Design

Overview

Motivation

Regular involvement in fitness activities is an integral part of a healthy lifestyle, yet MIT students, on average, do not meet minimum exercise requirements.

The Center for Disease Control recommends that that adults participate in 30 minutes of light-to-moderate exercise five times a week and twice-weekly strengthening activities.[1] According the to Mayo clinic: exercise controls weight, combats health conditions and diseases, improves mood, boosts energy, promotes better sleep, and improves an individual's sex life.[2] Exercise is even more important for a student than the average American. A study by the American College of Sports Medicine found that students who regularly engage in vigorous exercise get better grades.[3]

Yet, the majority of the MIT student body does not meet the recommendations of minimum exercise for a healthy lifestyle. Every year the administration conducts a survey to evaluate the quality of life at MIT. The results indicate that over half of the student body found that they exercise regularly or participated in club or intramural sports less than they would have liked[4] and around 64.2 percent of undergraduate students spend between 0 and 5 hours on physical activities each week.

When interviewed, MIT students pointed out the following barriers to exercise:

- Lack of interesting fitness activities.
    - "I don't just want to run on a treadmill"
- Lack of peer involvement.
    - "If someone doesn't hold me to my commitment to go, I probably won't make it"

Purpose

FitFriends will enable MIT students to connect and be healthy by creating, viewing, and joining opportunities to be involved in fitness activities.

We would like to build FitFriends because we believe that MIT students have a hard time getting a proper amount of exercise. Part of the problem is lack of motivation or lack of knowledge. With FitFriends, we can create a place for MIT students to feel comfortable connecting with friends and peers in order to help them to keep in shape. In doing so

---

[1] To stay healthy, adults aged 19-64 should try to be active daily and should do at least 150 minutes of moderate-intensity aerobic activity over one week and muscle-strengthening activities on two or more days a week. Source: Centers for Disease Control and Prevention's *2008 Physical Activity Guidelines for Americans*

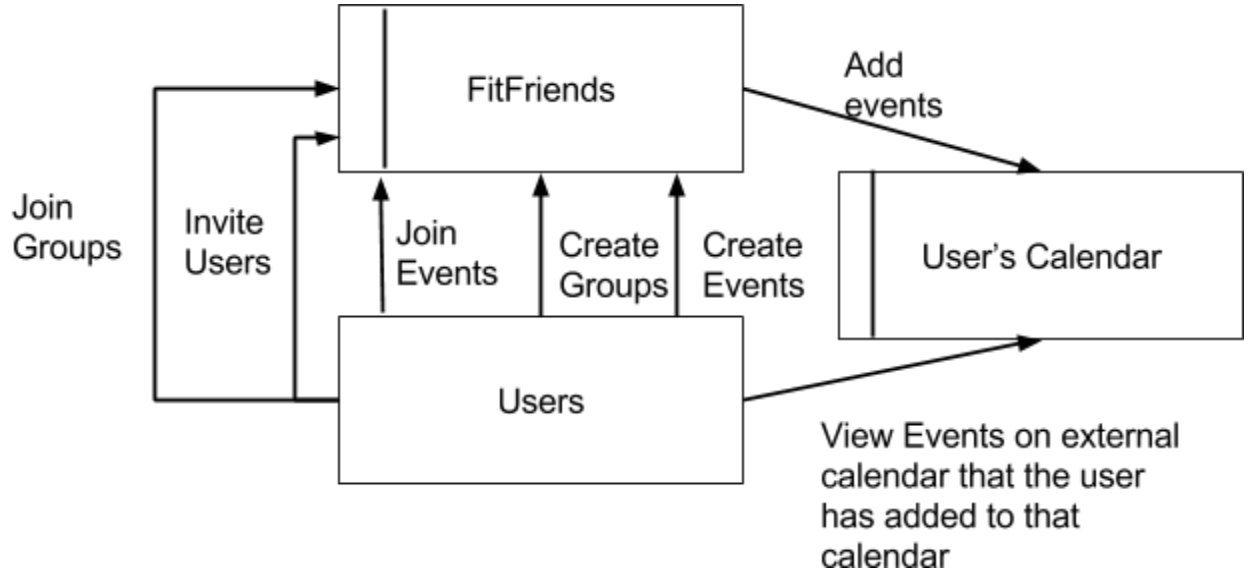
[2] "Money Talks When It Comes to Losing Weight, Mayo Clinic Study Finds." *Mayo Clinic*. N.p., n.d. Web.

[3] American College of Sports Medicine, 2011

[4] http://web.mit.edu/ir/surveys/pdf/2013_SQL_Survey_Highlights_050613.pdf, page3

FitFriends will speak to two purposes: fostering a sense of community and helping MIT students find fun healthy activities.

