FishBowl - Connecting Familiar Strangers Through A Location-Based Mobile Application

David Doan, Yasyf Mohamedali, Larry Zhang
Massachusetts Institute of Technology
Cambridge, MA
{daviddoan, yasyf, larryzhang}@mit.edu

ABSTRACT
This paper details the motivation, design, and field testing of FishBowl, a mobile iOS application that initiates and encourages social media interactions with strangers. The idea for this application was developed after an initial survey regarding social media and its role in maintaining and creating relationships.

This application seeks to privately use location data to promote the initiation, growth, and prosperity of newfound relationships. By recording the people that a user passes by every day, this application permits the user to later review and reconnect with those he or she met earlier in the day. Our goal for this application is to connect users on a more personal level by providing them with a means to build lasting relationships with new people.

Author Keywords
Familiar strangers; location; proximity; Bluetooth; privacy; mobile

ACM Classification Keywords
H.3.5 Information Storage and Retrieval: Online Informational Services
H.5.2 Information Interfaces and Presentation (e.g. HCI): User Interfaces

INTRODUCTION
Since the development of the Internet, keeping in touch with people has become significantly easier, especially with the introduction of social media. Although mass social media such as Facebook and Twitter allows people to connect to each other from around the world, it is also alienating people from their surroundings [1]. In addition to this, these services are primarily focused on large, established groups rather than forming new, more intimate ones.

A paper published in 1989 by a world-renowned anthropologist and psychologist, R.I.M. Dunbar, showed a correlation between the evolution of our neocortex and the maximum number of close relationships that can be maintained [2]. This number, which is around 150, is a harsh contrast to the average amount of friends a Facebook user accumulates, which can often be in the thousands. This paper further proves the growing disconnect of emotional and physical relationships between social media and reality.

In addition to the gap between social media "friend" and physical "friends", the majority of close interactions between people tend to be location based, restricting the demographic of a person's friend group to be mainly local. Most social media sites and applications tend to cast a wide net, making connections less personal.

However, even within a localized area, there tends to be a disconnect between strangers. This phenomena was first explored by Stanley Milgram in 1977 where he outlines the concept of "familiar strangers"[3]. This term describes an individual who is visually recognized through regular, behavioral activities but does not interact any further. It is difficult for current social media applications to bridge this gap due to the privacy restrictions we, as users, implement on our social profiles.

From these, we devised our three main research questions to be as follows.

- How can we encourage more personal interactions between familiar strangers?
- How can we build enough trust and enthusiasm between users to spark a conversation?
- How can we establish relationships between people in similar geographical locations?

To answer these questions, we decided to make a mobile application that aids in connecting familiar strangers with each other. Our mobile application is meant to supplement physically meeting people by providing a history of people who the user has met over the last 24 hours. Our service provides a way to promote the longevity of meeting new people, giving users all the information needed to establish relationships with strangers. Ultimately, we hope that our application will establish lasting, personal relationships between people in similar locations.

RELATED WORKS

This application draws from work in two primary fields: familiar strangers and location. The problem of familiar strangers has been studied, but no complete solution has been fully implemented. Location-based applications are also very plentiful, but many run into the problem of privacy. We will introduce works in both fields to better establish the design and implementation of our system.

Familiar Strangers

There has been academic literature that target problem similar to familiar strangers, but none that involve a solution similar to ours. An example of this is "Making Friends by Killing Them: Using location-based urban gaming to expand personal networks"[4]. In this paper, the authors organized and facilitated a murder mystery game between mutual friends with the intent of building a relationship between them. This solution gamifies meeting new people by using a SMS game.

However, their project takes a more active approach, where users need to go out of their way to interact with the application and with others. Our application tries to be more passive by running in the background and not interfering with the daily routines of our users.

Another academic paper, "What can 'People-Nearby' applications teach us about meeting new people" outlines how recognizable profiles and establishing trust between strangers is crucial in driving social interaction between two people.

As such, we hope to incorporate their findings by utilizing well-established and recognized Facebook profiles. We also build upon other trusted social media outlets, such as Twitter and Snapchat, so that users can more comfortably view others' public image.

A natural, crowd sourced solution is the "I Saw You"-style message boards (Figure 1) where anonymous users post descriptions of people that they saw. The idea is for the described person to see the board post and reply. This, however, is completely anonymous and usually, no relationships are formed beyond the bulletin post.

Our application plans to combat this issue is by providing a transparent social media platform for users to reference after each interaction (as mentioned above). Since the user can actually see the name and Facebook profile of his/her new acquaintance, a future interaction is more likely to occur.

Currently, there is one major commercial competitor in the market called Happn (Figure 2). It has been released overseas and in New York, but has not gained much traction since its initial release. Like FishBowl, it records people who the user has interacted with and allows the user to later go through and review them. Happn, however, is geared towards dating and romantic interests. As such it limits the number and types of profiles that appear on a user's phone. Because we are hoping to target all familiar strangers, our application has a drastically different user interface, and will record all interactions.

