CS 4720 - F18 - Final Project Report

Device Name: Dunsparce (Nexus 7) Platform: Android

Name: Jeremy Weng Computing ID: jhw2np

Name: Alicia Zheng Computing ID: arz7cn

App Name: Sous-Chef

Project Description:

Sous-Chef is the cooking partner you never knew you needed. Want a new recipe for chicken? Use our expansive search feature to get the best rated recipes online. Forgot the directions to an old favorite? Save and find it in your cookbook. Need some new inspiration? Give the device a shake to suggest a random dish! When your friends rave about the new mac and cheese recipe you forced them to try, you can share it with them in seconds by using Sous-Chef.

What we propose to do is create an app that will do the following:

- The system shall allow a user to search for recipes online
- The system shall allow a user to save searched recipes for future use
- The system shall allow a user to save their own recipe
- The system shall allow a user to see these saved recipes on the main activity
- The system shall allow a user to take or use an existing picture of a recipe and save it

We plan to incorporate the following features:

- Camera - The user can take a picture of the dish after making it.
- Device Shake - The user can shake their phone to suggest a random recipe
- Consume a pre-built web service - We will use an API to use a comprehensive list of existing recipes
- Data storage using SQLite - We will store used recipes and their pictures in a local database
- Open shared activity / features - The app will provide the user to share recipes through text

Wireframe Description:

Our wireframe shows a very basic layout that we envisioned when we first started. Surprisingly, we stuck fairly closely to the wireframe we submitted. After the launch screen there are two tabs. These tabs are now on the top. These tabs are split into a cookbook and a “find new”

section. Each of these had a list of cards, that could store thumbnails and other simple information if we chose to do so. Each card would have its own detail page, that would have additional functions like taking pictures and sharing, among others. In our wireframe, we actually failed to add in custom recipes and editing and deleting recipes, and we thought it would make more sense to not be so restrictive with the recipes. Overall, we stayed true to our vision despite having to adjust some goals once we realized how feasible they were.

Platform Justification - What are the benefits to the platform you chose?

Android provides many benefits, such as comparatively good guidance in laying out an app and a larger market. Most importantly for us, though, was that we simply preferred coding with Android. This is largely because neither of us have Macs.

Major Features/Screens - Include short descriptions of each (at least 3 of these)

For our major features and screens, we had a MainActivity that was divided into two fragments: a cookbook and a search tab. The cookbook tab displays a list of all currently saved recipes, while the search tab allows searching through an API. There is a floating action button on the bottom right which allows you to create a custom recipe (AddRecipe) for local storage and add it to the cookbook.

Clicking on any of the cards in the list will redirect to that recipe's details, and there are 6 buttons that show up depending on the situation. If the recipe is an API recipe, there is a special "Directions" button that redirects the user to the website containing those directions. The other 5 buttons are present depending on whether or not the recipe was accessed from the cookbook. There is also an edit button, which opens an EditRecipe screen, and a delete button, which deletes that specific recipe from the cookbook.

Optional Features - Include specific directions on how to test/demo each feature and declare the exact set that adds up to ~60 pts

Pre-built web service: One of the two fragments on our main activity screen is a search function, which searches through the Yummly API. All searches in the search bar are exclusively done through that API.

Camera: After searching for a recipe from the API or creating a custom recipe, clicking on any of the cards and then clicking on the right side camera button will allow you to take a picture of the recipe and store it as the recipe's picture.

Device Shake: When on the main activity screens, shaking the phone will bring up a random recipe from your own cookbook.

Data Storage using SQLite: All recipes in the cookbook are almost entirely stored in an SQLite database, including images.

Open shared activities/features: On the recipe details page, clicking the share icon on the right side will allow the user to email text containing the recipe's name, instructions, and such and share the recipe.

Testing Methodologies - What did you do to test the app?

We relied heavily on the emulator and used a Nexus 7 (Dunsparce) from Sherriff, sparingly. It was difficult to debug through a USB, since it made the app run much slower than would be efficient for actual use. We also reviewed each other's code often to catch errors before they become too big.

Usage - Include any special info we need to run the app (username/passwords, etc.)

Nothing in particular; you just need to give access for storage and camera (which the app will ask for when reaching the recipe details screen)

Lessons Learned - What did you learn about mobile development through this process?

1. Don't procrastinate.
2. Really, actually don't procrastinate.
3. If you start with unorganized code...it will keep spiraling until it's pretty unsalvageable without major refactorization. Our code at the moment is pretty scattered and there is bound to be a lot of redundancy.
4. Commit often! You never know when you'll want to go back to a previous version. Always err on the side of having more previous states to revisit.