LAKEHEAD UNIVERSITY

FACULTY OF ENGINEERING

Analysis Workflow of the Chocoholics Anonymous System

For the ChocAn Project

Instructor: Dr. Ayman Diyab
Lab Instructor: Mr. Mohammed AbuFoul

North Light Software

Full name	Student ID
Joshua Whitlock	1272349
Thanh Vu Nguyen	1244225
Kenneth Shahi	1261283

Thunder Bay, 2025

Contents

1 Outline

1. Understand How the System Works
   - Learn how ChocAn works with members providers and the DataCenter
   - Identify what the system does such as checking IDs recording services creating reports and sending payments
   - Break the work into smaller steps to see how data moves
   - Make sure everything connects correctly between users and the system
2. Identify Actors and Use Cases
   - List all people and systems that interact with ChocAn such as members providers managers the DataCenter Acme Accounting Services and the ProviderTerminal
   - Explain what each actor does
   - Identify all main use cases and use Record Service Provided as the main example for the analysis
   - Make simple diagrams to show how actors and system functions connect
3. Find Nouns and Create Classes
   - Find important nouns in the project description and requirements that show people data or actions
   - Turn these nouns into classes or attributes
   - Group classes as Entity Boundary or Control
   - Make a table to show what each class does including the ServiceRecord class used in Record Service Provided
4. Make CRC Cards and Responsibilities
   - Make CRC cards to show what each class does and who it works with using Record Service Provided as the main example
   - Keep each class focused on one main task
   - Make sure every class connects to at least one use case
5. Show How Classes Interact
   - Explain step by step how classes talk to each other during Record Service Provided
   - Include normal and error cases such as invalid codes or suspended members
   - Show how control moves from user input to system logic and stored data
6. Write Data Rules and Limits
   - List all data rules such as ID length fee size and date format
   - Write when special system tasks happen such as the Friday midnight batch
   - Make a table that summarizes the data rules and limits
7. Plan the Test Workflow
   - Write test cases for the Record Service Provided use case
   - Use correct and incorrect inputs
   - Write what the expected result should be
   - Make a checklist to confirm reports EFT files and updates work correctly
8. Finish and Review the Analysis
   - Review all use cases actors and classes

- Make sure all descriptions are clear and correct
- Check that all information fits together well
- Prepare tables CRC cards and test results for submission

2 Actors

This section lists the main people and systems that take part in ChocAn operations. Each actor sends or receives information during validation service entry reporting and weekly accounting.

Actor	Description
Member	A person who pays monthly fees and receives health services. Each member has a nine digit ID card and a status that may be valid suspended or invalid
Provider	A health worker such as a dietitian doctor or exercise expert who uses the terminal to record services and submit billing information
ProviderTerminal	The interface used by providers to validate IDs enter service information and communicate with the DataCenter
DataCenter	The main computer that manages all data for members and providers validates IDs records services creates reports and prepares weekly accounting files
ChocAn Operator	A system operator who adds updates or deletes member provider and service information in the DataCenter
Manager Accounts Payable	Reviews the weekly reports created by the system and receives a summary of all consultations and provider totals
Acme Accounting Services	Handles membership fee payments and updates member status every night

3 Use Cases

This part describes what the system does for each actor. Each use case represents a main task such as validating IDs recording services creating reports and running the weekly batch. Record Service Provided is used as the main example for the analysis. The Save Service use case shown in the requirements diagram refers to the same provider service entry behaviour as Record Service Provided.

Table 2: Use cases and brief description.

