PROJECT 3.1 DESIGN

Angel Batista, Christian Londono, Lucy Yu

OVERVIEW

Purpose and Goals (Angel)

- **Description of system to be built**

  We will build an app that aids groups of people in splitting their bills for an “events” like going to a restaurant or a taking a cab. The app will calculate debts owed between users and keep track of payments made.

- **Key goals and purpose (what problem does it solve?)**

  It is an extremely difficult task to split a bill evenly or fairly among a group of people. Even after doing so, it is even harder to track how much money one owes another person or vice versa over time. In attempting to do so, one has to make the following considerations:

  - What previous debt has accumulated between a person another over previous events?
  - When bills are split between different groups of people, how does this affect previous debt arrangements made with mutual individuals between the different groups?
  - How exactly do you split the bill when certain members of the group are expected to pay more than others?
  - When different forms of payments are used or accepted (such as cash or credit cards), how exactly do you split the bill especially when some users might end up paying more?

  The main goal of this app is to abstract all of these concerns. More specifically, the app aims to

  - Remove the need for a person to remember the debt that accumulates over time between himself/herself and all other persons.
  - Remove the need to calculate a proper splitting of a bill between multiple people.

- **Motivation for development (eg, deficiencies of existing solutions)**

  Some existing solutions to this problem include the following:

  1. “Splitwise” (http://splitwise.com/)

     Splitwise focuses organizing one’s expenses - whether its an apartment, a trip, or simply any bill. However, they have an extensive set up (i.e. you’re instantly prompted to input information about your apartment) and they attempt to determine the “fairness” of splitting the bill for you.

  2. “Billr” (http://billr.me/)

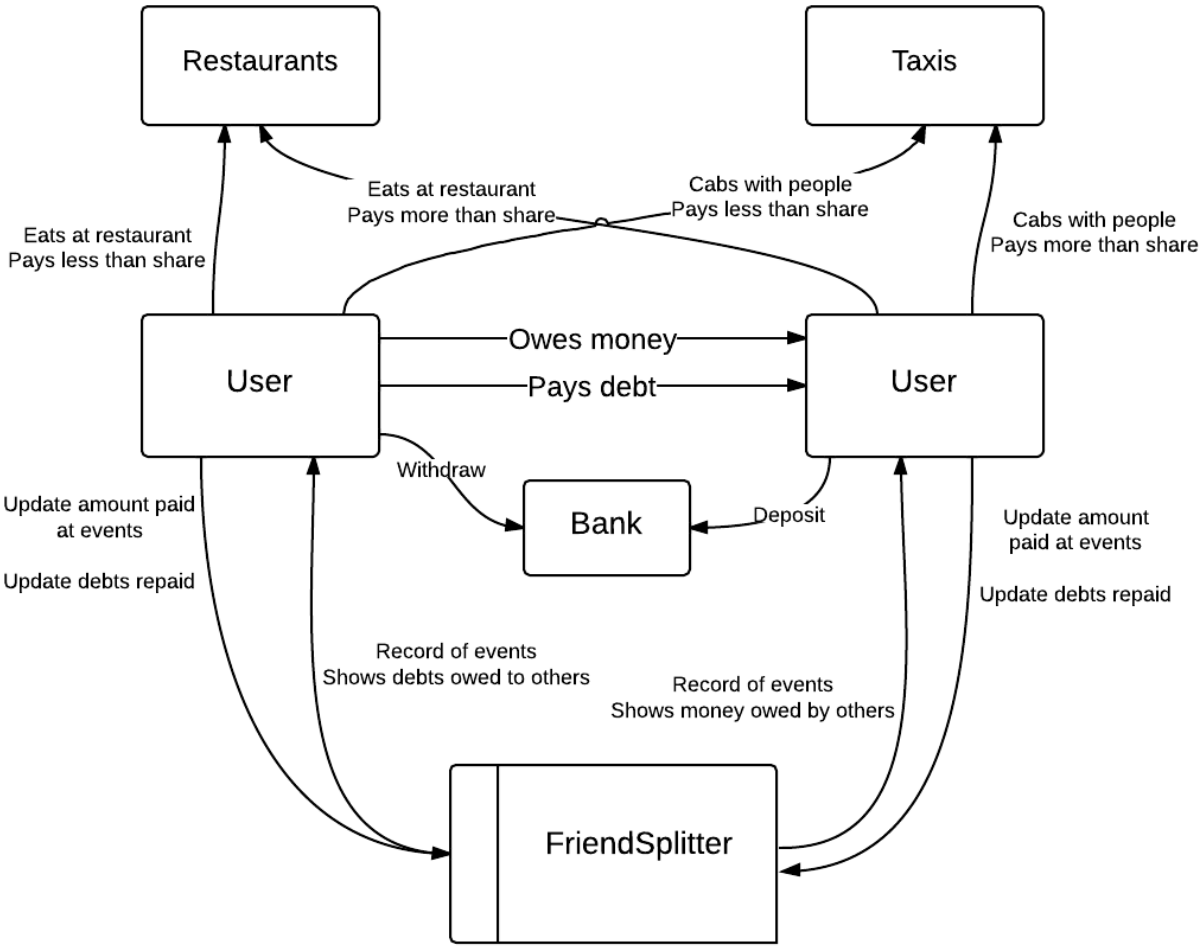
     Billr is essentially a calculator meant for splitting the bills at a restaurant or bar. However, that’s all it is. It doesn’t not track debt over time and it only applies to splitting a bill at restaurants and bars.

