Selfie with Strangers

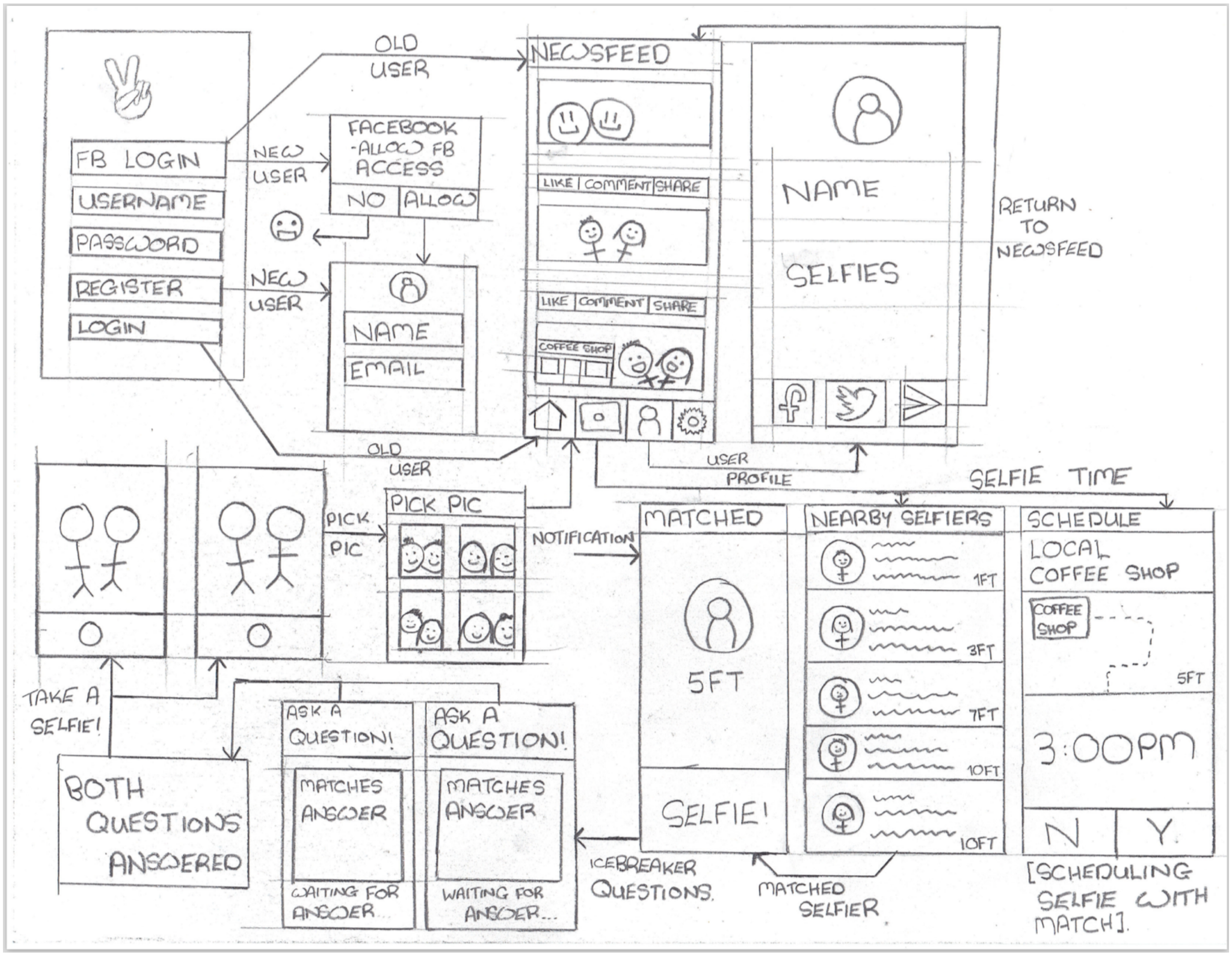
Design Specification

9 April 2015

Summary of the Project Idea

Selfie with Strangers is a low-pressure, low-risk way to meet people around you. Users find other users by being notified when they are close to each other, or by scheduling a meeting for a future time. Once users meet, they tell the other person about themselves and take an awesome selfie together.

Detailed Application Flow



Login:

- Receive information from form and query facebook for authorization
- If need to: Create a new user and save the user to the database
- Loads user session
- User is taken to Newsfeed

General Non-Login Views:

- Should have at least return to Newsfeed button

Newsfeed:

- Maintains a list of X most recent selfies taken in a region (or hold selfies from the previous week/day?)
- Can receive new Selfie to add to list
- Has buttons for View Profile, Search for Selfies, and Settings

User Profile:

- User is able to view all his/her selfies
- Contains basic user information

Search For Selfie:

- Gets location of user
- User gets option of taking a selfie with someone nearby, or someone at a future time and undetermined (but relatively nearby) location
  - Switch between Nearby or Future Views by swiping?
- Nearby:
  - App displays location and image of a nearby user.
  - Displays red border if that user is the one for Future Selfie ???
- Future:
  - Displays location near two people for Selfie and a time
  - Users must both agree for it to be scheduled
  - Decline by clicking NO or swiping?
- Receive decision and then either initiate nearby or future scheduling

