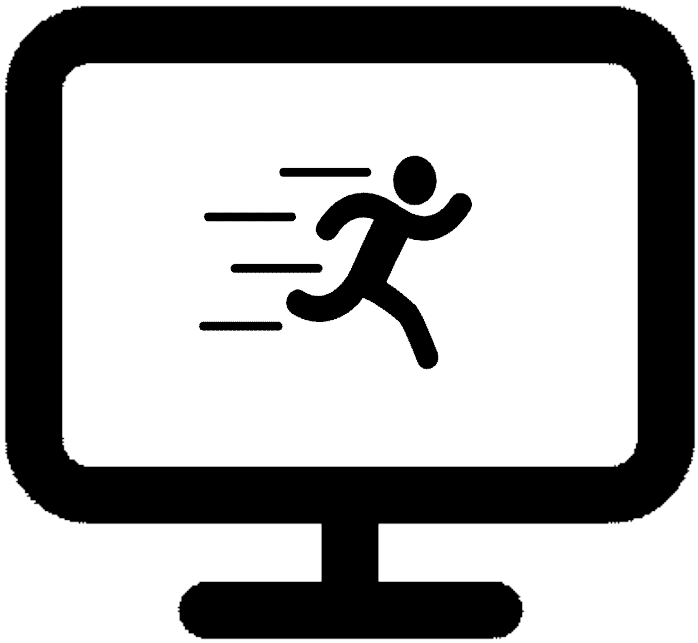
Yeungnam University

Customer Tracking



22311947
김서현
canna7610@gmail.com

[Revision history]

Revision date	Version #	Description	Author
28/05/2025	1.00	First Writing	김서현

= Contents =

1. Introduction

Customer Tracking System은 여러 가맹점의 고객 관리를 위해 설계된 고객 추적 시스템이다. 시스템의 사용자는 매니저와 점주로 나뉘며 1명의 매니저가 여러 명의 점주를 생성해 관리할 수 있다. 점주는 가게에 방문한 손님의 정보를 시스템에 입력하며, 시스템은 사용자의 요청에 따라 해당 정보들을 그래프의 형태로 제공한다. 시스템에서 제공된 정보는 매니저와 점주의 고객 관리를 도와 원활한 가게 운영을 가능하게 할 것이다.

해당 문서에는 Customer Tracking System의 설계를 위해 작성된 디자인 문서이며 여러 다이어그램들을 포함하고 있다. 작성된 주요 다이어그램으로는 Class Diagram, Sequence Diagram, State Machine Diagram이 있다. 각 다이어그램은 시스템의 구조와 동작 방식을 시각적으로 명확하게 표현하기 위해 작성되었다.

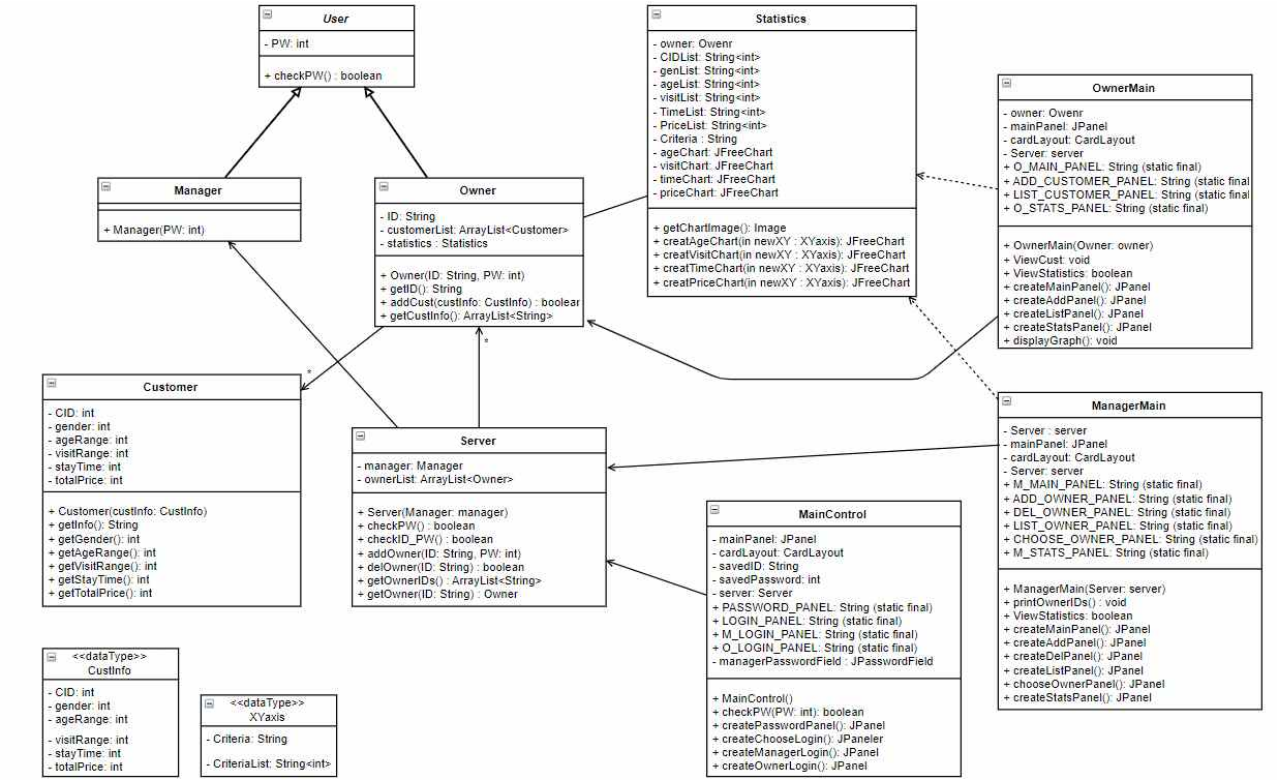
Class Diagram은 시스템에 사용될 각 클래스를 정의하고 클래스 간의 관계를 표현한다. 또한 각 클래스에 대한 자세한 설명은 다이어그램 아래의 Description 파트를 통해 작성되어 있으며 Description에는 클래스 요약, 주요 속성 및 메소드에 대한 설명이 존재한다.

