Design for the Chocoholic Anonymous System

A data processing system to support membership and services provided to the chocolate dependent population.

CS 300 PSU WINTER 20’

Professor Chris Gilmore

Group 1

Christian Andersen

Darika Batbayar

James Bao

Minjin Enkhjargal

Yan Li

Deep Patel

Tenzing Rither

**Term Project: *ChocAn***

Design Document

**Table of Contents**

**1** **Introduction 2**

*1.1* *Purpose and Scope 2*

*1.2* *Target Audience 2*

*1.3* *Terms and Definitions 3*

**2** **Design Considerations 4**

*2.1* *Constraints and Dependencies 4*

*2.2* *Methodology 5*

**3** **System Overview 6-8**

**4** **System Architecture 9**

*4.1* *Logical View of The ChocAn Architecture 10*

*4.2* *UML diagram for ChocAn System 11*

*4.3* *Layered System Architecture 12-16*

*4.4 Database System Interface 16*

**5** **Detailed System Design 17**

*5.1* *Admin Subsystem 17*

5.1.1 *Assisting with data structure* 17

5.1.2 *Reading and writing to disk* 18

5.1.3 *Making reports* 18

*5.2* *Member Subsystem 18*

*5.2.1 Store HIPAA Information 18*

*5.2.2 Add history to data structure 18*

*5.3* *Provider Subsystem 20*

5.3.1 *Confirm membership* 20

5.3.2 *Recording Services* 20

**1** **Introduction**

The design document will cover details about the design and structure of the ChocAn system. Throughout the document there are multiple sections covering multiple aspects about the ChocAn system, the following are brief summaries for each section. The Introduction will cover the purpose of this document, important terms, and to whom this documentation is addressing. The Design considerations will discuss the constraints, dependencies and methods for the ChocAn system. The System overview and architecture section will provide a broad understanding of the system. Finally, the detailed system design section will cover the specifications of the design for the software.

**1.1 Purpose and Scope**

The design document will go over the ChocAn system and important design details to consider when moving forward with the project. The structure and design will be discussed at a high level at this point in the design.

**1.2 Target Audience**

The target audience for this document is the ChocAn team of engineers and stakeholders. The engineers will benefit from the specifications of the design, in order to gain a better understanding about the functionality of the software. The stakeholders will be able to also have a high level overview of the structure of the project, so they can oversee the plan to make sure it meets customer expectations.

**1.3 Terms and Definitions**

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| Constraints | Limitations to consider for the project. |
| Dependencies | What dependent parts of the system will affect the project. |
| ChocAn Data Center | Data center responsible for processing member’s information. |
| System | ChocAn system. |
| ChocAn | Chocoholics Anonymous is an organization committed to helping people with chocolate addiction. |
| Interface | Layout and structure of the system’s interface. |

**Figure 1**. *Table of terms and definitions for ChocAn.*

**2 Design Considerations**

The design considerations will go over the constraints and dependencies our software needs to be aware of. The methodology our team will implement will also be discussed in this section.

**2.1 Constraints and Dependencies**

There were a couple of constraints and dependencies that the project needed to be aware of and adhere to. The system has functionalities and different tasks that need to be executed. The ChocAn is essentially a large database-interface software that needs to be able to store information. There are some constraints to be aware of, such as the format and number of characters/digits that each record information has to be able to hold. Enough memory must be allocated for each information. For example, the system must account for the enough characters for a member with a long name, or something similar.

Some dependencies that affect how our design is built is the scalability factor of the database. The number of active users of the system must be considered when creating a design. The data structures that have been decided for the program have been chosen to accommodate the varying number of members and providers, if they are added or removed from the system.

The design also must depend on making the database and patient information as secure as possible. The method of data abstraction is used to maintain privacy of the user’s data and sensitive information.

**2.2 Methodology**

The software engineering method that will be used for the development of the software will be the Waterfall method and some Agile methods during the implementation and testing phases.