  3. Notes, stickies, excel sheets.

     One of the simplest approaches to this problem is to simply write down and calculate exactly how much someone owes you. However, if you’re constantly writing this information down on random pieces of paper or in an excel file that is located somewhere on your computer, you can easily lose track of certain transactions or make simple calculation mistakes.

  All of these solutions fail to deliver a truly simple way of splitting any cost. Splitter’s goal is to be simple, quick, friendly, and universal. It will be applicable to any situation where a bill needs to be split, while allowing users to determine exactly how much they need to pay for a given bill (either through an even split of the bill or through a previously agreed upon amount).

Context Diagram (Lucy)



CONCEPTS

Key Concepts (Christian)

- **Account:**

  The concept of an account is just a way to keep track of different users. It allows the user the ability to edit personal information entered and keep its information private and secure if necessary.
- **Bill:**

  A bill is considered any transaction that describes the cost of an event.
- **Debt:**

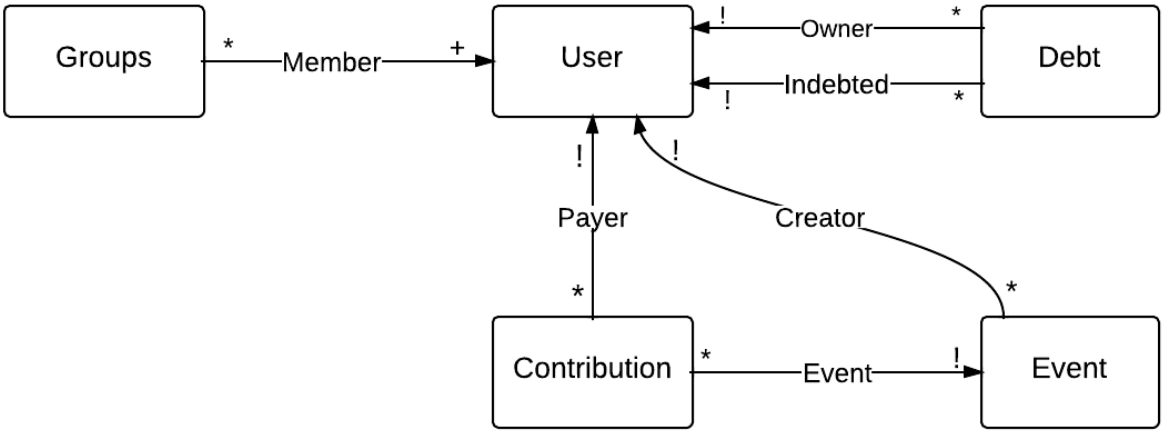
  The notion of money being owed to another person
- **Payment:**

  A payment is the concept of returning money back to someone who had lent it to you.
- **Groups:**

The notion of a selected group of users. This is generally a group that often has many interactions, whether they be debts or payments between each other. and frequently participate in events with each other.

- **Form:**
  The concept of a form is used when a user signs up, or when filling out various event details or transactions
- **Record**
  With respect to our application a record would keep track of all transactions, Debts or payments, between any two uses, as well as an overall track of total debt.

Data Model (Angel/Lucy)



The contribution Model may be a non-obvious model. We need a table connecting a User to an Event which also records how much money each user paid at an event. Since the number of users participating in an event is variable, we cannot store this data in a fixed number of columns in the Event table.
The debt model may also be non-obvious. In our application we keep track of accumulating debts between two users throughout many shared events.

