**ChocAn Data Center**

COMP 370

Software Engineering

Dr. Leon Pan

Winter 2017

Group 7

Sufiyan Samnani

300142163

Tanesh Manjrekar

300144052

Vishal Chandra

300142148

Mohamed Furqaan Attarwala

300144394

Table of Contents

[Software Project Management Plan 1](#_Toc478637665)

[Use Case Diagram 3](#_Toc478637666)

[Use Case Descriptions 4](#_Toc478637667)

[Functional and Non-functional Requirements 7](#_Toc478637668)

[Entity-Relationship Diagram (ERD) 8](#_Toc478637669)

[Data Flow Analysis 9](#_Toc478637670)

[Class Diagram 11](#_Toc478637671)

[Class-Responsibility-Class Cards 12](#_Toc478637672)

[Collaboration Diagrams 15](#_Toc478637673)

[Sequence Diagrams (equivalent to collaboration diagrams) 16](#_Toc478637674)

[Black box test Cases 18](#_Toc478637675)

[Screenshots 19](#_Toc478637676)

# Software Project Management Plan

1. **Project Summary**
   1. **Purpose:**

To streamline and simplify the process of storing data about Chocoholics Anonymous (ChocAn) members and providers, and to make it easier for the providers to bill ChocAn for providing services to its members.

* 1. **Scope:**

The ChocAn Data Center software will be accessible by providers and managers, and will be operating within the confines of the physical location of the ChocAn organization.

* 1. **Objectives:**

To automate the following processes:

* Keeping records of the members of Chocoholics Anonymous and its health care professionals (providers),
* Keeping detailed records of service appointments of members with their providers, containing the date of the appointment, fees, etc.,
* Generating weekly member, provider and summary reports each week and on demand for the manager,

through the ChocAn Data Center software.

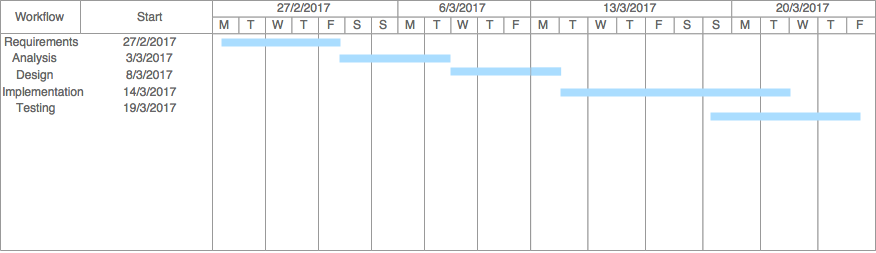
* 1. **Deliverables:**

Product:

The product will have the following major functionalities:

* Providing a login module to providers and managers (which uses a keyboard for input, instead of swiping a card, for the acceptance test)
* Validating provider and manager details and providing them access to the services they respectively need.
* Showing member and provider reports and a summarised report once a week.

Documentation:

* Use case diagram
* Description of all use cases
* Functional and non-functional requirements
* Software Project Management Plan
* Entity-Relationship diagram
* State chart
* Class-Responsibility-Collaborator cards
* Data flow analysis
* Class diagram
* Collaboration diagram
* Sequence diagram equivalent to collaboration diagram
* Black box test cases
  1. **Schedule:**

1. **References:**

Schach, Stephen R. *Object-Oriented and Classical Software Engineering 8th Ed*. 2011. McGraw Hill

Oracle. *DocumentSizeFilter*. 2008. Oracle: <https://docs.oracle.com/javase/tutorial/uiswing/examples/components/TextComponentDemoProject/src/components/DocumentSizeFilter.java>

MigLayout: <http://www.miglayout.com>

Apache Derby: <http://db.apache.org/derby>

Apache POI: <http://poi.apache.org>

JCalendar: <http://toedter.com/jcalendar>

1. **Primary roles and responsibilities:**

Requirements, Analysis, Design: Sufiyan Samnani, Mohamed Furqaan Attarwala

Implementation, Testing: Tanesh Manjrekar, Vishal Chandra

1. **Process Model:**

The waterfall software development model was used in the development of ChocAn Data Center.

