2010-09-27 Cmpt470 Group 8 Team Meeting

Project Proposal Notes

Project: Volunteer Manager System

What is it?
An all in one website where

- volunteer organizations can post volunteer jobs and manage volunteer jobs and schedules
- volunteers can search for volunteer jobs
- both volunteers and organizations can track their work

Why do it?

- Manually scheduling even a medium sized volunteer workforce is time consuming and therefore costly.
- Volunteer work has a very high attrition rate and there is a need to continually find new workers.
- Volunteers using volunteer work as a work reference need verifiable and complete information on their volunteer work.
- There is a need for a "one stop shop" for volunteer jobs.

Why use the internet?

- Fast application development
- Wide reach – can aggregate jobs from a wide variety of organizations
- Near real time tracking of hours completed
- Organizations can neatly combine advertising tasks with scheduling

What do we expect to learn from doing this?

- Learn how to map a real world problem to software products and processes
- Technical skills with respect to web development and database design (web framework development, database integration, web security best practices)

Do you have sufficient knowledge of the problem you are trying to solve?

- We have all worked in volunteer organizations but have no direct experience managing volunteers. Two of our group have contacts in the volunteer sector (Free Geek Vancouver and Habitat for Humanity) who we can contact for more information on the challenges involved in managing a large, continually changing group of volunteers.

Who would use this site? Are there any specific requirements based on these users?

- Both organizations and volunteers would use this site
- Site should be intuitive and easy to understand

Business Process Descriptions

TODO: translate into UML (Cal)

Roles:

Site admin: manages organization and volunteer accounts
Organization admin: manages jobs and scheduling for an organization
Volunteer: person who actually does the work

Goal: **Organization signs up to use site**

An organization admin:

1. Starts an application to be added to the site (this is a form on the site)

A site admin:

2. Checks the application, possibly contacting the organization
3. Sends email to prospective organization admin so they can create a log in

Inputs:

- Organizational data (verified by site admin) (info)

Outputs:

- An organization record is created (resource)
- One organization admin log in is created (resource)

Goal: **Create a job and advertise it**

An organization admin:

1. Creates a job record in the database
2. Makes job visible on the site home page (IE sets whether job can be searched or not by the public)
3. Makes job invisible on the site home page when they no longer want to advertise it
4. Makes job invisible in the organization home page on the site when the organization no longer needs to schedule it (jobs are kept indefinitely for record keeping)

Inputs:

- Job information (info)
- Set the visibility status of the job (info)

Outputs:

- Job record is created (resource)

Goal: **Contact an organization regarding a job**

The volunteer:

1. Searches for a job they are interested in on the main home page
2. Logs in or creates a new volunteer user account
3. Sends an internal message through the site to the organization expressing interest in the job

An organization admin:

4. Responds to the volunteer's expression of interest

Inputs:

- Volunteer provides contact information (at least) (info)

Outputs:

- Volunteer record is created (resource)
- Internal message is created for expression of interest (resource)

Goal: **Schedule a job**

Precondition: the date(s) and time(s) for the volunteer time are agreed on

An organization admin:

1. checks the relevant schedule for job(s) for the time of interest
2. adds the volunteer to the schedule
3. (optional) sends reminder to volunteer about the job

Volunteer:

4. checks their schedule for conflicts if they have more than one volunteer job
5. alerts organization admin if there is a conflict

Inputs:

- Date and time agreed upon by the volunteer and organization admin (info)

Outputs:

- Schedule record is created joining job, volunteer, date and time (resource)

Goal: **Record work done (or not done)**

Precondition: scheduled work date and time has passed

An organization admin:

1. Checks schedule
2. Indicates whether the volunteer did the shift or not (three states: null, work done, absent)
3. (Optional) Contacts volunteer offline as follow up

Inputs:

- Job either done or not done (info)

Outputs:

- Schedule updated with status of work (info)

Goal: **Volunteer gets summary report of work done**

Volunteer:

1. Logs in to their account
2. (Optional) enters dates, organization of interest or other filtering information
3. Prints out a formatted report that can be added to a resume

Inputs:

- Log in (info)
- Optional filtering information (info)

Outputs

- Printable report (info)

What we plan to do

We plan on analyzing and designing the database model first:

- Entities relating to organizations, volunteers and jobs
- Define the relevant relationships between these entities

Simultaneously we will design the look of the site and the relationships between the different site pages relating to the tasks outlined above.

Thirdly we will tie together the design and the database model to make a working site.

Fourthly test and deploy the site. We will test all components as well as do useability tests for the site on the production server. This step is in fact concurrent with the third step.

How we plan to implement the site

We plan to use mysql for our database storage and php to implement page generation from the database. In addition we will take advantage of web tools such as jquery (javascript) and smarty templates (php) to simplify development. All these tools are commonly used and are known to scale well.

As one of our learning goals is to learn framework design, a lightweight, modular framework has been developed. While there are some disadvantages to using a custom designed framework, this framework does offer some advantages for our work. Jquery and smarty are integrated into the framework from the start rather than being patched in after the fact. One of our team members, having authored the code, is an expert on how this framework functions. This should in fact simplify support should any framework related issues arise. The framework is deliberately made simple, leaving out many complicated an unnecessary features of other frameworks. Because it is relatively simple, potential performance issues of other more complex frameworks are avoided without losing many of the advantages offered by a component based model, view, controller design that will automate and streamline the coding process.

Help will be implemented using mediawiki a php based tool used by wikipedia that is easy to install and maintain.

Site Layout

Home page:

- Header / Menu: volunteer and organizational registration and log in links, job search widget, contact links
- Main page area: paginated list of volunteer jobs (will have contact links if someone is logged in)
- Footer: other links: legal, copyright notice etc

Note that log in forms can be integrated into the page design via css and javascript.

Volunteers:

- Volunteer sign up pages: register / update. Volunteer reports page

Organizations:

- Organization sign up / edit page, admin registration pages: list, remove, add / update
- Job manager pages: list (search, change status), add / update
- Schedule pages: show schedule for a period of time, modify schedule / assign work
- Help

Resources

The framework is being developed right now and will be available with documentation before we begin database design. The team member who worked on the framework will then be available to help trouble shoot and support the rest of the team in the design, code, test and release phases of the project.

We estimate that we will need approximately 15 hours per week per person for 4 weeks or 240 hours to complete the project. We provide as a rough estimate the following breakdown:

Database design: 20 hours
Visual site design: 40
Code / Test / Deploy cycle: 120 hours
Documentation: 60 hours

These estimates do not include the framework development which is happening now and is estimated to take an additional 20 hours to complete code / test and deployment. An additional 20 hours is estimated to be needed to train the rest of the team in its use and troubleshoot problems. The framework should be useable by the beginning of October approximately coinciding with our proposal submission.

Deliverables

- Mysql database
- Project website
- This proposal, project report, project poster, user help documentation

License

The site and additional materials will be licensed under the GNU General Public License.