RedPins
Requirements and Specification Document
3/22/2013, version 2.0
Members: Benjamin Le, Victor Chang, Jerry Chen, Soo Hyoung (Eric) Cheong, Andy Lee

Project Abstract:

RedPins allows users to find "pins" (or markers) that shows interesting attractions or popular events on a map around their proximity. Other users of RedPins who are familiar with that area can add pins anywhere on a huge map, detailing what attraction is at that location, along with the time and date it is relevant to. Users of all sorts may also share photos to that pin, add comments, other information, etc.

Document Revision History

Rev 2.0 2013-03-22 -version 2, post iteration 1
Rev 1.0 2013-02-13 -initial version

Customer

Our customers will be people who are not necessarily familiar with a certain area and the events that occur in their locality. When people are in a city, they are often unfamiliar with events that happen once a week and may not have formal establishments associated with them. Currently, Yelp solves the problem of finding food for an area by allowing users to search for nearby restaurants with certain criteria. Other than food, other attractions are often missed out on, mainly food stands, food trucks, sceneries, parades, hiking trails, weekend free markets (with closed streets), temporary events (public dances, etc.), etc. RedPins is a solution to bring these attractions that people in an area may often not know about and bring forth attention to things that are often missed upon. Even locals in the area can discover events that they have never heard of before.

Competitive Landscape

Although these are competing services on the market today, rather than directly competing with them, we plan on integrating these services into our app for a better experience.

1. Yelp
    a. Users rate and review local businesses and restaurants
    b. How are we different?
        i. Yelp lists established businesses only, our app is not entirely limited to that. Users can pin any local event and gathering throughout the area as well as established businesses as well.

        ii. Yelp is very location centric rather than event centric. For example, we can only review a venue for an event on Yelp (such as Oracle Arena) but not the event itself (such as a concert). RedPins will focus more on the events themselves.
    c. How can we work w/ them?
        i. When users review an event happening at a venue. We can let users review the venue and push the review on yelp w/ their yelp account credentials.
2. Facebook
    a. Allows users to create events and invite other Facebook users to join them.
    b. Users can also check into a location with their friends on their Facebook timelines
    c. How are we different?
        i. Facebook has a way to list places and events you can check into and post about, but is limited in terms locality (location and time).
        ii. Facebook is primarily used to organize events and to get people to join events.
    d. How can we work w/ them?
        i. Using the Facebook Login API to handle accounts
        ii. Find a users friends on RedPins using Facebook
        iii. Seed user preferences on their likes and dislikes using their Facebook information
3. Foursquare
    a. Users check in at local venues nearby, and check-ins give users points/badges
    b. How are we different?
        i. Doesn't allow advanced check-in, only at the time.
        ii. Foursquare doesn't help you find events that are currently going on in the area.
    c. How can we work w/ them?
        i. Further market research is required.

User Stories (Iteration 2)

<u>Actors Involved</u>

- Normal users with Facebook accounts
- Normal users without Facebook accounts but still want to use the app

Name	**Login with Facebook (need)**
Actors	User that wants to have an account w/o wanting to make one w/ email and password combination
Triggers	Click "Login With Facebook"
Events	- Click "Login With Facebook"

Exit Condition	Click the “Login With Facebook” button
Post-Conditions	Account is created, gains permissions
Acceptance Test	User gains the privileges of logged in users and is directed to their personal homepage.

Name	**Submit Event (need)**
Actors	Logged in user
Triggers	User clicks the “Add Event” button
Events	- User gets redirected to a Map where they can pin their event on the map - User then chooses a location based off where they place their pin or the address they enter in - After confirming the location, user is redirected to an event edit page where user can add tags to the event, add date, etc.
Exit Condition	User clicks submit or cancel
Post-Conditions	User is redirected to the event page he/she just created. All users are able to see this event, the location, and the time it happens.
Acceptance Test	-Events will be created and stored properly in our database, with the right info provided by the user -Posted events will show up in the right location (on our map view) at the right time with the right tags.

Name	**Search (need)**
Actors	Any user regardless of whether logged in or not
Triggers	Click the “Search” button or press anywhere in a search bar
Events	- User enters in search query - User clicks “Search” or a submit button
Exit Condition	User clicks “Search” or a submit button
Post-Conditions	Lists the search results in a scrollable ListView
Acceptance Test	-User will be able to search for an event in numerous ways (by title of the event, by tag, by

	location). -For example, a user would search for “Off the Grid” and the first event that is displayed in the list view is the “Off the Grid” event, since it is a full text match -A user may also search for partial strings, such as “Lawrence” that would bring up “Lawrence Hall of Science Picnic Day” or “Hiking Up Mount Lawrence”. -For now, we are assuming users’ queries are not typos, but in future iteration we may need to deal with typos, misspellings and other “erroneous” searches, possibly through external search libraries.

Name	*Rate user events (useful)*
Actors	Logged In Users who want to rate an event
Triggers	User clicks Thumbs up or Thumbs down button
Events	- User clicks Thumbs up or Thumbs down button
Exit Condition	- User clicks Thumbs up or Thumbs down button
Post-Conditions	- Event gains either one like or dislike
Acceptance Test	- Rating appears on the page of the event. There will be some sort of graphical indication to the user that he or she has rated this event, such as a bolding of the thumbs up/thumbs down button.
Error Conditions	-A rating may not register or be countered properly. What happens if multiple users rate an event at the same time? Methods need to be synchronized. -If a user clicks “like” twice, the expected behavior is ambiguous. Should the event be liked and immediately un-liked or should only the first like be registered?

