buddyUp

Team Amethyst:

https://github.com/b-cheung/EE461L-Project
Cooper Travis: Email: ctravis096@gmail.com, Github: CooperTravis
Johnny Gabriel: Email: jmgabriel@utexas.edu, Github: jmgabriel96
Brooke Paxman: Email: brooke.paxman@utexas.edu, Github: brookepaxman
Brian Cheung: Email: b_cheung@utexas.edu, Github: b-cheung
Sean Wang: Email: sean.wang@utexas.edu, Github: wang-sz
David Terral: Email: dterral504@gmail.com, Github: dterral504

Problem Statement and Vision

You want to go to an event or do an activity, but you don't have anyone to go with, and you don't want to go alone. It can be hard to meet new people or do new things, especially as an individual. buddyUp provides a good solution to this by finding a "buddy" to go to an event with or to plan an activity with. Now going to a concert, late night snack trips, or even going to the gym can be an opportunity to meet someone new rather than going solo. On the other hand, maybe it's hard to jump into a completely new hobby or activity, but a partner or mentor can help encourage you to push through any difficulties beginners may have.

Alternatives and Competitive Analysis

There seem to be several alternatives to buddyUp, but we bring something new to the table that is significantly different from what each of the other competitors offer.

One competitor with a focus on local events is Facebook Local. This provides users a way to discover new events and notify Facebook friends about these events, an excellent way to find ways to go out and have fun. But buddyUp offers more than a platform to find events near them. It gives users the ability to find a buddy for these events in case all your friends are busy so you can still go and and share the fun with someone more familiar than a complete stranger. Additionally, users can simply look for a buddy to do an activity with, without a set time or location. Maybe you want a buddy to play tennis with, but none of your friends want to. Maybe you like going on midnight snack trips and no friends are available to accompany you. It's safer and more fun to meet someone with similar interests and buddyUp provides the perfect platform for all of this.

This can also seem like just another dating app on the surface, like Tinder or Bumble, but there are significant differences in purpose and execution. In fact, Tinder's main focus is on matchmaking, while buddyUp centers around specific events or activities. Rather than trying to match with someone you'd like to date, buddyUp offers you buddies to pair up with based on a specific event or activity. For someone who may be intimidated by going alone to a big event, they can show up with someone they met before hand. Alternatively, you might want to try a new activity. Looking for a buddy with more experience and have a mentor of sorts to help you at your first time ice skating or rock climbing. Either way, buddyUp provides a way to do all this without any expectation of a romantic relationship. What you should expect is to have fun doing what you like or exploring your interests with others of similar mind.

Prelinks would be the most similar competitor for matchmaking based on events; however, we can still draw major differences. Prelinks solely focuses on matchmaking at nightlife events and purchasing tickets for those events, a majority being concerts and parties. Certainly, these two types of events are a large overlap between Prelinks and buddyUp, but buddyUp expands this idea further. Users aren't limited to just nightlife and can express a whole variety of their interests in the form of user created events and activities in general. Matching with somebody with a similar interest in an activity can even lead to setting up a larger event for others that may be interested in the same thing to attend the meet up.

Data Sources, Scraping, and Database

We plan to collect data using Facebook's Graph API to find Facebook events that student organizations may post or local activities around the area. Additionally, web scraping from do512.com, an Austin events website, provides good coverage for many events not posted on Facebook. To do this, we plan to test out Goose or Scrapy, open source projects implemented in Python, or even developing our own scraper just for do512 if the other two don't provide the best coverage.

We plan to use Amazon DocumentDB for our database backend to store general account information, such as user profiles, hobbies, interests, activites, and events (both upcoming and previous). After scraping for events, event information would also be stored in the database.

Requirements

Actor	**Goal**
UT Student	(1) Find local events/activities
	(2) Create events/activities
	(Formal 1) Find a buddy for an event/activity
	(Formal 2) Request a buddy for an event/activity
	(3) Message buddy
IM Team Leader	(4) Advertise/Fill out teams roster

1. The user wishes to find a local event/activity. He/she navigates to the explore page to find local events/activities.
2. The user wishes to create a local event/activity that others are attending. He/she opens the menu to create a new event.
3. As a UT Student, I want to be able to easily message buddies I find, so that I can take full advantage of the application and be able to meet up with the “buddies” I match with.
4. As an IM Team Leader, I want to be able to advertise and fill out my teams roster so that I can take part in the activities UT has to offer.