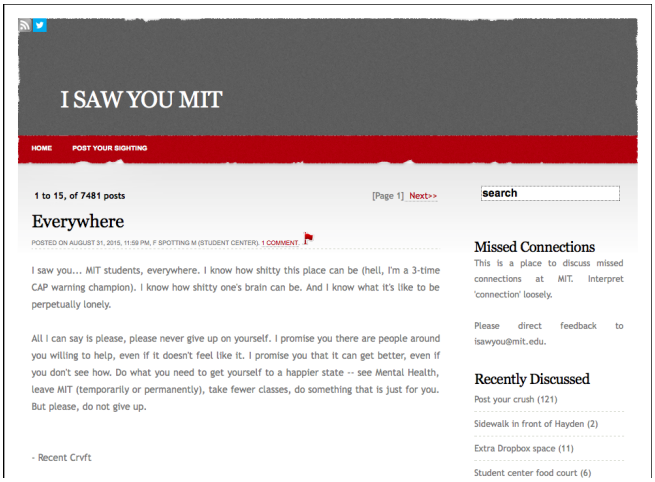

Figure 1. An example "I Saw You" blog for MIT students.

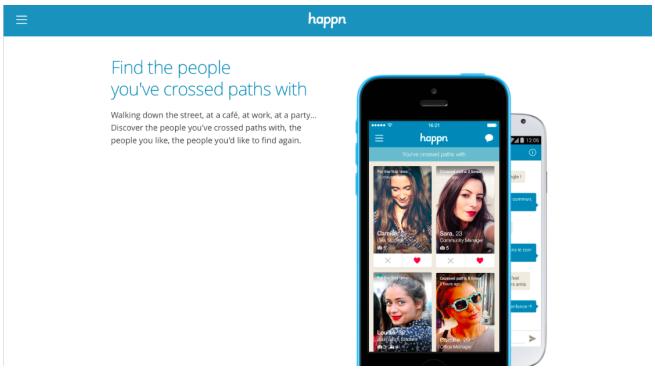

Figure 2. Happn, a location-based dating application.

Location

Currently, there are many location-based services, such as Apple's Find-My-Friends application or Facebook's Nearby Friends. While these applications already have widespread exposure, they are not commonly used.

Apple's Find-My-Friends is an iOS application that allows for the discovery of friends through location sharing. Users can easily see the exact locations of other people in their friend groups. Virtual groups can be easily created to notify and keep track of all of its members. Notifications can also be set to contact a friend or group when the user leaves or arrives a location. Although privacy settings are available, turning off or limiting location sharing defeats the purpose of this application.

Facebook's Nearby Friends is a highly underused service bundled into the iOS and Android apps, largely due to the fact that it is opt-in and not heavily publicized. When enabled, users are able to see friends in their respective cities, and receive notifications when traveling near a friend. While

reviews are positive from its users, privacy concerns and a lack of fine-tuned controls prevent widespread use.

In both cases, a big roadblock preventing the large scale use of these applications is the issue of location privacy. Most users are hesitant in publicly revealing their location, even to close family and friends. We hope to bypass this with our application by ensuring that a user's location information never leaves their device. At no point in time can users see the current location of other users.

BACKGROUND

While formulating the general idea for our application, we started by interviewing a number of students at MIT to assess their habits and preferences regarding interactions over social media. We asked questions about their personal experiences with social media, exploring the functions and limitations of social media in both large groups and in more personal, one-on-one communication. The questions covered topics from frequency of use to shortcomings of current systems. At the end, we asked the students to describe the last time they met somebody new, and how social media networks, such as Facebook, played a role. After the interviews, we performed an affinity analysis on our data, which provided us with several useful insights about social networking.

First, while current forms of social media are extremely streamlined and useful for planning events and organizing large groups of people, they are less targeted at handling more personal situations and individual relationships. Generally, people categorize social media as an outlet to post to the general public, sharing information with a broad range of people. It's extremely easy to plan an event through Facebook, or share a photo through Instagram. However, when it comes to actually building a relationship with someone, simply knowing them virtually isn't sufficient. Thus, we aim for our application to utilize social media to help in the development of these individual relationships.

Second, there was a very big disconnect between social media and interactions with strangers. Users primarily used social media as a form of communication with people they already knew (e.g. planning events with family, viewing photos/posts of friends). Very rarely was it utilized to meet or familiarize themselves with new people. And yet every single one of them had a past experience with a stranger that couldn't be built upon simply because they didn't remember or know the stranger's name. Thus, we want our application to bridge the two, allowing for users to extend their interactions with strangers and even develop them into worthwhile connections through social media.