BEHAVIOR

Feature Descriptions (Christian)

For Minimum Viable Product:

- **Debt Tracker** Automatic calculation of bill-split.
- **Event History** See all the events that you've taken part in.
- **Groups** It is sometimes the case that uses will participate in multiple events with the same group of people. Creating groups allows a way to keep track of these events by

these groups, so that debts, payments, and bills can be kept organized by these users.

If extra time, implement:

- **Facebook/google integration:** our hope is that to ease the strain of signing up and creating an account we will allow for facebook and google + integration. This will also alleviate the stress or pressure of adding friends or people that usually interact.
- **Uneven distribution of splits:** when attending an event the bill can be split unevenly, to allow for different circumstances.
- **IOUS:** allows you to transfer a debt, or portions of it to someone else.
- **Interest:** This can be implemented by IOUs so that they amount of money may accrue interest as time passes.

Security Concerns (Angel)

Key Security Requirements

There are two different types of security concerns:

- **Social Attacks**

  These types of attacks are ones where somebody performs a malicious or false act without breaking into the system. These attacks can occur from situations such as a user leaving his/her account open on a computer and someone else interacting with it. The attacker can then create false events or falsely settle debts.

- **Database attacks**

  These types of attacks are ones where somebody manages to modify or read the database in a way that a user did not intend to. For example, if a RESTful api allows for a user to settle a debt, then an attacker would try to enter the correct URL parameters in order to settle a debt.

Attack Mitigations

In order to mitigate social attacks, the following measures will be taken:

- Users will be logged out after a short period of inactivity.
- Before agreeing to an event/settlement, the user will have to confirm (perhaps with his/her password)

Database attacks will be mitigated with the following measures:

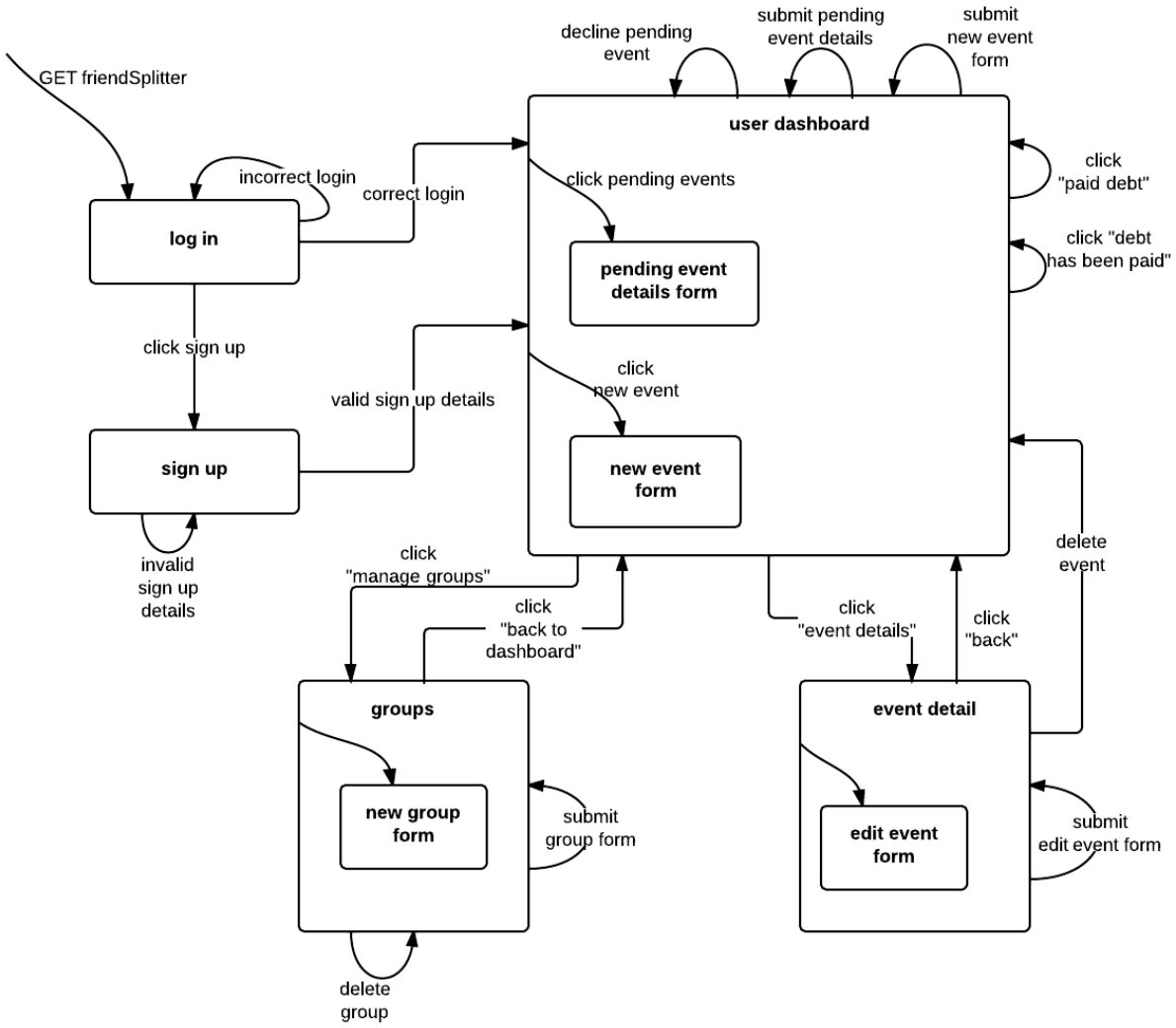
- Before controller actions are used, we will confirm the user is logged in and has the authority to make a change using the rails “before_action” method.

