Revised Project Proposal

GroupUs

Rui Ge rg3105

Ruimin Zhao rz2390

Hao Fang hf2345

Shenxiu Wu sw3196

October 9, 2018

1 Background

- Team OS: 3 Mac + 1 Win (We will use MacOS as the platform though)
- Application (Demo): Desktop APP (javafx)
- Language: Java

We came up with this idea of designing a simple platform for Columbia students to group together easily for different kinds of daily activities. We agreed that the three types of common group activities for us are eating, studying and going home together at night when studying till midnight.

2 Use cases

Here are the use cases we came up with so far including both MVP features and wish-list features.

- MVP
  - Sign up/in: Users could register using their Columbia email address.
  - Profile: Users could see past events statistics they posted/joined
  - Post: After log in, users could post group-up events
  - Search: After log in, users could search group-up events
  - Join: After log in, users could join group-up events
- Wish-list (Might Change)
  - Recommendation: When posting eating event, nearby restaurants can be shown in event content.
  - Invitation: Event poster could start event by sending out invitation
  - Notification: Event poster could receive notification for new ”join”

3 User Stories

3.1 MVP User Stories

There are some features to complete so as to deliver a minimum value product for the first iteration.

(i) Login: As a Columbia University student, I want to use this group-up application so that I can find some after-class group events among Columbia students. My conditions of satisfaction are (1) If I use the email address correctly ending with a suffix of @*columbia.edu* for registration, I should be able to sign up properly. (2) If I use email address with wrong format, I will get an error message and be reminded to use my @*columbia.edu* email.

(ii) Search: As a user who has logged in successfully, I want to have the place to search for events so that I can either join a group event or post one if I do not find one fits my need. My conditions of satisfaction is that (1) I could search for events by categories (study, eat, go home) and see their basic information (start and end time, event description). (2) The displayed events should be listed in the order of distance to the user's current location.

(iii) Join: As a user who has logged in successfully and has searched out a list of available events, I want to be able to join the event by clicking on the JOIN button. My condition of satisfaction is that I could see this joined events listed in my profile.

(iv) Post: As a user who doesn't find suitable events, I want to post a new event so that I can find someone else shares the common interest with me. My conditions of satisfactions are (1)I could have a profile to save the events I create.(2)I could specify the start and end time of the event, and give detailed description, location of the event. Both the plain text format and geolocation format of location will be saved into database, because the Yelp API requires plain text location while our "search nearby event" requires event geolocation information for distance calculation.

3.2 Wish-list User Stories

If time allows, we have a list of improvement on features to work on.

(i) Recommendation: For eating events, I hope this App could give us several real-time restaurant suggestions based on the final formed group's information and what we prefer (e.g. the number of people, Location, Cuisine, Dining Time), and also we want to know the weather condition near the restaurants we finally choose if possible.

(ii) Invite: As a user who has logged in successfully, I want to be able to invite certain users to my posted event, so that I could have a private group where members are pre-decided. My conditions of satisfaction are (1) I type in the email address of the user who I want to invite and I can send out an email notification to them. (2) This type of private event cannot be searched by users out of the invitation list.

(iii) Notification: As a user who has logged in successfully, I want to be informed whenever someone has joined my posted event. My conditions of satisfaction is that I could receive an email notification if someone has just joined my posted event.

4 Implementation

4.1 MVP Acceptance Testing

- Login:
  - **User use email address with suffix "@columbia.edu" to register**: The input combination of email address and password will be stored into the database successfully if this email address has not been used before.

    For example, user could type "ab1234@columbia.edu" for register email.
  - **User use email address with suffix other than "@columbia.edu" to register, or the email address already exist in the database**: The registration will not succeed and an error page would prompt out to remind the user.

    For example, user could type "ab1234@gmail.com" and fail to register.
  - **User use registered email address and correct password to login**: User could login and be directed to the user profile page where the

posted and joined events history can be seen there. Otherwise if the password and email address is not correct, user would be re-directed back to login page.

