**COMP 370 – Software Engineering**  
**Term Project**  
Problem 13.22  
Question: (Term Project) Perform the analysis workflow of the Chocoholics Anonymous product described in Appendix A.

Assumptions:

* If a provider resigns, the information about him cannot be deleted from the system before all his claims have been processed.
* Similarly, the information about a member or a service cannot be deleted until all claims concerning the member or service have been processed. Thus such deletions should be delayed until the end of the week. This requirement has been ignored in the analysis workflow since the timing of actions is not a concern of analysis.
* A “nice to have” requirement is that members should be informed of new providers, resignation of providers and changed provider details.
* Similarly, providers should be informed of new services, discontinuation of services and changed service details. The requirements workflow includes sending such information by email to the providers and members. This has been left out of the analysis workflow since it is not essential to the product.

**Functional Modeling**

Scenarios for the use cases of Chocoholics Anonymous depicted in Problem 11.24 Part 1 to 4 appear below the in following use cases descriptions.

**Figure 1.0: An extended scenario for the Manage Session use case**

|  |
| --- |
| A provider wishes to initiate a session to use the ChocAn system to verify members, submit claims and receive a provider directory. The provider switches on his or her terminal.   1. The system prompts the provider for his or her provider number. 2. The provider supplies his or her provider number. 3. The system searches for a provider with this number. 4. The system displays a message indicating the options available to the provider. 5. The provider can now use the system. 6. When the provider has finished used the system, he or she chooses the Quit option. 7. The system ends the session. |
| **Possible Alternative**   1. In step 3, the system cannot find the provider number. The system displays the message invalid number, and step 1 is repeated. |

**Figure 1.1: An extended scenario for the Verify Member use case**

|  |
| --- |
| A provider wishes to verify that the number on a member’s card is a valid ChocAn membership number of an active member.  The provider has switched on his or her terminal and entered his or her provider number correctly.   1. The provider swipes the member’s card through the card reader of the provider terminal. 2. The system searches for a member with this number. 3. The system displays a message indicating the status of the member on the provider terminal’s display, namely, Validated. |
| **Possible Alternative**   1. In step 2, the system cannot find the member number. The system displays the message invalid number, and the use case terminates. 2. In step 3, the member’s status is “suspended”. The system displays the message Suspended, and the use case terminates. |

**Figure 1.2: An extended scenario for the Submit Claim use case**

|  |
| --- |
| A provider wishes to submit a claim for a service provided to a ChocAn member. He or she has already verified the member as in the normal scenario of the Verify Member use case.   1. The provider enters the date the service was provided in the format MM-DD-YYYY, the service code and optionally comments about the service rendered. 2. The system verifies that the service code exists. The system displays the corresponding service name. 3. The provider confirms that this is the correct service rendered. 4. The system stores the following information about the claim submitted:   Current date and time (MM-DD-YYYY HH:MM:SS)  Service date (MM-DD-YYYY)  Provider number (max 9 digits)  Member number (max 9 digits)  Service code (max 6 digits)   1. The system displays the fee to be paid to the provider for the service rendered. 2. The provider has a form on which he or she can enter the same information as that stored by the system, as well as the service fee. |
| **Possible Alternative**   1. In step 2, the service code does not exist. The system displays the message Invalid Service code. The provider re-enters the service code and step 2 is repeated. 2. In step 3, the service name is not the name of the service for which the provider wishes to submit a claim. The provider re-enters the service code and step 2 is repeated. |

**Figure 1.3: Scenario for the Receive Provider Directory use case**

|  |
| --- |
| A provider has already switched on his or her terminal and entered his or her provider number correctly.   1. The provider submits a request for a Provider Directory through his or her terminal. 2. The system generates a list (report) of all services, ordered alphabetically according to the service name, including each service:   Service name (max 20 characters)  Service code (max 6 digits)  Service fee (max $999.99)   1. The system sends the list as an email attachment to the provider, and displays a message to this effect on the provider’s terminal. |

**Figure 1.4: First extended scenario of the Maintain Member use case (add a new member)**

|  |
| --- |
| A ChocAn operator wishes to add a new member.   1. The ChocAn operator selects the option to add a new member, and enters the new member’s details:   Member name (max 25 characters)  Member street address (max 25 characters)  Member city (max 14 characters)  Member province (2 letters)  Member zip code (7 digits)  Member email address (max 50 characters)   1. The system allocates a number to the new member, records the new member’s details, and sets the new member’s status to Active. 2. The system displays the new member’s member number and other details. |

