The ChocAn Data Center

Design Document

1 Introduction

In this document we (the developers) provide a detailed description of the system that makes up the ChocAn Data Center. We begin by outlining the design considerations and requirements informing system architecture, and describing the methodologies to be used in the implementation of the system. We then offer a broad view of the system itself, and expand on the shape of the subsystems that make up the whole. Following this we examine the subsystems themselves and explain the architecture underlying them and informing their role in the greater system. Finally we identify the ground level actors in the system and account for their purposes.

1.1 Purpose and Scope

This document offers a detailed characterization of the system comprising the ChocAn Data Center. It also serves as documentation of the development of said system, integral both during development itself and long afterward, if and when maintenance is required.

The scope of the document will mirror the scope of the Data Center itself. The system described in the submitted Requirements Document will be covered here in great detail, at every level from the six-digit service codes used by ChocAn service providers to the system as a whole. But beyond logging dollar amounts of the fees billed to ChocAn for services rendered its members, this document has nothing to say about financial particulars, communications, hardware specifications, security, or any operations of ChocAn, its members or its service providers.

1.2 Target Audience

This document is written for the development team charged with implementing the ChocAn Data Center, and for the future contractors, engineers and data scientists who are tasked with long-term use and maintenance of the system.

1.3 Terms and Definitions

1.3.0 ChocAn Simulator:

The full software suite which the end-user will experience and interact with, and to which the ChocAn Data Center provides its functional services. This is not part of the scope of the software to be described by this document.

1.3.1 ChocAn Data Center:

The software application underlying the ChocAn Simulator. The design of the ChocAn Data Center is the subject of this document.

1.3.2 Member:

A subscriber of ChocAn who pays the organization’s required monthly fee and obtains access to services from providers recognized by ChocAn.

1.3.3 Provider:

A health care professional who provides services (ref: sec 1.3.8) for ChocAn members. These providers can be dieticians, internists, exercise specialists, addiction treatment specialists or other care entities recognized by ChocAn.

1.3.4 Manager:

An individual responsible for ChocAn operations. They can request the report for any member or provider as well as the manager’s report at any time through the manager terminal. In interactive mode, they can also add, remove, and edit the records associated with members and providers.

1.3.5 Operator:

A data abstraction that forms the base of the derived classes ‘manager’ and ‘user’.

1.3.6 User:

A data abstraction that forms the base of the derived classes ‘provider’ and ‘member’.

1.3.7 Terminal:

The interface used by an end-user to interact with the ChocAn Simulator. Terminals are not within the scope of this design document.

1.3.5.1 Provider terminal: The terminal to be used only by providers.

1.3.5.2 Manager terminal: The terminal to be used only by managers.

1.3.8 I.D. Number:

A 9-digit identification number associated with a ChocAn-affiliated entity.

1.3.6.1 Member number: The I.D. number associated with a member.

1.3.6.2 Provider number: The I.D. number associated with a provider.

1.3.9 I.D. Number Status:

The status associated with verifying an I.D. number.

1.3.7.1 Validated: A validated I.D. number can be used to access an account or log a service.

1.3.7.2 Invalid: An incorrectly formulated or unused I.D. number is considered invalid.

1.3.7.3 Suspended: A suspended I.D. number is associated with a member who has been suspended (e.g. by having dues outstanding with ChocAn).

1.3.10 Service:

A consultation or treatment performed for a ChocAn member by a recognized provider that can be billed to ChocAn.

1.3.8.1 Service code: The 6-digit code ChocAn associates with a predefined service.

1.3.8.2 Service list: A list of services provided that week. Service lists are data fields of both Member and Provider classes.

1.3.11 Provider Directory:

A list of service codes and their associated service names and fees, provided in a file format. The provider directory does not include information associated with ChocAn providers.

1.3.12 Interactive Mode:

An operating mode of the ChocAn Data Center that offers a manager additional functionality to modify service data and providers' and members' records.

