interface	56
25g Application of Design Patterns	57
26 Final System Design.....	57
27 Object Design	58
27a Packages.....	58
27b Subsystem I	58
27c Subsystem II.....	58
27d etc.	58
IV Project Issues.....	58
28 Open Issues.....	58
29 Off-the-Shelf Solutions	59
29a Ready-Made Products.....	59

29b Reusable Components.....	59
29c Products That Can Be Copied	59
30 New Problems	59
30a Effects on the Current Environment.....	59
30b Effects on the Installed Systems.....	60
30c Potential User Problems.....	60
30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product.....	60
30e Follow-Up Problems.....	60
31 Migration to the New Product	60
31a Requirements for Migration to the New Product.....	60
31b Data That Has to Be Modified or Translated for the New System	60
32 Risks.....	61
33 Costs.....	62
34 Waiting Room	62
35 Ideas for Solutions.....	62
36 Project Retrospective.....	62
V Glossary.....	63
VI References / Bibliography	63

List of Figures

Figure 1 -Testing Correspondence	47
Figure 2 . Classes for the Smart Grocery application.....	52
Figure 3 . Mapping	55
Figure 3. UI for Smart Grocery	56
Figure 4. UML Class Diagram for Smart Grocery.....	57

List of Tables

Individual Use Cases	29
----------------------------	----

I Project Description

1 Project Overview

The Smart Grocery application allows users to broaden their variety of food available to them by taking information from other users as well as available groceries in their region to recommend a meal that they can cook. The user can also browse recipes to brainstorm what they would like to eat. Given their past purchases and their preferences, a machine learning algorithm will be able to help suggest cooking suggestions to the user. We use multiple sources in this application, which include phone GPS, other user data, and barcode scanners.

Those who want to cook more, and learn new recipes are encouraged to use this application. The app would take the user's preferences and keep a record of what they have cooked previously.

2 The Purpose of the Project

We hope to encourage users to cook more, and see what others are cooking around them. This project will allow users to be creative, and remember what they cooked before.

2a The User Business or Background of the Project Effort

Besides users, possible clients of this application are different grocery stores, and different recipe websites. In order for the user to see what groceries are available for them to cook with, grocery stores in the vicinity of the user's location must comply and show their products. SmartGrocery is a great way for grocery stores to boost their revenue as users would be encouraged to cook their own meals and not order take out/ go to a restaurant. Also, companies which publish people's recipes would be a necessary client for our application. In order for users to obtain ideas for what to cook, the application will need to access recipes found online. Both the grocery and recipe business are necessary clients for our application. Also online chefs would be able to gain traffic and popularity if they were to post their recipes.

The grocery stores and recipe websites will allow us to pull the items they have available and their recipes for our application, respectively. The user will be able to see what groceries are available, purchase them, and cook a new recipe using the ingredients they bought, the recipe finder feature, and seeing what others around them have been cooking.

Motivation

With this application, users can eat healthier, cook more, find new things to cook, and see what other people are cooking in their vicinity. Our goal for this application is to allow users to shop smarter, discover new cuisines, and cook better.

Considerations

The problem we are trying to solve with the creation of this application is finding new things to cook using data from other users, grocery stores, and online cooking forums (recipes). While this problem is not life threatening or super serious, we are hoping to make our users' lives easier and encourage them to cook more.

2b Goals of the Project

Content

Ultimately, we came up with this project idea because such an application does not exist. As a group, we like to cook, but were unable to find an application that shows available grocery items, ideas for recipes, and see what others around us are cooking, all in 1 place.

Motivation

Our main goal of this application is to encourage users to cook more, try new recipes, and see what grocery options are available in their vicinity.

We plan to develop our application in a timely fashion, and to learn as much as we can doing it. We wanted to create an application that we ourselves would use.

Examples

We want our application to be able to get our user's locations, instantly.

We want to be able to encourage people to cook more.

We want to be able to generate more revenue for grocery stores.

We want to be able to predict accurate recommendations for our users given what they have cooked previously & their preferences.

2c Measurement

We will be able to measure if our goals are met by hearing from customers that they enjoy our application & use it to cook more, shop locally, and try new recipes. A true success would be the app user recommending the application to one of their friends.

We can also measure success by hearing the grocery stores being satisfied with our services and reporting an increase in revenue.

3 The Scope of the Work

The scope of the work is to teach users how to cook better, save people time, and

encourage them to shop more at their local grocery store. This project will only concentrate on users purchasing from the grocery store, using data the user's input themselves, their location, and possible recipes from online.

3a The Current Situation

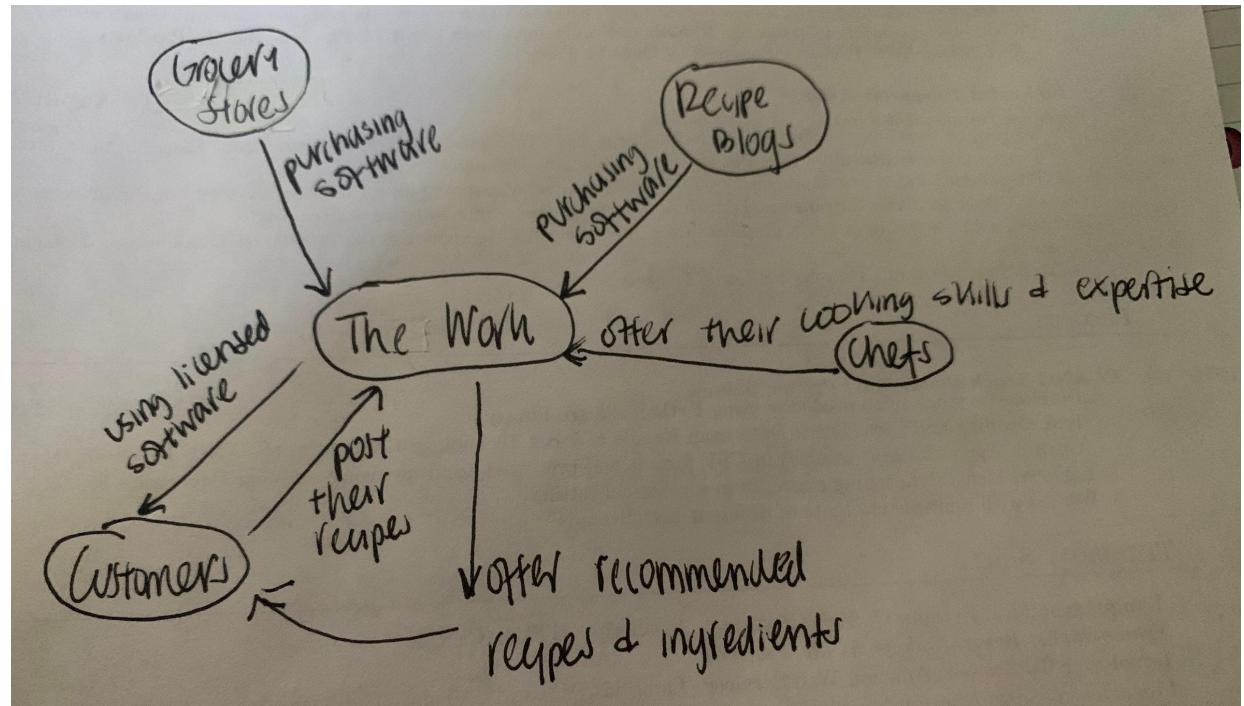
Content

Currently, users cannot buy groceries and find what recipes they can cook with them in the same application. They would need to see what groceries are available in their current vicinity, and then possibly decide what to cook or eat with them on a separate platform. Also, there is no application (that we know of) where users can see what others are cooking around them. Using the SmartGrocery application, all these steps are combined and can be done on 1 platform. This way, users can try new recipes, see what others are cooking, and eat out less. Current applications only take a users' location into account, they do not take other users' opinions for food options or ideas to cook.

Motivation

We wanted to create an application that allows users to perform the steps which we described above. This application enables users to eat out less by cooking meals more, and can find them a recipe for their favorite foods. The application will use machine learning to predict the user's preferences and foods they might like.

3b The Context of the Work



Considerations

Here we consider the customers who want to cook more, explore recipes, shop for ingredients, and post their own recipes. Chefs are different types of customers as they are not browsing for recipes, they are posting their own where the customers can comment.

3c Work Partitioning

Table 1- Work Events

Event Name	Inputs and Outputs	Summary
Grocery stores purchase software	Payment Details (Input) Location (Input) Product list (Output)	Grocery stores' create their accounts and show what inventory/ ingredients are available at a given location.
Recipe blogs purchase software	Payment Details (Input) Recipes (Output)	Recipe blogs create their accounts and will be able to advertise their content.
Customers sign up for an	User Details (Input)	Those who want to cook

account	Payment Details (Input) Interaction with Chefs (Output)	more will sign up for an account, input their location and food preferences. The payment details are only for when users want to buy groceries from the application.
Chefs sign up for an account	Chef details (Input) Access to post their cooking tutorials and interact with users (Output)	Chefs will be able to create their account, post their cooking tutorials, and see comments from other users.
Customers scan the barcode of their receipt	Barcode of grocery store receipt (Input) Suggestions for what to cook (Output)	Given the ingredients a user has already bought (barcode of the receipt), the user will receive a recommendation of possible recipes.
Customers receive recommended recipes	Their receipt ,food preferences, and what others by them are cooking (input) Recommended recipes based off our Machine learning algorithm (output)	Customers will receive recommendations of what to cook given our predictive Machine learning algorithm based on their prior preferences, ingredients list, and what others around them are cooking

3d Competing Products

Content

While there is no application exactly like SmartGrocery, there are a few which do mimic some of its features. First, Instacart and other independent grocery stores have their own applications where users can see what groceries and ingredients

are available. On some of these applications, delivery is an option, while that is not a feature of our application. Also, there are applications which allow users to find recipes, but not coinciding with what ingredients are in their vicinity at a grocery store. There are social media platforms where users can share what they cook, but not all in one application. Lastly, UberEats and DoorDash are 2 applications where users can have take out from their local restaurants delivered to their house.

Motivation

No platform on the market currently allows users to see what food they can cook given their grocery stores' availability by using their current location, and also sharing/swapping recipes with other users. We believe that our features sets our application apart from other ones available to users.

4 The Scope of the Product

The scenario describes a user that is new to the app and downloaded it because they don't want to deal with the headache of picking ingredients for a good price.