**Figure 1.5: Second extended scenario of the Maintain Member use case (update an existing member)**

|  |
| --- |
| A ChocAn operator wishes to update an existing member’s details.   1. The ChocAn operator enters the existing member’s member number. 2. The system searches for the details for the member and displays the details. 3. The ChocAn operator edits the details that must be changed. (The member number can never be changed.) 4. The system updates the member’s details. |
| **Possible Alternative**   1. In step 2, the system cannot find the member number. It displays the message invalid number, and the use case terminates. |

**Figure 1.6: Third extended scenario of the Maintain Member use case (delete an existing member)**

|  |
| --- |
| A ChocAn operator wishes to delete an existing member.   1. The ChocAn operator enters the existing member’s member number. 2. The system searches for the details for the member and displays the details. 3. The ChocAn operator selects the option to delete the member. 4. The system deletes the member’s details. |
| **Possible Alternative**   1. In step 2, the system cannot find the member number. It displays the message invalid number, and the use case terminates. |

**Figure 1.7: First extended scenario for the Maintain Provider use case (add a new provider)**

|  |
| --- |
| A ChocAn operator wishes to add a new provider.   1. The ChocAn operator selects the option to add a new provider, and enters the new provider’s details:   Provider name (max 25 characters)  Provider street address (max 25 characters)  Provider city (max 14 characters)  Provider province (2 letters)  Provider zip code (7 digits)  Provider email address (max 50 characters)  Provider type (Dietitian, Internist, or Exercise Specialist)   1. The system allocates a number to the new provider and records the new provider’s details. 2. The system displays the new provider’s number and other details. |

**Figure 1.8: Second scenario for the Maintain Provider use case (update an existing provider)**

|  |
| --- |
| A ChocAn operator wishes to update an existing provider’s details.   1. The ChocAn operator enters the existing provider’s provider number. 2. The system searches for the details for the provider and displays the details. 3. The ChocAn operator edits the details that must be changed. (The provider number can never be changed.) 4. The system updates the provider’s details. |
| **Possible Alternative**   1. In step 2, the system cannot find the provider number. It displays an error message, and the use case terminates. |

**Figure 1.9: Third extended scenario of the Maintain Provider use case (delete an existing provider)**

|  |
| --- |
| A ChocAn operator wishes to delete an existing provider.   1. The ChocAn operator enters the existing provider’s provider number. 2. The system searches for the details for the provider and displays the details. 3. The ChocAn operator selects the option to delete the provider. 4. The system deletes the provider’s details. |
| **Possible Alternative**   1. In step 2, the system cannot find the member number. It displays an error message, and the use case terminated. |

**Figure 1.10: First extended scenario for the Maintain Service use case (add a service)**

|  |
| --- |
| A ChocAn operator wishes to add a new service.   1. The ChocAn operator selects the option to add a new service, and enters the new service’s details:   Service name (max 20 characters)  Service code (max 6 digits)  Service fee (max $999.99)   1. The system records the new service details. 2. The system displays the new service details. |

**Figure 1.11: Second extended scenario for the Maintain Service use case (update a service)**

|  |
| --- |
| A ChocAn operator wishes to update an existing service.   1. The ChocAn operator enters the existing service code. 2. The system searches for the details for the service and displays the details. 3. The ChocAn operator edits the details that must be changed. 4. The system updates the service details. |
| **Possible Alternative**   1. In step 2, the system cannot find the service code. It displays an error message, and the use case is terminated. |

**Figure 1.12: Third extended scenario for the Maintain Service use case (delete a service)**

|  |
| --- |
| A ChocAn operator wishes to delete an existing service.   1. The ChocAn operator enters the existing service code. 2. The system searches for the details for the service and displays the details. 3. The ChocAn operator selects the option to delete the service. 4. The system deletes the service details. |
| **Possible Alternative**   1. In step 2, the system cannot find the service code. It displays an error message, and the use case is terminated. |