Sequence Diagram은 Analysis 과정에서 도출된 각 Use Case를 기반으로 작성되었다. Sequence Diagram은 각 Use Case가 시간의 흐름에 따라 어떻게 동작하는지를 표현한다. 이를 통해 시스템 내 참여자 간의 상호작용과 호출 순서를 직관적으로 확인할 수 있다.

State Machine Diagram은 시스템의 상태 변화를 설명하기 위해 세 가지로 나누어 작성되었다. 각각의 State Machine Diagram은 로그인 이전, 매니저 로그인 이후, 점주 로그인 이후로 나뉜다. 세 다이어그램을 모두 작성한 이후, 다이어그램들을 하나의 다이어그램으로 통합하여 시스템 전체의 흐름을 확인할 수 있도록 하였다.

해당 Design 문서에서는 이와 같이 여러 다이어그램들을 통해 Customer Tracking System의 설계 구조와 작동 방식을 명확하게 설명하고자 한다.

2. Class diagram



해당 시스템은 명확한 구현과 편리한 유지보수를 위해 시스템의 데이터 처리와 UI 제어를 완전히 분리하는 형태로 설계되었다.

시스템을 이용하는 사용자 클래스는 User로, User는 Manager와 Owner의 공통 속성을 정의한다. 매니저와 점주를 포함한 모든 사용자는 Server 클래스에서 추가/삭제/호출이 이루어진다.

입력받은 고객의 정보를 저장하기 위해 Customer 클래스를 별도로 설계하였다. Owner는 Customer 클래스를 통해 고객 정보를 저장하며 Statistics 클래스를 통해 다양한 통계 정보를 제공받는다.

사용자 인터페이스를 위해 세 가지 클래스가 설계되었다. 로그인 이전까지의 흐름과 메인 화면을 관리하는 MainControl 클래스는 사용자의 로그인을 담당하며, 시스템에 로그인한 사용자에 따라 ManagerMain 또는 OwnerMain UI 클래스를 호출한다.

1) User

Description	
Summary	시스템을 이용하는 사용자들을 통합적으로 관리하는 추상 클래스. 인스턴스 생성은 불가능하다. Manager와 Owner로 상속된다.
Attributes	- PW: int : 사용자의 비밀번호
Methods	+ checkPW() : boolean : 사용자의 PW를 체크하기

2) Manager

Description	
Summary	User를 상속받은 매니저 클래스. 시스템에 존재하는 점주의 정보들을 확인할 수 있다.
Attributes	
Methods	+ Manager(PW: int) : 매니저 객체를 생성하는 생성자

3) Owner

Description	
Summary	User를 상속받은 점주 클래스. 점주의 고유 정보를 관리한다.
Attributes	- ID: String : 점주의 아이디 - customerList: ArrayList<Customer> : Shop Owner가 생성한 Customer의 리스트 - statistics : Statistics : 고객 통계를 위한 변수
Methods	+ Owner(ID: String, PW: int) : 점주 객체를 생성하는 생성자 + getID(): String : 점주의 아이디 가져오기 + addCust(custInfo: CustInfo) : boolean : 새로운 고객 추가하기 + getCustInfo(): ArrayList<String> : 자신이 저장한 고객들의 모든 고객정보를 문자열 리스트 가져오기

<<dataType>> CustInfo
- CID: int - gender: int - ageRange: int - visitRange: int - stayTime: int - totalPrice: int

4) Customer

Description	
Summary	고객의 식별 아이디, 성별, 나이대, 입장 시간대, 머문 시간대, 구매한 물건의 총 가격대 등의 정보를 저장하고 관리한다.
Attributes	- CID: int : 고객의 식별 아이디 - gender: int : 고객의 성별 - ageRange: int : 고객의 나이대 - visitRange: int : 고객의 입장 시간대 - stayTime: int : 고객이 가게에 머문 시간 - totalPrice: int : 고객이 구매한 물건의 총 가격대
Methods	+ Customer(custInfo: CustInfo) : 새 고객을 생성하기 위한 생성자 + getInfo(): String : 자신에게 저장된 모든 정보를 문자열로 반환 + getGender(): int : 고객의 성별 가져오기 + getAgeRange(): int : 고객의 나이대 가져오기 + getVisitRange(): int : 고객의 입장 시간대 가져오기 + getStayTime(): int : 고객이 가게에 머문 시간 가져오기 + getTotalPrice(): int : 고객이 구매한 물건의 총 가격대 가져오기

