Project Leader:
James Almeida - jamespa@princeton.edu

Team:
Sumer Parikh - sumerp@princeton.edu
Danielle Pintz - dpintz@princeton.edu
Emanuel Castaneda - emanuelc@princeton.edu
Meaghan O'Neill - mconeill@princeton.edu

OVERVIEW

Check out our eXchange Prototype!:
https://jamesalmeida.proto.io/share/?id=21c47824-e34e-460b-82ea-9b173da86e97&v=6

"eXchange" is an app designed to completely revolutionize the eating club meal exchange process. It will give club members the ability to schedule and plan out meal exchanges with one another while giving the meal checkers of club the convenience of a completely electronic system. We hope that it will make meal exchanging a more social, convenient and public activity for students.

We will have two types of users: club members and meal checkers, each of which will have their own, separate user interface and verification process. Before getting started with specifics about the app, we should make one distinction very clear: there are two major moving parts in our app. The first part is the ability for users to coordinate meals and keep track of requests that they have. We hope that this will make meal exchanging significantly more convenient and streamlined for friends in different clubs. The second part is the ability for meal checkers to determine whether or not an exchange that was set up by members was actually completed or not. This is really the most important part of the app that we need to coordinate with the ICC and the individual clubs. Only after getting the meal checker's confirmation does an exchange actually get processed officially.

This app will be made on two platforms that share a back end. It will be available as an iOS app for the convenience of iPhone/iPad users, and will also be available as a webapp so that every member of an eating club can meal exchange on this system whether or not they own a smartphone (Yes, we do not consider androids to be "smart" phones).

REQUIREMENTS AND TARGET AUDIENCE

Currently, in order to meal exchange, members of a club must bring their exchange to the meal checker and fill out a paper slip by hand. These slips are then tucked away into a box or drawer and are susceptible to getting lost, misplaced, or manipulated, which can cause some members to be unjustly charged for an apparently incomplete exchange and can allow other members to falsely complete an exchange. Additionally, for members that meal exchange frequently, there is no system in place that tracks the exchanges they have started and completed in a given month. Because of this, it becomes difficult to remember all of one's previous exchanges and it is common for friends to simply forget to complete them. "eXchange" has been designed to solve all of these problems in addition to making meal exchanging a more social and interactive process. Therefore the target audience of the app, as alluded to in the overview, will be members of eating clubs and the officers of the eating clubs.

FUNCTIONALITY

Note: Many of the functionality decisions that we are working with now can be interactively observed on the proto.io link that has been added at the top of the design doc at the beginning of overview.

Scenario A: General Overview

Let's say Sally wants to meal exchange with her friend Diddy. Sally is in Cottage, and Diddy is in Cap & Gown. Sally wants to invite Diddy to eat dinner with her tomorrow! She opens eXchange, goes to her "friend" list (a person's eXchange friends consist of the the intersection of three groups of people: their facebook friends, their Princeton classmates, and people with eXchange accounts). When the friend list opens, she clicks on Diddy's name (lucky for her, Diddy is one of her best friends, so she's at the top of her friend list!) and clicks "request a meal." She specifies a meal and a date, then submits her request. When the request is sent, Diddy gets a notification on her phone, "Sally invited you to dinner at Cottage on March 23rd! Click to accept or reject the invitation." Diddy clicks on the notification, which opens the "Requests" page of the eXchange app, and she clicks on Sally's request and clicks "Accept." This brings her to her eXchange calendar, where all of her planned meal exchanges for the month are shown. Home meals (those meals taking place at her club, Cap) are in blue. Away meals (those meals taking place her friends' clubs) are in white. Diddy can export her eXchange calendar to her iCal or Google Cal at any time. When Diddy gets to cottage at dinnertime on March 23rd, the meal checker at Cottage sees Diddy and Sally's exchange on the list of exchanges happening at Cottage during that meal. The meal checker clicks on their exchange, and hits "Confirm" in order to verify that Diddy indeed came to Cottage to eat with Sally. They have a great dinner! One week before the end of the month (so in Diddy and Sally's case, the day after their dinner, March 24th), Sally and Diddy get a notification on their phone because they haven't completed their meal exchange. For Sally, it says: "Don't forget to eat dinner with Diddy at Cap before the end of March!" And for Diddy: "Don't forget to invite Sally to dinner before the end of March!"