- CSRF attacks will be prevented by using the rails “protect_from_forgery” method.
- Passwords will be encrypted to prevent an attacker from using a password if he/she manages to read them from the database.

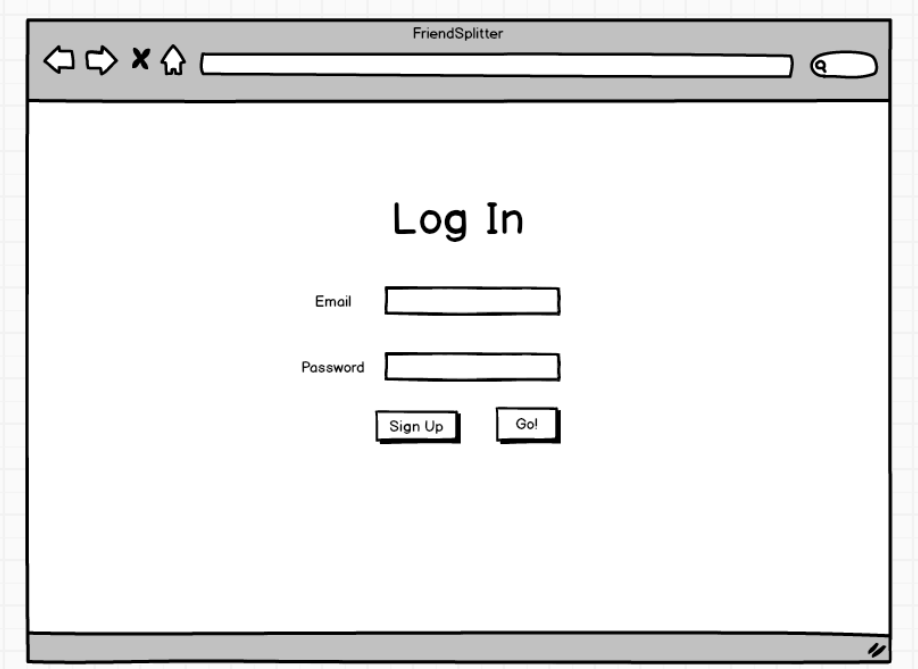
Threat model

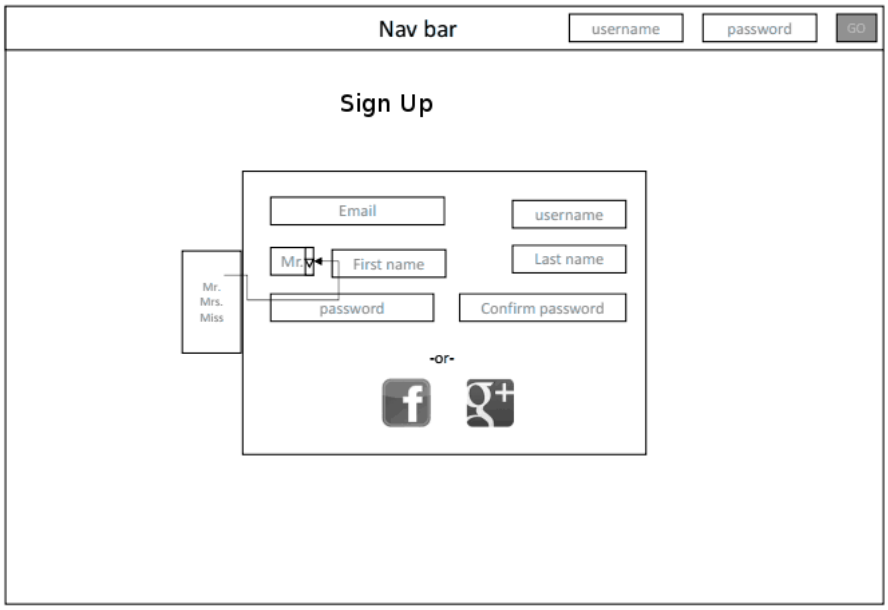
Since Splitter doesn’t handle money and is simply a record book of debts, an attacker is assumed to be someone that the victim knows. In fact, it is more likely that the attacker either owes the victim money and wants to eliminate that debt or is owed money and wants to increase that debt. However, the data isn’t too sensitive since it doesn’t reveal too much information from a user, nor does maliciously changing it cause serious real-world implications (at the worst case, a user might forget to collect a debt from a person).

User Interface (Christian/Lucy)