Use Case	Brief Description
Verify Member Number	Allows a provider to check if a member number is valid or suspended
Verify Provider Number	Allows a provider to confirm that their provider number is registered in the ChocAn system
Request Provider Directory	Allows the provider to view the list of services service codes and fees
Lookup Service Code	Retrieves the service name for a given service code
Lookup Fee	Retrieves the fee for a given service code
Record Service Provided	Allows a provider to enter a completed service including date service code and comments then store it in the DataCenter
Retrieve Services	Gets all services a provider completed within the week
Weekly Report Generation	Produces weekly reports for members providers and the manager
Weekly Accounting	Sends weekly fee totals and provider payments amounts to Acme Accounting Services
Print Financial Report	Prints the weekly financial summary
Add Provider	Allows the ChocAn operator to add a provider
Delete Provider	Allows the ChocAn operator to remove a provider
Update Provider	Allows the ChocAn operator to update provider information
Add Member	Allows the ChocAn operator to add a member
Delete Member	Allows the ChocAn operator to remove a member
Update Member	Allows the ChocAn operator to update member information
Add Service Code	Allows the ChocAn operator to add a new service code
Update Service Code	Allows the ChocAn operator to update a service code
Delete Service Code	Allows the ChocAn operator to delete a service code
Get Weekly Fees	Retrieves total fees for the current or previous week

4 Noun Extraction

4.1 Use Case List

Table 3: Use cases and their brief descriptions.

Use Case	Brief Description
Verify Member Number	Allows a provider to check if a member number is valid or suspended
Verify Provider Number	Allows a provider to confirm that their provider number is registered in the ChocAn system
Request Provider Directory	Allows the provider to view the list of services service codes and fees
Lookup Service Code	Retrieves the service name for a given service code
Lookup Fee	Retrieves the fee for a given service code
Record Service Provided	Allows a provider to enter a completed service including date service code and comments then store it in the DataCenter
Retrieve Services	Gets all services a provider completed within the week
Weekly Report Generation	Produces weekly reports for members providers and the manager
Weekly Accounting	Sends weekly fee totals and provider payments amounts to Acme Accounting Services
Print Financial Report	Prints the weekly financial summary
Add Provider	Allows the ChocAn operator to add a provider
Delete Provider	Allows the ChocAn operator to remove a provider
Update Provider	Allows the ChocAn operator to update provider information
Add Member	Allows the ChocAn operator to add a member
Delete Member	Allows the ChocAn operator to remove a member
Update Member	Allows the ChocAn operator to update member information
Add Service Code	Allows the ChocAn operator to add a new service code
Update Service Code	Allows the ChocAn operator to update a service code
Delete Service Code	Allows the ChocAn operator to delete a service code
Get Weekly Fees	Retrieves total fees for the current or previous week

4.2 Use Case Noun List

Contains nouns extracted from use cases.

Table 4: Table containing nouns extracted from use cases.

Use Case	Nouns
Verify Member Number (Provider)	Providers, Person, ChocAn, membership.
Verify Provider Number (Provider)	Providers, ChocAn system, accounting.
Request Provider Directory	Providers, services, service number, fees.
Lookup Service Code	Terminal, ChocAn DataCenter, list, service codes, descriptions.
Lookup Fee	Service Code, service fee.
Bill ChocAn	Provider's Terminals, bill, ChocAn database, services, codes, fees, provider number.
Calculate Weekly Fee	List, Service Codes, Terminal, Week, Fees.
Check Member Number (Server)	Member Number, Database.
Check Provider Number (Server)	Provider Number, ChocAn Database.
Check Service Code	Server, Service Code, Descriptions, Terminal.
Check Fee	List, Service Code, Fees.
Store Weekly Fees	List, Services, Fees, Fee Total, Provider, Week.
Weekly Report Generation	Provider, Fees, Week, Database.
Weekly Accounting	Fees, Report Generation, Acme Accounting Services.
Print Financial Report	Fees, Database, Week.

Use Case	Nouns
Add Provider	ChocAn Operator, provider, ChocAn database.
Delete Provider	ChocAn Operator, Provider, ChocAn database.
Update Provider	ChocAn Operator, Provider, Details, ChocAn database.
Add Member	ChocAn Operator, Member, ChocAn database.
Delete Member	ChocAn Operator, Member, ChocAn database.
Update Member	ChocAn Operator, Details, Member, ChocAn database.
Add Service Code	ChocAn Operator, Service Code, Service Directory.
Update Service Code	ChocAn Operator, Service Code, Service Directory.
Delete Service Code	ChocAn Operator, Service Code, Service Directory.
Get Weekly Fees	Fees, Database, Week.
Retrieve Services	Services, Provider, Week.
Save Service	Provider, Services, Terminal, ChocAn DataCenter.