Scenario B: Rescheduling a meal

Sally invites Diddy for a meal. Diddy clicks on her "pending" tab in the "exchange" section of the app, and sees that Sally wants to eat with her on Wednesday night. However, Diddy has precept on Wednesday night so cannot eat with Sally that night. Diddy does not want to reject the meal exchange (though she has the option to do that). Instead, she clicks on the meal exchange offer and hits the reschedule button. Once she does that, she selects a day and time and the exchange is sent to Sally, where she will find it in her "pending tab" to review.

Scenario C: Gossip Sally

Sally really enjoys knowing who is eating with whom in order to be in the know about any potential relationships/unlikely friendships that she can gossip about. She can go to the "Newsfeed" tab, which makes the app more social and interactive. When an exchange is proposed, a user will have the option to make it public or private. If it is public, than Sally can indulge in her gossiping addiction by seeing every public meal had by one of her facebook friends or every public meal exchange completed in her club Cottage. Luckily, if people know

about Sally's intrusive personality and do not want to feed her habit, they can simply make their meal private and Sally will have no idea it happened (Unless she physically sees them eating!).

Scenario D: Sally commits identity theft

Sally is in Cottage (as we have established by now). However, she really wants the world to think that she is in Colonial. When setting up her profile in the "me" tab (when she first gets the app), she selects her club as colonial. However, since we have a list of the netid's of all the people in Colonial, we know that Sally is not in Colonial and will prevent her from making that her club. She will then hopefully select Cottage, where she will pass the verification, and be able to begin using the app!

Scenario E: Sally becomes a meal checker

Since people are making their meals private and Sally cannot see them on the "newsfeed" tab, Sally becomes a meal checker so that she can physically see them walk in. She logs into Cottage's account on the app by clicking "meal checker login" on the home page of the app and providing the login details. She see's a list of people who are scheduled to eat at Cottage for that meal (member's name and guest's name). When people come, she can click on them on the list and hit "confirm". If they do not come, she can hit "deny" after the meal. If she clicks on an item on the list by mistake, she can hit "cancel" and nothing happens.

Scenario F: Paying your dues

At the end of the month, the treasurer of each club is sent a list of names of the member who have started a meal exchange at their club but not finished it (and the number of times the person does that). The treasurer can charge the person the appropriate amount.

DESIGN

iOS User Interface: As mentioned earlier our user interface will consist of both a web app version and an iOS version. Our iOS version will be built through Xcode and will be written in Swift. The target application will be universal, offering compatibility with both iPhone and iPad.

Web app front end: For the web application, the pages will be built from HTML (generated through Django's template system). The style will be defined with CSS sheets generated using Bootstrap, a open source style platform created by Twitter. Javascript will be the final component. (This is a rough framework we are still figuring out specifics as we have no prior experience in web development).

Process: The user interfaces will allow users to schedule meal exchanges, allow meal checkers to approve completed exchanges, and allow administrative accounts to edit the database.

These three processes will allow the removal of the current paper system for meal exchanges. This will result in a cleaner more reliable system that will prevent data loss (missing exchange slips) and help promote finishing exchanges to prevent unwanted student charges.

Database: Our database will be run using Django as our back end service. Our persistent data will consist of user accounts, administrative accounts, and meal exchange history plus pending exchanges. Our user accounts will wither consist of a two step verification process, described in more detail in earlier sections, or will be streamlined with the Princeton CAS verification system. All verified user accounts will be stored in our database. Administrative accounts will range from Princeton faculty, to Club officers, and meal checkers. Administrative accounts will hold varying levels of control over data maintenance. Faculty accounts will hold the power to add and remove student user accounts. Officer accounts will have the ability to add and remove verified users from their designated club. Finally, meal checker accounts will only hold the power to verify or cancel scheduled meal exchanges. Our app will provide the user interface to manipulate all levels of data, once the proper credentials have been provided. It will be hosted on either Amazon, Digital Ocean or Heroku.