5) Statistics

Description	
Summary	등록된 고객 정보를 바탕으로 기준에 따라 통계 기능을 제공한다. 통계 기준 선택 기능과 통계 결과 조회 기능을 제공한다.
Attributes	- owner: Owenr : 통계 기능을 사용하고자 하는 점주를 담아두는 변수 - CIDList: String<int> : 해당 점주가 가진 모든 고객의 CID를 저장 - genList: String<int> : 해당 점주가 가진 모든 고객의 성별을 저장 - ageList: String<int> : 해당 점주가 가진 모든 고객의 나이대를 저장 - visitList: String<int> : 해당 점주가 가진 모든 고객의 입장 시간대를 저장 - TimeList: String<int> : 해당 점주가 가진 모든 고객이 가게에 머문 시간을 저장 - PriceList: String<int> : 해당 점주가 가진 모든 고객이 구매한 물건의 총 가격대를 저장 - Criteria : String : 통계 기준을 담는 변수 - ageChart: JFreeChart : Criteria와 나이를 기준으로 하는 차트 - visitChart: JFreeChart : Criteria와 입장 시간대를 기준으로 하는 차트 - timeChart: JFreeChart : Criteria와 머문 시간을 기준으로 하는 차트 - priceChart: JFreeChart : Criteria와 구매한 물건의 총 가격대를 기준으로 하는 차트
Methods	+ Statistics(owner: Owenr) : 클래스를 만드는 생성자 + getChartImage(): Image : 생성된 그래프를 이미지로 가져오기 + creatAgeChart(newXY : XYaxis): JFreeChart : Criteria와 나이를 각 축으로 삼는 JFreeChart를 생성하기 + creatVisitChart(newXY : XYaxis): JFreeChart : Criteria와 입장 시간대를 각 축으로 삼는 JFreeChart를 생성하기 + creatTimeChart(newXY : XYaxis): JFreeChart : Criteria와 머문 시간을 각 축으로 삼는 JFreeChart를 생성하기 + creatPriceChart(newXY : XYaxis): JFreeChart : Criteria와 구매한 물건의 총 가격대를 각 축으로 삼는 JFreeChart를 생성하기

<<dataType>> XYaxis
- Criteria: String - CriteriaList: String<int>

6) MainControl

Description	
Summary	프로그램 최초 실행 시 Set PW → 로그인 → 매니저/점주 메인 화면으로 흐름을 연결해 주는 메인 컨트롤러 클래스.
Attributes	- mainPanel: JPanel : 화면 전환을 위한 메인 패널 - cardLayout: CardLayout : 패널 전환을 위한 레이아웃 매니저 - savedID: String : 로그인 화면에서 입력받은 아이디를 저장하는 변수 - savedPassword: int : 로그인 화면에서 입력받은 비밀번호를 저장하는 변수 - server: Server : 시스템에 저장된 사용자들을 담고 있는 변수 + PASSWORD_PANEL: String (static final) : 매니저 패스워드 설정 패널 식별자 + LOGIN_PANEL: String (static final) : 로그인 유형 선택 패널 식별자 + M_LOGIN_PANEL: String (static final) : Manager 로그인 화면 패널 식별자 + O_LOGIN_PANEL: String (static final) : Shop Owner 로그인 화면 패널 식별자 - managerPasswordField : JPasswordField : 매니저 비밀번호 입력 필드
Methods	+ MainControl(): 프로그램 최초 실행 시 매니저/점주 메인 화면까지의 UI를 생성하는 생성자 + checkPW(PW: int): boolean : 비밀번호가 유효한지 체크 + createPasswordPanel(): JPanel : 매니저의 비밀번호를 초기화하는 비밀번호 설정 화면 생성하기 + createChooseLogin(): JPanel : Manager 로그인과 Show Owner 로그인 중 하나를 선택하는 로그인 유형 선택 화면 생성하기 + createManagerLogin(): JPanel : 비밀번호를 입력받는 Manager 로그인 화면 생성하기 + createOwnerLogin(): JPanel : 아이디와 비밀번호를 입력받는 Show Owner 로그인 화면 생성하기

7) ManagerMain

Description	
Summary	매니저 메인 화면을 관리한다. 매니저 로그인 이후 매니저의 메인 화면을 보여주며 점주 관리, 통계 보기, 로그아웃 등의 기능을 수행하는 컨트롤러 클래스이다. 메뉴를 통해 다양한 패널로 화면 전환이 가능하며, 각 패널은 점주 추가, 삭제, 리스트 보기, 통계 확인 등을 담당한다.
Attributes	- Server : server : 현재 시스템에서 사용 중인 서버 - mainPanel: JPanel : 화면 전환을 위한 메인 패널 - cardLayout: CardLayout : 패널 전환을 위한 레이아웃 매니저 - Server: server + M_MAIN_PANEL: String (static final) : 매니저 메인 화면 패널 식별자 + ADD_OWNER_PANEL: String (static final) : 점주 추가 패널 식별자 + DEL_OWNER_PANEL: String (static final) : 점주 삭제 패널 식별자 + LIST_OWNER_PANEL: String (static final) : 점주 리스트 패널 식별자 + CHOOSE_OWNER_PANEL: String (static final) : 통계를 확인할 점주를 선택하는 패널의 패널 식별자 + M_STATS_PANEL: String (static final) : 점주별 통계 보기 패널 식별자
Methods	+ ManagerMain(Server: server): ManagerMain 클래스의 생성자. Manager로 시스템에 로그인 시 매니저 메인 화면의 UI를 구성하고 메뉴와 패널을 초기화 + printOwnerIDs() : void : 매니저가 소유한 점주들의 ID 출력 + ViewStatistics: boolean : 고객 통계 보기 + createMainPanel(): JPanel : 매니저의 기본 패널을 생성하기 + createAddPanel(): JPanel : 점주를 추가하는 패널을 생성하기 + createDelPanel(): JPanel : 점주를 삭제하는 패널을 생성하기 + createListPanel(): JPanel : 등록된 점주 리스트를 보여주는 패널을 생성하기 + chooseOwnerPanel(): JPanel : 통계 대상 점주를 선택하는 패널을 생성하기 + createStatsPanel(): JPanel : 통계를 보기 위한 기준을 선택하고 선택된 점주의 통계 정보를 보여주는 패널을 생성하기