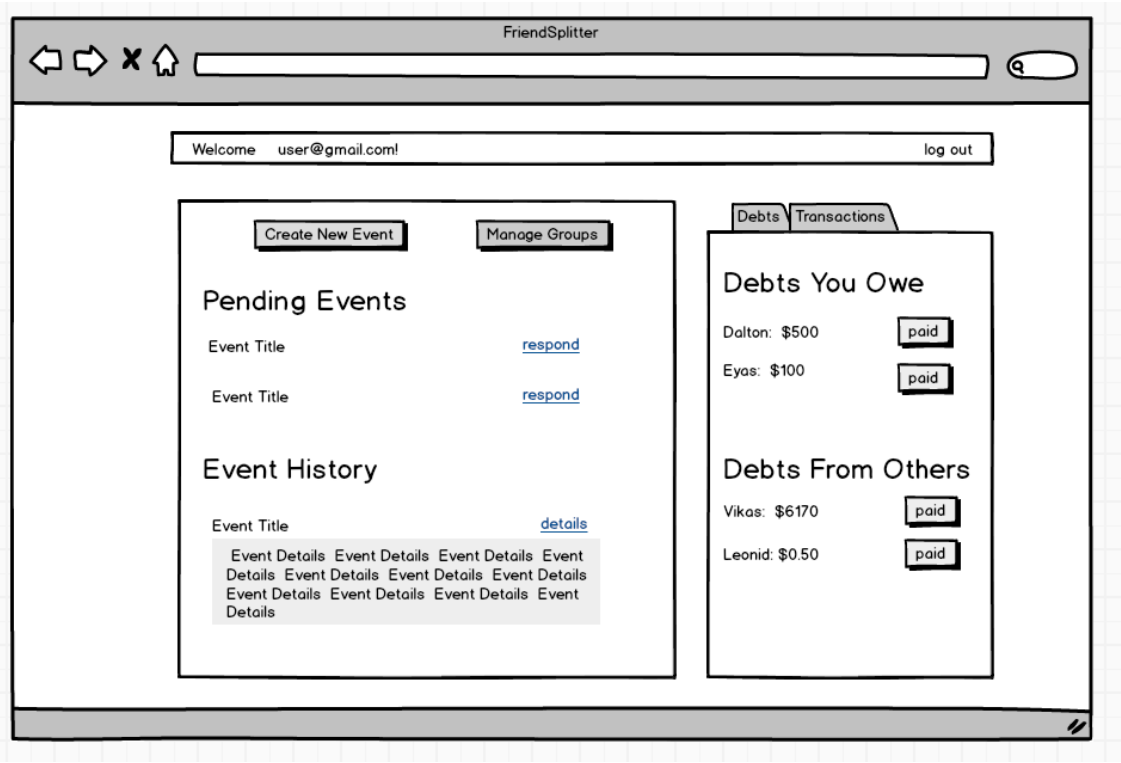


Page Flow

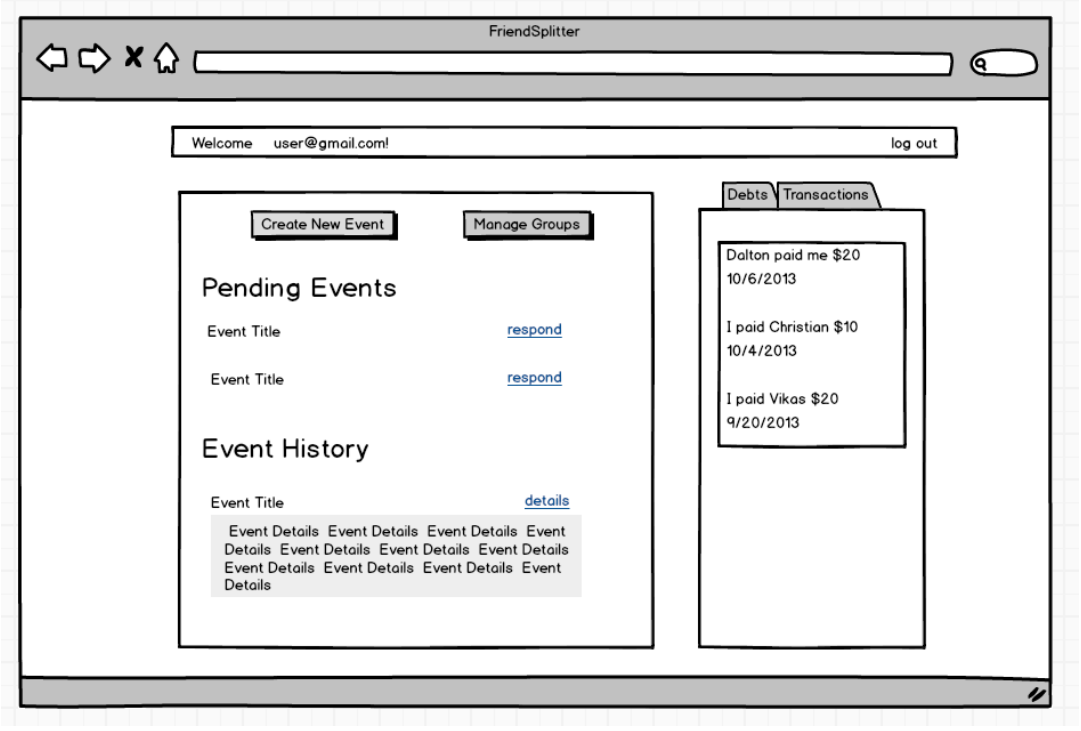

FriendSplitter
Log In
Email
Password
Sign Up
Go!


Nav bar
username
password