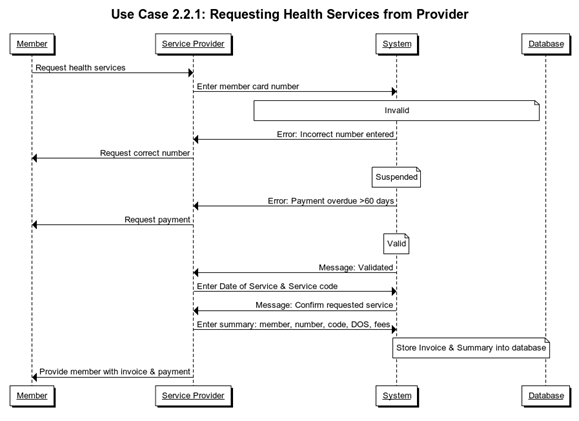
Waterfall is a linear approach to development, which will be used in the development of the ChocAn software. This includes strict planning and going through the plan step by step for the requirements specifications, design and evolution phases. Using the Waterfall method in these phases will allow easier management of the project. Each phase has specific deliverables and a review process which allows for each stage to be well planned out. This allows for processes and results to be well-documented, which will allow work to be completed carefully and completely. Additionally, since the client for the ChocAn system requires little to no communication, there is no need for a lot of client involvement. The client has provided an in-depth system project requirements document, which has been used to design the plan the system project. Thus, a lot of change and potential re-work is not expected. To follow this method for these phases, the development team has created a requirements specification and design document for the project.

The Agile method will mostly be used in the implementation, testing (validation and verification) phases. In these phases, it is important to implement the code for the system and continuously test to ensure that the system is on track to follow the specifications. This will allow a small-knit team to work closely. Since the ChocAn development team is a small team of 7 people, there will be close meetings and discussions to divide up the work and coding section.

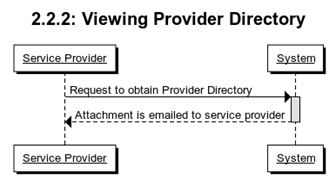
**3 System Overview**

Chocoholics Anonymous (ChocAn) data processing system helps to organize, collect, store and deliver information for the ChocAn organization. The system utilizes the special identification numbers to manage services and fees for each member and provider. The data is stored using external files at the ChocAn Data Center for constant updates of member and provider records. A summary of a list of services and fees are processed weekly. The payment management for ChocAn membership is handled by Acme, a third party organization, updates the membership records every evening. All other fees are managed as electronic transfer funds data to banking computers for crediting the providers.

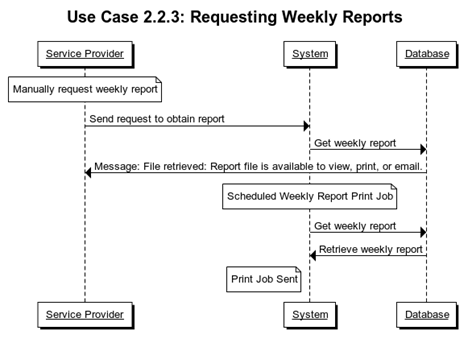
The ChocAn data processing system is implemented using the Java language. It is developed with an object-oriented programming approach. It should provide good maintainability if new changes are warranted.

****

**Figure 3.2.1:** *This is an overview of how the system works when the members request health services from their providers. The identification number must be validated for any ChocAn services.*

.****

**Figure 3.2.2:** *Service provider can request software product for a Provider Directory. The Provider Directory is sent to the provider as an email attachment.*