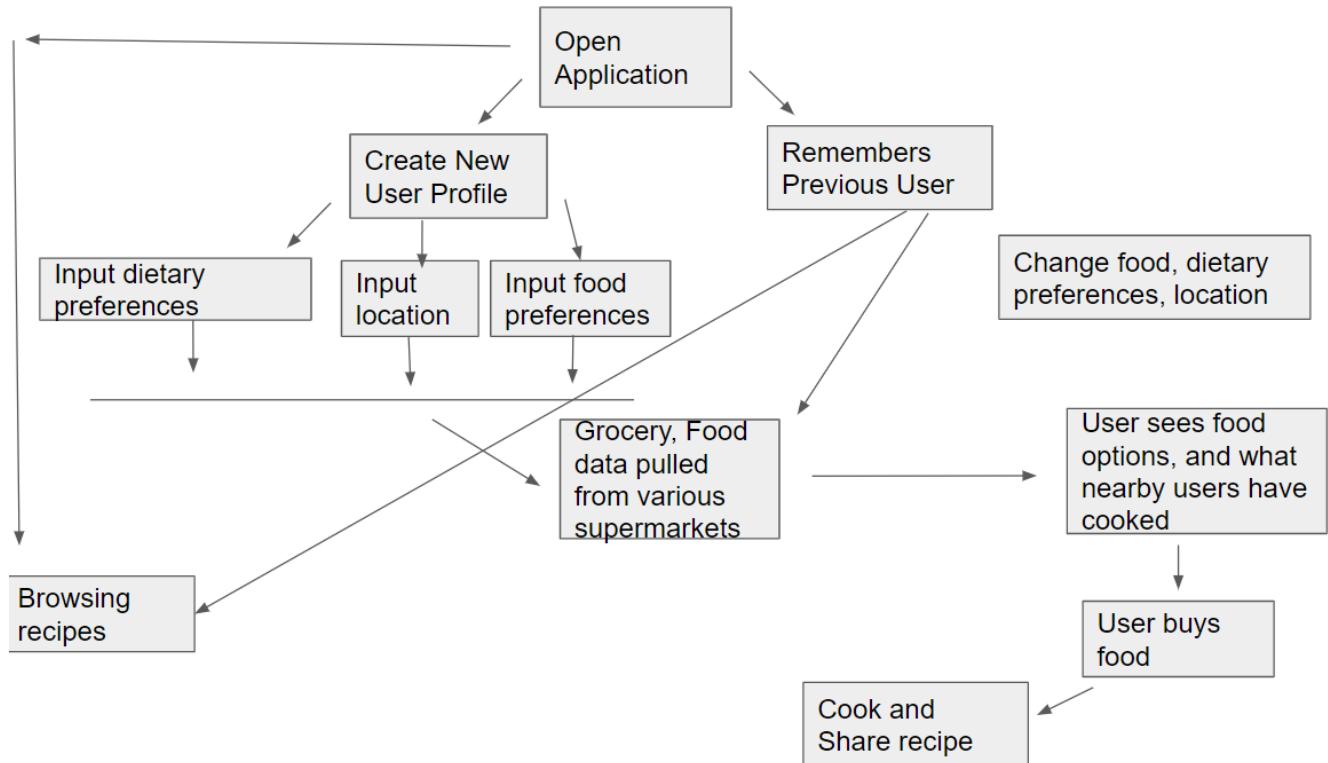
The User starts by creating an account on the application and then filling out a User Profile. This Profile will take into account dietary restrictions and personal preferences as well as Location. In the dietary restriction section, the user can describe what kind of food they cannot eat or prefer not to. In the personal preferences tab, the user can describe what kind of cuisine they prefer (American,

Italian, Thai, etc). They can also specify if they want more health conscious ingredients and recipes. Once these pieces of information have been inputted, the application will provide different types of recipes as well as a location of where to get the ingredients for a minimal cost. The user will choose whichever recipes they feel like eating for say a week and will be given a list of groceries as well as the location that offers them for the best price and best quality.

The option to log these entries will be provided. Once the user has used those ingredients to cook, they can rate the interaction for that set. Another user with similar preferences will be likely to recommend that meal as well. The ingredients will be updated with locations matching the new individual's recipe.

Another feature of the application is that the user can browse other users' recipes before cooking or getting groceries. In this feature, the user can plan ahead and get ideas of what to cook. This feature would include multiple options for people from quick and easy meals to make to more complex meals. Users would also be able to choose which dietary recipes they want.

4a Scenario Diagram



4b Product Scenario List

1. Create Account
2. Login
3. Input Dietary Information
4. Recommendations
5. Manual Browsing
6. Ingredient List
7. Purchase
8. Share Recipe

4c Individual Product Scenarios

Create Account: A new user that has never used the application will be prompted to create an

account to create a user that will store individual information. This application will be taking most of its information from gps and scanners that will save to a user profile that needs to be added into our system.

Login: If the user has created an account before, they would login to that account to access their profile to continue using the application. This would be the first step for most users that have already created an account but the application generally will remember the primary account on the mobile device to avoid logging in each time.

Input Dietary Information: The foundation of recommendations that will be provided to the user will be based on the initial information about the types of meals they would like to consume. If they have any form of allergy, or want to implement certain nutrients into their diet, this would be inputted here. This is the equivalent of building a user profile and giving it its unique value.

Recommendations: Based on information provided by the user individualized recipe recommendations will be provided to the user. These will be recipes that have good ratings from people with similar tastes in cuisine or similar dietary restrictions.

Search: For individual users that would like to take a more hands on approach to foods that they want to consume, the search feature will allow users to look for similar recipes to what they are searching for and provide reviews from other people. The user will then have the option to choose one of these and will be provided with an ingredient list.

Ingredient List: The application will sort through the websites of different stores and provide the location of which grocery store can provide the most ingredients for the cheapest total

cost. The user is then free to pick the location that best suits their needs.

Purchase: The user will perform a transaction to buy all of those items at the store which will not be recorded by the application.

Share Recipe: After the user has had the opportunity to purchase the ingredients as well as cook them into the recipe that was provided, they can rate the recipe and share it so that other people get recommended that recipe more.

5 Stakeholders

5a The Client

In our situation we will not have a client that we will go back to. This will be our own organization and we will manage and create this application. The reason is due to us using multiple stores API and information.

5b The Customer

Our customers will include the general public and will be open to anyone who has interest in cooking quick meals or other meals. The customer will be using this app to find receipts and cheap food for a tasty meal quickly. The customer will also have the option to get additional perks as future updates roll out.

5c Hands-On Users of the Product

Our hand-on-users include the customers and the people who want to use the app to

cook food and look up online recipes and also ones recommended by other people based on your region and also the professional cooks who want to include their videos on the platform.

The first user group and most important would be our daily users who just want to browse for recipes or look up foods in their areas that they can buy for a good price and see the stock availability. This user group will have similar features to the second user group in that they won't be able to post video recording, instead they will be able to post their recipes.

The second user group would be professional cooks or people who want to share their recipes as a video format with guided information and written following. This user group will also have the same features as the first user group however they will be able to post videos, and in these videos users will be able to make comments regarding their cooking.

5d Maintenance Users and Service Technicians

Maintenance is done specifically on smart grocery ends.. We will be updating our app and including new features that might come out in the future, we will also provide any customer service to people who have trouble with the app, however anything related to the food which includes spoiled food would be directed to the stores.

5e Other Stakeholders

Some other stakeholders would include professional chefs who would post their recipes, we

would also include a video option where one can post them cooking and explaining how to make the food. We will also include other experts like marketing experts to push our app to the masses. We will also be business analysts so we can make good decisions on future updates. Also we will have to team up with supermarkets.

5f User Participation

User participation will be needed once the core app is build, once the app is build user will have the option to input their recipes they have created and from there more and more recipes will be available for other users, however to encourage users to participate we will be inputting in default recipes along with other things, which include a basic start up tutorial

5g Priorities Assigned to Users

Key Users: Our key users would be our clients since without them our product would not be able to function how we intend it to. Without the information from the stores or the recipes it would be hard to impossible to use the smart grocery app

Secondary Users: Secondary would include anyone who would like to post their recipe to the app and share it with everyone, this would also include detailed information about the recipe and the store they used

Unimportant users: These people include our customers who just use the app for their benefit without posting anything.

6 Mandated Constraints

6a Solution Constraints

Description: The product will operate using Android and IOS mobile devices

Rationale: Users will be using the back facing camera to scan the receipts of the products they bought and also scan barcodes as well. This will be good since most phone cameras have technology that scans barcodes and also ingredients on the receipt

Fit Criterion: The product will be distributed in the app store and google play store

Description: The app will also operate using the location of user

Rationale: Users location will be used when collecting data based on region and recommending recipes.

Fit Criterion: User location must be enabled in order for the recommendation to work when using the app

Description: The app must implement a well running machine learning algorithm that analyze the ingredients bought and used on recipes

Rationale: users will be recommended recipes based on the region and user data.

Fit Criterion: The algorithm should given out recommendations based on user and region data and not some random recommendation.

Description: User will be able to post video content and recipes

Rationale: some user prefer watching videos compared to looking at text there we must

allow video sharing in the app

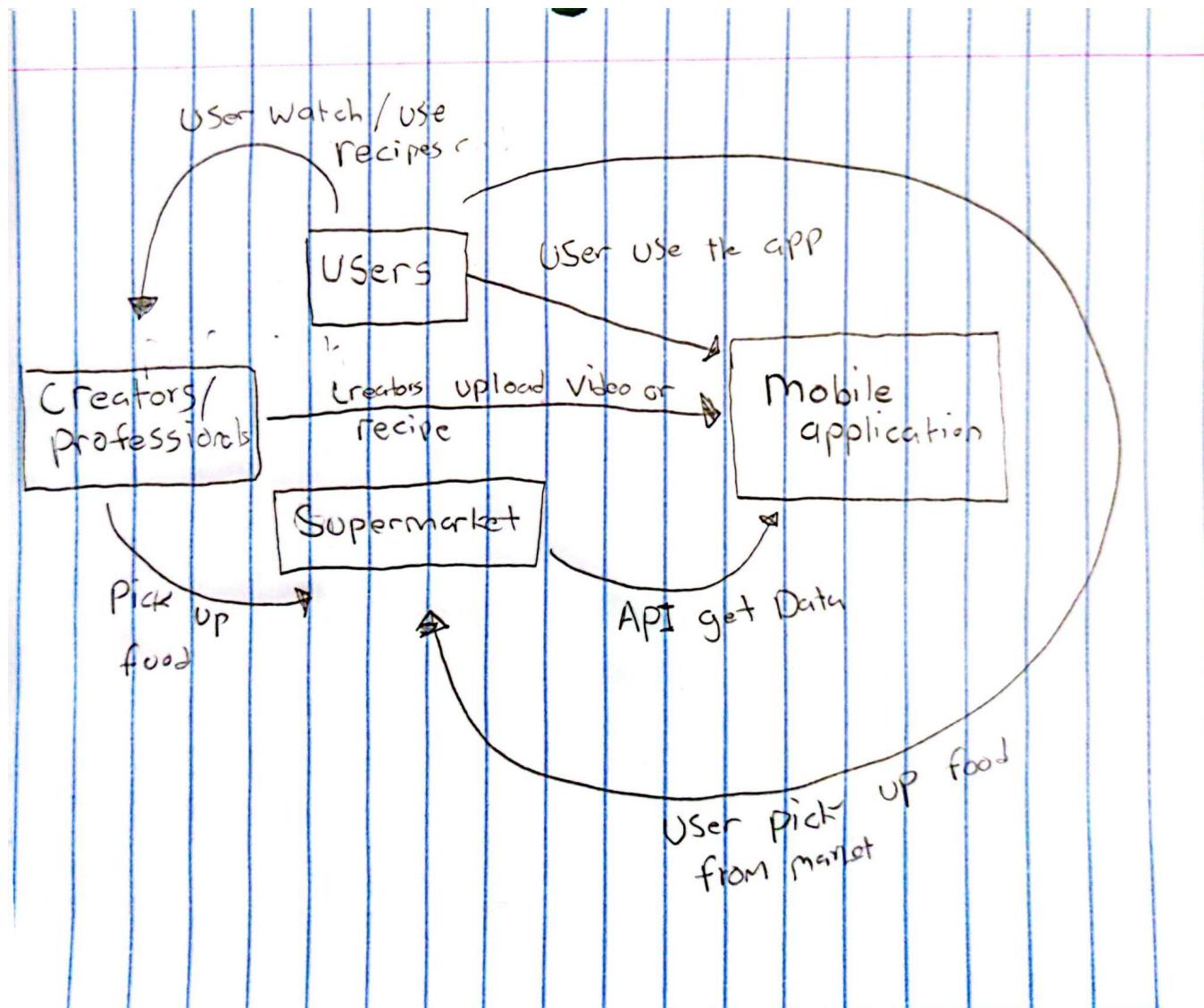
Fit Criterion: video upload and sharing should be stable and fast with no buffering

Description: User will be able to purchase foods for in person pickup

Rationale: since some users prefer ordering food online we will have a quick option to order ingredients based on what recipe the customer selected

Fit Criterion: order should be ready for pick up and ordering online should be done easily

6b Implementation Environment of the Current System



6c Partner or Collaborative Applications

Supermarket API - This will include any supermarket that has an API and they will provide us with inventory and other information about the products. Supermarkets include but are not restricted to Walmart, Target, and Amazon Fresh

Cooking website API - This will be needed when people browser for recipes in the

beginning to try out new recipes, however this will be limited

Google video API - this will be used for professional who want to upload there video to our platform and see how there video is doing

6d Off-the-Shelf Software

Smartphone

- This will include smartphones that have Android or IOS running on them because this product is a mobile app and also it will use feature for specifically on this device

Login API

- Login whenever using the app so you will be able to tell the user apart

6e Anticipated Workplace Environment

This product will be used in the kitchen or when looking up recipes at your home.

This product shows customers the recipes and also how to cook them so by standard most people who cook in their kitchen will look at this app, or when ordering food from their phones. There will also be a social aspect when regarding regular users and also professionals who upload videos about themselves cooking. Professionals and regular users will be able to communicate with each other in person or online.

6f Schedule Constraints

In order for the creation of the application we will need funding from investors, these will

include people who want to invest in our software and fund it. Once we receive the funding we can then start the development of the application. Also when professionals upload video content to our platform we expect that it will be cooking and useful for regular users if not and if it violates our rules we will forcibly remove that user and ban them from using our application. We will be monitoring these videos.

6g Budget Constraints

Most of the money will be spent on creating the application and also spreading the news about it so we will likely need to pay the developers and also the marketing team regarding this. Also with partners we might have to also pay them. When implementing the AI we might need a big database which will be hosted on the cloud so this will also cost some money. Other things will be needed to take into account are the software that we will be using and other tools. In total we expect our app to cost around \$250,000-\$400,000. This will be to pay the marketing team, developers and get tools.

7 Naming Conventions and Definitions

7a Definitions of Key Terms

Grocery Stores: Any brick-and-mortar store that sells produce and various ingredients items used to make food dishes.

Grocery Users or “Users”: A user of the service who is looking for various dishes they can potentially cook using ingredients they may purchase.

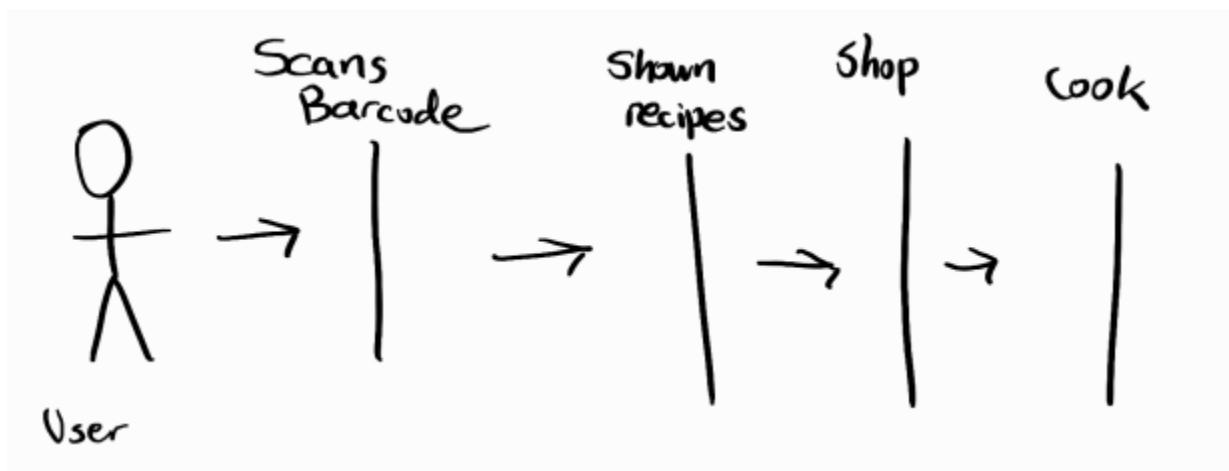
Recipe: A guide, which can be an instructional video or written guide, that shows a person how to cook a certain meal.

Meal: An amount of food that may be eaten, usually created by following a recipe.

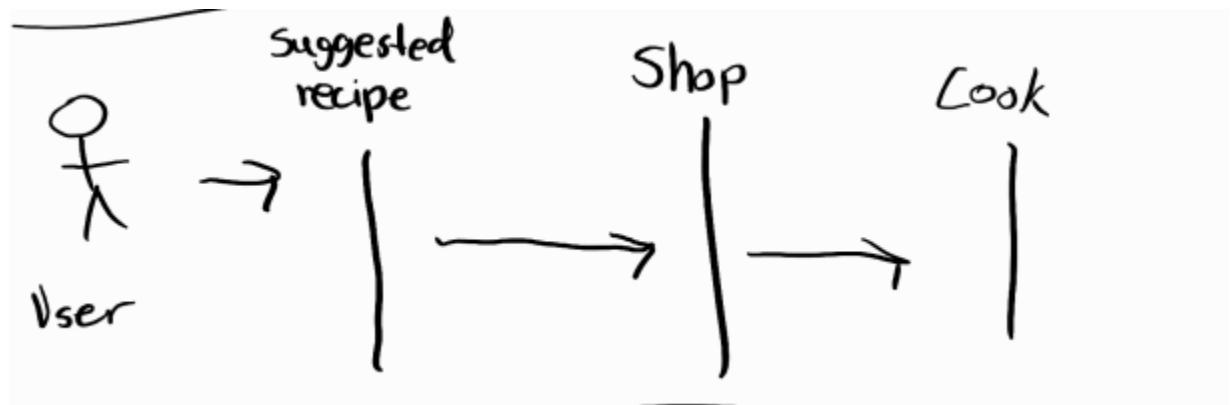
Barcode Scanner: Scanner that uses a mobile phone's camera to scan barcodes of ingredients.

7b UML and Other Notation Used in This Document

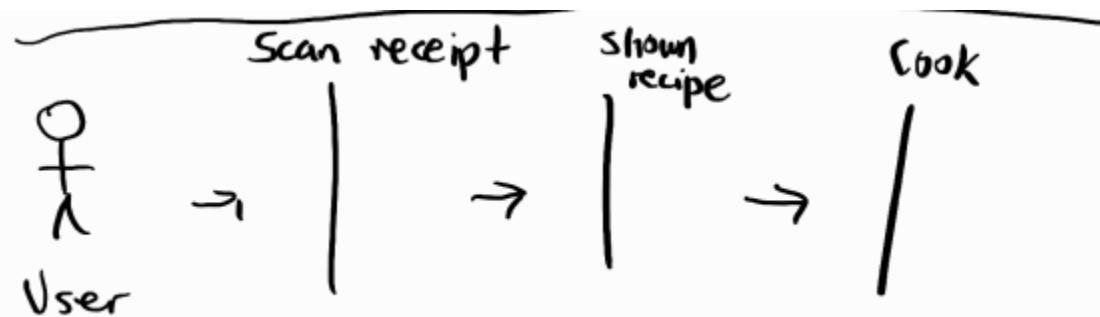
Sequence Diagram for Barcode Scan



Sequence Diagram for Recipe Suggestion



Sequence Diagram for Receipt Scan



7c Data Dictionary for Any Included Models

Grocery shopping

- Store visited
- Ingredients owned
- Ingredients to purchase
- Recipe chosen

8 Relevant Facts and Assumptions

8a Facts

- The average shopping trip takes 41 minutes. That is 53 hours per year.
- Sunday is the most common day to go grocery shopping, followed by Saturday.
- Americans spend over \$750 billion on groceries a year.
- Grocery stores have over 39,500 items on average.
- About 25% of grocery shopping dollars are spent on processed foods and sweets.
- About 36% of Americans cook at home daily.
- 1 in 4 people eat fast food every day.
- Americans consume 31% more packaged food than fresh food.

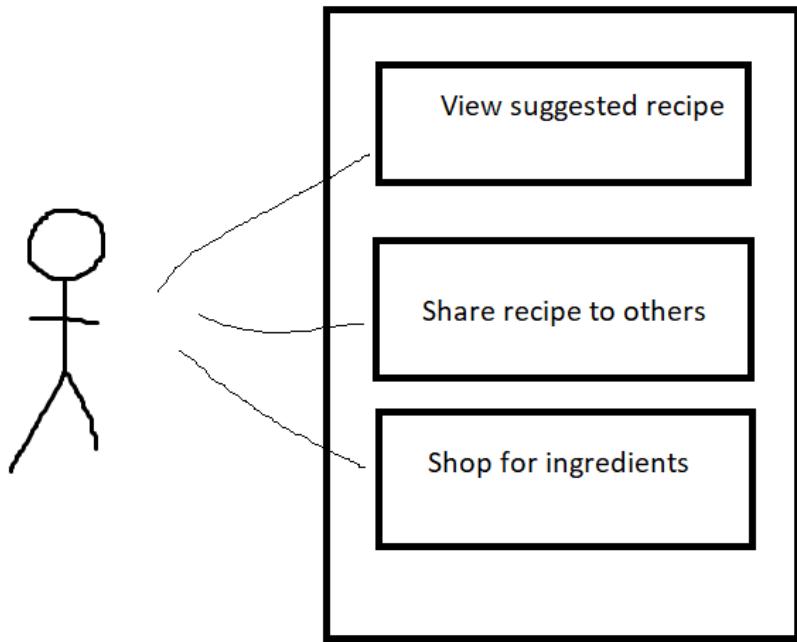
8b Assumptions

- Users of the application will own or have access to a mobile device which utilizes an internet connection, GPS, and a camera.
- The user will have access to a kitchen or cooking area.
- The user will be able to afford most cooking ingredients.
- The user lives in an area where there are grocery stores nearby.
- The user will have a little bit of cooking experience to follow the recipes.