8) OwnerMain

Description	
Summary	점주 메인 화면을 관리한다. 점주 로그인 이후 점주의 메인 화면을 보여주며 고객 관리, 통계 보기, 로그아웃 등의 기능을 수행하는 컨트롤러 클래스이다. 메뉴를 통해 다양한 패널로 화면 전환이 가능하며, 각 패널은 고객 추가, 고객 리스트 보기, 통계 확인 등을 담당한다.
Attributes	- owner: Owenr : 현재 시스템에 로그인한 점주 - mainPanel: JPanel : 화면 전환을 위한 메인 패널 - cardLayout: CardLayout : 패널 전환을 위한 레이아웃 매니저 - Server: server + O_MAIN_PANEL: String (static final) : 점주 메인 화면 패널 식별자 + ADD_CUSTOMER_PANEL: String (static final) : 고객 추가 패널 식별자 + LIST_CUSTOMER_PANEL: String (static final) : 고객 리스트 패널 식별자 + O_STATS_PANEL: String (static final) : 통계 보기 패널 식별자
Methods	+ OwnerMain(Owner: owner): OwnerMain 클래스의 생성자. Owner로 시스템에 로그인 시 점주 메인 화면의 UI를 구성하고 메뉴와 패널을 초기화 + ViewCust: void : 고객 정보 보기 + ViewStatistics: boolean : 고객 통계 보기 + createMainPanel(): JPanel : 점주의 기본 패널을 생성하기 + createAddPanel(): JPanel : 고객 정보를 입력해 새 고객을 추가하는 패널을 생성하기 + createListPanel(): JPanel : 등록된 고객들의 세부 정보 리스트를 보여주는 패널을 생성하기 + createStatsPanel(): JPanel : 통계를 보기 위한 기준을 선택하고 해당 기준에 따라 통계 정보를 보여주는 패널을 생성하기

9) Server

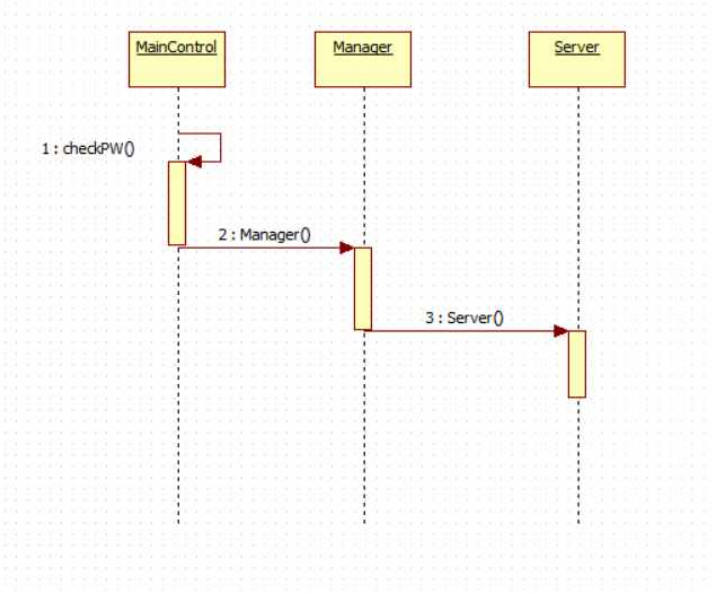
Description	
Summary	시스템의 모든 유저 데이터를 저장하고 관리하는 데이터 저장소. 별도의 서버가 존재하지 않는 해당 시스템에서 서버의 역할을 수행한다. 시스템에 존재하는 모든 유저의 데이터를 저장하고 관리한다.
Attributes	- manager: Manager : 해당 시스템의 매니저를 저장 해두는 변수 - ownerList: ArrayList<Owner> : 해당 시스템의 Owner들을 저장해두는 리스트
Methods	+ Server(Manager: manager): Server 클래스의 생성자 + checkPW() : boolean : 매니저의 비밀번호 체크하기 + checkID_PW() : boolean : 점주의 아이디/비밀번호 체크하기 + addOwner(ID: String, PW: int) : boolean : 새로운 점주 추가 + delOwner(ID: String) : boolean : 기존 점주 삭제 + getOwnerIDs() : ArrayList<String> : 내가 생성한 점주들의 아이디만을 저장한 아이디 리스트 가져오기 + getOwner(ID: String) : Owner : 해당 ID를 가진 점주 인스턴스 가져오기

3. Sequence diagram

이 시스템은 UI를 구현하기 위해 MainControl, ManagerMain, OwnerMain의 세 가지 클래스를 사용한다. 각 클래스들은 JPanel을 통해 UI를 구현하고, 패널을 생성하기 위해 createPasswordPanel(), createChooseLogin() 등의 여러 내부적인 함수들을 사용하였다. 해당 함수들은 MainControl, ManagerMain, OwnerMain의 생성자 내부에서 모두 사용되지만 가독성을 위해 아래 Sequence diagram에서는 표기하지 않았다. 각 생성자의 내부에서 사용되는 함수는 아래와 같다.