GO
Sign Up
Email
username
Mr.
First name
Last name
Mr.
Mrs.
Miss
password
Confirm password
-or-

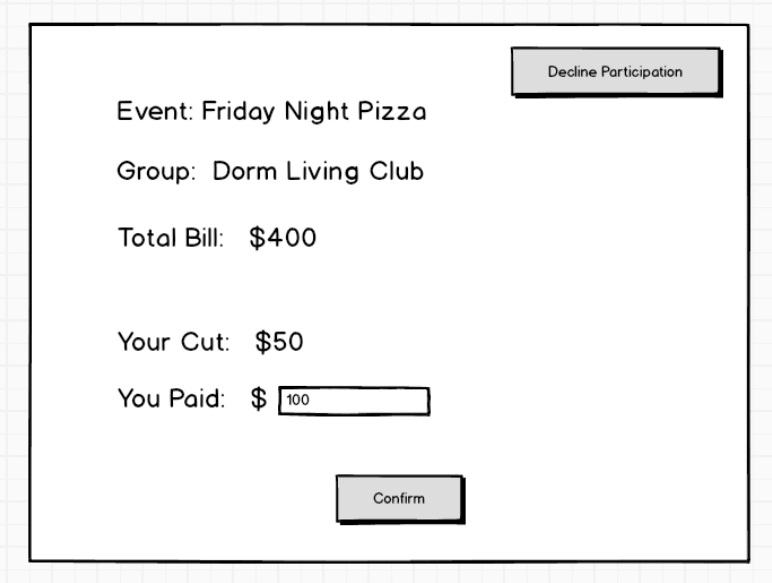
user dashboard debts tab:



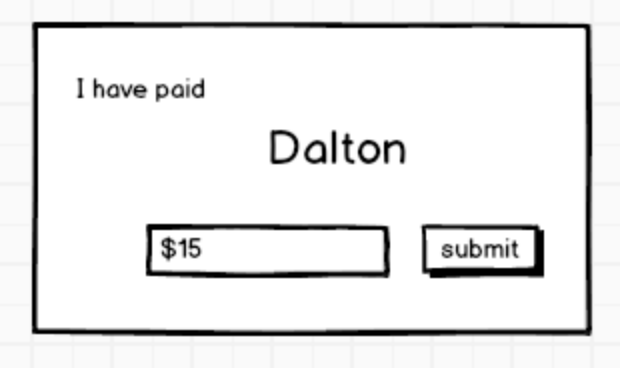
user dashboard transactions tab:



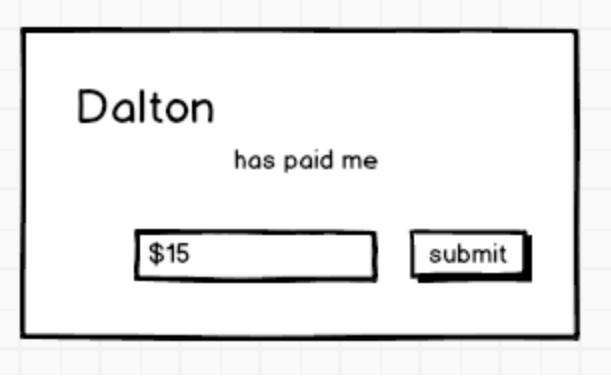
popup from dashboard when clicking “respond” under “Pending Events”:



popup from dashboard when clicking “paid” under “Debts you Owe”:

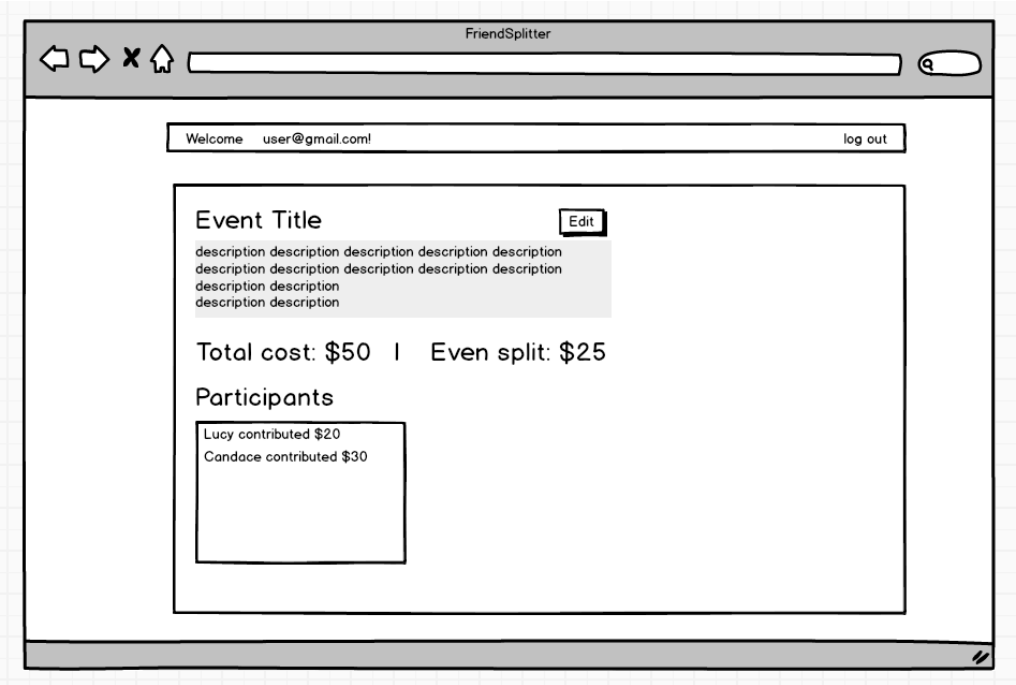


popup from dashboard when clicking “paid” under “Debts From Others”:



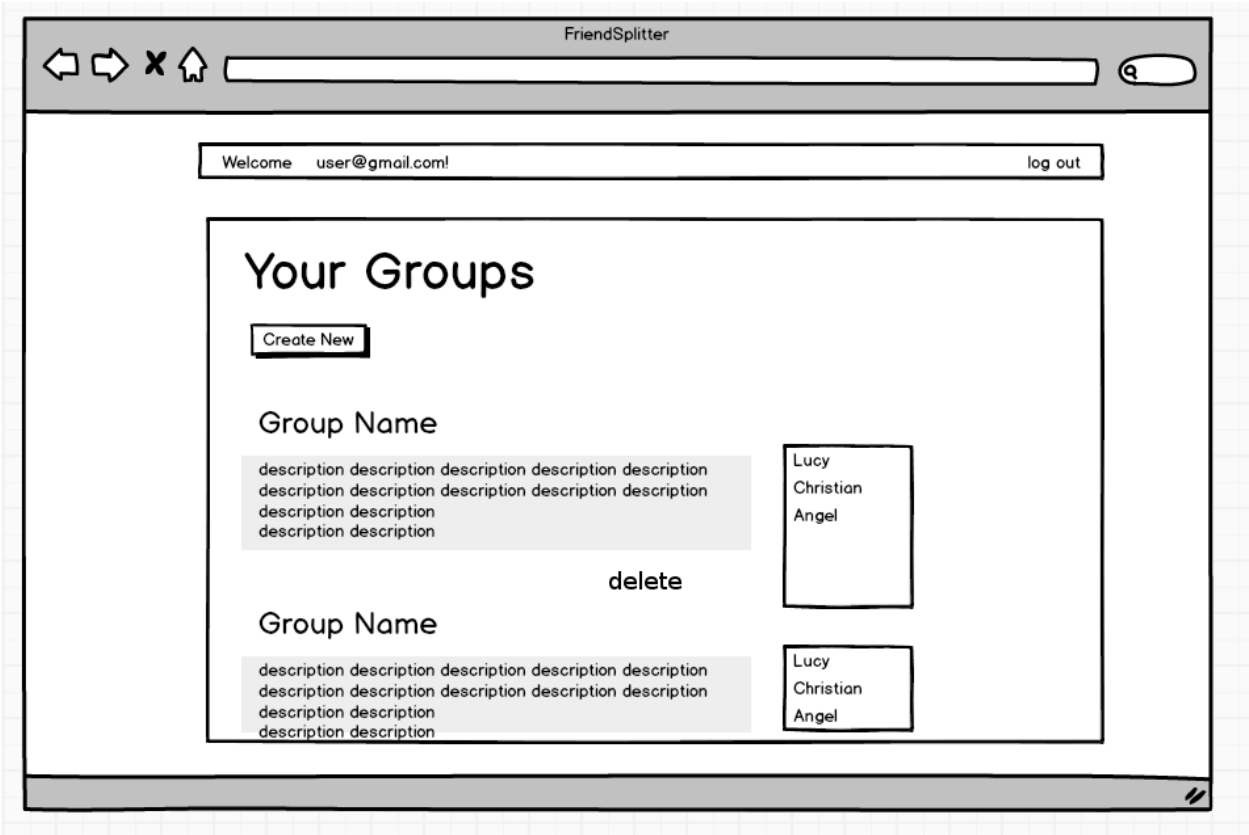
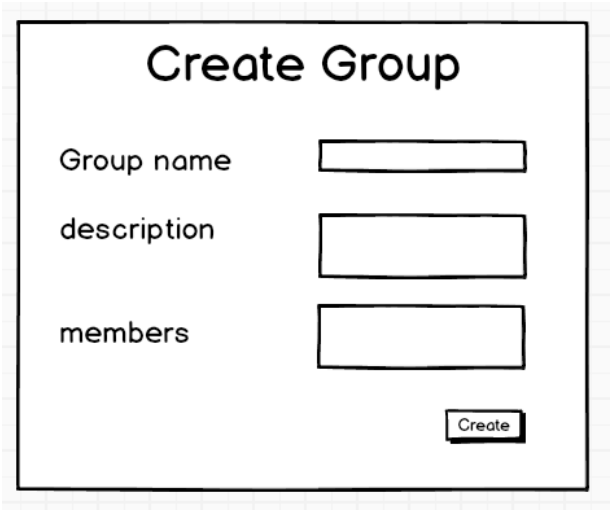
popup from dashboard when clicking “new event”:

Event details:



popup when clicking “edit”:

Groups:



popup when clicking “Create New” group: