Kathryn Siegel, Sylvan Tsai, Ari Vogel
6.170 Project 3.3: Final Version
November 3, 2013

Overview

primary author: Sylvan

System Description

- **Split.li**, an app that allows friends to split a bill at a restaurant

Key goals and purpose

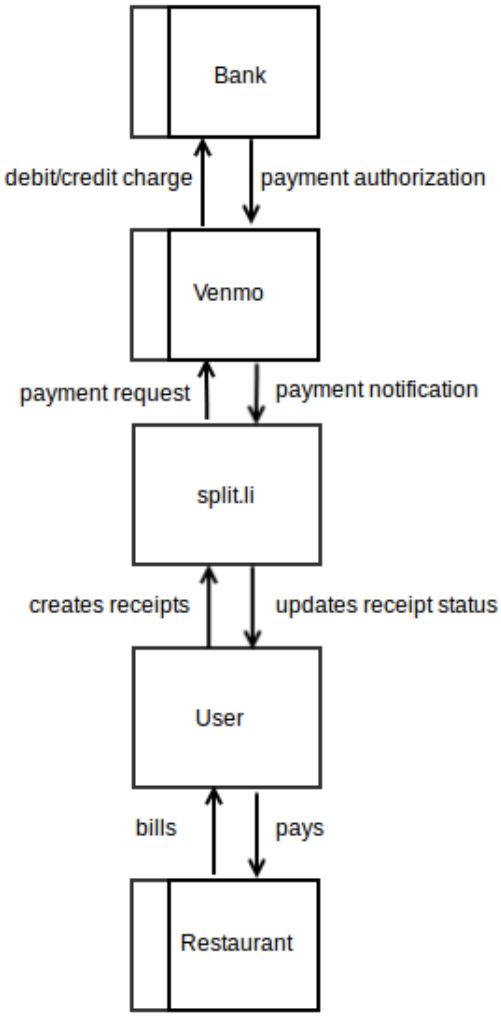
- Facilitates group outings by removing the hassle of calculating who-pays-what and who-owes-whom.
- Offers a convenient way of reimbursing someone in a timely manner.
- Offers a low-effort, automated way to remind others of what they owe and why.
- Caters to two different bill splitting settings, including splitting the bill evenly and attributing items to each person (itemized).

Motivations for development

- Provide an avenue for easily splitting a bill at a restaurant without having to do all the math and factoring in tax/tip by yourself. A night shouldn't be ruined by trivial calculations and searching for change.
- Better tracking of who still owes money to whom
- Allows for more reliable repayment - adds social pressure to paying someone back since they can see who has and has not reimbursed them, via Venmo..

Context Diagram

primary author: all



Concepts

primary author: all

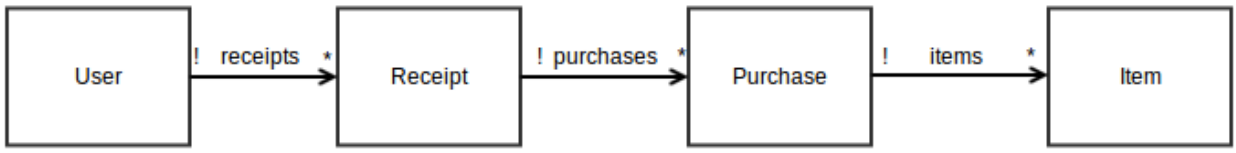
Key concepts

- **Receipt:** A receipt encompasses a set of transactions relating to a single real world receipt. A receipt is the top level organization tool which is named by the User.
- **Item:** Items represent an individual item that would appear on a real world receipt (eg, chicken wings, soup, quesadilla). This item has a price and belongs to a single purchase.
- **Purchase:** A purchase is collection of items, which will be charged to a single person. The purchase is used to associate items to a single user which is responsible for paying for all these items.
- **Charge:** Charging a user means creating a Venmo charge and sending it to the

specified person. When a charge is sent, the user will receive an alert from Venmo, prompting the user to complete or reject the charge. When the user chooses to complete the charge, Venmo will complete the transaction by transferring funds.

