

# **Testing Document and Specification**

## Test Plan

Audrey Tsafack  
Jessi Bautista Espino  
Kyra Haake  
Ricardo Alvarado

## Introduction

This document outlines the test plan for the fundraiser application. As outlined in the project Requirements Document, this system needs to provide a medium for users to use the application to donate to fundraisers. This application will also provide a mechanism for discussion amongst providers and students affiliated with the providers. Users will have the option to make donations either as a guest or by creating an account through this app. The testing procedures detailed in this document will ensure that the fundraiser application fulfills the customer's requirements by confirming that it satisfies the specifications mentioned in the Requirements Document.

During the testing process, the fundraiser application's performance will be evaluated in several areas, including guest browsing, guest donation, account creation, user profile page display, login functionality, and previous donation history. The outcome of this testing procedure will allow the application developers to determine project success based on the objectives specified in the Project Plan.

## Terminology

Throughout this document the terms user, student, provider, system, site and username/password will be used frequently therefore, formal definitions will be given.

System- the database and bulletin board that supports the backend of the website

User- people affiliated with UMKC (students, faculty, and staff with SSO username and password) who will exclusively use only the front-end of the website

Student- students of UMKC that are involved in this website

Provider- faculty and staff affiliated with UMKC and Commerce Bank

Site- the front-end website (main page)

Username/Password- unique identifiers that authenticate and validates a user

## Items Tested

Items that will be tested during the testing phase as laid out by the Project Plan will be but are not limited to:

- Ability to authenticate a user by their username and password.  
Test Case: 4.1
- Ability for a user to login via username and password and create a fundraiser  
Test Case: 4.2
- Ability for a user to login via username and password or as a guest to donate to a fundraiser  
Test Case: 4.3
- Ability for users to login if they forgot their password.  
Test Case: 4.4
- Ability for users to change their username.  
Test Case: 4.5

## Items Not Tested

There are features that will not be included in the current testing procedure. This does not mean that these features will not be implemented, but that they have not been implemented and are not available for testing. Those features include but are not limited to:

- Reading Questions from the Archive
- Querying the Database
- Administrative Features that include adding/removing a user, signaling on call status and selecting email preferences

## Approach

The overall method to this testing procedure is manual system testing. Each test case created will have a direct link to the requirements as laid out in the Requirements Document. Test cases that include similar Feature methods will be tested together. Examples of these features include logging in to submit a question, logging in to submit a survey and/or logging in to the back-end website. Test cases such as these test the security features of the system along with the ability to submit information to the database. Each test case will test the security

feature with invalid and valid data (usernames and passwords) to ensure that user requirement of valid users for these features is met.

The features that specify reading previously asked questions and searching by keyword will be tested together but in separate test cases. Searching by keyword through previously asked questions and answers will verify the system's ability to produce results that an individual can validate visually.

Manual system testing will continue throughout the second and third iteration of the project. For each iteration, both old and newly implemented features will be tested. Adding new features or functionality can sometimes interfere with the functionality of old features and to ensure product/project success, all features implemented should function as intended throughout the life of the software.

## **Item Pass/Fail Criteria**

The minimum requirements for this software system were laid out in the Requirements Document and the Project Plan outlined what the creators of the software considered project success.

Implemented features that meet the requirements as determined by the customer, meaning the feature does what the user wants it to do with very little difficulty, passes the testing procedure. Difficulty, as used here, is determined by user comprehension and user ability to use the feature with little to no training.

Features that contain major defects will fail the testing procedure and will be documented via an incident report and turned over to the developer for investigation and revision.

## Test Deliverables

In addition to the Test Plan, other test deliverables include the Test Specification which outlines the specific test cases and expected results of each test, and Test reports which is comprised of Incidents, Defects and Changes.

## Testing Tasks

The following list the testing deliverables and the activities required to produce the deliverable.

Deliverables	Activities
<b>Test Plan</b>	<ul style="list-style-type: none"><li>• Analyze Requirements for System Features</li><li>• Determine Testable/Non-Testable Features</li><li>• Develop Approach/Method for testing</li><li>• Determine Task and Estimate Efforts</li><li>• Develop Schedule for Testing</li></ul>
<b>Test Specifications</b>	<ul style="list-style-type: none"><li>• Analyze Requirements</li><li>• Define Test Cases for Testable Features as Outlined by the Test Plan</li></ul>
<b>Test reports</b>	<ul style="list-style-type: none"><li>• Implement Test Cases as Outlined by the Test Specifications</li><li>• Document Incidents and Defects</li><li>• Determine Severity of Incidents and Defects</li><li>• Determine Changes that Need to be Made to System</li><li>• Document and Submit Change Request to Developer</li></ul>