Project Description

Nonprofit organizations that deal with donations require impeccable management and coordination to connect the people who request assistance with the provided donations. Additionally, a nonprofit usually relies on funding of some sort, usually grants, to remain operational and provide services to those in need. Grant applications need to include specific metrics that prove the program is achieving the community's need in order to be awarded.

Our creation of a deployable software for a nonprofit organization would allow for easier management of all the moving pieces in a donation reception process. It would allow for people who were in need to easily request assistance. Items donated would be stored in a database, so a detailed list of inventory would be accessible at all times, improving oversight of the donation process. To continue the funding of the nonprofit using the software, the software would generate reports on useful metrics to prove its efficacy to those approving the grant requests. This would remove the human data processing element in the grant writing process giving the grant writer the ability to write more grants to receive additional funding.

Project Milestones

October 3, 2016: Initial Project Description

- Finalize the members of the group, have an idea of where they will fit in the design of the project.
- Establish the project to be completed, and outline the milestones for the first and second semesters.
- Form a general idea of the requirements of the project, and the tools we will use to complete it.
October 24, 2016: Project Proposal
- Finalize responsibilities for each member. Each team member should now know exactly what they need to accomplish for the semester and have a schedule for delivery times.
- Finalize hardware, software, and computing resources. We should have established the best technology available to use for the database, frontend, backend, and servers.
- Outline of all preliminary use cases. We should have a solid understanding of all the cases we need to provide for, and the requirements needed to fulfill those cases.
- Initial design of the front end, back end, and database.
- Understand how every piece of the software will fit together, and what each piece will be responsible for.

October 31, 2016: Project Proposal Video
- Make the problem visual and try to explain the importance to a wide audience.
- Describe how we will implement the software.

November 28, 2016: Website Mockup
- Create a wireframe for the website to show how the website should function.
- Display basic functionality for all users of the website (Donors, donees, employees).

December 9, 2016: End of Semester
- Ready for implementation
- Finalized designs of database, front end, and back end. Communication throughout is documented and outlined.
- Any additional edge use cases are included in the designs
- Any additional functionality under consideration is documented but sidelined, pending implementation progress.

February 2, 2017: Final Project Design
- Submit the final project design document.

March 3, 2017: Check in of Several Features
- The donor, donee, and user pages are all completely functional.
- Sign up forms for the donor and donee are complete and allow creation of new donors and donees into the system.
- The website functionality is complete in regards to the donors and donees interaction with it.

March 17, 2017: Start of Spring Break
- Implementation of the matching algorithm is complete.
- All of the website features not in the admin panel are complete at this point.
April 21, 2017: Feature Complete
- The product is feature complete.
- Allow two weeks for polish and detailed documentation after this point.

May 1, 2017: Final Project Video
- Complete and present the final project video to the class demonstrating the functionality and usefulness of our project.

May 3, 2017: Quad Chart
- Turn in quad chart for the project.

May 5, 2017: Stop Day
- Completed project.
- Include all the functionality for a useful donation management system.
- Provide useful documentation and installation options for anyone wishing to use our software to deploy their own instance of it.