II Requirements

9 Product Use Cases

9a Use Case Diagrams



9b Product Use Case List

Not Applicable

9c Individual Product Use Cases

Use case ID: Following Suggested Recipe

Name: Recipe

pre-conditions: User must select a recipe to make

post-conditions: User logs that they have made the meal

Initiated by: User

Triggering Event: User selects a recipe that they want to make

Additional Actors: N/A

Sequence of Events:

1. Application suggests recipe for user
2. User decides to cook recipe, therefore logging it
3. Application suggests ingredients at local stores / delivery
4. User acquires ingredients and cooks recipe
5. User can share photos of meal to other user's feed (optional)

Alternatives: N/A

Exceptions: If the app closes while displaying recipe instructions, the application will mark the meal as uncooked and mark the recipe as "unread", this is assuming the recipe has been abandoned

10 Functional Requirements

1 - Machine Learning

Description: A machine learning model to learn about the user's taste in food.

Rationale: ML model will help bring relevant recipes to the user

Fit Criterion: Machine learning must be compatible with the application.

Acceptance Tests: Machine learning test

2 - Users (Cooks)

Description: Cooks will be able to read suggestions on recipes and decide whether or not they need to follow them

Rationale: At-home-cooks will find it easier to decide what to cook

Fit Criterion: Cooks must have their recipe history and taste in food stored inside the database

Acceptance Tests: Cook Test

11 Data Requirements

3 - Scanned Items

Description: We will need the items that a user brought in order to give recommendations (ML).

Rationale: as we get more information and our models learn more we will be able to give our better recommendations to users

Fit Criterion: The items scanned will be converted into useable data and stored in a database

Acceptance Tests: Recommendation test

4 - User Data

Description: User data is needed for specific tasks

Rationale: user data will be used for logging into the application, also using it for recommendation of recipes and other features such as chat feature

Fit Criterion: The user data will be stored in a database on creation of an account or when user updates profile later

Acceptance Tests: Create an account test

5 - Recipes

Description: Data is needed for ML algorithm when recommending recipes

Rationale: Needed for the ML algorithm, since we are recommending recipes it is obvious we need data for what type of recipes.

Fit Criterion: The recipes will be stored in a database based on what type of recipe it is for example vegan, non-vegan, and vegetarian and also what region the food comes from, for example Italian, Chinese, etc. Also ingredients information will be needed

Acceptance Tests: Recipe Test

6 - Professional Cook information

Description: Professionals who upload their videos to the platform , we will need to store this data

Rationale: Based on recommendation we will be able to share to users which professional is close to their region and recommend video to them based on their preferences.

Fit Criterion: The videos and the professional information will be stored in a database.

Acceptance Tests: Video test

12 Performance Requirements

12a Speed and Latency Requirements

7 - Creation of account

Description: when a user creates an account we expect that all the data will be stored in the database in seconds.

Rationale: user will not be able to enjoy the experience if the app takes a long time for the user to be created

Fit Criterion: User data should be stored in seconds in the database

Acceptance Tests: Account Test

8 - Recommendations

Description: we expect that the machine learning algorithm will return a couple of recommendations to the user. This should be done in a few second or on page load after scanning items

Rationale: Having a fast machine learning algorithm will help people get recommendations that suit them and people will be able to see these recommendations more easily and quickly.

Fit Criterion: The machine learning algorithm should be done giving a recommendation based on the ingredients in less than 3 seconds.

Acceptance Tests: ML Test

12b Precision or Accuracy Requirements

9 - ML recommendations

Description: The machine learning algorithm will need to give out good recommendations based on the user. There are prior works done on predicting preferences using machine learning, such as transfer learning [1].

Rationale: The machine learning algorithm will also provide recommendations based on your preferences and users will also be able to get recommendations based on their region and who is cooking what around them.

Fit Criterion: The machine learning algorithm must output meaningful recipes based on one ingredient and should be accurate 99 percent of the time..

Acceptance Tests: ML output Test

12c Capacity Requirements

10 - Database for User

Description: The database will hold multiple tables about the user and their information

Rationale: The database needs to be large enough to hold in information about users and this database should be large in size.

Fit Criterion: Information will be inserted into the database and we will also see if we can delete it.

Acceptance Tests: Database user Test

11 - Database for ingredients

Description: The database will hold multiple tables about the user and their scanned items which will be used for Machine learning.

Rationale: The database needs to be large enough to hold in information what user it is and all the ingredients they have scanned.

Fit Criterion: Information will be inserted into the database and we will also see if we can delete it.

Acceptance Tests: Database ingredients Test

13 Dependability Requirements

13a Reliability Requirements

12 - 24/7 uptime

Description: The application should be available to regular users and professional users 24/7 of the time. We expect this because we know people will be on our app looking for recipes and loosing connection will hinder our goal for a good product

Rationale: The application must be on when scanning users ingredients and from there giving recommended recipes.

Fit Criterion: We expect the system to go down every 3 months, so we can update and maintain the app.

Acceptance Tests: Reliability Test

13b Availability Requirements

13 - Scheduled Maintenance

Description: The system must go down only during the scheduled maintenance of the application.

Rationale: if users use the application they will get an error message and professionals will not be able to view their video.

Fit Criterion: The System will go through maintenance at least four times in a year.

Acceptance Tests: maintenance Test

13c Robustness or Fault-Tolerance Requirements

14 - Scanned Item recommendations offline

Description: Even in a situation where the system goes down to expect that user will get recommendation based on what ingredients they input, however the scope of recipes will be limited to data based on their own previous recipes they made.

Rationale: based on what ingredients are scanned it user will receive recipes that they can make

Fit Criterion: Downtime will only have four time a year

Acceptance Tests: offline ML test

13d Safety-Critical Requirements

15 - Health

Description: The user will have to sign a agreement that the application has no control over an user health

Rationale: Smart grocery help individuals with cooking not with diet management

Fit Criterion: The user will receive the recipes and also recommendations based on what they previously bought and people near the region

Acceptance Tests: Health Check Test

14 Maintainability and Supportability Requirements

14a Maintenance Requirements

17 - Database Management and Cleaning

Description: The database must be managed appropriately. It must be cleaned and updated every 1-2 days (24-48 hours). It will remove duplicate accounts & accounts that have not been active in over 2 years.

Rationale: By managing the database, there will be user satisfaction & more space for new, active users.

Fit Criterion: Cleaning will occur every 1-2 days at 12 AM CDT.

Acceptance Tests: Database Management Test

18 - Inventory Management

Description: Stores must update their inventory to ensure the items available on the app are actually in store. Inventory must be managed properly and updated every 1-2 days (24-48 hours). New products must be added to the application, and ones that are not must be removed.

Rationale: In order for customers to be able to see updated inventory and the most available products, the store will update inventory.

Fit Criterion: Stores will update their inventory every 24-48 hours at 12:00 AM CDT.

14b Supportability Requirements

19 - Technical Support

Description: There should be 24/7 technical support for store owners, chefs, and regular users of our application, in different forms such as phone, chat, and email. If there are potential issues with the Smart Grocery application, we will be able to address them quickly by having technical support.

Rationale: There may be potential issues with our application so by having good technical support we can solve them quickly and efficiently.

Fit Criterion: Users must be able to have available support and should not wait more than 5-10 minutes to get connected with a specialist. At the end of the support session, the user should feel like they should be able to use the application as designed.

Acceptance Tests: Tech Support Test

20 - Website Option

Description: A website that works on desktop and mobile should be available for users that prefer a website or in case the application is down.

Rationale: Users will be able to access the cooking videos and chat function on their computer, or on a website on their mobile device. Chefs will be able to type out recipes and edit their cooking tutorials and post directly from their editing devices. Stores will be able to post their inventory and accept payment on their own devices.

Fit Criterion: The website should work on all operating systems, computers, and mobile devices.

Acceptance Tests: Website Option Test

14c Adaptability Requirements

21- Operating System

Description: Smart Grocery must work on all operating systems and devices so that anyone can use our product.

Rationale: In order to ensure that everyone can use our product, it will work on all operating systems and devices.

Fit Criterion: The website should work on all operating systems, computers, and mobile devices.

Acceptance Tests: Operating System Test

14d Scalability or Extensibility Requirements

22- Database Scaling

Description: Given there may be an increase in the amount of data loaded from the users as well as to handle all traffic on the mobile application & website, the database has to be scaled to demand.

Rationale: In case of a great increase in users rapidly or spontaneously, the database and application must be able to handle a high capacity of traffic & have enough storage space.

Fit Criterion: Scaling should double in size as needed, and the database must be able to scale when the maximum capacity is reached.

Acceptance Tests: Database Scale Test

23 - Inventory Scaling

Description: Given there may be an increase in the amount of products loaded from the stores, inventory has to be scaled to demand.

Rationale: In case of a great increase in products, the store's inventory and application must be able to handle a high capacity of input & have enough storage space.

Fit Criterion: Scaling should double in size as needed, and inventory must be able to scale when the maximum capacity is reached.

Acceptance Tests: Inventory Scaling Test

24 - Machine Learning Scaling

Description: A machine learning algorithm must be flexible to learn new ingredients and cuisines without issue so that the system can give users good recommendations.

Rationale: This feature allows Smart Grocery to keep its recommendations diverse and help users to try new cuisines as well.

Fit Criterion: Machine learning software must be compatible with the application.

Acceptance Tests: ML Scaling Test

14e Longevity Requirements

25 - Smart Grocery Longevity

Description: The longevity of Smart Grocery depends on stores, chefs, and regular users being involved in our application, and using it. If the stores see an increase in sales and people buying their products, they will continue the contract with Smart Grocery.

Rationale: We will update and provide users with the best functionality and experience in order to retain user satisfaction and engagement.

Fit Criterion: The application process will be iterative - we will continuously update it & try to keep the user's engagement and exceed their expectations.

Acceptance Tests: User Engagement Test

15 Security Requirements

15a Access Requirements

26 - Chef Access

Description: Chefs will be able to post videos and their recipes. They will be able to also interact with users who are interested in their cooking tutorials.