TIMELINE

NOTE: We have seriously front-loaded this timeline and made it very optimistic. Each date in the timeline represents what we want complete by that date (except ‘over spring break’)

Over spring break:

- Danielle, James, Meaghan will begin to learn iOS
- Sumer will learn html, javascript and css
- Emmanuel will begin creating initial UI framework

March 21:

- Have the website up and running
- Finish our preliminary elevator speech
- Fully flesh out the desired functionality in proto.io
- Meet with the ICC meal exchange task force to finalize all of our functionality

March 23:

- iOS - Completely finish the exchange screen - allow users to request one another for an exchange
- Webapp - begin work on it

March 25:

- iOS - Completely finish the newsfeed screen - allow users to see who has been meal exchanging and like or comment on these things
- webapp - finishes exchange screen

March 27:

- iOS - Completely finish the “Me” screen (picture, name, club)
- iOS - Completely finish the meal checker screens
- Webapp - finish newsfeed.

March 28:

- iOS - Prototype complete
- iOS - Basic UI segues, using mock data

April 1:
- iOs app complete and debugged with hardcoded data
- Split into two teams: backend and web app
- “Me” screen finished on webapp

April 3:
- Parse and store all of the eating club data (member names, years, netids, etc.)
- Begin implementing web app with similar functionality to iOS app

April 8th:
- Integrate all of the data into the app and fully replace all hard coded data
- Have the basic functionality of the web app implemented

April 11th:
- Create user authentication-system for both apps

April 14th:
- Time permitting, add extra functionality (ex. Email alerts for members who need to complete meal exchanges in case they don’t check the app regularly towards the end of the month)

April 18:
- Alpha complete
- bug -fixing

April 25:
- Beta complete
- bug-fixing

Final Demos and Deans Date:
- Bug fixing complete: project in perfect running condition

RISK FACTORS AND MITIGATION:

We believe that these are the big risks that we will face that could cause delays or may make things go wrong:

Risk: Currently four members out of five in our team have no experience in iOS development apart from the demos in class.
Mitigation: The members who will be working on the iOS app who will learn to use swift and the basics of iOS development

Risk: We are working with one back end and trying to interface it with two interfaces (iOS and webapp). We may end up with two incomplete apps instead of one complete one or may not be able to make both work together properly
Mitigation: If need be, we will scrap one of the apps to make sure that at least one of the systems has complete functionality.

Risk: While we have positive responses from eating club hierarchies, we have not formally talked to them and received their approval as to whether this app will be taken up.
Mitigation: We are meeting with the ICC(Inter Club Council) at their meeting on the 21st of March in order to set things up. We are also in contact with Spencer Jones, the person who coordinates meal exchanges at the moment.

Risk: We have to understand the nuances of all the eating clubs meal policies early. For example, we know that tower limits the number of non-members who can eat there every day. This would be a complication that we must account for so that extra people do not show up to tower and get turned away. Charter has every member meal check so we would have to nuance the system so that for charter, a member can only validate meals when they are meal checking and not at any other time. Also, some clubs have members nights when only members can eat at that club (ex. Cap on thursdays).
Mitigation: We must learn all these nuances early and account for them. Luckily, we have members of 3 different clubs on the team, so we have a full grasp on the policies of those clubs. We must talk to members (preferably officers) of the other clubs to fully understand their procedures and whether we have to do something extra to account for them.

Risk: Chicken and egg/network effect when attracting users
Mitigation: We will either make it the official platform for meal exchanging (if the ICC approves it). Otherwise, if it does not become the official platform, we will start using the member interface to plan meals with our friends and popularize it in our clubs (listserv, work of mouth etc.) so that at least the meal set up part of the functionality is used.

Risk: Data of the members of eating clubs may not be provided by the clubs
Mitigation: If it is the official meal exchange platform, then the clubs will have to provide us with that data. If it isn't, then clubs which don't provide us with member lists will not be included in the system.