# Party MGR

Intro

Technical Plan Team Members Delegation of Tasks Components Dependency Graph

Completion Schedule

GUI Prototype Changes ­ 4/14/2015 GUI Prototypes

Home Page Event Detail: Event Overview: Settings:

iOS Mockups DB Design

Test Plan Version

### Intro

Party MGR will be developed as an iPhone app and a website that target youth parties for kids between the ages of 3 and 11. The service will handle electronic party invitations but its main focus will be on the gift buying aspect. The host of the party will be given the option to specify a list of the top gifts their child wants. Invitees will then be able to contribute towards these gifts and the child will be able to get more meaningful gifts.

### Technical Plan

This section will describe the software and services we will use when developing the software for Party MGR.

|  |  |
| --- | --- |
| **Category** | What are we using? |
| **Communication** |  |
| Email | Gmail |
| Web Conferencing | Google Hangouts |

|  |  |
| --- | --- |
| Instant Messaging | Slack |
| **Collaboration** |  |
| Document Collaboration | Google Drive |
| File Sharing/Data Tracking | BitBucket |
| Developmental Task Delegation | Asana |
| **Web Development** |  |
| Environment | Mac OS, Linux, Windows |
| Browsers Supported | Chrome, IE11+, Safari, Firefox |
| Frontend | HTML 5, CSS 3, JQUERY |
| Backend | PHP |
| Database | MySQL |
| **iOS Development** |  |
| Environment | Mac OS |
| Devices Supported | iPhone |
| Device OS Supported | iOS 7+ |
| Software | Xcode |
| Development Language | Objective­C |

### Team Members

Vania Nettleford: Project Manager (nia724@gmail.com) Noel Whitaker: Test Manager (noelw55@gmail.com)

Steven Harrigan: Database Manager (def.monk@gmail.com) Stephen Hayes: Documentation Manager (schayes4@gmail.com) Julian Sexton: Test Manager (protojas@gmail.com)

### Delegation of Tasks

Vania Nettleford

Tasks assigned for the past week:

* Assign development tasks in Asana
* Bug fixes
* Continue testing the site for bugs Tasks completed for the past week:
* Assigned development tasks in Asana
* fixed bugs with contributions (saving to backend correctly & redirecting after displaying a message)
* optimized invitation code Tasks assigned for next week:
* Assign development tasks in Asana
* Bug fixes

Noel Whitaker

Tasks assigned for the past week:

* Work on assigned bug fixes
* Update Account Info DB/Endpoint to allow payment preference to be posted back expect a string to be sent

Tasks completed for the past week:

* Update Account Info DB/Endpoint to allow payment preference to be posted back expect a string to be sent
* Bug fixes

Tasks assigned for next week:

* Bug fixes

Steven Harrigan

Tasks assigned for the past week:

* Test the site for bugs
* Replace forgot my password legacy code with new endpoint Tasks completed for the past week:
* Testing (ongoing)
* Fixed UI issues with site
* Logo on website screenshots to show John Tasks assigned for next week:
* Add logo to the site
* Fix ui issues with the site
* Bug fixes

Stephen Hayes

Tasks assigned for the past week:

* Organize weekly deliverable
* Add in term of service & privacy policy
* Fix bugs in app

Tasks completed for the past week:

* Organized weekly deliverable
* Added terms of service, privacy policy, FAQs, and about to app
* Fixed bugs in app
* Connected forgot password to backend
* Added SSL certificate to server Tasks assigned for next week:
* Organize weekly deliverable
* Fix settings connecting to backend in app
* Bug fixes
* Improve UI for contribution thank you modal

Julian Sexton

Tasks assigned for the past week:

* Implement user wants to contact us
* When a user registers/signup add in a checkbox and next to it add “Accepts terms of service & privacy policy” before a user can register

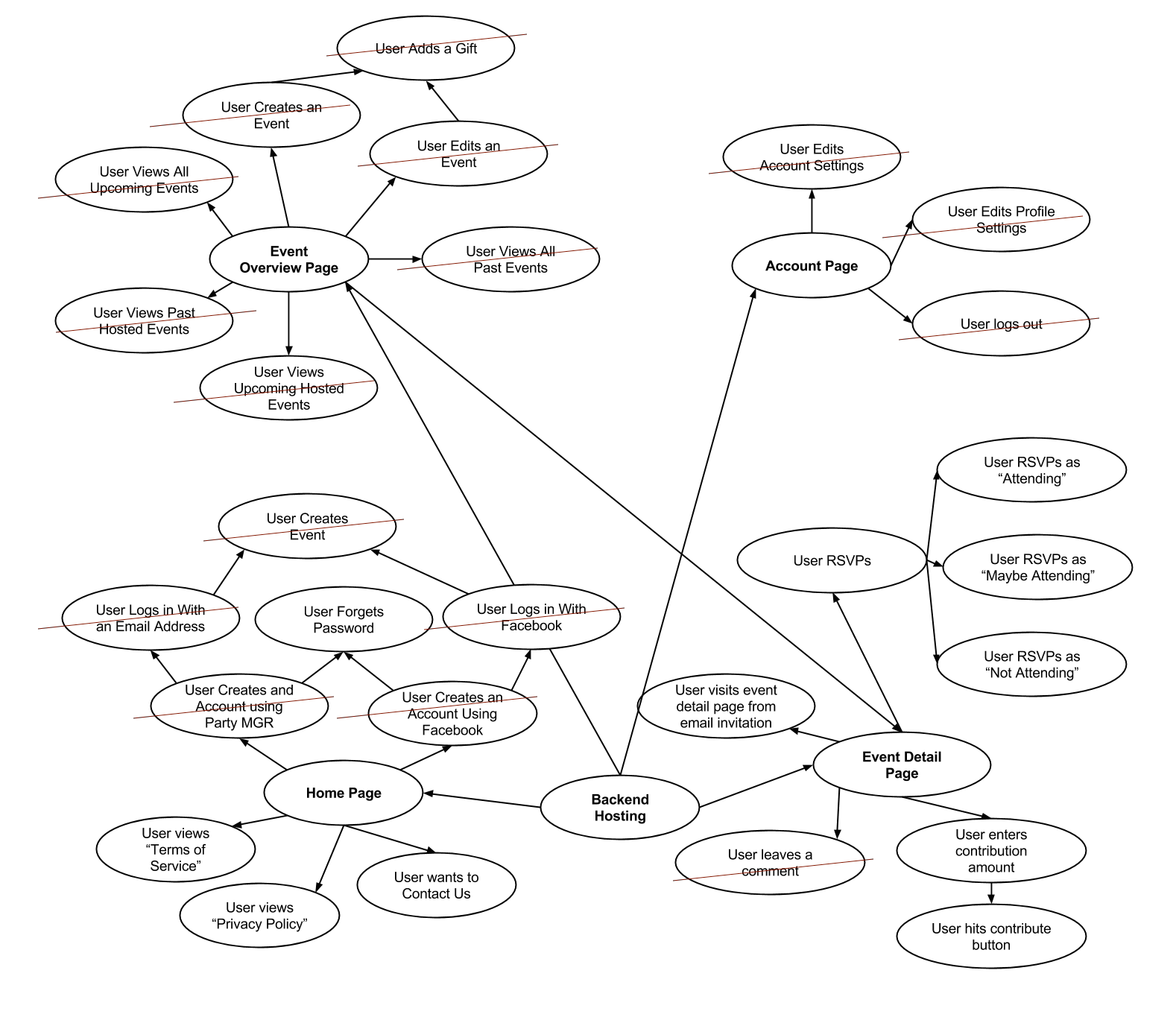
Tasks completed for the past week:

* When a user registers/signup add in a checkbox and next to it add “Accepts terms of service & privacy policy” before a user can register
* Implement user wants to contact us
* updated hotmail import code Tasks assigned for next week:
* fix bug with invitations modal

### Components

* Back End: make APIs for front end to connect with
  + Login/Register
  + Home Page
  + Event Detail Page
  + Administration Page
  + Event Administration Page
  + Account Page
* Login/Register: allows user to log in/sign up on the site via Facebook or using email address and password
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
  + Integrate with Facebook API
* Home Page: allows user to see a list of their upcoming event they are invited to or hosting
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
* Event Detail Page: allows users to see more detail about a specific event
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
* Payments: allow users to contribute money to a gift from an event
  + Website frontend layout
  + iPhone app frontend layout
  + Backend integration with 3rd party payments company (Stripe, Paypal)
* Administration Page: allows for users to see events they are hosting, create new events, and search for friends’ events
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
* Event Administration Page: allows a user to edit/update an event they are hosting
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
* Account Page: allows user to manage their account settings and information
  + Website frontend layout
  + iPhone app frontend layout
  + Integrate with Backend
* Contact Us
  + Website frontend layout
  + iPhone app frontend layout
* Privacy Policy
  + Website frontend layout
  + iPhone app frontend layout
* About
  + Website frontend layout
  + iPhone app frontend layout
* FAQs
  + Website frontend layout
  + iPhone app frontend layout
* Terms of Service
  + Website frontend layout
  + iPhone app frontend layout
* Database Setup
  + Backend API support
* Hosting/Domain name
* Security
  + Use bcrypt as a salt and SHA­512 to store password securely
* Logo

### Dependency Graph



**Completion Schedule**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Date** | **Website Frontend** | | **iPhone App Frontend** | **Backend API** | |
| 9/30/2014 | ­ | ~~Base layout~~ |  |  | |
| 10/7/2014 | ­ | ~~Base layout~~ |  |  | |
| ­ | ~~Site navigation~~ |
| 10/14/2014 | ­ | ~~Color scheme~~ |  |  | |
| 10/21/2014 | ­ | ~~User creates an~~ |  |  | |
|  | ~~account using Party~~ |
|  | ~~MGR~~ |
| ­ | ~~User creates an~~ |
|  | ~~account using~~ |
|  | ~~Facebook~~ |
| ­ | ~~User logs in with~~ |
|  | ~~Facebook~~ |
| ­ | ~~User logs in with~~ |
| ~~email address~~ |
| 10/28/2014 | ­ | ~~User visits event~~ |  |  | |
|  | ~~detail page~~ |
| ­ | ~~User RSVPs~~ |
| 11/4/2014 | ­ | ~~User enters~~ |  |  | |
|  | ~~contribution amount~~ |
| ­ | ~~User logs out~~ |
| ­ | ~~User edits account~~ |
| ~~Settings~~ |
| 11/11/2014 | ­ | ~~User visits event~~ |  | ­ | ~~Database setup~~ |
|  | ~~detail page~~ | ­ | ~~User creates an~~ |
|  | ~~(updated)~~ |  | ~~account using~~ |
| ­ | ~~User views~~ |  | ~~Facebook~~ |
|  | ~~upcoming hosted~~ | ­ | ~~User creates an~~ |
|  | ~~events~~ |  | ~~account using~~ |
| ­ | ~~User views all~~ |  | ~~Party MGR~~ |
|  | ~~upcoming events~~ | ­ | ~~User logs in with~~ |
| ­ | ~~User creates an~~ |  | ~~Facebook~~ |
|  | ~~event~~ | ­ | ~~User logs in with~~ |
| ­ | ~~User edits account~~ | ~~email address~~ |
| ~~settings (updated)~~ |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ­ | ~~User edits profile~~ ~~settings~~ |  | |  | |
| 11/18/2014 | ­ | ~~User edits an event~~ |  | | ­  ­  ­ | ~~User creates an~~ ~~event (local)~~ ~~User edits an~~ ~~event (local)~~  ~~User views event~~ ~~detail (local)~~ |
| 11/25/2014 | ­ | ~~User edits an event~~ ~~(continued)~~ |  | | ­  ­  ­  ­  ­  ­ | ~~User creates an~~ ~~event (local)~~ ~~User adds a gift~~ ~~(local)~~  ~~User views all~~ ~~past hosted~~ ~~events (local)~~ ~~User views all~~ ~~upcoming hosted~~ ~~events (local)~~ ~~User views all~~ ~~upcoming~~ ~~attending events~~ ~~(local)~~  ~~User views all~~ ~~past attending~~ ~~events (local)~~ |
| 12/2/2014 |  | | ­ | ~~Initial project~~ ~~setup~~ |  | |
| 12/9/2014 |  | |  | | ­ | ~~User edits~~ |
|  | ~~account settings~~ |
| ­ | ~~User edits profile~~ |
| ~~settings~~ |
| 12/16/2014 |  | |  | | ­ | ~~User invites~~ |
|  | ~~guests~~ |
| ­ | ~~User leaves a~~ |
| ~~comment~~ |
| 12/23/2014 |  | |  | |  | |
| 12/30/2014 |  | | **Winter Break** | |  | |
| 1/6/2015 |  | | ­ | ~~User creates an~~ |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | ­ | ~~account using~~ |  | |
| ~~Party MGR~~ |
| ~~User creates an~~ |
|  | ~~account using~~ |
|  | ~~Facebook~~ |
| ­ | ~~User logs in~~ |
|  | ~~with Facebook~~ |
| ­ | ~~User logs in~~ |
| ~~with email~~ |
| ~~address~~ |
| 1/13/2015 |  | ­ | ~~User views all~~ |  | |
|  | ~~past hosted~~ |
|  | ~~events~~ |
| ­ | ~~User views all~~ |
|  | ~~upcoming~~ |
|  | ~~hosted events~~ |
| ­ | ~~User views all~~ |
|  | ~~upcoming~~ |
|  | ~~attending~~ |
|  | ~~events~~ |
| ­ | ~~User views all~~ |
| ~~past attending~~ |
| ~~events~~ |
| 1/20/2015 | * ~~User leaves a~~ ~~comment on the~~ ~~event page~~ * ~~User invites guests~~ * ~~User adds a gift~~ |  | |  | |
| 1/27/2015 |  | ­  ­ | ~~User leaves a~~ ~~comment~~  ~~User creates an~~ ~~event~~ | ­ | ~~User hits~~ ~~contribute button~~ ~~(continued)~~ |
| 2/3/2015 |  | ­ | ~~User edits an~~ |  | |
|  | ~~event~~ |
| ­ | ~~User adds a gift~~ |
| 2/10/2015 | * ~~User enters~~ ~~contribution amount~~ * ~~User hits contribute~~ ~~button~~ | ­  ­ | ~~User enters~~ ~~contribution~~ ~~amount~~ ~~User hits~~  ~~contribute button~~ |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2/17/2015 | ­ ~~User hits contribute~~ ~~button (continued)~~ | ­ | ~~User hits~~ ~~contribute button~~ ~~(continued)~~ |  | |
| 2/24/2015 | * ~~Guest enters~~ ~~contribution amount~~ * ~~Guest contributes to~~ ~~an event~~ * ~~User hits contribute~~ ~~button (continued)~~ | ­  ­  ­ | ~~User visits~~ ~~event detail~~ ~~page~~  ~~User RSVPs~~ ~~User leaves a~~ ~~comment~~ | ­  ­  ­ | ~~User forget~~ ~~password~~ ~~User enters~~ ~~contribution~~ ~~amount~~  ~~User visits event~~ |
| ­ | ~~User creates an~~ |  | ~~detail page from~~ |
|  | ~~event~~ |  | ~~email~~ |
| ­ | ~~User adds a gift~~ | ­ | ~~User hits~~ |
|  | ~~contribute button~~ |
| ­ | ~~User creates an~~ |
|  | ~~event (server)~~ |
| ­ | ~~User edits an~~ |
|  | ~~event (server)~~ |
| ­ | ~~User views event~~ |
|  | ~~detail (server)~~ |
| ­ | ~~User RSVPs to~~ |
| ~~event (Attending,~~ |
| ~~Maybe~~ |
| ~~Attending, Not~~ |
| ~~Attending)~~ |
| 3/3/2015 | * ~~User forgets~~ ~~password~~ * ~~User enters in~~ ~~information and~~ ~~contributes to an~~ ~~event using stripe~~ | ­ | ~~User hits~~ ~~contribute~~ ~~button~~ | ­ | ~~User hits~~ ~~contribute button~~ ~~(continued)~~ |
| 3/10/2015 | ­ ~~User resets~~ ~~password~~ | ­  ­ | ~~User edits~~ ~~account settings~~ ~~User edits~~  ~~profile settings~~ | ­ | ~~User visits event~~ ~~detail page from~~ ~~email invitation~~ |
| 3/17/2015 | * ~~User visits event~~ ~~detail page from~~ ~~email invitation~~ * ~~User enters in~~ ~~information and~~ ~~contributes to an~~ | ­  ­ | ~~User visits~~ ~~event detail~~  ~~page from email~~ ~~invitation~~  ~~User forgets~~ ~~password~~ |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | ~~event using stripe~~ | | ­ | ~~User edits an~~ |  | |
| ~~(continued)~~ | |  | ~~event~~ |
| ­ | ~~User edits~~ |
|  | ~~account settings~~ |
| ­ | ~~User edits~~ |
|  | ~~profile settings~~ |
| ­ | ~~User contributes~~ |
| ~~to an event~~ |
| 3/24/2015 | ­  ­ | ~~User views terms of~~ ~~service~~  ~~User views privacy~~ ~~policy~~ | ­ | ~~User contributes~~ ~~to an event~~ | ­ | ~~Backend hosting~~ |
| 3/31/2015 | ­ | ~~Logo~~ | ­  ­  ­ | ~~Logo~~  ~~User visits~~ ~~event detail~~  ~~page from email~~ ~~invitation~~  ~~User forgets~~ ~~password~~ |  | |
| 4/7/2015 ­ | ­ | ~~User wants to~~ | ­ | Polishing | ­ | Polishing |
| End |  | ~~contact us~~ | ­ | Bug fixes | ­ | Bug fixes |
| ­ | Polishing | ­ | ~~Terms of~~ |
| ­ | Bug fixes | ~~service, privacy~~ |
| ­ | ~~SSL Certificate~~ | ~~policy~~ |

### GUI Prototype Changes ­ 4/14/2015

* Updated completion schedule

### GUI Prototypes

* GUI prototypes contain mockups for current pages of the website
* Along with each GUI mockup are use cases for how the user and the system will interact on that page

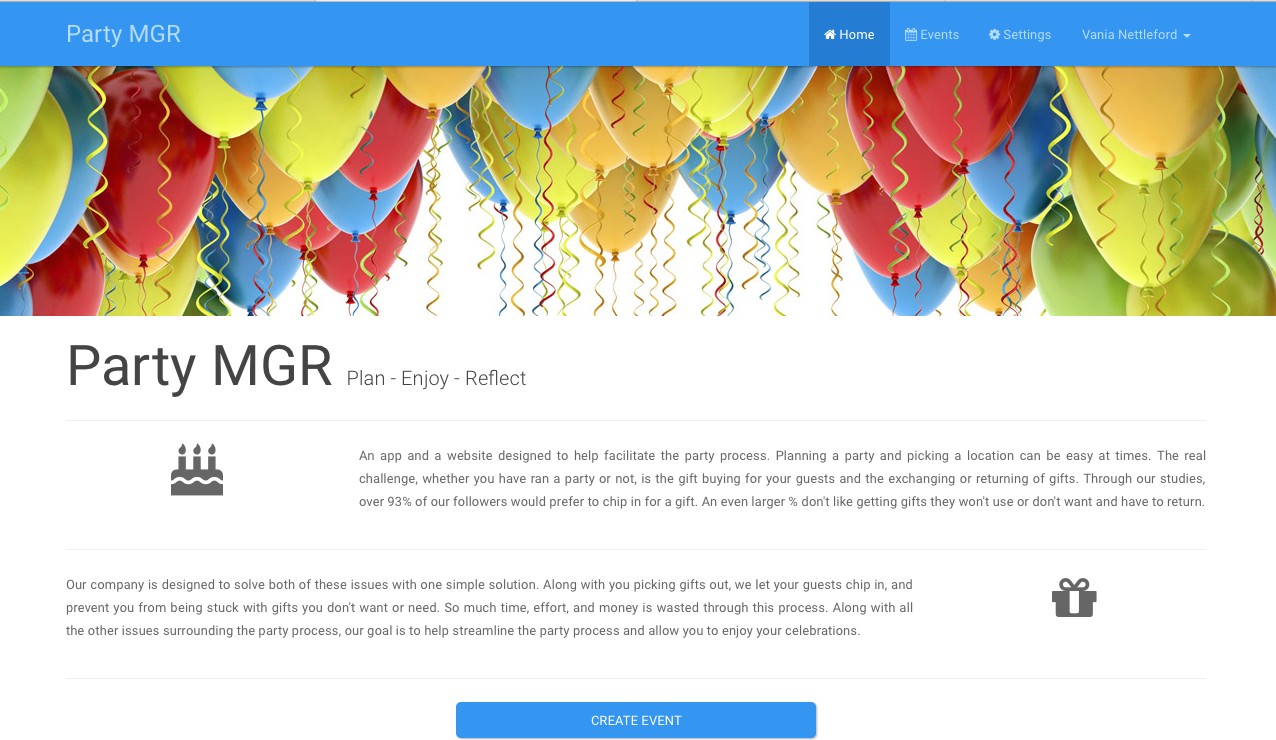
# Home Page

1. User Views “Privacy Policy"
2. User Views “Terms of Service”
3. User Wants to Contact Us
4. User Registers
5. User Creates An Account Using Facebook
6. User Creates An Account Using Party Mgr
7. User Logs in With Facebook
8. User Logs in With An Email Address
9. User Forgets Password
10. User Creates Event

This screen is shown the first time a user visits the Party MGR page or if a user isn’t logged in.

