Chocoholics Anonymous

## Test Plan

12/12/2016

Zachary Brunette

Bryant Graff

Benjamin Haos

## 

## 

## Overview

This document details testing procedures for the software product to be provided for Chocoholics Anonymous.

## Objectives and Tasks

The objective of these tests is to ensure that the product functions according to the defined specifications. The product must verify a Member ID, provide a service, login, verify a member, charge for service, get the provider directory, and process the necessary reports for providers, members, and managers.

## Scope

The scope of the testing includes validation of all entities, testing of the date and time object, and testing for validation in each use case.

## Strategy

Each component of the system will be unit tested. Any data stored as a result of user interactions with a component will be verified for integrity and validity. Each form will be evaluated as a whole by evaluating the information provided upon submission. These forms also have built in error handling that will need to be tested for effectiveness.

Due to the limited availability of developers on this project, all members of the team will be responsible for testing procedures, unless otherwise stated. Any such tasks will be assigned when the appropriate component or system is ready to be tested and will be assigned as needed. Any component must be unit tested before it is submitted. When a sufficient number of components are available to allow for integration testing then an integration test will be performed. When enough components are present to test the application as a whole, then an integration test will be performed on the whole system. Finally, when all components are present a final integration test will be performed, whereupon it will be passed on for acceptance testing.

### Unit Testing

Unit testing will be performed for the billable, provider, user, and service entities as well as the date/time object. Zach will be responsible for unit testing. For each entity, the test will begin by creating and testing a valid object against that object’s validation process. Then, the valid object will be tested by invalidating one of the fields in an object at a time and testing for an invalid result. Each field will be thoroughly tested to make sure any invalid character in a field will return false. Fields with a required length or character type will be tested with invalid lengths and character types to confirm an invalid response. The tests will take place before the program starts when the maven plugin is activated.

### Integration Testing

As components are completed, it will be necessary to test how they interact with each other. This will be done by feeding information relevant to these components through the whole process and evaluating the result. Several sets of data will be used and will test the limits of valid information and just exceed the limits to verify that the process functions as intended. The Choco Test Data Controller tests the database for each entity to make sure it adds the objects correctly, can let the user know if the object already exists, and can fetch each object from the database. Ben and Brian were responsible for integration testing.

### Acceptance Testing

When the product is ready for deployment, the client will evaluate and test the product to ensure that it meets specifications and is ready for operational use. The client will take the necessary time to validate that each requirement has been met, and communicate with us any changes, if necessary.

## Hardware Requirements

Chocoholics Anonymous Terminal

Internet Connection

## Control Procedures

Problems that are encountered during testing will be reported back to Zach. Change requests will be reported to Ben and Bryant with the criteria for the changes.

## Features To Be Tested

Each use case will be tested. The “Verify Member ID” use case will be tested by making sure the member ID contains exactly nine numbers. The “Provide Service” use case will be tested by making sure the provider ID, member ID, and service ID have the correct amount of numbers in the ID. The “Login” use case will be tested by confirming the correct amount of numbers in the ID and that the ID exists. The “Verify Member” use case will be tested by making sure each field meets the required specifications. The “Charge for Service” use case will be tested by making sure the billable’s fields are all within the specifications.

## Features Not To Be Tested

The use case “Get Provider Directory” will not be tested because the option will not be available until a service is in the system. The reports use cases will not be tested because the options will not be available until a billable is in the system.

## Significantly Impacted Departments

The significantly impacted departments are data management.

## Tools

The tool that will run the tests and find bugs is the Maven software that runs the program. The tests will be run every time the software is used.

## Approvals

Approvals must be taken from the lead developer, lead tester, and client.