4.3 Noun List

A list of nouns extracted from the use cases.

- ☒ Providers
- ☒ Person
- ☒ ChocAn
- ☒ Membership
- ☒ ChocAn System
- ☒ Accounting
- ☒ Services
- ☒ Service Number
- ☒ Fees
- ☒ Terminal
- ☒ ChocAn DataCenter
- ☐ List
- ☒ Service Codes
- ☐ Descriptions
- ☒ Service Fee
- ☒ Provider's Terminals
- ☒ Bill
- ☒ ChocAn Database
- ☒ Codes
- ☒ Provider Number
- ☐ Week
- ☒ Member Number
- ☒ Database
- ☒ Server
- ☒ Fee Total
- ☒ Report Generation
- ☒ Acme Accounting Services
- ☒ ChocAn Operator
- ☐ Details
- ☒ Member
- ☒ Service Directory

4.4 Noun Grouping

A section grouping the nouns extracted from the use cases.

4.4.1 Member

- Person/Member

- Member Number
- Membership

4.4.2 Provider

- Provider
- Provider Number
- Terminal/Provider's Terminal

4.4.3 Provider Services

- Services
- Codes/Service Codes
- Service Directory
- Service Number
- Service Fees

4.4.4 Service Fees

- Fees/Service Fees
- Fee Total

4.4.5 Billing

- Accounting
- Report Generation
- Bill
- Acme Accounting Services

4.4.6 ChocAn Database

- ChocAn/Server/Database/ChocAn Database/ChocAn System/ChocAn DataCenter

4.4.7 ChocAn Employees

- ChocAn Manager
- ChocAn Operator

4.5 Candidate Class List

- Member
- Provider
- Services/Provider Services
- Service Fees
- Billing
- ChocAn Database
- ChocAn Employees

4.6 Candidate Class Expansion

4.6.1 ProviderTerminal

4.6.1.1 Terminal

- Attributes:
  - Provider
  - reportRan: bool

- Methods:
  - Provider init(providerID)
  - verifyProviderID(providerID)
  - verifyMemberID(memberID)
  - recordTransaction(serviceID, memberID, time)
  - updateServiceList()
  - emailServiceList()
  - emailReceipt()
  - checkProviderTimer()

4.6.1.2 Member

- Attributes: public:
  - memberID: int private
  - memberName: string
  - memberAddress: string
  - memberPhone: string
  - memberEMail: string

4.6.1.3 Provider

- Attributes: public
  - providerID: int private
  - providerName: string
  - providerAddress: string
  - providerPhone: string
  - providerEMail: string

4.6.1.4 ServiceInstance

- Attributes:
  - rendered: Service
  - customer: Member
  - provider: Provider
  - time: arr(Date, Time)
- Methods:
  - save() // Saves an instance of a service locally

4.6.1.5 ServiceList

- Attributes:
  - List: dict{id: Service}
- Methods:
  - refresh() // Re-pulls from the database.

4.6.1.6 Service

- Attributes:
  - ID: int
  - Fee: float(2)
  - Name: string
  - Desc: string

4.6.1.7 ProviderReport

- Attributes:
  - weeklyServices: arrserviceInstance
- Methods:
  - sum(date, date) // Gets all service instances in a range and totals // the fee.

4.6.2 ChocAn Mainframe

4.6.2.1 EmployeeTerminal

- Attributes:
  - employee: ChocAnEmployee
  - dbInstance: ChocAnDB
- Methods:
  - ChocAnEmployee, ChocAnDB init(EmployeeID)
  - registerMember()
  - editMemberInfo()
  - viewMemberInfo()
  - suspendMember()
  - hireEmployee()
  - updateEmployeeInfo()
  - viewEmployeeInfo()
  - fireEmployee()
  - registerProvider()
  - editProviderInfo()
  - viewProviderInfo()
  - suspendProvider()
  - registerService()
  - editServiceInfo()
  - viewServiceInfo()
  - suspendService()
  - initiateManagerReport()

4.6.2.2 MainScheduler

- Attributes:
  - timerRan: bool
- Methods
  - checkAcmeTimer()

4.6.2.3 ReportGenerator

- Attributes:
  - services: arrServiceInstance
  - date: date
- Methods
  - generateReport(ChocAnEmployee, date1, date2, providerID(optional)) // Called by ChocAn managers over a date range. Can specify a provider.
  - generateWeeklyReport() // Automatically called weekly for the billing class.

