Team Work

Plan

Stakeholders

- The only stakeholders for Fireplace is its users. Since it is a personal application driven by the interest of the users, users are responsible for the content provided.

Resources

- Computational: We are limited to the storage space and speed of SQLite3 in development and PostGres in production. Since Fireplace is simply a To-Do List application and doesn't store or display any images, video, or audio, this shouldn't be a problem. Also, the Google Geocoding API allows only 2500 requests each day, so we can only convert 2500 addresses to geographic coordinates each day. Users will only be able to enter an address when creating a tag, and during our development and early production phase we don't expect users to be storing that many locations.
- Cost: Creating this application will cost us no money. The electronic resources we'll be using (Ruby on Rails, Heroku, Balsamiq, Google Geocoding, HTML5 Geolocation) are all available free online. Additionally, Team Spider will be developing the application in its entirety, requiring no outside services.
- Time: We only have a few weeks to design and develop Fireplace, and in particular we have less than a week to develop a minimum viable product. The team members also have other commitments and responsibilities, such as classes and extracurricular activities, so they won't be able to spend all of their time on this application.

Tasks

Task Breakdown

1. Design data model(5%), design app behavior (20%)
2. Design models(15%), design ui(20%)
3. Create models(10%), create controllers(10%)
4. Create views(10%)
5. Make modifications and add features for final product (10%)

Note: Percentages are rough estimates of the fraction of total project work hours that will be dedicated to each task. Note that a large amount of project work has been dedicated to designing the app behavior and ui, since the defining features of our app center around its simple interface.

Primary Responsibilities

All team members will participate in all areas of the project, but to ensure that tasks don't fall through the cracks, we have assigned team members to be in charge of various aspects of the project.

- **Jon:** UI layout/design -- task list page, user settings page
- **Michelle:** UI layout/design -- task edit page, tag edit page
- **Tyler:** Models -- task filtering methods, task/tag edit methods
- **Rebecca:** Models -- database interactions

Timeline

- Preliminary design (Tuesday, November 12)
- MVP implementation (Sunday, November 17)
- Revised design (Sunday, November 24)
- Final implementation (Sunday, December 8)

Risks

Risk: Difficulty integrating with Google Geocoding

Mitigation: Start working with the API early so that if we have issues we can get help or consider switching to a different address-to-geographic-coordinates API.

Risk: HTML5 Geolocation not providing accurate enough information about the user's current location

Mitigation: Start working with the API early so that if we have issues we can consider switching to a different location-tracking API.

Risk: Difficulty developing features in parallel

Mitigation: Use development branches in GitHub and only merge a branch with the master branch when the feature is fully functional

Minimum Viable Product

- Subset of features to be included:
- creating and deleting tasks
- view relevant tasks
- creating tags
- specifying tags and time/location attributes when creating tasks and tags
- Issues postponed: + user indicating a task's importance, duration, and due date
- ability for user to view specifically important tasks, long term tasks, all of today's tasks, all of tomorrow's tasks, etc.
- Simplified data model?