# Use Case Diagram



# Use Case Descriptions

|  |  |
| --- | --- |
| Use case ID | Validate\_provider |
| Use case name | Validate Provider Credentials |
| Actors | Provider |
| Brief description | The system asks the provider for his/her credentials and verifies it |
| Triggers | Provider terminal is switched on |
| Pre-conditions | Waiting for provider to input his/her credentials |
| Post-conditions | Provider logged in |
| Flow of events | * Provider is asked to enter his/her provider number upon system start * The system checks whether the provider credentials are valid or not * If valid, then the provider is logged in |

|  |  |
| --- | --- |
| Use case ID | Check\_consultation\_details |
| Use case name | Check consultation details |
| Included use cases | * Validate\_user * Validate\_service |
| Actors | Provider |
| Brief description | The system verifies the member number and the service code |
| Triggers | Provider inputs consultation details |
| Pre-conditions | Waiting for provider to input details |
| Post-conditions | * Validated * Invalid credentials/Member suspended |
| Flow of events | * Provider needs to input the ChocAn member’s number and the service code * The system will validate these details against the credentials stored in the database * The provider verifies that the correct service code has been input * If the verification is successful, then “Validated” is displayed on the screen |

|  |  |
| --- | --- |
| Use case ID | Show\_Provider\_Directory |
| Use case name | Show Provider Directory |
| Actors | Provider |
| Brief description | The system displays the Provider Directory to the provider |
| Triggers | Provider requests for Provider Directory |
| Flow of events | * Provider requests the system to show the Provider Directory to look up the service code * Upon receiving this request, the system displays the Provider Directory |

|  |  |
| --- | --- |
| Use case ID | Validate\_manager |
| Use case name | Validate Manager Credentials |
| Actors | Provider |
| Brief description | The system asks the manager for his/her credentials and verifies it |
| Triggers | Provider terminal is switched on |
| Pre-conditions | Waiting for manager to input his/her number |
| Post-conditions | Manager logged in |
| Flow of events | * Manager is asked to enter his/her credentials upon system start * The system checks whether the credentials are valid or not * If valid, then the provider is logged in |

|  |  |
| --- | --- |
| Use case ID | Show\_reports |
| Use case name | Show generated reports |
| Included use cases | * Generate\_weekly\_member\_report * Generate\_weekly\_provider\_report * Generate\_weekly\_summarized\_report |
| Actors | Manager |
| Brief description | The system generates reports and shows them to the manager |
| Triggers | The manager requests for reports |
| Flow of events | * The manager requests for either weekly member, provider or summary report * The system generates the respective report * The report is then shown to the manager |
| Use case ID | Add\_update\_delete\_members |
| Use case name | Add, update or delete members |
| Actors | Operator |
| Brief description | The operator can add, update or delete ChocAn members |
| Triggers | When operator wants to add, update or delete information about members upon the following events:   * New member signs up * Member resigns * Member details change |
| Pre-conditions | System is in interactive mode during the day |
| Post-conditions | Details added, deleted or updated |
| Flow of events | * System is in interactive mode during the day * The operator chooses to add, update or delete members depending upon the trigger |

|  |  |
| --- | --- |
| Use case ID | Add\_update\_delete\_providers |
| Use case name | Add, update or delete providers |
| Actors | Operator |
| Brief description | The operator can add, update or delete ChocAn health care professionals (providers) |
| Triggers | When operator wants to add, update or delete information about health care providers upon the following events:   * New member signs up * Member resigns * Member details change |
| Pre-conditions | System is in interactive mode during the day |
| Post-conditions | Details added, deleted or updated |
| Flow of events | * System is in interactive mode during the day * The operator chooses to add, update or delete health care professionals depending upon the trigger |