**Figure 1.13: First extended scenario for the Request Report use case (provider report)**

|  |
| --- |
| The ChocAn manager wishes to print provider report.   1. The ChocAn manager selects the provider report, and enters the end date of the week he or she requires and the provider number. 2. The system generates a provider report including the following information:   Provider name (max 25 characters)  Provider number (max 9 digits)  Provider street address (max 25 characters)  Provider city (max 14 characters)  Provider province (2 letters)  Provider zip code (7 digits)  For each service provided, the following information, sorted according to claim submission date and time, is included:  Service date (MM-DD-YYYY)  Claim submission date and time (MM-DD-YYYY HH:MM:SS)  Member name (max 25 characters)  Member number (max 9 digits)  Service code (max 6 digits)  Service fee (max $999.99)  Total number of consultations with members (3 digits)  Total fee for week (max $99,999.99)   1. The manager chooses to print the report. 2. The system prints the report. |
| **Possible Alternatives**   1. In step 1, the ChocAn manager enters an invalid provider number. The system displays an error message and the use case terminates. 2. In step 3, the ChocAn manager chooses not to print the report. The use case terminates. |

**Figure 1.14: Second extended scenario for the Request Report use case (member report)**

|  |
| --- |
| A ChocAn manager wishes to print a member report.   1. The ChocAn manager selects the member report, and enters the end date of the week he or she requires and the member number. 2. The system generates a member report including the following information:   Member name (max 25 characters)  Member number (max 9 digits)  Member street address (max 25 characters)  Member city (max 14 characters)  Member province (2 letters)  Member zip code (7 digits)  For each service provided, the following information, sorted according to service date, is included:  Service date (MM-DD-YYYY)  Provider name (max 25 characters)  Service name (max 20 characters)   1. The manager chooses to print the report. 2. The system prints the report. |
| **Possible Alternatives**   1. In step 1, the ChocAn manager enters an invalid member number. The system displays an error message and the use case terminates. 2. In step 3, the ChocAn manager chooses not to print the report. The use case terminates. |

**Figure 1.15: Third extended scenario for the Request Report use case (accounts payable report)**

|  |
| --- |
| The ChocAn manager wishes to print an accounts payable report.   1. The ChocAn manager selects the accounts payable report, and enters the end date of the week he or she requires. 2. The system generates an accounts payable report including the following information:   For each provider to be paid that week:  Provider name (25 characters)  Number of consultations (max 6 digits)  Total fee (max $99,999.99)  Total number of providers who provided services (max 6 digits)  Total number of consultations (max 9 digits)  Overall total fee (max $99,999.99)   1. The manager chooses to print the report. |
| **Possible Alternative**   1. In step 3, the ChocAn manager chooses not to print the report. The use case terminates. |

**Figure 1.16: Scenario for the Run Accounting Procedure use case**

|  |
| --- |
| The accounting procedure is run every week at midnight on Friday.   1. For each provider, the system generates a provider report (First extended scenario for the Request a Report use case – provider report) and sends the report as an email attachment to the provider. 2. For each member, the system generates a member report (Second extended scenario for the Request a Report use case – member report), and sends the report as an email attachment to the member. 3. The system generates an accounts payable report for the ChocAn manager (Third extended scenario for the Request a Report use case – accounts payable report), and sends the report to the manager as an email attachment. 4. The system generates the following EFT data for each provider who must be paid for the week:   Provider name (max 25 characters)  Provider number (max 9 digits)  Total fee for week (max $99,999.99) |

**Entity Class Modeling**

Candidate entity classes are determined using noun extraction.

Description of software product in a single paragraph: The ChocAn system allows providers to verify a person’s ChocAn membership status, and to submit a claim for a service provided to an active member. The data required by the system can be maintained by a ChocAn operator. The ChocAn manager can request various reports.

Identify the nouns/noun phrases: The ChocAn system allows providers to verify a person’s ChocAn membership status, and to submit a claim for a service provided to an active member. The data required by the system can be maintained by a ChocAn operator. The ChocAn manager can request various reports.

With regard to the nouns in the previous paragraph:

* ChocAn System: the software product itself
* Provider: possible entity class
* Person: can be a provider, member, operator or manager.
* Membership status: possible attribute of member
* Claim: possible entity class
* Service: possible entity class
* Data: describes all attributes of entity classes
* ChocAn operator: actor, no information needs to be stored about this actor
* ChocAn manager: actor, no information needs to be stored about this actor

**Figure 1.17: The entity class diagram of the Chocoholics Anonymous product**



|  |  |
| --- | --- |
| **Control Classes:**  ChocAn System  Claim Submitter  Member Maintainer  Provider Maintainer  Service Maintainer  Provider Directory Generator  Provider Report Generator  Member Report Generator  Accounts Payable Report Generator  EFT Report Generator | **Boundary Classes:**  Provider Interface  Operator Interface  Manager Interface  Scheduler Interface  Provider Directory  Provider Report  Member Report  Accounts Payable Report  EFT Data Report |

**Dynamic Modeling:**

Statecharts for the ChocAn product are shown in Parts I, II, III and IV.

**Figure 1.18: The statechart for the Chocoholics Anonymous product**



**Figure 1.19: The Provider subsystem statechart**



**Figure 1.20: Maintenance subsystem statechart**



**Figure 1.21: Reporting subsystem statechart**



We now realize the use cases.

**Use case Manage Session:**

**Figure 1.22: Class diagram showing the classes that realize the Manage Session use case of the ChocAn software product**



**Figure 1.23: Collaboration diagram of the realization of the scenario of the Manage Session use case of the ChocAn software product**



**Figure 1.24: The flow of events of the realization of the scenario of the Manage Session use case of the ChocAn software product**

|  |
| --- |
| The provider supplies the necessary details – (Step 1). The software product finds the provider – (Steps 2, 3) and displays the provider options – (Step 4). The provider selects Quit – (Step 5) and the product ends the session – (step 6). |

**Figure 1.25: A Sequence diagram equivalent to the collaboration diagram from Part VI of the Manage Session use case scenario**



**Use case Verify Member:**

**Figure 1.26: Class diagram showing the classes that realize the Verify Member use case of the ChocAn software product**



**Figure 1.27: Collaboration diagram of the realization of the scenario of the Verify Member use case of the ChocAn software product**



**Figure 1.28: The flow of events of the realization of the scenario of the Manage Session use case of the ChocAn software product**

|  |
| --- |
| The provider supplies the member number – (Step 1). The software product finds the member details – (Steps 2), and displays the member status– (Step 3-4). |

**Figure 1.29: Sequence diagram equivalent to the collaboration diagram for Verify Member use case**



**Use case Submit Claim:**

**Figure 1.30: Collaboration diagram showing the classes that realize the Submit Claim use case (Figure 1.20) of the ChocAn software product**



**Figure 1.31: Collaboration diagram of the realization of the scenario of Figure 1.2 for Submit Claim use case of the ChocAn software product.**



**Figure 1.32: The flow of events of the realization of the scenario of Figure 1.2 for Submit Claim use case of the ChocAn software product.**

|  |
| --- |
| The provider supplies the member number – (Step 1). The software product finds and transfers the corresponding member – (Steps 2 to 3), and displays the member’s status – (Step 4). The provider supplies the service code and other data, which the product uses to find and return the service name to the provider – (Steps 5 to 10). The provider confirms the service – (Steps 11 to 12), and the product creates a new claim and displays the service fee – (Steps 13 to 16). |

**Figure 1.33: A Sequence diagram equivalent to the collaboration diagram of Figure 1.31. The flow of events of the realization of the scenario of Figure 1.32 for Submit Claim use case of the ChocAn software product.**



**Figure 1.34: Class diagram showing the classes that realize the Receive Provider Directory use case (Figure 1.3)**



**Figure 1.35: Communication diagram of the realization of the scenario of Figure 1.3 of the Receive Provider Directory use case of the ChocAn software product.**



**Figure 1.36: The flow of events of the realization of the scenario of Figure 1.3 of the Receive Provider Directory use case of the ChocAn software product.**

|  |
| --- |
| The provider requests the provider directory – (Step 1). The software product transfers the request – (Step 2). The product obtains the service details – (Steps 2-4), and uses this information to create the service directory – (Steps 5, 6), which it then sends by e-mail – (Step 7). It then displays an acknowledgement to the provider – (Steps 8 and 9). |

**Figure 1.37: Sequence diagram equivalent to the Collaboration diagram of Figure 1.35. The flow of events is shown in Figure 1.36.**



