

Bailee Crandley
Ryan Filgas
Dylan Hanson
Jesse Olsen-Jacobsen
Jeremiah Tilse

Term Project: *ChocAn*

Test Plan Document

Table of Contents

1 Introduction - Bailee	2
1.1 Purpose and Scope	2
1.3 Target Audience	3
1.4 Terms and Definitions	3
2 Test Plan Description - Ryan	3
2.1 Scope of Testing	3
2.2 Testing Schedule	4
2.3 Release Criteria	4
3 Unit Testing	5
3.1 Units - Bailee	6
3.2 Strategy - Ryan	7
4 Smoke Testing - Jeremiah	8
4.1 Access	8
4.2 Manager Interface	8
4.2.1 Provider Directory	8
4.2.2 Providers	8
4.2.3 Members	8
4.2.4 Reports	8
4.2.5 Manager and Third Party Logins	9
4.2 Provider Interface	9
4.2.1 Member Verification	9
4.2.2 Member billing	9
4.2.3 Request Provider Directory	9
4.3 Third Party Access	9
4.3.1 Acme Accounting Services	10
4.3.2 EFT	10
4.4 Automated Systems	10
4.4.1 Main Accounting Procedure	10

4.4.1.1 Member Report	10
4.4.1.2 Provider Report	11
4.4.1.3 EFT Data	11
4.4.1.4 Summary Report	11
5 System Testing - Jesse	12
5.1 Manager Adds Member	12
5.2 Manager Adds Provider	13
5.3 Manager or Provider Request Provider Directory - View/Export	13
5.4 Provider Renders Service to Member	14
5.5 Manager Runs EFT Report	15
5.6 Manager Runs Member Reports	15
5.7 Manager runs Provider Reports	15
5.8 Manager Removes Member	16
5.9 Manager Removes Provider	16
5.10 Manager Edits a Member	16
5.11 Manager Edits a Provider	16

1 Introduction - Bailee

This document will describe the testing for the ChocAn data processing software that our team will be designing. It will go over the general test plan, unit testing, smoke testing, and system testing.

1.1 Purpose and Scope

The purpose of this document is to provide a roadmap for testing as the software is being developed. It will reduce the probability of bugs and defects slipping through to the final product. It will also simplify the debugging process by ensuring that each new component added to the system passes all tests before moving on to the next component. This document will mostly be used during the development of the software, but it will also be useful afterwards for maintaining the software and for the customer to understand what has been tested and what has not.

1.3 Target Audience

The primary audience for this document will be the team members who are working on building and testing the software. However, this document will also be useful to anyone maintaining the software, as well as the customer that ordered the software.

1.4 Terms and Definitions

The following definitions may be useful in comprehending this document:

Term	Definition
ChocAn	Chocoholics Anonymous: the company that the software is being developed for
EFT	Electronic Funds Transfer
Provider	Healthcare professional
Member	Patient receiving services

2 Test Plan Description - Ryan

The test plan will be broken into scope, schedule, and release criteria. In general the scope of testing will test all normal user functionality that uses our software, and excludes systems that our team isn't implementing, or libraries that our software makes use of. Our development schedule will follow aggressive checkpoints to complete the project by our deadline.

2.1 Scope of Testing

The scope of the test plan will first and foremost be focused on user stories for valid user input. These will include generation of EFT reports, add/modify a provider or member, add/modify provider directory, request provider directory, check member status, and log service provided for member. Following this, invalid input will be tested for the authorization system in place to ensure isolation. Systems provided by Acme, the EFT system, and the scheduling system are out of scope for this project. Data structure

methods will not be tested, as there is high confidence the standard template library is error free.

2.2 Testing Schedule

As the project deadline is December 3rd, the programming and test plan will need to follow an aggressive schedule.

- Monday 11-22-21 - A barebones structure should be built to support component integration for classes and their methods.
- Monday 11-22-21 - All base level objects should be created: User, Provider, Service Provided, Service Received
- Wednesday 11-24-21 - Unit tests should be developed and implemented for all base level objects to support testing of all related methods.
- Friday 11-26-21 - Mid level manager classes should be complete. Editing methods for provider manager, member manager, provider directory, reporting methods for providers, members, eft data, and provider directory.
- Sunday 11-28-21 - Develop and implement unit tests for mid level structures above. Complete Chocann access including internal calls for all chocann software methods. Completed user access system.
- Tuesday 11-30-21 - Develop and implement unit tests for the chocann access class.
- Thursday 12-2-21 - Completed user interface, and test user stories. Tie up loose ends. Write the project report.
- Friday 12-3-21 - DEADLINE: Tie up any loose ends and polish. Turn in the project.

