Unit 1

Suppose that the product for Chocoholics Anonymous of Appendix A has been implemented exactly as described. Now the product must be modified to include endocrinologists as providers. In what ways, will the existing product have to be changed? Would it be better to discard everything and start again from scratch?

The existing software product has to be changed by including an endocrinologist as a provider. Since the endocrinologist is also a Health Care Professional/Provider, the endocrinologist must also have the provider data fields such as the name, number, street address, city, state, zip code, date of service, date and time data were received by the computer, member name, member number, service code, fee to be paid. So also, the endocrinologist would provide new kinds of services which would be different from the services that are provided by other Health Care Professionals hence, new types of services associated with a particular service code would have to be created.

It would not be necessary to discard the everything and start again because adding a new type of provider (an endocrinologist) is not a complex issue. We can take advantage of Object-Oriented Programming by incorporating abstraction, inheritance, and polymorphism to our program. A provider can be a dietitian, internist, exercise experts, or an endocrinologist (polymorphism) by creating a “provider” superclass and using abstraction to make sure that its methods are implemented in its subclasses (dietitian, internist, exercise experts, or an endocrinologist). Then, the subclasses inherit the methods and fields of its superclass. Regression testing should be performed to ensure that previously developed software still works with the new changes.

Unit 2

Which software life-cycle model would you use for the Chocoholics Anonymous product described in Appendix A? Give reasons for your answer.

The most suitable software life-cycle model to use for the Chocoholics Anonymous product is the iterative-and-incremental life-cycle model which involves the implementation of a small set of software requirements and iteratively improving developing versions until the complete product is ready to be deployed. This allows the product to be used right away even before an endocrinologist provider and its services are added to the product. “At the end of each iteration, there is a working version of part of the overall target software” (Schach, 2011, p. 50).

Iterative-and-incremental life-cycle model is most suitable because our organization is only responsible for the Chocoholics Anonymous data processing Software while a different organization is responsible for designing the communications aspect and implementing the EFT component. This life-cycle model allows us to check the robustness of the architecture early on and to make the necessary changes that will ensure that the different components of the product are compatible and can adapt to changes.

So also, as cited in Schach (2011), a report carried out by Rubenstein (2007) showed that there is empirical evidence that the Iterative-and-incremental life-cycle model works. The results showed that one of the factors responsible for successful projects in 2002, 2004, and 2006 was the iterative process Schach (2011).

Unit 3

What differences would you expect to find if the Chocoholics Anonymous product of Appendix A were developed by an organization at CMM level 1, as opposed to an organization at level 5?

CMM Level 1 is the initial level. At this level, there is no sound software management practices hence, the process is unpredictable and could lead to software faults, especially when staff changes. The data processing section that is being developed for this product is liable to be faults. For example, the data types for incoming data may be arbitrarily used without informing the company in charge of communications. Eventually, it may be realized that the data types supplied may be different than expected when it is time to integrate the different parts of the product. The programmers would have to respond to such crisis's situation since their tasks were not preplanned, and this will cost money and time.

CMM Level 5 is the optimization level. This level is characterized by continuous improvement of the software development process. At this level, the organization is likely to employ the use of object-oriented programming techniques in creating the product which allows for encapsulation and code re-use. This will certainly save more time and money compared to CMM level 1. However, severe changes in the program are not expected at this level since the company would have pre-planned for such situations. So also, changes in staff is unlikely to lead to problems in developing the product because there would have been proper documentation the existing code, so a new programmer will be able to understand the code that was written by an ex-employee. The knowledge gained from each project is applied to future projects, hence this process incorporates a positive feedback loop which results in a steady improvement in quality and productivity Schach (2011).

Unit 4

What type of team organization would be appropriate for developing the Chocoholics Anonymous product described in Appendix A?

The appropriate team approach for the chocoholic Anonymous product would be the Democratic ream approach because it encourages the programmers to work together as a team and find bugs and faults in the program with little or no ego being displayed.