We also wanted to explore the role of location in social media, so with the same group of students, we performed a second interview and affinity analysis. We came upon another interesting observation: people like knowing where other people are, but "don't like sharing their own location". The main issue is privacy: people like to have control over who can see their current location, and when they can see it. As described above, this was the primarily reason why apps like Apple's Find-My-Friends and Facebook's Neary Friends failed: people are uncomfortable with the publicity of their location.

It is interesting, though, because many people recognized the value of allowing others to see their location, whether it be for planning meetings, or finding people in public areas. Even so, they still valued their privacy over any of the merits of public location information.

And so, from these points, we outline the purpose of Fish Bowl: for users to keep a log of all of the users (including strangers) they meet throughout the day, and provide them with the means to connect with them through social media (namely, Facebook). While the application uses location data, no information about the user's location is shared with other users. The scope of Fish Bowl encompasses virtually any user, though we initially targeted the students within the MIT community.

These are the three major use cases we envisioned for this application.

- User A meets User B and the two interact. At a later time, User A would like to follow up with User B, but cannot remember User B's name or contact information.
- User A is in the vicinity of User B, and would like to know more about User B before approaching him or her.
- User A is in a heavily populated area, and desires to know more about the people around him or her.

While the main goal of our application is to address the issue of familiar strangers (the first use case), we realized that there are other situations in which our app could be beneficial.

SYSTEM DESCRIPTION AND IMPLEMENTATION

We developed FishBowl for iOS using the Swift programming language. We chose to develop in iOS due to its quality of developer tools, inherent aesthetics, and powerful pre-existing distribution platform.

We wanted to build an extremely intuitive, low maintenance application in order to increase adoption rate. This meant simplifying the user interface and optimizing battery efficiency.

User Interface

The main user features of our application are as follows.

- A timeline of people who the user has interacted with over the last 24 hours
- User profiles with several social media outlets: Facebook, Twitter, Snapchat, and iMessage
- Maps showing current and previous interactions between two users

As we were building our user interface, we utilized several iterations of paper prototypes (Figure 3). In order to better understand how potential users would use our app, we user tested our prototypes by asking testers to do simple tasks such as view the last interaction or add a phone number to the profile. Feedback from our paper prototyping testing were crucial to determining our final design. Simple additions such as brief instructions or sending push notifications were invaluable in keeping our application intuitive.

Figure 3. Initial paper prototypes.

Timeline

The timeline (Figure 4) is the main screen and contains a list of profile pictures and timestamps of interactions with other users. It consists of a tableview that is updated every time an interaction between two phones occurs. This allows for live updating without having the user manually refresh the timeline. We decided that, in order to avoid cluttering the timeline, we would only display interactions from the past 24 hours. It was possible for the same user to appear multiple times throughout the day, as people can run into each other more than once in a day.

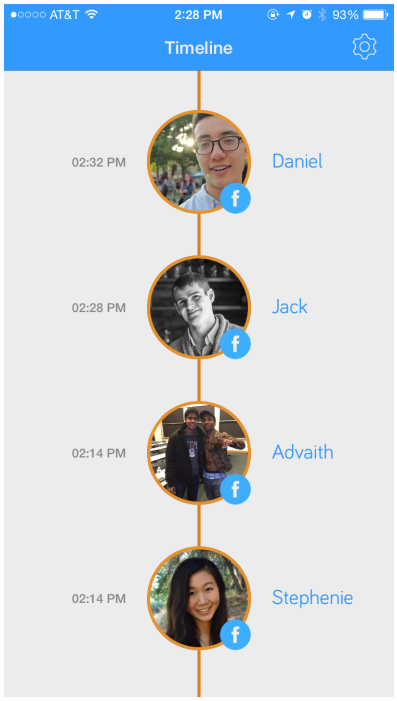

Figure 4. Screenshot of our timeline.

Later, we also added a small feature that would add a Facebook badge to the interaction if the user was already a Facebook friend. This feature was key to helping distinguish strangers from friends.

User Profiles and Interaction Map

User profiles (Figure 5) contain all pertinent information about the user and is viewable when clicked from the timeline. These profiles displays the number of Facebook mutual friends between the users as well as all social media outlets, including as Facebook, Twitter, Snapchat, and iMessage. The Facebook outlet opens the Facebook application and redirects to the user's profile. The Twitter outlet opens an in-app Twitter feed using the Twitter SDK. The Snapchat outlet copies the user's username and opens the Snapchat application. Finally, the Messaging outlet opens a screen so that iMessages can be sent to the user's phone number.