Context

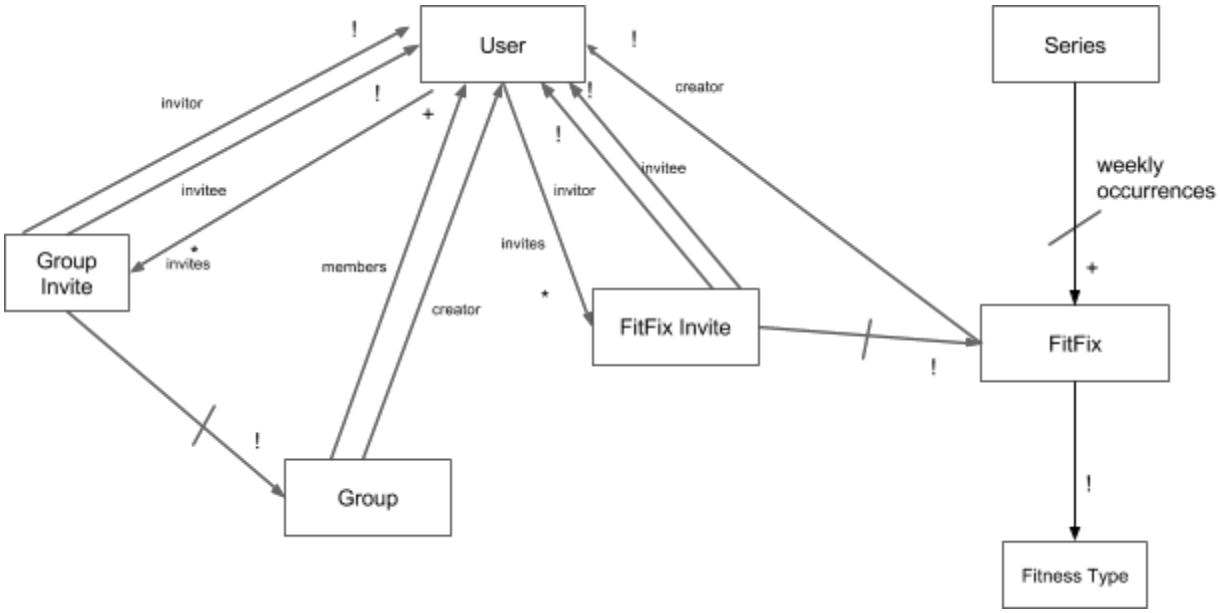


Design Model

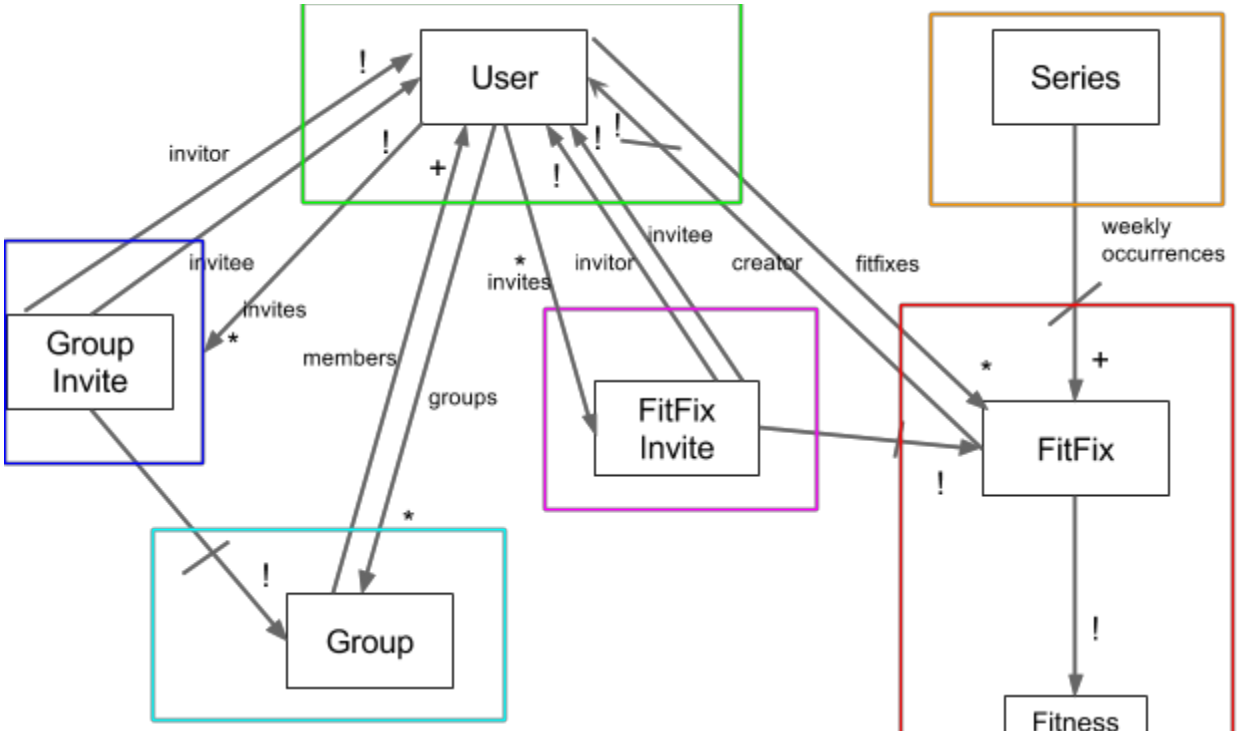
Concepts

FitFix: an exercise activity posted by an organizer and attended by users at a specific time and place. The purposes it motivates are fostering a sense of community and helping MIT students find fun healthy activities.
Group: a set of people who plan FitFixes together on a recurring basis and can be invited to an event together. The purpose it motivates is fostering a sense of community.
Commit: an individual making a promise to attend an FitFix. The purpose it motivates is helping MIT students find fun healthy activities.

Data Model



Contour Diagram



FitFix: key = objectID
Series: key = objectID
User: key = objectID
Group Invite: key = objectID
FitFix Invite: key = objectID
Group: key = objectID

Schema

```
userSchema = mongoose.Schema({
        email: {type: String, required: true},
        local:{
                username: {type: String, required: true, unique: true, lowercase: true},
                password: {type: String, required: true}
        },
        fitfixes: [{type: mongoose.Schema.Types.ObjectId, ref:'Fitfix'}],
        groups: [{type: mongoose.Schema.Types.ObjectId, ref:'Group'}]
});
```

```
fitfixSchema = new mongoose.Schema({
        author: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
        title: String,
        startDate: Date,
        duration: Number,
        numWeeks: Number,
        description: {type: String, default: ""},
        location: String,
        type: String,
});
```

```
var groupSchema = new mongoose.Schema({
        name: String,
        description: {type: String, default: ""},
        members: [{type: mongoose.Schema.Types.ObjectId, ref: 'User'}],
});
```

```
var fitfixInvite = new mongoose.Schema({
        fitfix: {type: mongoose.Schema.Types.ObjectId, ref: 'Fitfix'},
        inviter: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
        invitee: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
        accepted: {type: Boolean, default: false}
});
```

```
var groupInvite = new mongoose.Schema({
        group: {type: mongoose.Schema.Types.ObjectId, ref: 'Group'},
        inviter: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
        invitee: {type: mongoose.Schema.Types.ObjectId, ref: 'User'},
        accepted: {type: Boolean, default: false}
});
```

```
var series = new mongoose.Schema({
        fitfixes: [{type: mongoose.Schema.Types.ObjectId, ref:'Fitfix'}],
});
```

Behavior

Security Concerns

Sketchy

Undesirable users may try to use the app to meet up with individuals. We will mitigate this security risk by using passport to ensure that users are part of the MIT community (i.e ensuring that they have an @mit.edu email address, noting that this opens the app to any cross registered students as well) to participate in the app and view any information. We would like to think that the MIT community is not comprised of sketchy individuals, however we sadly know that this is possibly not the case. Therefore, we will encourage users to be smart about the information that they post to the app and plan meeting locations for FitFixes that are in highly trafficked public areas and during daylight hours (for outdoor activities).

Malicious

There is also the possibility that malicious users try to access the private information of MIT students by injecting code into various forms throughout the app. The major concerns that we are considering are:

I. Database injection- our app uses many input forms that allow the user to interact with the app. These forms are susceptible to injection attacks from users looking to manipulate the database.

II. Cross Site Scripting - malicious users may use this the input fields to run scripts that we did not write within our app.
III. Cross Site Request Forgery - this risk is not as great in this app as we are not working with highly personal data. However, a malicious user may set up hidden requests to the app. This would be particularly harmful in conjunction with Cross Site Scripting.

We will mitigate these security risk by sanitizing the inputs to all forms in the app. We recognize that sanitation does not guard against all malicious activities. However, we do not believe that the app will gain great attention from users outside the MIT community and therefore are not highly concerned. Additionally, the app does not store large amounts of highly personal data and were this information to be obtained, the damage could be controlled very easily.

Security Policy

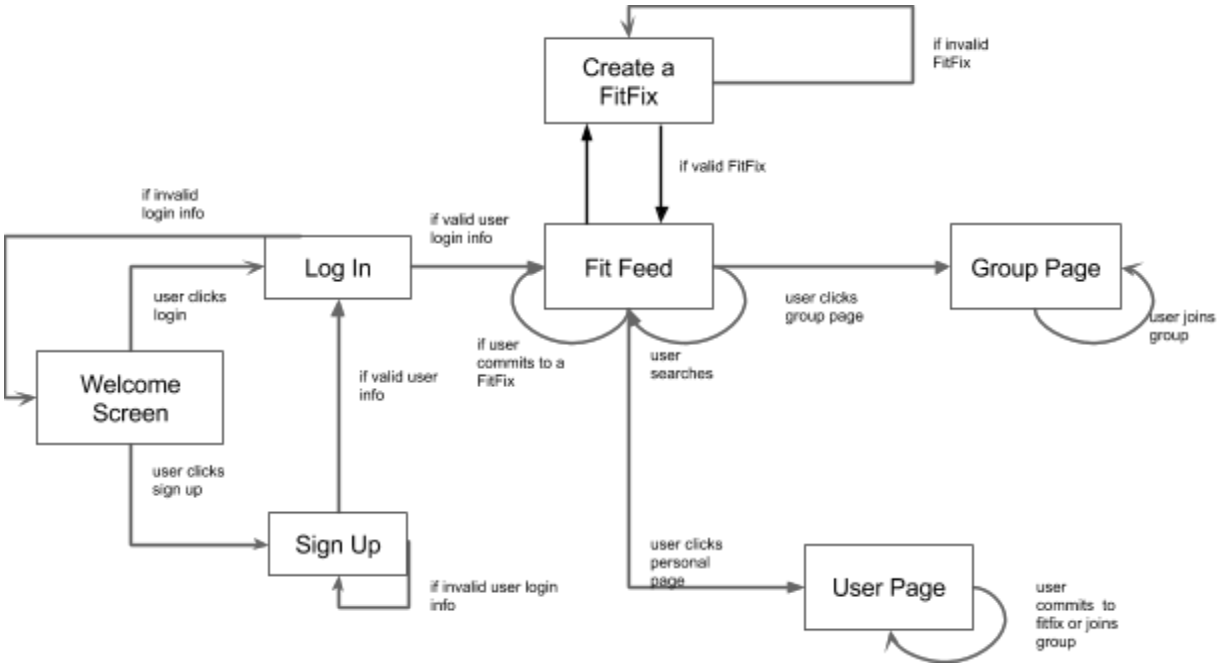
- Only the user that created the FitFix can edit any of the details or delete the FitFix
- Only users in a group can invite that group to a FitFix
- Only the user can see his or her stats
- Only a user in a group can edit the details of that group
- Only a user can see his or her own calendar

Threat Model

- A user that is not part of a group or is not an author of a FitFix can construct requests to eavesdrop or tamper with other groups and FitFixes.
- A user with basic credentials can construct requests or make requests through the forms.
- A malicious user may attempt to gain access to a user's account through the use of trial and error of passwords from a known username
- A malicious user may create FitFixes in unsafe locations at unsafe times of the day or in unsafe locations
- A user in the system may attempt to create an overwhelming number of invites or fitfixes rendering the system useless.
- A user might try to use FitFriends as a dating service.
- A user might try to promote his/her personal training services.