1.3.13 Member Report:

A summary of the services obtained by a member over the past week.

1.3.14 Provider Report:

A summary of the services rendered members by a certain provider over the past week.

1.3.15 Summary Report:

A compendium of the providers and associated total fees to be paid by ChocAn over the past week.

2 Design Considerations

This system is intended to be a simulator to interact with the ChocAn Data Center. We will be designing the software that handles the input manipulation and reporting of data entries in the ChocAn Data Center.

2.1 Constraints and Dependencies

We are only developing the database entry software. The user terminals will be designed and developed by an outside firm. Further, ChocAn has an accounting firm which handles the charging of members and crediting providers.

2.2 Methodology

We are using the waterfall development method because this project has unchanging requirements and a solidified goal. This means that we don't have to worry about changes as we are developing and implementing this product. Also the clear structure of waterfall allows for efficient use of time. We implemented the waterfall development method by creating a requirements document first that was approved before we began designing the software in a document. Before continuing to implement the design, we laid out a test-plan for the system and each component.

3 System Overview

The system of the ChocAn Data Center is best described as a set of seven objects: **Operator**, **Manager**, **User**, **Provider**, **Member**, **Service**, and **ID**. The first five of these form a hierarchy, and the remaining two are contained or used by those in the hierarchy. All seven objects are described in broad terms in this section, and defined in greater detail in subsequent sections.

Note that the terms used in this section refer to the data abstractions within the software system, not their physical counterparts. When we say that a “Member” has the capability to generate a member report, or that a “Provider” is able to edit provider data, we mean that those methods are implemented within the Member and Provider classes, respectively. A “real life” member of ChocAn is *not* a user of the ChocAn Data Center, and ChocAn service providers do not have access to the database except when logging services.

3.1 System Hierarchy

The bulk of the system is built in a class hierarchy (see fig. *1*), at the top of which is the Operator base class. An Operator class object contains identification information and the ability to pass that information to other objects. Derived from the Operator are two subclasses: Manager and User.