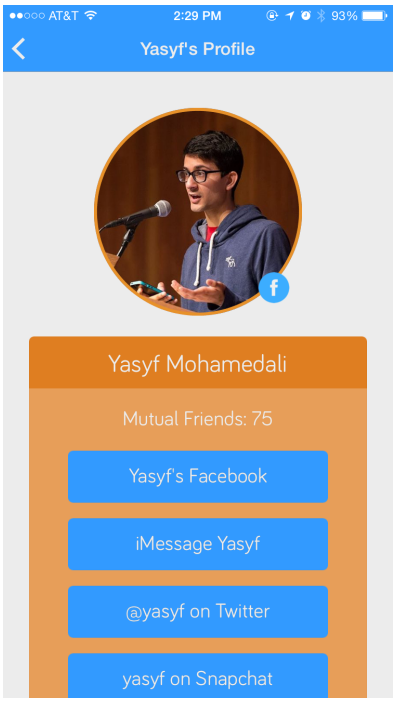

Figure 5. Screenshot of a sample user profile.

In addition to these integrations, we outfitted each profile with a map (Figure 6) showing the location of the current interaction. A button on the top left of the map gave the option to show all previous interactions between the two users.

The user's profile is also editable through the settings page. This allows users to change their phone number, Twitter handle, and Snapchat username.

User Flow

After downloading and launching the application, the main screen consists of the FishBowl logo and a Facebook login button (Figure 7). We decided to use Facebook as our main method of registration to simplify our backend, while keeping profile information consistent across all users. Logging

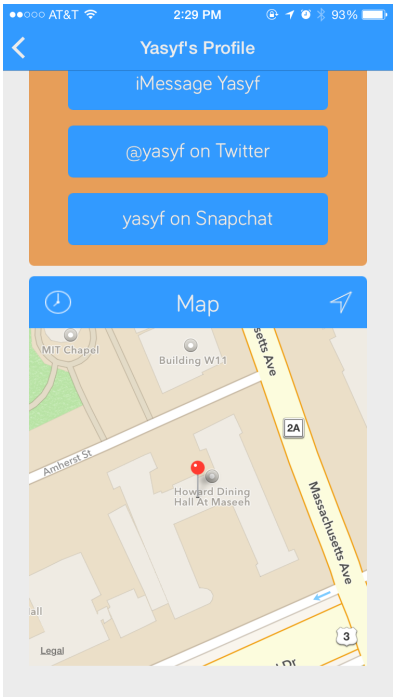

Figure 6. Screenshot of a sample interaction map.

in with Facebook is also a very efficient and streamlined process, and thus removes any unecessary hassle in the setup of FishBowl.

Figure 7. Screenshot of our initial login screen.

After authorizing our application to use the user's basic Facebook information, the application then displays a form to input any optional information regarding other social media outlets: iMessage, Twitter, and Snapchat. Then the applications prompts the user to authorize both Bluetooth and location usage (both are needed to successfully run the app).

The user simply has to leave the application running in the background. Whenever a user interacts with someone else, the two users' profiles show up on each other's timelines with a timestamp and picture. The user can then view the other's profile by simply tapping on the picture.

Notifications

Because daily users were crucial to our field study, we implemented push notifications (Figure 8) to remind the user about FishBowl. A notification would appear towards the end of the day to provide a summary of the people a user saw that day. We also implemented notifications that would alert the user whenever he or she passed by a stranger. This feature will most likely be removed in the future, when a broader variety of users are using the app. Finally, we implemented notifications to remind the user to turn on Bluetooth or keep the application running in the background.

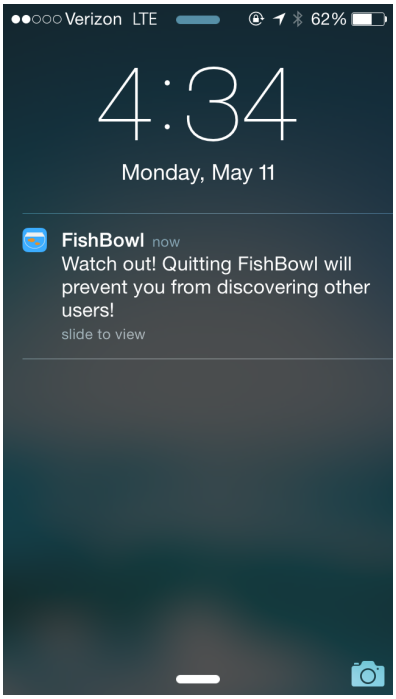

Figure 8. Screenshot of a custom notification.

Technical Implementation

Bluetooth

Because of the nature of our application, the core of FishBowl is designed around Bluetooth Low-Energy technology. Each client device acts as an iBeacon (Broadcaster) and a receiver (Discoverer). By running the application in the background, a user can go about his or her daily routine as usual without needing to actively interact with the app. Due to the hardware requirements for our implementation (Bluetooth 4.0), only iOS devices from the iPhone 4S and later are supported.