Rationale: In order to gain communication with their cooking community, chef's will be able to access the name of the user's account.

Fit Criterion: Chefs only have access to the user's name.

Acceptance Tests: Chef Access Test

27- User Access

Description: Users will be able to see the store's availability, options, and recipes. They also will be able to see the chef's and the video recipes they post.

Rationale: Users can see what the chef posts, and interact with them. They also can see the availability of the product's in a particular store.

Fit Criterion: Users only have access to the store's inventory, and the chef's content they themselves post. If the chef interacts with them, they can engage also.

Acceptance Tests: User Access Test

28 - Store Access

Description: Store's will be able to see the name of the user's buying their food. They will be able to handle the purchasing of their product's using our data protection protocols.

Rationale: Store's can see the name of the customers buying their food, and the transaction associated with it.

Fit Criterion: Store's can only see the name of the customer's buying their food, and the payment method.

Acceptance Tests: Store Access Test

15b Integrity Requirements

29 - Password Encryption

Description: All user accounts (store, general users, chefs) will be encrypted within the database.

Rationale: All passwords will be encrypted so that no user data is compromised.

Fit Criterion: Before being added to the database, all passwords must be encrypted.

Acceptance Tests: Password Encryption Test

30 - Data Authentication Encryption

Description: While verifying account data with the store the user buys groceries with, verification data will be encrypted.

Rationale: This is to prevent user account data that comes from the store's database from being compromised.

Fit Criterion: All verification data must be encrypted while the application is interacting with the store's database.

Acceptance Tests: Verification Data Encryption Test

31 - Duplicate Account Prevention

Description: All users that register with the application must register using a unique email and username.

Rationale: In order to keep database integrity and only have 1 account registered to 1 email.

Fit Criterion: If there is an account registered with an email, do not store a duplicate account in the database.

Acceptance Tests: Valid Credentials Test

15c Privacy Requirements

32- Privacy Policy

Description: All data will be stored and protected with compliance to relevant privacy laws.

Rationale: All user information should be secure in compliance with privacy laws and maintain app credibility.

Fit Criterion: General access to user information will be restricted.

Acceptance Tests: N/a.

34- User Privacy Policy

Description: All users of our application (general users, chefs, stores) must read and agree with our privacy policy before creating an account. We will explicitly state what data is collected and why it is needed for our application. If there is a change to any policies, users will be notified.

Rationale: We do not want to collect user data without consent of the user.

Fit Criterion: All users must agree to the privacy policy before registering.

Acceptance Tests: User Privacy Policy Acceptance Test

15d Audit Requirements

N/A

Description: N/A

Rationale: N/A

Fit Criterion: N/A

Acceptance Tests: N/A

15e Immunity Requirements

35 - Network Security

Description: In order to protect it from cyber attacks, the system will have a firewall to protect it.

Rationale: This will prevent any third-party attacks & viruses against our application.

Fit Criterion: There is a firewall system installed that can protect our systems.

Acceptance Tests: Network Security Test

36 - Database Backup and Encryption

Description: Our application's database will be encrypted and will be backed up to a secondary database on the cloud in case the data needs to be restored.

Rationale: The data encryption will prevent data breaches and ensure user data is protected and stored inside the database. In case the primary database is compromised/fails, there will be a backup of the data.

Fit Criterion: The database is encrypted and backed up to a secondary database, preferably using a cloud database that utilizes pods that can auto-backup when needed.

Acceptance Tests: Database Backup and Encryption Test

16 Usability and Humanity Requirements

16a Ease of Use Requirements

37- Easy Interface Rule

Description: The interface must be easy enough to navigate that anyone above the age of 12 can use and understand it.

Rationale: For improved workflow and ease of use.

Fit Criterion: Any person or user should be able to understand and navigate the interface.

Acceptance Tests: Usability Test

38 - Six Click Rule

Description: Any action or user task should not take more than 6 steps in the application.

Rationale: For improved workflow and ease of use.

Fit Criterion: No more than 6 clicks to complete an action in the application.

Acceptance Tests: Clicks Test

16b Personalization and Internationalization Requirements

39 - Language Setting

Description: Because the application development will take place in the United States, the application will use American English as its primary language.

Rationale: Majority of our users understand American English, and prefer that dialect over British English.

Fit Criterion: The user will understand the text in the application.

Acceptance Tests: N/a

16c Learning Requirements

40 - System Training Model

Description: The System must teach how to cook the recipe before the user cooks it.

Rationale: The user needs to know the steps of how to make the recipe before cooking.

Fit Criterion: The application must provide a video or list of instructions for the recipe.

Acceptance Tests: Recipe Check Test

41 - System Training Model

Description: The System must have a Question and answer forum page where users can find the most asked questions and their answers.

Rationale: Every single question on the user forum must have at least 1 solution/answer.

Fit Criterion: The users will be able to ask questions and answer them in the discussion forum.

Acceptance Tests: Question Check Test

16d Understandability and Politeness Requirements

42 - Simple Language Rule

Description: The language used in the application must be easy to understand and easy to follow.

Rationale: All users should have no trouble understanding the words and phrases in the application.

Fit Criterion: No slang or idioms used in the application.

Acceptance Tests: Language Test

16e Accessibility Requirements

43 - Voice Assistant Mode

Description: The application must contain a voice assistant which can read the instructions of the recipes to the user.

Rationale: The product system must be available to everyone.

Fit Criterion: The user must be able to set the volume of the voice assistant.

Acceptance Tests: n/a

16f User Documentation Requirements

44 - Installation Instructions

Description: The application must have an installation instruction document which guides the user through setup.

Rationale: The installation of the application must be easy to understand and be fast.

Fit Criterion: n/a

Acceptance Tests: n/a

16g Training Requirements

45 - Chef Training

Description: Every chef must undergo training on how to post videos and recipes, and how to interact with users.

Rationale: The chefs must be familiar with the product functionality because they will be able to gain popularity.

Fit Criterion: The chef must post a recipe or video to the application within the first week or they need to complete training again.

Acceptance Tests: Chef Recipe Test

46 - Store Training

Description: Every store must undergo training on how to post their products and how users can make purchases through our application.

Rationale: The stores must be familiar with the product functionality because they will be able to increase their sales and profits.

Fit Criterion: The store must show their inventory/products availability in the first week or they must go undergo training again.

Acceptance Tests: Store Training Test

17 Look and Feel Requirements

17a Appearance Requirements

47 - Color Scheme

Description: The color scheme of the application must follow the application logo, which is red & yellow

Rationale: Using consistent, vibrant colors in our application will enhance the user experience.

Fit Criterion: The application will have mostly red & yellow in the background and widget colors, with neutral colors to complement, to make sure not to overwhelm the user.

Acceptance Tests: Color Test

48 - Font Size

Description: The font used in the application must be easy to comprehend for the user.

Rationale: If we use a cursive font that doesn't follow conventional standards, users might not be able to read the application (i.e. instructions, sign in , etc). This is unprofessional and unacceptable.

Fit Criterion: Arial or Times New Roman font will be used in the application.

Acceptance Tests: Font Test

49 - Font Color

Description: The font used in the application must be a color that is easily readable for the user.

Rationale: If eccentric and non neutral colors are used, users might not be able to read the text in our application. This is unprofessional and unacceptable.

Fit Criterion: Besides white, the font color can be anything neutral besides white.

Acceptance Tests: Font Color Test

17b Style Requirements

50 - Style Guidelines

Description: Given the guidelines in the previous section, we want our application to be modern , professional, and interactive.

Rationale: We want our application to stand out from other applications on the market, but want it to be credible and professional. Users will be buying and handling food using our application, and we want to create an interactive experience.

Fit Criterion: the requirements in 17a have been met.

Acceptance Tests: n/a

18 Operational and Environmental Requirements

18a Expected Physical Environment

51 - Works in Store Environment

Description: The application shall be used by stores by posting their availability.

Rationale: The application should work even in a store that might have spotty service.

Fit Criterion: The application should work at any food store even if it does not have good wifi, and is loud.

Acceptance Tests: N/a

52 - Works in Home Environment

Description: The user will browse food availability, recipes, and chef tutorials in their home where they can ultimately cook their food.

Rationale: The application should be adjusted to any brightness levels based on natural lighting or LED lights in the user's home.

Fit Criterion: The application should work at any food store even if it does not have good wifi, and is loud.

Acceptance Tests: n/a

18b Requirements for Interfacing with Adjacent Systems

N/a

Description: N/a

Rationale: N/a

Fit Criterion: N/a

Acceptance Tests: N/a

18c Productization Requirements

53 - Product Weight

Description: The weight of the application cannot be greater than 100 megabytes.

Rationale: The application should not slow down the user's mobile phones or web browser.

Fit Criterion: N/a

Acceptance Tests: N/a

18d Release Requirements

54 - Update to System

Description: Unless there is a critical bug that needs to be fixed, the application should be updated monthly.

Rationale: A month is enough time to develop new tests for the software, and create new features.

Fit Criterion: Monthly maintenance ensures that the application works fast and efficiently for the user.

Acceptance Tests: N/a

19 Cultural and Political Requirements

19a Cultural Requirements

55 - Cultural Variety

Description: The product will be recommending multiple cultural cuisines and people will be able to recommend the cuisines of other users.

Rationale: Due to the nature of the application tracking user's food habits given certain dietary restrictions, it would not be uncommon for many people to eat a specific cultural cuisine more frequently.

Fit Criterion: Users can look at other reviews from other people on different cuisines which gives an opportunity to broaden their food variety if they want to make it.

Acceptance Tests: N/a

19b Political Requirements

56 - Political Correctness

Description: The application will not be offensive to any culture, food, or user.

Rationale: The application is friendly and open to everyone, and shouldn't push users away through any offense.

Fit Criterion: Comments on food will be regulated to avoid profanity or racism.

Acceptance Tests: Language Test

20 Legal Requirements

20a Compliance Requirements

57 - Legal Compliance

Description: If any aspect of the application is deemed to be illegal or violates any terms, it will be altered.

Rationale: This is to ensure that we don't get lawsuits against developers or the company.

Fit Criterion: Maintaining flexibility to change any possible legal issue.

Acceptance Tests: N/a

20b Standards Requirements

58 - Health Inspected Groceries

Description: The partnered groceries that are included in the application shall comply with Health Inspection Standards.

Rationale: We can only recommend ingredients from a grocery that have passed health regulations and inspections.

Fit Criterion: When partnering with these companies this will be checked before displaying data from their inventory.

