CMSC436 (Team 18)

Final Project Report

PotLuck

Sierra Seabrease

Kurnal Saini

Rahul Khanna

Isha Angadi

[App Walkthrough Video - https://www.youtube.com/watch?v=G_BuIAYr2wM&ab_channel=Kurnalsaini](https://www.youtube.com/watch?v=G_BuIAYr2wM&ab_channel=Kurnalsaini)

Overview

Remember being a freshman in college — bored of dining hall food, no access to your car, barely any money left in your account, and nothing fun or nutritious to eat in your dorm mini-fridge? We remember the feeling! That's why we created **PotLuck**! PotLuck connects hungry college students with over-zealous college students who love cooking food! Once a student whips up a dish, they can post a portion of it on PotLuck and another hungry student can find their dish, select a pick-up location, grab food from their peer or choose to sit down and eat with them! PotLuck is our modern take on the barter system that brings together cooking and socializing with a few simple clicks. Terps on campus can use our app to enter information about the dish they have cooked and fellow terps can request to pick it up from them through our map feature — combining good company and great food!

Running the code

To run our application, follow these directions: open the 'nextbigthing' folder, then open the PotLuck folder, then open the PotLuck subfolder. In this folder, open PotLuck.xcworkspace and hit the run button on the top left. As a prerequisite, please make sure to have cocoapods installed since the team uses Firebase as a dependency. In order to configure this run **sudo gem install cocoapods** and then **pod install**.

Goals

The primary goal of this application is to connect terps on campus who have cooked extra food to share it with fellow terps on campus who cannot cook or do not have access to food. PotLuck will curb food waste, while allowing terps to connect with each other! Our app will showcase a map and list view of available dishes around the University of Maryland (UMD) campus, with an interface that allows the user to add, delete, or pick up dishes.

Logging In

When you first open our app, you are greeted with our login page where you can sign in if you already have an account (Figure 1). If you don't have an account there is a sign up button at the bottom of the screen which will redirect you to our sign up page. The sign in page uses Firebase Authentication to create a user and also uploads a user document to FireStore. The login screen retrieves the user information from FireStore by doing a **where** query with the inputted email. The user data is placed in an **EnvironmentObject** so that we can easily access the user uid later on when uploading and accessing dishes to the database. The reason this is needed is that the uid is needed to add a dishes subcollection for that user.

