GrubSplit

Teamwork Plan

By: Amanda Zhou, Marcos Pertierra, Matthew Miklasevich, Jorrie Brettin

Stakeholders

1. Users
   The users will use GrubSplit to order food from a specific restaurant and easily split the bill (including tax, tip, and delivery fee).
2. Food Providers
   These restaurants are the ones that provide the food that the users ordered.
3. Delivery Provider (Delivery.com)
   This is the service through which the order is placed - they contact the restaurant with the order and handle delivery.

Tasks

1. Design Doc/Teamwork Plan (Due 11/13)
   a. Jorrie - Motivation and Concepts
   b. Matt - Data Model and Security Concerns
   c. Amanda - Wireframing
   d. Marcos - Data Model and Teamwork Plan (tasks and MVP)
2. Revised Design (Due 11/19)
   Each team member will revise his/her sections from the Design Doc. Expected Effort: 1-2 days per person
3. MVP implementation (Due 11/23)
   a. Jorrie - Views and client-side JS. Expected Effort: 3-5 days
   b. Matt - Integrating with 3rd party APIs (Delivery.com, Venmo). Expected Effort: 3-5 days
   c. Amanda - Routing, User model and testing. Expected Effort: 3-5 days
   d. Marcos - Grub model and testing. Expected Effort: 3-5 days
4. Final version of code (Due 12/6)
   a. Jorrie - Updating Grub UI to include ratings, blacklist, map (delivery tracking/time-estimate). Expected Effort: 4-6 days
   b. Matt - Delivery tracking feature. Expected Effort: 4-6 days
   c. Amanda - Notifications feature. Expected Effort: 4-6 days
   d. Marcos - Blacklisting/Blocking user feature. Expected Effort: 4-6 days

Risks

One risk is that people do not pay the order creator. We aim to mitigate this risk by enabling the order creator to easily send notifications to the users that haven't completed the charge. Additionally, in the case where a participant refuses to pay, the order creator can just keep the food.

Naturally, with an order creator, there is a risk of the order creator receiving payments for the food from others, but keeping all the food to themselves. We mitigate this risk basing our app on trust - groups are not meant to be anonymous. In the general use case, users will be

ordering with their peers. In this case, it seems unlikely that an order creator would siphon money from their friends and keep all the food for themselves.

Finally, our reliance on 3rd party APIs for menu availability and food delivery. We chose Delivery.com because it had a reliable API and the services that we wanted. But, if we were to lose the functionality that we desire, we could switch to another service (Postmates or Grubhub, given appropriate API access).

Minimum Viable Product

GrubSplit's MVP will allow a user to:

- sign up and log in
- search for restaurants by type of food and by keywords
- create a new order (or join another order if given invite)
- invite friends to join his/her order
- complete an order which will use GrubHub/Delivery.com and Venmo APIs to automatically
    1) order the food from the restaurant
    2) request payments from other users in the order for their food items
- keep track of who has paid

The concepts that will be included are Grub (the shared online order), SubGrub (a set of food items added to the Grub), GrubLeader (the user that creates the Grub), and GrubJoiner (a user that is invited to and joins a Grub).

We have postponed implementation revolving risk mitigation in our app. This includes the ability to blacklist or block users who have not paid for past purchases. Additionally, our MVP will not include the delivery tracking feature and any corresponding UI elements.