Formal Use Case 1
Name: Find a buddy for an event/activity
Actors: UT Student
Entry Condition: UT student is looking for a buddy to accompany him/her at a specific event
Exit Condition: UT student matches with a buddy or decides to go alone
Event Flow

Trigger: UT student wants to find a buddy to go with them to an event

1. UT student selects the event
2. System searches for other users attending the event who are also looking for a buddy
3. System presents user with search results
4. UT student browses search results for a buddy they would be comfortable with, swiping to match with another user or to pass
5. UT student messages buddy to set up plans
6. UT student confirms the match
7. System saves event and match to user profile

A4.1. There is nobody the user would be comfortable with
A4.2. User exits and decides not to match with any buddy for this event

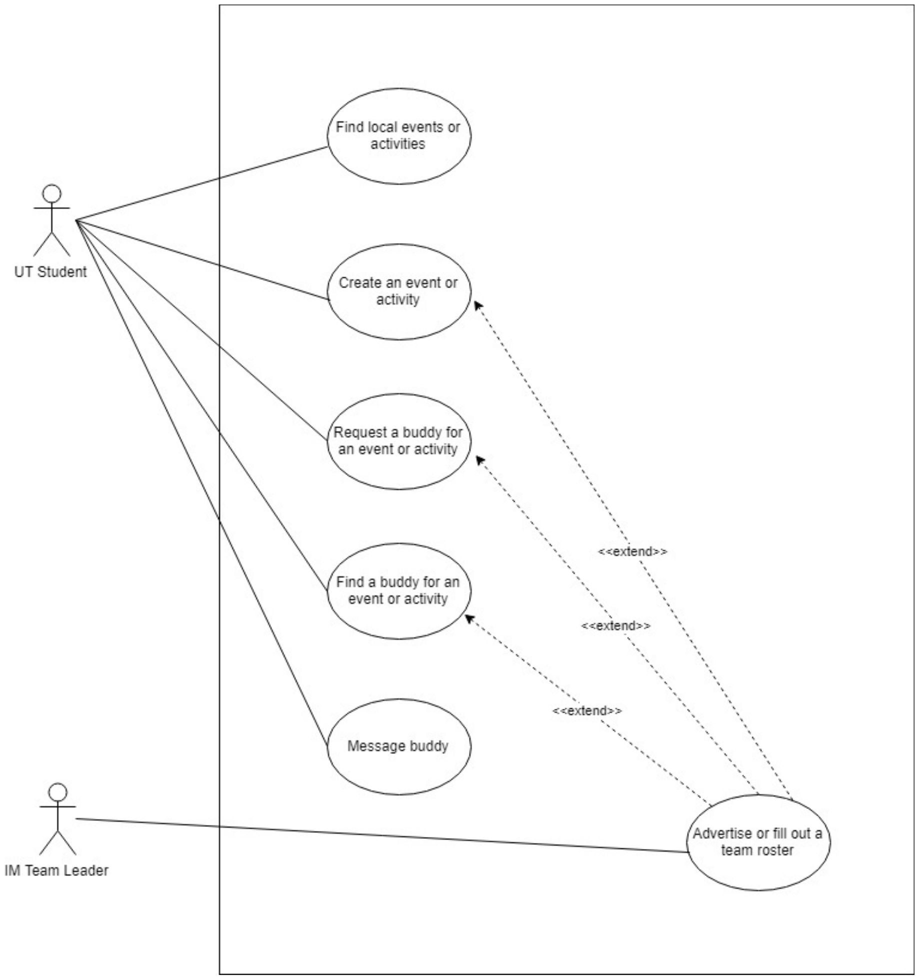
A6.1.a. User discovers that they are not comfortable with the match
A6.1.b. User rejects the match
A6.2.c. User is returned to 3

A6.2.a. Other user that is matched does not want to be matched with this user
A6.2.b. Other user rejects the match
A6.2.c. User is returned to 3