So also, since Chocoholic Anonymous is only responsible for the data processing software while other organizations will be responsible for other parts of the software, a democratic team will help the different parties involved in the project to seamlessly integrate the different parts of the product together. “Every programmer must encourage the other members of the team to find faults in his or her code. The presence of a fault must not be considered some- thing bad but a normal and accepted event; the attitude of the reviewer should be appreciation at being asked for advice, rather than ridicule of the programmer for making coding mistakes” (Schach, 2011, p. 110).

Unit 5

What types of CASE tools would be appropriate for developing the Chocoholics Anonymous product described in Appendix A?

Computer Aided Software Engineering (CASE) tools are tools that will assist the programmers in developing the software. The CASE tools that would be appropriate for developing the Chocoholics Anonymous product include:

- Front-end tools: will assist the developers during the earlier workflows of the process to determine the requirements of the software, analyze it, and design the workflows.
- Back-end tools: will assist the developers with the implementation workflow and postdelivery maintenance.
- Data dictionary: will assist the developers in storing information such as variable names and types, and where they are defined, as well as procedure names and parameters and their types. It can store data such as a provider's number, member's number, service codes, as well as their types, for Chocoholics Anonymous.
- Consistency checker: can be combined with the data dictionary to check that every data item in the specification document is reflected in the design and vice versa.
- Report generator: is used to generate the code needed for producing a report. It will Assist Chocoholic Anonymous in generating reports on the service providers, members, service provided, accounts manager etc.
- Requirements management workbench: allows the system analysts to organize and track the requirements of a software development project e.g. RequistePro.
- Structural editor: helps to speed up the implementation of the product by assisting the developers in detecting syntax faults as soon as it is typed by the programmer. With

structural Editors such as Eclipse (for Java, C/C++, Ruby, etc.), time is not wasted on futile compilations as syntax errors are made visible by the editor.

- Email: is can be used as a means of communication between the team members. For confidential and security reason, it is most appropriate to use the organization's email for such communication.
- Online documentation: provides the developers with information about the operating system, editor, programming language, and so on. It is desirable that such information is available online because it is more convenient to search online, the query is quicker, and it is easier to update an online manual than to keep track of all hard-copy versions.

Unit 6

Explain how you would test the utility, reliability, robustness, performance, and correctness of the Chocoholics Anonymous product in Appendix A.

- Utility: This is the extent to which the client's needs are met when using a correct product under conditions that are allowed by its specifications. To test for utility, we can test for:
    - How easy the product is to use: we can test for utility by having the clients perform some operations and work through the user interface of the product while taking note of how easily the client can enter and access information.
    - Whether the product performs useful functions: we can test for utility by checking that important functions of the Chocoholic Anonymous product such as swiping a member's card actually works, and ensuring that reports generated are correct and well-formatted.
    - Whether the product is cost effective compared to competing products: This would be difficult to test since this a custom software that was specifically developed for Chocoholic Anonymous. However, ready-made products available for sale to the general public with similar functionality to our product can examined for their price and specifications and compared with our projected budget and previous utility tests to determine the viability of the product.
- Reliability: This is a measure of the frequency and criticality of failure of the product. To test for reliability, we can run the product with different sets of data which meets the requirements in the specification document. For example, we can swipe a number of valid and dummy membership cards on the same machine and keep record of the type of

failure, the how often the failure occurs, and how long it takes to recover from such failure.

- Robustness: This is a function of a number of factors such as, the range of the operating system, the possibility of unacceptable result with valid input, and the acceptability of effects when the product is given invalid inputs. We can test for robustness by using well designed test cases to check for all possible faults that the product could have. If an error should occur, the software should print and error message as to why it occur instead of crashing.
- Performance: This is the extent to which the product meets its constraint of response time and space requirements. We can test the response time by using well designed test cases that would test the limits of the system. The response time should be calculated, recorded, and compared with the product specifications. We can test the space requirement by compiling the product and calculating the size of all the needed files for the system to determine if it meets the size requirements.
- Correctness: This is a test to check if the product satisfies its output specifications, independent of it use of computing resources, when operating under permitted conditions. We can use correctness proofs to ensure that the product is correct and that it satisfies its specification. Cost-benefit analysis can be used to determine the economic viability of the correctness proving on the project.

Unit 7

Suppose that the Chocoholics Anonymous product of Appendix A was developed using the classical paradigm. Give examples of modules of functional cohesion that you would expect to find. Now suppose that the product was developed using the object-oriented paradigm. Give examples of classes that you would expect to find.

