Project Summary Document

## Introduction

Team Fuel set upon a project to design and develop software that could be used by foodbank organizations to track donations as they arrived at an organization, recording the type and quantity of food received. We included provisions for tracking quantities of food with the same expiration date so that these could be found in storage facilities and cycled out of the foodbank. Our categories were meant to be expandable as capabilities were expanded to include dietary concerns, both medical and cultural. The layout of the Donation Management System lent itself to use by only a single organization or it could be scaled upward to accommodate use by many related organizations.

## Project organization

* Adam laid out our initial data models and prepared most of the Software Design Document when it was originally submitted.
* Aaron created Lo-Fi Wireframes, Object Models, and contributed coding for CSS and HAML page layouts. Aaron also did the coding for column sorting and worked on establishing search capabilities.
* Huong has been a force throughout the project consistently. She has been very effective with BDD and TDD development and reworking the User Stories as they are now. Huong also contributed to CSS and scripting to display pages.
* Michael has been very involved in the project, voicing observations often that he had hoped were helpful. Michael has contributed heavily to documents during all stages of the project and to research into ways of expanding the reach of the project possibilities. Michael developed the Inventory MVC and worked hard to make cross table referencing a reality but ran short of time.
* Lauren has been a valuable resource for the team because of her prior experience with MVC and a well rounded background that was very useful for our team to utilize. Lauren created the team’s repository and filled it early on with MVC entities for much of the code that would be expanded to become our team project.

## Risk analysis

We experienced as many as two of the catastrophic risks temporarily over the course of the project. Some team members experienced failures with Heroku and with Cloud9. Some team members were unable to contribute at various times throughout the project. There is nothing that could have been done within the scope of the academic team project to mitigate those risks. We have no budget and we cannot foresee when events occurring in the lives of our other team members may present exceptional challenges and difficulties.

Overall we experienced some form of at least one of the risks from each of our three categories during the course of the team project. Mostly we were constantly pressed for time to complete the demands of the project as well as homework and video/reading assignments. I am sure that most of us felt that they were constantly trying hard to get one assignment finished so that they could begin working on another. At least one of us will never take a class during the Summer again.

## Hardware and software resource requirements

Michael found and made use of the Faker gem to populate our database tables with a seeds.rb script for testing purposes. Huong added many different gems to facilitate her testing with Cucumber, Capybara, Nokogiri, coffee script, etc. actually far too many to list, way beyond the scope of this document.

Lauren added FactoryGirl, a version number for Ruby and some related others for test vs production modes. There are no special instructions for running the application if not in test.

## Work breakdown structure

Our work breakdown structure has simply been based on recording one’s intentions to be working on a particular aspect of the project by putting their name on the card that corresponds to that work.

## Project schedule

### Milestone 1

Development of create, read, update, and edit operations and database persistence were underway. We had agreed upon basic database table structure and had an workable object model. We had some tests developed and a list of User Stories that contained many of our core functional items.

### Milestone 2

At code milestone 2, we had a running application that lacked some features. We were in flux about what we would be able to complete by the time we had allotted and this led to ambiguity in what we needed to develop testing for. The existence of an actual test suite was lagging behind. Finalizing user interface design was underway.

### Milestone 3

We met our goals for Milestone 3. Coding was completed and we had an application that had been pared down to the features we had decided we could deliver. There may have been some additional small tweaks included after August 11, but the application remains largely unchanged since then.

## Monitoring and reporting mechanisms

Our team met weekly in one of the smaller conference rooms of Mammel Hall. We also made use of Trello, as touched on above, for tracking which aspects of the project that members were currently working on or researching to be contributed to in a future cycle. We used some email for communications, and some text messaging occasionally, but we mostly relied upon Slack for asynchronous messaging between team members.

In retrospect it may have been beneficial to have tried some other tools as well. Some team members found various things lacking with each of the tools that we used. Git was overly complex for most of us to use to capabilities that characterize some of the popular purchased version control systems. Google docs lack many capabilities and have no version control at all. Cloud9 seems to have no experience with the importance of file dates and times, other than allowing unix-like utilities from the command line.

## Miscellaneous

Michael worked on having the application set-up to be monitored by New Relic while it is deployed on Heroku, but is currently unable to login to Heroku from Cloud9 and was therefore unable to add newrelic.yml and the line in the Gemfile requiring “gem 'newrelic\_rpm'” to be loaded with the application. This is Heroku ticket #501474.