**Use case Maintain Member:**

**Consider the add new member scenario of Figure 1.4. The Collaboration diagram is shown in Figure 1.39, the flow of events diagram in Figure 1.40 and the corresponding sequence diagram in Figure 1.41.**

**Figure 1.38: Class diagram showing the classes that realize the Maintain Member use case – add new member (Figure 1.4) of the ChocAn software product**



**Figure 1.39: Collaboration diagram of the realization of the Add a Member scenario of Figure 1.4 of the Maintain Member use case of the ChocAn software product.**



**Figure 1.40: The flow of events of the realization of the scenario of Figure 1.4 of the Maintain Member use case (add new member) of the ChocAn software product.**

|  |
| --- |
| The member supplies the necessary details, entered by the ChocAn operator – (Step 1). The software product transfers the details, creates a new member record – (Steps 2 to 3), and returns and displays the new member number – (Steps4-6). |

**Figure 1.41: A sequence diagram equivalent to the Collaboration diagram of Figure 1.39.**



Now consider the update a member scenario of the Maintain Member use case (Figure 1.5). The Collaboration diagram is shown in Figure 1.42, the flow of events in Figure 1.43 and the corresponding sequence diagram in Figure 1.44.

**Figure 1.42: Collaboration diagram of the realization of the update a member scenario of Figure 1.5 of the Maintain Member use case of the ChocAn software product.**



**Figure 1.43: The flow of events of the realization of the scenario of Figure 1.5 of the Maintain Member use case of the ChocAn software product.**

|  |
| --- |
| The member supplies the changed details, entered by the ChocAn operator – (Step 1). The software product transfers and updates the details – (Step 2 and 3), and returns and displays an acknowledgement – (Steps 4 to 6). |

**Figure 1.44: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.42.**



Finally, consider the delete a member scenario of the Maintain Member use case (Figure 1.6). The Collaboration diagram is shown in Figure 1.45, the flow of events in Figure 1.46 and the corresponding sequence diagram in Figure 1.47.

**Figure 1.45: A Collaboration diagram of the realization of the delete a member scenario of Figure 1.6 of the Maintain Member use case of the ChocAn software product.**



**Figure 1.46: The flow of events of the realization of the scenario of Figure 1.6 of the Maintain Member use case of the ChocAn software product.**

|  |
| --- |
| The member supplies his or her member number, entered by the ChocAn operator – (Step 1). The software product transfers and deletes the member details – (Step 2 and 3), and returns and displays an acknowledgement – (Steps 4 to 6). |

**Figure 1.47: A sequence diagram of the realization of the delete a member scenario of Figure 1.6 of the Maintain Member use case of the ChocAn software product.**



**Use case Maintain Provider:**

Consider the “Add a New Provider” scenario of Figure 1.7. The Collaboration diagram is shown in Figure 1.49, the flow of events in Figure 1.50 and the corresponding sequence diagram in Figure 1.51.

**Figure 1.48: Class diagram showing the classes that realize the Maintain Provider use case (Figure 1.7) of the ChocAn software product**



**Figure 1.49: A Collaboration diagram of the realization of the add a provider scenario of Figure 1.7 of the Maintain Provider use case of the ChocAn software product.**



**Figure 1.50: The flow of events of the realization of the scenario of Figure 1.7 of the Maintain Provider use case of the ChocAn software product.**

|  |
| --- |
| The provider supplies the necessary details, entered by the ChocAn operator – (Step 1). The software product transfers the details and creates a new provider record – (Step 2 and 3), and returns and displays the provider number – (Steps 4 to 6). |

**Figure 1.51: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.49.**



Now consider the update a provider scenario of the Maintain Provider use case (Figure 1.8). The Collaboration diagram is shown in Figure 1.52, the flow of events in Figure 1.53 and the corresponding sequence diagram in Figure 1.54.

