Butterfly

Design Document

Team 19

Alec DeFano, Muhammad Hamza Farrukh, Travis Kent, Nicholas Ramser, Khanh Tran

**Purpose**

Many social applications are bloated and distract users from the main reason they installed it in the first place, to allow communication and to bring them together. Our goal is to build a service that will make it easier for users to find or establish communities based on similar interests. Butterfly will also provide users with customization options and the tools necessary to easily organize and promote events within those communities.

Functional Requirements:

**General User Functions:**

As a user, I would like to:

1. login using an existing account (Facebook, Google+).
2. create my own communities and be the community founder.
3. see a brief tutorial about the app.
4. see who in my contacts list is already using the app.
5. control the frequency and manner of receiving notifications.
6. search communities based on event activity levels.
7. search communities based on category or name.
8. search communities based on my stated interests.
9. view automatically generated suggestions based on my interests.

**User Profile Functions:**

As a user, I would like to:

1. edit my profile page to display interests and personal information.
2. choose what to display to other users.
3. upload a profile picture or import one from the web.????????????????????/
4. see my participation rating.
5. give another user a point.
6. see the profile of other users of the app.
7. see whether the user is a leader or member of any community.

**Calendar Event Functions:**

As a community member, I would like to:

1. have the group’s calendar synced to my own calendar.
2. view listed events.
3. see information about the event.
4. be able to RSVP an event.
5. be notified about community events.

As a community leader, I would like to:

1. add events on to the community calendar.
2. view events on the community calendar.
3. edit events on the community calendar.
4. delete events on the community calendar.

**Attending Event Functions:**

As a user attending community events, I would like to:

1. share pictures and videos of events I have attended.
2. view how many members are attending events.
3. view which members are attending events
4. be able to check in when I arrive for an event.

**Neighborhood Functions:**

As a user in the neighborhood, I would like to:

1. view a list of all communities in the neighborhood.
2. view a calendar with all planned events of communities in the neighborhood.

**Community Functions:**

As a community member, I would like to:

1. invite anyone to my communities.
2. leave the communities that I am a part of.
3. message other people in my communities.
4. see a community board of messages.
5. see a list of members in the community.
6. create a crew within of a community.
7. create a hangout to meet other members.
8. browse hangouts that have already been created.
9. join existing hangouts.

As a closed community member, I would like to:

1. send a notification to all members of the community.

**Community Board Functions:**

As a community member, I would like to:

1. see a feed which shows recent activity in the community.
2. see a board of user messages.
3. be able to comment on user messages.

**Hangout Functions:**

As a community member, I would like to:

1. create a hangout.
2. leave a hangout.
3. edit my posted hangouts.
4. view hangouts posted by other community members.

**Crew Functions:**

As a member of a crew, I would like to:

1. have a chat with other members in my crew.
2. invite members to my crew.
3. see a list of members in my crew.
4. see the profile page of the members in my crew.
5. send a notification to members of my crew.

**Community Leader Functions:**

As a community leader, I can:

1. remove users from the community.
2. post large planned events for the community.
3. post special messages on the community board.
4. use tags and keywords to describe the community.
5. edit the profile page for the community.
6. upload or import from the web a community profile picture.
7. appoint other users to position of leader.
8. leader the community board.

Nonfunctional:

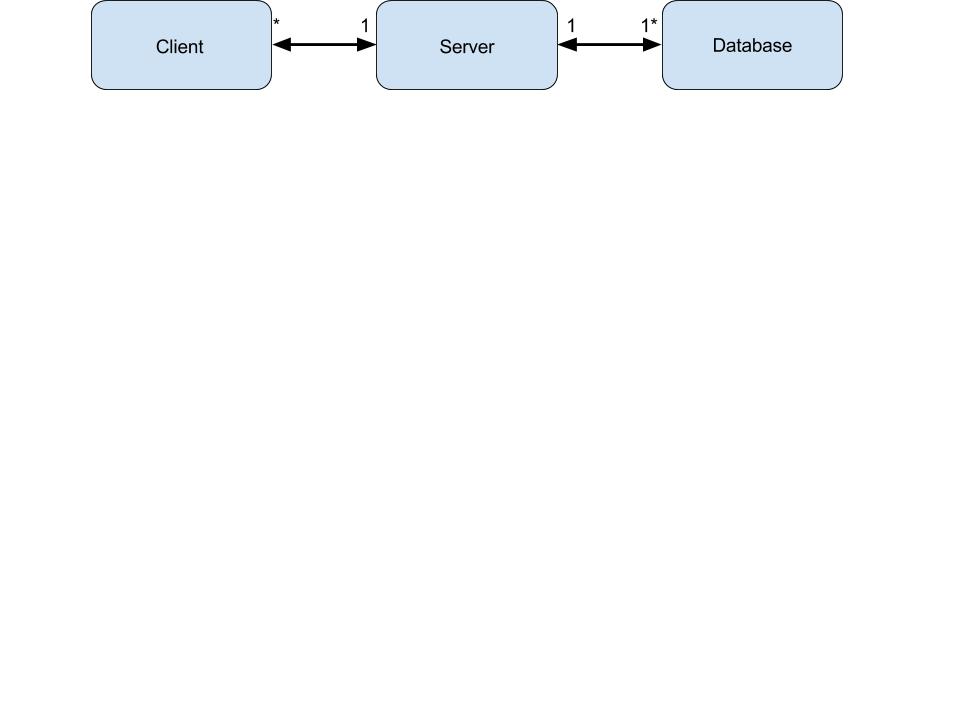
1. The user interface will be an Android interface.
2. The user interface should be clean and simple.
3. The app will pull its content from a backend server.
4. The app will display very little to no embedded advertisements.
5. The backend end server will maintain a database of communities and users.
6. The server must be able to handle a large amount of traffic.
7. A donations option to be available if more budget is required for maintenance.
8. Integrate the use of Google Maps and geolocation (if time available).

**Design Outline**

Our project, Butterfly, will utilize the Client-Server Model:

* Client
  + Provides user interface
  + Stores user’s groups
  + Stores calendar containing the events from the user’s groups
  + Syncs calendar with the server
* Server
  + Interprets requests from client and provides response from the database
  + Manages community calendars
  + Manages community events
  + Manages community boards
* Database
  + Stores all of the information like users, groups and events

The Android Client and Web Server will have a many-to-one-client-server relationship - one server for many users. The Server will respond to requests from the Clients and either store, retrieve or modify the information in the Database.



**Design Issues**

**Functional:**

How specific will a group’s categorization be?

**Option 1:** The group’s creator can write in any categorization when the group is formed

**Option 2:** The group’s creator will choose from a set of general categories (e.g. sports,

games, food, social) and write in a more specific description

**Option 3:** The group’s creator will first choose a general category, which will allow them to choose from a set of predefined subcategories (e.g. sports – soccer, baseball, Frisbee)

**Decision:** Option 3 is the best solution. Having the creator write in their own category for every group will make them hard to search for by category, and if we have pre-defined sub-categories, we may not capture all the different types of groups users will create.

Who is allowed to post events in a community?

**Option 1:** Only leaders should be able to post events

**Option 2:** Anyone in a community can post events

**Option 3:** Leaders can choose who can post events

**Decision:** Option 1 is the best solution because the leader will be able regulate the community much better than random people in the community. Also if a leader is busy or unavailable for certain periods of time (for example during spring break or over the summer) they can make other users leader. This way the community will always be active and this will also prevent spam.

How should users sign in?

**Option 1:** Email address

**Option 2:** Another social account (Google+, Facebook)

**Option 3:** All the above

**Decision:** Option 3 at first glance seems to be the clear best solution to this design issue because it’s the inclusive, flexible option. But, by having users use their account of a popular service like Google+, it removes the need for our app to handle user signup and account creation. Also, we would want to distance ourselves from other social services that are not conducive with our ideals like Facebook.

When are crews removed from the system?

**Option 1:** Crews are removed when there are no more members

**Option 2:** Crews are removed when there is only one member left

**Option 3:** Crews are removed when there is only one member left and there are

no pending requests to join the crew. When a crew is created, the creator must

send at least one invitation.