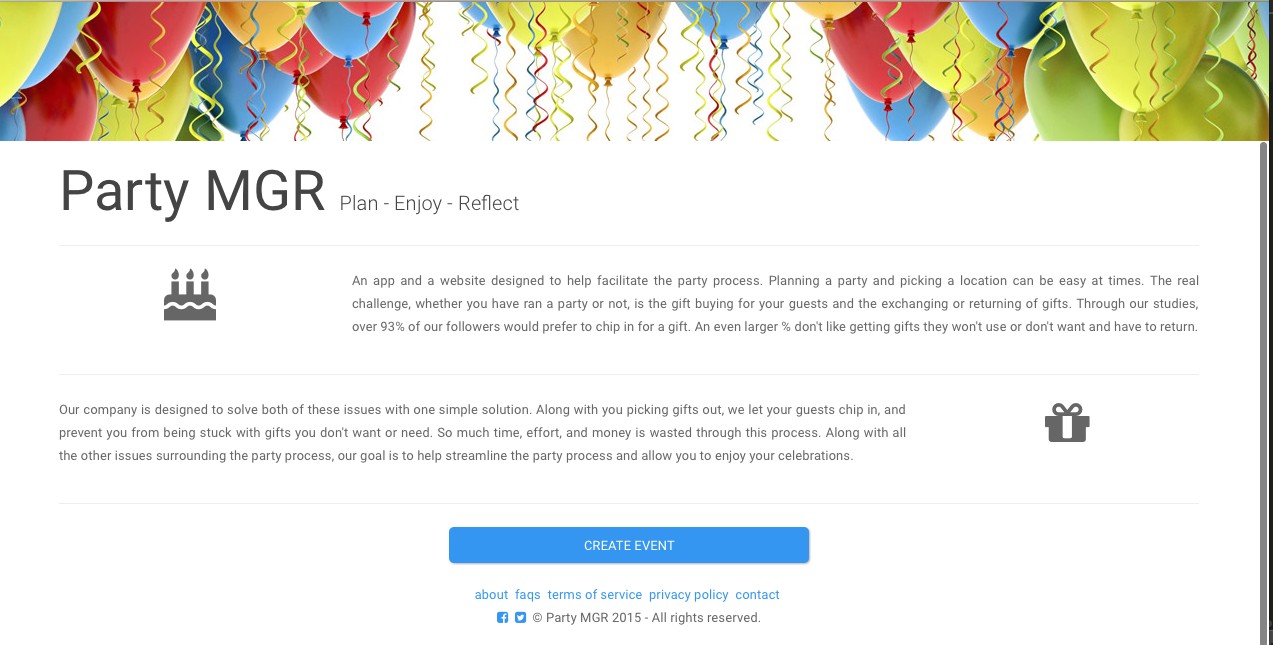
### Before

**User Views “Privacy Policy"**



**During**

User presses “Privacy Policy” and is taken to a page displaying the privacy policy.

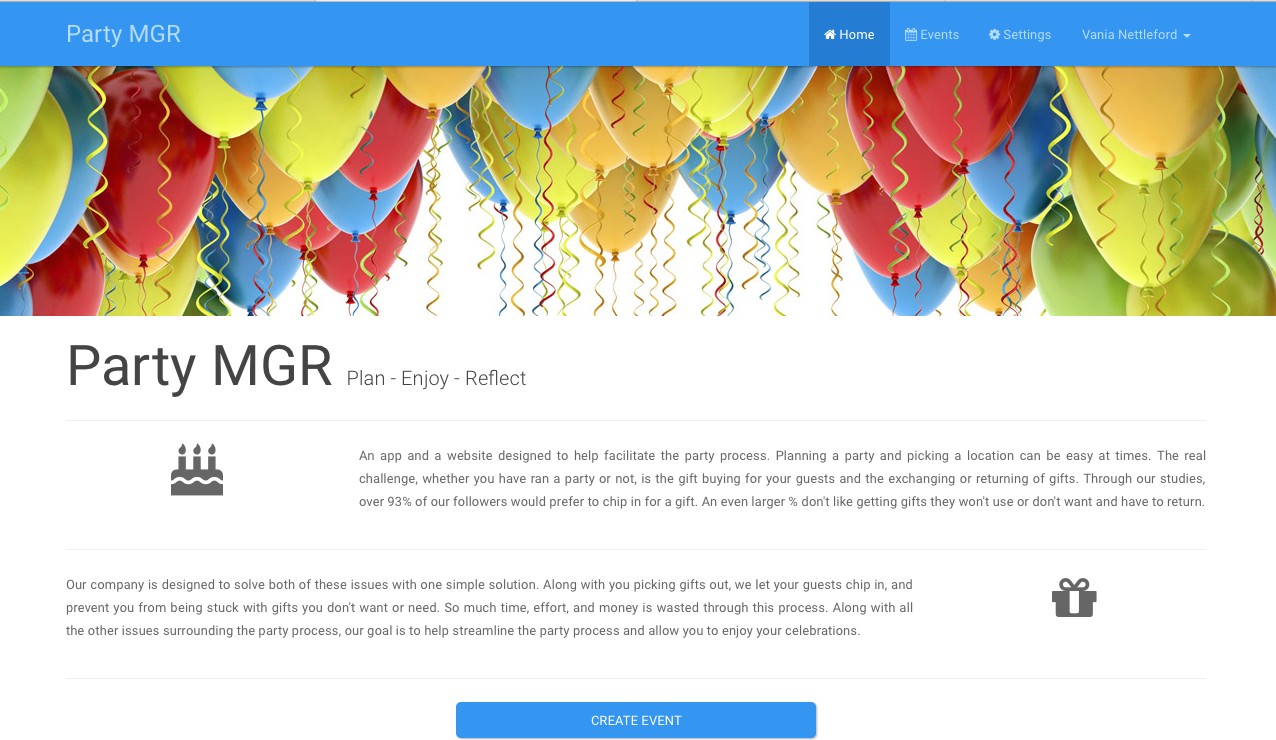


### After



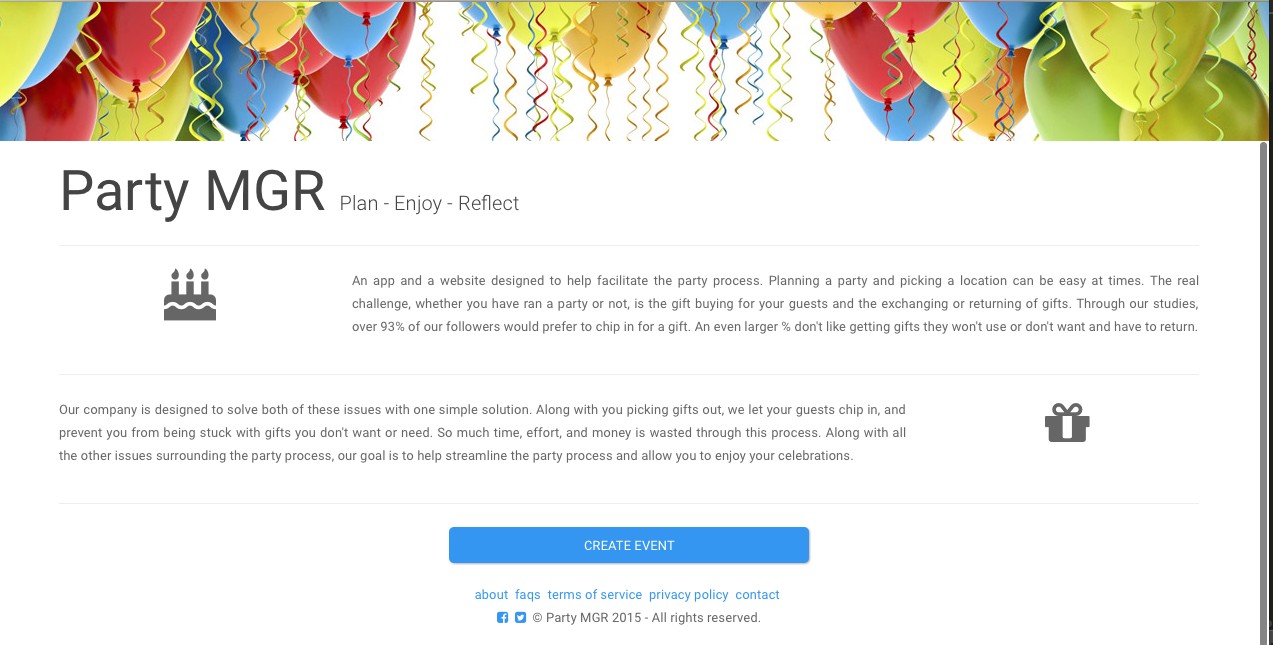
**Before**

**User Views “Term of Service"**



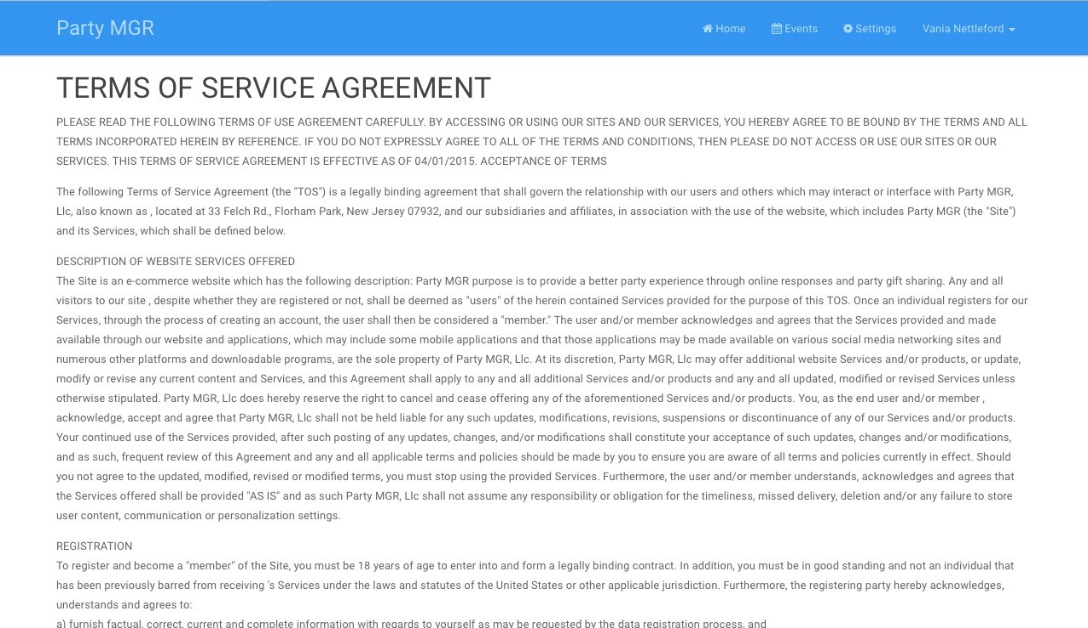
**During**

User presses “Terms of Service” and is taken to a page displaying the terms of service.



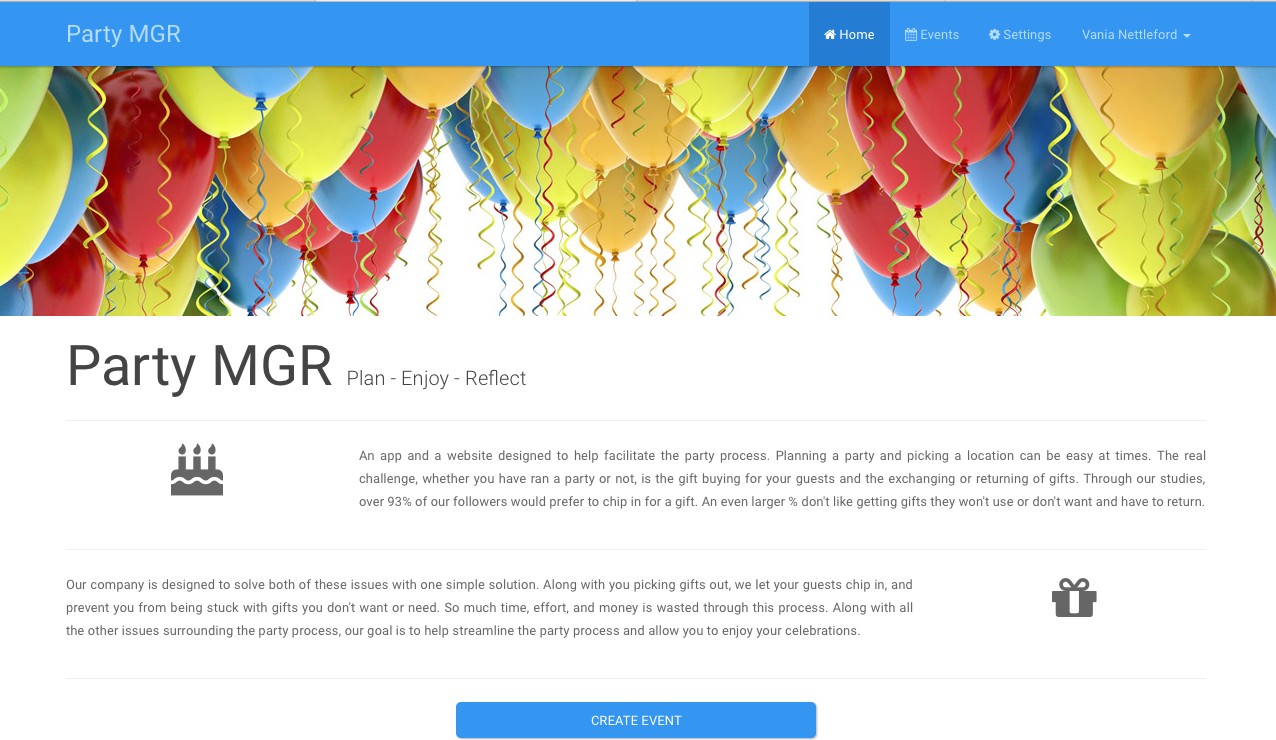
Terms of Service

### After



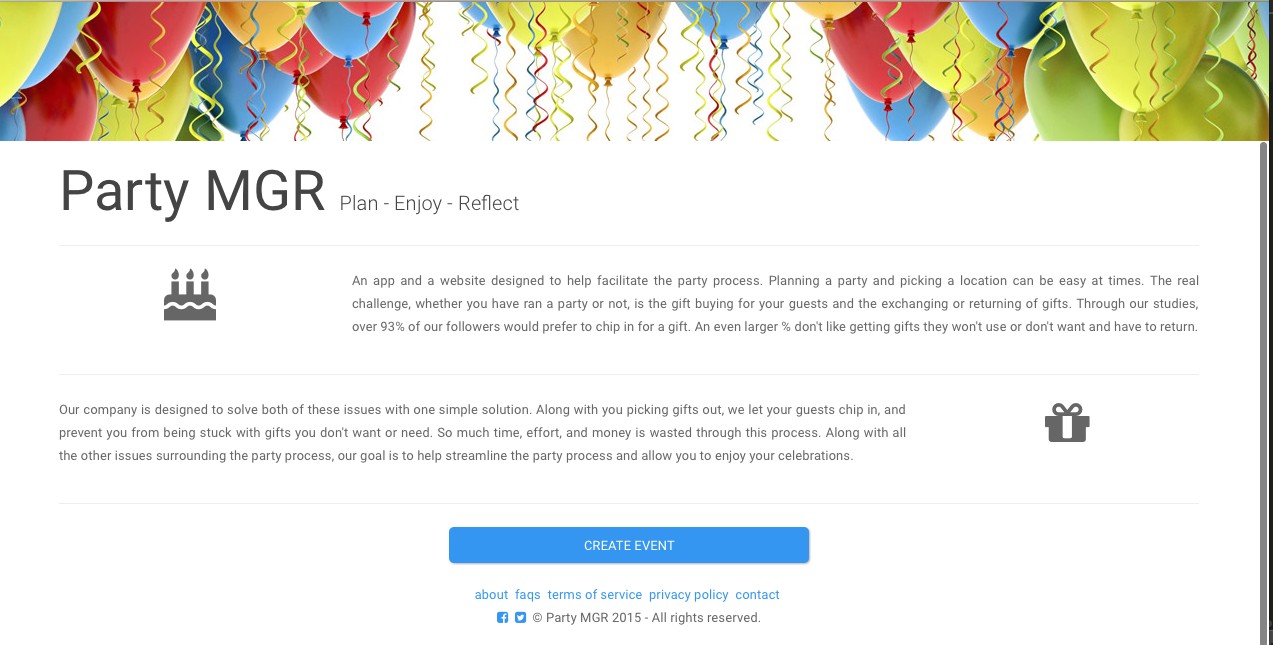
**Before**

**User Wants to Contact Us**

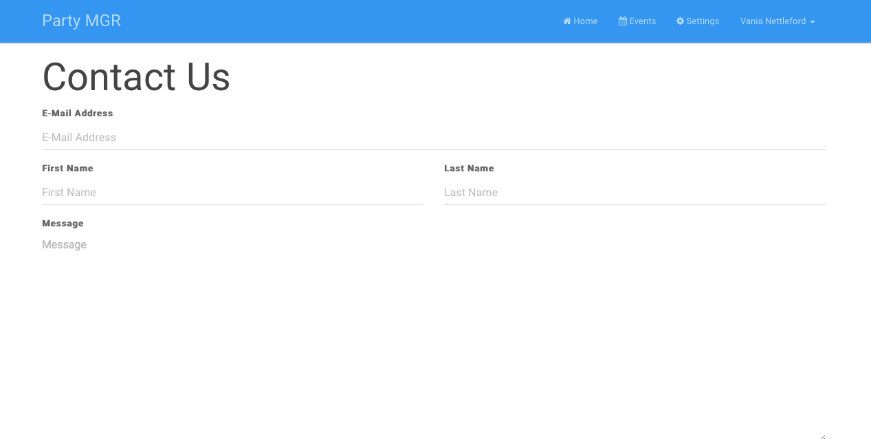


**During**

User presses “Contact” and is taken to a page with information to contact the site.