4.6.2.4 Billing

- Attributes:
  - payouts: dict{providerID: FeeTotal}
  - period: date
- Methods:
  - weeklyBill(date) // Gets the total fee for each provider for the previous week from the ReportGenerator and sends it to ACMEAccounting.

4.6.2.5 ChocAnDB

- Methods:
  - addEmployee()

  - removeEmployee(employeeID)
  - updateEmployee(employeeID)
  - viewEmployee(employeeID)
  - addMember()
  - removeMember(memberID)
  - updateMember(memberID)
  - viewMember(memberID)
  - addProvider()
  - removeProvider(providerID)
  - updateProvider(providerID)
  - viewProvider(providerID)
  - addService()
  - removeService(serviceID)
  - updateService(serviceID)
  - viewService(serviceID)
  - getServiceList()
  - getWeeklyServices(date)
  - getServices(date1, date2)

4.6.2.6 ChocAnEmployee

- Attributes:
  - ID: int
  - name: string
  - role: string

5 Class Diagram

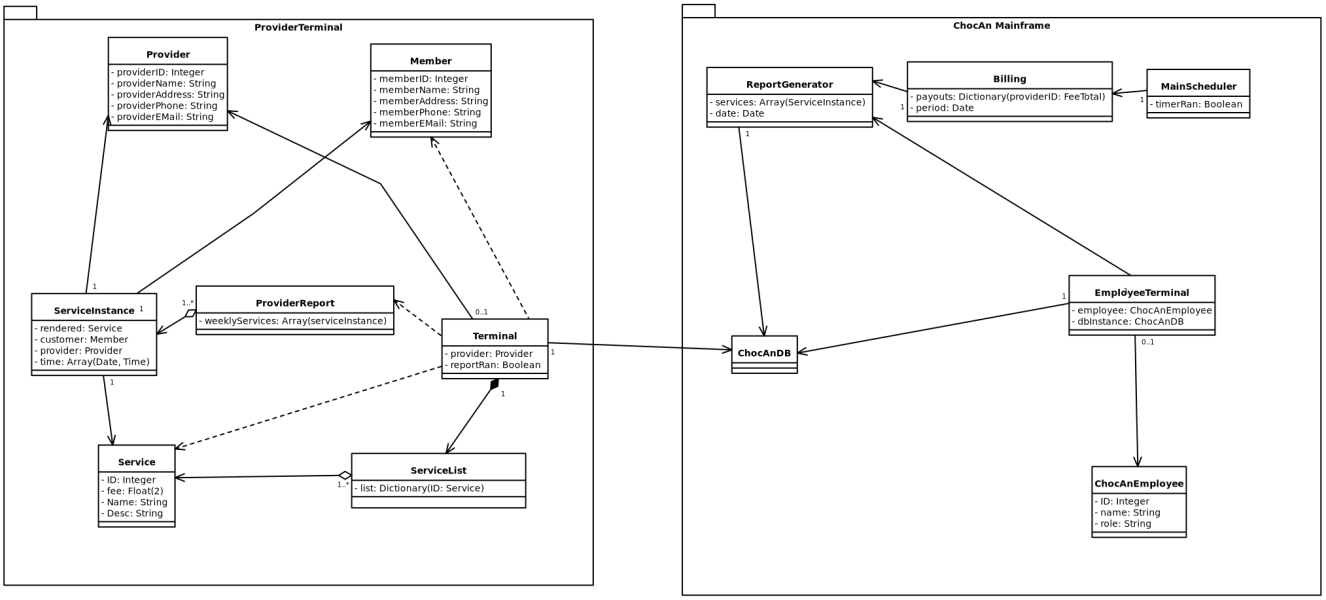

Figure 1: Class Diagram

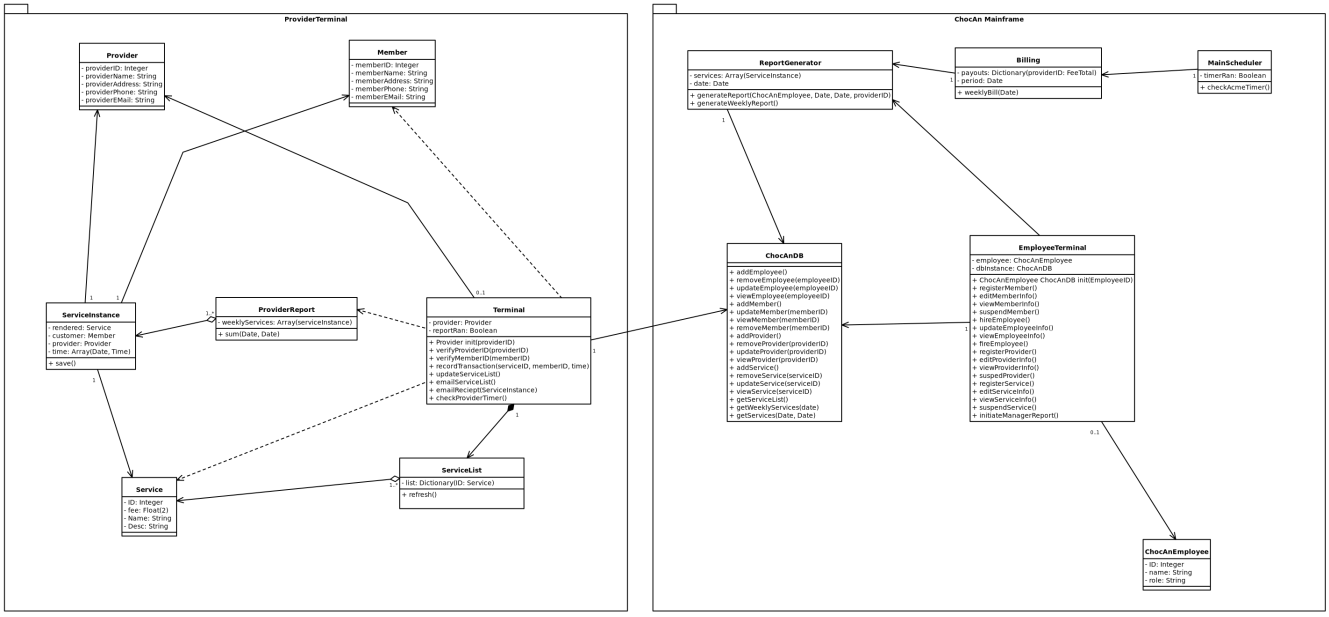

Figure 2: Detailed Class Diagram

6 Entity, Boundary and Control Classes

Type	Class	Purpose
Entity	Member	Stores member information and membership status
Entity	Provider	Stores provider information and provider number
Entity	Service	Represents a service offered by ChocAn including service name, service code and fee used during service lookup
Entity	ServiceRecord	Stores one completed service with provider number member number date of service service code fee and comment
Entity	Directory	Holds all service names service codes and fees used during Record Service Provided
Entity	Report	Holds weekly report information for members providers and the manager
Entity	EFTFile	Holds weekly payment information for each provider
Boundary	ProviderTerminal	Receives provider input validates IDs and displays service information
Control	ValidationControl	Handles provider and member number validation through the DataCenter
Control	BillingControl	Handles service lookup confirmation building of ServiceRecord and submission
Control	DataCenter	Stores ServiceRecord entries performs validation supports reporting and runs weekly accounting

7 CRC Cards

Lists the system main classes what each one is responsible for and which other classes it works with. This helps define how data moves between different parts of the system during Record Service Provided.