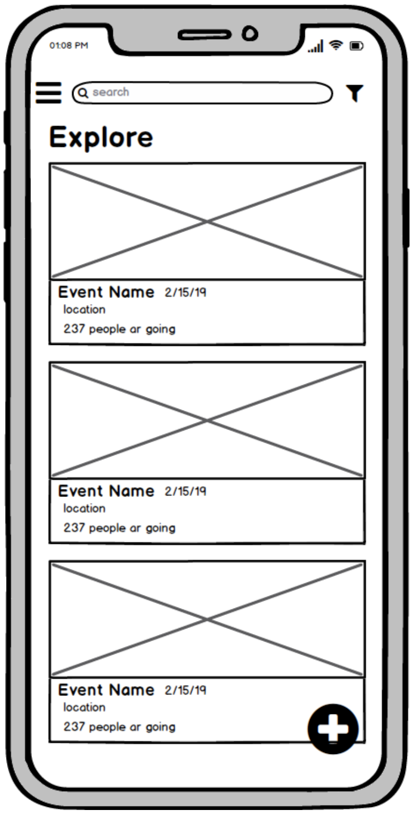
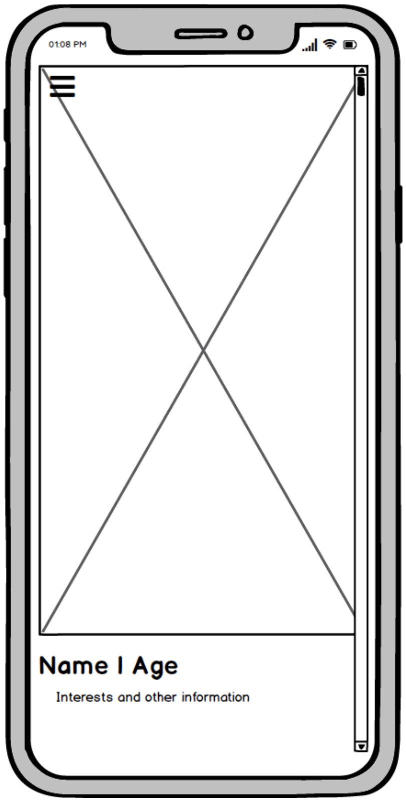
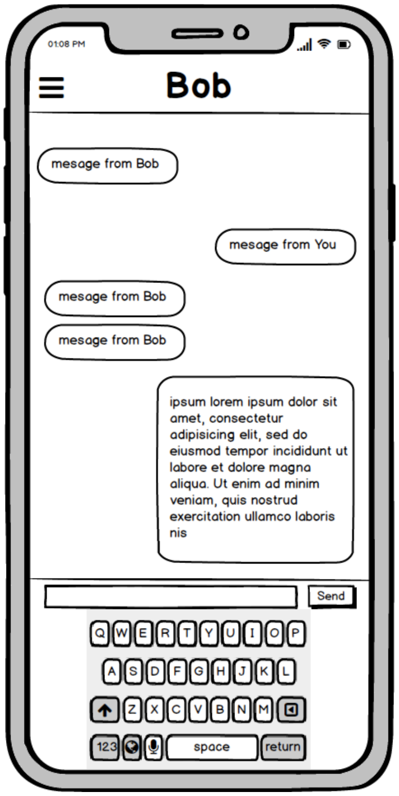
Formal Use Case 2
Name: Request a buddy for an event/activity
Actors: UT Student
Entry Condition: UT student is looking for a buddy with similar interests to do things with
Exit Condition: UT student has set up the request
Event Flow

Trigger: UT student wants to find somebody to partake in activities of shared interest

1. UT student starts the request process
2. UT student enters information about the activity or possible event
3. UT student confirms that this information will be public to others
4. System saves this to user profile


UT Student
IM Team Leader
Find local events or activities
Create an event or activity
Request a buddy for an event or activity
Find a buddy for an event or activity
Message buddy
Advertise or fill out a team roster
<<extend>>
<<extend>>
<<extend>>

Interface

Explore Page	**Find a buddy page**	**Messages**
 01:08 PM search Explore Event Name 2/15/19 location 237 people ar going Event Name 2/15/19 location 237 people ar going Event Name 2/15/19 location 237 people ar going		 01:08 PM Bob mesage from Bob mesage from You mesage from Bob mesage from Bob ipsum lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nis Send Q W E R T Y U I O P A S D F G H J K L Z X C V B N M 123 space return
The explore page allows users to find local events near them. The user can scroll through these events and tap on them for more details. The user can also create events and activities by tapping the plus button. The events will be recommended based on location, interests, other people, and filtered preferences.	The app will provide a set of potential buddies who are attending the same event and have similar interests. The user can reveal more information about the potential buddy by scrolling down. The user can swipe left if he/she does not want to be paired up with the potential buddy. The user can swipe right if he/she does want to be paired up with the potential buddy. A pair is made when both parties swipe right on each other.	After pairing up with a buddy, both parties can message each other to further the interaction.

Planning and Scheduling

Phase 1 - Database Design	● Use Amazon DocumentDB to design a database structure to store events and user profiles ● Design REST API (test using Postman) ● Begin User Interface layout design
Phase 2 - Data scraping and DB Implementation	● Use Facebook Graph API to populate events database with events specifically from UT orgs ● Web scraping of do512 to populate events database ● Carry out unit tests of code ● Refine API ● Create bare-bones, mobile application with React Native
Phase 3 - Application and User Base Connection	● Refine UI ● Allow users to see full events database and add themselves to specific events ● Implement BuddyChoice function
Phase 4 - Refine Project	● Refactor and refine code ● Final UI and functional testing ● Finish final project report ● Create presentation

Tools, Software, Frameworks

- Language/Framework: React Native
- Cloud Platform: AWS
    - Possible Databases: Amazon DocumentDB, Amazon Neptune
    - Authentication: Amazon Cognito
- Testing Tools: Postman, Mocha, unittest

Feasibility

- Needs a user base
    - We can grow our user base by providing an intuitive way for users to find local events (provided through Facebook events and do512.com). As more users use the app, the value and usability also grows.
    - Inviting friends to the platform would help.
- Developing our own web scraper in the case others don’t work
    - Real-time scraping