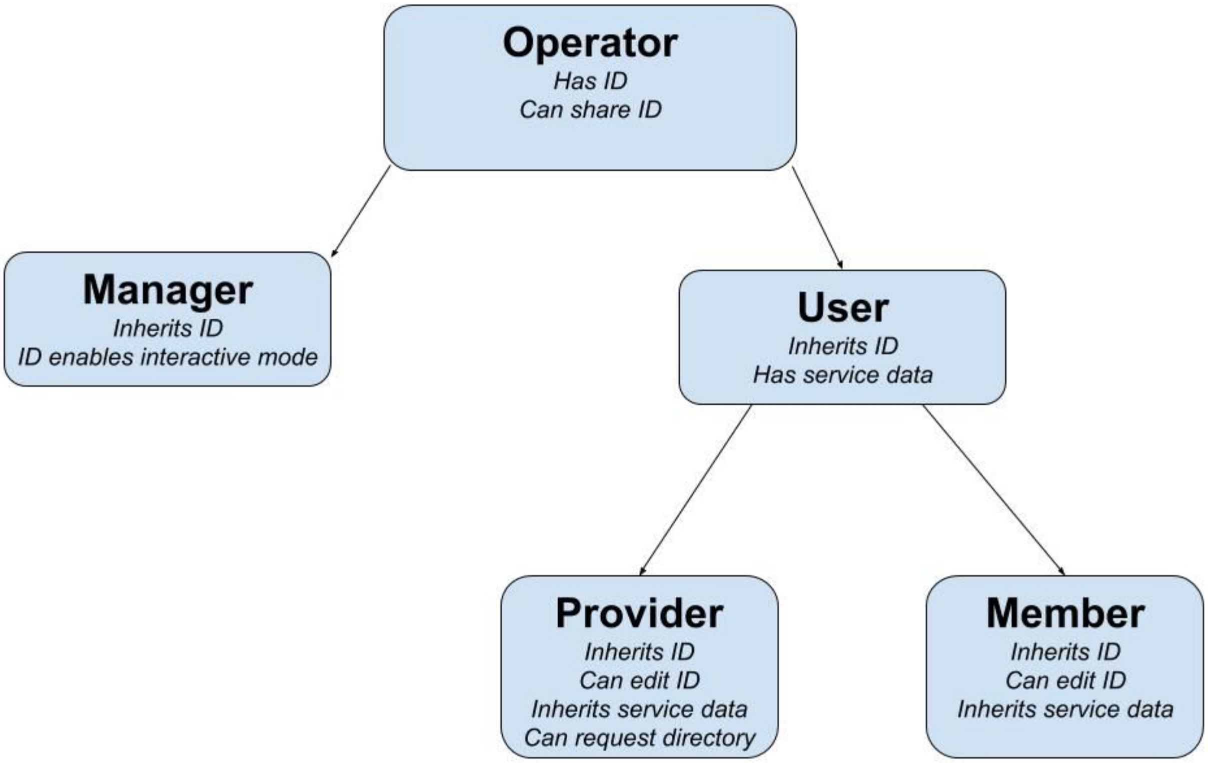


Figure 1: the class hierarchy comprising the majority of the major objects in the system.

The User class is further abstracted via two additional derived classes: Member and Provider.

Again, the reader is advised not to take the names ‘Operator’ and ‘User’ at face value. Members do not ‘operate’ or ‘use’ the system beyond giving their member I.D. to a provider who uses the provider terminal. But Members exist as important objects in the class hierarchy.

Similarly, Manager objects derived from the Operator class don't need to do much, even though managers are the primary end-users of the system. For example, much of the functionality used by managers in interactive mode will be implemented as methods of the Member and Provider subclasses.

3.1.1 Operators:

Operator objects exist solely to provide a base of the hierarchy. They store identification information common to all actors in the hierarchy, and offer methods to access that information. Specifically, the Operator contains a field for an 'ID' struct, which will be introduced in section 3.2 and described in detail in section 5.1.

3.1.2 Users:

User objects represent the common aspects of Providers and Members. Specifically, they will handle the bulk of the data manipulation associated with Service objects, which are introduced in section 3.2 and described in detail in section 5.2.

3.1.3 Managers:

Manager objects are the most limited of the three "bottom level" subclasses of the hierarchy. They have no significant functionality beyond their base, and exist simply to identify a user of the ChocAn Simulator as a manager and allow that manager access to the system in interactive mode.

3.1.4 Providers:

Beyond the Operator and User bases, Provider objects employ functionality to manage their unique data. They contain information pertaining to the services they have provided ChocAn members. See subsection 5.3.2 for a detailed description of the Provider class.

3.1.5 Members:

Beyond the Operator and User bases, Member objects do not contain much unique information or functionality. They exist to store and manipulate the data associated with a ChocAn member. See subsection 5.3.1 for a detailed description of the Member class.

3.2 Other Structures

Two other major objects outside the primary class hierarchy exist within the system: the Service and ID structures.

3.2.1 Service:

A Service object is simply a collection of data describing a service performed by a provider for a ChocAn member. Since managers should be able to modify service data, and since this capability should not be extended to service providers, this object should be implemented as a class, objects of which are used or contained by Member and Provider objects via the User base class. Fields and methods of the service class are described in section 5.2.

3.2.2 ID:

The ID struct is simply a collection of data applicable to objects of the Operator class, i.e. Users (Members and Providers) and Managers. Data comprising the ID struct is defined in section 5.1.

3.3 Using the Hierarchy

In section 4.2 we describe the functionality available via the manager terminal, including methods to add, edit, and remove member, provider and service data from the database. This functionality will be accomplished using standalone wrapper functions that exist outside the hierarchy described above. These wrappers will invoke the corresponding methods of the appropriate object in the hierarchy. For example, to change a ChocAn member's address, a manager would log in to the Simulator and select the "edit member" option in interactive mode. This menu option calls a wrapper function, which in turn invokes the proper Member class method to edit the address field.