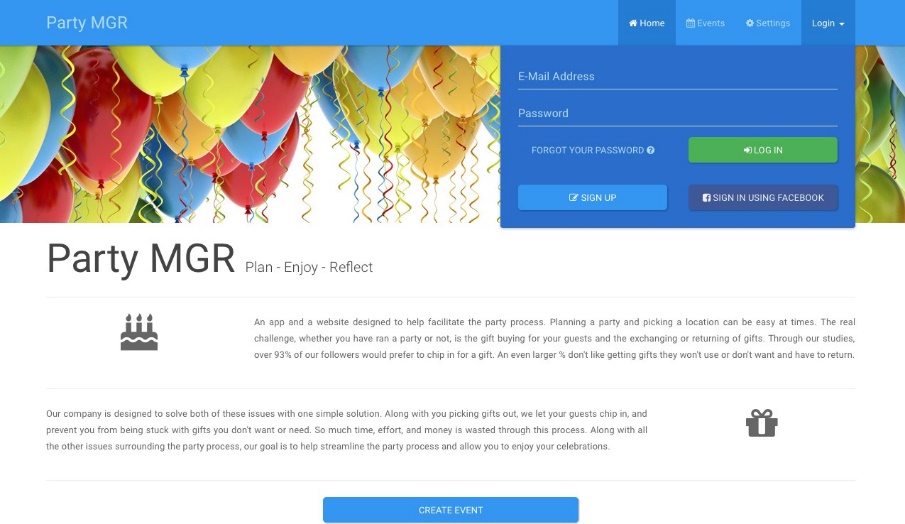


### After



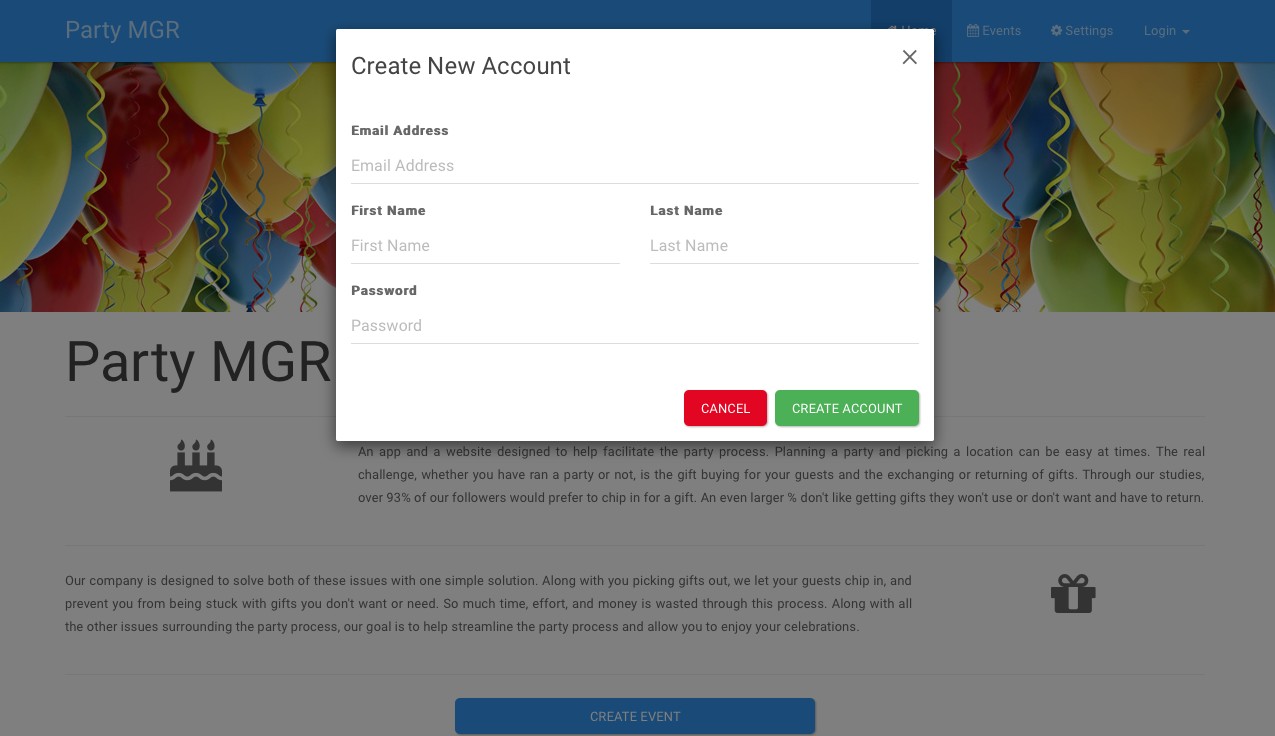
**User Wants to Register**

**Before**

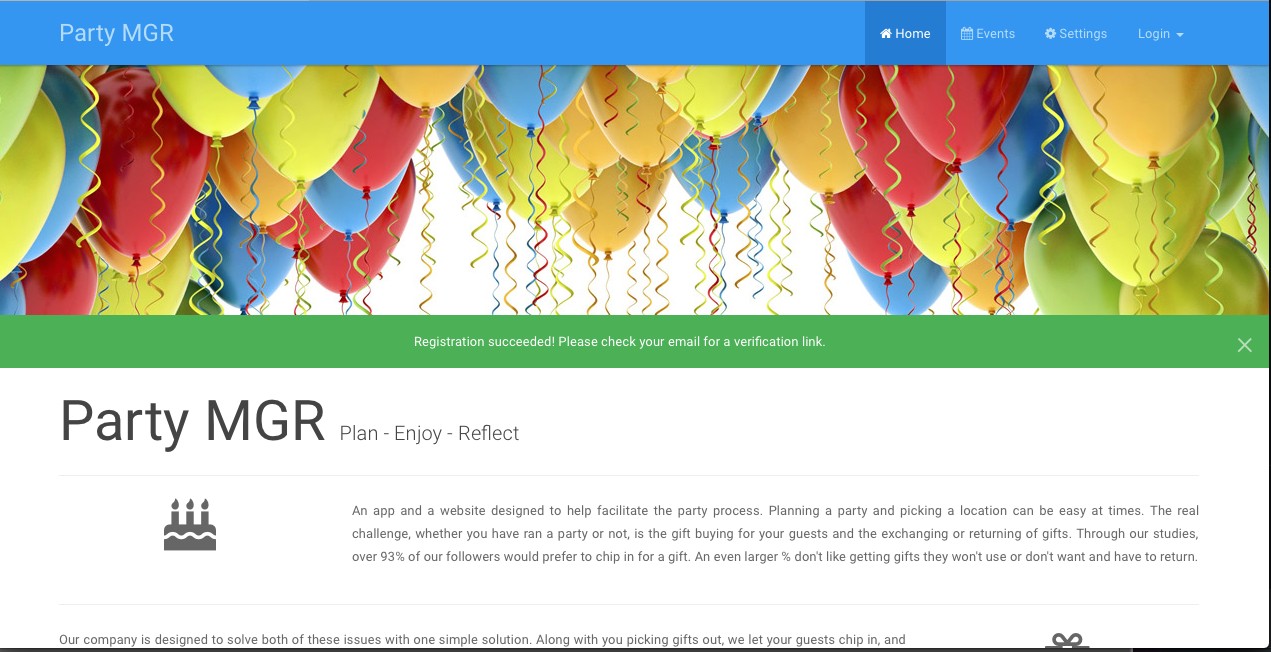


**During**

User presses “Login/Register” button and is taken to the registration page.

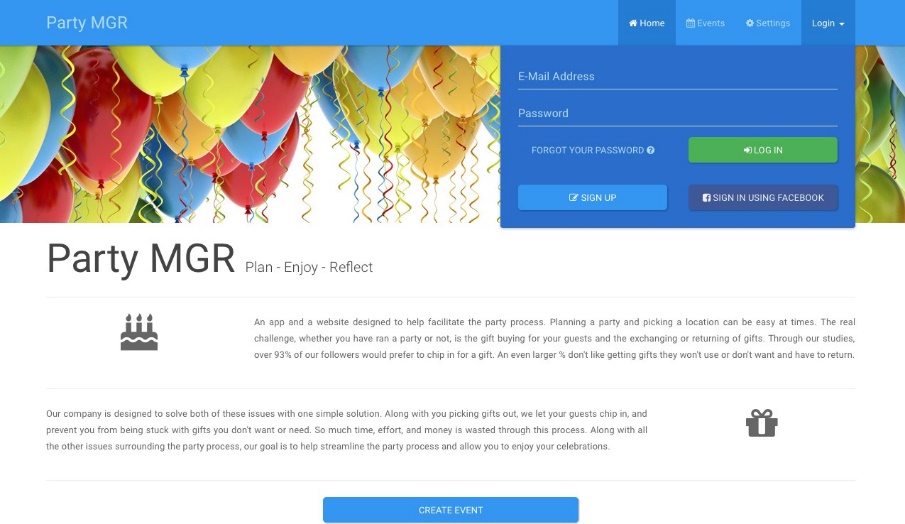


### After



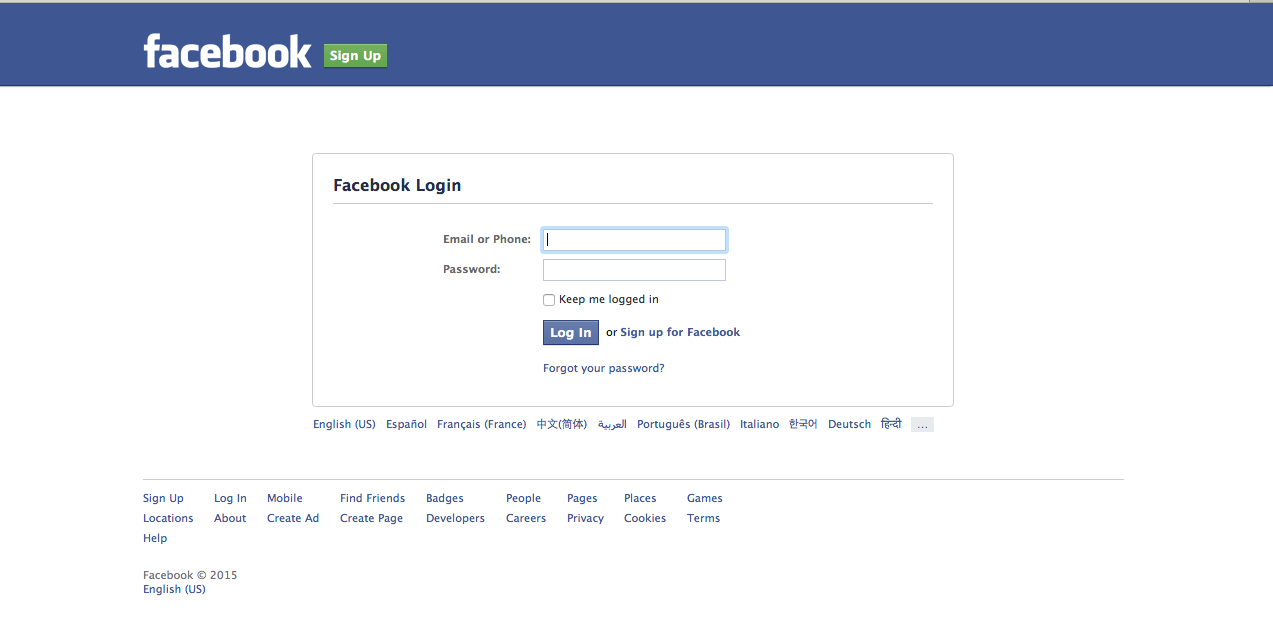
**Before**

**User Creates An Account Using Facebook**

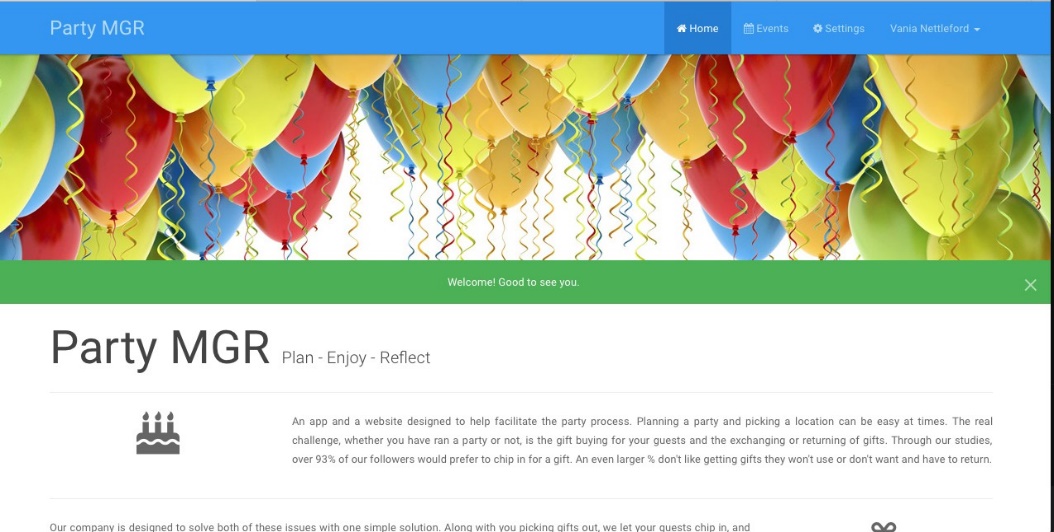


**During**

User presses “Create an Account with Facebook” and is redirected to Facebook for authentication. The system retrieves user’s name and email upon authentication with Facebook. After successful authentication the user will be redirected to the home page.

