**House Rules | Functional Requirements**

**Project Description**

For our group project, we aim to create an android app capable of facilitating the process of hosting and/or attending social events for our users. The app makes managing events easier by allowing the host to include any rules that they wish their attendees to abide by so that expectations of the event and attendees can be established prior to the start of the event. As such, we have named our app House Rules due to the ability our users will have to host/attend a unique social event that they feel will suit their wishes.

In the past, both individuals and groups alike have experienced difficulties raising awareness about their upcoming events, whether they be a new local band, small business owner, or even a large company. Likewise, people looking to attend events have encountered difficulties of their own, whether it’s a group having trouble meeting up or an individual simply looking for something fun to do for the weekend. Our app’s purpose is to suit the needs of all people who encounter these issues as it is specifically tailored for people who want to host/attend social events.

The goals for our app are split into two main categories, namely those for hosts and those for attendees, although a user can be both a host to an event and an attendee to different event. First of all, one of our primary goals is to have all users create a House Rules account which they will use to access the entirety of the app. Next, the user will have the option to either host an event or search for nearby public events through a search radius around their current location. Hosts will have the ability to set a date, time, location, send invites, approve/deny invite requests, edit the house rules, and be able to set their events to be public or private. Hosting a public event would mean that anyone looking for an event to go to, who is not invited, would be able to see all the details of the event except for the exact location, whereas invited users can see all the details of the event. Hosting a private event on the other hand would make it so only invited users can see the details of the event and the event will not be visible to any other users. Furthermore, once the event has been created, people who were invited to the event will be able to see the event location in a map. Hosts will also be able to see the number of invites they sent as well as how many users have accepted/denied their invitation.

Now, as for people who want to attend an event, they will have the option to search for nearby public events. When searching for a public event, all nearby events will show up along with all the event details except for the location. The user will then be able to decide what event they would most like to attend and request an invitation from the host. If the invitation request is approved, then the user will have access to all the event details just like any other invited user does, as well as location. On the other hand, if it is denied then they will not be able to see the location of the event.

There are several challenges that our group might encounter along the way as well as some far reaches that we hope to include in the second iteration of our project. First of all, we hope to be able to use the Facebook API and Google API to allow users the option to log into the House Rules app. Also, we hope to use the Google Maps API to not only show the location of an event on a map, but to also give users directions to the event. For the Facebook and Google APIs we will need to learn how to integrate them into the design of our project as well as learn how they work, and for this reason we are unsure if we will leave them for our second iteration or if we will be able to integrate them into our first iteration. A feature we would like to include in our project but will leave for the second iteration as well are user ratings and reviews for hosts of events. We believe the ratings would be helpful for attendees deciding whether they would like to attend an event by seeing whether the host has received generally positive or negative ratings from past users who have attended their events. To supplement this, the host reviews would allow users to leave written reviews about their experience at the host’s event and would also be helpful for other attendees to know what they can expect besides what has been outlined in the house rules of the event. Of course, only users who have attended the events would be able to rate and review the host’s event so as to avoid fake reviews and unfair ratings from users who did not attend the hosts’ event. Finally, we hope to create this app using Android Studio, but before we can create the app, we will need to learn how to use Android Studio to develop our app.

Upon successful completion of our first iteration, we hope to have a functioning prototype that will allow users to host specialized social events, search for nearby events to attend, respond to event invitations, and be able to use the necessary features outlined in our project description to be able to accomplish these various uses.

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Description** | **Author** |
| Inception Draft | January 23, 2017 | First iteration of draft | Edward Guardado III |
| 2nd Draft | January 27, 2017 | Alterations and Additions | Whole Group |

**Vision**

We envision a next level social event application that will provide clients with many options to host or join local social events. We wish to distinguish ourselves from other applications such as facebook events, email invitations, and physical invitations to provide clients with a simplistic UI to allow them attend social events with relative ease.

**Problem Statement**

Traditional forms of invitations lack many features that can help clients. Such features include directions to an event, live information during an event, and reaching a target group for public invitations. Our application looks to bring an overall solution to these lack of features to improve and create better overall social events.