Device-to-device communication is handled by exchanging globally unique identifiers which are generated by a central server upon registration. When a user signs up and provides any additional information, all data, including Facebook tokens, is posted to this server, which returns a new user identifier that the devices stores indefinitely. Any updates made to user data are immediately posted to the central server. When a user obtains the identifier of another user, it queries the central server to retrieve all associated metadata, which is then cached locally on the device.

The central server presents a JSON API which clients consume. Since the only publically-accessible endpoints involve registration and querying for user metadata given a UUID, no authentication is required. The only other publicly-accessible endpoints interact with the Facebook Graph API, to query statistics such as number of mutual friends between two users. These endpoints require a Facebook authentication token, and simply passes requests through. To interact with this API, FishBowl uses a custom-built client based off the open-source `SwiftHTTP` library. In addition to the client-facing endpoints, there are additional endpoints used for administrative purposes, such as resetting demo devices, and sending push notifications to all beta testers.

The software architecture for communication is divided into two major components, a Broadcaster and a Discoverer, both of which take advantage of the CoreBluetooth and MultiPeerConnectivity frameworks in order to build a mesh network which devices use to communicate with each other. Through a series of asynchronous callbacks, background threads use the Broadcaster and Discoverer services to react to encountering new devices. After collecting relevant information, they commit finalized Interaction objects to a CoreData-based persistent store.

Broadcaster

The Broadcaster service exists to help user devices advertise themselves to nearby devices through a variety of mediums, including the currently connected WiFi network, ad-hoc WiFi networks, and Bluetooth. To ensure communication with a maximal number of devices, we use all available communication mediums to broadcast a device's presence. To ensure consistency of communication, a single set of identifiers are used for all FishBowl services, allowing other instances of the application to filter out broadcasts from other applications. When in the foreground, the Broadcaster advertises itself as a peer with the MultiPeerConnectivity framework, under the `personlog-disc` service identifier. The peer identifier is set to be the FishBowl ID of the broadcasting user. Initially, while running in the background, the plan was to register devices as iBeacons with unique device identifiers, but due to artificial limitations on background behavior, this was abandoned. When running in the background, the Broadcaster advertises itself as a Bluetooth Low Energy peripheral with a single service (`F8F1A882-14FF-4F5D-A4A2-0308AB0644D8`). The synthesized peripheral has one characteristic (`C7F7729A-F744-49E7-AE94-649D14FE2327`), which has a read-only value that is set to the FishBowl ID.

Discoverer

The Discoverer service serves to scan for the broadcasts of other devices, recording a local interaction when appropriate. Once again, when in the foreground, the MultiPeerConnectivity framework handles all discovery of other peers. In order to effectively discover in a quick yet power-efficient manner when backgrounded, we take advantage of operating system-level abstractions which allow our application's threads to remain asleep while in-between interactions. By registering specific services with the kernel to seek out on remote devices, FishBowl uses minimal power between discoveries, letting the built-in CoreBluetooth framework handle the hardware interactions. The application is only activated for a short period of time upon discovering a Bluetooth peripheral with the specified UUID. The FishBowl ID of the other device is recorded, and the GPS chip is powered up to full strength for up to 30 seconds (but often far less) in order to get an accurate location fix. The next time the application is foregrounded, all pending interactions are processed, with the central server being queried and all data being committed to persistent storage.

Battery Conservation

In order to ensure a bounded loss in energy from getting a highly accurate GPS fix, we use two tactics. The first is to precalculate a low-power location fix based on non-GPS heuristics every ten minutes, so as to speed up the higher-precision operations when they occur. The second is to use two concurrent queues with a lock on a shared indicator to ensure that, given sufficient time, an interaction is recorded, whether or not the hardware has reported a location report within the desired accuracy threshold. This is demonstrated in the code below.

```
class PeerTask:NSObject {
  let callback:() -> Void
  var completed = false
  let peer:Peer

  init(peer:Peer, callback:() -> Void) {
    self.callback = callback
    self.peer = peer
    super.init()
    dispatch_after(
      dispatch_time(
        DISPATCH_TIME_NOW,
        Int64(30*Double(NSEC_PER_SEC))),
        dispatch_get_global_queue(
          DISPATCH_QUEUE_PRIORITY_HIGH,
          0
        ),
        self.run
      )
  }

  func run() {
    if !completed {
      completed = true
      callback()
    }
```

```
  }
}
```

```
func didFindPeer(peer:Peer) {
  let peerTask =
        PeerTask(peer:peer, callback:{
    for callback in self.peerCallbacks {
      callback(peer)
    }
  })
  onNewLocation({
    dispatch_async(
      dispatch_get_global_queue(
        DISPATCH_QUEUE_PRIORITY_HIGH,
        0
      ),
      peerTask.run
    )
  })
  goGPSHighPower()
}
```

FIELD STUDY

Distribution

By using a combination of analytic and developer tools, we were able to distribute our alpha version to testers while collecting invaluable data. We decided to use TestFlight to distribute our application in order to patch bugs quickly without having to go through the Apple review process. A website (www.fishbowlapp.me) was created to collect names and emails to invite people to test FishBowl (Figure 9).