Taking Selfie:

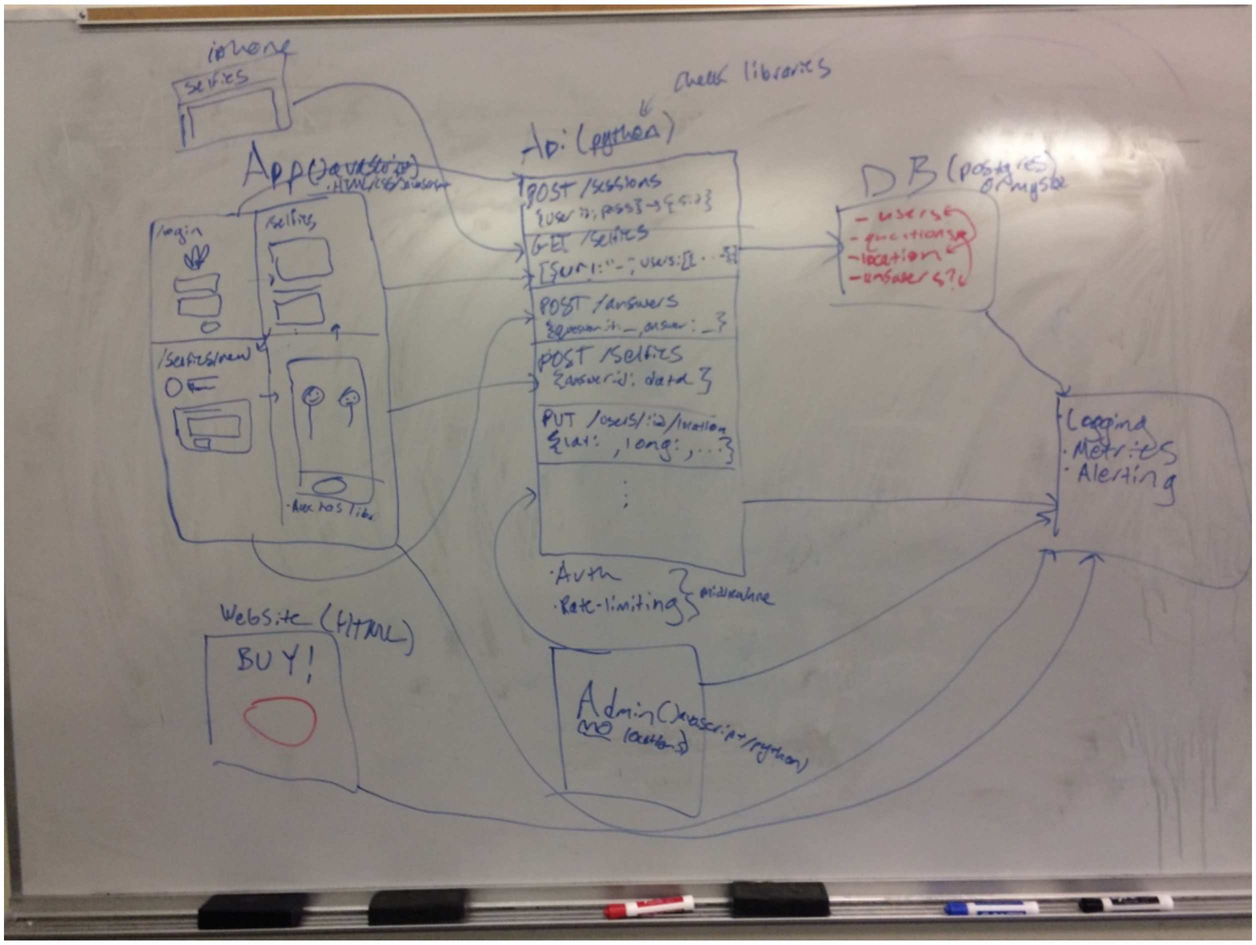
- Answer Question:
  - Display question from database to each user to answer
  - Save answer to database
- When Users Meet:

    - Each user will be required to enter the answer that the other person gave, in order to verify that they have met the correct person
- Take Picture (s):
    - Pictures taken by either user are temporarily stored
- Pick Picture:
    - Display pictures taken by both users
    - If both users allow pictures to be posted to newsfeed, then the selected pictures is posted to the newsfeed
    - The answer to the question is associated with the picture, as well as an optional caption

Settings:
- User is able to update user information and update security preferences
- User is able to specify whether or not selfies will be posted to the newsfeed

Aleksandr Burkatovskiy - 3/27/15 and Caleb Larson - 3/29/15

Architecture Diagram



Bruce Spang - 4/9/2015

Components and API

(include the relevant libraries for each component description)

App service

- express.js/ejs

Api service

- Flask

Login

- Facebook Javascript SDK for the login in button
- oauthlib 0.7.2 - For our own login setup