# Functional and Non-functional Requirements

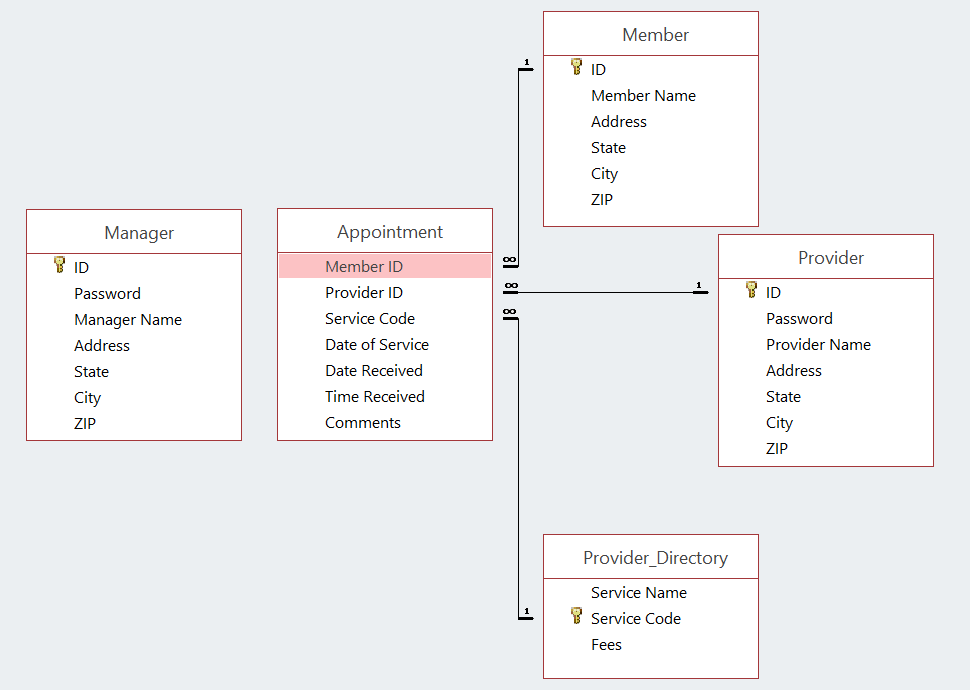
Functional requirements:

* The system shall validate a provider’s credentials and allow him/her to login.
* The system shall allow the provider to input the member number and it will check whether the member number is valid or invalid and whether the member’s status is active or suspended.
* The system shall allow the provider to input the service code and it will check whether the service code is valid or not.
* The system, upon request by a provider, shall generate a file called Provider Directory containing service codes, service names and fees.
* The system shall generate member and provider reports, and a summarized report of those reports, at midnight on every Friday. These reports are available to the manager upon request at any time.

Non-functional requirements:

* Performance: The system software should be fairly responsive.
* Availability: The system software should be available at all times, and even during the day, it should be in an interactive mode.
* Recoverability: The system software should be able to recover (commit and rollback) stored data in case of failure.
* Security: Since the system deals with sensitive information of members, providers and managers, the system should be secure enough to withstand any unauthorised attempts to access the system.
* Usability: The system software should be convenient enough to be used by both, tech-savvy and comparatively inexperienced consultants and managers.

# Entity-Relationship Diagram (ERD)



# Data Flow Analysis

Provider data flow analysis:



Manager data flow analysis:



# Class Diagram



# Class-Responsibility-Class Cards

|  |
| --- |
| CLASS  **Initializer Class** |
| RESPONSIBILITY   1. Start **LoginTerminal class** 2. Start **Scheduler class** 3. Provide default font 4. Provide home directory |
| COLLABORATION   1. **LoginTerminal class** 2. **Scheduler class** |

|  |
| --- |
| CLASS  **LoginTerminal Class** |
| RESPONSIBILITY   1. Validates credentials 2. Start **ProviderTerminal/ManagerTerminal class** |
| COLLABORATION   1. **Initializer class** 2. **ProviderTerminal class** 3. **ManagerTerminal class** 4. **DatabaseHelper class** |

|  |
| --- |
| CLASS  **ProviderTerminal Class** |
| RESPONSIBILITY   1. Validate input 2. Send billing details to **DatabaseHelper class** 3. Display fees 4. Generate Provider Directory |
| COLLABORATION   1. **DatabaseHelper class** 2. **LoginTerminal class** |