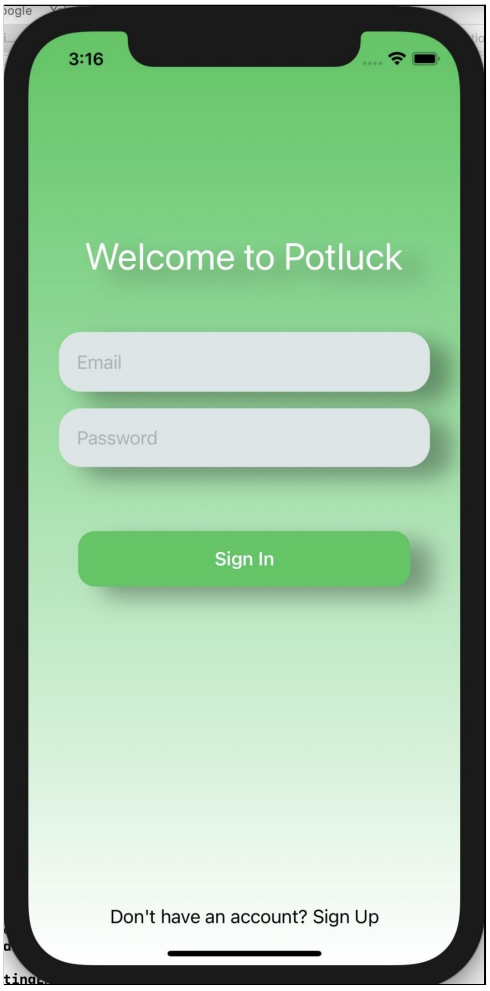


Figure 1. Login

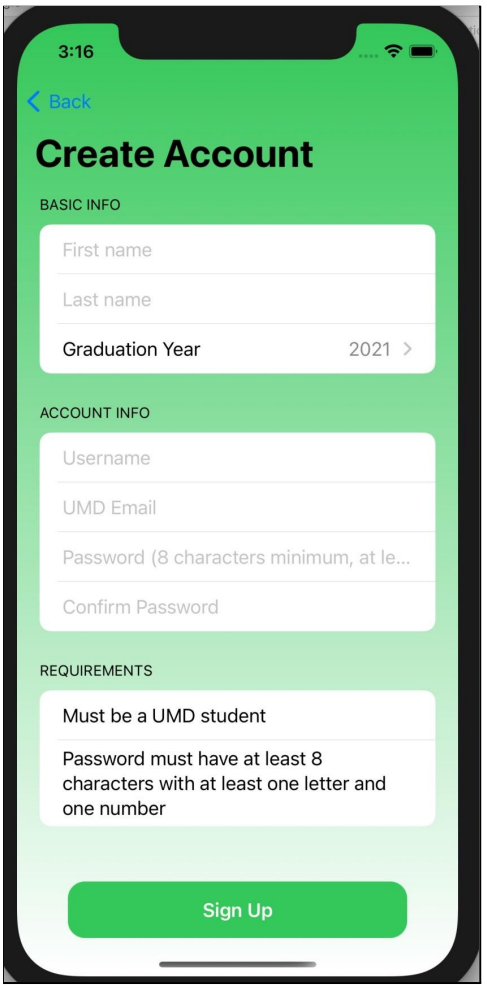


Figure 2. Sign Up

Creating an account

When creating an account (Figure 2), we authenticate the user by requiring them to have an email ending in .umd.edu or .terpmail.umd.edu since our target audience is on-campus UMD students.

Navigating the application

Our app has a navigation bar on the bottom of the screen with separate tabs for adding a new dish and picking up a pre-existing dish on the map (Figure 3). It also consists of a header with the user's account information, our logo, and a creators credits page (Figure 4). Our simple design is easy to navigate and fun to use as well!

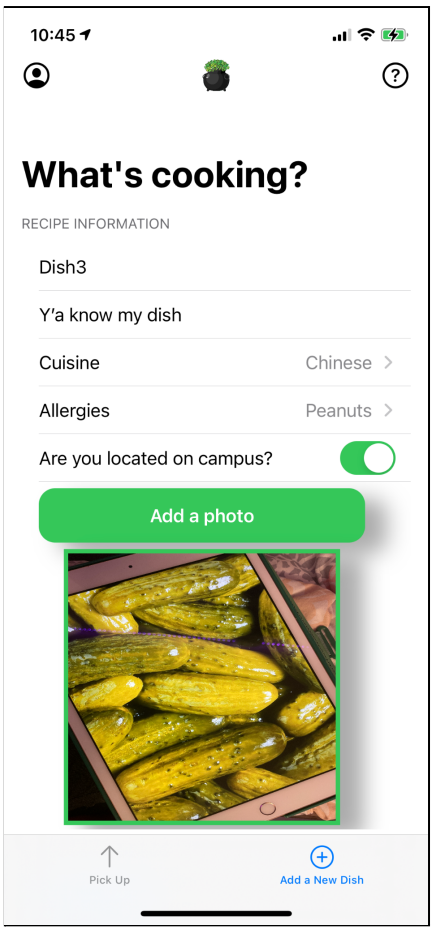
Figure 3. Navigation | Figure 4. Header

Uploading Dishes

When uploading a dish (Figure 5a), the user is required to enter its name, ingredients, potential allergies, and share their locations to collect the pick-up.

Uploading Pictures

In addition to all of the information about their dish mentioned above, users have an option to click and upload a picture of their dish (Figure 5b). This was a stretch goal that we accomplished.