Examples of modules of functional cohesion that can be found when using the classical paradigm include:

- **verify_member_number**: checks if a member's card has an invalid number or if the member was suspended.
- **add_member**: used for registering a new member
- **update_member_record**: updates a member's record after a health care service has been provided to a member.
- **delete_member:** used for deregistering a member
- **add_provider**: used for registering a new provider
- **update_provider_record**: updates a provider's record after a health care service has been received from a provider.
- **delete_provider:** used for deregistering a provider.
- **look_up_service_code**: used in a Provider Directory by a provider to look up the appropriate six-digit service corresponding to the service provide.
- **store_information**: used to write new entered information to disk.
- **look_up_fee**: used for looking up the fee to be paid for a service provided.
- **send_provider_list**: sends an alphabetically ordered list of service names and corresponding service codes and fees to the provider as an email attachment.

- **send_member_report**: sends a list of services provide to a member
- **send_summary_report**: sends a report to the manager for accounts payable.

Examples of classes that can be found when using the object-oriented paradigm include:

- **Member** class: stores member information and performs all member related tasks. It would contain methods/functions such as **add_member**, **verify_member_number**, **send_member_report**, and **delete_member**.
- **Provider** Class: stores provider information and performs all provider related tasks. It would contain methods/functions such as **add_provider**, **update_provider_record**, **send_provider_list**, **delete_provider**.
- **Service** Class: stores information about the kinds of services available at ChocAn and performs service-related tasks. IT would contain methods/functions such as **look_up_service_code** and **look_up_fee.**
- **Report** Class: used for generating and sending financial report. It would contain methods/functions such as **send_summary_report**.

Unit 8

Suppose that the Chocoholics Anonymous product of Appendix A is developed using the classical paradigm. What parts of the product could be reused in future products? Now suppose that the product is developed using the object-oriented paradigm. What parts of the product could be reused in future products?

If the product was developed using classical paradigm, we could reuse the following parts of the product for future products:

- The data input routines: For example, we could use the **add_member** and **update_member_record** to register a new gym member and update his or her information respectively.
- Validation routine: The **verify_member_number** module can be used to check a gym member's card for a valid gym membership pass after swiping the card on a card reader/terminal.
- The input lookup routines: The **look_up_service_code** module can be used to check for the different gym membership plans available and their prices.
- The report routine: The **send_member_report** can be used to compile and send a summary of a member's gym activities.

Of course, the routines may have to be edited to better contain more specific information related to the gym.

On the other hand, if the product was developed using the object-oriented paradigm, we could reuse the **Member** class, **Provider** class, **Service** class, and the **Report** class. Some of the methods or functions can be reused in other products. For example, the

verify_member_number can be used in a shopping mall to validate gift cards and check its balance. Other things such as the specification document, test cases, design template etc. can also be reused.

Unit 9

Consider the Chocoholics Anonymous project described in Appendix A. Why is it not possible to estimate the cost and duration purely based on the information in Appendix A?

It is not possible to estimate the cost and duration purely based on the information in Appendix A because the information given focuses more on the basic layout for the functionality of the product and omits important details that are necessary for cost and duration estimation.

In Appendix A, there is no mention about the possibility of human factors such the resignation of a critical staff member during product development. If a critical staff member resigned during the project, time and money will be spent in attempts to fill the vacated position and integrate the replacement into the development team, or in reorganizing the remaining team members to compensate for the loss.

Other factors that could affect the estimates that are not mentioned in Appendix A include; the skill levels of the programmers, the complexity of the project, the size of the project, familiarity of the development team with the application area, the hardware on which the product is to be run, and the availability of CASE tools (Schach, 2011).

“Another factor is the deadline effect. If a project has to be completed by a certain time, the effort in person-months is greater than if no constraint is placed on completion time; hence, the greater the cost” (Schach, 2011, 276). The shorter the deadline given, the greater the effort and the cost, but no deadline is mentioned in Appendix A.