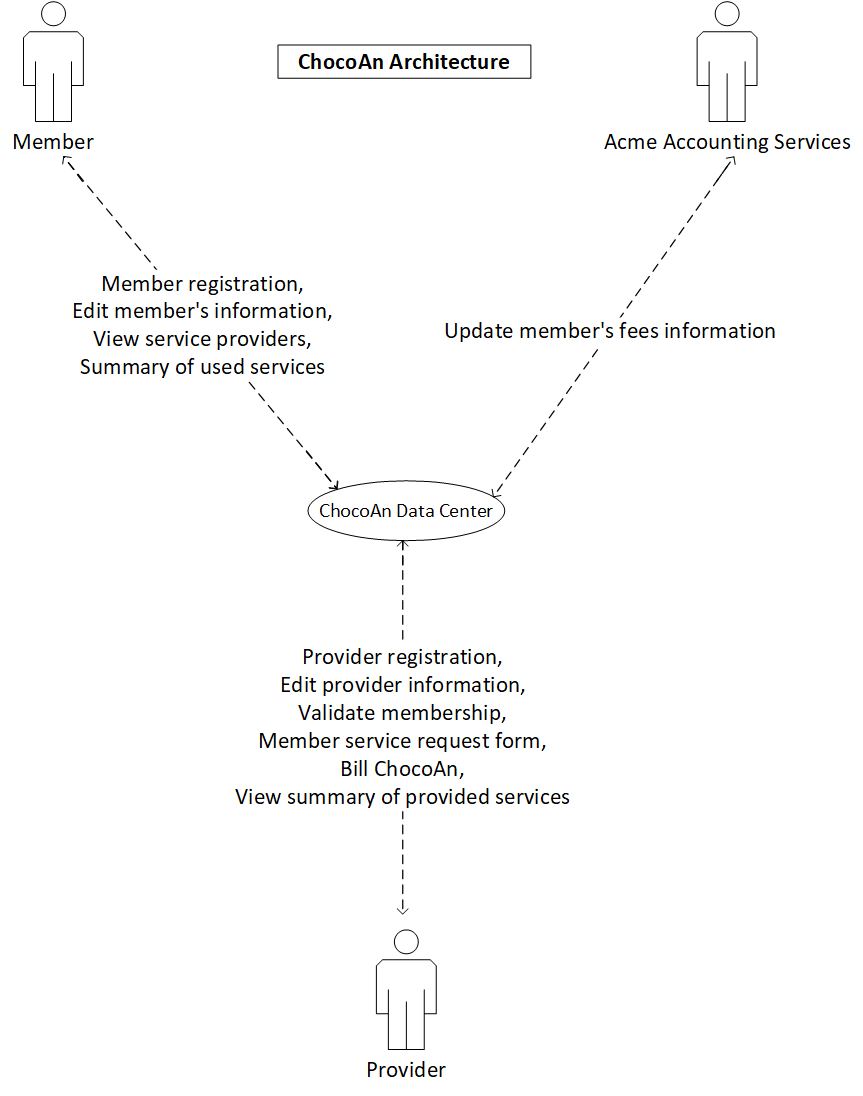
****

**Figure 3.2.3:** *Service providers can retrieve weekly reports from the system. The reports are generated and saved automatically weekly in the system.*

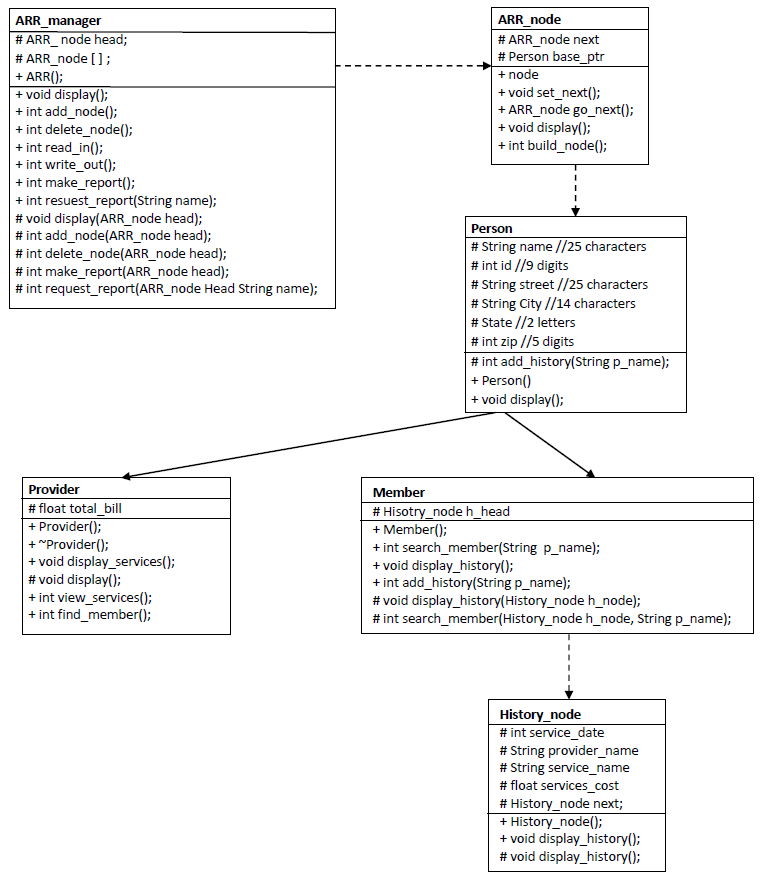
**4 System Architecture**

In the system architecture section, the structure and organization of the ChocAn system’s architecture will be discussed. A layered system design will be implemented, and a UML is provided for an overview of the system. The systems and subsystems that will be further explained in this section include: the information system, the member system, the acme system, the ChocAn system, ChocAn manager system, and the data collection system.

**4.1 Logical View of The ChocAn Architecture**



**4.2 UML Diagram for ChocAn System**



In the ChocAn system, a person is a virtual abstract base class. Member and provider classes are derived from the person class. The developer will use the single inheritance methodology and object oriented approach to provide functionality. The member has history\_node class which only displays the history of member’s used services. Provider class can view members by typing their id, charge members for used services, and total amount owed from members, and verify amounts has been paid for services used by the members from ChocAn. Member class can view member information and contain the history node. So when the provider enters the information. It will be stored in member class.