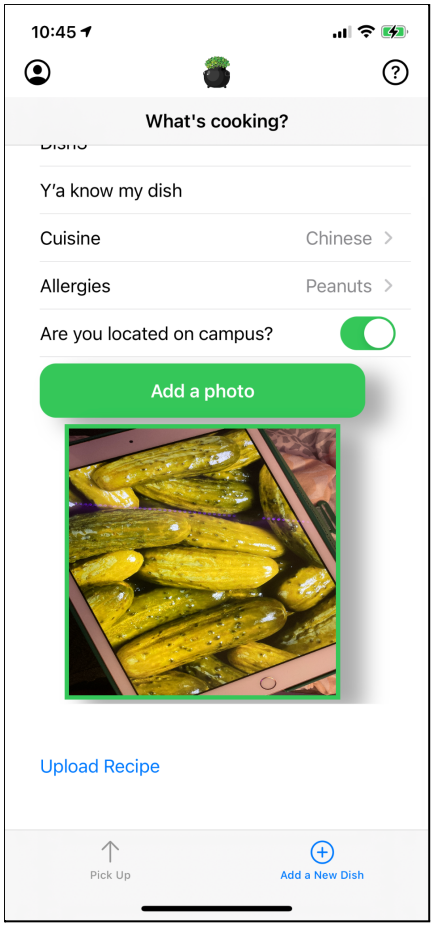


Figure 5a. Add A Dish | Figure 5b.Add a Photo

The team utilized Firebase Storage to send image data to their bucket on the cloud. In the callback for uploading an image, we then updated the document for the dish with the URL for the image so we could load it on the client side later.

User Dish Rating

Users can try many dishes but having information about the quality and taste of those dishes through a rating system is beneficial for new and existing PotLuck users since this allows us to maintain a decent quality of the dishes being shared.

Figure 6. Confetti on-click

After a user rates a dish and submits it to the database, a celebratory confetti animation is executed as feedback for performing that action.

Displaying Available Dishes

Our app design was originally based around just listing the available dishes on campus without having any visual aid to that. We were able to complete our stretch goal and actually display a map above our list of dishes to visually see how the dishes surrounding the user's current location.

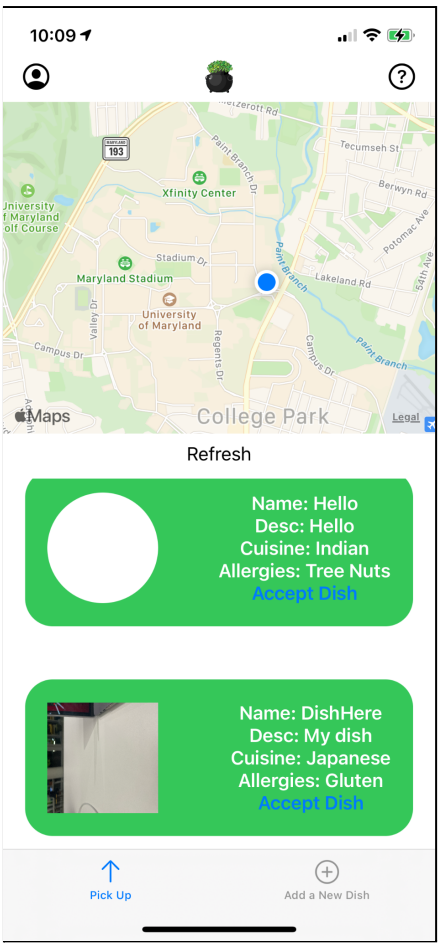


Figure 7. User Profiles and map location

In addition, we used a Dispatch Group to asynchronously load all the user's dishes in the database into an array so that we could display it in a ScrollView. If a user had uploaded an image, we would use **URLSession.shared.dataTask** to download and display it. Otherwise, we would render an empty white circle to symbolize that the dish had no image corresponding to it. The user could also accept dishes from this view as well.

User Profile

The user profile utilized the **EnvironmentObject** that was instantiated after logging in to display their metadata on the page. In addition, by using the UID for the respective user, a query is executed that allows the page to display all their active dishes.

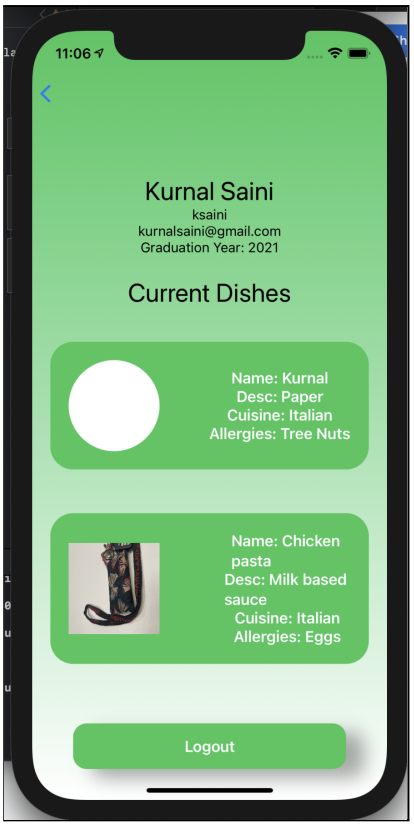


Figure 8. User Profiles

Storing the Information

Although we initially were going to use CloudData to store our app's information, we ran into some issues and made the switch to Firebase during our development process. When a user signs up, all of the user's information is stored on our Firestore Database (Figure 9). When a user adds a new dish, that dish's information is added to that specific user's list of dishes. The image that is captured with the dish information is uploaded and stored on Firebase Storage (Figure 10).