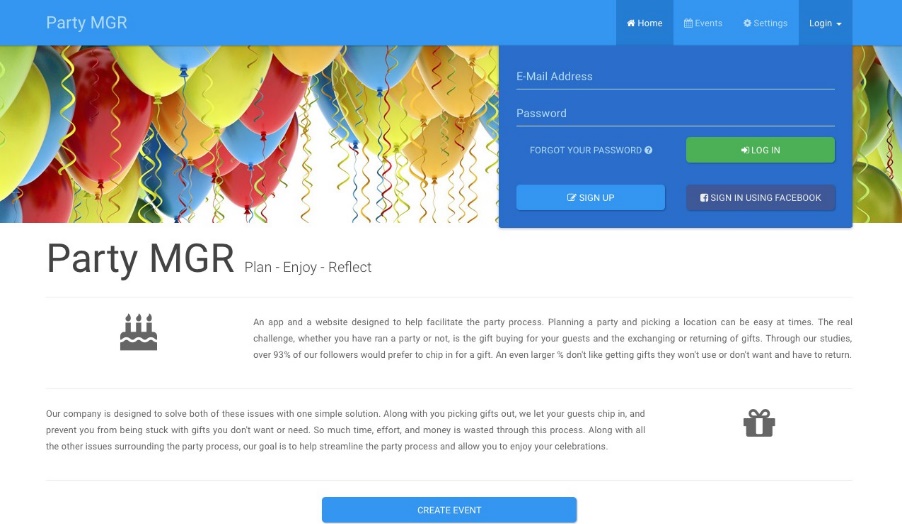


### After

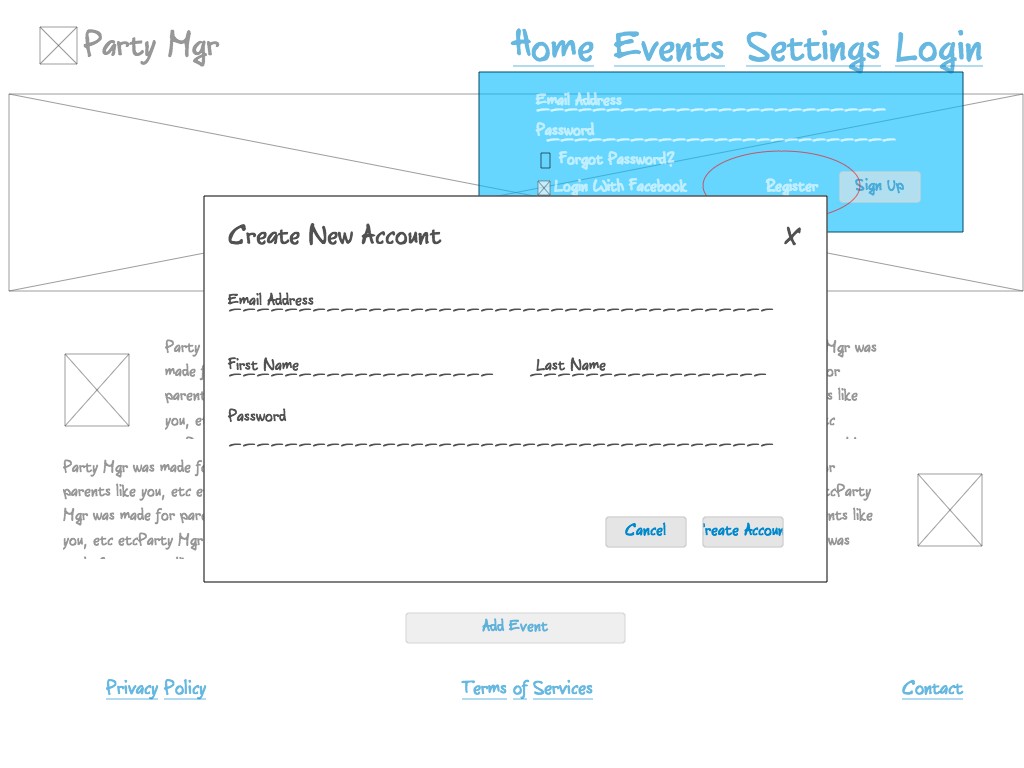


**Before**

**User Creates An Account Using Party Mgr**

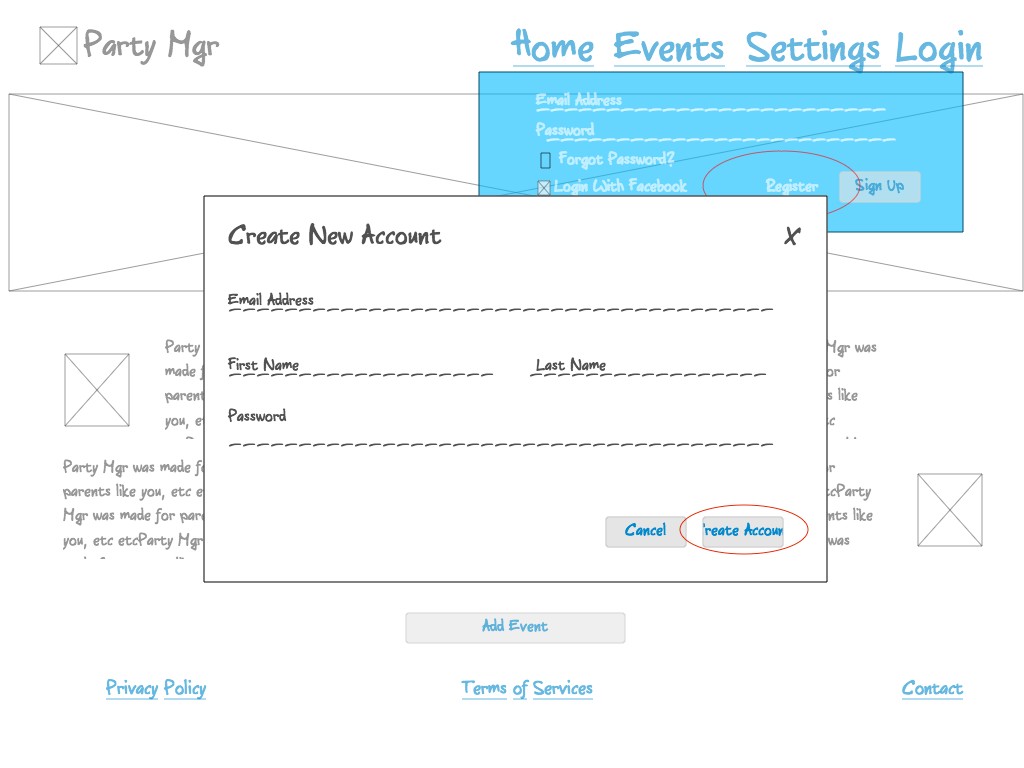


**During**

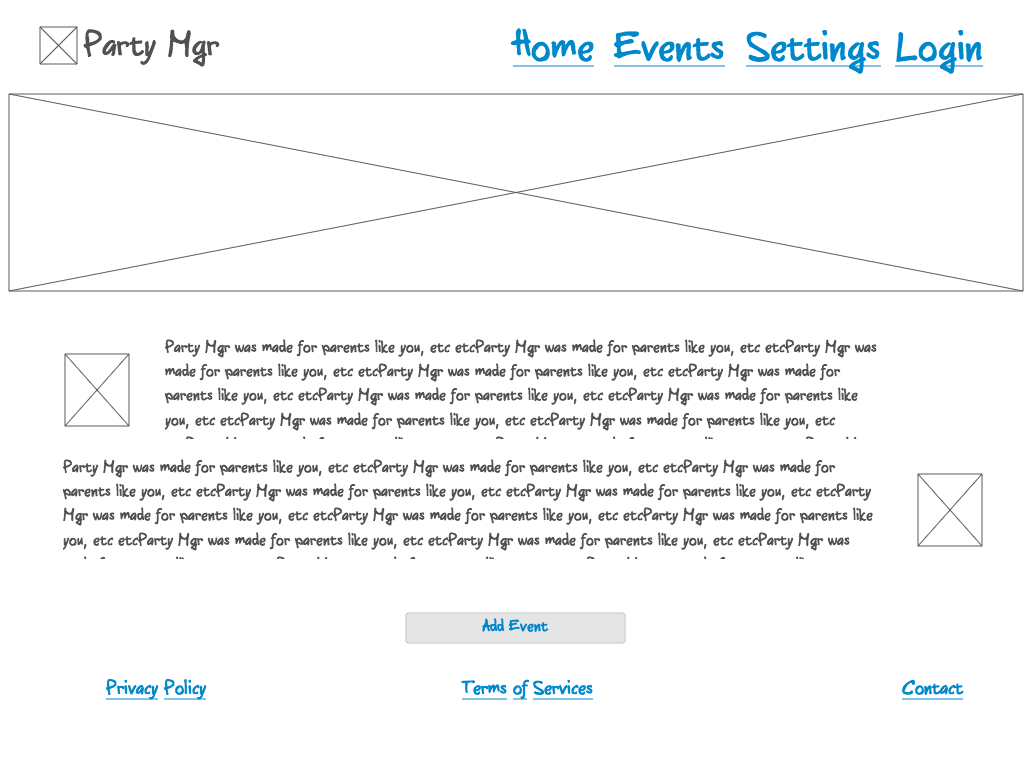


**After**

User presses “Create Account” button, system rejects if not all the fields are filled in. If all fields are filled in, system will create a user account in the database. User will then be redirected to the home page.

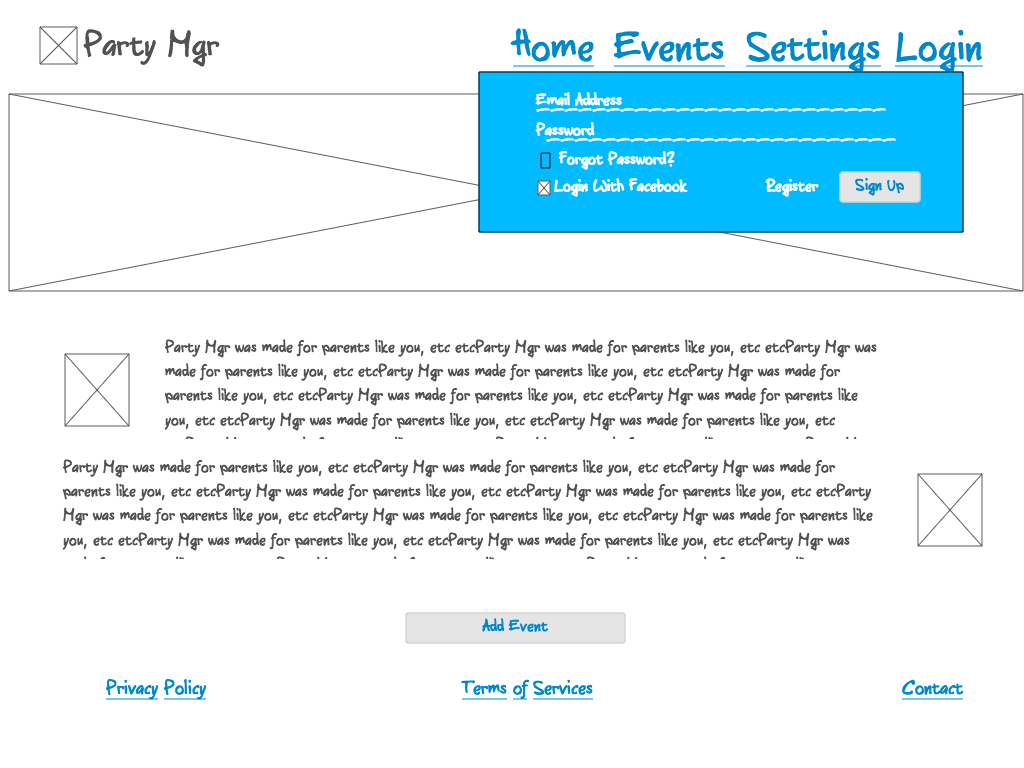


Upon successful creation of the account the user is brought to the home page.



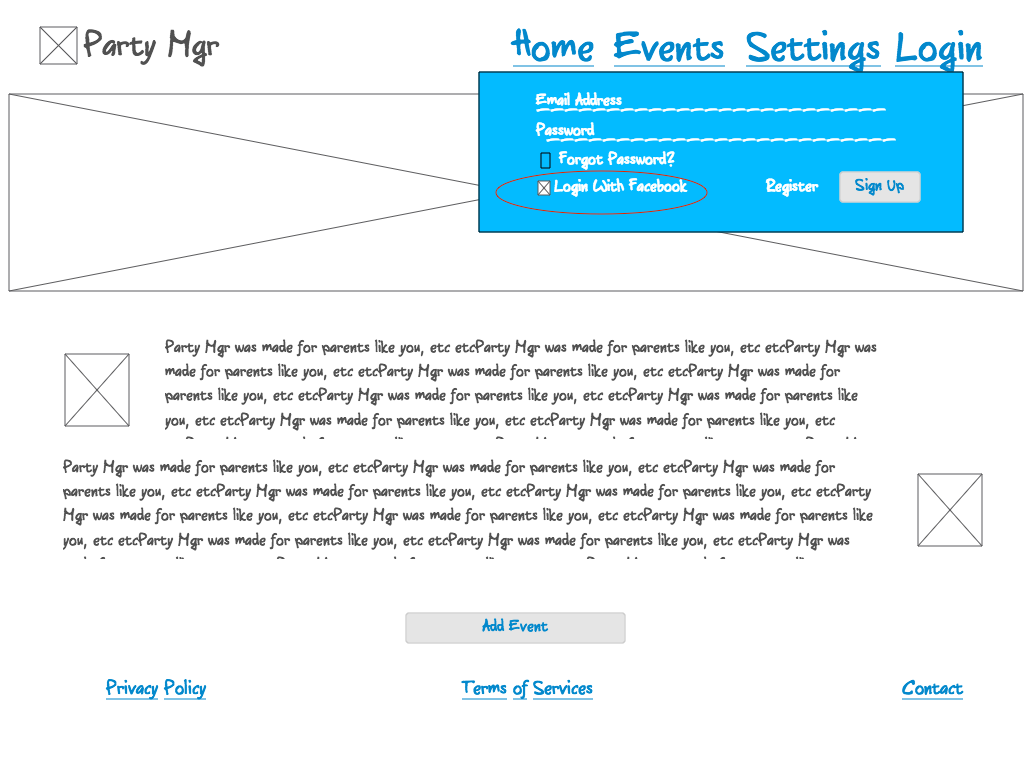
### Before

**User Logs In Using Facebook**



**During**

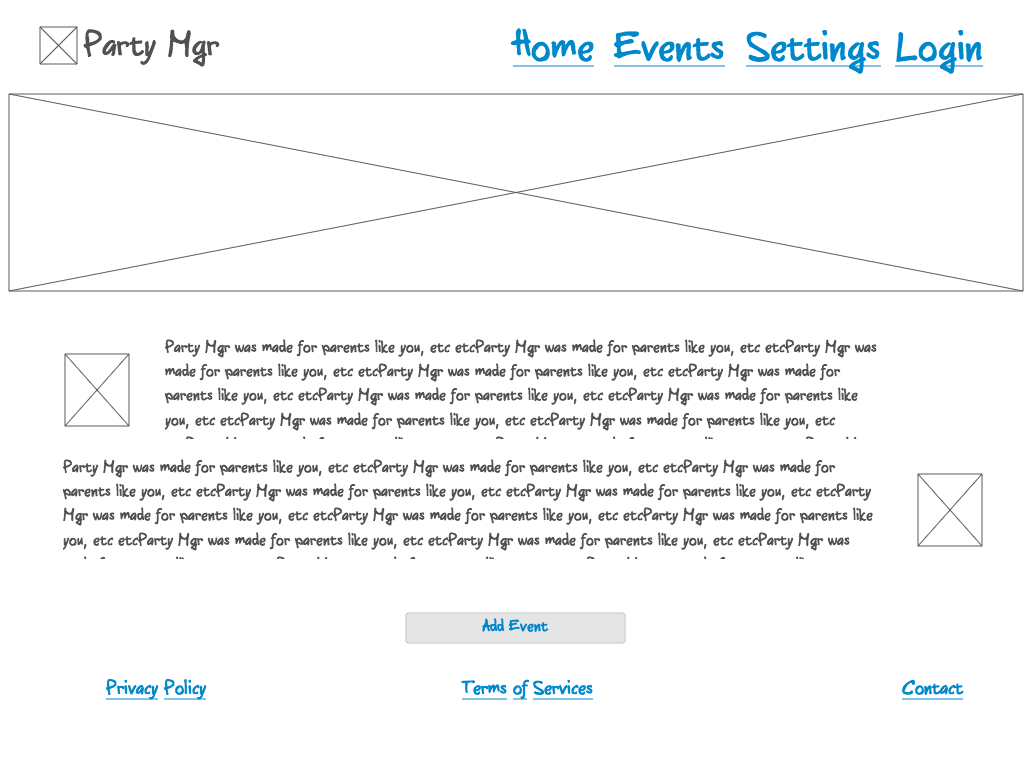
User clicks “Login With Facebook”



### After

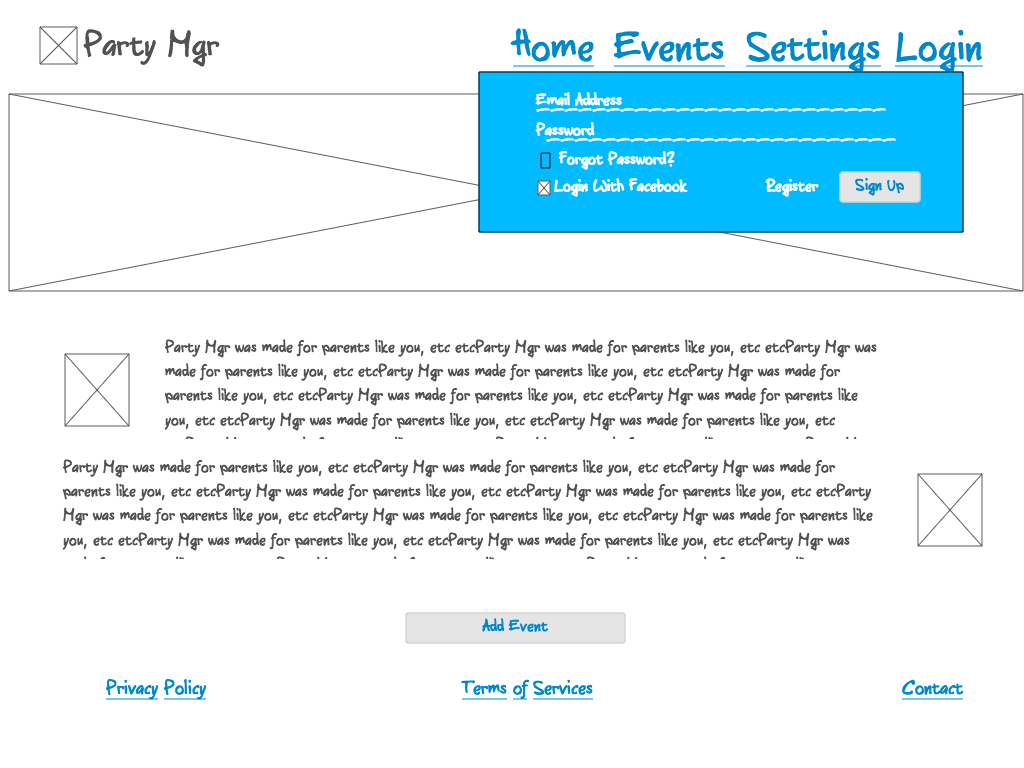


Upon successful creation of the account the user is brought to the home page.



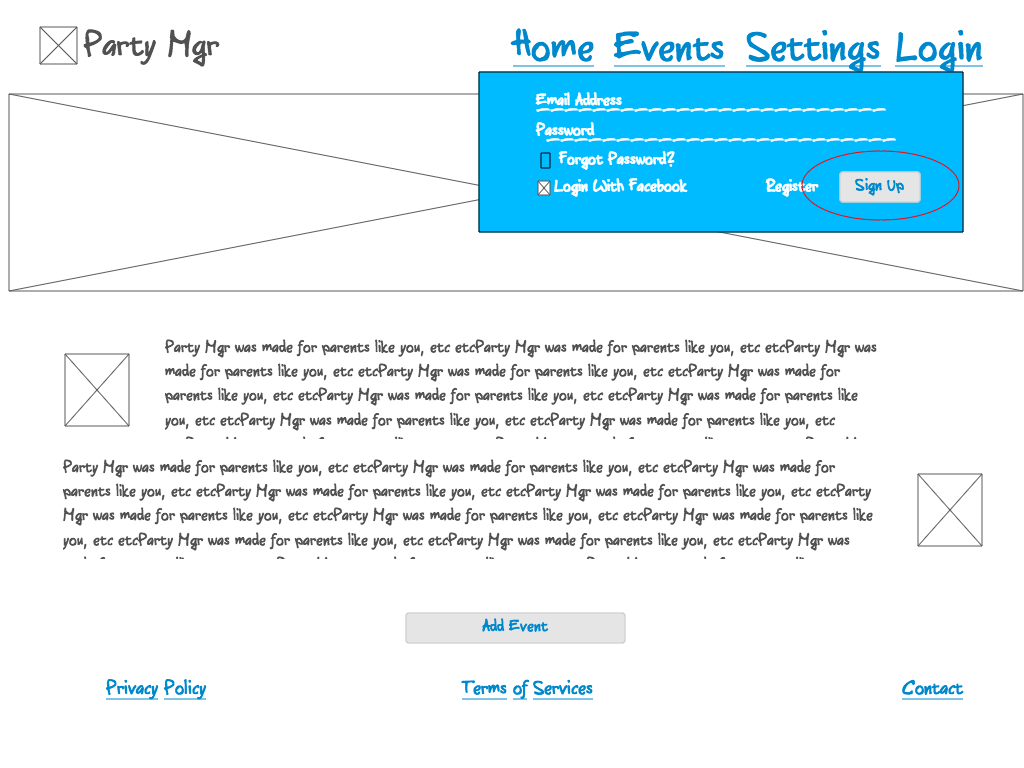
**User Logs In Using Email**

### Before



**During**

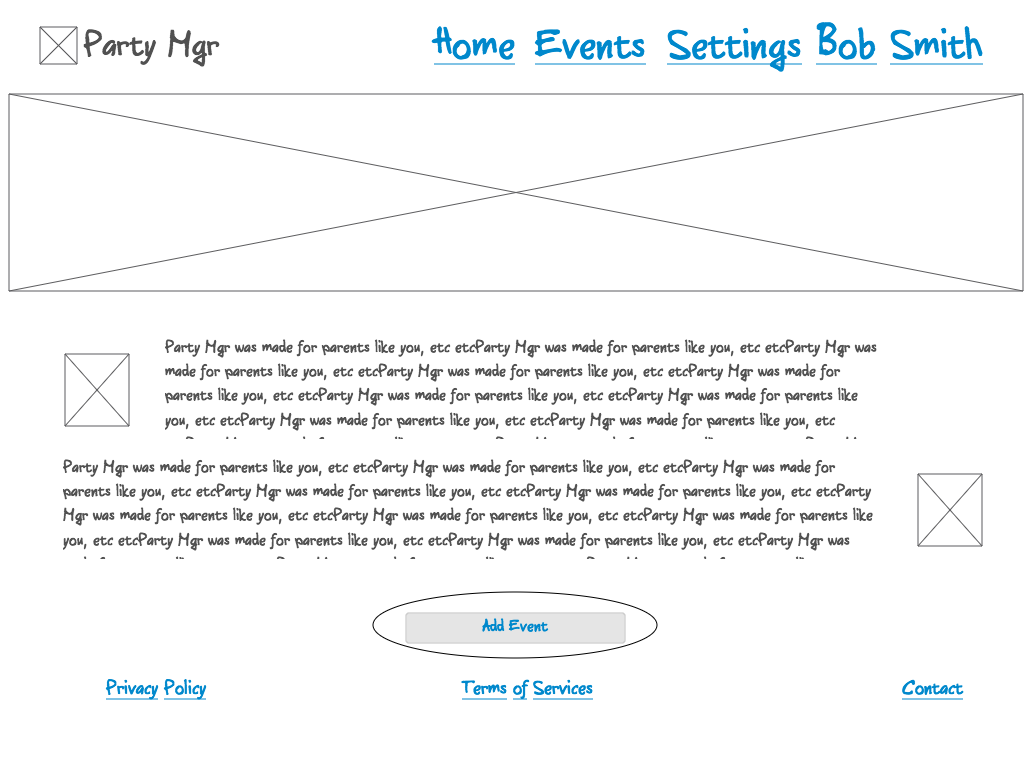
User presses “Login with Email” button and is taken to login prompt.