So also, when considering techniques such as the COCOMO or COCOMO II, they require a lot of detail for estimates that are not provide in Appendix A. They require knowledge of the company’s metrics for previous work in software development. FFP is also not feasible to use

because it is based on the calculation of function points in the program, which are in turn based on the file, flows, and processes of the product. We are also lacking information about the efficiency of the development process within the organization. Hence, it is impossible to estimate the the cost and duration based on the information given in Appendix A.

Unit 11

Perform the requirements workflow for the Chocoholics Anonymous project in Appendix A.

Table 1 – Initial glossary of the Chocoholic Anonymous product.

TERMS	DESCRIPTIONS
Accounts Payable	Accounts of providers sent to the manager for crediting for services rendered to ChocAn members
Accounting Procedure	A procedure that is run at midnight on Fridays at the ChocAn Data center. It reads the weeks' file of services provided and prints a number of reports.
Acme Accounting Services	A third party organization responsible financial procedures such as recording payments of membership fees, suspending members whose fees are overdue, and reinstating suspended members who have now paid what is owing.
Addiction	Excessive physical or psychological dependence on chocolate.
ChocAn Data Center	The location of the ChocAn records storage where members' numbers are verified and where the main accounting procedure is run.
ChocAn Operator	An individual working for ChocAn, allowed to perform basic functions such as adding new members, deleting resigned members, and updating member records.
Chocoholics Anonymous (ChocAn)	An organization dedicated to assisting its members to get rid of chocolate addition.
Consultation	A meeting of a member with a provider.
Dietitian	An expert in nutritional health care.
EFT Component	An acronym for electronic funds transfer.
Exercise Expert	A fitness trainer who creates plans designed to improve the health of ChocAn members.
Interactive mode	A mode that allows the software to respond to inputs from an operator in order to add new member, delete members, and update member records.
Internist	A specialist in non-surgical diagnosis and treatment of diseases.
Manager	A staff responsible for administering all or parts of the organization. The manager requires information about the services provided to members during a particular week, as well as the fees to be paid to the providers.
Member	An individual who has registered for ChocAn services.
Member Number	A nine-digit number that uniquely identifies a member.

Membership Card	A plastic card embossed with the member's name and number, and incorporating a magnetic strip on which the information is encoded.
Membership Fee	The amount of money a member must pay in order to be entitled to ChocAn services.
Provider	A health care provider (Dietitian, Internist, or Exercise expert) that provides services to members.
Provider Directory	An alphabetically ordered list of service names and corresponding service codes and fees.
Provider Number	A nine-digit number that uniquely identifies a Provider.
Service	A treatment or consultation provided by a provider.
Service Code	A six-digit number corresponding to a service provided.
Terminal	A device used to read membership cards, similar to a credit card device.
Treatment	Health services administered to a member by a provider.
Member Suspended	The status of a member's account when membership fees are outstanding for at least one month.
Validated	A key word that appears on the card reader after a provider passes a member's card or inputs a member's number into the card reader in order to bill ChocAn, after providing health care service to a member.

Figure 1 – Use-case diagram for the requirements of the Chocoholics Anonymous project