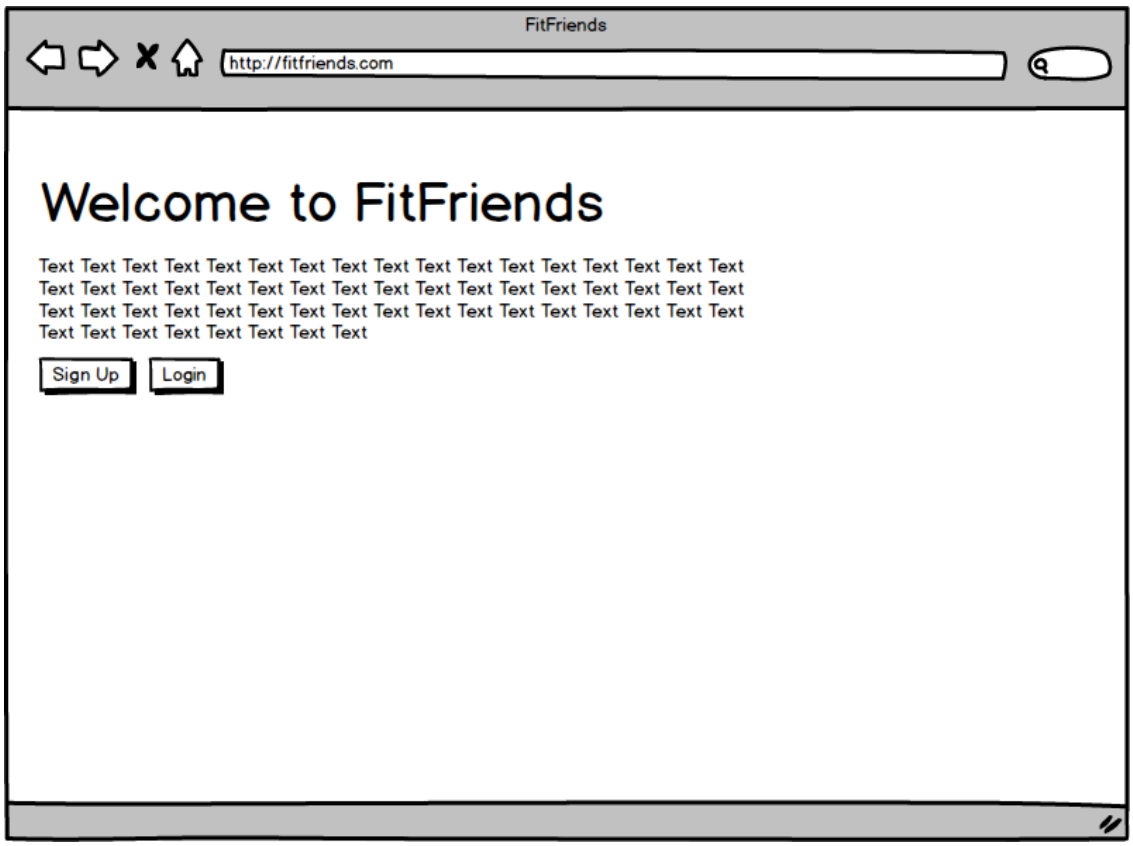
User Interface

State Diagram

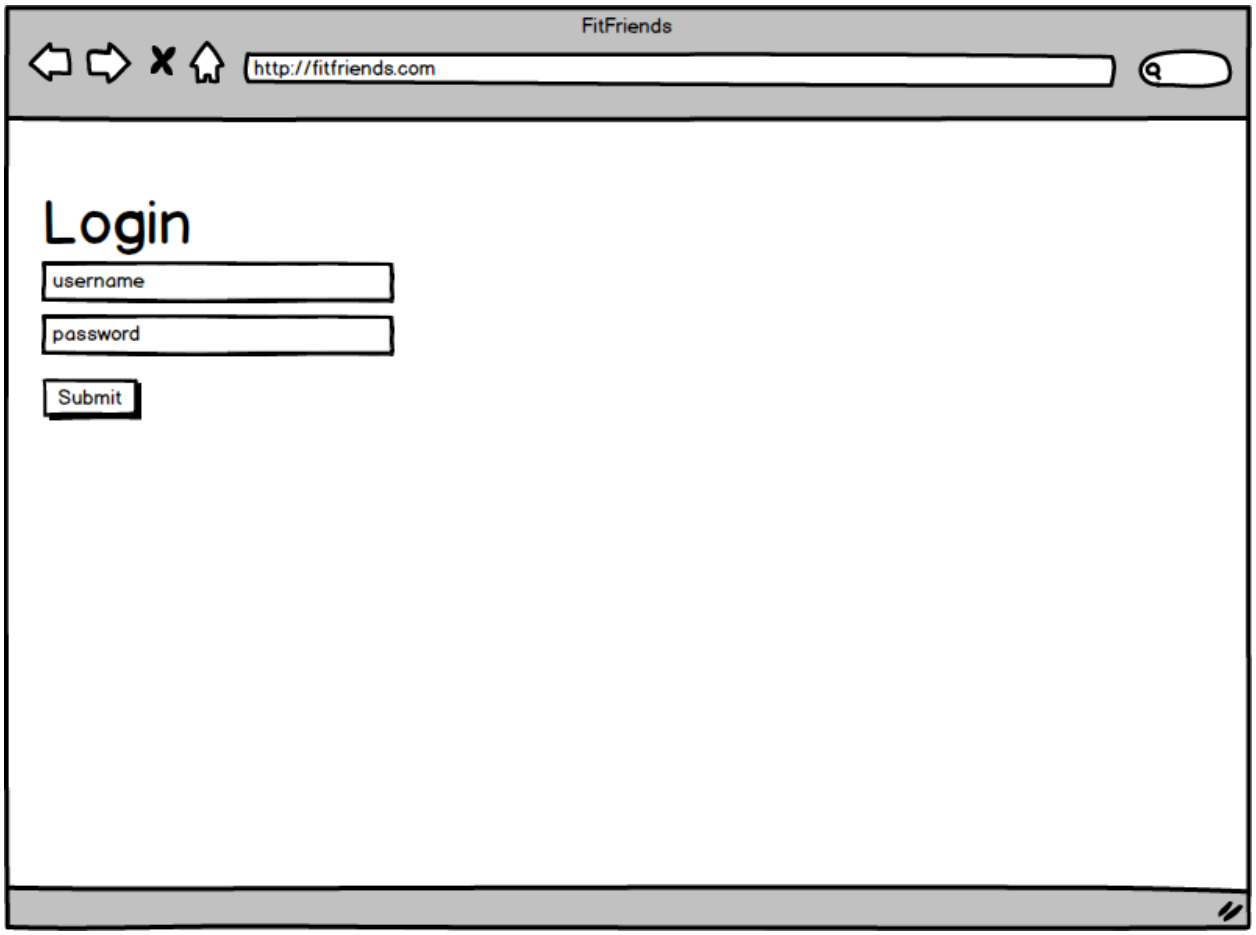


Mockups

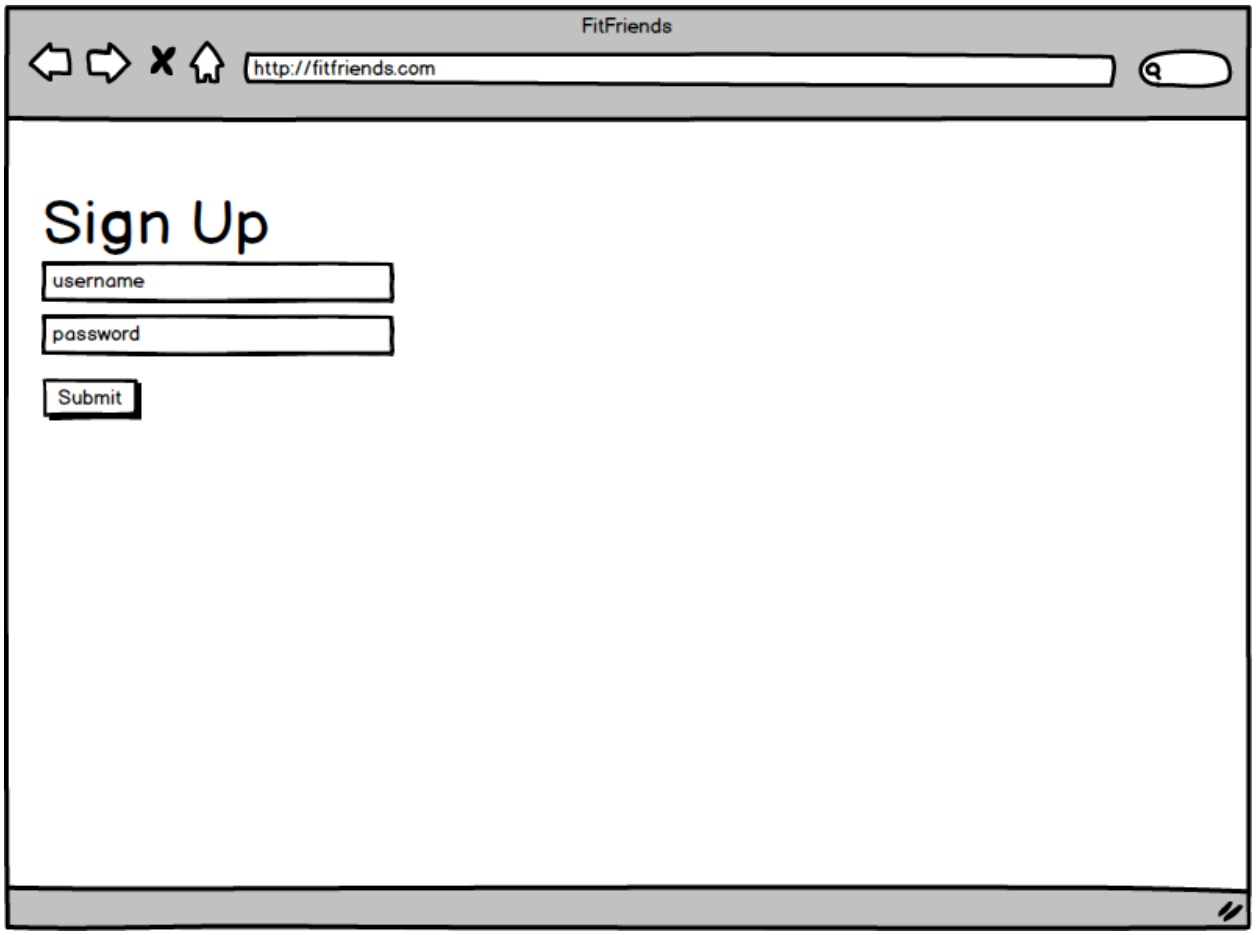
Welcome



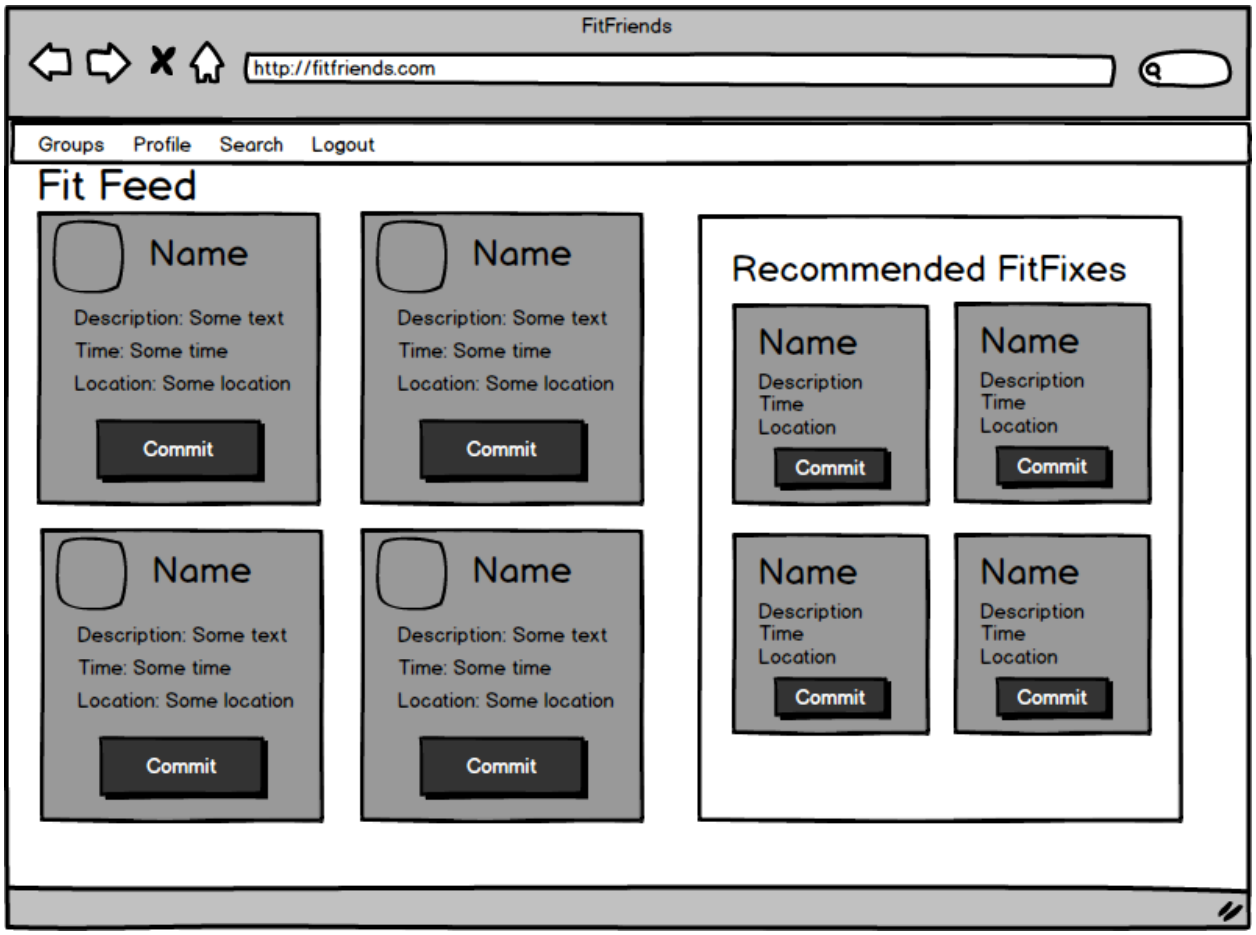
Login


FitFriends
http://fitfriends.com