All this is to say that a fair amount of functionality (adding and removing to the database in particular) will exist outside the major system actors described in detail in this document.

4 System Architecture

The primary objects in the system are interrelated (or contained) in the hierarchy defined above. They work with each other to inform the structure of the weekly reports (the member, provider, and summary reports--see sections 4.4 and 5.5) to be printed by the system and generated on-demand by the manager. *Fig. 2* depicts a UML diagram of the relationships between the objects of the system and the reports that need to be generated.

A more detailed description of the data encompassed within each object is given in section 5.

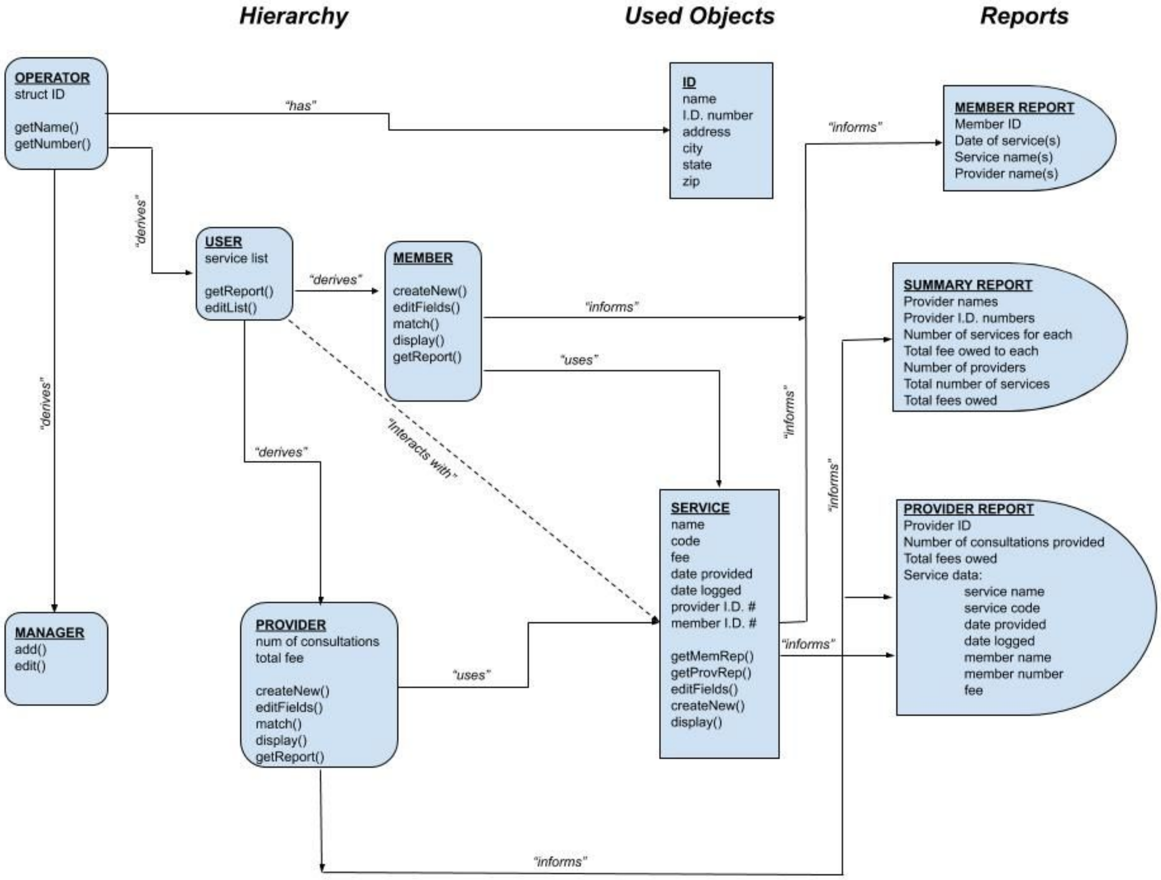


Figure 2: a broad-brush diagram of the relationships between the primary objects in the system and the reports generated by the Data Center.