### After

* User presses “Sign in” button. System checks if email/password combination is valid. If valid, system brings user to home page. If invalid, system shows an error to the user.
* User checks “Remember me” button, browser will automatically sign the user into the site in the future.

Upon successful login user enters into the main site.



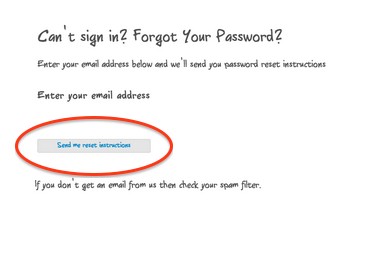
### Before

**User Forgets Password**

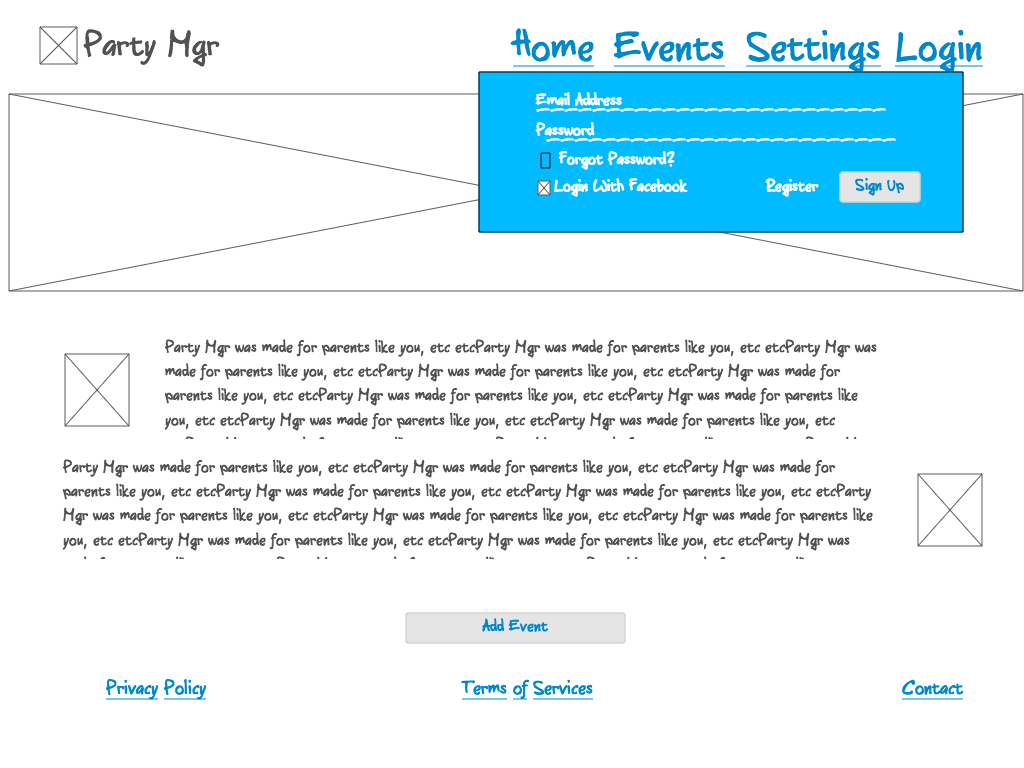


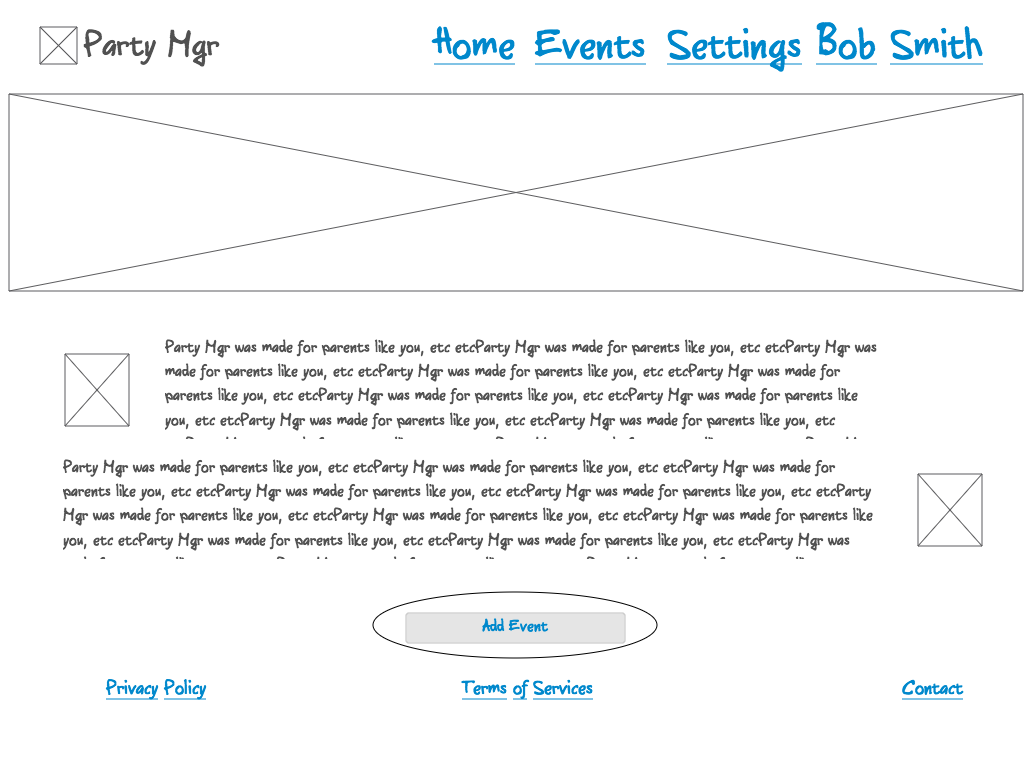
**During**

User presses “Forgot Password?” button. System sends user an email which allows them to reset their password.



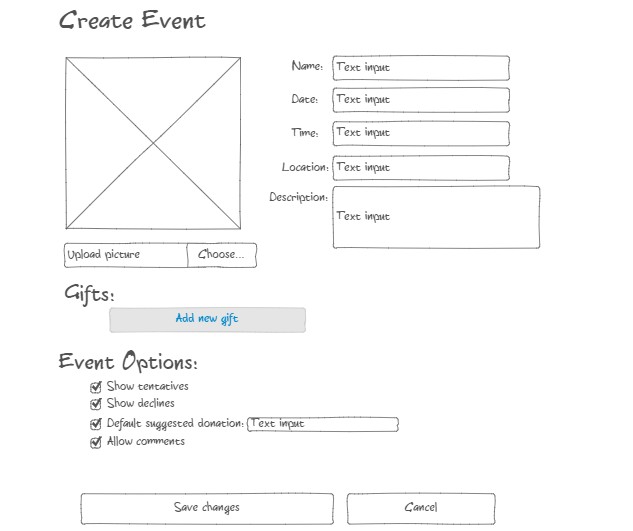
### After



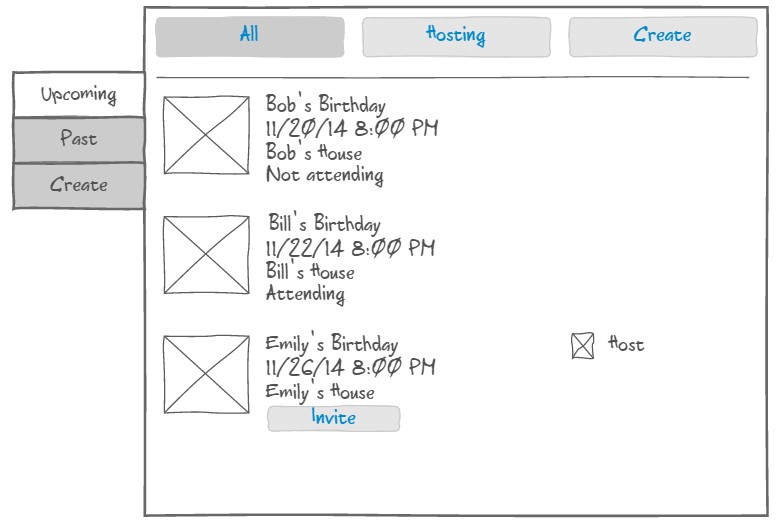
Before:

During:

# User Creates Event



After:



# Event Detail:

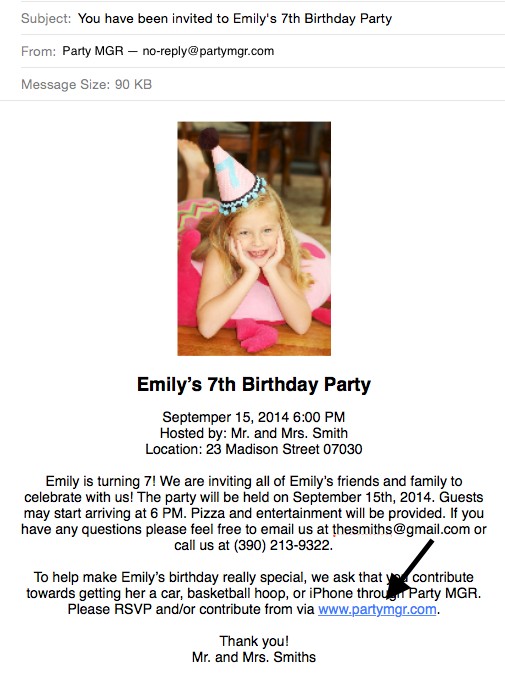
1. User visits event detail page from email invitation
2. User clicks on “Attending” as their RSVP status
3. User clicks on “Maybe Attending” as their RSVP status
4. User clicks on “Not Attending” as their RSVP status
5. User enters a contribution amount
6. User hits contribute button
7. User leaves a comment

### Before

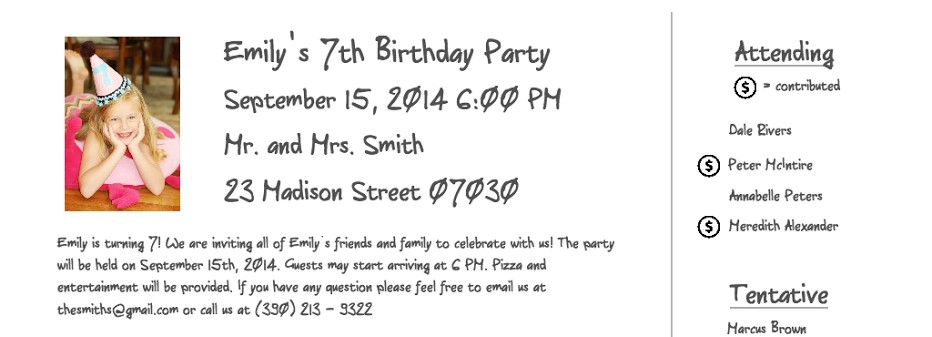
**User visits event detail page from email invitation**



**During**

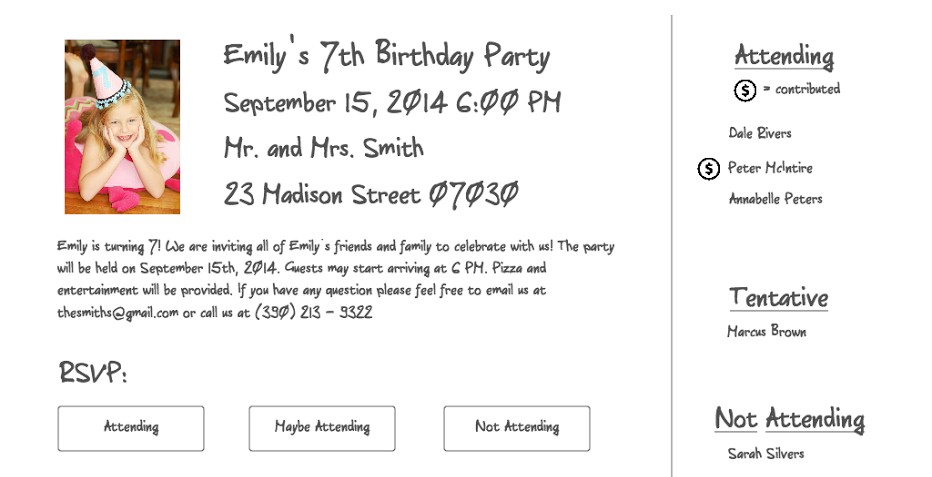


**After**



**User clicks on “Attending” as their RSVP status**

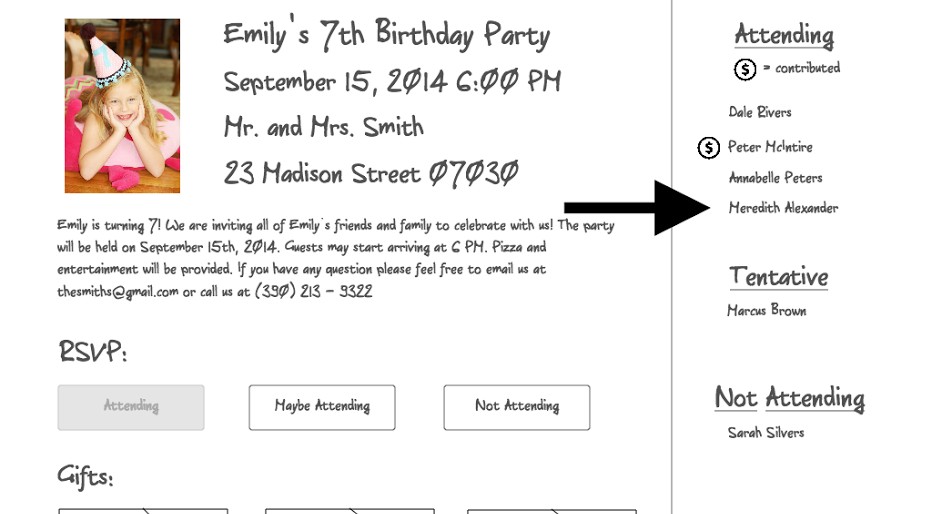
**Before**



**During**

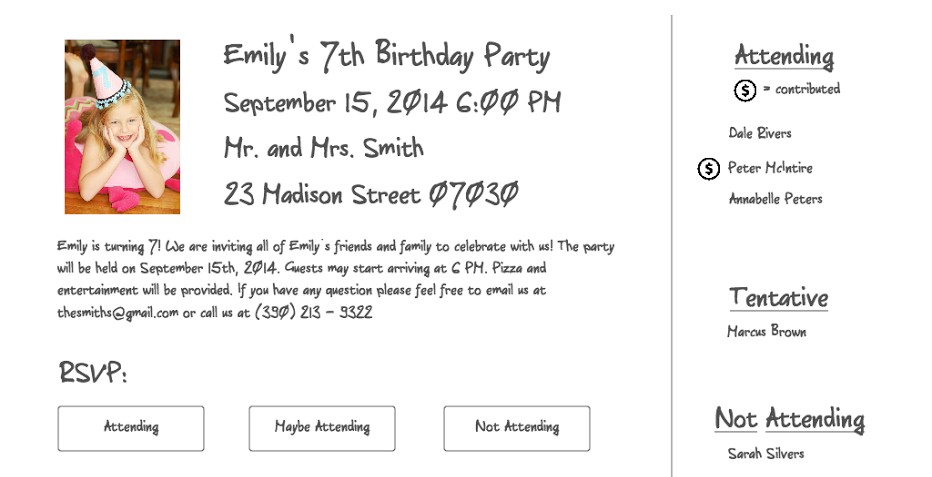


**After**

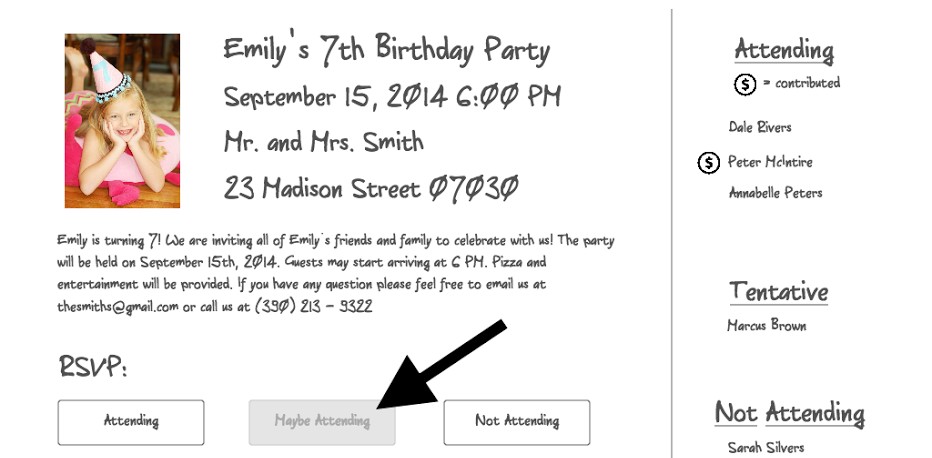


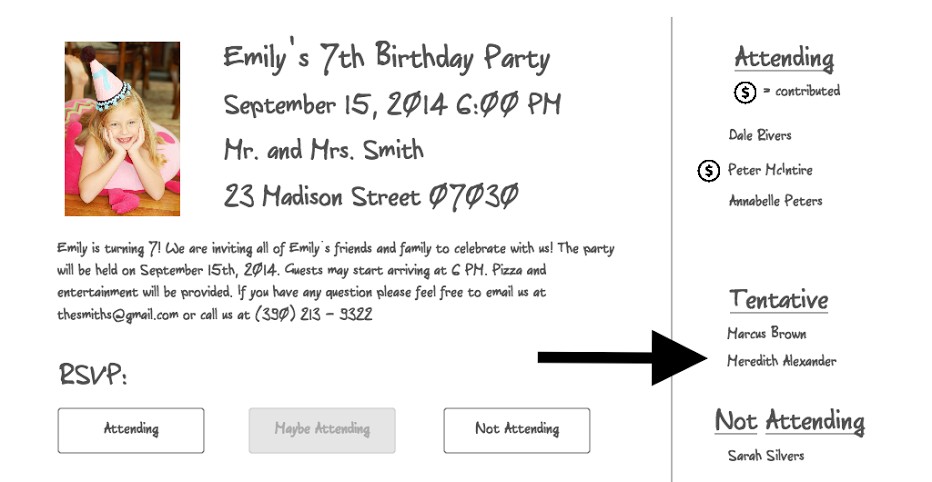
**User clicks on “Maybe Attending” as their RSVP status**

**Before**



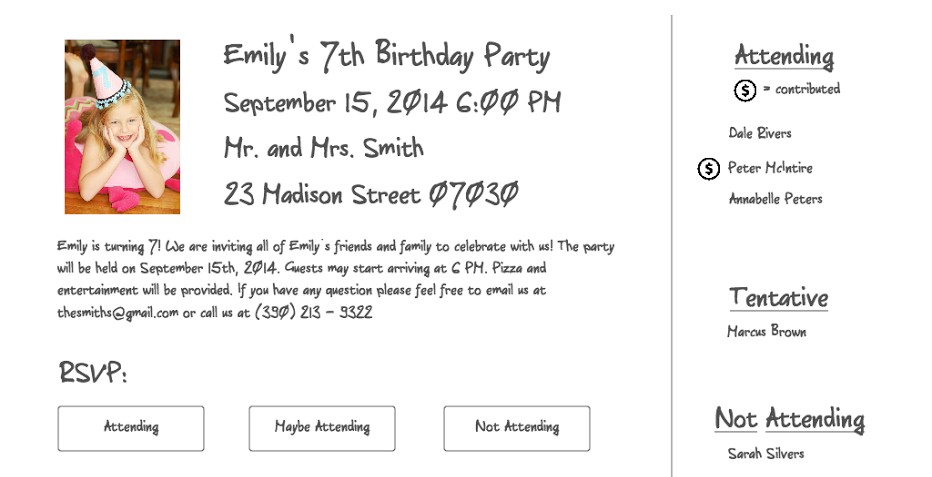
**During**



**After**

**User clicks on “Not Attending” as their RSVP status**

**Before**



**During**



**After**



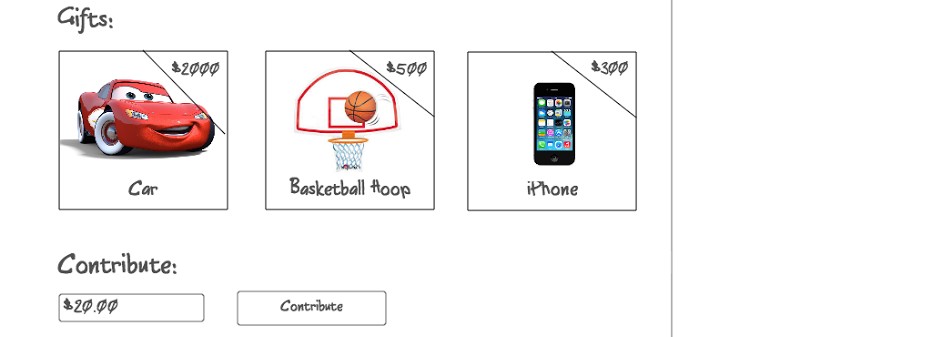
**User enters a contribution amount**

Before:



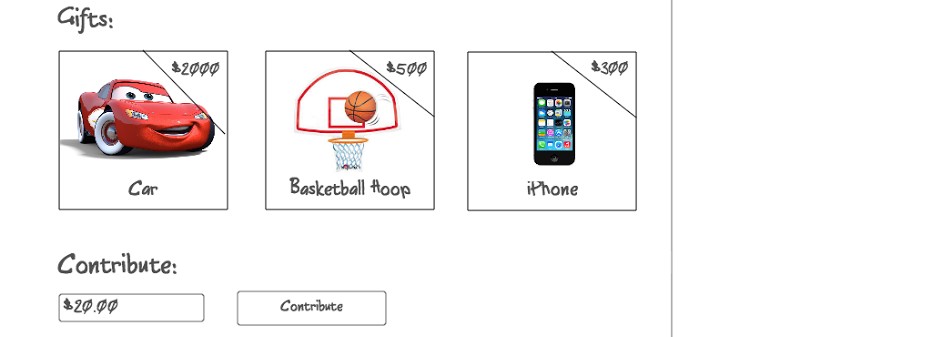
### During



**After**

**User hits contribute button**

**Before**

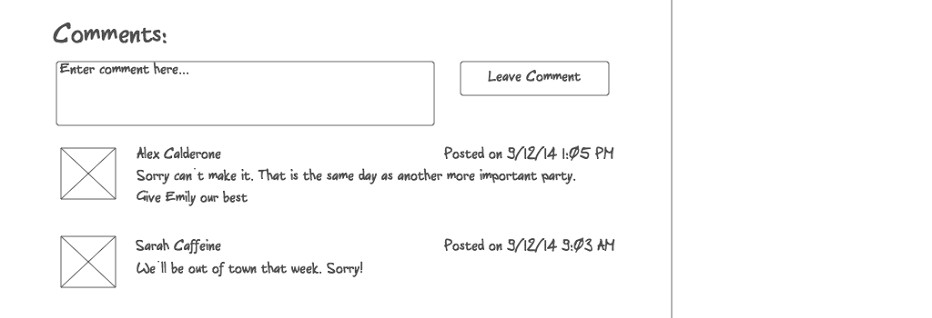


**During & After**

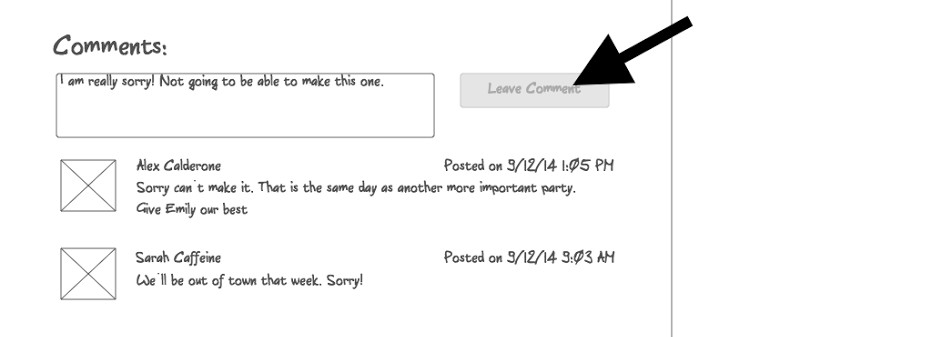
System redirects to 3rd party payment system which processes and confirms purchase. Likely Stripe or Paypal.

**User leaves a comment**

### Before



**During**



**After**



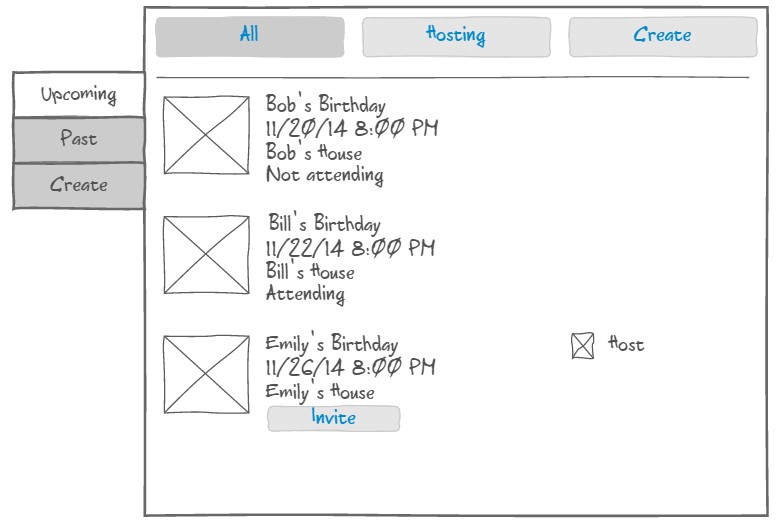
# Event Overview:

* 1. All ­ Upcoming
     1. Invite
  2. All ­ Past (not shown)
  3. Hosting ­ Upcoming
     1. Invite
  4. Hosting ­ Past
  5. Create Event
     1. Add Gift
        1. Save Gift
  6. Save Event

### Before

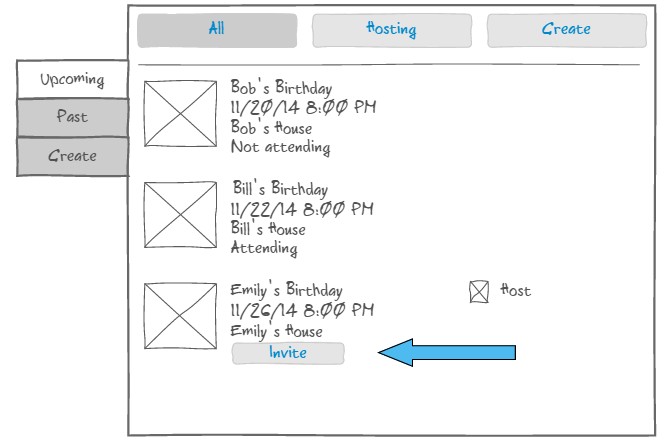
**All ­ Upcoming**

User is on “Upcoming” tab on the left, and “All” tab on the top. Website shows a list of events that the user is attending in the future, as well as events that the user is hosting.



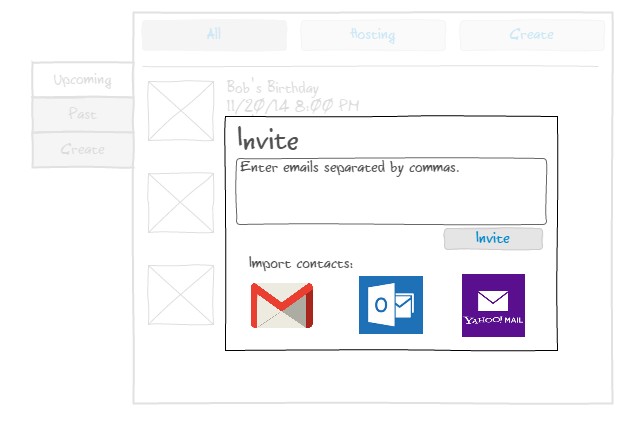
### During

User clicks the invite button on an event he or she is hosting.



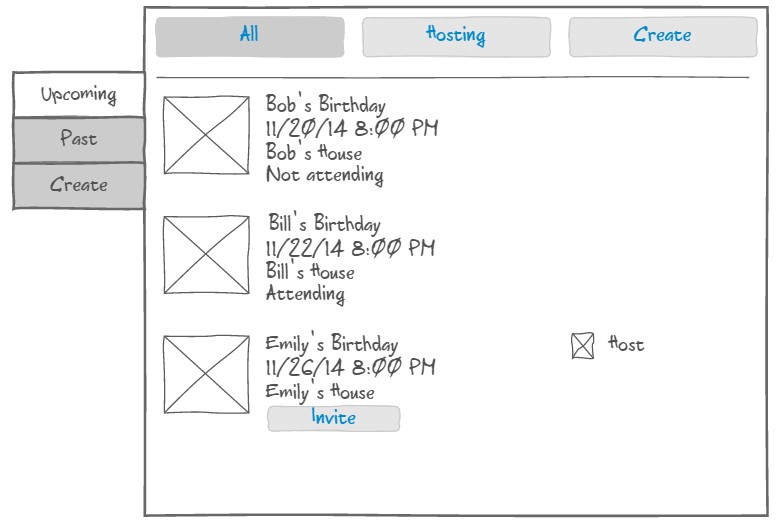
### After

Website displays a modal with invite options. The user can enter emails to invite, or import contacts.



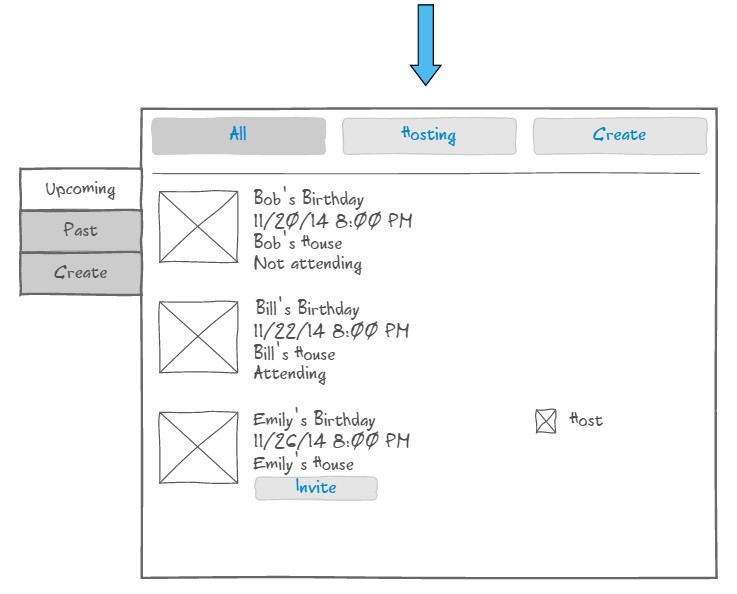
### Before

**Hosting ­ Upcoming**



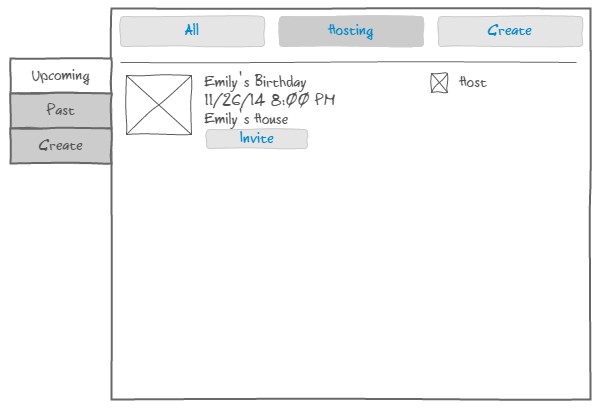
**During**

The user clicks the Hosting Button.



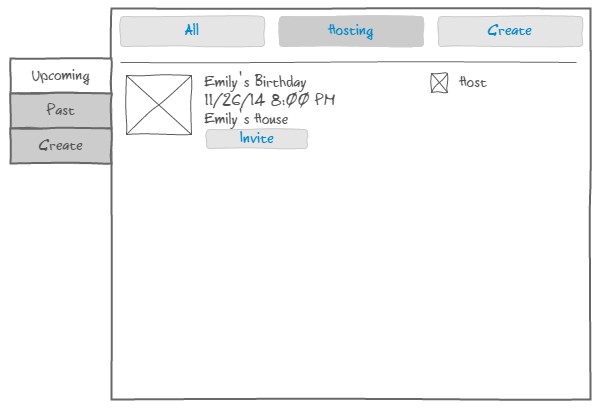
### After

The webpage displays only events that the user is the host for.



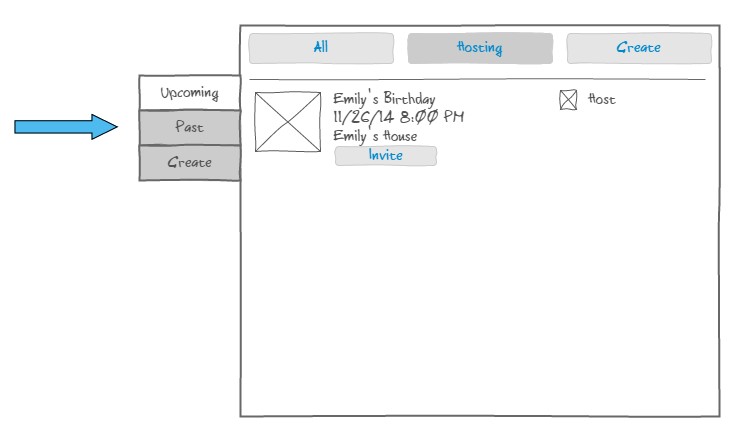
### Before

**Hosting ­ Past**

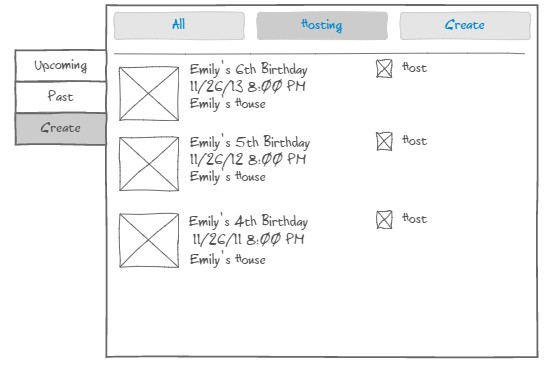


**During**

User clicks the “Past” tab

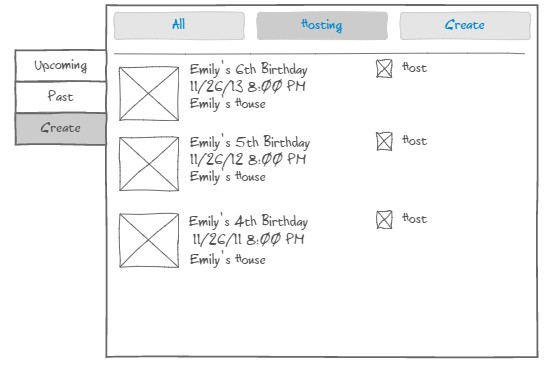


### After



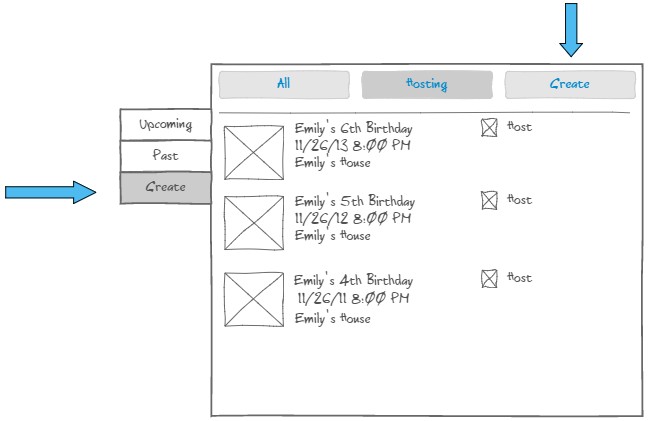
**Before**

**Create Event**



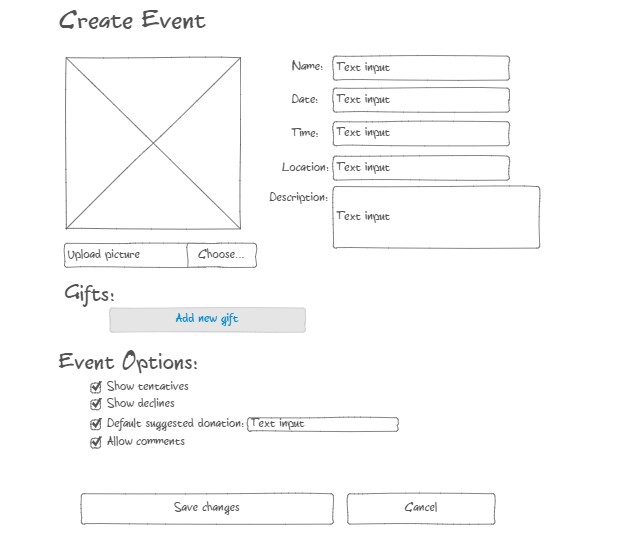
**During**

User clicks the create button on the left, or the create button on the top.