CLASS ProviderTerminal (Boundary)
RESPONSIBILITY 1. Receive provider number member number date of service service code and comment 2. Display validation messages such as “Validated”, “Invalid Number” and “Member Suspended” 3. Request service name and fee from Directory through BillingControl 4. Send completed service information to BillingControl 5. Display final confirmation after the ServiceRecord is stored COLLABORATION 1. ValidationControl 2. BillingControl

CLASS ValidationControl (Control)
RESPONSIBILITY 1. Validate provider numbers through DataCenter 2. Validate member numbers and membership status 3. Return validation results to ProviderTerminal COLLABORATION 1. DataCenter 2. ProviderTerminal

CLASS BillingControl (Control)
RESPONSIBILITY 1. Receive record service request from ProviderTerminal after validation 2. Request service name and fee from Directory using the service code 3. Build a ServiceRecord with provider number member number date service code fee and comment 4. Send the ServiceRecord to DataCenter for storage 5. Return success or failure to ProviderTerminal COLLABORATION 1. ProviderTerminal 2. Directory 3. DataCenter 4. ServiceRecord

CLASS DataCenter (Control)
RESPONSIBILITY 1. Store ServiceRecord entries with timestamps 2. Validate provider and member numbers 3. Retrieve service information for reporting and accounting 4. Support weekly processing and report generation COLLABORATION 1. ProviderTerminal 2. ValidationControl 3. BillingControl 4. Directory 5. Member

CLASS DataCenter (Control)
6. Provider 7.ServiceRecord

CLASS ServiceRecord (Entity)
RESPONSIBILITY 1. Hold all fields for one recorded service provider number member number date of service service code fee comment and timestamp 2. Provide data for weekly reporting and accounting COLLABORATION 1. BillingControl 2. DataCenter

8 Example Use Case Record Service Provided

This section shows one use case in full detail. It explains step by step how the Record Service Provided process happens from logging in to saving a completed record.

8.1 Step by Step Description

1. Provider enters provider number into the ProviderTerminal
2. ProviderTerminal sends the provider number to ValidationControl
3. ValidationControl checks the DataCenter and returns “Validated” or “Invalid Number”
4. Provider enters member number into the ProviderTerminal
5. ValidationControl checks the DataCenter for member status
6. If the member is suspended the ProviderTerminal shows Member Suspended and stops
7. Provider enters the date of service the service code and a comment
8. BillingControl requests the service name and fee from the Directory
9. Directory returns the service name and fee
10. ProviderTerminal displays the service name and fee for confirmation
11. Provider confirms the service information
12. ProviderTerminal sends the service details to BillingControl
13. BillingControl builds a ServiceRecord using the provider number member number date service code fee and comment
14. BillingControl sends the completed ServiceRecord to the DataCenter
15. DataCenter stores the ServiceRecord with a timestamp
16. ProviderTerminal displays Service Recorded Successfully

9 Weekly Processing

1. At midnight on Friday the system starts automatic weekly processing
2. The DataCenter reads all ServiceRecords from that week
3. A Member Report is created for each member showing all services they received
4. A Provider Report is created for each provider with the list of services performed and the total fees
5. A Summary Report is created showing the total number of providers total consultations and the total amount paid
6. An EFT File is created for each provider containing the amount to be paid for the week
7. The reports and EFT files are stored so they can be reviewed by the manager

All reports stay stored in files for review and testing.

10 Data Rules and Limits

Table 11: Data restrictions.

Field	Rule or Limit
Member Number	9 digits
Provider Number	9 digits
Service Code	6 digits
Service Name	Up to 20 characters
Comment	Up to 100 characters
Fee	Up to $999.99
Weekly Total	Up to $99,999.99
Date Format	MM/DD/YYYY
Timestamp Format	MM/DD/YYYY HH:MM:SS
Weekly Batch Run	Every Friday at 12 AM
Input and Output	Keyboard input and screen output only
File Output	One file per Member Report Provider Report Summary Report and EFT File

11 Message Flow (Step by Step Version)

This message flow shows how information moves through the system when a provider records a service for a member, using the Record Service Provided use case.