**Decision:** Option 3 allows us to remove unnecessary crews (crews with only one

member) while still allowing easy creation of crews. Option 1 could possibly leave

many unnecessary crews which would take up space in our system, and option 2 makes the creation of crews more difficult, since there is only one member to start and will thus be removed upon creation.

How are members notified of hangouts?

**Option 1:** Every member of a community is notified whenever a hangout is created.

**Option 2:** Hangouts are posted to a community board for each member to see.

**Option 3:** Hangouts with overlapping timeframes are internally matched and there is a separate option for browsing.

**Decision:** We will choose option 3. The problem with option 1 is the potential for an overwhelming amount notifications to be pushed to community members. Option 2 has a similar problem in that posting each hangout to community board could allow spam posts to take over the community. Option 3 is the best solution because it automatically combines and closes similar hangouts, while also giving members the ability to choose whether or not they want to view hangouts. This way community members are not annoyed by constant notifications, and the community is not cluttered with hangout posts.

How should users be able to invite friends to a group?

**Option 1:** Users can search for existing Butterfly users by name

**Option 2:** Users can invite people via email address

**Option 3:** Users can invite people via text message

**Option 4:** Users can create a friends list and invite users from the list

**Option 5:** Combination of searching for Butterfly users, inviting by email, and text

messages.

**Decision:** A combination of options 1 and 2 will be the best decision for this. We are avoiding option 3 giving users friends list to keep communication inside the groups. When searching for people to add to a group, a list of autocomplete suggestions will come up from people you already have interacted with or if their email or phone number is already associated with a Butterfly name.

**Non-Functional:**

What language should be used for the server?

**Option 1:** C++

**Option 2:** Java

**Option 3:** Ruby

**Option 4:** Python

**Decision:** We chose Java for our server since it a language everybody on our team has had previous experience. We did not think it would be a wise use of our time to get everybody acclimated to a new language when there was more important development work to do in the short timeframe of the semester. Additionally, since we developed an Android application, which is also written in Java, it made was convenient to choose Java for the server as well.

**Design Details**

Class Level Design

class-level-diagram.png

Description of Classes

1. Neighborhood
   1. List of all communities in the area
   2. Has a calendar for all communities in the neighborhood
2. Community
   1. A group of people with an interest that connects them
   2. Contains a list of all users that are members or leaders
   3. Has a calendar of events planned
   4. Hangouts can be created and searched within the community
   5. Crews for chatting with and notifying specific members of the community can be created and viewed
   6. Community members can send invites to other people
   7. Members can post messages to the board and comment on other users’ posts
   8. Users can view and join open communities
   9. Members of closed communities can notify all other members
   10. Members of open communities must first create a crew to send notifications
3. Crew
   1. Small clique of people from one community to chat with
   2. Users can be invited to the crew
   3. Users can create a crew and invite their friends within the community
   4. A crew is created when community members are matched for a hangout
4. User
   1. Contains all the information like email, phone, calendar, lists of communities and crews, and rating
   2. Users can either be regular members or leaders of the communities they are active in
   3. Users who are leaders can delete posts and comments from the community board
5. Calendar
   1. Displays all the events the user has RSVP to or the events in a community or neighborhood
6. Event
   1. Has all the information about the event like name, place, time and description
   2. Events can be posted by leaders of communities only
7. Board
   1. All community posts are listed here
8. Post
   1. It is a message that goes on the board which can be commented on by the users
9. Message
   1. Messages are sent to all members of a crew
10. Comment
    1. Comments are messages in response to posts sent to the board
11. Hangout
    1. Users can create a hangout for meeting up with other members of a community
    2. Hangouts have a range of start times and a range for number of members
    3. Hangouts are matched based on the start times and member range and combined
    4. When the range of members is met, the hangout is deleted and a crew with all users from the hangout is created
12. Hangout Finder
    1. Contains a list of all of the Hangouts in the current Community
    2. Has methods for matching up hangouts with overlapping times and member ranges