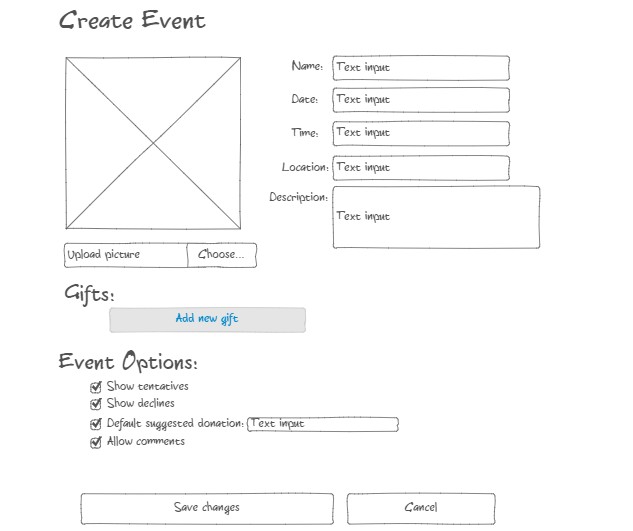
### After

User is directed to a “Create Event” Page.



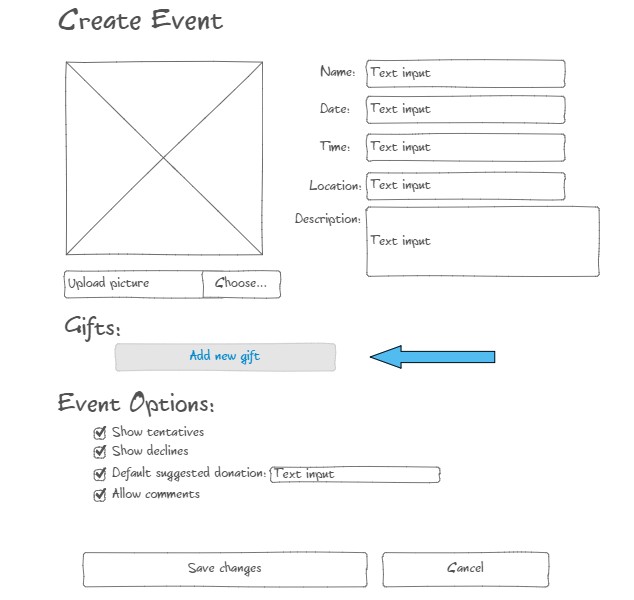
### Before

**Create Gift**



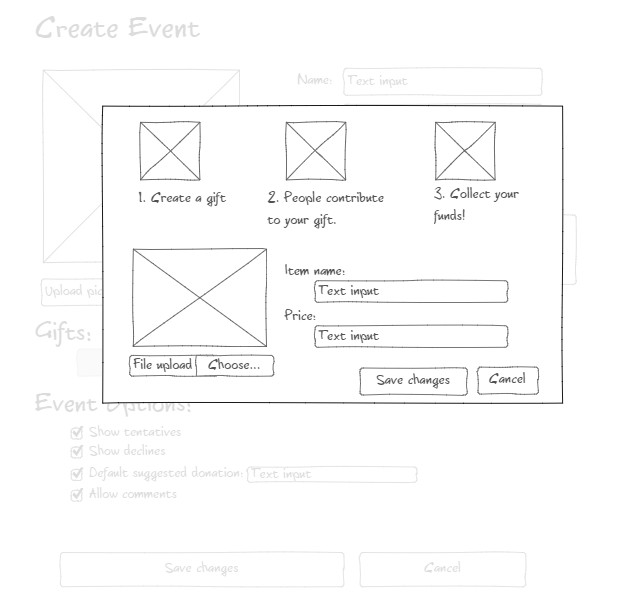
**During**

User clicks the “Add new gift” button.



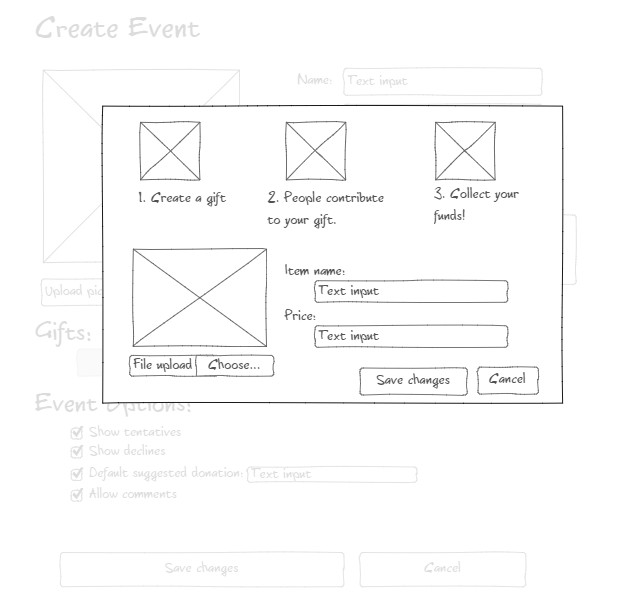
### After

Webpage displays a modal that gives the user options for adding gifts.



### Before

**Saving Gift**



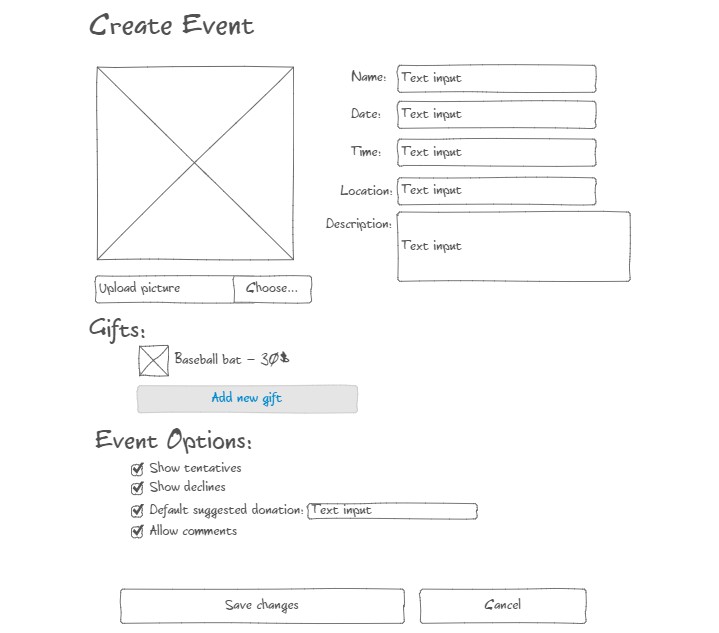
**During**

User clicks the Save changes button.



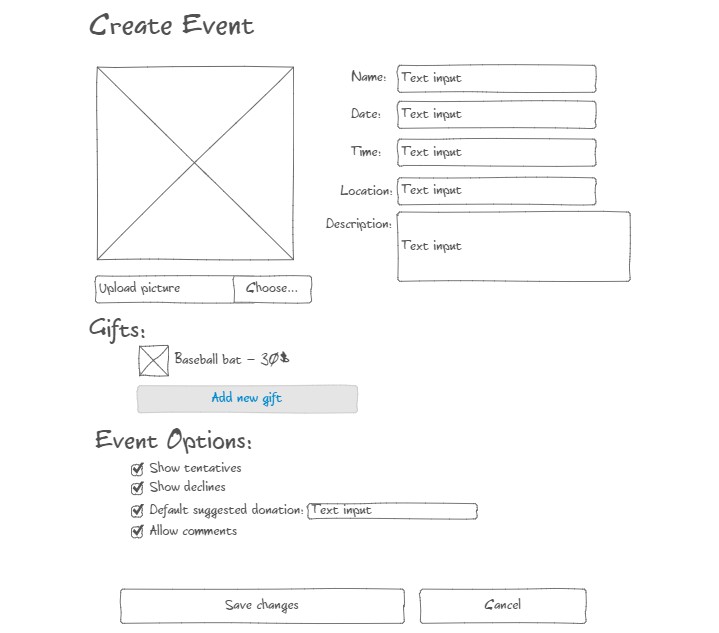
### After

The webpage adds the gift to the list, and closes the modal.



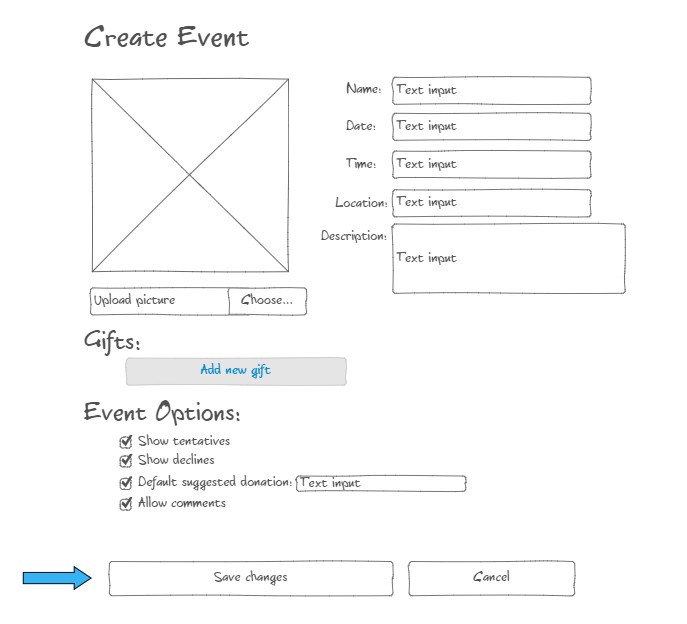
### Before

**Save Event**

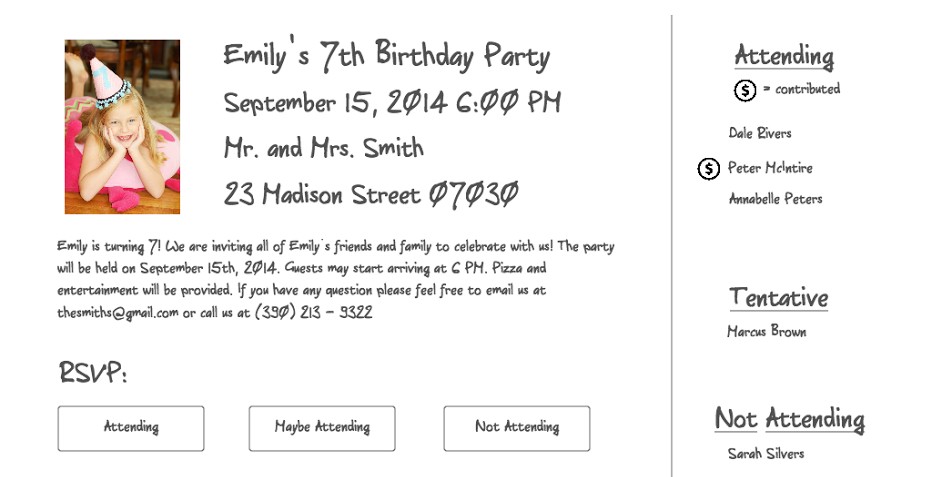


**During**

The user clicks the save changes button on the Create Event page.



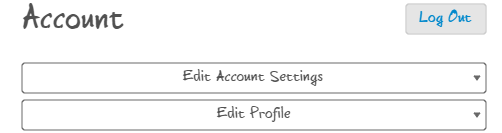
### After



**Settings:**

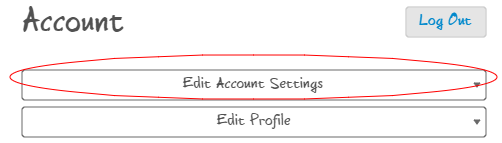
1. User enters Account Page
2. User clicks account settings to expand form
3. User clicks edit profile to expand form
4. User edits and saves changes on account settings
5. User cancels edits on account settings
6. User edits and saves changes on profile
7. User cancels edits on profile
8. User logs out
9. User cancels logout

**User enters Account Page**



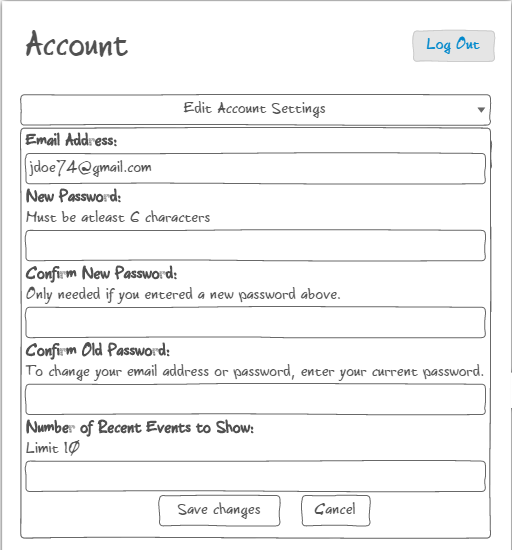
### Before

User clicks account settings to expand form



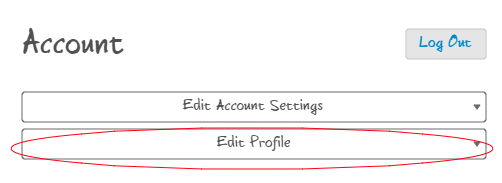
### After

With already saved settings



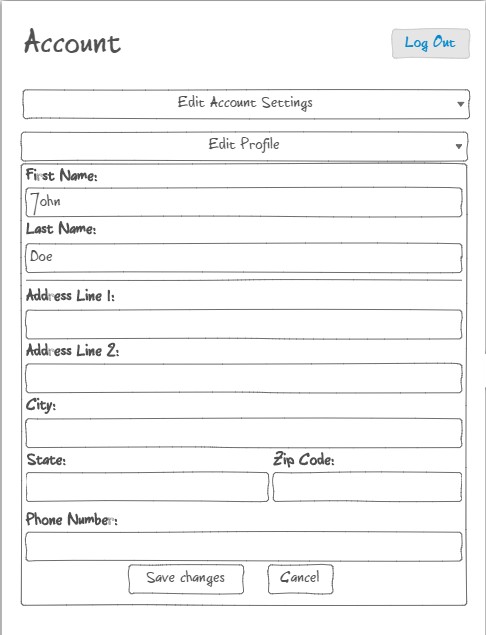
### Before

**User clicks edit profile to expand form**



**After**

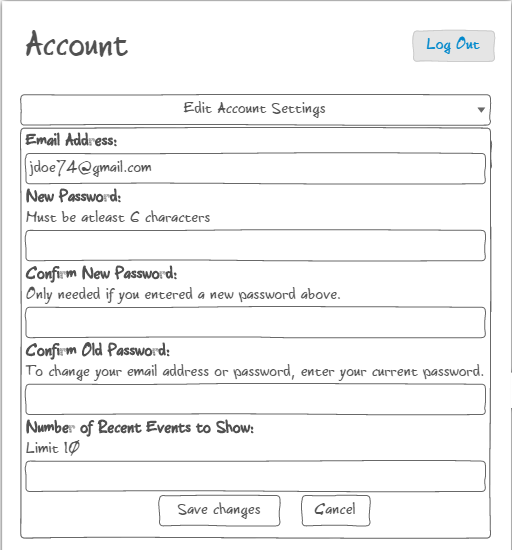
With already saved information



## User edits and saves changes on account settings

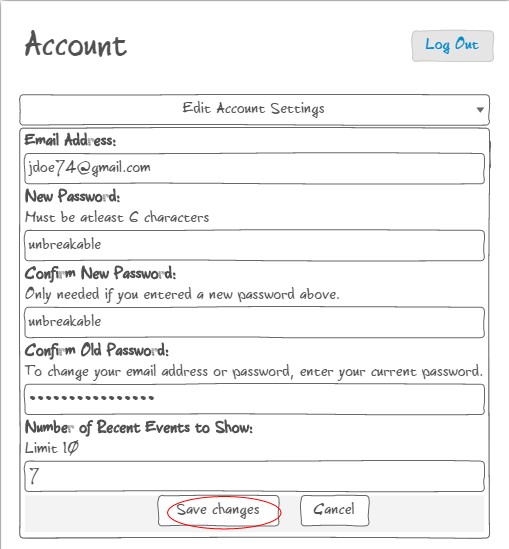
### Before

Form with already saved information



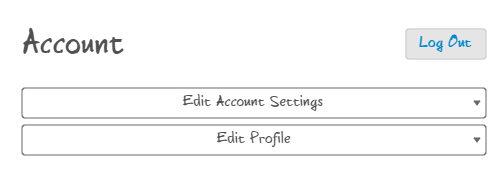
### During

User edits form and clicks save changes



### After

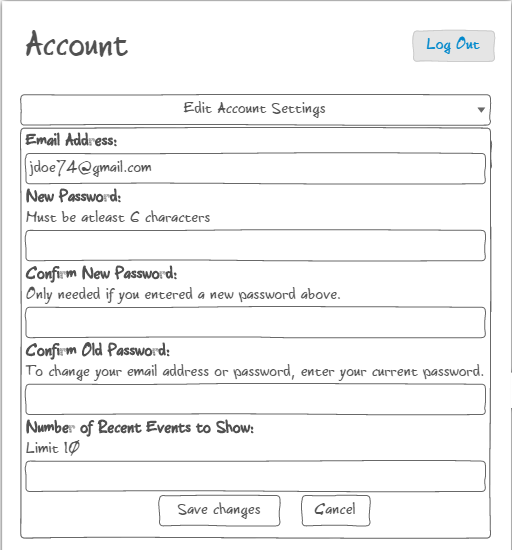
Account settings collapses



### Before

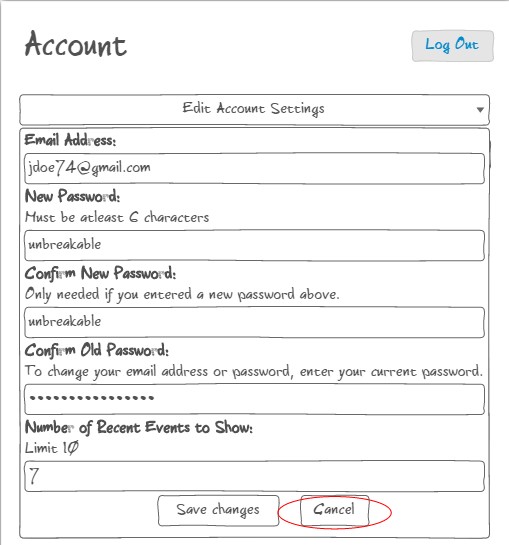
**User cancels edits on account settings**