11.1 Step by Step Message Flow

1. The Provider enters their provider number into the ProviderTerminal
2. The ProviderTerminal sends the provider number to ValidationControl
3. ValidationControl asks the DataCenter to verify the provider number
4. The DataCenter returns the validation result to ValidationControl
5. The ProviderTerminal displays “Validated” or “Invalid Number”
6. The Provider enters the member number
7. The ProviderTerminal sends the member number to ValidationControl
8. ValidationControl checks the member status through the DataCenter
9. The ProviderTerminal displays “Validated”, “Invalid Number” or “Member Suspended”
10. If both numbers are valid the Provider enters the date of service service code and a comment
11. The ProviderTerminal sends the service code to BillingControl
12. BillingControl requests the service name and fee from the Directory
13. The Directory returns the service name and fee to BillingControl
14. BillingControl sends the service name and fee to the ProviderTerminal for confirmation
15. After the Provider confirms the ProviderTerminal sends all service details to BillingControl
16. BillingControl builds a ServiceRecord using the provider number member number date service code fee and comment
17. BillingControl sends the completed ServiceRecord to the DataCenter
18. The DataCenter stores the ServiceRecord with a timestamp
19. The DataCenter sends a confirmation to the ProviderTerminal
20. The ProviderTerminal displays Service Recorded Successfully

11.2 Summary of Interaction Order

From	To	Action
Provider	ProviderTerminal	Enter provider number, member number and service details
ProviderTerminal	ValidationControl	Send ID validation requests
ValidationControl	DataCenter	Validate provider and member numbers
ProviderTerminal	BillingControl	Send service code and service details
BillingControl	Directory	Request service name and fee
BillingControl	DataCenter	Submit completed ServiceRecord
DataCenter	ProviderTerminal	Return validation and confirmation messages
ProviderTerminal	Provider	Display results and confirmation

12 Test Workflow

This section checks if the ChocAn system works properly. Each test follows the main steps of the **Record Service Provided** use case. The goal is to confirm that validation, service lookup and record storage all work correctly.

Table 13: System Test Classes

CLASS ChocAn System Test Class
RESPONSIBILITY 1. Test Verify Provider Number using a valid and invalid provider numbers 2. Test Verify Member Number using a valid, invalid and suspended member numbers 3. Test Lookup Service Code to confirm the correct service name and fee appear 4. Test Record Service Provided to ensure BillingControl builds a ServiceRecord correctly and the DataCenter stores it 5. Test that the ProviderTerminal displays Service Recorded Successfully after storage
COLLABORATION 1. ProviderTerminal 2. ValidationControl 3. BillingControl 4. DataCenter 5. Directory 6. ServiceRecord

12.1 Test Workflow (Sample Test Cases)

Table 14: Test Workflow Example

Test ID	Use Case	Test Condition	Expected Result	Class Responsibility Checked
AB-BC-001	Record Service Provided	Member status is suspended, valid provider, correct date and service code	ProviderTerminal displays **Member Suspended** and stops the transaction. No ServiceRecord is created.	Confirms ValidationControl correctly interprets Member status
AB-BC-002	Record Service Provided	Valid provider and member, non existent 6 digit service code (example 000000)	ProviderTerminal displays **Invalid Service Code**	Confirms BillingControl handles invalid service code lookup
AB-BC-003	Weekly Processing	Friday midnight batch run	DataCenter creates Member Reports, Provider Reports, the Summary Report, and the EFTFile	Confirms DataCenter reporting and weekly output generation

13 Responsibility Table

This section connects each system function to the class that handles it.

Table 15: Responsibility Breakdown

Task	Who Does It
Verify Provider Number	ProviderTerminal ValidationControl DataCenter
Verify Member Number	ProviderTerminal ValidationControl DataCenter
Record Service Provided	ProviderTerminal ValidationControl BillingControl DataCenter
Lookup Service Code	ProviderTerminal BillingControl Directory
Lookup Fee	ProviderTerminal BillingControl Directory
Generate Reports	DataCenter Report
Create EFT File	DataCenter EFTFile
Maintain Member and Provider Records	DataCenter Member Provider
Manage Directory	Directory DataCenter
Run Weekly Batch	DataCenter Report EFTFile
Error Handling for IDs and Service Codes	ProviderTerminal ValidationControl BillingControl DataCenter

Test Checklist

1. Validate provider numbers using valid and invalid inputs
2. Validate member numbers using active, suspended and invalid numbers
3. Test service code lookup through BillingControl and confirm the correct service name and fee appear from the Directory