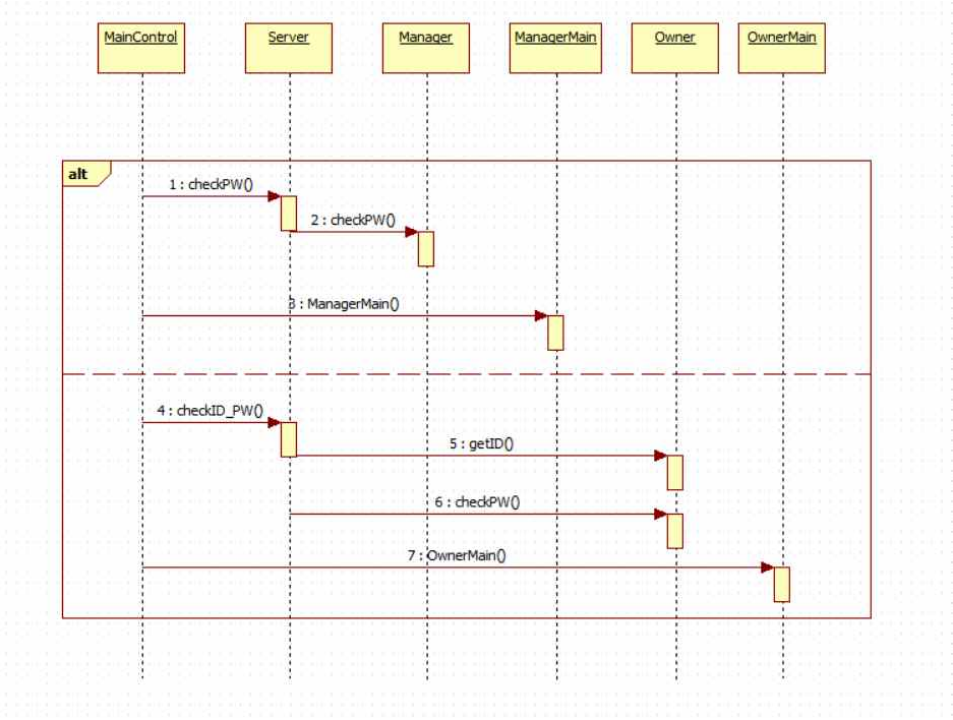
Class Constructor	Used Methods
MainControl()	+ createPasswordPanel(): JPanel + createChooseLogin(): JPanel + createManagerLogin(): JPanel + createOwnerLogin(): JPanel
ManagerMain()	+ createMainPanel(): JPanel + createAddPanel(): JPanel + createDelPanel(): JPanel + createListPanel(): JPanel + chooseOwnerPanel(): JPanel + createStatsPanel(): JPanel
OwnerMain()	+ createMainPanel(): JPanel + createAddPanel(): JPanel + createListPanel(): JPanel + createStatsPanel(): JPanel

1) Set PW



Set PW는 시스템 매니저의 비밀번호를 초기화하기 위한 기능이다. 메인 화면을 관리하는 MainControl 클래스는 사용자에게서 입력받은 비밀번호가 유효한지를 확인한다. 만약 유효하다면 해당 비밀번호를 가진 Manager를 인스턴스를 만든다. 또한 Server를 생성해 만들어진 Manager를 Server에 저장한다.

2) Login

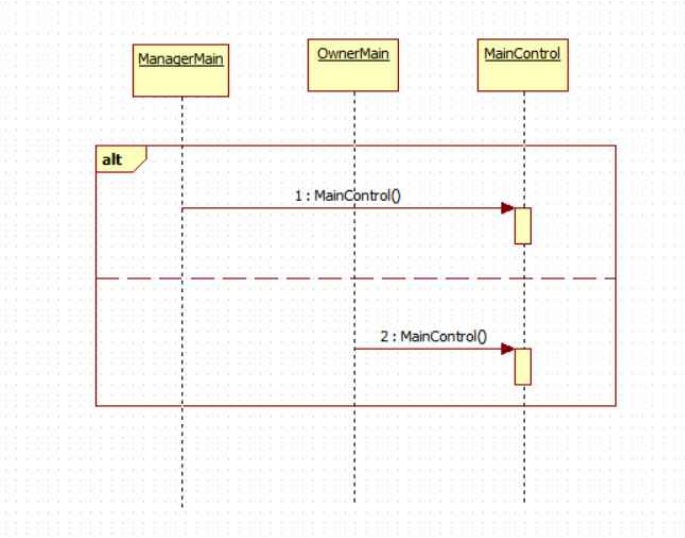


Login은 매니저 또는 점주가 시스템에 로그인하기 위한 기능이다. MainControl 클래스에서 로그인 시, 누구로 로그인하는 것을 선택하느냐에 따라 Login의 흐름이 두 가지로 나뉜다.

사용자가 선택한 유형이 매니저 로그인이라면 Server 인스턴스에 저장된 Manager의 비밀번호와 사용자가 입력한 비밀번호가 일치하는지를 확인한다. 확인 절차가 무사히 끝난다면 매니저 메인 화면을 클래스인 ManagerMain을 생성한다.

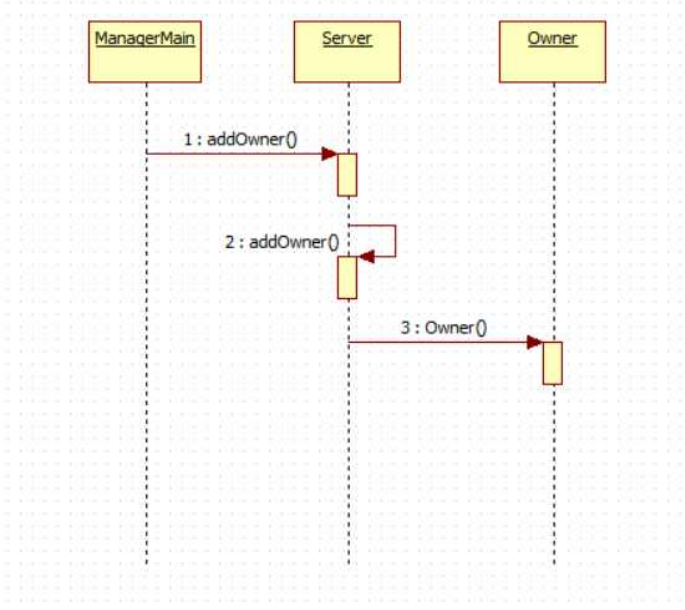
사용자가 선택한 유형이 점주 로그인이라면 Server에 저장된 모든 Owner를 돌며 해당 Owner에 저장된 아이디/비밀번호가 사용자가 입력한 아이디/비밀번호와 일치하는지를 확인한다. 만약 일치하는 Owner가 존재한다면 해당 Owner를 로그인 점주로 가진 OwnerMain을 생성한다.

3) Logout



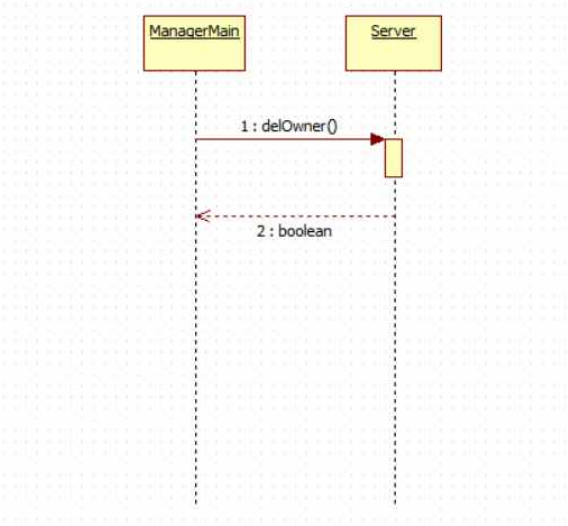
Logout은 현재 로그인 상태에서 로그아웃하고 처음의 로그인 창으로 되돌아가기 위한 기능이다. 현재 사용자가 사용 중인 사용자 메인 화면 클래스에서 (사용자가 매니저일 경우 ManagerMain 클래스, 사용자가 점주일 경우 OwnerMain 클래스) 로그인 흐름을 관리하는 MainControl을 다시 생성한다.

4) Add Owner



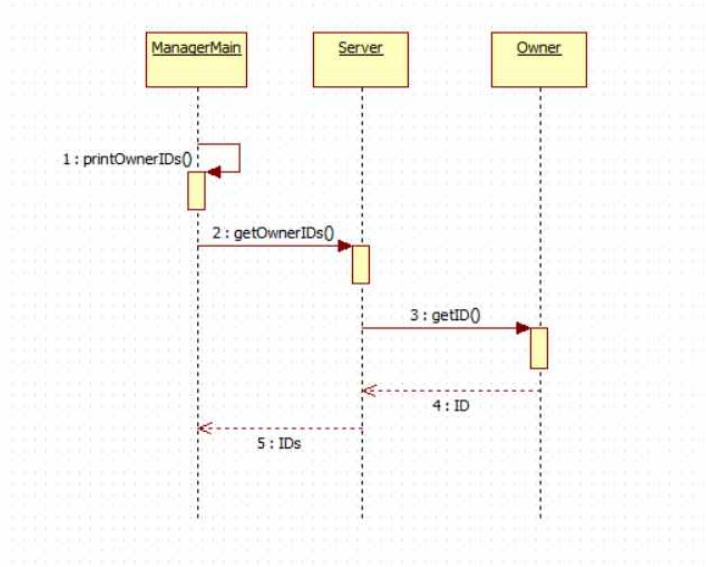
Add Owner는 매니저가 새로운 점주를 추가하기 위해 사용하는 기능이다. ManagerMain 클래스는 Server의 addOwner() 함수를 실행한다. Server는 addOwner를 통해 ManagerMain에게서 넘겨받은 아이디와 비밀번호가 유효한지를 판단한다. 만약 유효하다면 Owner 객체를 생성한다.

5) Delete Owner



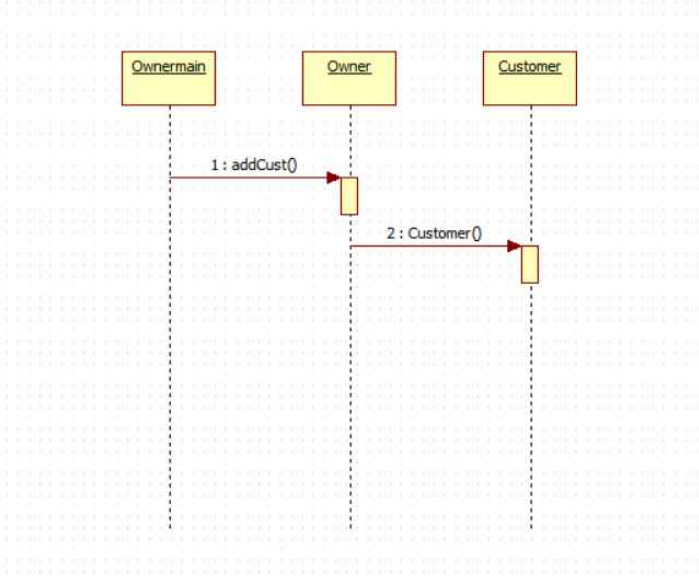
Delete Owner는 기존의 점주를 삭제하기 위한 기능이다. ManagerMain은 Server의 delOwner() 함수를 실행한다. Server에 삭제하고자 하는 Owner가 존재한다면 삭제한다. 삭제에 성공했다면 True, 실패했다면 False를 반환한다.