|  |
| --- |
| CLASS  **ManagerTerminal Class** |
| RESPONSIBILITY   1. Validate member/provider ID 2. Display member/provider/summary report |
| COLLABORATION   1. **DatabaseHelper class** 2. **LoginTerminal class** 3. **MemberReport class** 4. **ProviderReport class** 5. **SummaryReport class** |

|  |
| --- |
| CLASS  **Scheduler Class** |
| RESPONSIBILITY   1. Generate weekly member/provider/summary reports |
| COLLABORATION   1. **Initializer class** 2. **MemberReport class** 3. **ProviderReport class** 4. **SummaryReport class** |

|  |
| --- |
| CLASS  **MemberReport Class** |
| RESPONSIBILITY   1. Generate member report |
| COLLABORATION   1. **DatabaseHelper class** 2. **ManagerTerminal class** 3. **Scheduler class** |

|  |
| --- |
| CLASS  **ProviderReport Class** |
| RESPONSIBILITY   1. Generate provider report |
| COLLABORATION   1. **DatabaseHelper class** 2. **ManagerTerminal class** 3. **Scheduler class** |
| CLASS  **SummaryReport Class** |
| RESPONSIBILITY   1. Generate summary report |
| COLLABORATION   1. **DatabaseHelper class** 2. **ManagerTerminal class** 3. **Scheduler class** |

|  |
| --- |
| CLASS  **DatabaseHelper Class** |
| RESPONSIBILITY   1. Validation of details 2. Input billing details 3. Fetch appointment details |
| COLLABORATION   1. **LoginTerminal class** 2. **ProviderTerminal class** 3. **ManagerTerminal class** 4. **MemberReport class** 5. **ProviderReport class** 6. **SummaryReport class** |

# Collaboration Diagrams

Provider collaboration diagram:



Manager collaboration diagram:



# Sequence Diagrams (equivalent to collaboration diagrams)

Provider sequence diagram:



Manager sequence diagram:



# Black box test Cases

Test case 1:

|  |  |
| --- | --- |
| Component | LoginTerminal |
| Input | ID: 111111111  Password: FNKMKC |
| Output | Invalid credentials |

Test case 2:

|  |  |
| --- | --- |
| Component | ProviderTerminal |
| Input | Member ID: 111111111  Date of service: 2017-03-29  Service code: 123456  Comments: <blank> |
| Output | Fees due are 999 |

Test case 3:

|  |  |
| --- | --- |
| Component | ProviderTerminal |
| Input | <”Generate Provider Directory” button clicked> |
| Output | <Provider Directory> |

Test case 4:

|  |  |
| --- | --- |
| Component | ManagerTerminal |
| Input | Member ID: 111111111 |
| Output | <Member report> |

Test case 5:

|  |  |
| --- | --- |
| Component | ManagerTerminal |
| Input | Provider ID: 111111111 |
| Output | <Provider report> |

Test case 6:

|  |  |
| --- | --- |
| Component | SummaryTerminal |
| Input | <”Generate Summary Report” button clicked> |
| Output | <Summary report> |

# Screenshots

Database:

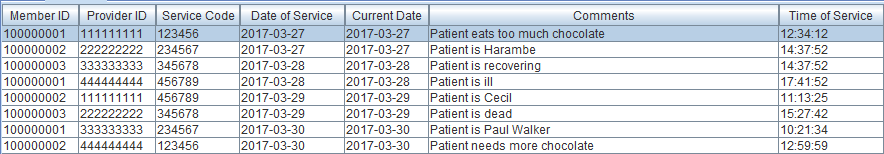


Figure 1 Appointment table

Screenshots/Database/Manager%20Database.png

Figure 2 Manager table

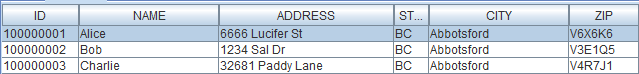


Figure 3 Member table



Figure 4 Provider table

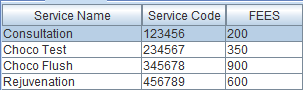
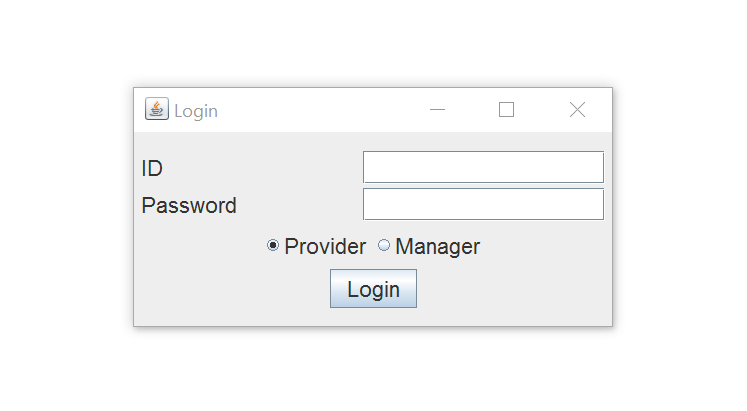
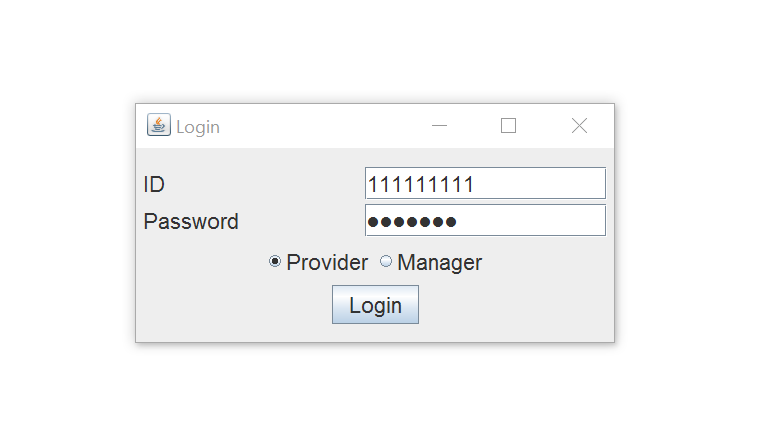
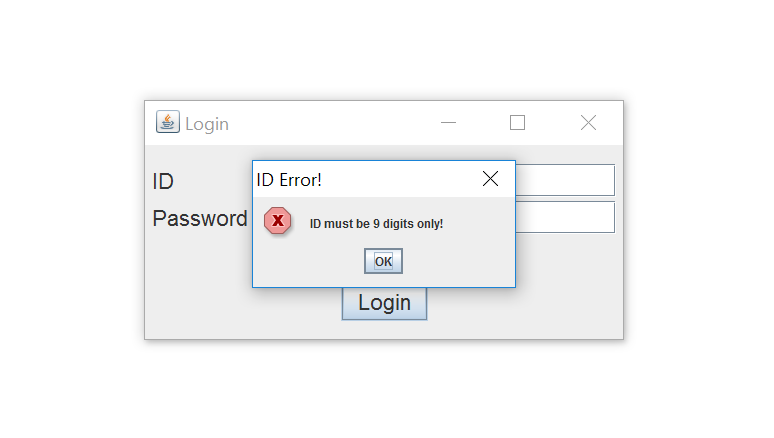


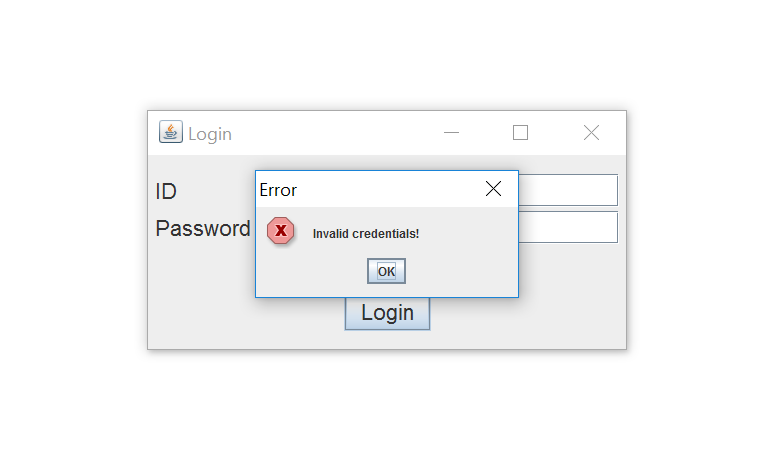
Figure 5 Provider Directory table

Login Terminal:

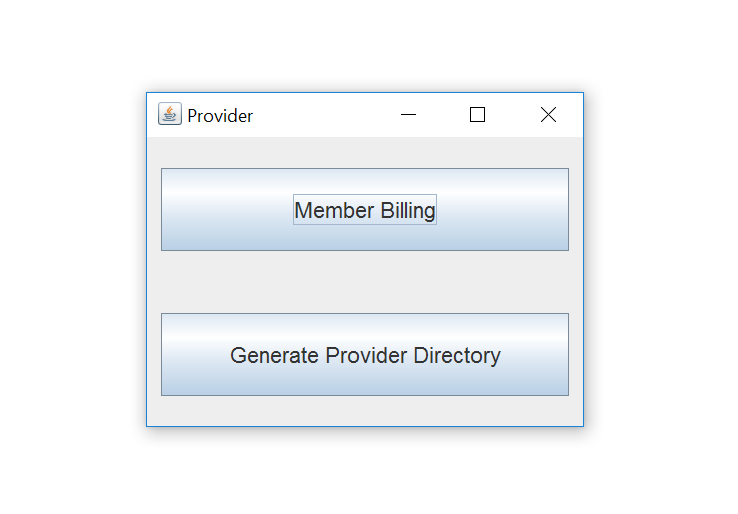


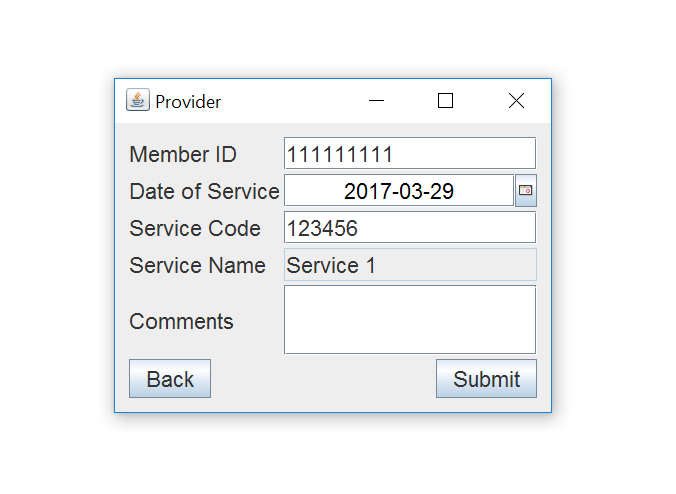


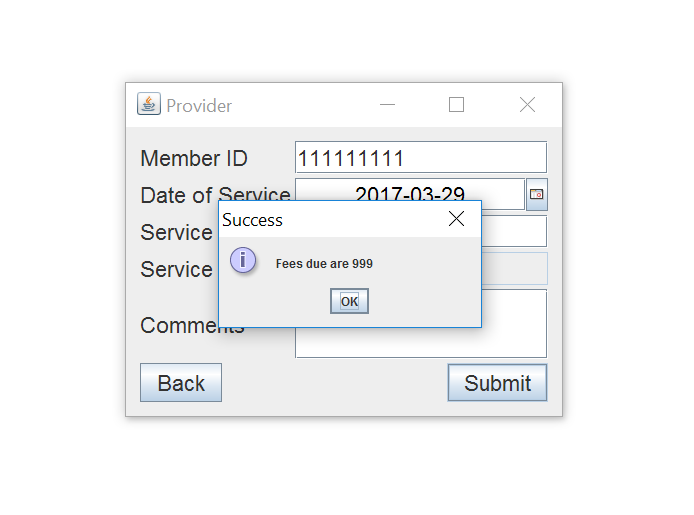




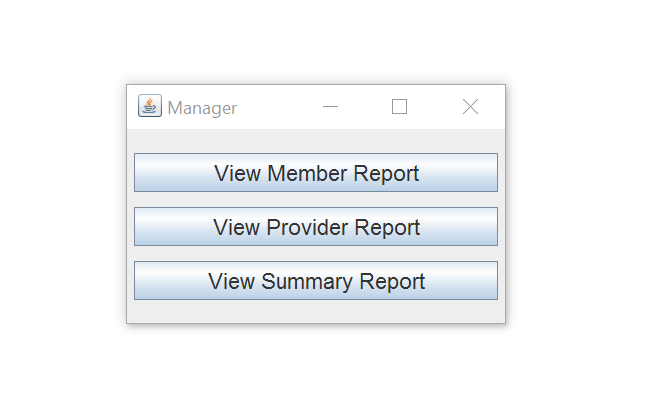
Provider Terminal:

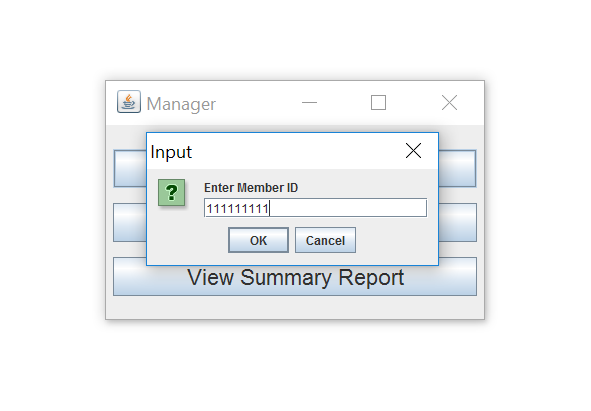