2.3 Release Criteria

Faults as a result of critical valid user activity and inputs will not be tolerated. At a very basic level the managers need to be able to add and delete member and provider data. If

editing doesn't work predictably, this is an annoyance, but an acceptable result to get the software to market. The following criteria must be met:

- A manager must be able to use an initial account to create others.
- Providers must be able to log in under limited access.
- A manager must be able to create and delete member and provider accounts and directory entries in addition to service data for members and providers.
- A provider must be able to log new service reports in their data and the members data.
- Providers and managers must be able to view and output the provider directory.
- Managers must be able to run member reports, provider reports, and EFT reports.

3 Unit Testing

This section will describe the individual components of the software that will be tested. It will also describe the strategy used for testing.

3.1 Units - Bailee

1. Manager interface:
 - a. Edit Members
 - i. Add Member (with and without valid data)
 - ii. Edit Member info (with and without valid data)
 - iii. Display Member
 - iv. Delete Member (with valid member id and invalid member id)
 - b. Edit Providers
 - i. Add Provider (with and without valid data)
 - ii. Edit Provider (with and without valid data)
 - iii. Display Provider
 - iv. Delete Provider (with valid provider id and invalid provider id)
 - c. Edit Provider Directory
 - i. Add Entry (with and without valid data)
 - ii. Remove Entry (with and without valid data)
 - iii. Display All
 - iv. Export
 - d. Edit Manager Logins
 - i. Add Manager (with and without valid data)
 - ii. Remove Manager (with and without valid data)
 - iii. Display Managers
 - e. Run Reports
 - i. Export Member Reports
 - ii. Export Provider Reports
 - iii. Export Summary Reports

- f. Schedule Reports
 - i. Generate reports every Friday at midnight
- 2. Provider Interface:
 - a. Billing
 - i. Bill Member (with and without valid data)
 - ii. Export Billing Record
 - b. Directory Lookup
 - i. Display Provider Directory
 - ii. Display Entry
 - iii. Export Directory
- 3. Authentication:
 - a. Authenticate users
 - b. Reset authentication

3.2 Strategy - Ryan

Unit testing for the project will be completed in different phases. Classes will be tested individually for expected input and output. Integration tests will be limited to basic testing of user stories when the project is nearing completion. As each unit is being tested and there aren't very many pieces, basic integration tests will happen at each incremental phase, and finally at the end of the build. Testing of data structures will be excluded as the project will use constructs from the standard template library. The project should end up with code coverage somewhere between 80-90%.

Unit testing will be done in four phases:

1. Testing of base objects: member, provider, service_provided, service_recieved
2. Testing of intermediary structures: member_manager and provider_manager classes. Testing of provider directory.
3. Testing of chocann access class.
4. Testing of user interface.

4 Smoke Testing - Jeremiah

This section describes the core components and basic tests needed to verify that the build meets all necessary requirements and is ready for further testing. If these basic tests do not pass, then build is rejected.

4.1 Access

The user, either manager, provider, or third party organization, must be able to access the system.

1. Manager must be able to access the manager menu.
2. Provider must be able to access the provider menu.
3. Third party organizations must be able to access their prospective menus, if they have more than a singular task.

4.2 Manager Interface

The manager interface must provide access to all tasks that fall under a manager's responsibilities.

4.2.1 Provider Directory

A manager must be able to add, delete, and modify the services listed in the Provider Directory.

4.2.2 Providers

A manager must be able to add, delete, and modify provider accounts.

4.2.3 Members

A manager must be able to add, delete, and modify member accounts.

4.2.4 Reports

The manager must be able to run all reports.

- Summary report
- Provider report
- Member report

- Provider Directory list

4.2.5 Manager and Third Party Logins

A manager must be able to create, remove, and modify access for other managers and third parties to access the system.

4.2 Provider Interface

The provider interface must provide access to all tasks that fall under a provider's responsibilities.

4.2.1 Member Verification

A provider must be able to verify the status of a member via card swipe or keyboard entry.

4.2.2 Member billing

A provider must be able to bill a member for a service provided. The following steps are to be taken when billing a member:

1. Verify a members status
2. Enter date
3. Use provider directory to lookup a service, enter its code, and verify the service name and code after
4. Enter comments if needed

4.2.3 Request Provider Directory

A provider must be able to submit a request to have the Provider Directory sent via email.

A file must be created with the requesting provider name and ID, and the Provider Directory listed alphabetically by service name, with corresponding service codes and fees, so that the directory may be emailed by the automated system.

4.3 Third Party Access

Third party organizations must be able to access the system to perform tasks as needed.

4.3.1 Acme Accounting Services

Acme Accounting Services must be able to record payments of member fees, and update a member status if needed.

4.3.2 EFT

The organization responsible for processing EFT data must be able to access the necessary files for processing.