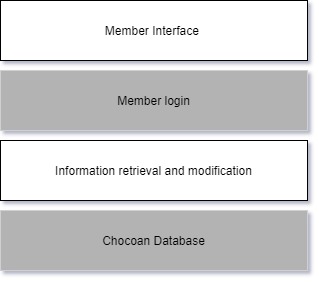
The person class is abstract base class which provides dynamic binding features with all common utilities that has been implemented in all the derived classes throughout the hierarchy. Some other responsibility of person class will be to add members, provider, and update member and provider information.

ChocAn developers use an array of linear link lists as data structure to manage the database system. All the data will be stored as text file. ARR class will manage the data with help of the node class. ARR class contains the node class and node class contains the personal class.

ARR class will act as a hub of ChocAn data center. It will allow providers, members and acme to call specific services based on their roles. Which will call the specific function from the backend at runtime.

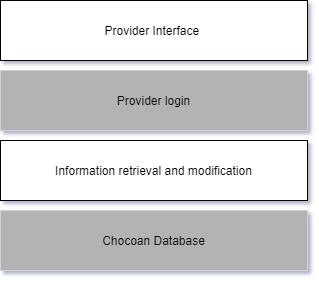
**4.3 Layered System Architecture**

Layered information system architecture design provides a simple and secure interface for ChocAn members. They have their ChocAn member card that encrypts important personal information. The members give their member cards to the provider to check for validation through a terminal. The members receive a summary report sent as an e-mail that concludes the week’s list of services. All the records are stored safely in the ChocAn database.

****

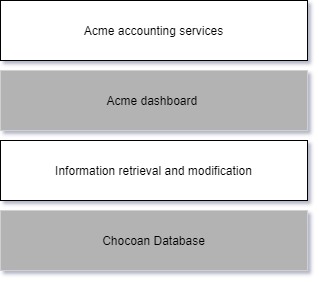
**Figure 4.3.1:** *Member layered system architecture*

Providers can log into their home screen using provider id. It allows providers to see the dashboard that applies to id. Provider can use the feature by entering the specific single character key that will be provided by the ChocAn system. Providers can fill in the information about the services they are providing to the members, can check the total amount owed to members, modify the provider information and check the provider directory.



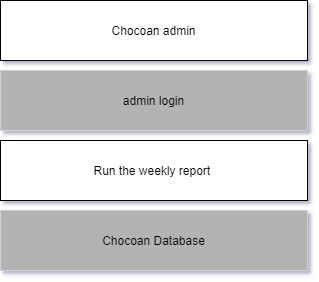
**Figure 4.3.2:** *Member layered system architecture*

An Acme person can log in with their single id code that will allow them to go to the acme dashboard. An Acme person will only be able to access membership fees information and allow them to change membership status.



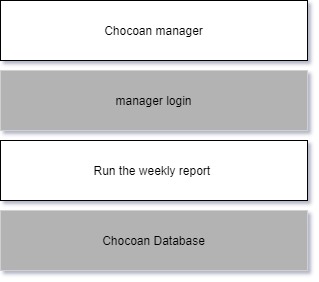
**Figure 4.3.3:** *acme layered system architecture*

ChocAn admin is authorized person of the ChocAn data center to run the weekly billing reports that are sent out to members. ChocAn admin will provide a key to log into their system. The interface allows them to run the report weekly. hcocoan admin can verify if the member has been paid the amount into the bank.

****

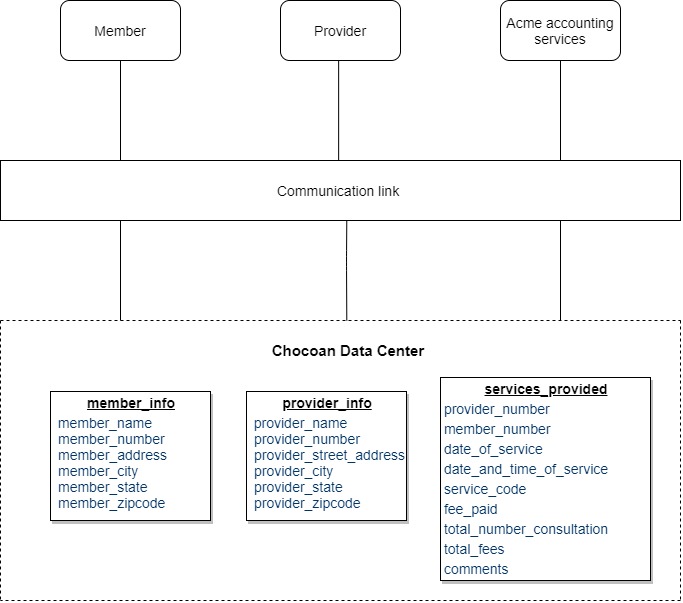
**Figure 4.3.4:** *ChocAn admin layered system architecture*

ChocAn managers have a unique key to log into the system. It will allow them to check the summary of members' visits and total fee of all service received and provided by the particular member the management is interested in.