4. Enter service details and confirm that BillingControl creates a ServiceRecord and the DataCenter stores it correctly

5. Verify all system messages (“Validated”, “Invalid Number” “Member Suspended”)
6. Test invalid inputs (bad IDs, wrong date format and non existent service codes)
7. Run the Friday weekly batch and confirm member, provider and summary reports are generated
8. Check that each provider’s EFTFile total matches their total fees for the week
9. Confirm no emails or real payments are sent (file output only)

14 Deliverables

- Summary of actors and use cases
- Noun extraction and initial class identification including ServiceRecord
- Entity Boundary Control (EBC) class tables
- CRC cards for the main classes
- One fully detailed use case (Record Service Provided)
- Message flow for the chosen use case
- Sequence diagram for the chosen use case
- Weekly processing summary
- Data rules and limits table
- Responsibility table
- Test workflow and test checklist
- Glossary of system terms

15 Glossary

Table 16: Glossary of Terms

Term	Definition
Member	A ChocAn user who receives services. Identified by a nine digit member number. Status may be *Valid*, *Invalid*, or *Suspended*
Provider	A health care professional who provides services to members and records those services through the ProviderTerminal
ProviderTerminal	The interface used by providers to validate IDs, enter service information and communicate with the Data Center
DataCenter	The main computer system that stores member provider and service data processes validations stores ServiceRecord entries and generates weekly reports
Directory	The list of all valid service names service codes and service fees used when providers enter completed services
ServiceRecord	A stored record representing one completed service including date of service, provider number, member number, service code, fee, comments and timestamp
Report	A weekly output file created for each member, provider and a summary report created for the manager showing totals for the week
EFTFile	A weekly “Electronic Funds Transfer” payment file containing each provider’s name, provider’s number and total amount owed for that week
Weekly Batch	The automated processing that runs every Friday at midnight to read all ServiceRecord entries, create weekly reports and create the EFTFile
ValidationControl	The control class that validates provider numbers member numbers and membership status using the DataCenter
BillingControl	The control class that looks up service information builds a ServiceRecord and sends it to the DataCenter
Directory Management	The functions that allow the ChocAn operator to add update or delete service codes from the Directory
Acme Accounting Services	The external organization that updates member payment status every night and tracks suspended and reinstated members

16 Sequence Diagram

17 Communication Diagram

18 Pseudo Code (Record Service Provided) Algorithm

1. Ask ValidationControl to validate the provider number
2. If provider number is invalid: Display “Invalid Number” STOP
3. Ask ValidationControl to validate the member number and get the member status
4. If member status is “Suspended”: Display “Member Suspended” STOP
5. If member status is “Invalid”: Display “Invalid Number” STOP
6. Ask BillingControl to look up the service name and fee from the Directory using the service code
7. If the service code does not exist: Display “Invalid Service Code” STOP
8. Display the service name and fee to the provider and request confirmation
9. If the provider does not confirm: Display “Cancelled” STOP
10. Ask BillingControl to store a new ServiceRecord in the DataCenter The record must include: - provider number - member number - date of service - service code - fee - comment
11. If the DataCenter confirms that the record was stored successfully: Display “Service Recorded Successfully” Else: Display “Error while storing service”

END Algorithm

19 Reflection

(Kenneth) Software Life-Cycle Models: Preparing this presentation made it easier to understand how different software development models guide the building of a system. Many people assume all projects follow the same process, but looking at Agile, XP, Scrum, Synchronize and Stabilize, and the Spiral Model showed that each method has its own purpose. Agile focuses on flexibility, Scrum helps teams stay organized, XP encourages simple design and good communication, and the Spiral Model aims to reduce risks early.

During the ChocAn term project, the ideas from these models became more noticeable. The system was not created all at once. The actors, use cases, diagrams and class responsibilities were updated step by step as the work became clearer. This slow and steady progress is similar to Agile and Scrum, where a system grows through small improvements. The many corrections made to fix confusing parts of the sequence and communication diagrams also match the ideas in XP which support simple design and regular improvement whenever something is unclear.

Even though the ChocAn project is small, comparing it to real development models showed how important it is to use small steps, clear communication and early checks. Making this presentation helped build a better understanding of these models and showed how their ideas naturally appear during system design. This made the project feel more organized and gave a clearer view of how real software development moves forward.