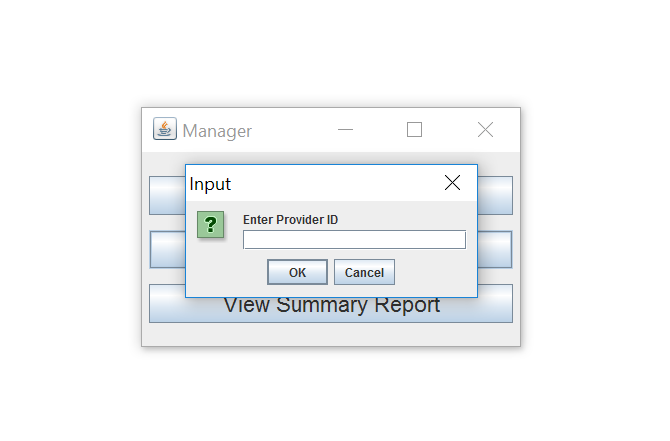




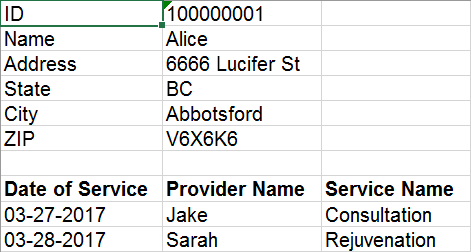
Manager Terminal:



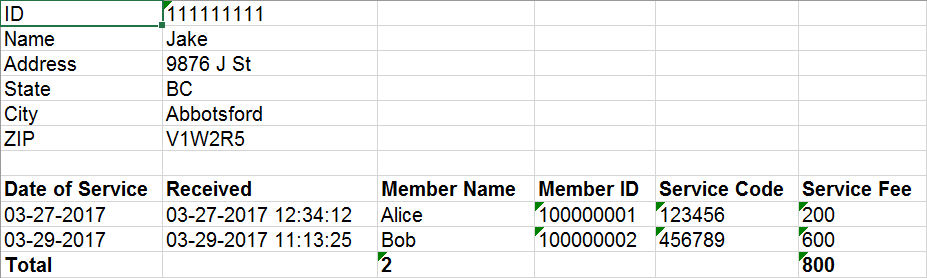




Member Report:



Provider Report:



Summary Report:

