Say Hello App Backlog

“Dethaw for the world of icebreakers.”

Team 17

Sakshay Bhatnagar
Derek DeWitt
Levi Heilman
Soumya Kumar
Levi Miller
Nicholas Rishel

Problem Statement

In a social gathering people often want to talk to each other. It is not always easy to start a conversation; often they are unsure who is receptive to conversation.There is no tool to help people find others open to random conversation, retain them in their contacts, and be aware of their attendance at nearby events.

Background

Today, everyone uses social media to meet up with new people. But there is no other app that allows people to meet up with someone new depending on their location. Our app allows the user who is in a completely new environment or social event (where they may only know one or two people) to meet up and start chatting with a complete stranger. Some of the examples of an event could be meeting new people in a classrooms,bars,concert. The bottomline is to help people socialize in large group gatherings. Therefore, the application would be very beneficial anytime there would be an awkward social interactions, such as callouts, bars, orientations, weddings, hangouts, etc.

There are apps like Hotspot and WhosHere that very similar in concept. However, Hotspot allows user to create events before reaching the location. The limitation in this app is that it shares that events with only the user's friends. Whereas our app allows the user to create or "check-in" to an event and to start a conversation with someone they have never met before. Whoshere does not help organize people in terms of an event. It only allows the user to display his or her location to meet up with someone new. Our app will not only create an event it will also allow others to check-in to that event as well as display nearby events that other users are apart of.

System Model

The first interaction, shown in figure 3.1, shows how events are created. A coordinator will open the application and create a new event. The coordinator will then be able to edit the event by changing the name, location, and additional information. This new event will be sent to the server and be added to the database of events. Once the event is in the database, clients will be notified that this event has been created and will be given the option to join. Clients will also be able to search for events via a menu.

The second interaction, shown in figure 3.2, shows how clients will be able to join available events. After clients have received the notification or selected an event from the menu, they will be able to join the event by clicking a "Join" button. Then, they will be added to the group of attendees of the event. The attendees that are already part of the event will receive a notification that the new person has joined.

The third interaction describes what people will be able to do once they are in an event. Users

will be presented on screen with a list of attendees in the event. At this point, they can click on an attendee and initiate a conversation. Conversations will be one on one and users will be able to see which attendees are already in a conversation.

The fourth interaction, shown in figure 3.3, shows how attendees will share their contacts with each other after a conversation or at any point during an event. Both attendees will have to accept the offer of sending contacts. Once they have both accepted, their contact information will be sent to the server and back to the opposite attendee.

The fifth and final interaction, shown in figure 3.4, shows how the GPS location of the client will be sent to the server. The client will get the GPS information from the device and then send this information to the server. This information is used in order to show the client available events in a relatively close proximity. The Geo Location interaction will be used throughout the other interactions in order to keep the available events menu updated.

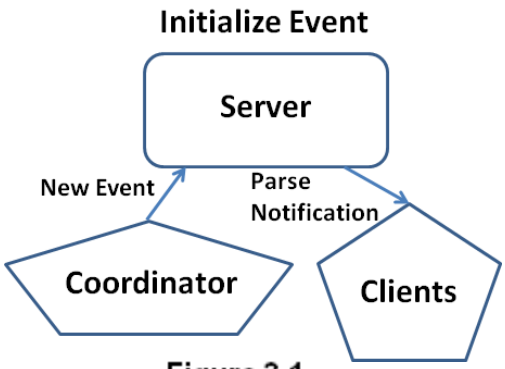

Figure 3.1

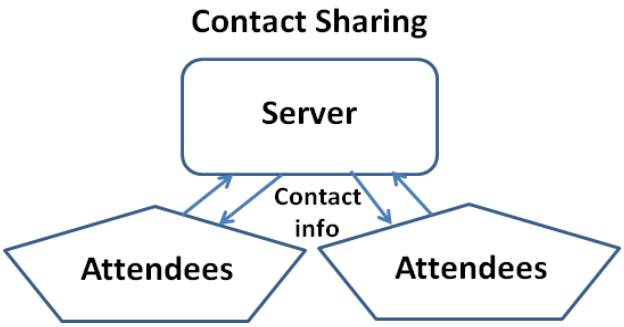

Figure 3.2

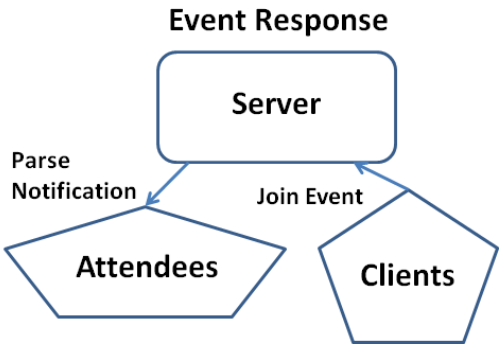

Figure 3.3

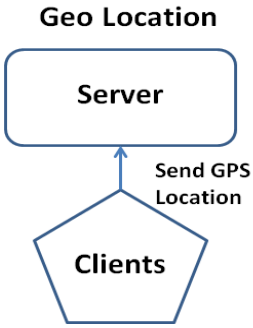

Figure 3.4

Requirements

I. Must be done
   A. Functional
      1. Allow the user to create an event
      2. Display nearby events that have been created
      3. Display users (name and picture) that are already "checked-in"
      4. Allow the user to "check-in" to an event
      5. Allow users to post a description of their specific location at the event
      6. Allow users to start a text conversation with other "checked-in" users
      7. Allow users to confirm personal contact information that may want to be shared
      8. Allow users to exchange all or parts of their saved contact information
      9. Allow users to record a brief summary or important info from a conversation
      10. Display all users that you have had a conversation with (along with any shared contact information)
      11. Display brief summary associated with corresponding conversations (if available)
   B. Non-Functional
      1. 1. Usability - Follow mobile standards for size, location, etc, easy to use and interact, minimal button taps to execute any features.
      2. 2. Security - Should be secure to use. Private data should not be visible. User should be at full control of what he wants to show to which users. should have privacy control.
      3. 3. Performance/Response time - Should give users high response time on interaction with the app and allow low utilization of system resources to work as a lightweight/fast program.
      4. 4. Scalability - The app must be scalable and should be able to handle large amount of traffic at time and need.
      5. Robustness - The program should be robust and should not crash.
II. Will be done if time allows
   A. Functional
      1. Allow the user to mark other users as "favorites".
      2. Allow users to easily connect through social media (facebook, LinkedIn, etc)
      3. Allow users to post a simple greeting that other "checked-in"users will see
      4. Send a notification to the user when a "favorite" is nearby or has "checked-in" to a nearby event

B. Non-Functional
    1. Should work offline. - The system should be accessible offline and and record user interaction and should soon update it online once it hooks to the network.
    2. Platform Compatibility for multiple devices - should be available on broad area of platforms to start with iOS, Web and Android(Android is our first target device) and other smartphones.
    3. Privacy - should give a greater control on individual to control their privacy level like for example let a user easily hide a user from other specific users etc.
    4. Redundancy/Backup - should allow users to backup their data and recover their profile in case of emergency.
    5. Extensibility - Next version of the software should carry over the last version’s feature and in some cases to able to improve on those features.