Master Test Plan

1. Introduction
   This document is the Master Test Plan (MTP) for the giv2giv project, introduced at the conclusion of the 10th iteration.
   1. Document identifier
      Introduced February 25, 2013 by the giv2giv student developer group at the University of Virginia (composed of Rebecca Boswell, Mark Cheung, Loren Fryxell, Douglas Milvaney, Eric Tsai, and Jason Ya).
   2. Scope
      The purpose of this test effort is to cover the efforts to be made during alpha and beta testing of the giv2giv product, done in collaboration between the student team and the customer. It is assumed that the project will not have reached full functionality by the beginning of this testing phase and will likely not have completed development by the end of the testing efforts. Testing efforts will focus on the minimum requirements and also test the desired/optional features that have been implemented (Reference 1). Testing will begin with component testing on individual aspects of the system (e.g. log-in, charity selection, etc.) and continue with acceptance testing for overall system functions, similarly to the fashion in which development was done.
   3. References
      1. "Service Learning Practicum Requirements Document: giv2giv" (available on collab)
   4. System overview and key features
      Available in requirements document (reference 1).
   5. Test Overview
      i. Organization
         Testing is set to begin concurrently with development until the end of iteration 12 (March 25th) at which point development will cease and the focus will shift to testing (iterations 13-15). Issues raised during testing will be brought to the customer's attention and discussed with the development team.
      ii. Master test schedule
         During the alpha testing phase, individual components should be tested with "on-the-fly" development as they are worked on by the team. Once the beta testing phase is reached, components will be subjected to acceptance testing using requirements testing where a model user is taken through the system and attempts to follow typical user flow (i.e. account creation, log-in, charity addition, donation, etc.). Issues discovered will be addressed as necessary by component and then re-tested for acceptance.
      iii. Integrity level schema
         This is not applicable as the default integrity level scheme is used.
      iv. Resources summary

The testing staff includes the student giv2giv development team, the customer, and the team mentor. The tools used will include the CakePHP unit testing module.

v. Responsibilities
This is not applicable, as all team members will participate equally in testing components.

vi. Tools, techniques, methods, and metrics
Testing will be done on the pre-configured VirtualBox image containing Ubuntu with Apache and MySQL preconfigured. CakePHP unit testing tools will be used to facilitate unit testing for each of the components. Metrics for success will largely be based on acceptance testing and whether components/the system works satisfactorily.

2. Details of the Master Test Plan
    1. Test processes including definition of test levels
        i. Management, Acquisitions, Supply, Development, Operation, Maintenance
        Test tasks: account creation, log-in, charity addition, charity allocation, account edit, admin account controls, Facebook connect, Facebook share, Twitter connect, Twitter tweet

Task	**Test Design**
Method	User account creation
Inputs	Username, password, contact info, payment method
Outputs	Validation page, created account
Risks	Allows invalid inputs, fails to create account, invalid redirect, invalid payment account

Task	**Test Design**
Method	Log-in
Inputs	Username, password
Outputs	Profile page, Account logged in
Risks	Allows invalid inputs, fails to log in, invalid redirect

Task	**Test Design**
Method	Charity addition
Inputs	Charity page, logged-in user
Outputs	Allocation page
Risks	Allows invalid inputs, fails to add charity, invalid redirect, access by invalid user

Task	**Test Design**
Method	Charity allocation
Inputs	Logged-in user, at least one charity added
Outputs	Allocation numbers
Risks	Allows invalid inputs, fails to change allocation

	percentages, invalid redirect, access by invalid user

Task	**Test Design**
Method	Account edit
Inputs	Logged-in user
Outputs	Modified profile information
Risks	Allows invalid inputs, fails to modify information

Task	**Test Design**
Method	Admin account controls
Inputs	Logged-in admin user
Outputs	Access to admin page
Risks	Access to invalid user, invalid redirect

Task	**Test Design**
Method	Facebook connect
Inputs	Facebook username/password
Outputs	Facebook account connected
Risks	Allows invalid inputs, Facebook account not connected

Task	**Test Design**
Method	Facebook share
Inputs	Connected Facebook account
Outputs	Share story on timeline
Risks	Allows invalid inputs, fails to share story, incorrect story shared

Task	**Test Design**
Method	Twitter connect
Inputs	Twitter username/password
Outputs	Twitter account connected
Risks	Allows invalid inputs, Twitter account not connected

Task	**Test Design**
Method	Twitter tweet
Inputs	Connected Twitter account
Outputs	Story tweeted on Twitter
Risks	Allows invalid inputs, fails to tweet, incorrect story tweeted

2. Test documentation requirements
   Additional test documents will outline the specific methods to be used in achieving these test cases and the number of levels necessary for each.
3. Test administration requirements

Anomalies discovered during testing will be noted and issues will be made on Redmine for the student development team to address. Once the anomaly has been addressed, the component/system will be retested.

4. Test reporting requirements
   This is not applicable as test reports will be sent to Professor Bloomfield on a scheduled basis.

3. General
   1. Glossary
      This is not applicable for this document.
   2. Document change procedures and history
      Revision 1.1 – February 25, 2013