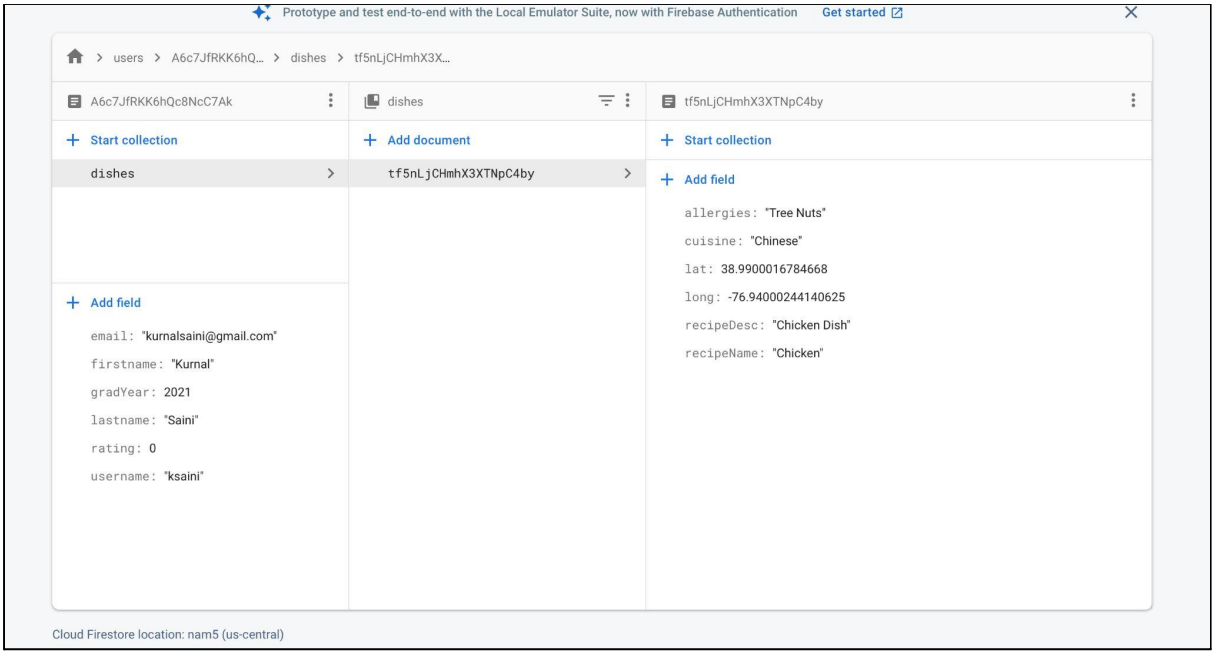


Figure 9. Firebase Firestore Database

Figure 10. Firebase Storage Database

Development process

PotLuck was developed over the course of the online semester. During this unique semester, we learnt the importance of communicating effectively to smoothen the process of working in a development team in an online setting. A couple of weeks before each Milestone, the team would meet to go over the goals for each milestone. Through the week, we would connect regularly to go over our progress either over Zoom or our group chat.

During Milestone One, we were able to complete all of our goals successfully. We made the navigation bars to navigate through the app — this was created such that each team member would have their own tab to work on. We created branches for each team member to avoid future merge conflicts. Additionally, we set up a user authentication system that only allows UMD emails to sign up since the app's target audience is on-campus UMD students. One of the tabs provided the users with the ability to enter metadata about the dish they have cooked. This tab consisted of a selector for cuisines, allergies, and an option to upload the dishes to the database. Finally, the last goal of this milestone was to create a scrollable list of the food dishes currently in the database.

Before beginning Milestone Two, the team went over their progress for Milestone One and then assessed the goals for Milestone Two. On going over the list of goals, we realized there were many small tasks to be completed before the big tasks for the milestone. We created a User class in order to add a new User to the database, got their current latitude and longitude using location services, made tables for Users, Dishes, and added a getter based on ID. We created a GUI for accepting and rating a Dish which consisted of —a toggle to set and/or display whether a dish is currently available, a separate tab to display detailed information about the dish's ingredients. On this page, we made the toggle control to share information regarding the availability of the dish. Lastly, we added the map feature in this milestone to improve the user experience. We created a map with markers that represents where dishes are available.