Acceptance Tests: Health Check Test

21 Requirements Acceptance Tests

21a Requirements – Test Correspondence Summary

Test	Req 1	Req 2	Req 3	Req 4	Req 5	Req 6	Req 7	Req 8	Req 9	Req 10	Req 11	Req 12	Req 13	Req 14	Req 15	Req 16	Req 17	Req 18	Req 19	Req 20	Req 21	Req 22	Req 23	Req 24	Req 25	Req 26	Req 27	Req 28	Req 29	Req 30	Req 31	Requirements
Machine Learning	x																															
Cook		x																														
Recommendation		x																														
Account			x	x																												
Recipe			x																													
Video			x																													
Machine Learning Output				x																												
Database User					x																											
Database Ingredients						x																										
Reliability						x																										
Maintenance						x																										
Offline Machine Learning							x																									
Health Check							x																									x
Database Management								x	x																							
Tech Support								x																								
Website Option									x																							
Operating System										x																						
Database Scaling										x																						
Inventory Scaling										x																						
User Engagement										x																						
Chef										x																						x
User Access										x																						
Store Access										x																						
Password Encryption										x																						
Verification Data										x																						
User Privacy Policy Acceptance											x																					
Network Security											x																					
Data Backup											x																					
Usability											x																					
Clicking											x																					
Recipe Check											x																					
Question Check											x																					x
Language											x																					
Store Training											x																					
Color											x																					
Font											x																					
Font Color											x																					

Figure 1. Testing Correspondence

21b Acceptance Test Descriptions

Machine Learning:

Description: The test requires that user pattern behavior be analyzed through machine learning and for it to scale properly per individual instance.

Cook:

Description: This test checks if users are planning a meal through recommendations or through individual ingredient searching.

Recommendation:

Description: This test ensures that the users recommendations are being populated with items that are based on ingredients that have been previously scanned. This test works closely with the Machine Learning Test to improve accuracy for the user.

Account:

Description: This test makes sure that the user can create an account and access it.

Recipe:

Description: This test ensures that correct ingredients are provided for the recipe to make a whole meal. This test works closely with the Machine Learning Test.

Video:

Description: This test checks that a video is being displayed to the user on what they are viewing or being taught on how to cook.

Database User:

Description: This test checks that the database is storing all user information and is updating any credentials if the user so desires. This test works closely with the Account test.

Database Ingredients:

Description: This test checks that the database is storing all ingredient information and is accessing them whenever the Machine Learning Algorithm accesses. This test works closely with the Account and Machine Learning Tests.

Reliability:

Description: This test checks if the system is staying up or if it has gone down based on an estimation on how often the system will go down.

Maintenance:

Description: This test checks that if the system went down it can be rebooted back up, and while the system is up keeping all assets and entities up to date that it has access to not including database entries directly.

Offline Machine Learning:

Description: This tests if local data can be provided to users even if the system goes down so the user can continue to have a temporary set of recommendations.

Health Check:

Description: This checks that data being accessed for the user has passed all the dietary restrictions and Health checks that they provide on their initial profile.

Database Management:

Description: This test checks that the database has been optimized by removing old or duplicate pieces of data to keep speed as efficient as possible.

Tech Support:

Description: This test checks that there is a stable connection for the user to use at any point where they can contact Tech Support for assistance or submit a ticket for a service.

Website Option:

Description: This test makes sure that the browser version of the application is running smoothly. This test works closely with the Operating System Test.

Operating System:

Description: This test checks that the application is running smoothly on iOS and android OS.

Database Scaling:

Description: This test checks if the database can scale accordingly to handle larger loads of data influx or loss. This test works closely with the Database Management test.

Inventory Scaling:

Description: This test checks if the inventory scaling is accounted for and provides a detailed report of supply to the grocery store.

User Engagement:

Description: This test checks how many users use this app within a 2 week range on an interval.

Chef:

Description: This test checks that chef's have passed a basic qualification or training that will be provided as well as being able to interact with the user better for tutorials.

User Access:

Description: This test checks that the store's availability options and recipes are visible on the user's end.

Store Access:

Description: This test checks that the store has adequate access to information of the user without violating the users privacy. This test also ensures that the stores get more access to data from Smart Grocery on which items are being bought through us the most. This test works closely with the Inventory Scaling Test.

Password Encryption:

Description: This test verifies a user's password has been encrypted and is secure.

Data Verification:

Description: This test ensures that personal information such as inputted dietary restrictions are encrypted and aren't accessible to others. This test works closely with the Account Test.

User Privacy Policy Acceptance:

Description: This test returns true or false whether the user has accepted the provided Privacy Policy.

Network Security:

Description: This test continuously checks if any security breach has occurred in the data or any corruption of data thereof.

Usability:

Description: This test checks that the interface can navigate from each page to another without getting stuck.

Clicking:

Description: This test makes sure that a user can reach where they need to from the home point of their app in 6 clicks or less.

Recipe Check:

Description: This test checks that the user has completed all the prerequisites before attempting to cook the meal. This test works closely with the Chef Test.

Question Check:

Description: This test checks that questions have been answered in a forum and if they are not, they will be monitored until they are.

Language:

Description: This test checks that language is proper and formal throughout and comments made by other individuals doesn't involve racist or hateful speech.

Store Training:

Description: This will ensure that a set of instructions for training have been provided and completed on an external source of partners that know how to implement our application to their grocery.

Color:

Description: This test checks that the color scheme is Red and Yellow for the application throughout.

Font:

Description: This test checks the consistency of font based on Title, Header, and description.

Font Color:

Description: This test checks that the Font Color is correct and is consistent throughout.

III Design**22 Design Goals**

Our main design goal is to be able to recommend recipes to the user based on their location and ingredients as fast as the user can go through the suggested recipes page. This experience for the user should feel instantaneous, but in reality, the machine learning model runs in the background when the user is not using the application and prepares the recipes beforehand. The machine learning model will run on the servers and send the results to the user's mobile device.

23 Current System Design

N/A

24 Proposed System Design

24a Initial System Analysis and Class Identification

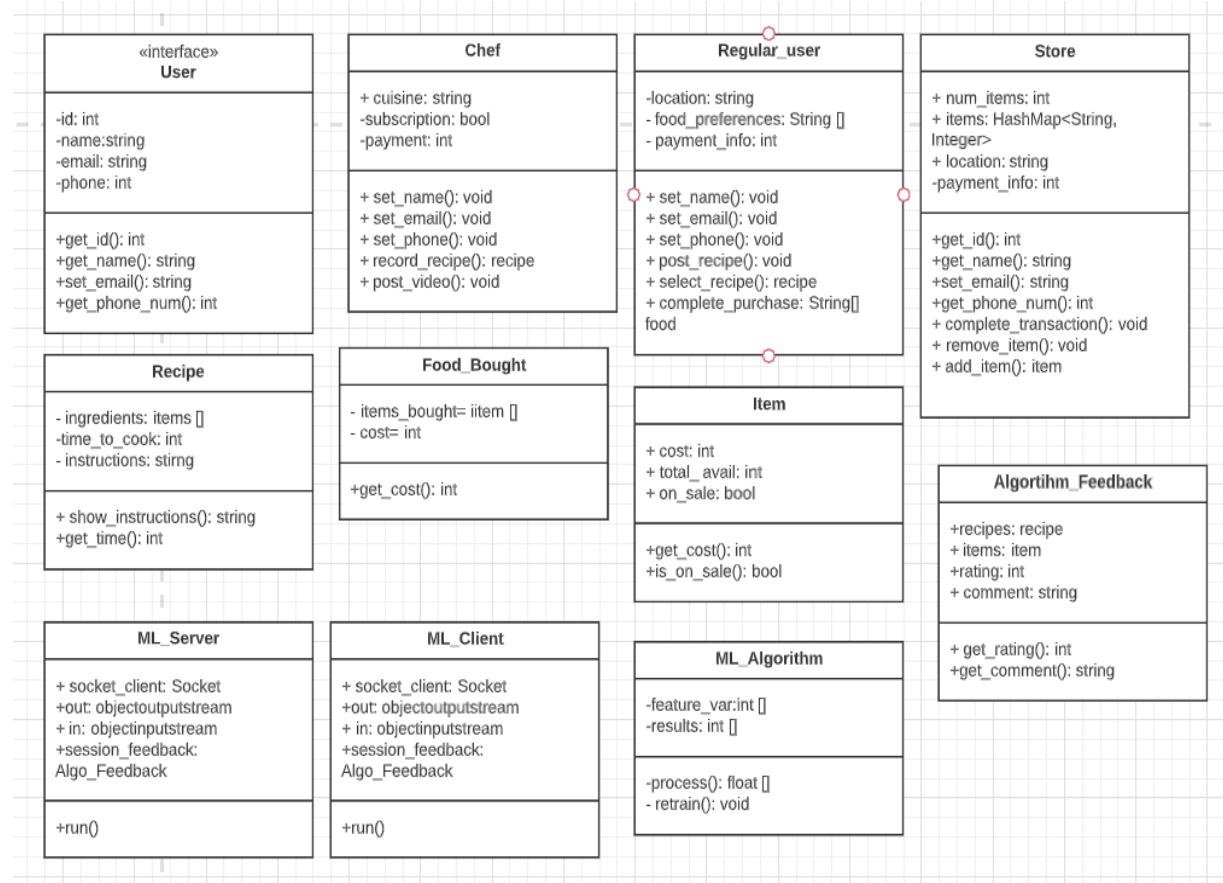
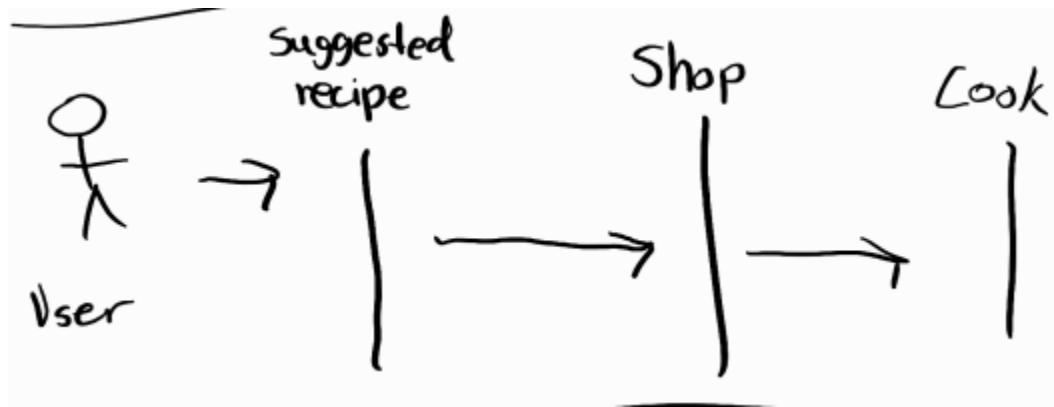


Figure 2 . Classes for the Smart Grocery application

24b Dynamic Modelling of Use-Cases



24c Proposed System Architecture

The proposed system architecture would be client-server for sending the food preferences of users to the machine learning algorithm and receiving food preferences for the user in return. The centralized server would contain the database as well as the machine learning algorithm would take in the real time recommendation and return the possible recommendations, food recipes of the user.

