

MyKitchen UTD CS Expo Presentation Script

Opening/ Introduction

0:00 Garrett - Hello everyone, and welcome again to the UTDesign Expo. My name is Garrett Griffiths and I am joined by my fellow co-founders Thinh Le, Krish Suchdev, Elliott Gorman, and Rowenna Switzer.

The project we have to present today is the MyKitchen app, a mobile app intended to address the management of food inventory in a household setting. We have found, as soon-to-be graduates, that we are, and will continue to be, too busy to properly prepare a meal. Between not having the right ingredients for a recipe, discovering that something has recently expired, or simply not having a good idea of what to eat, we default to the classic meals such as instant ramen or restaurant takeout. Given the move in recent years to produce digitized convenience in transportation with Android Auto or Apple CarPlay, or in meeting contexts with Zoom and MS Teams, we have recognized an opportunity to address this issue with the hope that users will be able to save money on takeout and enjoy the health and mental benefits of cooking a good meal for yourself.

To accomplish this, we decided to develop the “MyKitchen” mobile app using apple’s SwiftUI to create a platform with which to store and manage one’s personal pantry. Between the ability to efficiently manage the food items in one’s pantry, create recipes and track it’s ingredients, and generate multiple shopping lists to expedite outings to the store, we have created a pleasant user experience on a smooth UI to properly address this issue. Now Thinh will help to explain the means by which the specifics of food storage and management are handled.

1:30 Thinh - Thank you Garrett.

The Food or Pantry view of the app is essentially a list of ingredients that are available in the pantry, for example: eggs, meat, or canned food. Users can manually enter the information of an ingredient into the app or they can use the barcode scanning feature, a convenient alternative to manual entry. For food items that do not always have a barcode, like produce, users can enter the information of those items manually. They can enter information such as name and quantity, expiration date, where the item is in your kitchen, or how it should be categorized. For food items that do have a barcode, users can use the automatic data entry feature, which is the barcode scanning feature, to enter those food items’ information into the app. Krish can explain some details on how the user can use these features to better go about their day using the Recipes tab.

(Discuss the broad setup of the Food views - Explain the different options for storing data [defaults vs no defaults, ability to track location and expiration, etc]; touch on the inclusion of barcode scanning as a solution to the problem of efficiency on the user experience)

2:45 Krish -

Thanks Thinh. So our app also offers a recipes tab where we'll display user-added and curated recipes. You won't need to be a professional chef; each recipe features a list of ingredients and instructions on how to make it. That may sound familiar to recipes you could find online, but they don't know which ingredients you already have and which ones you would need to get, which makes it difficult to figure out which recipes you can make right now. So instead of going through that tedious process of scrambling through your pantry to see if you have each ingredient on the recipe, we'll do it for you in just one tap. Each recipe has a check ingredients button that automatically checks your pantry to divide the list of ingredients into two sections: ones you already have in your pantry and ones you need to buy. This takes quantity and units into account as well, so if a user has an ingredient but not enough of it for the recipe, we'll tell them how much they still need to buy. Basically, the goal with the recipes tab is to allow users to find out, within seconds, which recipes they can make right now and if they can't, what ingredients they should buy.

Throughout the app, our user interface has been designed to follow Apple's Human Interface Guidelines. What this means is that the look and feel of our app is consistent with the iOS platform so that the app behaves in ways that people would expect. As you can see in the screenshot, we have a tab bar that separates each of the primary features. The first tab is the home tab, which has all the features that you heard Thinh talk about earlier. The second is the recipes tab, which is where everything that I just talked about is.

And the last is the shopping lists tab, which I'll hand over to Elliott to talk about.