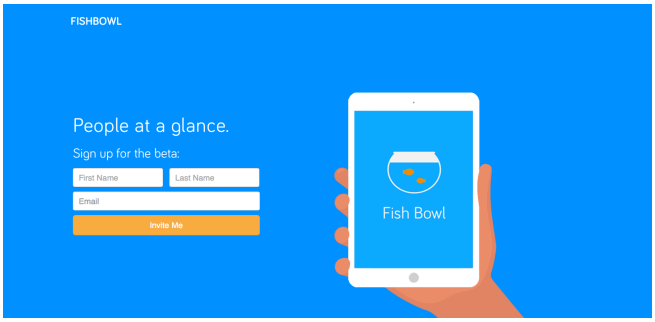

Figure 9. Screenshot of our beta testing website (www.fishbowlapp.me).

Demographics

We wanted to target several different groups of MIT students. Of the 85 users who signed up for an invitation, 70 installed and actively used FishBowl. The basic demographics our test group are as follows.

- Gender: 51% male, 49% female
- Age: 18-21 years old
- Education: Predominantly MIT undergraduates

Application Analytics

In order to be notified about bugs and crashes, we utilized Twitter's Fabric SDK with Crashlytics. This allowed us to immediately address any bugs or crashes that may potentially impair user experience. Crashlytics, paired with TestFlight, allowed us to push fixes fast and reduce any downtime our application.

User Analytics

We used Localytics, Facebook, and Fabric integrations in order to collect invaluable usage data. From this, we were able to find some interesting statistics.

Daily Users

We began distributing our first version on April 7th, 2015 and gained a steady 4-6 users a day. As shown in Figure 10, there are small dips during the weekend which can be explained by our users staying in or not interacting with large groups of people. The steep drop in our graph is most likely due to a bug that caused the application to crash after the first recorded interaction. After releasing the updated fix around April 28th, there was a steep increase in daily users which hovered at around 23 daily users.

Overall, this result indicates that the usage of our app was pretty steady. Our highest peak reached 30 users in a day with an average of 22 users/day (excluding the period with the bug). As our app functions best with more users, this was very beneficial.

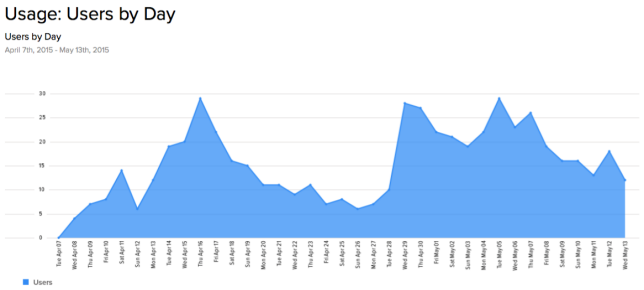

Figure 10. Daily users per day.

Occurrences by Day

Occurrences are interactions that happen between two phones. The shape of the frequency of interactions (Figure 11) correlates with the amount of users and the drop can be explained by the same bug as described above. This makes sense, as the number of interactions should increase with more users actively using the app. There is an average of approximately 50 interactions per day before and after the bug. Our user data also shows that over 80% of our users only have 1 to 2 interactions a day.

The low number of interactions per user is most likely attributed to the limited number of users we were able to reach during our testing period. We hope that the frequency of interactions per user will be resolved with more users.

Sessions by Hour

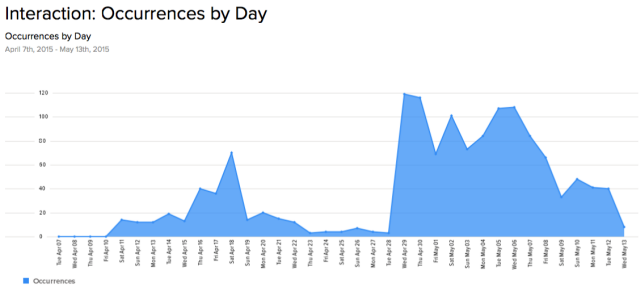

Figure 11. Interactions by day.

Sessions are instances in which the application is opened. As shown in Figure 12, there seems to be a steady rate of sessions per day when the application was fully functional. Most of the sessions were between the hours of 7P.M. to midnight. This is most likely because people will check the application at the end of the day, when they can see a summary of the interactions throughout that day.

The average session length is 6 seconds with over 70% of users closing the application without exploring further. These statistics may be attributed to our lack of diversity in testers: since many of the testers were already friends, they were not interested in exploring each other's user profiles. Hopefully, as a larger diversity of users start using the app, we will see that session lengths will get longer.

