Project 2

Main requirements

(1) Talk with the TA about what non-trivial data your app sends to the cloud

(3) Demonstrate that your app correctly handles intermittent network connectivity

(1) Show the code that stores your data on the cloud

(3) Demonstrate that your code successfully loads the data and uses it for an asynchronous task

Group functionality

- Main activity change: if no group data, show Start page with Create Group and Join Group buttons
    - Create group starts a new list, generates Group ID and user inputs password to be encrypted
        - Adds everything locally and to the cloud as new group
        - Starts session
    - Join group asks for group ID and password
        - Checks against cloud data, if match pulls up list and stores locally
        - Starts session
    - If group data exists, just pull up the list like usual locally
- Add group data to propList class (just fields/attributes)
- Encrypt group password
- Group data is sent and stored on the cloud
- Proplist is local, and also sent and stored on the cloud
    - Handles network problems; data is updated via cloud, but user views it locally (so no issues if network is down)
- Async task checks local proplist against cloud proplist using last change time
    - If there is a change locally, proplist is uploaded to the cloud and cloud's last change time updates
    - If there is a change on the cloud, proplist changes are downloaded locally and local last change time updates
    - Run async task on startup, then every 30 seconds when using app, then every night when not using app

Division of work (getting started)

- Michael – getting cloud stuff started
- Reilly – changing main activity to check for group and direct to create/join page if needed
- Anna – add groupID, password, and lastUpdatedTime attributes to propList; get lastUpdatedTime on add and edit property and update it in propList
- Rex – figure out how to encrypt password and generate random group ID