Newsfeed

- Probably just need Flask for this, it’s only pulling in data

Search for Selfies

- Geolocation: urlopen from urllib, this gives us the options of using something like http://freegeoip.net/json/<IP> to get a JSON object with the geolocation data of that IP
    - Example use here: http://stackoverflow.com/revisions/26165487/1
- Making a unique ID for this match in python (this makes it easy to assign a question to this match and for finding pictures)
    - UUID: https://docs.python.org/2/library/uuid.html

Taking a Selfie

- getUserMedia API for taking a picture
    - http://davidwalsh.name/browser-camera
- Flask has an API for uploading files
    - Found here: http://flask.pocoo.org/docs/0.10/patterns/fileuploads/

Machine Learning

- Scikit-Learn/Scipy/Numpy - These libraries will be used to build statistical models for matching users to each other, suggesting ice-breaker questions, and possibly for suggesting times and locations.

Posting a Selfie

- Upload to Facebook
    - Facebook Javascript SDK for posting a link: https://developers.facebook.com/docs/javascript/quickstart/v2.3#dialogs
    - Facebook Javascript SDK for actually uploading it to Facebook https://developers.facebook.com/docs/javascript/quickstart/v2.3#graphapi
    - Worth looking into if we are truly dedicated to Python:
        - https://github.com/pythonforfacebook/facebook-sdk
        - http://facebook-sdk.readthedocs.org/en/latest/api.html

Coding Details

- Vagrant for the dev environment
- Ansible for provisioning
- Scales/graphite/nagios for metrics

Zac May - 3/31/15 and Wesley Fung 4/1/15

Database Schema

```
CREATE TABLE users (
id              VARCHAR(32),
username        VARCHAR(32),
name            TEXT NOT NULL,
password_hash   TEXT NOT NULL,
created         DATETIME NOT NULL,
updated            DATETIME,
PRIMARY KEY(id),
UNIQUE(username)
);



CREATE TABLE user_locations (
user_id          VARCHAR(32) REFERENCES users(id),
latitude         FLOAT NOT NULL,
longitude        FLOAT NOT NULL,
when             DATETIME NOT NULL
);

CREATE TABLE questions (
id               VARCHAR(32) NOT NULL,
question         TEXT NOT NULL,
PRIMARY KEY(id),
UNIQUE(question)
);

CREATE TABLE locations(
id               INTEGER AUTO_INCREMENT,
name             VARCHAR(125) NOT NULL,
lat              VARCHAR(10) NOT NULL,
long             VARCHAR(10) NOT NULL,
PRIMARY KEY(id)
);

CREATE TABLE answers (
id            VARCHAR(32) NOT NULL,
selfie_id     VARCHAR(32) REFERENCES selfies(id),
question_id   VARCHAR(32) REFERENCES questions(id),
user_id       VARCHAR(32) REFERENCES users(id)
answer        TEXT NOT NULL,
created       DATETIME NOT NULL,
PRIMARY KEY(id)
);
```

```
CREATE TABLE ratings (
selfie_id       VARCHAR(32) REFERENCES selfies(id),
rater_id        VARCHAR(32) REFERENCES users(id),
ratee_id        VARCHAR(32) REFERENCES users(id),
rating          INTEGER NOT NULL,
created         DATETIME NOT NULL,
PRIMARY KEY(selfie_id, rater_id, ratee_id)
);

CREATE TABLE selfie_users (
selfie_id VARCHAR(32),
user_id VARCHAR(32) REFERENCES users(id),
PRIMARY KEY(selfie_id, user_id)
);

CREATE TABLE selfies (
id              VARCHAR(32),
outcome_id      VARCHAR(32) references proposals(id),
proposal_id     VARCHAR(32) references proposals(id),
created         DATETIME NOT NULL,
feedback        VARCHAR,
PRIMARY KEY(id)
);

CREATE TABLE feedback (
user_id         VARCHAR(32) REFERENCES users(id),
selfie_id       VARCHAR(32) REFERENCES selfies(id),
feedback        TEXT NOT NULL,
created         DATETIME NOT NULL,
PRIMARY KEY(user_id, selfie_id)
);

CREATE TABLE outcome_codes (
outcome_id      VARCHAR(32),
description     TEXT NOT NULL,
PRIMARY KEY(outcomeid),
UNIQUE(description)
);
```

```
CREATE TABLE proposals (
id              VARCHAR(32),
proposer_id     VARCHAR(32) REFERENCES users(id),
recipient_id    VARCHAR(32) REFERENCES users(id),
location        POINT,
meeting_time    DATETIME,
accepted        BOOLEAN,
created         DATETIME,
PRIMARY KEY(id)
);

CREATE TABLE distance_histories (
id                      VARCHAR(32),
user1_distance          FLOAT,
user2_distance          FLOAT,
accepted                BOOLEAN,
PRIMARY KEY(id)
);
```

Aleksandr Burkatovskiy - 3/27/15, Oskar Singer - 3/30/15, Bruce Spang - 3/30/15

Revision History