Possible Actions of a Community Member

user-sequence-diagram.png

Possible Actions of a Community Leader

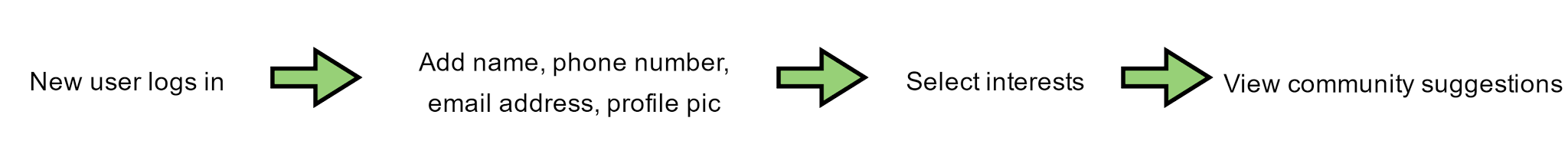
moderator-sequence-diagram.png

Create Selection Flow



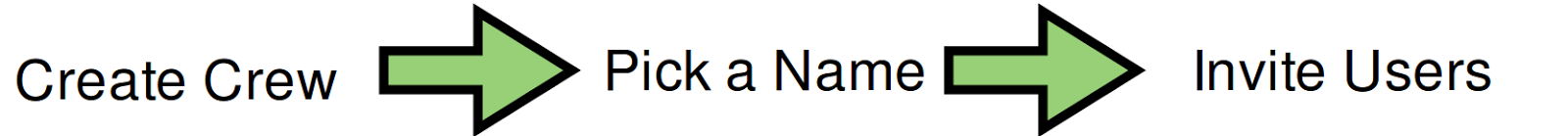
This flow shows the dialogues a user will be guided through after choosing the create a group button.

New User Selection Flow



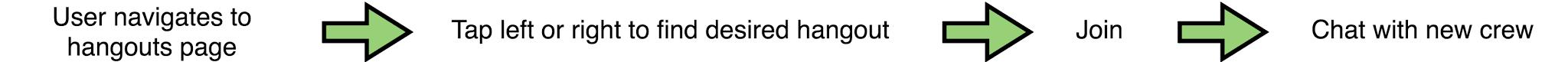
This diagram shows what happens to a new user when they log in for the first time.

Create Crew Flow



From the communities page, members can create a new crew using this process.

Join Hangout Flow



These are the steps a user will go through to join a hangout.

|  |  |
| --- | --- |
| Login Screen  UI Mockup.png  This will be the first screen a user sees when they enter the app. They will have the option to remember login information. The steps afterwards are shown in the new user flow if it is the person is creating a new account. | My Home Page  Home Page.png  This page will show all the groups a user is a part of. There will also be a button leading to different pages for notifications, discovering other groups, and viewing notifications from anything they are a part of. There is also a button to reach the settings located at the top left. |
| Community  Community Home Page.png  This is the home page for each community. At the top there is a back button which leads back to the user’s home page. Next to it the community’s name and image is displayed. There are tabs for the list of crews in that community the user is a part of, the community board, and the community calendar. At the bottom there is a button for changing the user’s community options and allowing the user to leave the community. A button leading to a list of community members and leaders is on the bottom right. On the bottom middle, there is a button which leads to the hangouts page. If a user is not yet a member of the community they are viewing, they will not be able to access the hangouts or settings, and those buttons will be replaced with a join community button. | Discover    This page shows the communities in the user’s neighborhood based on the interests of the user. There is a search button at the top right which allows the user to search for specific communities by name or by category. The number of members of the community is displayed to the right of the community name |

|  |  |
| --- | --- |
| Activity  Activity Page.png  The activity page serves as a notification page. Here is where a user will be able to see any activity going in inside a group. Clicking on the bolded words will take a user to the respective page mentioned. The calendar will also show each user's events they have planned in one synchronized calendar. | Hangouts    The hangouts page allows community members to either create a new hangout or join existing ones. They are able to join hangouts by using the browsing interface to find the hangout that best suits them. If a member would rather create a new hangout then they simple tap the create button in the lower left. |

|  |
| --- |
| The crew page allows simple communication between the members of a crew. The community name and crew name are located at the top. The button at the top left will lead the user back to the community home page. Messages sent to the crew chat will send notifications to other crew members. Notification options can be accessed with the button on the bottom left. The button on the bottom right leads to a page showing members of the crew. |