**Key High Level Goals**

With this app, we wish to give users a simpler event hosting experience by allowing them to easily create, share, and search for events. Not only that, but users will also be able to personalize said events with a variety of attributes, such as their date, time, and location. Our app would be aimed toward anyone who wants to host and share such events, from party planners trying to reach a wider audience to teachers who wish to host study sessions for students. With the existence of so many “social” networks, we wish to create a more interpersonal experience for our users, by allowing them to branch out and meet others in the real world, as opposed to “liking” them from behind a screen.

**System Features:**

* Account Creation
* Event Sharing/Creation
* Invitation Sharing/Creation

**USE CASES**

**UC1 | Creation of Account**

**Primary Actor:** User

**Stakeholders and Interests:**

* User: Wants to create an account because they are interested in attending/creating an event(s)
* User Authorization Service: Make sure that the User has a valid email, Google account, or Facebook account.

**Preconditions:** User has downloaded the application and possesses a valid email account or alternative.

**Success Guarantee (Postconditions):** User is registered and information is saved onto the server securely.

**Main Success Scenario:**

1. User opens House Rules Application.
2. Log in page is displayed
3. User clicks button titled “Sign Up”**.**
4. Application changes screen to registration form with Facebook and Google buttons
5. If Facebook login is used, signup handled by Facebook login services.
6. If Google login is used, signup handled by Google login services.
7. If User decides to fill out form they will follow these steps:
   1. Enter full name
   2. Enter screen name
   3. Enter Email
   4. Enter Birth Date
   5. Enter Password
8. Form is sent to SQL database.
9. Screen switches to “Congratulations” page and a verification email is sent to user’s email.
10. Screen Switches to home page of the application.

**Extensions (or Alternative Flows):**

1. User clicks “Sign Up” and they are already registered.
   1. An option to go back will be enabled to return to the login page
2. User uses Facebook or Google to login and forgets their password, email, or just accidentally ends up clicking it.
   1. An option to go back to the “Sign Up” screen is enabled.
3. User opts to use our form and doesn’t have a valid Email or an email already taken.
   1. The User will be informed of the issue and Asked to enter a valid email
   2. The User should enter a valid email and be able to move forward

**UC2 | Creation of Event**

**Primary Actor:** User

**Stakeholders and Interests:**

* User: Wants to create and invite other users to an event based on their privacy settings (whether they want the event to be public, shared with only friends, etc…), the time of the event, the event’s location, and a description of the event.
* System: Will validate the location and time of the event and send out invites based on the user’s privacy settings.

**Preconditions:** The user has input valid times and locations into the respective fields and has set who the event should be shared with.

**Success Guarantee (Postconditions):** User creates the event, the event’s information is saved onto the server, and the information about the event is shared with the correct users.

**Main Success Scenario:**

1. A signed in user opens a form to create an event.
2. User would enter the information about their event, including its location, time, privacy settings, and description.
3. Form is sent to SQL database.
4. Information about the event is made available to those with whom the user wishes to share.

**Extensions (or Alternative Flows):**

1. User accidentally chose the option to create an event
   1. An option to go back to the home screen will be available
2. User accidentally invited another user
   1. The host will always have the ability to invite and un-invite users so long as the event is active
3. The user forgot to invite a certain user
   1. The host can still send invitations to users so long as their event is active

**UC3 | Responding to an Invitation**

**Primary Actor:** User

**Stakeholders and Interests:**

* User: Wants to respond to an event from the invitation of another user(s) (whether they want to accept or decline).
* System: Will take User’s decision and will add the user the list of people that will or will not be attending the event.

**Preconditions:** The user has an invitation from a given event and indicates that they would or would not like to attend the event.

