The ChocoAn Simulator

Requirements Document

Table of Contents

1 Introduction

This document introduces the fundamentals aspects and planning for the Chocoholics Anonymous software development. Our goal with this document is to provide the readers with an overview of the software that we are creating, whom we are creating it for, who will be using the software, its required (non)functional requirements, and all the important milestones and deliverables. This document will provide the readers with all the important information needed to understand how our Chocoholics Anonymous software will work.

1.1 Purpose and Scope

The purpose of this document is to lay out the foundational aspects of the Chocoholics Anonymous software and break it down into its most critical components. The document will take you through step by step what the software requirements are from the organization and highlight our intent to meet them. This document will not include any technical information, but more of a broad overview of requirements and core functionality.

1.2 Target Audience

The main audience of this document is those who want to know how to develop software and how to design software. In addition, it also provides ideas for learning and development, so it is suitable for anyone who wants to learn from the basics.

1.3 Terms and Definitions

The terms and definitions will be put here in advance. To ensure the readability of subsequent articles.

- ChocAn
  - Chocoholics_Anonymous (Usually abbreviated to ChocAn)
- Provider
  - Usually on behalf of management
- EFT
  - Electronic_funds_transfer(Usually abbreviated to EFT)
- Service_Code
  - Code that is automatically generated when a user uses a service.(6 digits)

2 Product Overview

We are developing software to help people who are addicted to all kinds of chocolate reduce their chocolate consumption. Software development is primarily about users and development companies. Our software is members-only and offers a variety of online services.

This part of the document will introduce the information about the user of the program and our client. It will highlight the role of the stakeholder in the development or maintenance of the software. Additionally, we will include how the user is going to use the system product.

2.1 Users and Stakeholders

We will list out the stakeholders and the users for the product which will consist of our client, Chocoholics Anonymous, the members, and the providers. We will provide information on Chocoholics Anonymous's role in the development of the software. In addition, we will list out what information from the system will be available for the user like the patient and the provider.

2.1.1 Chocoholics Anonymous

Chocoholics Anonymous is our main customer and we must satisfy their requirements to make a deliverable product.

- **2.1 Use cases** - How the user is going to use the product and get service
  - Patient(member)
    - Get membership information
    - Provide service date
    - Get the six-digit service code
    - Access to service
  - Patient(Nonmember)
    - Failed to get member information
    - Apply for membership
    - Get membership information
    - Provide service date
    - Get the six-digit service code
    - Access to service

2.1.2 Provider

- **2.1 Use cases** - How the user is going to use the product and get service
  - Provider (Service Fee)
    - Provider(Service fee)
    - Provide service

    * Get service fee
- Patient (Membership Fee)
    * Provider (Membership fee)
    * Provides membership services, all servers are used only by members.
    * Charge for membership through EFT.

3 Functional

This section serves to outline the functional requirements for the Chocoholics Anonymous software application. It will highlight the core functionality of the software and break down the application into major tasks that will largely contribute to how the software works.

3.1 Member Validation

The first functional requirement for the software is to be able to validate the member's member code, then input the member's service code for their care. This in turn will allow the provider to access the member's file and annotate all of the critical information regarding their specific care event and store it on a disk.

3.1.1 Prompt User

One of the most important aspects of this functional requirement is for the software to be able to prompt the member for their member code, read it in, and then verify it. The software will be able to determine the member's status in the system with their member code, such as;
(Member Suspended, Invalid Number, Validated)
Once the member's code is validated, all logged information will be added to that member's account. The provider will then log the date that the care was rendered.

3.1.2 Provider Directory

The second key component to this functional requirement is the ability for the provider to reference a Provider Directory in order to determine the service code for the service that was provided to the member. The provider will input the corresponding service code and the system will verify it and display service back to the provider for additional verification and it will be added to the member's account.

3.1.3 Additional Comments

Lastly for this component, the software will allow the provider to enter any additional comments specific to the care that was given to the member and then all the information about that care event will be recorded onto a disk in the following format:

- Current date and time (MM-DD-YYYY HH:MM:SS)
- Date service was provided (MM-DD-YYYY)
- Provider number (9 digits)
- Member number (9 digits)
- Service code (6 digits)
- Comments (100 characters) (optional)

3.2 Provide Information

The main function of this function is to collect information and display all information on the display. The provision of information is mainly divided into three parts: Chocoholics Anonymous, Provider and EFT. All data stores should be in, invoked and used from the database.