<table>
<thead>
<tr>
<th>Name</th>
<th>Duration</th>
<th>Start</th>
<th>Finish</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Proposal</td>
<td>5d</td>
<td>10/1/2016</td>
<td>10/16/2016</td>
</tr>
<tr>
<td>Project Proposal Video</td>
<td>11d</td>
<td>10/13/2016</td>
<td>10/24/2016</td>
</tr>
<tr>
<td>Databases ERD Diagram</td>
<td>20d</td>
<td>11/1/2016</td>
<td>11/14/2016</td>
</tr>
<tr>
<td>Program Designs Complete</td>
<td>40d</td>
<td>10/17/2016</td>
<td>11/19/2016</td>
</tr>
<tr>
<td>User Profile End</td>
<td>10d</td>
<td>11/16/2016</td>
<td>12/4/2016</td>
</tr>
<tr>
<td>Build database for testing</td>
<td>40d</td>
<td>10/1/2016</td>
<td>12/4/2016</td>
</tr>
<tr>
<td>Donna Backend</td>
<td>30d</td>
<td>10/23/2016</td>
<td>2/15/2017</td>
</tr>
<tr>
<td>Donna Backend</td>
<td>30d</td>
<td>10/23/2016</td>
<td>2/15/2017</td>
</tr>
<tr>
<td>User Backend</td>
<td>30d</td>
<td>10/23/2016</td>
<td>2/15/2017</td>
</tr>
<tr>
<td>Online database script</td>
<td>10d</td>
<td>12/6/2017</td>
<td>2/15/2017</td>
</tr>
<tr>
<td>Donor and donor signup forms</td>
<td>10d</td>
<td>12/6/2017</td>
<td>2/15/2017</td>
</tr>
<tr>
<td>Setup testing environment</td>
<td>10d</td>
<td>2/6/2017</td>
<td>3/15/2017</td>
</tr>
<tr>
<td>Add expiration to signup forms</td>
<td>10d</td>
<td>3/6/2017</td>
<td>3/16/2017</td>
</tr>
<tr>
<td>Sheet signup form in admin panel</td>
<td>20d</td>
<td>2/26/2017</td>
<td>3/8/2017</td>
</tr>
<tr>
<td>Implement matching equations</td>
<td>10d</td>
<td>2/8/2017</td>
<td>3/8/2017</td>
</tr>
<tr>
<td>Admin panel feedbacks</td>
<td>20d</td>
<td>2/1/2017</td>
<td>3/8/2017</td>
</tr>
<tr>
<td>Tail Report Generation</td>
<td>20d</td>
<td>2/27/2017</td>
<td>4/2/2017</td>
</tr>
<tr>
<td>Sheet Menu Generator</td>
<td>20d</td>
<td>2/27/2017</td>
<td>4/2/2017</td>
</tr>
<tr>
<td>Donor list Generator</td>
<td>20d</td>
<td>2/27/2017</td>
<td>4/2/2017</td>
</tr>
<tr>
<td>Donor report and details</td>
<td>10d</td>
<td>2/27/2017</td>
<td>4/2/2017</td>
</tr>
<tr>
<td>Payment and documentation</td>
<td>10d</td>
<td>3/1/2017</td>
<td>3/5/2017</td>
</tr>
</tbody>
</table>

Budget

Our project can be done at no cost for development. Only when the software we build is deployed will we require a server to run it. All the tools and programs used to build it can be found for free.
Work Plan

- Backend developer: Cheng-Yeh Lee
- Backend developer: Taylor Hockersmith
- Testing: Ryan Ahlgren
- Frontend web developer: Michael Bechtel
- Database designer: Cheng-Yeh Lee
- Database designer: Dustin Horvath
- Web designer: Rusty Riedel
- Backend architect: Dustin Horvath

Github Link

https://github.com/rustyriedel/donationDatabase

Final Project Design

We intend on implementing the donation database software with common industry standard technologies. For the front end we will use HTML5, CSS3, JavaScript, and Bootstrap. For the backend of the software to run all the server side logic we will use php 5. This is a typical software stack for web development.

Our software will run on a web server with a website and database component maintained by a nonprofit organization. Donors and hopeful donees will register with the website before use. As an additional measure, we will also have an email authentication step in order to ensure that the information given by the user is valid.

Donees registered with their site will be asked to complete a need assessment survey to gather information that will be used by the matching algorithm to evaluate the donee's needs and allocate donated goods accordingly. The survey will request details such as name, telephone number, address, email and demographic information such as age, ethnicity, income and number of members in their household. The organization maintaining the software will be provided access to metrics which we intend to provide to assist in the grant-writing process, as well as tools to track the current inventory of donations (e.g., to examine what types of goods are in surplus, and what goods need to be delivered to donees).