4.1 Logging In

When the system boots, the user will be prompted to identify themselves via their I.D. number. The developers recommend a more secure login system be implemented as a layer over the one we present in this document, but that is outside the scope of the Data Center.

The user's I.D. number will be used to identify them as a ChocAn manager or service provider. In the fully functional ChocAn Simulator, this distinction will not be as important in the login context, since the two types of users (manager and provider) will be differentiated by the hardware being used (the manager terminal vs the provider terminal). Since hardware is outside the scope of the Data Center, we assign managers I.D. numbers beginning with '1', while service providers and members will have I.D. numbers beginning with '0'. See section 4.5 for details.

4.1.1 Manager Login:

If a manager logs in to the ChocAn Simulator, they will be presented with one of two menus, depending on the time of day (we model this as an option being presented to the user upon identification as a manager).

4.1.1.1 In interactive mode: If the user is a manager logging in during ChocAn's regular operating hours, the system's main process will run in interactive mode, the functionality of which is described in section 4.2.

4.1.1.2 Outside interactive mode: If the user is a manager logging in outside ChocAn's regular operating hours, the system will offer limited functionality (see section 4.2).

4.1.2 Provider Login:

If a user logs in with an I.D. number that the system identifies as being associated with a provider, they will be presented with the options described in section 4.3.

4.2 The Manager Terminal

Upon logging into the system, a manager will be presented with a text-based menu for the operator to work through in order to complete the chosen task.

This section describes the functionality available to a user identified by the system as a manager in interactive mode. Functionality that is also available to managers outside interactive mode is noted as such.

Note that most of the functionality described here will *not* be implemented within the Manager class, despite the fact that it represents work that can only be done by a manager. Instead we implement standalone wrapper functions that invoke methods of the appropriate classes to access their data.

4.2.1 Manager-Interactive Tasks:

4.2.1.1 Add: The user may create a new profile with a valid, unused I.D. number that the system will identify as a manager.

4.2.1.2 Edit: The user may edit the data associated with a manager.

4.2.1.3 Remove: The user may delete the profile of an existing manager.

4.2.2 Provider-Interactive Tasks:

4.2.2.1 Add: The user may create a new profile with a valid, unused I.D. number that the system will identify as a provider.

4.2.2.2 Edit: The user may edit the data associated with a provider.

4.2.2.3 Remove: The user may delete the profile of an existing provider.

4.2.2.4 Retrieve: The user may look up a provider's profile by searching the system for their name.

4.2.2.5 Display: The user may display the identification data associated with a provider.

4.2.2.6 GetReport: The user may generate any provider's weekly provider report at any time. This functionality is also available to managers outside interactive mode.

4.2.3 Member-Interactive Tasks:

4.2.3.1 Add: The user may create a new profile with a valid, unused I.D. number that the system will identify as a member.

4.2.3.2 Edit: The user may edit the data associated with a member.

4.2.3.3 Remove: The user may delete the profile of an existing member.

4.2.3.4 Retrieve: The user may look up a member's profile by searching the system for their name.

4.2.3.5 Display: The user may display the identification data associated with a member.

4.2.3.6 GetReport: The user may generate any member's weekly member report at any time. This functionality is also available to managers outside interactive mode.

4.2.4 Summary Report:

The user may generate the weekly summary report at any time. This functionality is also available to managers outside interactive mode.

4.3 The Provider Terminal

Upon logging into the system, a service provider will be presented with a menu of three choices. This section describes the functionality on that menu.

