# The ChocAn Simulator

Test Document

Table of Contents

[1Introduction 2](#__RefHeading__457_848818598)

[1.1Purpose and Scope 2](#__RefHeading__459_848818598)

[To demonstrate to the customer that the program works as it is intended before putting the product into production. This document will outline the tests that will be administered for the program. 2](#__RefHeading__527_848818598)

[1.2Target Audience 2](#__RefHeading__463_848818598)

[1.3Terms and Definitions 2](#__RefHeading__465_848818598)

[2Test Plan Description 3](#__RefHeading__467_848818598)

[2.1Scope of Testing 3](#__RefHeading__469_848818598)

[2.2Testing Schedule 3](#__RefHeading__471_848818598)

[2.3Release Criteria 3](#__RefHeading__838_341687267)

[3Unit Testing 4](#__RefHeading__473_848818598)

[3.1For customers/member 4](#__RefHeading__475_848818598)

[3.2For provider 5](#__RefHeading__840_341687267)

[3.3Services 6](#__RefHeading__842_341687267)

[3.4Generate reports 7](#__RefHeading__844_341687267)

[4Integration Testing 8](#__RefHeading__477_848818598)

[4.1Command line 8](#__RefHeading__846_341687267)

[4.2Service entry 8](#__RefHeading__848_341687267)

[4.3Ask for provider directory 8](#__RefHeading__850_341687267)

[4.4Create reports 8](#__RefHeading__852_341687267)

[4.5Customer action 8](#__RefHeading__854_341687267)

[4.6Provider action 9](#__RefHeading__856_341687267)

# Introduction

This is the test document for the Choc-An system. It is intended to show developers and customers that the program works as it is intended before I release the product.

## Purpose and Scope

To demonstrate to the customer that the program works as it is intended before putting the product into production. This document will outline the tests that will be administered for the program.

## Target Audience

Audience members include developers, managers, clients and customers, and even possibly users of this program. In this application the audience is Brian Breniser, Chris Gilmore, and the teaching assistant of CS 300

## Terms and Definitions

Ascii – American standard code for information interchange → Definition of character, as opposed to int, float, etc.

# Test Plan Description

This part of the document will describe the scope, schedule, and release criteria of the Choc-An system.

## Scope of Testing

The document is meant to illustrate the tests performed on only the required pieces of the Choc-An simulator by Brian Breniser. The test documentation will not include the rest of the application.

## Testing Schedule

The project we are working on is small enough our testing schedule will be incremental to when pieces of the program get done. As I complete a module I will test it, or rather, as I think of a module to write, I will write test code first and then write the module. As I test each piece for verification I will bring the new code into the overall project and test that as a whole.

## Release Criteria

When I've decided that the program is validated and verified the software will be complete. Using prior documentation such as requirements and design docs (for validation), and testing doc (verification), I will consider the program done. December 5th is the deadline for the project, but I expect it will be done sooner.

# Unit Testing

## For customers/member

Display – display 0 – arbitrary number of items. Tests:

1. Displays 0 entries
2. Displays 1 entries
3. Displays 1000 entries (as a theoretical max number)
4. Reports error when no file found
   1. Does not break upon not finding a file

add – 1 JSON entry to the file

1. 1 Addition only creates 1 entery
2. 100 Additions creats 100 enteries
3. Data is verified (correct data entered)
4. Add more than allowed maximum ascii characters per entry
   1. Does not allow, error messages instead
5. Add zero characters per entry
   1. Does not allow (when applicable)

delete – 1 JSON entry to the file

1. Will delete the specified file
   1. tests a set of data, delete one, test new set to verify

delete all – Remove file and start over

1. Delete the whole file and start over with a blank
   1. Have a set of data, delete, check if zero
   2. Will create empty file

modify – only change 1 JSON entry in the file

1. Modify between 1 and all atributes
   1. take data
   2. apply modify
   3. check for correct changes
      1. add more than allowed maximum characters
         1. error messages
      2. Add zero characters
         1. error message

## For provider

Display – display 0 – arbitrary number of items. Tests:

1. Displays 0 entries
2. Displays 1 entries
3. Displays 1000 entries (as a theoretical max number)
4. Reports error when no file found
   1. Does not break upon not finding a file

add – 1 JSON entry to the file

1. 1 Addition only creates 1 entery
2. 100 Additions creats 100 enteries
3. Data is verified (correct data entered)
4. Add more than allowed maximum ascii characters per entry
   1. Does not allow, error messages instead
5. Add zero characters per entry
   1. Does not allow (when applicable)

delete – 1 JSON entry to the file

1. Will delete the specified file
   1. tests a set of data, delete one, test new set to verify

delete all – Remove file and start over

1. Delete the whole file and start over with a blank
   1. Have a set of data, delete, check if zero
   2. Will create empty file

modify – only change 1 JSON entry in the file

1. Modify between 1 and all atributes
   1. take data
   2. apply modify
   3. check for correct changes
      1. add more than allowed maximum characters
         1. error messages
      2. Add zero characters
         1. error message

## Services

Display – display 0 – arbitrary number of items. Tests:

1. Displays 0 entries
2. Displays 1 entries
3. Displays 1000 entries (as a theoretical max number)
4. Reports error when no file found
   1. Does not break upon not finding a file

add – 1 JSON entry to the file

1. 1 Addition only creates 1 entery
2. 100 Additions creats 100 enteries
3. Data is verified (correct data entered)
4. Add more than allowed maximum ascii characters per entry
   1. Does not allow, error messages instead
5. Add zero characters per entry
   1. Does not allow (when applicable)

delete – 1 JSON entry to the file

1. Will delete the specified file
   1. tests a set of data, delete one, test new set to verify

delete all – Remove file and start over

1. Delete the whole file and start over with a blank
   1. Have a set of data, delete, check if zero
   2. Will create empty file

## Generate reports

Services for week

Member report

Provider report

Summary report

For all:

1. Check if report is correctly created
   1. Will create fake data
   2. will generate report
   3. will compare to known good outcome
2. Will create empty (with message) files for no services
3. Will display error when file not found

# Integration Testing

## Command line

The command line takes ascii input and will not break on non-recognized input, such as a command not recognized, or numbers outside any ranges (all input will be in ascii characters, not int, float, or any other data type)

## Service entry

Take in a series of data to conclude with saving one entry for the day/week. This test will be passed if the command creates a viable JSON entry into the file, it does not have to read from the file, necessarily.

## Ask for provider directory

Will return a neatly organized list of providers. We do not need to write to the provider directory with this command. This command cannot return junk, and will return a message of “empty list” if there are no providers. It will return “no file found” if no provider file was found.

## Create reports

Will create the various reports. 1) services for week, 2) member report, 3) provider report, 4) summary report

These reports will be non-empty JSON files, even if there are no events I will populate the file with “no objects” message.

## Customer action

Add, delete, and modify a customer. These tests should perform each action and test whether an action was broken during development.

## Provider action

Add, delete, and modify provider list. These tests should perform each action and tests whether an action was broken during development.