6) Owner list



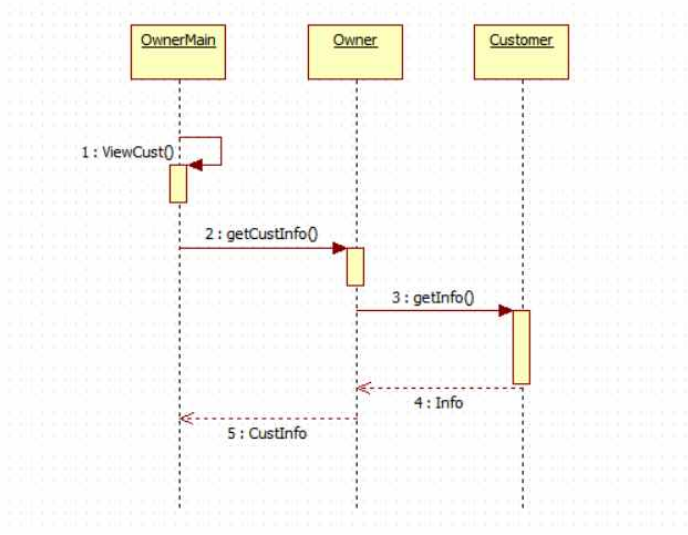
Owner list는 서버에 존재하는 모든 점주의 ID를 출력하는 기능이다. ManagerMain은 printOwnerIDs()를 통해 점주의 아이디를 출력하게 된다. ManagerMain은 Server에게 모든 점주의 ID를 요청하고 Server는 자신이 소유한 각 Owner들에게 ID를 요청한다. ManagerMain은 Server가 반환한 모든 ID를 자신의 화면에 출력한다.

7) Input customer



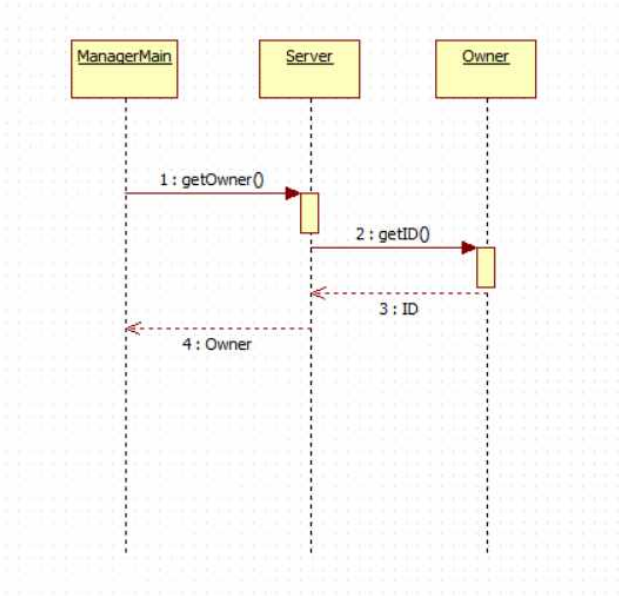
Input customer는 점주가 새로운 고객을 추가하기 위해 사용하는 기능이다. 점주 메인 화면인 OwnerMain 클래스는 Owner의 addCust() 함수를 실행한다. Owner는 ManagerMain에게서 전달받은 정보를 바탕으로 Customer 인스턴스를 생성한다.

8) View customer



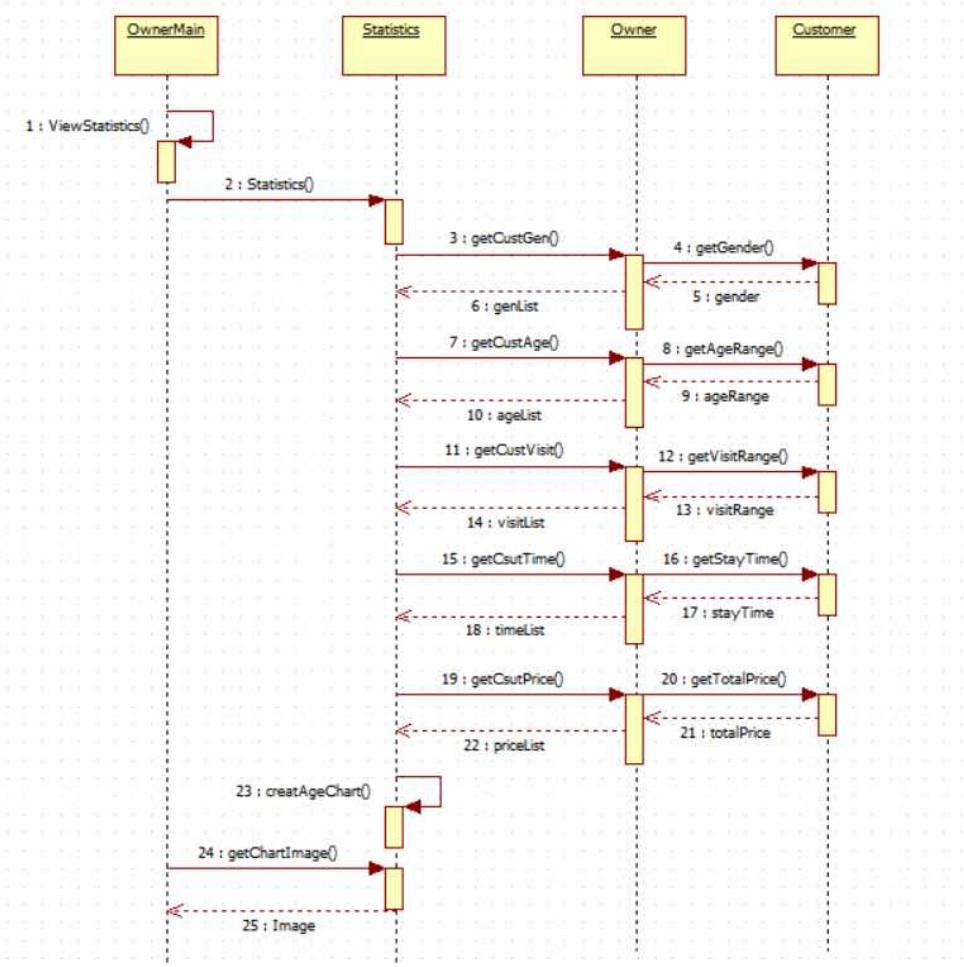
View customer는 점주가 자신이 소유한 고객들의 정보를 보기 위해 사용된다. OwnerMain이 ViewCust() 함수를 실행하면 Owner는 getCustInfo() 함수를 통해 자신이 소유한 Customer 인스턴스들에게 정보를 요청한다. Customer가 반환한 정보들을 Owner가 정리해 다시 OwnerMain에게 전달한다. OwnerMain은 최종적으로 전달받은 고객들의 정보를 자신의 화면에 출력한다.