4.3.1 Requesting a Provider Directory:

At any time, a provider may use their terminal to request a copy of the current provider directory from ChocAn. The directory will be sent to the provider as an email attachment (communications functionality is not implemented--we model it here by displaying a table of service names and associated codes).

4.3.2 Requesting ID Verification:

When a member goes to a provider to obtain services, they will present their member I.D. The provider will scan the I.D. (this functionality is outside the scope of the Data Center--we model it here as text input of the member I.D. number) and the Data Center will determine and display its status as validated, invalid or suspended.

4.3.3 Logging a Service:

4.3.3.1 Enter Service Code: Once the service has been performed, the provider will log it in the Data Center by using the provider directory to obtain the associated service code.

4.3.3.2 Enter Date and Comments: The provider should then enter the date the service was provided and any applicable comments.

4.3.3.3 Record Service: The Data Center will write a record of the service provided and display the appropriate fee for the provider's records.

4.4 Weekly Reports (Overview)

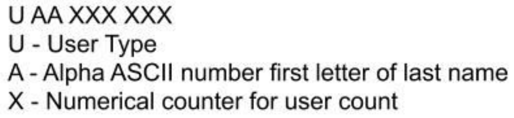
Every Friday at midnight, a series of automated reports will be generated. They present an itemized accounting of services to ChocAn members and providers, as well as a summary of the week's transactions.

The reports comprise data contained within User and Service objects. The Data Center requests a report via a wrapper function, which invokes the method of the appropriate User (Member for a member report, Provider for a provider report), which in turn produces the appropriate Service data.

Weekly compilation of the reports is outside the scope of the Data Center; implemented here is the ability of managers to generate them on demand (a feature that will remain available to end-users of the ChocAn Simulator). The contents of the reports are described in detail in section 5.5.

4.5 Data Structure

ID numbers will be generated by the system and returned to be the only way to reference users. The data structure will be built based on that ID number. It will be formatted UAA XXX XXX.



U AA XXX XXX
U - User Type
A - Alpha ASCII number first letter of last name
X - Numerical counter for user count

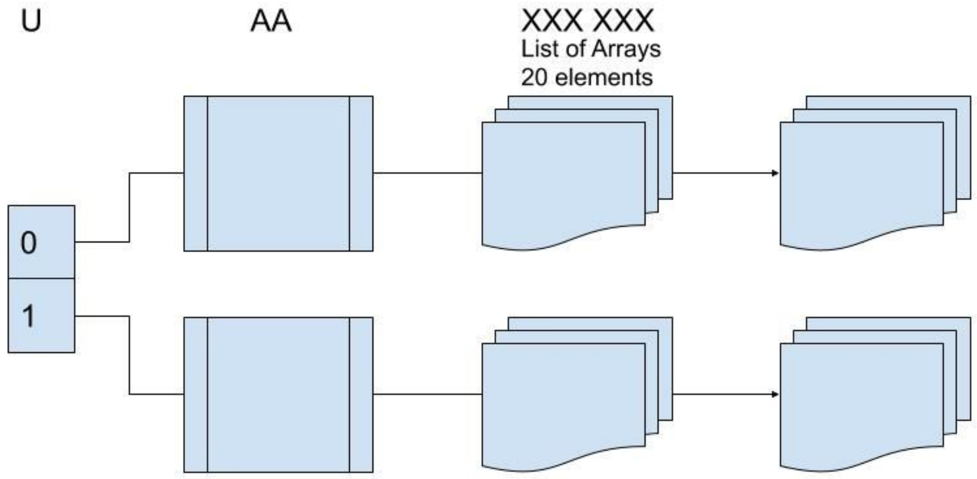


Figure 3: a basic diagram of the structure of the user database.

5 Detailed System Design

This section describes in detail the components of the subsystems described in sections 3 and 4. The reader is advised to note our usage of caps when referring to these system actors. We refer to the ChocAn employees who have access to the interactive mode of the Simulator as “managers” while the data abstractions within the system are “Managers”. A ChocAn service provider is referred to as a “provider” while the corresponding data abstraction is a “Provider”. A person operating the ChocAn Simulator is a “user”; the base class of Member and Provider is “User”. Tread lightly.