- Search:
    - **User type a keyword that could not match any existing event description**: A reminder message "No result" will be displayed and user will be re-directed back to the search page.
      For example, when user type "searching partner to travel outside", due to it's not in the range of our categories, the user will be told none post match his or her request.
    - **User choose one of the three available categories to see the search results**: The events listed should all lie in the chosen category and the order is consistent with its distance to user's location from near to far. And for each event box, basic information including name, category, time, and event image (if applicable) will be displayed.
      For example, when a user studying in NWC library who wants to find a study partner, he or she can choose the study category to find any interesting post. Result from the order is consistent with its distance, the App will firstly show those people who also studying in NWC and seeking a studying group at the same time. Say, the first event box maybe indicate – name: seeking a study group on the 1st floor of NWC library; category: study; time: 3pm - 12am.
- Join:
    - **User click on the JOIN button on the event box**: User should be directed to profile page and see the joined event box as part of his/her event list in his profile page. And the joined events listed are in order of the start time.
      For example, when a user click the JOIN button to join an event that will start right away (earlier than any other events he/she ever joined), the user will be directed to profile page and see this event being listed at the first position in the list of joined events.
    - **User try to join an event that has already been joined by her/him**: User will be reminded of this and be directed back to the search page.
      For example, when user try to click the JOIN button to join the event

that already exist in his/her profile history, an error message window will prompt out saying “You have already joined this event” right away.

- Post:
    - **User enter basic information of the new event including event name, category, description, start time, end time, location, additional information about location, and event image (optional), and then click on POST button in event posting page**: User could be directed to user profile page and see the posted event listed in the event list. And if user go to search page and search for that event, that event should also show up normally and is available for being joined.

      For example, user could type – Event name: Go back to home together; Category: going home together; Description: Is there someone will back home from NWC library tonight at 11:30pm? We can walk together via Broadway Street; Start time: 11:30pm; End time: N/A; Location: Columbia University; Additional Locational Information: NWC library first floor.
    - **User enter an end time that is earlier than current time**: User will be reminded of this error and be redirected back to the event posting page again.

      For example, when user type – Event name: Form a study group for Leetcode practice; Category: studying; Start time: one hour ago. End time: till today midnight. The user would finally be notified the entering start time is illegal and please choose a time valid.

4.2 Wish-list Acceptance Testing

- Recommendation:
    - **User enter basic information of the formed group or their forecast of demand (e.g. the number of people, Location, Cuisine, Dining Time), and then click on SUBMIT button to get the recommendation result**: Our APP will return 3 restaurant suggestions based on a set of preferences of users’. The return information will include restaurant name and its location. If user determine one restaurant to go, our back-end will automatically query the neighborhood

weather condition of that restaurant if possible and return to users.

For example, the user could type – Location: Manhattan; Cuisine: Japanese; The number of people: three; What time: today, 7pm. Our return suggestion would be: "Hello! Here are the Japanese restaurant suggestions for 3 people, for today at 7 pm: 1. Sushi Nakazawa, located at 23 Commerce St, 2. Jin Ramen, located at 3183 Broadway, 3. Nikko, located at 1280 Amsterdam Ave. Enjoy your meal!"

  - **User enter a dining time which is earlier than current time or the location is out of US**: User will be reminded of this error and be requested to re-write a correct information.

    For example, user could type – Location: Manhattan; Cuisine: Japanese; What time: yesterday, 7pm. Our return suggestion would be: "The dining time is illegal, please re-enter a correct one."

- Invite:

  - **User enter email address of a registered user and click on SEND button to invite the user to the private event** : An invitation email containing information of the event will be sent to the user's email and return with the feedback of successful invitation.

    For example,user wants to invite ab who is already a registered user of our app with registered email "ab1234@columbia.edu",after user types in the correct address and click on SEND button, an email with the event information will be sent to the target user and return a message "invitation sent successfully".

  - **User enter invalid email address(not registered) to invite and click on SEND button** : User will get an error message indicating that "User does not exist" and will be directed back to the initial invite interface after an interval.

    For example, user enter email address of the person with correct suffix "ab1234@columbia.edu" but isn't a registered user in records or "ab1234@gmail.com" with suffix other than "@columbia.edu". After user click on SEND button our app will return an error message "User not exist" and go back to the invite interface.

- Notification:

  - **User get notifications about posted event**: When someone joins a posted event, the initiator of the posted event will get a notification

by email which informs the user that someone has just joined your event. The email will contain the information of the event and the newly joined member. The user can get notifications even he/she isn't at login status.
For example, when user Jack joins a posted event with the topic "reviewing class big data together on Saturday". The launcher of the event Simon will get an email with content: "New member: Jack Event: reviewing class big data together on Saturday"

4.3 Technologies

- Back-end Development: Java (javafx)
- Database: AWS RDS
- API: Google Geolocation API, Yelp API
- Demo: Desktop UI window
- Code testing: JUnit
- CI/CD pipeline: GitHub

4.4 Potential Challenges

- Technology: We are new to Desktop APP development, so there might be a learning curve for us to get familiar with the language as well as the platform.
- Time Management: Before the first iteration, apart from completing the Desktop APP development, we should also finish the corresponding unit test and CI/CD pipeline definition.