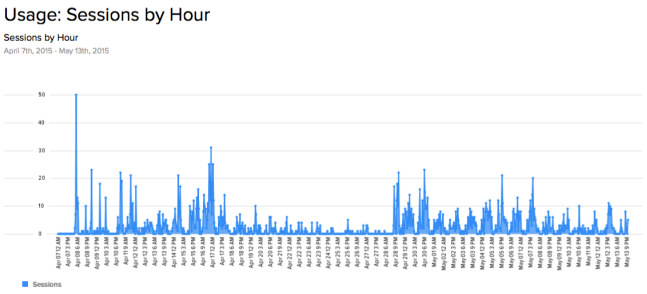

Figure 12. Sessions by hour.

Retention

Retention is the measure of returning users per day. Figure 13 shows a retention rate of approximately 40% after 1 week and 30% after 2 weeks. This data, however, is skewed because our bug largely affected the amount of returning members between the dates of April 18th - April 28th.

Even so, the implementation of our notifications system proved somewhat effective. As shown in Figure 14, the notifications encouraged the usage of the application. Out of 39 of users who launched the application through a notification, 23 users viewed a profile and 11 users viewed a social media outlet.

Meeting Strangers

Throughout the three weeks of user testing, there were a total of 53 interactions with users that did not know each other (not Facebook friends). Within this subset of interactions, 44%,

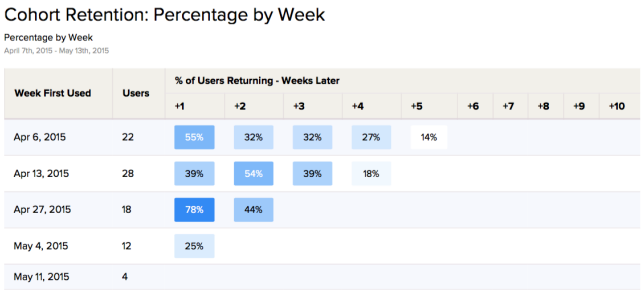

Figure 13. Retention of users by week.

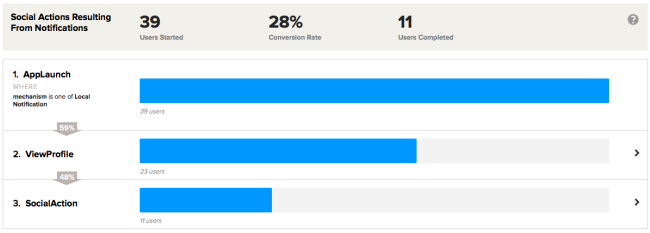

Figure 14. Notifications leading to social media ouletsx1.

or 23 users, then viewed the FishBowl profile of the stranger. Of those users, 57% of them proceeded to interact through a social media outlet (Facebook, Twitter, etc.)(Figure 15).

These users represent strangers that then went on and somehow connected through some form of social network or communication. Ultimately, this statistic represents the main purpose of our application: to initiate the communication between strangers. Unfortunately, the number of strangers that did end up interacting through social media was small. However, given the number of users that were already friends, this was relatively expected. Furthermore, we don't expect every single stranger to reach out to each other, as that would be an absurdly large number of social media interactions. However, the fact that at least a few strangers interacted indicates that our application was at least partially successful in bringing strangers together.

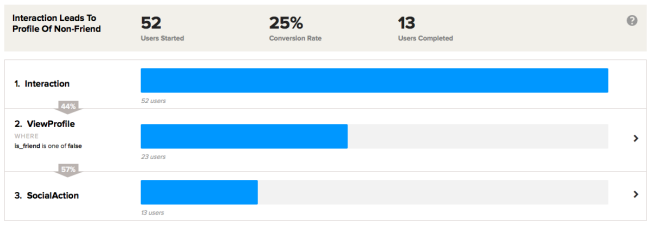

Figure 15. Conversion of meeting strangers to viewing profile.

Qualitative Analysis

In addition to quantitative data, we also conducted an exit survey of all of our testers. We asked the users to reflect on their experience. From this, we were able to see how each user used the application and their general thoughts and feedback. The results of this survey are explored further in the Discussion section.

DISCUSSION

In general, we received relatively positive feedback from our users. The following are some quotes from our users.

- "I was actually quite surprised by the number of times I run into people without knowing it."
- "It was extremely intuitive to use; I liked how simple the design was."
- "Most of the people on my timeline were already my friends."
- "When I saw someone, but didn't know he was around, I went to look for him."
- "I think the concept of the app is a really interesting idea and the design/layout of it is great, but I can't really see a practical use of it in my everyday life."
- "I really like the idea. I'd like to see the developed further over the summer!"