4:45 Elliott - Thank you, Krish. One of the major issues we identified with apps similar to MyKitchen was the ease of turning recipes into a shopping list. Many apps utilize features to do just this, but in our findings they neglect to consider items you may or may not have, often causing users to buy more items which they already had or causing users to manually edit these lists after the fact. One of the main purposes of MyKitchen is to provide ease of pantry management. As part of this, it's nice to have recipe-based shopping lists that *incorporate* food that you already have. As Krish discussed, this is why we developed a recipe view that can automatically tell you what you do and don't have to make a given recipe. Using this, we developed a shopping list feature that automatically pulls ingredients *you don't have* for a recipe directly into your shopping lists in the MyKitchen app. This feature will allow users to easily manage buying food for recipes, without worrying about buying too much. We also intend to integrate the shopping lists with Grocery APIs so that your lists can automatically be populated within a Grocer's online order page, which Garrett will discuss later.

Another quality of life feature we've added is the ability to search items in the pantry. The purpose being a simple method by which users can quickly learn if they have any particular

item. This makes the questions like, “Do I have whipping cream at home?” much easier to answer, and allows users to more easily manage their pantry.

Our hope with all of the features we've discussed is to provide users with easier pantry management, and to save time when creating shopping lists. To learn more about what empowers these features, Rowenna is going to discuss the internal database.

6:45 Rowenna -

We chose to implement our data store in Swift Combine, which is a framework in the iOS SDK for creating reactive data models that integrate natively with SwiftUI. Our choice to use Combine allowed us to define our database directly in terms of Swift objects, which maximized our ability to iterate in the short timeframe afforded for this project. Changes to the model used by our views in Swift code were immediately reflected in our data store, by the nature of our framework choice. In moments of app inactivity, we save the entire data store to the file system encoded in JSON. Whenever the app launches anew, the data store is retrieved from the file system. This simple setup is ideal for small amounts of data, which is okay because we don't expect our users' pantries to contain hundreds of thousands of items.

Furthermore, the ease of converting the data store, as well as objects within it, to and from JSON keeps the door open for future development involving servers and other platforms.

Our data store is, in the Combine lingo, “Published”, which means that observers, like our SwiftUI views, can register themselves to be notified of any changes in the data. For anyone who remembers RSS, it's like the database is vending an RSS feed of changes to the data, and our SwiftUI views are subscribed, so they can update themselves in a timely manner in response to changes in the data. Of course, it is through the views that our users modify their data as well, so in fact we are using two-way bindings, where the data store is also signed up to be notified when a view would like to update the data.

Regarding privacy of said data, none of your data ever leaves your phone. There are no binaries that we've dropped in from other companies that might see your data. Even the barcode scanner is implemented natively with the iOS SDK and simple lookups. We take our users' privacy seriously, and even in the future, as more platforms and data sharing features are added, user data would only be transmitted optionally, at the request of the user.

Finally, Garrett will discuss what the app's features mean for the users and the future of this project.

8:45 Garrett -

Thanks, Rowenna.

Overall, our goal is to provide the most seamless and efficient user experience possible to ensure that time spent towards using MyKitchen for food inventory management cuts down on the time that would be spent doing it manually, as much as possible. While the feature set we have developed thus far is indeed conducive to helping the user with food inventory management, meal preparation, and other aspects previously outlined, we also recognize that the critical necessity for an app such as this is the ease of user experience and the efficiency at which it can be properly employed. That given, we have plans to support this project going forward with prospective features such as Receipt Scanning, which would allow users to take receipts from their store visit and scan them to bulk-import items to save a great portion of time in inventory management. We also have plans to implement Pantry Sharing, to enable multiple people across different devices to access the same pantry, as we understand that having a single device to manage your food inventory across multiple people is not the ideal scenario. Additionally, integrating Grocery APIs into the app could allow users to export shopping lists directly to store webpages for easy purchasing, or import their online receipt into the food storage for easy bulk imports. These features, as well as many others throughout our continued support of this project, will combine to create a truly streamlined experience for kitchen management.

Thank you all for allowing us to take some of your time, and we hope that the remainder of the expo is a pleasant experience.