5.1 *The ID Structure*

The ID struct exists to group identification data that can be associated with a ChocAn member or a service provider. By inheritance, all objects of the Operator class (that is, all Managers, Users, Members and Providers) possess an ID.

Note that Manager objects contain an ID even though manager identification data is not used by the Data Center, except to log in at the manager terminal (see section 4.1). More specifically, the first digit of the nine-digit I.D. number contained within an ID struct is the key element differentiating the three low-level actors in the system: Managers, Members and Providers.

An ID struct comprises the following fields:

● Name	*up to 25 characters*
● I.D. number	*9 digits*
● Street address	*up to 25 characters*
● City	*up to 14 characters*
● State	*2 characters*
● Zip code	*5 digits*

5.2 *The Service Class*

The Service class is a data abstraction representing a service provided to a member. Service objects contain fields representing data on the service provided as well as on the providers and members giving and receiving the service.

Service objects are accessed in the hierarchy primarily through methods contained within the User class (see section 5.3). The methods listed below will be invoked primarily by those User methods.

5.2.1 Service Class Fields:

The Service class includes the following fields:

● Name	*up to 20 characters (not 25!)*
● Service code	*6 digits*
● Date provided	*MM-DD-YYYY*
● Date logged	*MM-DD-YYYY*
● Member I.D. number	*9 digits*
● Provider I.D. number	*9 digits*
● Applicable fee	*up to $999.99*
● Comments	*optional, up to 100 characters*

5.2.2 Service Class Methods:

The following methods are available to users of the Simulator identified as managers (not providers, except where noted), who will invoke them via wrapper functions outside the Service class.

Service class methods allow the caller to:

- Submit data for a member report

- Submit data for a provider report
- Edit the Service's data fields
- Create a service (this method should only be invoked from the provider terminal, when a provider logs a service (see subsection 4.3.3))

5.3 The User Class

The User class is derived from the Operator class, and acts as the base of the Member and Provider classes. The User class contains the bulk of the operations that make use of the data contained by objects of the Service class. Notably, virtual User class methods will exist to generate both member reports and provider reports (see *Fig. 2*) via the derived Member and Provider classes.

5.3.1 The Member Class:

The Member class is derived from the User class, which is derived from the Operator class. It inherits an ID struct, and contains a list of Service objects representing the services performed for the member that week.

Note that the service list is updated when the member receives a service. It will be reset weekly when the member report is generated. However, since weekly generation of the reports is outside the scope of the Data Center, we present the user with an opportunity to reset the service list when a member report is manually generated.

5.3.1.1 Member class fields: Fields of the Member class include:

- An ID struct (see section 5.1)
- A list of services the member has received that week

5.3.1.2 Member class methods: The methods implemented here are available to users of the Simulator identified as managers (not members), who will invoke them via wrapper functions outside the Member class. Member class methods allow the caller to:

- Change the Member's ID
- Add Services to the Member's service list
- Remove Services from the Member's service list
- Match the Member's I.D. number when searching
- Display the Member's ID and service list
- Generate a member report (extended from virtual User class function)

5.3.2 The Provider Class:

The Provider class is derived from the User class, which is derived from the Operator class. It inherits an ID struct, and contains a list of Service objects representing the services provided to ChocAn members for that week.

Note that the service list is updated when a service is provided. The list will be reset weekly when the provider report is generated. However, since weekly generation of the reports is outside the scope of the Data Center, we present the user with an opportunity to reset the service list when a provider report is manually generated.