Location

In regards to location, we did not get any complaints regarding the use of location information. People were satisfied with the level of privacy our app provided. In fact, they found the map illustrating the locations of interactions extremely useful and interesting. "It was cool knowing the patterns in where I would meet certain people."

As indicated by these positive responses towards the interaction maps, its fair to conclude that location information definitely adds a lot of value to our application. These data not only contribute to a better understanding of the interaction itself, but provide a lot of insight into the patterns of interactions (e.g. if two people are in the same class together, they will often have interactions in the same room).

Interacting with Other Users

Although our application was able to connect strangers to each other, there wasn't a way for the users to physically follow through and form a connection with each other. "I would see someone I wouldn't know and check out their profile but wouldn't want to add them on Facebook or iMessage them." This sentiment was shared among many users. In addition, even though we tried to achieve a diverse testing group, most of our testers were already friends, which defeated the purpose of our application for many users.

Another feature that turned out to be extremely helpful was our implementation of the notification system. Because we intended our application to run in the background, a lot of users forgot about it, and would not reopen the app. However, with our daily digest notification, users often would browse through their timeline and see who they met throughout the day. One user stated that he "never used the app prior to the notifications update", but then "opened it almost every day afterward." As we can see in the spike of users on April 28, our notification system helped improve our user retention and overall usage. Although this was never part of our original research questions, it proved very effective in keeping users active.

Other Effects

One interesting user case that we did not predict was using the application for finding nearby friends. One user "went to look for" a friend that showed up on the timeline because the user didn't "but didn't know he was around." This effect is particularly interesting, as it applies to the situation addressed by Facebook's Nearby Friends. A user is notified when a friend is closeby, and is shown where the interaction occurs. Thus, he or she can go look for and physically connect with that friend.

Another very unexpected result was the gamification of "collecting interactions." This may have been a side effect of having the majority of our users were already friends. Two users commented that they would run around campus trying to find people simply because they wanted to have longer timelines. In the future, it will be possible for us to build off of this to add another layer of incentive for users to meet strangers.

CONCLUSIONS

In sum, it is reasonable to say that our application's attempt at solving the familiar stranger's problem was partially successful. It was effective in initiating the first step of virtual communication (through social media sites like Facebook), but it failed at bridging the gap into the real world. Users who found strangers through the application still did not actively try to meet up with the strangers that they met.

However, it is important to realize that even this first step of virtual communication can play a key role in building these novel relationships. As brought up in our first round of interviews, when people met strangers, it was often very difficult to follow up afterwards. While our app may not fully initiate the personal and lasting relationship we were originally intending, it still solves the problem of "not knowing [the stranger's] contact information." We hope that with further development and implementation, we can overcome this gap and play a role in fulling initiating personal, as opposed to just virtual, relationships.

Future Direction

Looking forward, there are two major areas for improvement that we gathered from our user feedback.

The first is an overhaul of the user interface, focused on increased functionality and ability to handle large numbers of interactions. Many user feature requests involved better filtering capabilities, such as only showing people who are not current Facebook friends, or people who go to the same university. People also often requested the ability to customize how far back their timeline went, so that users had a possibility of viewing a history of *all* interactions. We also hope to incorporate advanced analytics and statistics about historic interactions, such as "total number of people seen today" and "most frequently interacted." Furthermore, the timeline for displaying recent interactions does not present a very good user experience when the number of items increases throughout the day. To combat this, a future enhancement involves "clustering" interactions by time and location, such that the

user experiences nested timelines that they can navigate more easily.

The second major area for improvement is in the application backend. There are many areas of the communication infrastructure which are not scalable or efficient, and we plan to dedicate time to improve the robustness of our systems.

At this point in time, we still have approximately 30 crash-inducing bugs that are being reported from user devices through Crashlytics which remain to be triaged and rectified.

We hope to continue our work on FishBowl, and eventually plan on submitting our application to the App Store for review and distribution to the general public.

ACKNOWLEDGMENTS

We would like to thank all of our testers for giving us feedback and helping us improve our application.

We would also like to thank Edward Barrett and Frank Bentley for guiding us through the process of making our app. Your experience and insight was extremely valuable, and your suggestions were very much appreciated.

REFERENCES

1. Toch Eran, and Inbal Levi. What Can People-Nearby Applications Teach Us about Meeting New People? (2012): 1. Web.
2. Dunbar, R.I.M. Neocortex Size as a Constraint on Group Size in Primates. Journal of Human Evolution 22.6, (1992): 469-93. Web.
3. Milgram, Stanley. The Familiar Stranger: An Aspect of Urban Anonymity (1977): 51- 53. Addison-Wesley.
4. Coe, Josh, and Monchu Chen. Making Friends by Killing Them: Using location-based urban gaming to expand personal networks. (2010): 1-4. Web.