

TEAM WORK

PLAN

Stakeholders

- Ben
- Cristina
- Joanna

Roles are as noted in the tasks section

Resources

- No monetary costs
- Stripe takes in a percentage of donations transacted
- Deployment on Heroku for this stage of the project is free
- Every team member has more free time during the beginning of the project than the end

Tasks

Ben's Tasks:

Context diagram
Security concerns
Operations
Thank you notes
Search
Scaffolding/Setting up routes, etc...
Stripe payment
Testing

Cristina's Tasks

Object model
User interface
Wireframes/implement wireframes
Upcoming events/featured organizations
Offline donations
Private vs. public donations
Donation history

Joanna's Tasks:

Purpose and goals
Event model
Feature descriptions
Follows/email notifications of new events
Social buttons

Taking meeting notes/agenda

Calendar of intermediate due dates:

We will finish our assigned sections of the documentation by 10 p.m. the night before they are due, and one person will commit a final version sometime the next day before the time the documentation is due. For code, we will make sure to push up a valid working copy of code to Heroku by 10 p.m. the night before the due date.

Risks

- not many technical risks
- most risks involve getting users and getting organizations to use our product (for product to succeed, we need to gain trust of non-profit organizations and get several to sign up and use our product)
- we want to prevent people from creating fake organizations that just collect money from unsuspecting people (if this were to go live, we would have a moderation/verification process for each organization)

Minimum viable product

- user accounts (two types - one for users + one for non-profit organizations)
- non-profits can create events
- users can sign up for events
- people can donate to users
- Stripe payment

TEAM CONTRACT

(found in TeamContract.pdf in Deliverables folder)

MEETINGS

(found in Meetings folder inside Deliverables folder)

REFLECTION

Peer review

(emailed individually to TAs)

Evaluation

Our project came together pretty well. Much of the underlying code was very similar to some of the class's earlier projects, so we were able to reuse some of our old code. We fortunately didn't have many code conflicts while working separately, and the few that we did were fixed pretty

easily.

Key Lessons learned

1. Commit often and utilize Github's notifications/issues features
2. Use existing gems for certain features
3. Divide up work explicitly and evenly during group meetings
4. Take the initiative to fix little bugs even if they aren't necessarily in your assigned section
5. Set intermediary deadlines for when work should be done (and update the group if you aren't going to make the deadline)
6. Have open communication