For Milestone Three, our final milestone, we focused on brainstorming and drawing a clear path to follow for the final submission of our PotLuck app. We brainstormed efficient

implementations for user-authentication methods for UMD students. We also compared different options for our rating system. We began by deciding that rating options will be available for the user's dishes rather than the user itself. At the end of this milestone, we chose a thumbs up and down rating system but on reassessing our options, we chose the 5-star rating system instead since we agreed this gave us better information about a user's dish. We also created a logo for our application — this would later be used in our header.

We ended our final milestone with a strong note by brainstorming ideas for a cohesive application flow, prioritizing a design that is user centered. Our choices of setting up a header with account information on the top right, a navigation bar at the bottom of the screen allows for natural user flow. Small decisions like matching the color palette of the application to the logo increased the application's visual appeal. In addition to making our application functional, we also made it fun! By adding surprise confetti and application credits animations, we kept PotLuck practical and interesting.

Throughout the semester, if at any point, there were instances that required deliberation, we would meet through zoom to go over those issues and brainstorm potential solutions. After brainstorming a few solutions, we would choose the best path to implement. For example, at the end of Milestone Three, we learnt that we would have to change our database due to a problem with adding our User type to the core data database, to combat this problem — we decided to switch from CoreData to Firebase. This solution helped us organize the data we collected easily and run the app smoothly.

During our last day of development, our app was running smoothly and each group member was able to actively contribute to the same working project file. This changed, for a reason

still unknown to us, our app began crashing whenever we would try to run it. We began trying to go back to previous commits but the crashing was occurring on every commit now. This was a major setback as we needed to take the time to debug and understand why our project is no longer working. One of our group members was able to resolve the issue on their end by fully deleting their workspace and reinstalling the pods. This solution, however, did not work for everyone and even after recloning our whole project, we were unable to run our app. This led to the final development changes having to be emailed to our member who had the working project and they were able to integrate them into our final application.

One of the main features we used that was not covered in class was the use of the device's camera and photos library. We employed this feature by providing the user with an option to snap and upload a picture of the dish they have cooked. Implementing this feature took a lot of research in which we learned that there isn't a complete way to do this with only SwiftUI. We ended up using UIKit just to be able to take the image and then we connected this into our SwiftUI views.

Throughout the development process, we learnt that frequent check-ins, effective communication, and productive brainstorming sessions made it possible for us to meet our milestone and stretch goals.

Potential future directions

While we were able to implement all of the goals we originally set out, we realized that there were so many features we would love to add to PotLuck. Currently, we have a 5-star dish rating system, a potential next step could be creating a review option — with this feature, users can leave reviews about the different dishes they have tried and their overall experience. Another potential next step is adding a share feature that connects with users' and UMD's

dining halls' social media accounts. Through this feature, users will have a community that exists on PotLuck and outside of PotLuck in the real world as well. Some other additional features we would like to add include deleting dishes, adding a quantity for number of pickups per dish, user profile pictures, and better user feedback on erroneous operations.

Resource Citations

1. Choudhry, Asad Ali. "Utility Class to Upload Images, Videos & Files to Firebase Storage-SwiftIOS." Handy Opinion, 29 Oct. 2019, handyopinion.com/utility-class-to-upload-images-videos-files-to-firebase-storage-in-swift-ios/.
2. "How to Build a Login Page in SwiftUI #1 - Mastering Text Fields and Understanding @State." *BLCKBIRDS*, 21 Feb. 2020, blckbirds.com/post/login-page-in-swiftui-1/.
3. "How to Use Camera and Library to Take Photos in SwiftUI." IOS App Templates, 5 Feb. 2020, www.iosapptemplates.com/blog/swiftui/photo-camera-swiftui.
4. Hudson, Paul. "Adding a Custom Star Rating Component." Hacking with Swift, 16 Nov. 2019, www.hackingwithswift.com/books/ios-swiftui/adding-a-custom-star-rating-component.
5. Jeevansays: et al. "How to Navigate between Views in SwiftUI by Using an @ObservableObject."BLCKBIRDS,14 Jan. 2021, blckbirds.com/post/how-to-navigate-between-views-in-swiftui-by-using-an-observableobject/.