****

**Figure 4.3.5:** *ChocAn manager layered system architecture*

**4.4 Database system interface**



**5 Detailed System Design**

The ChocAn Data Processing System is discussed in a highly detailed manner in this section. All the classes and data structures to store the database of members and providers of ChocAn are defined here.

The system consists of three main components to organize the flow of information to store the members and providers information. The design uses a linear linked list data structure to maximize the efficiency of finding an individual patient or member within the database. To maximize code efficiency, both patient and member objects are designed as derived classes of a virtual “person” base class. “Person”-type objects will be instantiated in the “add node” helper function, but will be “cast” as either a “patient” or a “member” using dynamic binding.

**5.1 Admin Subsystem**

The admin component of the system is found in the main ARR\_manager class and it holds onto the entire data structure that organizes the database as well as simulating operators at ChocAn data center who edit the members list and providers list. The admin system is a critical system because the object of this class is created in the main class in order to interact with the user and allow access to the database in a secure manner. This class helps the programmer to create an abstraction and hide all the private information behind a single interface. Due to the complexity of the class, the functionalities are further broken down into the following subcomponents:

*5.1.1 Assisting with data structure* The ARR\_manager has a protected data member called “head” which is used as a starting point for the data structure. This class creates an array of size two. At index 0 of the array, the members are stored in a linear linked list and at index 1, the providers are stored in another linear linked. The index acts as a head pointer to each linear linked list and ties the two different subsystems that are mentioned later in the design.

*5.1.2 Reading and writing to disk* In this program, disks are external text files in .txt format. The database that holds onto the entire list of members and providers and their associated information are all stored in a text file. The ARR\_manager class has a read\_in() function that goes and finds the text file and then reads in the information into the data structure by assigning variables to each data specified in other classes this class points to. When the operators at ChocAn add or remove a member or a provider, this class also has a write\_out() function that updates the text file so the current database stays up to date to simulate a permanent change done on disk.

*5.1.3 Making reports* The last subcomponent of the Admin System is to create reports. The datacenter runs a weekly scan and creates reports for members that received some type of services. The make\_report() function looks at the entire members’ list and creates a separate report for each member with all the service names as well as the cost associated with it. If a manager at ChocAn data centers requests a report to be made for a certain member, the request\_report() function is used to take the name of the member and then searches in the database to find all history associated with that specific member.

**5.2 Member Subsystem**

Each member in the linear linked list at index 0 has a head pointer to point to another linked list of services they receive. If a member uses a specific service, the name of the service is recorded and gets time stamped by the CPU of the computer and stored in a history\_node. This secondary structure is organized in a chronological order with the most recent services added at the end of the list to create a queue.

*5.2.1 Store HIPAA Information* When a new member joins the ChocAn organization, a new account is created for the member along with all the health personal information. The “add\_history” data member in the Person class, whether it’s for adding a member or a provider, is protected. This only allows access of this stored information only if there’s a helper function or through inheritance. The Member class is a child of the Person class, so it stores the member’s HIPPA information securely through Person class. With each member created, a member profile is stored in an ARR\_node. A list of members is stored as a linear linked list.

*5.2.2 Add history to data structure* In the member class, there is a protected data member called “h\_head” which is a pointer to a linear linked list of health service history. Each node in the linear linked lists contains a particular service received by the member that day. The add\_history(String name) function is utilized to add service to the history for the member which is ordered by alphabetical order. Also, the history of a particular member can be retrieved by the member name via the search function.

**5.3 Provider Subsystem**

The provider component of the system is found as provider class. The provider will work as a class that displays provider information. The node class will manage the provider class. The node class will help to create a linear link list of provider directory. There are two main subcomponents of the provider: confirmation membership and recording services.

*5.3.1 Confirm membership* the membership\_verification() function in the ARR\_ class will allow providers to verify that the member exists in the ChocAn

system. find\_member() function will find the member from within the array of linear link list data structure. find\_member() function will be useful when providers want to add the details of the used services.

*5.3.2 Recording Services* member class contains an array that contains a hash function. After the membership is confirmed and a new member is added to the database if necessary, the record\_services() function in the ARR\_class will use a hash function to store the service in the services\_history array object contained in the provider class. This way, all of the services within the database are organized by care providers and can be indexed according to the time, location, and member name. The hash function will use the time, location, and member name to assign a position for each service.