**Figure 1.52: A Collaboration diagram of the realization of the update a provider scenario of Figure 1.8 of the Maintain Provider use case of the ChocAn software product.**



**Figure 1.53: The flow of event of the realization of the update a provider scenario of Figure 1.8 of the Maintain Provider use case of the ChocAn software product.**

|  |
| --- |
| The provider supplies the changed details, entered by the ChocAn operator – (Step 1). The software product transfers and updates the details – (Step 2 and 3), and returns and displays an acknowledgment – (Steps 4 to 6). |

**Figure 1.54: A sequence diagram equivalent to the Collaboration diagram of Figure 1.52.**



Finally, consider the delete a provider scenario of the Maintain Provider use case (Figure 1.9). The Collaboration diagram is shown in Figure 1.55, the flow of events in Figure 1.56 and the corresponding sequence diagram in Figure 1.57.

**Figure 1.55: Collaboration diagram of the realization of the delete provider scenario of Figure 1.9 of the Maintain Provider use case of the ChocAn software product.**



**Figure 1.56: The flow of events of the realization of the scenario of Figure 1.9 of the Maintain Provider use case of the ChocAn software product.**

|  |
| --- |
| The provider supplies his or her number, entered by the ChocAn operator – (Step 1). The software product transfers and deletes the provider details – (Steps 2 and 3), and returns and displays an acknowledgement – (Steps 4 to 6). |

**Figure 1.57: A sequence diagram equivalent to the Collaboration diagram of Figure 1.55.**



**Maintain Service – use case:**

Consider the “Add a New Service” scenario of Figure 1.10. The Class diagram is shown in Figure 1.58, the Collaboration diagram in Figure 1.59, the flow of events in Figure 1.60 and the corresponding sequence diagram in Figure 1.61.

**Figure 1.58: Class diagram showing the classes that realize the Maintain Service use case of the ChocAn software product.**



**Figure 1.59: A Collaboration diagram of the realization of the ‘Add a service’ scenario of Figure 1.10 of the Maintain Service use case of the ChocAn software product.**



**Figure 1.60: The flow of events of the realization of the scenario of Figure 1.10 of the Maintain Service use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the necessary service details, entered by the ChocAn operator – (Step 1). The software product transfers the details and creates a new service record – (Steps 2 and 3), and sends an acknowledgement to the manager – Steps (4 to 6). |

**Figure 1.61: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.59.**



Now consider the “Update a service” scenario of the Maintain Service use case (Figure 1.11). The Class diagram is shown in Figure 1.58, the Collaboration diagram is shown in Figure 1.62, the flow of events in Figure 1.63 and the corresponding sequence diagram in Figure 1.64.

**Figure 1.62: A Collaboration diagram of the realization of the “Update a service” scenario of Figure 1.11 of the Maintain Service use case of the ChocAn software product.**



**Figure 1.63: The flow of events of the realization of the scenario of Figure 1.11 of the Maintain Service use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the changed service details, entered by the ChocAn operator – (Step 1). The software product transfers and updates the details – (Steps 2 to 3), and returns and displays an acknowledgement – (Steps 4 to 6). |

**Figure 1.64: A Sequence diagram equivalent to the collaboration diagram of Figure 1.62.**



Now consider the “Delete a service” scenario of the Maintain Service use case (Figure 1.12). The Class diagram is shown in Figure 1.58, the Collaboration diagram is shown in Figure 1.65, the flow of events in Figure 1.66 and the corresponding sequence diagram in Figure 1.67.

**Figure 1.65: A Collaboration diagram of the realization of the “Delete a service” scenario of Figure 1.12 of the Maintain Service use case of the ChocAn software product.**



**Figure 1.66: The flow of events of the realization of the scenario of Figure 1.12 of the Maintain Service use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the service code, entered by the ChocAn operator – (Step 1). The software product transfers and deletes the service – (Steps 2 to 3), and returns and displays an acknowledgement – (Steps 4 to 6). |

**Figure 1.67: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.65.**



**Request Report – use case:**

Consider the “Provider Report” scenario of Figure 1.13. The Class diagram is shown in Figure 1.68, the Collaboration diagram in Figure 1.69, the flow of events in Figure 1.70 and the corresponding sequence diagram in Figure 1.71.