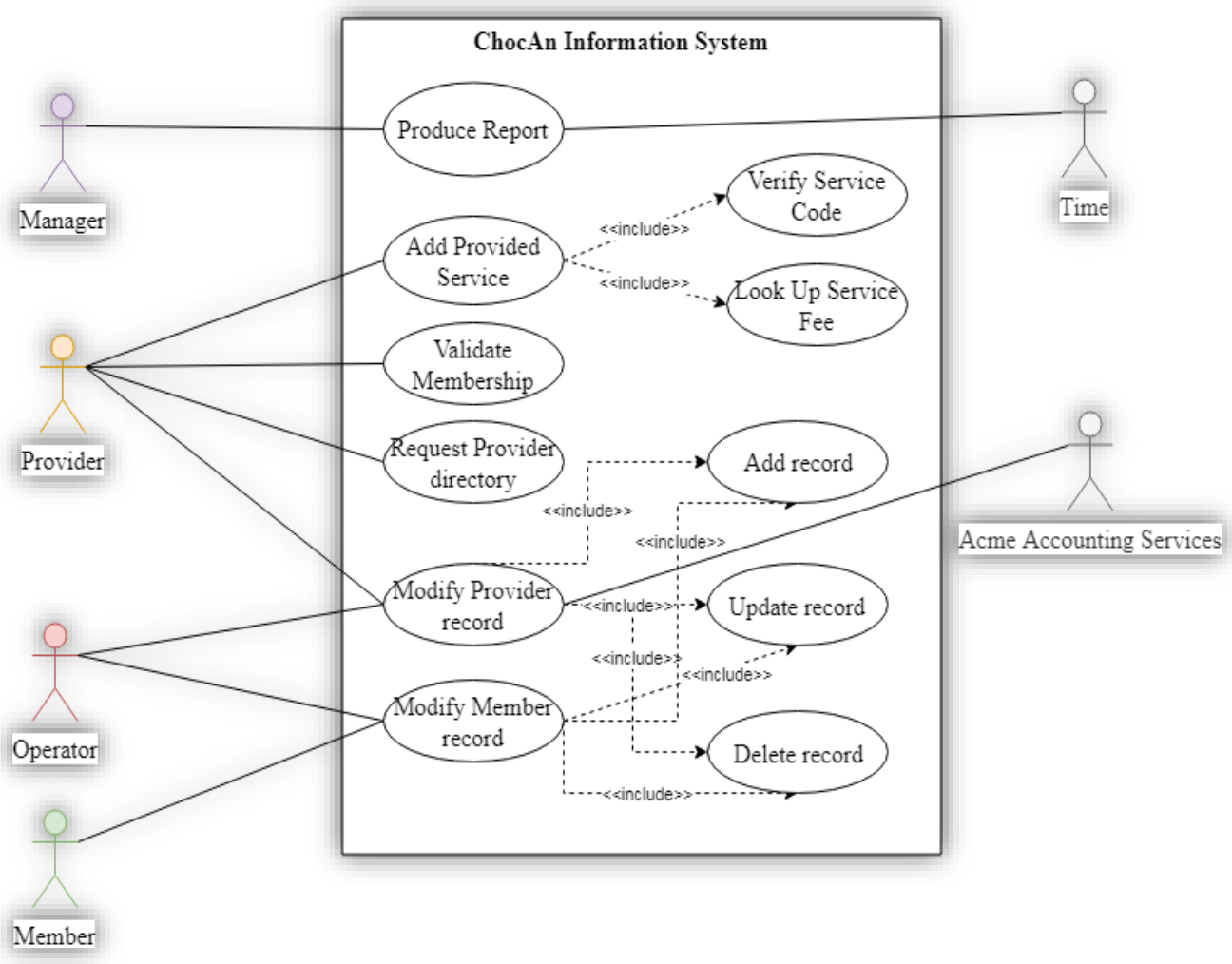

Table 2 – Description of "Produce Report" use-case

Brief Description: This sends a summary of the report for accounts payable to the manager, and sends an email attachment to the members and providers.
Step-by-Step Description: 1. The system reads the summary of the report when: (i) A manager requests a report anytime during the week. (ii) It is midnight on Fridays, where the file of services provided are read by the main accounting procedure at the ChocAn Data Center. 2. The system creates the following reports:

(i) Each member who has consulted a ChocAn provider during that week receives a list of services provided to that member, sorted in order of service date. The report, which is also sent as an e-mail attachment, which includes:
- 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).

(ii) Each provider who has billed ChocAn during that week receives a report, sent as an e-mail attachment, containing the list of services he or she provided to ChocAn members, which includes:
- 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 week (up to $99,999.99).

(iii) A summary report with the following information:
- List of every provider to be paid in the current week
- The number of consultations each had
- His/her total fee for the current week
- The total number of providers that provided services in the current week
- The total number of consultations by all providers for the current week
- The overall fee total for the current week.

3. The system writes the EFT data to disk on Fridays at midnight and the report is printed by the system.

Table 3 – Description of "Add Provided Service" use-case

Brief Description: This sends a summary of the report for accounts payable to the manager, and sends an email attachment to the members and providers.