Donors will select from a number of items listed on the donor website and/or donate funds directly for the nonprofit to purchase what is needed most with little effort by the donor. We refer to this selection of items as a pledge, and will be given a referral number for keeping track of it. The donated physical goods will be automatically matched to one or more donees by the matching algorithm. The donor will be provided with a receipt for tax deductible purposes upon request.
Physical donations pledged by a donor will either be dropped off at the nonprofit's donation office, mailed into the nonprofit's donation office, or dropped off at a predetermined points by the nonprofit for pickup.

Employees / volunteers working for the nonprofit that is using our software will enter the donations received into the inventory database via an easy to use employee webpage. Each pledge of physical donations made by a donor will have a reference number that will be entered to ensure the items that were pledged for are indeed delivered. If fewer items than pledged were delivered, then the system will post the deficit on the donor request page for other donors to acquire. An estimate of each donations value will be entered into the database along with each donation to provide the value of a tax receipt if requested as well as provide the important cost metric for grant writing. This web page will also include a feature to generate a report of the inventory for manual audits of the inventory if any mistakes were made. Any necessary corrections to the inventory database can be made from this webpage.

Reports can be generated and can then be used for various purposes. If requested by a donor, a tax receipt can be generated which will have the relevant information regarding their donation. This would include the amount they donated, the date of the donation, the name of the donor and the logo and contact information of the nonprofit organization.

Inventory reports can be generated by employees/volunteers if needed for manual audits of their inventory. In the case that an item has fallen below its specified threshold, an admin would be notified and donation request page would be populated automatically with the low quantity items.

Metrics used in grant writing will be generated by our system so that they can be used in future grants. These metrics would include the number of people who received donations, the total cost of donations, and the demographics of all donees. An example field to generate would be the percentage of the people served who were under the age of 18. If the software generates all these metrics, it will allow for the grant writer to choose what metrics they think are pertinent to their grant proposal without having to do any number crunching themselves. This will allow for the grant writer to write more grants in less time with the goal of getting more funding for the nonprofit.

A list of donations can also be generated and then sent to all donees who requested a specific item. The order in which these lists are sent out will be handled by our server side algorithm. Our algorithm is not fully developed yet, but one of the major factors will be the donees who have been in the system the longest but not received aid will have higher priority.

To avoid the difficult to achieve security standards that would be required for handling a donor's credit card details, any financial transaction will take place on an external site by using a donate button. Using a third-party service such as PayPal donate, or Google Wallet would link
the donor to a form on PayPal or Google’s respective site that is secured to handle sensitive credit card information. To use this donation method, each nonprofit using the software will have to set up an account with PayPal or google so they can receive the funds. Detailed instructions on how to set up an account will be provided in the final documentation of the software. Using this method for obtaining monetary donations ensures that the nonprofit can receive funds without the concerns of handling sensitive financial information. Further, this decision ensures that even in the event that the server running the software is compromised, financial damages and breaches of confidentiality will be minimized.

In order to ensure the security of our users’ passwords, we will hash our users’ passwords with a per-user randomly generated salt to ensure that even in the event that our entire database is compromised by a malicious third party, recovering our users’ passwords would be a monumental effort.

As a backend security measure, we will use sanitation functions on user input to prevent the possibility of a SQL injection. To further protect from SQL injection, we will obfuscate the names of our database’s tables so that an attacker without access to SQL output would be unlikely to be able to successfully alter or delete database records.

The front end interface will all be responsive. We will create responsive web pages that will display appropriately depending on what device the content is being displayed on. In 2016 approximately 10% of internet users are mobile device users only. This number of mobile only users is expected to grow rapidly in the future, so it is imperative that we create a website that is easily usable for these users. Using a mobile first approach, we can design each interface to work with the smallest mobile device first, ensuring that it looks perfect and performs well. From there we can modify the format for larger devices by allowing more content for its higher resolution and richer content if the device supports it. This mobile-first strategy also minimizes computing resources required by mobile devices, where computing resources are most scarce, ensuring a better experience for all users regardless of the nature of device they use to interact with the software.

