Geoffrey Pard

CS 496 – 400

FINAL PROJECT

DUE 12/4/16

JUNK TRUCK

Access Code Repository on Github here: https://github.com/gpard77/cs496_final

Video of functionality here: http://web.engr.oregonstate.edu/~pardg/CS496_final/project.html

Synopsis:

For the final project, I created an app tentatively titled "Junk Truck." The premise is that it is an Uber-ish way to make money by hauling other people's junk. Users can sign up for an account and follow a public feed of job listings. An admin creates this list and divulges job details to registered volunteers. The details include pickup and drop locations along with the offered payment. A volunteer can select a job and when the requirements of the job have been met payment is released and the job is removed from the public feed.

Details of API:

The backend of this project is coded in Python and served with the assistance of Google App Engine. The app utilizes Google's non-relational ndb database through the datastore. There are two major entities: Jobs and Members.

Jobs include the following properties:

- Caption – a heading for a posted job
- Street – street address of pickup location
- City – city address
- Zip Code
- Drop – location of disposal

- Offer – the amount offered for payment

Members include the following properties:

- User Name
- Passcode
- First Name
- Last Name
- Jobs – a list of jobs that a member has selected

API ROUTES:

Get

- https://pardg-cs496-junktruck.appspot.com/job Returns list of jobs
- https://pardg-cs496-junktruck.appspot.com/job/{id} Returns specifics about a job
- https://pardg-cs496-junktruck.appspot.com/job/search?caption={?} Search by caption
- https://pardg-cs496-junktruck.appspot.com/job/search?zip_code={?} Search by zip code
- https://pardg-cs496-junktruck.appspot.com/member Returns list of members
- https://pardg-cs496-junktruck.appspot.com/member/{id} Returns specifics about a member
- https://pardg-cs496-junktruck.appspot.com/member/search?user_name={?} Search by user name
- https://pardg-cs496-junktruck.appspot.com/member/search?last_name={?} Search by last name

Post

Curl representations as follows:

Add a Job

- curl --data "caption=?" --data "street=?" --data "city=?" --data "zip_code=?" --data "drop=?" --data "offer=?" -H "Accept: application/json" https://pardg-cs496-junktruck.appspot.com/job

Add a Member

- curl --data "user_name=?" --data "passcode=?" --data "first_name=?" --data "last_name=?" --data "jobs[]=?" -H "Accept: application/json" https://pardg-cs496-junktruck.appspot.com/member

Put

Remove a Job from Member's Job List

- curl -X PUT -H "Accept: application/json" -d "" https://pardg-cs496-junktruck.appspot.com/member/{?id}/job/remove/{?id}

Delete

Delete a Job from Public Job List

- curl -X DELETE https://pardg-cs496-junktruck.appspot.com/job/{?id}

Delete a Member from Volunteer List

- curl -X DELETE https://pardg-cs496-junktruck.appspot.com/member/{?id}

Account System:

New to Android development, I tried to stay away from third party libraries so as to build up my foundational android knowledge. Therefore, I did not use third party authentication for this project. I created a system of authentication against the API and datastore entries saved on the server. Essentially, I created an Admin login and that administrator exists as a sort of overseer of the public feed. Once authenticated by the server, an administrator can add and delete jobs from the public feed. All other users create a user name and passcode to enter the site. Once logged in, admin rights are unavailable to them. Also, a logged in user has no access to any other user's information.