Login
username
password
Submit

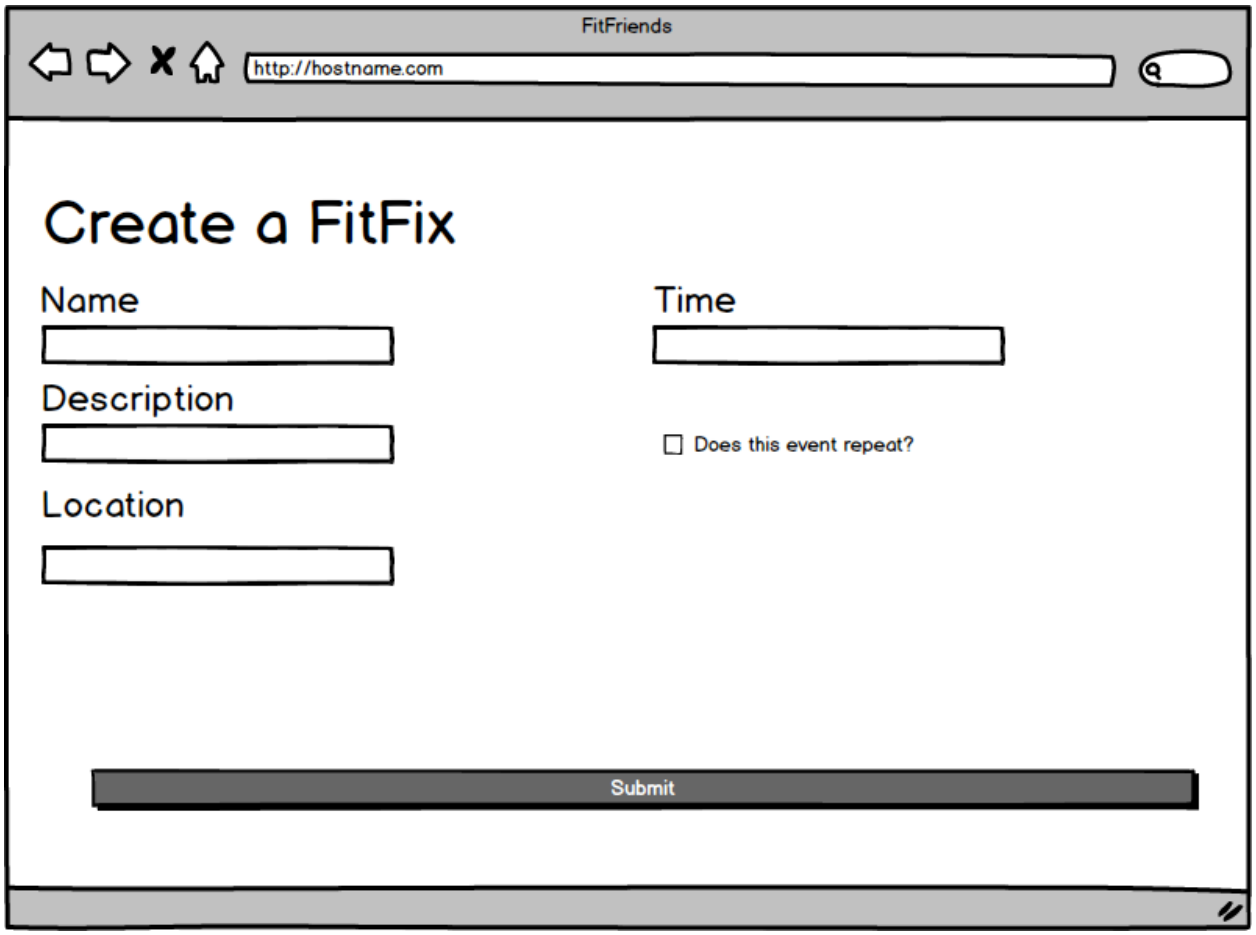
Sign Up


FitFriends
http://fitfriends.com
Sign Up
username
password
Submit

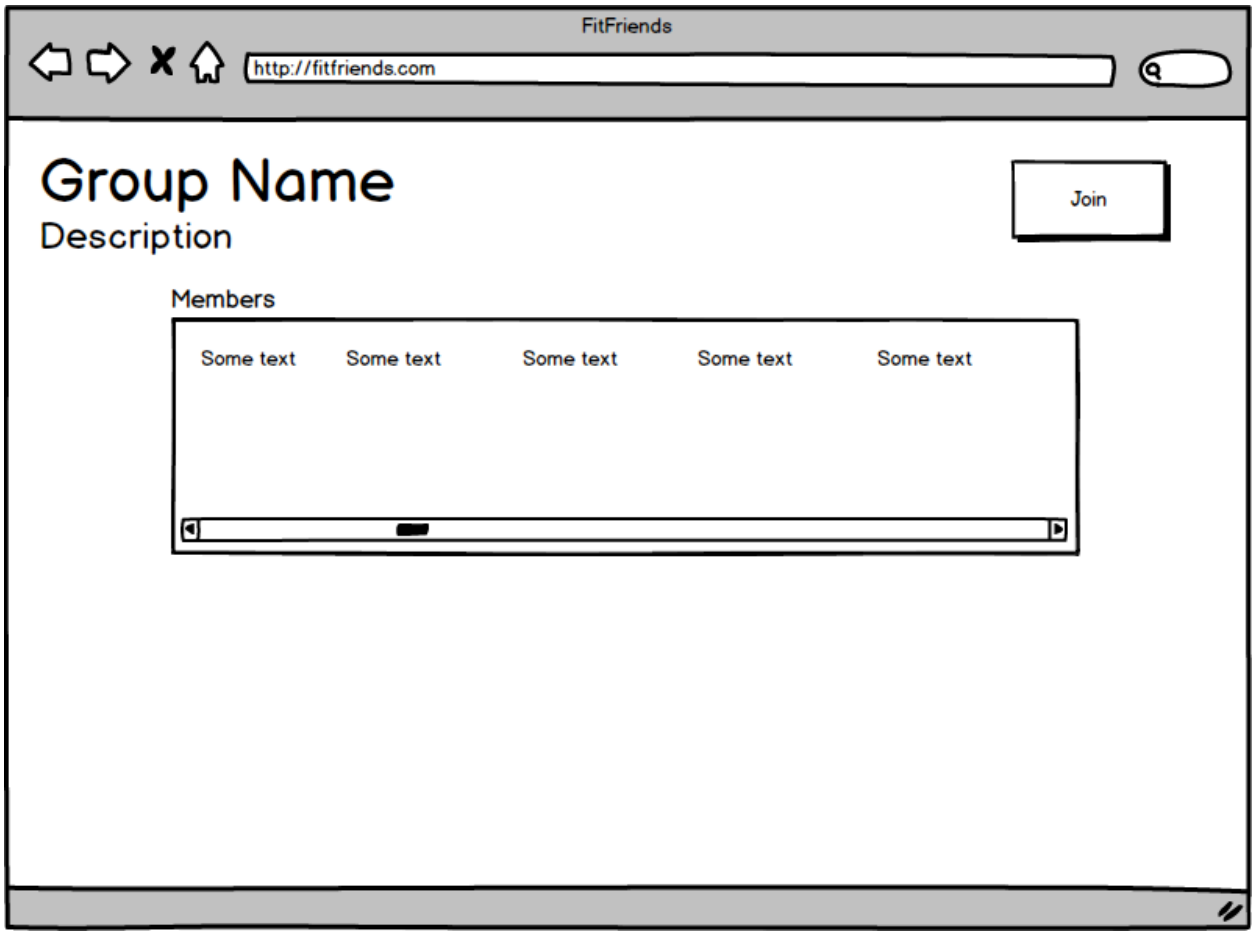
FitFeed


FitFriends
http://fitfriends.com
Groups
Profile
Search
Logout
Fit Feed
Name
Description: Some text
Time: Some time
Location: Some location
Commit
Name
Description: Some text
Time: Some time
Location: Some location
Commit
Name
Description: Some text
Time: Some time
Location: Some location
Commit
Name
Description: Some text
Time: Some time
Location: Some location
Commit
Recommended FitFixes
Name
Description
Time
Location
Commit
Name
Description
Time
Location
Commit
Name
Description
Time
Location
Commit
Name
Description
Time
Location
Commit