We intend for organizations who wish to use our software to download one zip archive of the software before being guided through a brief installation process similar to that of Wordpress. This installation process would confirm the presence of required dependencies such as MySQL, create default configuration files, and initialize the structure of the database and tables used by the software. The configuration files will contain details such as the nonprofit’s name, path to an image of their logo, and a sharable link to their 3rd party donate button. Our software will look at these fields in the configuration files and populate the skeleton webpages accordingly to fit their nonprofit specifically.

A non profit organizations’ website typically has a lot of content about the organization to show what good it is doing for the community, explain the services they are providing, recognize donors and volunteers for their efforts, and notify donor and donees of upcoming events. For a
nonprofit to set up our system will require someone who is tech savvy enough to install our deployable software on a server and handle all the other networking details to run the server. Our assumption is if they have such a tech savvy volunteer or employee setting up the software, they can also handle the creation of content using the easy to learn front end web languages. This will allow the nonprofit to create whatever content they desire without restrictions applied by our software. Additionally, all of our code will be well documented so any changes they would want to make to the provided software, such as changing the webpage styling or the format of the page will be of minimal effort.

Donation database use case diagram
Software architecture diagram

Database Entity Relation (ER) diagram
### Donee interface website page

**Item Requesting Form**

<table>
<thead>
<tr>
<th>Item</th>
<th>Category</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Item 1)</td>
<td>(Category 1)</td>
<td></td>
</tr>
<tr>
<td>(Item 2)</td>
<td>(Category 2)</td>
<td></td>
</tr>
<tr>
<td>(Item 3)</td>
<td>(Category 3)</td>
<td></td>
</tr>
<tr>
<td>(Item 4)</td>
<td>(Category 4)</td>
<td></td>
</tr>
<tr>
<td>(Item 5)</td>
<td>(Category 5)</td>
<td></td>
</tr>
</tbody>
</table>

**Special Requests**

Request special items not shown above

[Request Donation]

---

### Donor interface page

**Item Donation Form**

<table>
<thead>
<tr>
<th>Item</th>
<th>Category</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Select Item)</td>
<td>(Autopopulate category here)</td>
<td></td>
</tr>
</tbody>
</table>

**Special Donations**

[Name of special item not available above]
[Number of item]

[Request Donations]
Volunteer (user) interface page
Ethical Issues

As our database will be storing potentially sensitive information about our users, it is ethically imperative that our database security is strong enough to ensure that this information is kept confidential to protect the privacy and finances of our users.

From a financial standpoint, we do not plan to handle any money donations directly in our program/website. Instead, we are going to use a PayPal donation button, or something similar, so that all transactions are handled securely and privately by financial company/institution.

From a Utilitarian standpoint, we are making a software that will help nonprofit organizations be more efficient in their operations and, hence, allow them to be more effective in helping people and potentially address/reach a wider target audience.

Intellectual Property Issues

As this project is being undertaken for the benefit of charitable organizations of many scales, it will be published as a free project on Github. While it has not yet been researched, a free license will be used to protect the use and distribution of the finished product; this will likely be the GNU GPLv3 Public License, the MIT License, or the Apache Foundation License.

Our development group has no intentions to charge for or restrict the use of the completed product, but license protection will be required to prevent third parties from using code to produce and sell their own competing software products without citation or mention of its source. We believe that profiting from this project would divert from its original design of providing a useful service to organizations that do good for others.

Change Log

- Changed meeting time to match the schedule for this semester.
- Changed the date of the first milestone in the second semester to March 3rd. This is our planned date for the completion for a large amount of work.
- Changed the description for the March 17th milestone to reflect our new goals as listed on the gantt chart, namely a completed implementation of the matching algorithm.
- Added milestones for the final project description, final project video, and the quad chart.
- Added information about deployment to the description of the May 5th milestone.
- Replaced the old gantt chart with the newly revised one.
- Changed Ryan’s role to testing of code and software features for completeness, correctness, stability, and ease of use for the end users.
- Added the database ER diagram.
- Updated the mockup pages to our actual web pages we are using in the project.