**Figure 1.68: Class diagram showing the classes that realize the Request Report use case of the ChocAn software product.**



**Figure 1.69: A Collaboration diagram of the realization of the ‘Provider Report’ scenario of Figure 1.13 of the Report Request use case of the ChocAn software product.**



**Figure 1.70: The flow of events of the realization of the scenario of Figure 1.13 of the Provider Report use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the relevant provider number and end date – (Step 1). The software product transfers the details, finds and returns the provider – (Steps 3 and 4), then finds and returns the relevant claims – (Steps 5 and 6). It also finds and returns each member who has received services – (Steps 7 and 8), and the relevant services – (Steps 9 and 10). The software product then creates the report and displays it to the manager – (Steps 11 to 14). The manager then requests a printout, which is performed – (Steps 15 to 17), and an acknowledgement displayed – (Steps 18 to 19). |

**Figure 1.71: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.69.**



Now consider the “Member Report” scenario of the Request Report use case (Figure 1.14). The collaboration diagram is shown in Figure 1.72, the flow of events in Figure 1.73 and the corresponding sequence diagram in Figure 1.74.

**Figure 1.72: A Collaboration diagram of the realization of the ‘Member Report’ scenario of Figure 1.14 of the Report Request use case of the ChocAn software product.**



**Figure 1.73: The flow of events of the realization of the scenario of Figure 1.14 of the Provider Report use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the relevant member number and end date – (Step 1). The software product transfers the details, finds and returns the member – (Steps 2 to 4), then finds and returns the relevant claims – (Steps 5 and 6). It also finds and returns each provider who has provided services to that member – (Steps 7 and 8), and the relevant services – (Steps 9 and 10). The software product then creates the report and displays it to the manager – (Steps 11 to 14). The manager then requests a printout, which is performed – (Steps 15 to 17), and an acknowledgement displayed – (Steps 18 and 19). |

**Figure 1.74: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.72.**



Finally consider the “Accounts Payable” scenario of the Request Report use case (Figure 1.15). The collaboration diagram is shown in Figure 1.75, the flow of events in Figure 1.76 and the corresponding sequence diagram in Figure 1.77.

**Figure 1.75: A Collaboration diagram of the realization of the ‘Accounts Payable Report’ scenario of Figure 1.15 of the Report Request use case of the ChocAn software product.**



**Figure 1.76: The flow of events of the realization of the scenario of Figure 1.15 of the Provider Report use case of the ChocAn software product.**

|  |
| --- |
| The manager supplies the end date – (Step 1). The software product transfers the end date, finds and returns the providers who provided services – (Steps 2 to 4), then finds and returns the relevant claims – (Steps 5 and 6). It also finds and returns the services provided by each provider – (Steps 7 and 8). The software product then creates the report and displays it to the manager – (Steps 9 to 12). The manager then requests a printout, which is performed – (Steps 13 to 15), and an acknowledgement displayed – (Steps 16 and 17). |

**Figure 1.77: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.75.**



**Run Accounting Procedure – use case:**

Consider the “EFT Report” scenario of Figure 1.16. The Class diagram is shown in Figure 1.78, the Collaboration diagram in Figure 1.79, the flow of events in Figure 1.80 and the corresponding sequence diagram in Figure 1.81.



**Figure 1.79: A Collaboration diagram of the realization of the ‘EFT Report’ scenario of Figure 1.16 of the Run Accounting Procedure use case of the ChocAn software product.**



**Figure 1.80: The flow of events of the realization of the scenario of Figure 1.16 of the Run Accounting Procedure use case of the ChocAn software product.**

|  |
| --- |
| The scheduler runs the accounting procedure – (Step 1). The software product transfers the end date, finds and returns each provider – (Steps 2 to 4), then finds and returns the relevant claims – (Steps 5 to 6). It also finds each service and returns the corresponding fee – (Steps 7 to 8). The software product creates and saves the report (Steps 9 to 11) and sends an acknowledgement to the scheduler – (Steps 12 to 13). |

**Figure 1.81: A Sequence diagram equivalent to the Collaboration diagram of Figure 1.79.**



**Increment the class diagram**

In the course of realizing the various use cases, interrelationships between many of the classes became apparent; these interrelationships are reflected in the class diagrams of Figures 1.22 through 1.78.

**Figure 1.82: Class diagram combining the class diagrams of Figures 1.22 through 1.34.**



**Figure 1.82: Class diagram combining the class diagrams of Figures 1.22 through 1.34.**



**Figure 1.84: Class diagram combining the class diagrams of Figures 1.68 through 1.78.**



This concludes the analysis workflow.