24d Initial Subsystem Decomposition

User

This interface contains shared attributes amongst the chefs, application users and stores. An interface is used for better object-oriented structure with the program. The chefs, application users and the stores will be using polymorphism with this interface since they contain the same methods, but different usage.

Chefs

This class contains essential chef information as well as allows them to update their database regarding the amount of application users who are subscribed to them, and can post their recipes and videos.

Store

This class contains essential store information such as items, payment information, and location. It also contains functions like `check_inventory()`, `add_to_cart()`, and `pay()`.

Application Grocery User

This class contains essential application grocery user information as well as functions for them to browse recipes, post recipes, browse and buy food from different stores, and give feedback regarding the machine learning algorithm and the predicted food

preferences. The user has a lot of interaction with the app, so this class will contain many functions to allow them to have the smoothest interaction possible.

Recipe

This class contains a list of instructions on how to make a recipe. There also will be methods for chefs to add videos of them cooking their recipes.

Food Bought

This class is a list of items that the grocery application user has bought from the store, and the int cost. No methods are needed as this class is designed for modularity.

Item

This class contains the item information in a store; it only has the cost, if it's on sale, and the number of items available. Inventory (number of items) must be updated frequently.

ML Algorithm

This class contains the actual machine learning algorithm needed to predict the food recommendations of a user based on their food preferences, and past recipes they have cooked. The feature_var is the preferences that the application grocery user will record as a binary. Therefore, it is stored as an array of items. The results array is a list of food items that the user is predicted to like.

ML Server/Client

These two classes function like any normal server client class that is implemented in Java. The client will give the server the food preferences and the server will analyze the data using the ML_Algorithm class. They contain the necessary attributes for the server client connection to be possible as well as a serializable and run function to send information back and forth between each other.

Grocery application User/Algo Feedback

This class contains information for the application user to give the machine learning algorithm feedback regarding usability or for the application user for their preferences. These classes have getters that the application grocery user will fill out in the app.

25 Additional Design Considerations

25a Hardware / Software Mapping

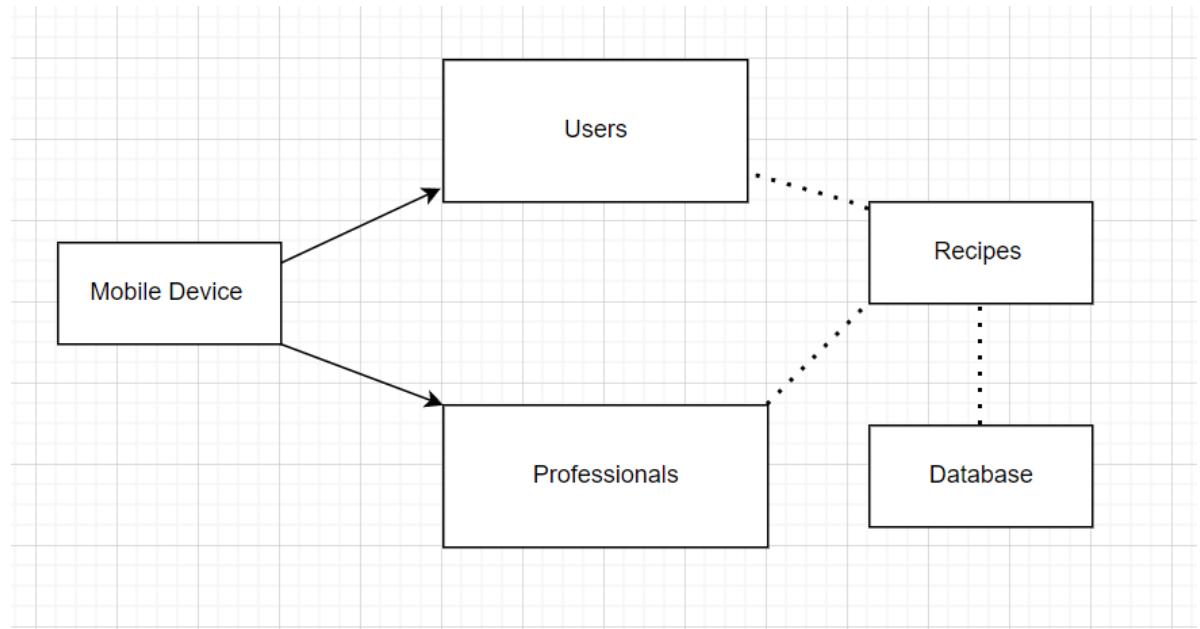


Figure 3 . Mapping

Arrow - Contains

Dotted line - uses

25b Persistent Data Management

Data management will be needed and this will be our top priority since many of our features are reliant on our data, so we will need to protect the data and during other situations as well. The data will be separated to multiple servers in case of server breaches or server shutdowns. If one server is shutdown or not working we expect the other servers to pick up the load. Each server will have 10 to 15 terabytes of data to store all the information regarding the user and recipes and more. The data will be stored in all the servers. The main server will be called server1 and will be the driving force and in case of lost connection or a shutdown the other servers will turn on and keep the system working.

25c Access Control and Security

The mobile devices that will be connected to the applications network will be done with valid ID credentials. Only devices with valid and certified credentials will be allowed to gain access to our application, those who use invalid credentials will not be able to use the application. Those who try to breach or make unauthorized attempts will alert the network administrator and inform them that something is happening. All the data being stored in our database will be private information of our users, therefore we want to ensure maximum security. We will also add encryption to the user's password and username and other information that is private to the user.

25d Global Software Control

Certain requirements will be needed to prevent certain issues from coming for like duplicates that might send an error to our database or an API.

25e Boundary Conditions

If there is a shutdown of our systems, the service should still be working regarding the features of looking up recipes and will also be able to get recommendations from the machine learning algorithm. However, professionals or users will not be able to share their recipes or other information. All the data will be updated every week and be sent to our backup servers.

25f User Interface

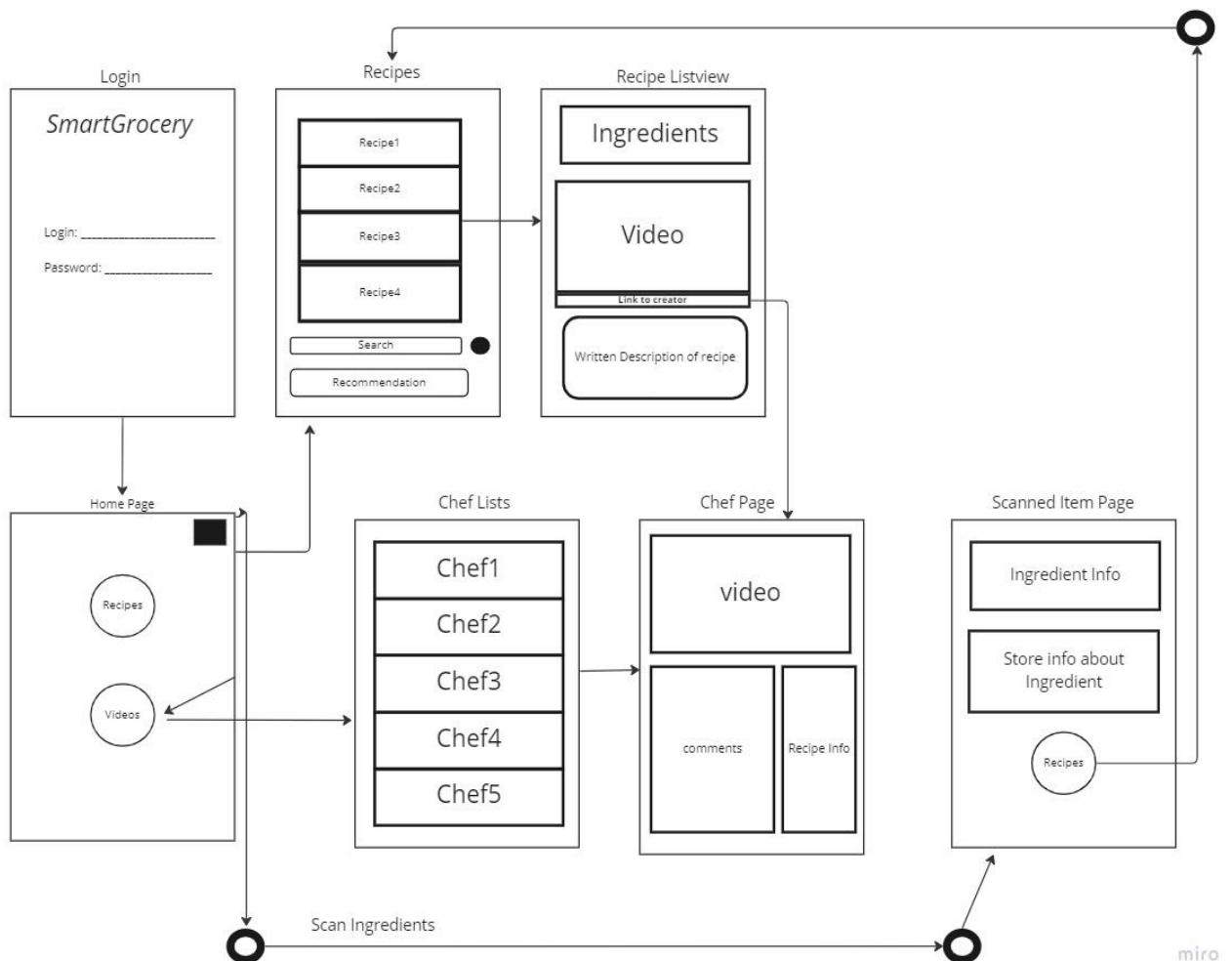


Figure 4. UI for Smart Grocery

25g Application of Design Patterns

All the design patterns have been documented already and the ones we will be using are within the communication diagram.