3.2.1 Collecting Information

Functions that need to provide information necessarily need functions that collect information, so the first aspect is to collect different data from different databases. At the same time, some data processing is needed in this process.

3.2.2 Information Output

The content provided for different requirements is different. Therefore, different output interfaces need to be provided for different requirements.

The output interface is divided into three types: User interface, management interface, and integrated information interface.

- User Interface
  - Current date and time (MM-DD-YYYY HH:MM:SS).
  - Date service was provided (MM-DD-YYYY).
  - Provider number (9 digits).
  - Member number (9 digits).
  - Service code (6 digits).
  - Comments (100 characters) (optional).
- Management interface
  - 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).

- Integrated information interface
  - 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 $99,999.99$)

All output is output through these output functions.

3.2.3 User Interface

For users, a separate interface is required to display the time of submitting tasks and required services. Therefore, a separate interface is required to ensure user data input and system feedback to user input information.

3.2.4 Provider Interface

For providers, we also need different interfaces to ensure that information is different from that of the user, so separate data input is required to ensure that the provider has enough information to flag the user.

3.2.5 EFT

For EFT, we still need some functions to ensure the correct access to EFT charges and the normal payment of membership fees. You don't need to provide any data interface, but you do need to ensure that the database information is up to date.

3.3 Billing

Because billing is in everyone's interest, we need to separate billing to ensure the security of the information. All charges fall into two main categories: membership fees and service fees.

3.3.1 Membership Fee

As for the membership fee, our fee does not need to be reflected in the weekly bill, but we need to have a separate email reminder and judge whether to extend the membership validity through EFT. Membership fees should be independent and separate from service fees. Be sure to count your profits separately each month. Instead of being included in the fee along with the service charge. We need separate parameters to ensure the independence of membership fees.

3.3.2 Service Charges

For service charges, because different types of charges are involved, a total charge variable is required, as well as specific data for other detailed charges. Specific data content is collected by the EFT database.In addition, because billing data is private, we must ensure the security of the data when we receive it.

4 Non-Functional Requirements

Non-functional requirements are constraints on the functionality of the system. Although the system could still work without meeting the requirements, it may not meet the customer's expectations. Therefore, it leads to an incomplete deliverable product.

4.1 Security

In order to follow HIPAA guidelines, the system must ensure that there are no information leaks about the patients. The system must ensure that only the correct people have access to sensitive information about the patient. This could be programmatically done by encrypting patient information. This ensures that only the right people have access to the data.

4.1.1 Time Authentication

First of all, calling the system time can help us avoid many unnecessary errors. For example, we can obtain the correct range of birth ages by the time, instead of random numbers. This reduces runtime errors due to incorrect birthdays and ages.

Second, through time authentication, we can better confirm whether the service time is consistent with the time provided by the user. In this way, we can verify that the consumer is using the service correctly.

Finally, time verification can also avoid the problem that people in different regions use the same account (Prevent account and password sharing problems).

4.2 Scalability

This system will most likely have to be upgraded in the future and be implemented with other systems. We will have to make the system modular enough so it still functions while changes are being made.

- The system should not break when a function is taken out
- The system will need to be able to be ready to be upgraded
  - If the system is not scalable, then there will be less incentive to purchase this system compared to other ones, as it is not future-proofed.

4.2.1 Dynamic Data

We will store the data dynamically to save space and only use the necessary space. It will take longer to access a certain patient's data; however, it will be more efficient for saving memory. Additionally, it will be easier to add more patients to the database. It will be important to know the cost of the storage space to know if it's more cost-effective to use static memory vs dynamic storage.

4.2.2 Extensibility

Software scalability architecture design, the ability to sustainably expand and improve system functions with minimal impact on the existing system. For future upgrades to the system, our software will still be usable and stable. This can be measured by the system uptime.

4.2.3 Low coupling

Use low coupling to ensure that the code has more space to add new functionality at a later stage. We also use a simpler and more stable interface to interact with another module so that future code can be extended more easily without having to change more core code.

