CS 4720 - Final Project Documentation

Device Name: Murkrow Platform: Android

Name: Samantha Chu Computing ID: syc5pg

Name: Claire Frank Computing ID: cmf3ap

App Name: HooWantsBrunch

Platform Justification

We chose the Android platform for our final app. One major benefit was that both partners could use Android Studio, whereas Sam could not use XCode because she is too proletariat for a MacBook. Furthermore, neither of us enjoyed connecting everything on XCode - the ViewControllers and navigation confused and aggravated the both of us during the iOS mini-app.

Major Features/Screens:

1. The View Restaurant Reviews screen will allow the user to view all of the reviews that have been posted so far, sorted by restaurant. The user can also select a restaurant from the drop down menu and view the reviews for that restaurant only.
2. The Write a Review screen allows the user to post a review to any restaurant. They select the restaurant they would like to review using a dropdown menu, and type in the review content. They have the option to upload or take a picture to include with the review.
3. The Send Email screen allows the user to send an email to the restaurant of their choice.
4. The Find Restaurants screen allows the user to find the brunch location that is geographically to where he/she is at the moment.

Optional Features

GPS / Location-awareness (15 points)

The GPS is used to determine the user's location in correlation to the brunch locations in Charlottesville. It locates the user and then compares the location to all of the brunch locations, and the closest one is returned. To test, check that all of the markers appear for the restaurants by scrolling through the map. All of the restaurants are in Charlottesville, so it should not be too hard to find them all. Also, check with an external google maps to see if the returned closest restaurant specified at the top of the activity is actually the closest restaurant to your location. If the feature opens on the Rotunda and does not return the nearest restaurant, this means that the device has not loaded its location yet. Return to the main screen and press the "Find nearest restaurant" button again. It should load up your location and the nearest restaurant.

Data storage with SQLite (20 points)
The restaurant and review content are stored in an SQLite database. When the review is submitted, it is saved to the database, and the information is loaded from the database and parsed in the View Restaurant Reviews screen. The user can test the data storage by submitting reviews, and then checking to see if their reviews have been saved to the database. In the "Restaurant Reviews" page, there is a dropdown menu of all of the restaurants. The user can choose the restaurant, and they will see all reviews of that restaurant. The photos associated with that restaurant will be pulled from the database.

Camera (15 points)
The Submit Review screen allows the user to submit a review, and include a photo with their review. In the "Submit a Review" feature, the user can write a review of a restaurant. They are also given the option to include a photo, either a photo from the library or they can take a picture. Once the photo is chosen/taken, it shows up on an ImageView on the Submit Review screen. We recommend that the camera is tested by submitting a review with a photo you just took, as well as submitting a review with a photo from the library. Once the review is submitted, the path of the photo is saved to a file, and the path is stored into the database.

Open shared activity/features (10 points)
Our app opens the device's email function in our "Contact Restaurant" feature. The user can choose a restaurant in the spinner and write them a short email. When the user presses the send button, it prompts them to choose the email app of their choice to actually send the email. To test, we highly recommend changing the "To" email address to your own email address when you go to actually send the email. This is so you can actually test that the email was sent, and so we don't annoy all of the restaurants.

Testing Methodologies
We tested our app in several ways. To test the sign-in feature, we used several Google accounts to try to login, including creating our own (see Usage section). To test the Find Restaurants activity, we tested on both the device and emulator, which allowed us to change our locations. We then checked to see if the feature was accurately stating the nearest restaurant by using Google maps to find out which restaurant was actually closest. To check the email activity, we changed the "To" field of the actual email once inside the actual email to our own emails to check that the email was actually sent (we suggest you do the same when you test). We had to test this feature on the device, because the emulator would not pull up any email apps to use. To check that the write reviews and view restaurants/reviews features worked, we created several test reviews for each restaurant, some with photos and some without, and checked that they were actually being uploaded onto the database and presented to the user on the device. We checked this on both the device and the emulator.

Usage

Our app requires using a Google account for sign-in. Feel free to use your own, however we created an account to use for testing and you are more than welcome to use it as well. The email is murkrow.cs.4720@gmail.com and the password is SherriffRox. Also, if the Find Restaurants feature opens with the map focused on the UVA rotunda, this means that the device has not grabbed its gps location yet. Go back to the main activity page and reopen the Find Restaurants feature. The map will now show the device's location.

Lessons Learned

We learned that doing sign-in with Google accounts is a lot easier than trying to design your own sign-in system. Not only is it better for the user because it is one less account they have to worry about, but it is also easier for the developers because Google makes fantastic tutorials to follow. We also learned that you need to check if the device has loaded a user's location before trying to plot it on a Google map because otherwise it will cause the app to crash. We now understand that accessing a device's location is not instant and we need to build in time for that.

Furthermore, in HCI it was taught that changes cost 100 times less when implemented before coding is done. Before, this seemed like an outrageous. However, because our original design and plan were changed to fit practical purposes so many times, we now understand why it is important to have a concrete plan that has gone through many stages of design, before implementing it in code. It is a huge hassle to revise things over and over again. We would probably be 100 times less frustrated if we did not have to implement these changes to our original proposal.

Wireframe Explanation

When the user opens the HooWantsBrunch app, they are brought to the login page, which features the app icon (but larger because we but a lot of effort into it and want to show it off). The user presses the sign-in button, which prompts them to sign-in using their Google account. Once the user signs in, they are taken to the main activity page, which features four buttons for them to click. The first button, "Add a Review" takes the user to the Add a Review activity. There, the user chooses the restaurant they wish to review from the drop down menu, then types in their review. The user can then choose to take a photo and add it to the review. Once the user is satisfied with their review, they hit the submit button, which takes them to the View Restaurants activity. The second button on the main activity page, "View Restaurants" also takes the user to the same page. There, the user can choose a restaurant from the drop down menu and view the reviews posted for that restaurant. At the bottom of the screen, the user can see the three photos most recently posted in reviews about that restaurant. Back at the main activity page, third button, "Find Closest Restaurant to Me," takes the user to the Find Restaurants activity.

There, the user is greeted with a Google map which shows the user their location, as well as the locations of the brunch restaurants included in the HooWantsBrunch app. At the top of the activity page also features a line of text which tells the user which restaurant is closest to their location. If the device has not loaded the user's location before the map is opened, the map shows the location of the UVA rotunda and tells the user that it cannot calculate the closest restaurant, prompting them to reload the activity. Back at the main activity page the fourth button, "Contact Restaurants," takes the user to the Email Restaurants activity. There, the user chooses a restaurant to email from the drop down menu. The user will note that not all of the restaurants included in the HooWantsBrunch app are in this menu; this is because not all of the restaurants have email addresses for patrons to contact. Once the user has chosen the restaurant, they then write their message in the text box. Once they are done, they click the submit button, which causes the device to prompt the user to continue with their email app of choice, such as gmail. The device then takes the user to that app with the email populated with the information they wrote in the HooWantsBrunch app. The user can then send the email.