With already saved information



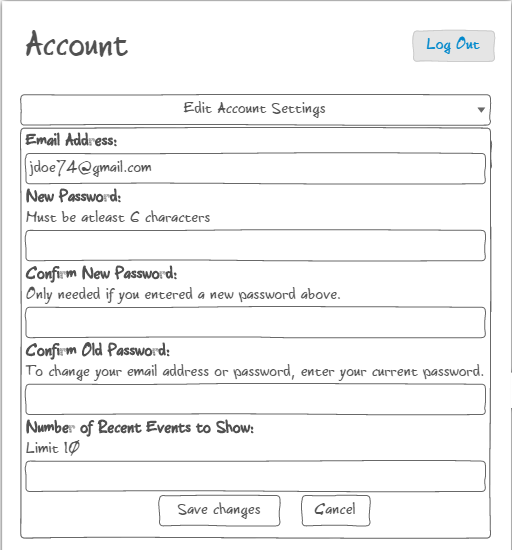
### During

User edits form and clicks cancel



### After

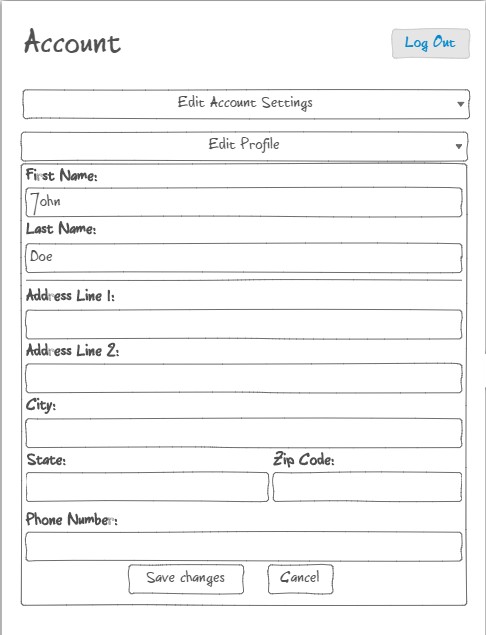
With original saved information



## User edits and saves changes on profile

### Before

With already saved information



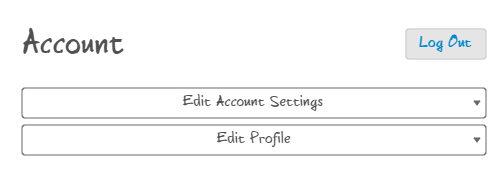
### During

User edits form and clicks save changes



### After

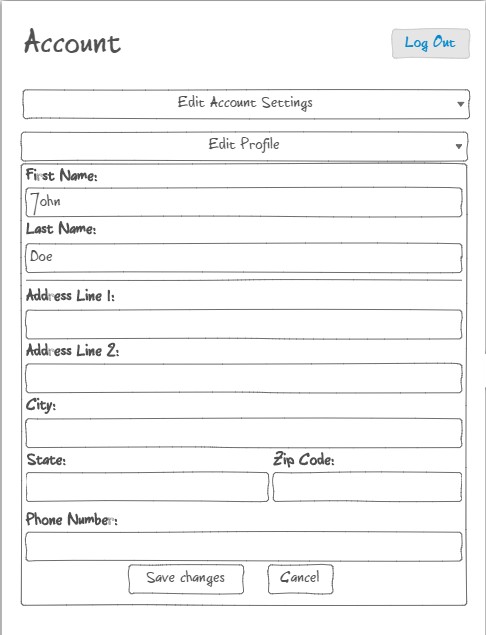
With collapsed profile



### Before

**User cancels edits on profile**

With already saved information

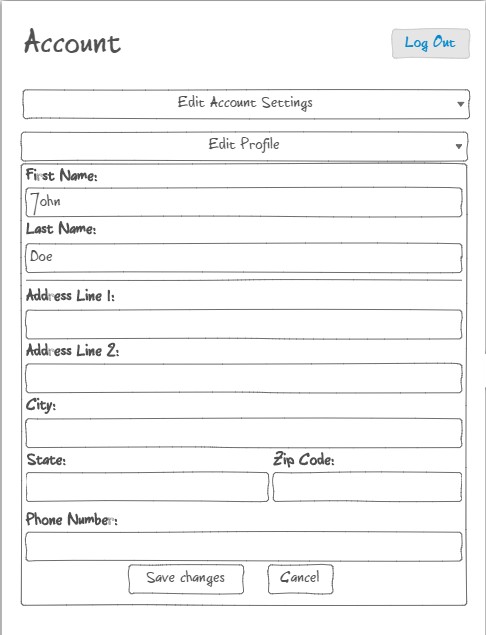


### During

User edits the form and clicks cancel



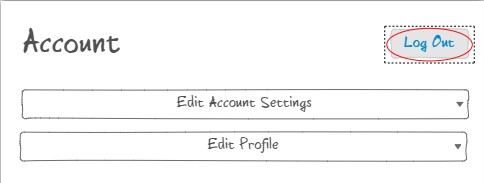
With already saved information



### Before

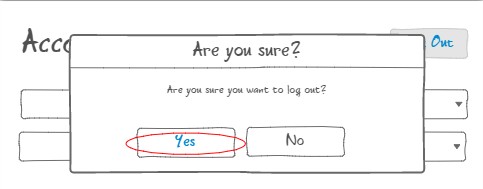
User clicks logout button

## User logs out



### During

User clicks the yes button confirming log out

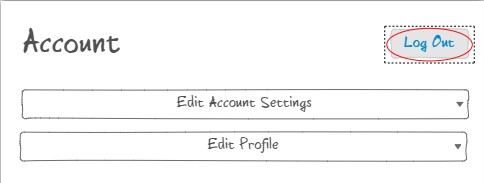


User returned to login/homepage

### Before

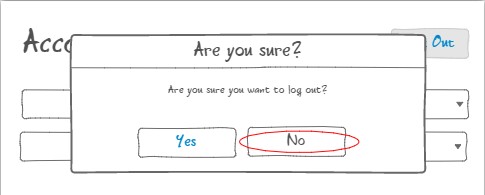
User clicks logout button

## User cancels logout



### During

User clicks the no button canceling log out



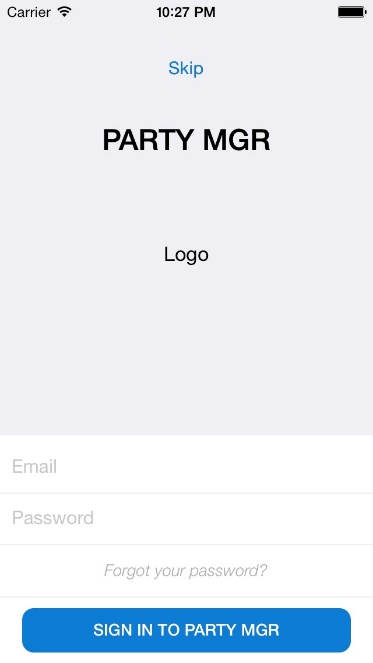
### After

Dialog closes and user returned to account page

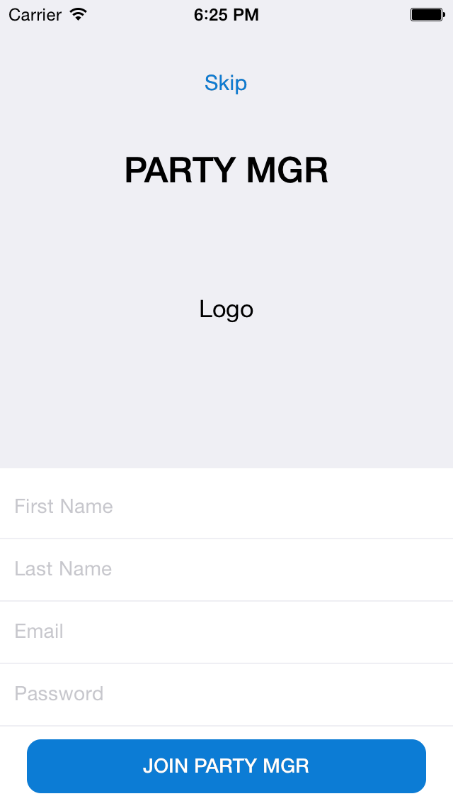


# iOS Mockups

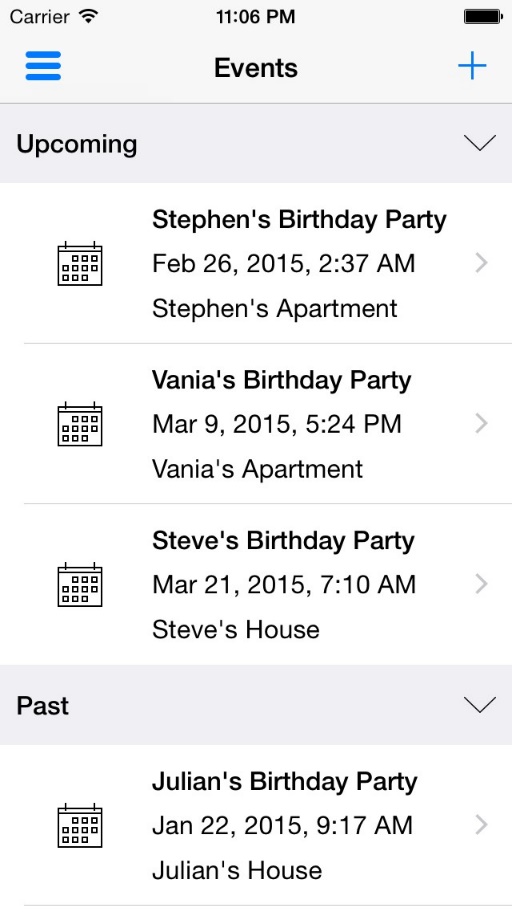
Login Screen



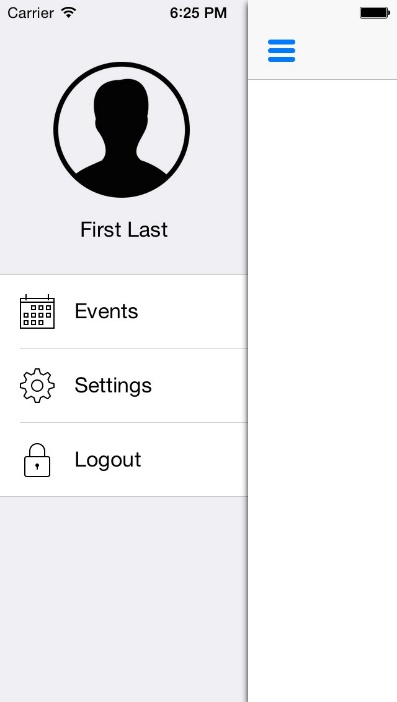
Sign Up Screen



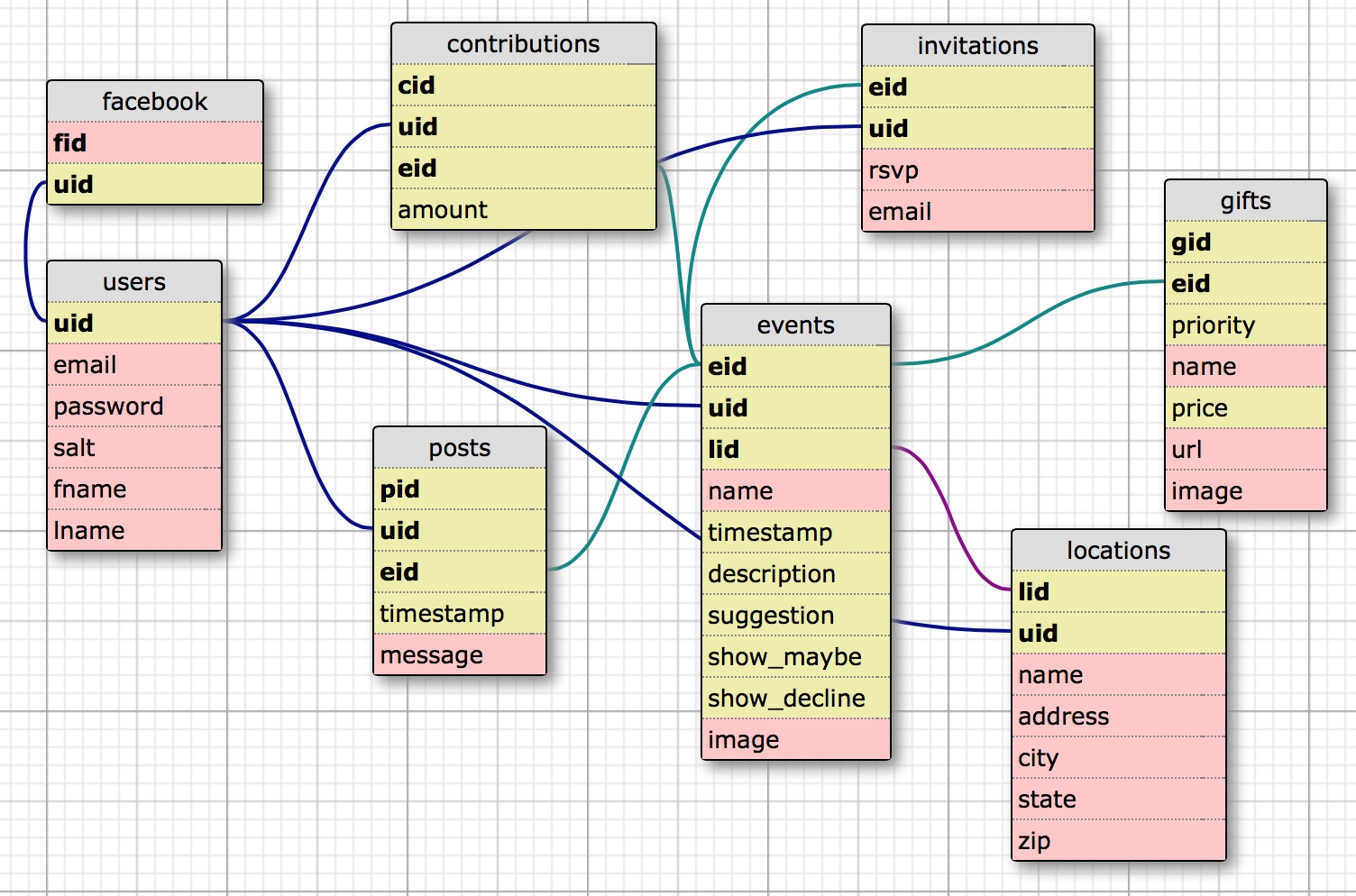
Event Overview Page



Menu/Navigation Tab



### DB Design



contributions(*cid, uid, eid*, amount)

* + cid: contribution id, unique for each entry, auto incrementing
  + uid: foreign key, single user has many contributions, contributions by single user
  + eid: foreign key, single event has many contributions, contributions for a single event
  + amount: decimal money value contributed

events(*eid, uid*, *lid*, name, timestamp, description, show\_maybe, show\_decline, image)

* + eid: event id, unique for each entry, auto incrementing
  + uid: foreign key, single user has many events, events owned by single user
  + lid: foreign key, single event has a single location
  + name: name of the event supplied by user
  + timestamp: time event is to occur, stored as unix timestamp
  + description: description of the event supplied by the user
  + show\_maybe: preference of showing maybe responses
  + show\_decline: preference of showing decline responses
  + image: image provided by the user gifts(*gid, eid*, priority, name, price, url, image)
  + gid: gift id, unique for each entry, auto incrementing
  + eid: foreign key, events can have many gifts, each gift associated with one event
  + priority: integer value associated with how important a gift is
  + name: name of gift supplied by user
  + price: price of gift required
  + url: external url of gift for linking exactly what they plan to buy
  + image: image of gift supplied by user locations(*lid*, *uid*, name, address, city, state, zip)
  + lid: location id, unique for each entry, auto incrementing
  + uid: foreign key, one user can have many locations, each location has one user
  + name: name of location supplied by user
  + address: address of location supplied by user
  + city: city location is in
  + state: state the city is in
  + zip: zip code of city/state posts(*pid, uid, eid*, timestamp, message)
  + pid: post id, unique for each entry, auto incrementing
  + uid: foreign key, each user can have many posts, post has one owner
  + eid: foreign key, each event can have many posts, each post belongs on one event
  + timestamp: unix timestamp for when the message was posted
  + message: text supplied by user for post

users(*uid*, fid, email, password, salt, fname, lname, phone, lid)

* + uid: user id, unique for each entry, auto incrementing
  + fid: facebook id, supplied by facebook API for identification
  + email: email address of user
  + password: using sha512, hash(concatenate(hash(supplied pass), salt)). invitations(*eid, uid,* email, rsvp)
  + eid: foreign key, each invitations is associated with an event
  + uid: foreign key, each invitation is associated with a user
  + email: email address of user
  + rsvp: status of attending, maybe attending, or not attending

### Test Plan Version

Functionality testing will be performed regularly, to ensure that updates to the site did not damage functionality in other parts of the site.

* + - Functionality of every link on each page
    - Functionality of payment methods
    - Functionality of event creation and editing
    - Functionality of email broadcasts
    - Functionality of RSVP
    - Functionality of account creation
    - Functionality of search bars
    - Functionality of login with Facebook/Gmail
    - Functionality of photo upload
    - Functionality of website and database communication
    - Functionality of import from Google, Yahoo, Facebook, and Outlook Security concerns with respect to user input:
    - Prevention of injection via photo upload
    - Input validation in login areas
    - Input validation in search bars and input areas
    - Make sure users cannot spam other people’s emails via button press (set a time limit on email reminders)
    - Ensure that emailed links to events cannot be altered to grant access to unauthorized users
    - Hash passwords using bcrypt for salts, or by using current timestamps, email addresses, and a random number generator.

Much of this testing can and should be automated. It has been decided that since unit testing software is either too general or too costly, unit tests will be created

by the team for different modules of the site. Ideally, these unit tests will not change (except to make them more strict), so that all methods and modules of the site have a standard to which they should conform.

* Unit tests for each method, both Javascript and PHP, including all possible forms of unfavorable input
* Unit tests for input on all form elements, which include special characters and null strings, and invalid or incorrect inputs. (Ideally, this process will be automated if possible, but if not, then test cases should be documented and performed on a regular basis)
* Unit tests for working links between pages