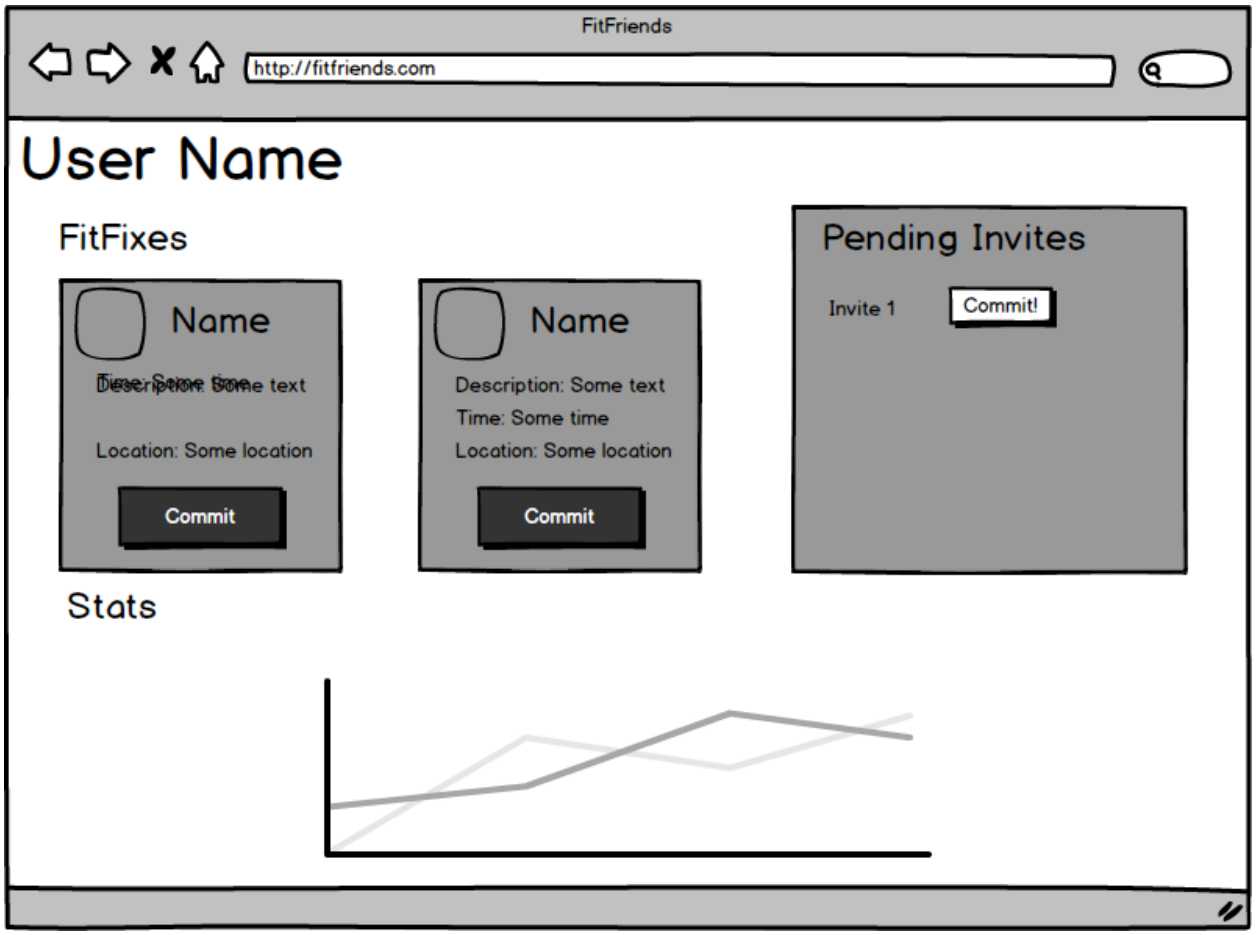
Create a FitFix



Group Page


FitFriends
http://fitfriends.com
Group Name
Description
Join
Members
Some text
Some text
Some text
Some text
Some text

User Page



Challenges

Design Challenges

Groups

There were a couple different way we considered making groups, but what we kept in mind when making our final decision was that the purpose of this application is to create a sense of community. The decision we had to make was how to populate groups.

Invitation Only

One option we considered was making groups invitation only. In this case, the user that makes the group and any subsequent user that is invited can add others. This option is good because the groups are a select number of people. However, this limits the sense of community because we are creating private sects of people when the goal is create a sense of community.

Public Groups

Another option to populate groups is to make groups public. This option is beneficial because people can join any group they want. If a user is interested in running, they can join any and all groups that go running on a recurring basis. The downside of this option is that a group may become overpopulated or users may join an overwhelming number of groups. In order to mitigate this, we can implement a leave group option so that users can be in a reasonable number of groups in order to be active in each one.

We decided to go with public groups for this design challenge.

Invites

Users can commit to any event they would like, since they are all public, but we would also like to implement a way for users to invite their friends or their groups to certain events. We considered who should be allowed to invite other users and/or groups to FitFixes. The three options we came up with were, the creator of the FitFix, a user who has committed to the FitFix, or any user of FitFriends.

Creator

The creator of the group is the one who set up the FitFix. If this user is the only one who can invite users, then the users invited would likely be a small group of people plus those who saw the FitFix on their FitFeed so the number of people attending would not be overwhelming. On the other hand, letting only the creator invite other users limits the reach. The users invited would likely be friends of the creator or people who have already displayed an interest in the type of FitFix. We would like more people to be aware of each FitFix which is why we decided not to go with this option.

A Committed User

If any committed user can invite friends to a FitFix, then FitFixes would probably be attended by a lot of small groups of friends. People would feel more comfortable going to them with a buddy or even get motivated by the user who invited them. This option increases the reach compared to the creator option, but still, the reach is somewhat limited.

Any User

The last option we considered was to let any user invite any other user to a FitFix. This option is ideal because we anticipate that in general, a user would invite someone else if he or she has already committed to said FitFix, but we also allow for users to invite other users to events they themselves might not be interested in or unable to attend, but know that their friend might want to.

The Any User option creates the greatest sense of community within FitFriends which is why we decided to go with it.

Recurring Events

We would like to implement recurring events for our users because we understand the FitFixes will likely happen on a recurring basis. We needed to decide how to implement recurring events and add it into our data model. We decided on adding a boolean flag to FitFixes in order to denote whether an event is recurring. The next decision was how to set an end date. We chose between no end date or an end date that corresponded with the academic calendar.

No End Date

If we decided on no end date then users would never have to worry about forgetting to create events. However, schedules change when semesters end. Another concern we had was users could forget about FitFixes and it would take us space in our database. Even if they did not forget, the way the database would handle an infinite number of events is unclear.

Default End Date

We also considered setting a default end date that corresponds to the end of the semester. With this choice, we take changing schedules into account and we don't have to worry about events becoming stale. The difficult part of this choice is that semesters do not always end on the same day every year. Instead, we will choose two dates that approximately correspond to the end of both semesters. It will make chron jobs much easier.

We decided to implement a default end date to solve this challenge.

Authentication

Choosing a method of authentication is particularly important to us because we would like to limit our application to the MIT community as mentioned earlier. We considered using MIT certificates and Passport.js in order to implement this.

MIT Certificates

We considered MIT certificates since it automatically limits user to the MIT community. It is also a very secure method of authentication and would help us in many of our security concerns. However after doing some research, we found that certificates do not support node.js. There are workarounds for this limitation, but more research told us that it would take a lot of effort that may not be worth our time given the scope of the project.

Passport.js

Passport.js is the other option we considered as a solution to the authentication problem. Passport.js offers an easy way to authenticate users in a secure way. There are also ways to verify users via email which we would like to implement to mitigate security risks and to verify MIT email addresses. The downside to this option is that we need to implement email verification in order to only allow MIT email address, which may prove to be a challenge in itself.

We decided to go with Passport.js to solve this challenge.