4.4 Automated Systems

Some tasks within the system are automated and must be tested to be sure that the tasks run when scheduled. For the purpose of testing, the scheduled times may be temporarily changed. Once the tests have passed, the times must be returned to their required values and verified for correctness. If feasible, tests should be performed to make sure that the automated tasks run at their required times.

4.4.1 Main Accounting Procedure

At midnight on Friday the week's file of services provided are read and reports produced.

4.4.1.1 Member Report

A file for each member that has consulted a provider during the week must be created so that an email can be sent to the member. The following information must be included in the file:

- Member name (25 characters)
- Member number (9 digits)
- Member street address (25 characters)
- Member city (14 characters)
- Member state (2 letters)
- Member zip code (5 digits)
- For each service provided, the following details are required:
 - Date of service (MM-DD-YYYY)
 - Provider name (25 characters)
 - Service name (20 characters)

4.4.1.2 Provider Report

A file for each provider that billed ChocAn during the week must be created so that an email can be sent to the provider. The following information must be included in the file:

- Provider name (25 characters)
- Provider number (9 digits)
- Provider street address (25 characters)
- Provider city (14 characters)
- Provider state (2 letters)
- Provider zip code (5 digits)
- For each service provided, the following details are required:
 - Date of service (MM-DD-YYYY)
 - Date and time data were received by the computer (MM-DD-YYYY, HH:MM:SS)
 - Member name (25 characters)
 - Member number (9 digits)
 - Service code (6 digits)
 - Fee to be paid (up to \$999.99)
- Total number of consultations with members (3 digits)
- Total fee for the week (up to \$99,999.99)

4.4.1.3 EFT Data

A file containing EFT data for each provider that billed ChocAn during the week must be created so that the provider is paid. The following information must be included in the file:

- Provider name
- Provider number
- Total amount to be transferred

4.4.1.4 Summary Report

A file containing a summary report for the week must be created so that it may be printed and given to the manager. The following information must be included in the file:

- A list of every provider to be paid that week, including:
 - The number of consultations
 - The total fees for that week
- The total number of providers who provided services
- The total number of consultations
- The total of the fees for consultations provided

5 System Testing - Jesse

This section describes the testing of units that interact with other components of the system. Each unit will have listed test parameters.

5.1 Manager Adds Member

Refer to the description of unit test cases.

- 5.1.1 Manager adds member with valid information

- 5.1.2 Manager adds member with invalid name

- 5.1.2.1 Manager adds member with empty name

- 5.1.2.2 Manager adds member with name that's too long

- 5.1.2.3 Manager adds member with invalid name format

- 5.1.3 Manager adds member with invalid street address

- 5.1.3.1 Manager adds member with empty street address

- 5.1.3.2 Manager adds member with street address that's too long

- 5.1.3.3 Manager adds member with invalid street address format

- 5.1.4 Manager adds member with invalid city

- 5.1.4.1 Manager adds member with empty city

- 5.1.4.2 Manager adds member with city that's too long

- 5.1.4.3 Manager adds member with invalid city format

- 5.1.5 Manager adds member with invalid state

- 5.1.5.1 Manager adds member with empty state

- 5.1.5.2 Manager adds member with state that's too long

- 5.1.5.3 Manager adds member with invalid state format

- 5.1.6 Manager adds member with invalid zip code

- 5.1.6.1 Manager adds member with empty zip code

- 5.1.6.2 Manager adds member with zip code that's too long

- 5.1.6.3 Manager adds member with invalid zip code format

5.2 Manager Adds Provider

- 5.2.1 Manager adds provider with valid information
- 5.2.2 Manager adds provider with invalid name
 - 5.2.2.1 Manager adds provider with empty name
 - 5.2.2.2 Manager adds provider with name that's too long
 - 5.2.2.3 Manager adds provider with invalid name format
- 5.2.3 Manager adds provider with invalid street address
 - 5.2.3.1 Manager adds provider with empty street address
 - 5.2.3.2 Manager adds provider with street address that's too long
 - 5.2.3.3 Manager adds provider with invalid street address format
- 5.2.4 Manager adds provider with invalid city
 - 5.2.4.1 Manager adds provider with empty city
 - 5.2.4.2 Manager adds provider with city that's too long
 - 5.2.4.3 Manager adds provider with invalid city format
- 5.2.5 Manager adds provider with invalid state
 - 5.2.5.1 Manager adds provider with empty state
 - 5.2.5.2 Manager adds provider with state that's too long
 - 5.2.5.3 Manager adds provider with invalid state format
- 5.2.6 Manager adds provider with invalid zip code
 - 5.2.6.1 Manager adds provider with empty zip code
 - 5.2.6.2 Manager adds provider with zip code that's too long
 - 5.2.6.3 Manager adds provider with invalid zip code format