- **Itemize:** This is the process of creating a receipt from a real world receipt, by inputting the people paying, and the items each people are responsible for. After a receipt is completed in an itemized fashion, Split.li will divide tax and tip in the appropriate proportions so each user is paying an amount representative of what they owe.
- **Split Evenly:** This process of creating a receipt from a real world receipt simply requires an input of the total amount, and the users to send charges to. Then Split.li divides the costs equally among all inputed users, and all the other parties are charged.
- **Tax and Tip:** Tax and tip are applied to all purchases. This is where Split.li comes in handy as it can distribute tax and tip proportionally for itemized receipts. For evenly split receipts, tax and tip are divided equally.
- **Venmo Account:** In order to use our application, the user must have a Venmo account. This is because we rely on venmo to complete transactions between the user and others who wish to split the bill. Venmo is secure and takes care of many concerns with keeping track of payments and completing transactions.

Data Model



- Textual Constraints
    - If more than one user shares a purchase (e.g. two people split an appetizer), separate purchases are entered into the database with evenly split cost.
    - A VenmoCharge cannot be sent unless the Receipt contains a Purchase of value greater than zero.
    - A Purchase must have a value that is greater than zero.
    - A Purchase is madeBy one of Split.li's users and draws specific details from its items.
    - An item is madeBy one of the users.
    - A receipt sends Venmo charges upon the User's request, and tracks whether or not the charges within the receipt have been sent.