5.3.2.1 Provider class fields:

Fields of the Provider class include:

An ID struct (see section 5.1)	
A list of services provided that week	
The number of services provided that week	*3 digits*
The total fee owed to the provider for that week	*up to $99,999.99*

5.3.2.2 Provider class methods:

The methods implemented here are available to users of the Simulator identified as managers (not providers), who will invoke them via wrapper functions outside the Provider class. **This functionality is not accessible via the provider terminal!**

Provider class methods allow the caller to:

- Change the Provider's ID
- Add Services to the Provider's service list
- Remove Services from the Provider's service list
- Manage the Provider's other data fields (number of services, total fee)
- Match the Provider's I.D. number when searching
- Display the Provider's ID and service list
- Generate a provider report (extended from virtual User class function)

5.4 The Manager Class

As it is derived from the Operator class, the Manager class possesses an ID struct (a name, I.D. number, address etc… see section 5.1). The only important piece of this data, as far as the Data Center is concerned, is the I.D. number. I.D. numbers stored in Manager objects begin with a '1', which identifies them as being associated with manager users, who should be able to access interactive mode and other functionality.

In interactive mode, users of the system identified as managers should be able to add, edit and remove Manager objects from the database, as they can do with Provider, Member and Service objects. It should be noted that this functionality is not specified in the project requirements, but the developers recognize that the list of ChocAn managers must necessarily change over time (since humans tend to move, find new jobs, and get fired from time to time), and it would be best for the capabilities of the Data Center to reflect this.

5.5 Weekly Reports (Detail)

The reports described in this section will be requested automatically every Friday night at midnight by the ChocAn Simulator and compiled by the Data Center. Since the Simulator is outside the scope of the system described here, we model only the reports' ability to be generated on demand by a user identified as a manager. Note that a manager may request these reports at any time, even outside interactive mode (which is only available during ChocAn business hours).

Note also that these reports are only available to managers--they are directly inaccessible to ChocAn members and ChocAn service providers, even though the Member and Provider classes described in sections 3.1, 4.2 and 5.3 contain methods to fetch the data in the reports.

Data comprising these reports will be compiled via the virtual User class method getReport(), which will be overridden by getReport() methods in the Member and Provider classes, both of which are derived from User.

5.5.1 Member Reports

Member reports are issued to each member who received at least one service from a ChocAn provider that week. They include general information specific to the member to whom the report is issued, as well as information applicable to each service provided.

Note that the member reports do not include administrative data used by managers and providers. For example, the service I.D. number, the date the service was logged, and any financial information should be hidden from the member reports.

5.5.1.1 Identification Data: Data that will head the member report include an ID struct object containing the member's

- Name
- I.D. number
- Street address
- City
- State
- Zip code

5.5.2.2 Service Data: Data that will be printed for each service received that week include:

- The date the service was received
- The name of the service provided
- The name of the associated service provider

5.5.2 Provider Reports

Provider reports are issued to each provider who performed at least one service for a ChocAn member that week. They include general information specific to the provider to whom the report is issued, as well as information applicable to each service provided.

5.5.2.1 General Data: Data that will head the provider report include:

- An ID struct object containing the provider's
  - Name
  - I.D. number
  - Street address
  - City
  - State
  - Zip code
- The number of services provided over the week
- The total fees owed to the provider by ChocAn for the week

5.5.2.2 Service Data: Data that will be printed for each service provided that week include:

- The date the service was provided
- The date the service was logged
- The name of the service provided
- The associated service I.D. number
- The name of the member who received the service
- The I.D. number of the member who received the service
- The fee associated with the service provided

5.5.3 Summary Reports

Summary reports include information applicable to each provider who performed at least one service to a ChocAn member that week. They also include information on the total number and fees of services rendered by all providers.

5.5.3.1 Separate Data: Summary report data for each provider include:

- The provider's name
- The provider's I.D. number
- The number of services logged that week
- The total fee owed by ChocAn for services rendered

5.5.3.2 Aggregate Data: Additional summary report data include:

- The number of providers listed in the summary report
- The number of services accounted for in the summary report
- The sum of fees owed to providers for the week