Torry Yang ty5kj
Patrick Anderson psa5dg
Gengar-Android
Free Food App

Our app is designed to connect students looking for free food with event coordinators looking for increased interest in their organizations. The concept is very simple. When an event coordinator plans to host an on-grounds event that will also provide free food to any and all guests, they will create an event on our app with a picture of the food or the organization logo, as well as a title, description, date & time, and location for the event. The ideal use of this app is for a student to open it when they have some free time to see if there are any ongoing events that are serving free food. They can then go to the event, grab some food, and then learn more about the event, which they might not previously have done had they not known that there would be initial incentive to check it out.

We decided to use Android over iOS mainly for coordination concerns regarding the fact that one member of our group does not own a Mac OS device and therefore cannot easily program iOS apps. Programming in Android made it much easier for the both of us.

The three main features of our application are the event feed, the event info page, and the event creation screen. When the event feed initially loads, it pulls all of the data from our web server and stores it in a local database on the device which allows for more convenient storage. Refreshing the feed by pulling down will then update the local database with any events that might have been created at the time. When the user clicks on any of these events, they will receive a more detailed description of the event, full of information about the place and time of the event as well as any further descriptions. Creating an event involves a simple process of filling out some fields in the proper page, as well as attaching an image of the food items the user wishes to advertise. A final confirmation button sends this event to our remote database for others to view from their devices.

Aside from standard programming convention and proper coding techniques, we used Stetho to work with the SQLiteDatabase, as well as Postman to test POST requests on the web service.

At times we would use toasts to display current statuses of our app, and constantly made sure to “code a little, test a little” when working through difficult situations. Understanding which parts of our code worked and which parts didn’t was essential to our success.

No information for usage is necessary.

One of the biggest lessons we learned was our timeline management. This project was naturally much more difficult than the past couple of projects because of the mere fact that we were spending much more work on it over a longer period of time. This meant that a “cramming” approach was not nearly as viable as it could have possibly been for a shorter assignment. Luckily, the milestones helped us a lot on this regard, as it held us accountable for our success in the application. Furthermore, we learned that communication was key for smoothing any situations out that were causing us difficulties. At times we would be working independently on separate parts of the application and would need to know what the other person was doing. Being able to clearly state the necessary help we each needed from each other made our lives much easier. Technically speaking, we learned several new ways to build apps, from implementation of local databases to the connection of a web service. There were several requirements for the app, and therefore a lot of new knowledge was packed into our brains and then transmitted onto the project.

We also learned that we cannot expect the UVA servers to be running at midnight the night before the project is due, which frustrated us significantly, although was begrudgingly accepted by us as a combination of high hopes and poor planning on our end.

The wireframe PDF should be located in the GitHub repo.