Behavior

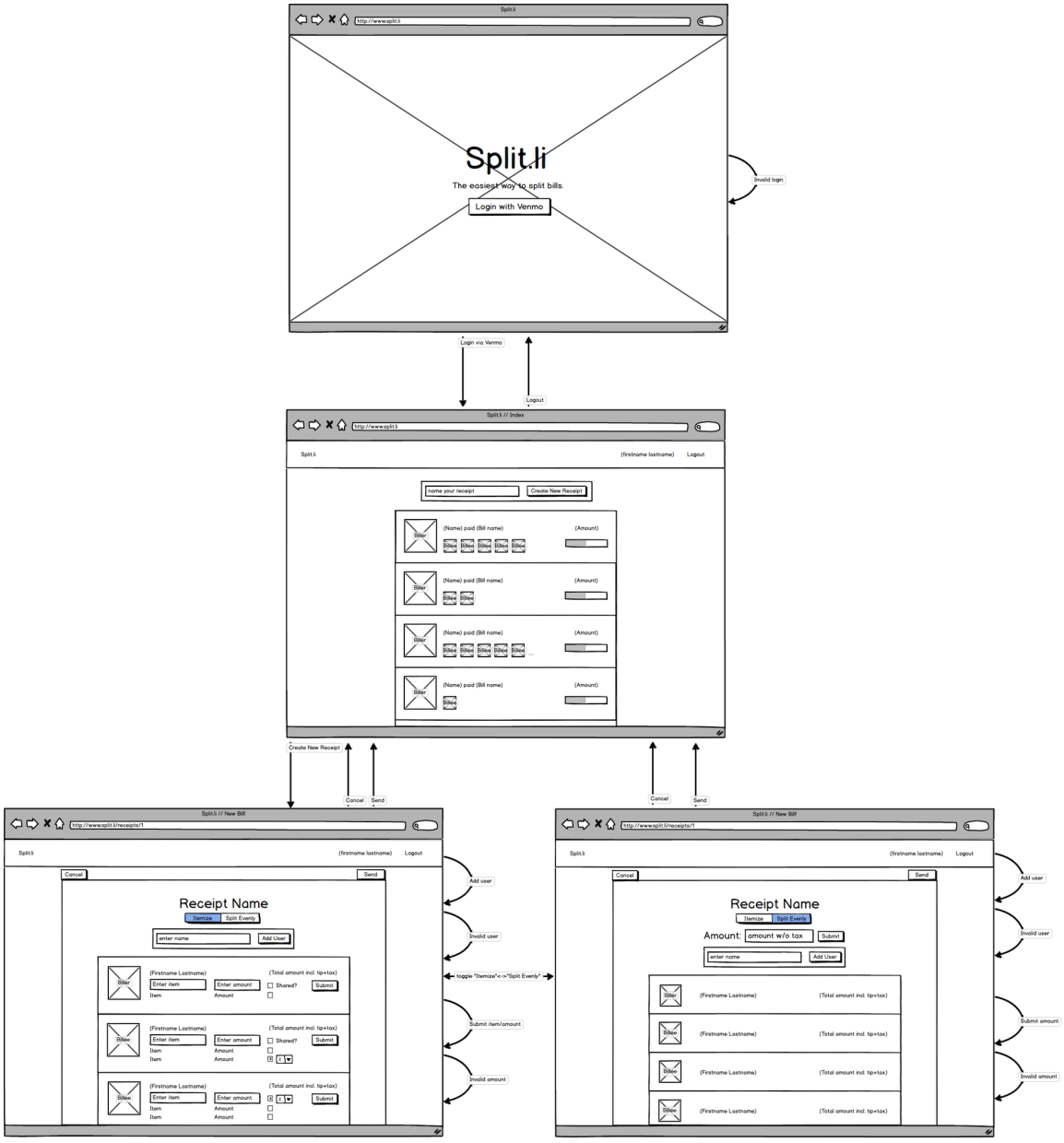
Feature Descriptions

primary authors: Sylvan and Katie

- **Venmo integration.** Sign-up and log-in done through Venmo, allowing reimbursement through Venmo charges and payments.
- **Bill calculator.** Remove the hassle of adding purchases, dividing tax, and scrambling for tip dollars: Split.li will do the math for you.
- **Split evenly function.** Don't care about the details? Split.li can just split the total bill evenly among everyone.
- **Itemized function.** Did certain people order more than others? Split.li will allow you to detail what each person ordered, aggregate each item's cost into a single payment, and then automatically calculate tax and tip.
- **Switching between different charging formats.** Clicking the tab at the top of either receipt editing page creates a completely new receipt of the other format.
- **More reliable reimbursement.** Split.li sends Venmo charges immediately after a receipt is entered, encouraging friends to send payment before even leaving the restaurant.
- **Mobile-friendly.** Use Split.li anytime at your convenience from your mobile browser.
- **Bill splitting history.** Split.li keeps track of all if the payments that you split with your friends in the timeline-style history feature on the main index page.

User Interface

primary author: Katie



(to view full-size, click here: http://web.mit.edu/ksiegel/www/mockup_6170.png)

Security Concerns

primary author: Ari

- Security Requirements:
  - Receipts can only be viewed and edited by the user who created the receipt.
  - Must know other person's number to charge them.
- Potential Risks:
  - Hacker causes all charges to be sent using his venmo account, so completed payments will go to the hacker's account.

    - Hacker gains another user's access token, and utilizes this to send fraudulent payments outside of Split.li.
    - User sends charges that exceed that actual amount of money a friend owes.
- Threat Model:
    - Can assume no interest from state actors or criminal syndicates, since only minimal profile information is stored (Venmo user id). (This sentence taken from example presented in class).
    - All charges must be approved, and include information on who the charge is sent by, so fraudulent charges are difficult to create convincingly.
    - Spamming is likely, as charges can be sent to any phone number.
- Mitigations:
    - Minimal information is stored, only the Venmo user ID and information regarding the receipt composition. By not storing access_tokens, a breach of the database would not release sensitive information.
    - User access control to prevent viewing of basic receipt information by everyone except the receipts creator.
    - Rely on Venmo for secure and accountable transactions.
    - All payments must be confirmed by other users through Venmo (including by text message) this puts responsibility for authentication of purchases on Venmo.
    - Implement session timeouts to avoid insecurities involved with copying cookies or leaving a computer open.
    - Utilizing SSL encrypts cookies, so hackers would need to decrypt a cookie to get a user's access_token.
    - Split.li utilizes rails' protect_from_forgery method to further encrypt information being sent by the app.
    - Split.li's use of Venmo for all payment concerns ensures that securing OAuth tokens will secure users payment permissions, as we do not keep any balances between users.
    - Split.li does not expose OAuth tokens to prevent fradulent Venmo charges