**Success Guarantee (Postconditions):** The server is updated to reflect the user’s decision.

**Main Success Scenario:**

1. A user opens up a form to respond to an invitation from a given event.
2. The user will respond by either accepting the invitation or declining it.
3. If the user accepts the invitation
   1. The system will add the user to the list of people attending the event
   2. The user will have access to all the details of the event
4. If the user declines the invitation
   1. The system will add the user to the list of people who declined to attend the event

**Extensions (or Alternative Flows):**

1. User accepted the invitation accidentally
   1. The user will have the option to decline the invitation even after accepting it, so long as the hosts keeps the invitation offer active
   2. After declining the invitation, the system will remove the user from the list of people attending the event and instead add the user to the list of people who will not attend
2. User declined the invitation accidentally
   1. The user can at anytime change the invitation status from declined to accept so long as the host keeps the invitation offer active

**UC4 | Searching for an Event**

**Primary Actor:** User

**Stakeholder and Interests:**

* User: Wants to search for an event to attend that is near their current location
* System: Will give the user search radius options and will take the user’s input for a search radius

**Preconditions:** The user has the app downloaded and has an account set up that allows them to access the event search feature.

**Success Guarantee (Postconditions):** The system will show the user a list of events that are nearby according to the input search radius. The user will then be able to decide if they would like to request an invitation to an event.

**Main Success Scenario:**

1. The user chooses the option to search for nearby events
2. The system will prompt the user to choose a search radius (in miles)
3. After choosing a valid search radius, the system will then display a list of events located within the search radius

**Extensions (or Alternative Flows):**

1. The user chose the option to search for an event on accident
   1. The user will have the option to go back to the home screen

**UC5 | Requesting an Invite**

**Primary Actor:** User

**Stakeholder and Interests:**

* User: Wants to request an invitation to attend an event displayed by the system’s search for nearby public events list
* System: Will deliver the user’s invite request to the host of the event

**Preconditions:** The user has entered a search for nearby public events that yielded a list of events for the user to choose from

**Success Guarantee (Postconditions):** The system will deliver the user’s invite request to the host of the event

**Main Success Scenario:**

1. The user chooses an event from the list of nearby public events displayed by the system
2. The user sees the details of the event including the house rules and chooses the option to request an invite
3. The system then delivers the request to the host

**Extensions (or Alternative Flows):**

1. The user accidentally chooses to send an invite request
   1. User can cancel the request at anytime so long as it has not been accepted or denied

**UC6 | Responding to a Request for an Invite**

**Primary Actor:** User

**Stakeholder and Interests:**

* User: The user is the host of an active event who receives an invite request from another user who would like to attend the host’s event
* System: Provides the host with the invite request, along with the option to accept or decline the request. The system will also send the user a notification of whether their request has been accepted or declined.

**Preconditions:**  The host has created an event and another user has requested an invitation for the event.

**Success Guarantee (Postconditions):** The host will have accepted or declined the invitation request delivered by the system initially. The system will then send the user who sent the invitation request a notification of whether their request was approved or denied.

**Main Success Scenario:**

1. The host receives a notification when a user sends an invitation request
2. The system give the host the option of accepting or declining the request
3. If the host accepts the request:
   1. The user who sent the request will be notified that their request was accepted
   2. The user will now have access to all the details of the event
4. If the host declines the request:
   1. The user who sent the request will be notified that their request was declined

**Extensions (or Alternative Flows):**

1. The host accidentally accepts the request
   1. The host always has the ability to decline invitations at any time so long as the event is still active
2. The host accidentally declines the request
   1. The host can send the user (and any other user) an invitation request at any time

**Glossary:**

**Possible terms to include:**

1. User - Someone who has downloaded the app and wants to either host a event, search for a event, or respond to an invitation.
2. Host(s) - A user(s) throwing a event
3. Invite Request - An invitation to an event
4. Public Event - An event in which anybody can see and request
5. Private Event - An event in which a user must be invited to see
6. House Rules - The rules in which the Hosts want users attending the event to follow whether they be game rules or event rules
7. Event Search - Search for public events based on attributes such as types and names
8. Public Events List - The format in which public events will be listed in order to protect location security

**SCHEDULE:**

|  |  |
| --- | --- |
| **ITEM** | **DUE** |
| Tech Req | January 31, 2017 |
| Development Officially Begins | February 1, 2017 |
| Pre SuperBowl Hurah - CodeSaturday!!! | February 4, 2017 |
| Half Way Meeting For First Iteration | February 11, 2017 |
| Launch Day - Beta | February 24, 2017 |
| Demo Day 1 | March 6, 2017 |
| Demo Day 2 | March 8, 2017 |
| Final Release | March 17, 2017 |