Interests, summer activities, college plans

Full time availability

Start date and end date

Vacation time – one week + 2 days off

Current Android developer

* Started work on Android app
* Laid some of the fundamental groundwork (interaction with Parse backend, different app screens, sending/receiving/playing links)
* Work remaining: polishing existing functionality + advanced features
  + Address book integration + friend suggestions
  + Data updates and push notifications
  + Support for different devices
* Moving to new project
* Will still be involved with Android and help you get up to speed with code base

Interesting problems involved in building the mobile app

* Messages and data concurrency
  + Message thread between users A and B, who are interacting in real time on the app
  + To save an object in the cloud, you either call saveInBackgroundWithBlock on a new object of a particular class or an existing, local instance (previously retrieved) of an object
    - The server copy of the object is overwritten by the local copy
  + User A writes and sends a message to B
    - Updates local copy of message thread M -> M + A
    - Posts message to server (takes ~few seconds to reach)
    - Server sends push notification/update to user B’s phone which automatically updates link on B’s phone
    - At the same time, B is updating her copy of message thread M -> M + B
    - If B presses send before update from A has reached her, then she will save to Parse M + B instead of M + A + B as desired
    - Fundamental issue: network latency
* Data update queue
  + Lets say user has push (remote) notifications switched off so we load data regularly every 10 seconds
  + After user sends a song, we want to immediately show sent link loading in inbox
    - If we already have a data update underway, what do we do to ensure that the user sees the new link in his/her inbox?
      * 1) wait for the previous data update to complete before starting a second update
      * 2) interrupt/cancel the previous one
    - Current solution: queue of data updates with different priority levels
      * Data update cancelled if newer data update in queue has higher priority level
* Paging content and performance
  + Want content to load completely so user who wants to scroll through all links gets a responsive experience, but also do not want long load time
  + Solutions
    - Cache data on file system in semi-persistent manner
    - Use paged server requests to only request 50 links at a time and require user to click “load more”
    - Load low or mid res artwork for all links and then replace with high res images as user scrolls

Exciting times for LinkMeUp

* Version of 2.0 of iOS app coming out this week
* Marketing and growth phase
* Moving from Parse backend to AWS + scaling up

Offer - $5000 for June 25 – Aug 28 (9 weeks + 2 days) + 0-20% bonus