Challenges

Design Challenges

primary authors: Ari and Katie

- How to split a bill between individuals with minimal effort
    - Option 1: Split a bill evenly by inputting a total amount.
        - Pros: This is the simplest implementation, and the user just has to input friends' information and send the charges.
        - Cons: The functionality is somewhat limited, as people can only split a bill evenly
    - Option 2: Itemize each purchase; have the user input all of the items each person purchased, and then split the bill exactly along those lines.
        - Pros: This ensures that everyone pays their fair share, as one person who

has purchased more than everyone else will accordingly be charged a greater amount.
        - Cons: This requires more effort on the part of the user, since he/she has to input every single item that every person purchased. Sometimes, this is unnecessary, as people may just want to evenly split the bill.
    - Decision: we decided to implement both options and give the user the option of whether to split a purchase evenly or to itemize the purchase.
        - This allows for our app to cater to a wider range of situations--i.e. both the even split situation and the itemized situation--without ever compromising on the usability of the app.
        - We also chose to erase any information that the user has inputted into the receipt when he/she changes formats, because of the differences between the two formats.
- How to implement itemized purchases
    - Option 1: Input items individually, with a field for the person
        - Pros: Works well with model of a user going down a receipt and adding each item, then asking who had that item. Easy to have multiple people sharing an item
        - Cons: Must input each users name multiple times thus making the user do more work.
    - Option 2: Input people then add items to each person
        - Pros: Only need to type in each users name once. Intuitive to think of each billee and what items they must pay for.
        - Cons: Makes designing the interaction for splitting an individual item more difficult.
    - Decision: We chose option 2 to make itemizing a receipt as easy as possible. This design minimizes the typing that a User must do, as each user must only be typed in once. Also, when considering how this will be used, we thought that this design aligns with a user's process of splitting a bill: going person by person and adding their contribution to the total bill.

- Account for tax and tip splitting such that each user pays a "fair" amount.
    - Option 1: let users input the additional tax and tip for each payer.
        - Pros: customizable as the user of the app can choose exactly how much each person will pay for tax and tip
        - Cons: user must still do some mental math to decide how much each payer is paying, and make sure enough tax and tip have been included.
    - Option 2: When a receipt is completed, tax and tip will be calculated for each person (proportionally to what their purchase amount) and added to their expense. Then a VenmoCharge will be created.
        - Pros: the only work for the user is adding tax and tip percentages as necessary (tax varies by states), no mental math required
        - Cons: removes ability of the user to set tax and tip amounts on a person

by person basis.
    - Option 3: Allow users to mark whether they have separately calculated tax on their own using a checkbox, and have users input the percentage tip each time they create a receipt.
        - Pros: Allows the app to apply to a greater range of situations. For example, when splitting the whole bill, the user may or may not want to have the submitted total include tax.
        - Cons: Creates another input that the user has to worry about.
    - Decision: We chose a mix between option 2 and option 3.
        - As an application that strives to remove the human element from calculating a bill, it makes more sense to for the app to calculate the tax and tip for the user, if they choose for the app to do so. However, sometimes the user may want to exclude tax in the final calculation, so we allow them to do so.
        - We have different defaults for the two different types of receipts (split evenly and itemized). For receipts that are split evenly, we thought that it would be more common that users would include tax in their total amount, so that is the default setting. For receipts that are itemized, we thought that it would be more common that users would not include tax in each inputted item, so that is the default setting.

References

- Third-Party Gems
    - twitter-bootswatch-rails, twitter-bootswatch-rails-helpers
    - nokogiri, rest-client
    - therubyracer
    - omniauth-venmo
    - rack-ssl-enforcer