4.3 Ease of Use

We want our software to be easy to use and access for all types of users because the audience is in healthcare. Because everyone will be using this system, we will need to make sure that it is minimal and simplistic enough while not sacrificing it's functionality. To achieve this, it means that the system will need to have an intuitive user interface and satisfactory user experience.

4.3.1 User Interface

We will design an intuitive layout for the providers and patients to use. It will utilize data abstraction so users will use the program as intended without thinking too much about the mechanism behind it. This will be measured by how long it will take for the users to learn how to use the system. The user interface is highlighted in section 3.2.1. This makes sure that the system is intuitive and simple enough for the user to use.

4.3.2 User Experience

For Chocoholics Anonymous to continue to use our system, we will make the system a good experience for all users. This can be measured by the number of users that contact us for support for the software, and the number of users that initially use the software versus the number of users that want to continue using the software.

5 Milestones and Deliverables

This section's purpose is to describe each milestone that we want to achieve in a certain amount of time. In addition, this section will highlight our work plan and timelines that we need to complete to efficiently achieve each milestone. By following each milestone and work plan, we will create software that will meet Chocoholics Anonymous's requirements within the deadline. Our milestones will be mostly code-based.

5.1 Milestone/Deliverable 1

Within the deadline, as a group, we will review the product's requirements and description. By separating each part of the requirements into chunks within time, we will be able to focus on one part at a time and focus on creating efficient and effective software. The plan consists of what database we will use and how we will approach each requirement or milestone. In addition, we will identify the stakeholder and users of the system, and we will include the functional and non-functional requirements in our requirement document.

- Requirement Document Due: October 25, 2021

5.2 Milestone/Deliverable 2

We will deliver a design document that will discuss the description of the system to allow us to understand clearly what needs to be built into our software. The document will consist of design considerations, system overviews, and detailed system design and architecture.

- Design Document Due: November 1, 2021

5.2.1 Milestone/Deliverable 2 Stage 1

Firstly, the milestone we would like to achieve within the two weeks after planning would be implementing a system that shall determine whether to validate the service code for the customer to receive health service with ChocAn. If the service code, our software shall implement an error message that will inform the provider that the service code is invalid. Otherwise, if the card can validate the service code, the system shall collect the information needed to record to the disk-like current date and time, date service, provider and member number, service code, and comments. In addition, the system should be able to collect multiple service codes since there is a possibility that the customer can receive multiple services with ChocAn. Also, the information should be limited by a certain amount of characters or digits as well.

- Validate the service code (October 25 - November 1)

5.3 Milestone/Deliverable 3

After working on most of the system, we will deliver a test plan where it will consist of details of testing the system like unit testing, smoke testing, and system testing. We will also include our test plan description like our schedule and criteria.

- Test Plan Due: November 15th

5.3.1 Milestone/Deliverable 3 Stage 1

After completing the first milestone, we would additionally add where the system would reveal the information for each member that consulted a ChocAn provider through an email. The information should consist of the member's name, number, address, city, state, zip code, and the services that were provided to them. In addition, the services that were provided also include more information such as date of service, provider name, and service name. By implementing the system, the system shall organize the services' information by the date. For error checking, we also need to check if the information fits with the number of characters that ChocAn requested, like for the member's state it should be two characters.

- Email Attachment Section (November 1 - November 15)

5.4 Milestone/Deliverable 4

- Project Report and Final Deliverables: December 3, 2021

At this deadline, we would present the final product to the client with a project report and final deliverables. The final product should meet the requirements that Chocoholics Anonymous requested, and be able to display certain information for the members and the providers.

5.5 Milestone/Deliverable 4 Stage 1

Lastly, our last milestone for coding is generating the bill for each service that the provider provided. The information should consist of the provider's name, number, address, city, state, zip code, the total number of consultations with members, and total fee. In addition, the system details each service that the provider provided including the data that we collected before like date of service and member name, and the provider must input the fee for each member. To make this part of the system efficient, we have to take previous data from other parts of the system to make the provider's job much easier rather than the provider typing it all over again. By collecting other data from other milestones, it would improve performance and efficiency for the system and satisfy our client's needs and requirements. In the end, the system should meet all the requirements that Chocoholics Anonymous requested. The system shall collect the information from the member and the provider effectively. Also, be able to generate a bill to show the fees for the members and the week.

- Generate Bill Section (November 15 - December 3)