Name	*Bookmark an event (useful)*
Actors	Logged In User
Triggers	- User clicks “Bookmark” button
Events	- User sees event in the ListView when they submit a search query or accesses the event page. - User accesses the event page by clicking on the

	event in the list view - User Clicks Bookmark icon - User can see this event in the bookmarked list when “bookmark” tab is clicked on the home page
Exit Condition	- User clicks “Bookmark” button
Post-Conditions	- User has bookmarked event added to his/her list of bookmarks
Acceptance Test	- All of the user’s bookmarked events appear on his/her bookmark list
Error Conditions	-If a user has bookmarked an event and that event does not appear on the user’s bookmark list.

Name	*Adding comments to an event (useful)*
Actors	Logged in user
Triggers	User A is on an event page and wants to leave a comment for the event
Events	User A writes comment into a text box and clicks “Add Comment” button
Exit Condition	The add comment button is clicked after the user has entered some string comment into a text field.
Post-Conditions	A comment is left on the list of comments for a page
Acceptance Test	The comment is made, and other users will also be able to see the newly posted comment, along with the user’s name and a timestamp. Comments will be ordered in some way, most likely by time.

Name	**Delete Event (Need)**
Actors	User who created the event initially
Triggers	User goes to event that he created and clicks a “delete” button.
Events	- Delete button is clicked. - Confirmation page pops up
Exit Condition	User confirms that he wants to delete the event.

Post-Conditions	Event is deleted from our data storage.
Acceptance Test	- Event disappears from the list of all events, and will no longer appear in user searches or on maps. - Users who have bookmarked said event will most likely receive a notification of some kind that their bookmarked event has been deleted (to be done in the next iteration). For now, the event will just be removed from the user's list of bookmarked of events.
Error Conditions	-If the user attempts to delete the event but the event does not get removed from the list views/map views.

Name	*Switch From ListView to MapView and vice versa (useful)*
Actors	Any user, regardless of being logged in or not, who have submitted a search query
Triggers	User clicks on "Map" or "List" Button, depending on what mode they are in.
Events	- User submits search query - User clicks on "Map" or "List" Button
Exit Condition	- User clicks on "Map" or "List" Button
Post-Conditions	- The view changes from a list of events to a map with pins on it and vice versa.
Acceptance Test	- The view changes properly upon the button click. The events are listed properly in list view or the pins are displayed properly in map view.
Error Conditions	-In Iteration 1, we experienced crashes of the app when switching between list and map views too fast. This will be debugged through automated UI testing.

Name	*Search nearby (useful)*
Actors	Any user regardless of whether logged in or not
Triggers	Click "Nearby" button
Events	-User clicks "Nearby" button -Collects GPS information of phone and finds pins

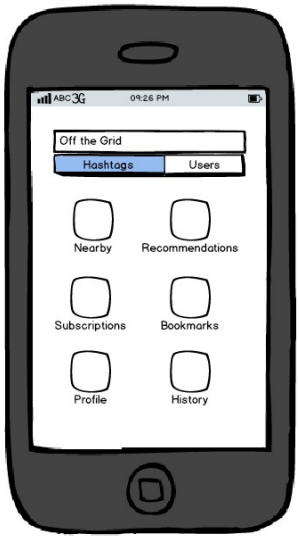
	nearby.
Exit Condition	- User clicks “Nearby” button
Post-Conditions	- User gets redirected to a ListView of nearby events
Acceptance Test	User is able to search and the correct results (events that are close to the user) are returned, within a few seconds.
Error Conditions	-If the “nearby” locations are actually not close to the user at all.

Future Planned User Stories

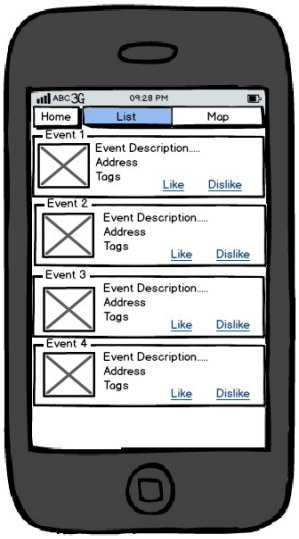
- Users can possibly be able to mark themselves as attending events, and there would be a guest list on event pages, similar to Facebook event guest lists. Other users would also be able to see the guest list, or at least see the amount of people attending.
- Karma system for users who constantly pin good, attractive events, or flag spam/repeat events, or leave helpful comments/advice for events.
- Complete Yelp integration, users will be able to review venues that pinned events are held at on Yelp. This could be integrated with a simple button on an event page that would take the user to a Yelp link of the corresponding establishment and allow the user to write a review on there.
- More integration with maps (providing directions to events/venues). Similar to how Yelp has directions to establishments on their individual pages, our event pages would have directions to the event, hopefully through Google Maps API.
- Account creation system. Users would be able to create their own accounts and subscribe to other users’ activities on the application.
- Recommendation algorithms that will be based off user preferences. This was originally planned to be integrated with Facebook, but this is essentially searching by tags, functionality that we already have planned.

User Interface Requirements

- 1st iteration: Home page, search events (list view and map view), adding events page, and event pages (as shown below)
- 2nd iteration: Users’ bookmarked events list, searching for users and the separate list views for displaying users
- Later iterations: User profile pages, subscription list for all events, possible “attending” guest lists, adding photos to events

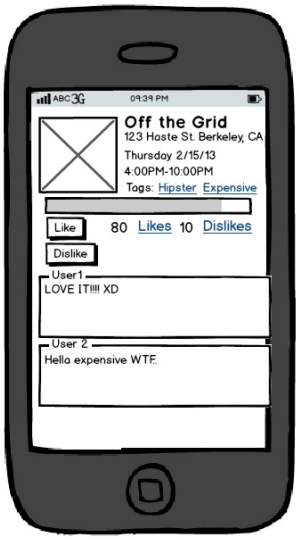


Home page



Event List View

Events Map View



Event Page