26 Final System Design

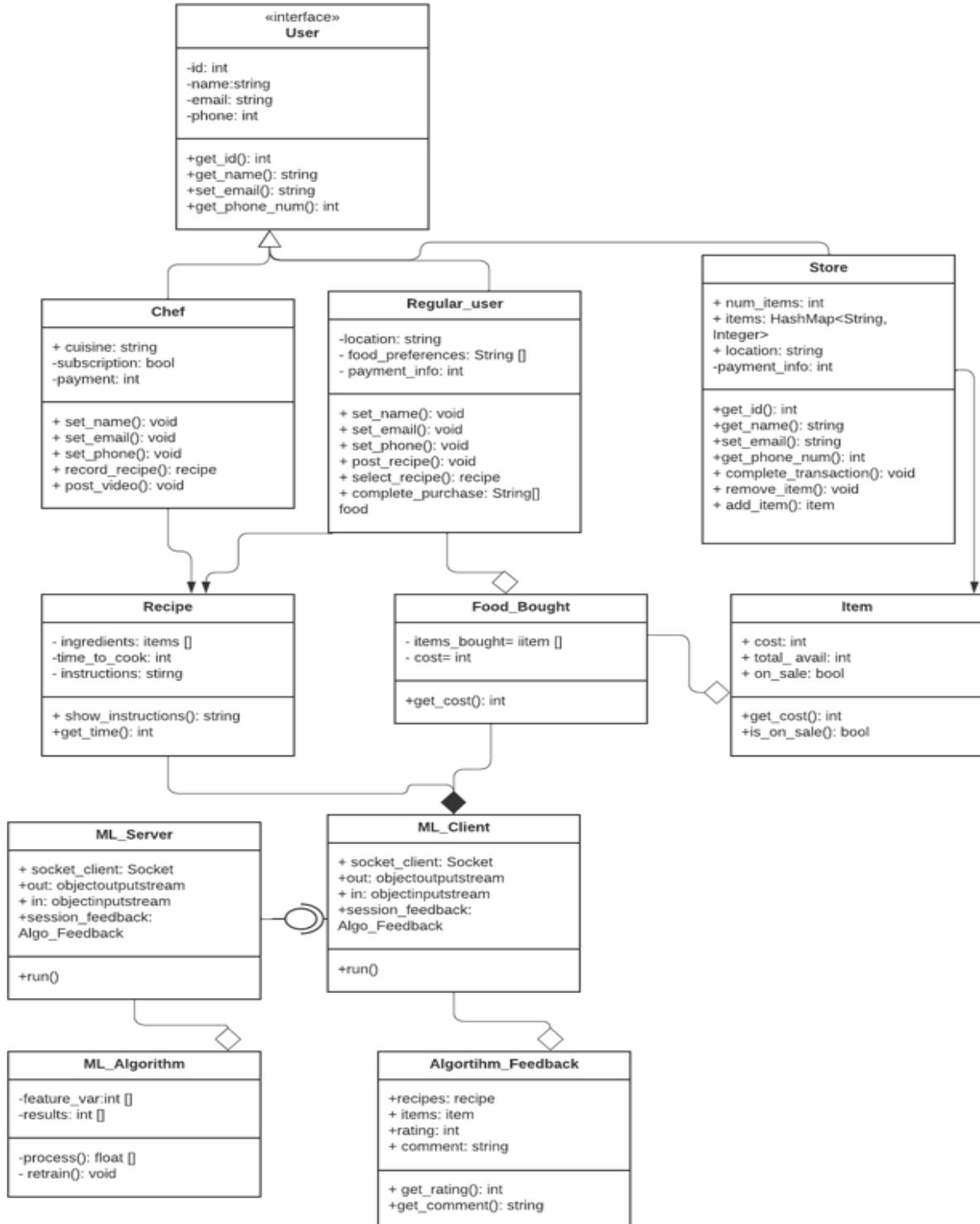


Figure 4 UML Class Diagram for Smart Grocery

27 Object Design

27a Packages

We will be using packages to implement the Database functionality into the application and also be using packages regarding the login and signup features. Another important package will be used for the styling of the application and also the video calling features.

27b Users

The user subsystem will get information regarding the user such as the ID number, name, email, and phone number. Other information such as which ethnicity the user is and also what sex the user is. All the information will be stored into a database. The user will be able to update their information. Since we have users and chiefs there will be 2 classes that will be inherited from the User subsystem class.

The first user class will be the regular user. This class will have the same things as the user class, regarding their private information. This class will also have contents regarding feedback.

The second class is the professional and they will have similar variables and also contain professional id.

27c Recipes

The Application will be recommending users recipe's based on a value system in the ML model. The usage of certain ingredients which gets stored as well as any manual searches that the user would like to input will be included to make a decision on recipes to be recommended.. The data for new recipe queries will be updated separately as an object that will be stored under the individual user in the database and be assigned a value as well. The recipes that are posted by chef's will be standalone objects that will be cycled in and out of new recommendations outside the standard user's scope if it is not being recommended already.

27d Machine Learning Connection - Server and Client

The information regarding the users recommendation will be given based on a client and server connection which will be done through an online or cloud based system. We expect the stored data from our cloud database about the user to train the model and then send a list of recommendations to the user. However, if a local database system is implemented to expect that they use a client and server base and send the data through that to the machine learning model.

IV Project Issues

28 Open Issues

An issue that can be raised is how local groceries are to handle any large demand increase if they do not have the supply ready. The application, which recommends ingredients, and certain products, will naturally find patterns of ingredients of dishes that people are liking and has a higher chance to recommend it. This means certain items will have a marginally higher demand and the stores supplying these ingredients can get overwhelmed and not be able to provide on short notice.

29 Off-the-Shelf Solutions

29a Early Alerts

The application's machine learning model will account for any outstanding increase or decrease in the recommendation for any items that exceeds a preset threshold. If this trend is found by the model, then any associating store that sells this product will be notified earlier of the influx and will provide recommendations to increase supply to meet demand. The ultimate decision will lie in the hands of the vendors if they would like to follow through.

29b Spreading Location Out

In the case that the Early Alerts does not prove to be as effective or the vendor cannot withstand the demand for any product, the vendor can inform us of the issue and we can provide the same ingredient recommendation for users if it is also available in other nearby locations for purchase.

30 New Problems

The application is built to handle individuals that build a dependency on the service that is provided so there are potentially concentrated amounts of high frequency users which means any issue that appears will be a repeated one until addressed and resolved.

30a Effects on the Current Environment

Due to the fact that this is an additional tool to be utilized for users to improve quality of life and increase profit for local businesses, this application should not affect either of these 2 factors. The application serves to be supplementary to the base exchange in products that already exist. Regular maintenance on the application and analysis of trends in usage of the application will help regulate any issues in the current environment.

30b Effects on the Installed Systems

The software that is being used is a standalone software application that is not built around any software or hardware that currently exists. Therefore the implementation of the software should not affect any current installed systems.

30c Potential User Problems

Since the application starts with a user-created base profile and provides ingredients based on location and availability as well as user's pattern usage and liked foods, any consistent user will be given different recommendations with a weaker accuracy for a short period of time if relocated to a different area. The application uses the machine learning model to track foods that the user likes but needs to base ingredients and cuisines based off of local stores' availability. This would need to be addressed based on short-term and long-term relocations as 2 separate cases.

30d Limitations in the Anticipated Implementation Environment That May Inhibit the New Product

Many stores currently have products that are common amongst numerous other stores in similar locations. Guiding users to certain stores based on availability and region eventually can lead to a bias against other competing local stores that are not receiving adequate recommendations and it can hurt their business. The spread of recommendations should be checked frequently that it is even to the degree of still providing the best option for the user while not hurting competing businesses.

30e Follow-Up Problems

If sensors were to be implemented as part of a future plan in each of the store aisles for more accurate inventory tracking, then a problem may arise with the hardware that would need to be maintained and redeployed on a regular basis as well as a fail-safe system to act as a standing backup in the event of damage to the sensor.

31 Migration to the New Product

31a Requirements for Migration to the New Product

N/A

31b Data That Has to Be Modified or Translated for the New System

N/A

32 Risks

Lack of Chefs

Risk Factor: High

Probability: Low

Rationale: The chances are low because this application is collaborating with the chefs, and they can post their own recipes and videos. These chefs can gain popularity by users being subscribed to them, and by gaining followers.

Lack of Application Grocery Users

Risk Factor: High

Probability: Low

Rationale: As mentioned earlier, this application is primarily focused on encouraging individuals to cook more by trying new recipes and seeing what others around them are cooking. Given there is no all in one application like ours, we expect there to be a lot of interest surrounding it.

Lack of Stores

Risk Factor: High

Probability: Low

Rationale: The chances are low because this application is collaborating with the stores, and they have their own products and customers. These stores can make a lot more profit by using our application, as more people will buy groceries.

Data Theft/ Data Loss

Risk Factor: High

Probability: Low

Rationale: The chances for a data theft or data loss to occur are quite low because the system has all possible security measures (Firewalls, Protection Software). The data will also be backed up the database in case of a data breach or loss.

Inaccurate Machine Learning Analysis

Risk Factor: High

Probability: Low

Rationale: The chances for the machine learning algorithm to be inaccurate are low because the machine learning team will train the model everyday with new data sets everyday. In case a new data

set fails, the old one will be reverted.

33 Costs

The cost of this project would require about 50 computers for employees to develop on, a database with enough bandwidth to serve its region of the world, and servers with enough power to run the machine learning models for the users. Along with physical infrastructure, we will also need to purchase licenses to have our application listed on the app store. One overlooked portion that we anticipate costing a lot of money is advertising and getting the app's name recognized.

34 Waiting Room

An idea that can be implemented later is a feature that tracks how many calories are in each portion of the meal being cooked by the user. Nutritional information on the app would also be good information to be displayed.

Another idea is giving the user the ability to find what type of food they prefer the most and giving them a breakdown of their ingredients by ranking them.

35 Ideas for Solutions

APIs and also other tools can be used to make this app much easier to make for example taking Yummly Api, which provides recipes. Also using the camera on mobile phones to scan the ingredients.

Developing this application can be done in many different ways, however we prefer using Dart and the flutter framework since this will be easier for the developers and will maintain code for both Android and IOS devices. This will prevent developers from taking extra steps and save money and time.

Regarding the database we also hope that the developers would use a cloud based database like Microsoft Azure SQL-Database or firebase realtime-database, however a local database can also be used.

36 Project Retrospective

As most projects in the industry there were some troubles throughout the process of this project. Some things we noticed were how some requirements weren't that easy to build as we planned. The amount of data that is being collected is too large for a single database to hold, therefore multiple databases would be needed. Another setback we noticed was how if the user does not want to share their information the app becomes unusable for them. This type of problem gave us insight on how we and other developers should handle these situations.

Also we noticed some issues regarding machine learning and how we could implement the idea of a machine learning algorithm into a mobile device and that it would not hinder the application in any way.

V Glossary

Android Studio: An integrated development environment for Google's Android operating system

API: Application Programming Interface, allows two separate programs to interact with each other

AWS: Amazon Web Services, public cloud offering

Client Server Model – Application structure where tasks are distributed between the providers of resources, known as the server, and the requesters, known as the client

Dart: A programming language designed for client development, such as for the web and mobile apps

Flutter: An open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, macOS, Windows, Google Fuchsia, and the web from a single codebase

Machine Learning – a computer algorithm that can automatically improve by use of data and by self-learning. This is a subset of Artificial Intelligence.

MySQL: open-source relational database management system

Public Cloud: Computing Infrastructures managed by a third-party provider

Private Cloud: A cloud computing consisting of hardware and software dedicated to one single organization

Real Time Data – Data that a program receives while the user is interacting with it. This is usually processed by an algorithm to enhance user experience.

VS Code: Source-Code editor made by Microsoft for Windows, Linux and macOS

VI References

1. Cai, Shijun, et al. "A Machine Learning Learning-Based Predictive Model"