Step-by-Step Description:

1. The provider uses the terminal to key in the date the service was provided in the format: MM-DD-YYYY.
2. The provider uses the "Provider Directory" to look up the appropriate six-digit service code corresponding to the service provided.
3. The "Verify Service Code" use-case is invoked
4. The providers verifies the code visually or returns to step 2 to input a new code
5. The provider can enter additional comments or leave the field blank
6. The system writes a record that includes the following information:
    - 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).
7. The "Look up Service Fee" use case is invoked.
8. For verification purposes, the provider enters into a form the following information:
    - Current date and time
    - The date the service was provided
    - Member name and number
    - Service code
    - Fee to be paid.

Table 4 – Description of "Validate Membership" use-case

Brief Description: This allows a provider verify the status of a member through the terminal.

Step-by-Step Description:

1. The provider swipes the member's card through the terminal
2. The terminal uses the member's number stored in the member's card to communicate with the ChocAn data center.
3. The system returns the status corresponding the member's number.
4. The terminal displays either of the following statuses on the terminal screen:
    (i) "Validated" if the number is valid
    (ii) "Invalid number" or "Member suspended" if the number is not valid or fees are owed respectively.

Table 5 – Description of "Request Provider Directory" use-case

Brief Description: This sends a provider directory to a provider upon request.
Step-by-Step Description: 1. The provider requests for the provider directory from the system 2. The software prepares an alphabetically ordered list of service names and their corresponding service codes and fees. 3. The list is sent to the provider as an e-mail attachment.

Table 6 – Description of "Modify Provider Record" use-case

Brief Description: This edits a provider's information stored in the ChocAn database.
Step-by-Step Description: 1. The operator is edits the provider's information in either of the following ways: (i) A new provider joins ChocAn and the "Add record" use-case is invoked. (ii) A current provider needs to make changes to his/her information and the "Update record" use-case is invoked. (iii) A provider resigns from ChocAn and the "Delete record" use-case is invoked.

Table 7 – Description of "Modify Member Record" use-case

Brief Description: This edits a member's information stored in the ChocAn database.
Step-by-Step Description: 1. Member records are modified in the following ways (i) The operator is edits the member's information in either of the following ways: - A new member joins ChocAn and the "Add record" use-case is invoked. - A current member needs to make changes to his/her information and the "Update record" use-case is invoked. - A member discontinues ChocAn services and the "Delete record" use-case is invoked. (ii) The Acme computer updates the relevant ChocAn Data Center computer membership records each evening at 9 P.M.

Table 8 – Description of "Verify Service Code" use-case

Brief Description: This is used to validate an inputted service code.
Step-by-Step Description: 1. The provider uses the Provider Directory to look up the appropriate six-digit service code corresponding to the service provided. 2. The provider keys in the service code. 3. The software displays a message to the terminal corresponding to either of the following: (i) The name of the service corresponding to the code (up to 20 characters) and asks the provider to verify that this is indeed the service that was provided. (ii) An error message is displayed if the entered code is non-existent.

Table 9 – Description of "Look up Service Fee" use-case

Brief Description: This is used to display a service fee to the terminal.
Step-by-Step Description: 1. The service code is used by a provider to look up the service fee. 2. The fee is sent to the terminal 3. The fee is displayed on the terminal screen.

Table 10 – Description of "Add record" use-case

Brief Description: This is used to add a new member or provider record to ChocAn database.
Step-by-Step Description: 1. The system is run in interactive mode to allow an operator add new members or provider. (i) For each member, the following information is recorded: - 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 provider, the following information is recorded: - 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:

Table 11 – Description of "Update record" use-case

Brief Description: This is used to edit an existing record in ChocAn database.
Step-by-Step Description: 1. The system is run in interactive mode to allow an operator update a members or provider's information. For each record: (i) If the record is that of a member, the member's information is updated with the new inputted information. (ii) If the record is that of a provider, the provider's record is updated with the new inputted information.

Table 12 – Description of "Delete record" use-case

Brief Description: This is used to delete an existing record in ChocAn database.
Step-by-Step Description: 1. The system is run in interactive mode to allow an operator delete a members or provider's record. For each record: (i) If the record is that of a member, the system deletes the member's record from the database. (ii) If the record is that of a provider, the system deletes the provider's record from the database.

References

Schach S. R (2011). *Object-Oriented and Classical Software Engineering, 8th ed*. New York: McGraw-Hill, 2011.