5.3 Manager or Provider Request Provider Directory - View/Export

- 5.3.1 Manager request to view provider directory
 - 5.3.1.1. Manager successfully displays the provider directory
 - 5.3.1.2. Manager unable to display the provider directory
 - 5.3.1.3. Manager attempts to display an empty provider directory
- 5.3.2 Manager successful request to export provider directory
 - 5.3.2.1. Manager successful write of provider directory to file

- 5.3.3. Manager unsuccessful request to export provider directory
 - 5.3.3.1. Manager unsuccessful write of provider directory to file
 - 5.3.3.2. Manager attempts to write empty provider directory to file
 - 5.3.3.3. Manager attempts to write to invalid file
- 5.3.4. Provider successful request to view provider directory
 - 5.3.4.1. Provider successfully able to view provider directory
- 5.3.5. Provider unsuccessful request to view provider directory
 - 5.3.5.1. Provider attempts to display empty provider directory
 - 5.3.5.2. Provider unable to display provider directory
- 5.3.6. Provider successful request to export provider directory
 - 5.3.6.1. provider successfully writes provider directory to file
- 5.3.7. Provider unsuccessful request to export provider directory
 - 5.3.7.1. Provider attempts to write empty provider directory to file
 - 5.3.7.2. Provider attempts to write provider directory to invalid file

5.4 Provider Renders Service to Member

- 5.4.1. Provider successfully renders service to a member
 - 5.4.1.1. Provider renders valid service to valid member
- 5.4.2. Provider adds rendered service with invalid service code
 - 5.4.2.1. Provider adds rendered service code with wrong format
 - 5.4.2.2. Provider adds rendered service with empty service code
 - 5.4.2.3. Provider adds rendered service with mismatched service code
- 5.4.3. Provider adds rendered service with invalid service date
 - 5.4.3.1. Provider adds rendered service with empty date
 - 5.4.3.2. Provider adds rendered service outside possible dates
 - 5.4.3.3. Provider adds rendered service with invalid date format
- 5.4.4. Provider adds rendered service with invalid service fee
 - 5.4.4.1. Provider adds rendered service with empty fee
 - 5.4.4.2. Provider adds rendered service with invalid fee format
 - 5.4.4.3. Provider adds rendered service with too large of a fee

5.4.5. Provider adds rendered service to invalid member ID

5.4.5.1. Provider adds rendered service with empty member ID

5.4.5.2. Provider adds rendered service with invalid member ID format

5.4.5.3. Provider adds rendered service with too long of a member ID

5.4.6. Provider adds rendered service to invalid member name

5.4.6.1. Provider adds rendered service with empty member name

5.4.6.1. Provider adds rendered service with too long of member name

5.5 Manager Runs EFT Report

5.5.1 Manager successfully runs EFT report

5.5.1.1. Manager writes EFT report to file

5.5.2 Manager unsuccessfully runs EFT report

5.5.2.1. Manager attempts to write empty EFT report to file

5.5.2.2. Manager attempts to write EFT report to invalid file

5.6 Manager Runs Member Reports

5.6.1 Manager successfully runs member report

5.6.1.1. Manager writes member report for week to file

5.6.2 Manager unsuccessfully runs member report

5.6.2.1. Manager attempts to write member report for week to invalid file

5.6.2.2. Manager attempts to write empty weekly member report to file

5.7 Manager runs Provider Reports

5.7.1 Manager successfully runs provider report

5.7.1.1. Manager writes provider report for week to file

5.7.2 Manager unsuccessfully runs provider report

5.7.2.1. Manager attempts to write weekly provider report to invalid file

5.7.2.1. Manager attempts to write empty weekly provider report to file

5.8 Manager Removes Member

5.8.1 Manager enters existing member ID

5.8.2 Manager enters member ID that does not exist

5.8.3 Manager enters member ID of invalid length

5.8.4 Manager enters invalid member ID format

5.9 Manager Removes Provider

5.9.1 Manager enters existing member ID

5.9.2 Manager enters member ID that does not exist

5.9.3 Manager enters member ID of invalid length

5.9.4 Manager enters invalid member ID format

5.10 Manager Edits a Member

5.10.1 Manager enters valid member ID

5.10.2 Manager enters invalid member ID

5.10.2.1 Manager enters member ID that does not exist

5.10.2.2 Manager enters member ID of invalid length

5.10.2.3 Manager enters invalid member ID format

5.11 Manager Edits a Provider

5.11.1 Manager enters valid provider ID

5.11.2 Manager enters invalid member ID

5.11.2.1 Managers enters provider ID that does not exist

5.11.2.2 Manager enters provider ID of invalid length

5.11.2.3 Manager enters invalid provider ID format