9) Choose owner



Choose owner는 매니저가 통계를 보고 싶은 점주를 선택하기 위해 사용된다. ManagerMain은 입력받은 ID와 함께 Server의 getOwner() 함수를 실행한다. Server는 자신이 소유한 Owner들에게 ID를 요청하고 Owner에게서 전달받은 ID와 ManagerMain에게서 입력받은 ID를 비교해 일치하는 Owner를 리턴한다.

10) View statistics



View statistics는 매니저 혹은 점주가 등록된 고객들의 정보를 통계로써 확인하고 싶을 때 사용된다. 매니저의 경우, Use case #9의 점주 선택 단계인 Choose owner 단계가 먼저 일어난다.

OwnerMain이 ViewStatistics() 함수를 실행하면 Statistics 인스턴스가 생성된다. 해당 인스턴스에서는 선택된 Owner를 통해 해당 Owner가 소유한 Customer들의 정보(성별, 나이, 방문 시간, 머문 시간, 구매 가격대)를 가져온다.

이후 creatAgeChart() 함수에서 가져온 Customer들의 정보를 통해 나이를 기준으로 하는 통계 그래프를 생성한다. 만들어진 그래프들은 getChartImage() 함수를 통해 이미지의 형태로 OwnerMain 클래스에 리턴된다.

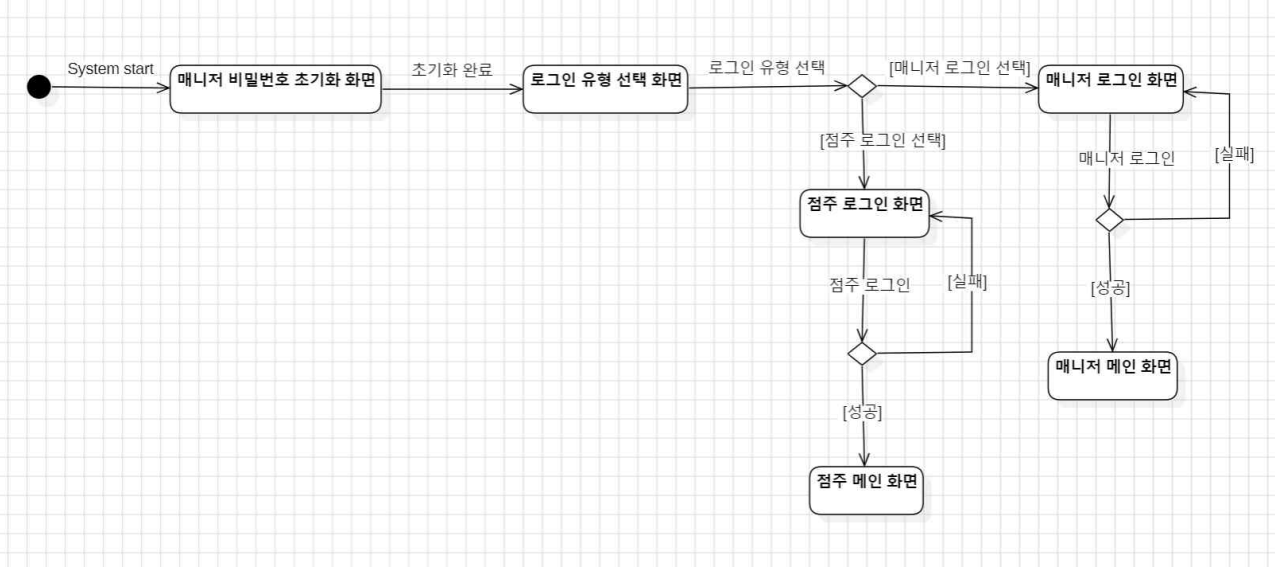
만약 이 Use case를 사용하는 객체가 Ownermain이 아니라 ManagerMain이더라도 이 모든 과정은 동일하게 수행된다. 또한 선택된 통계 기준이 나이일 때만 createAgeChart() 함수가 실행된다. 통계 기준이 달라지면 creatVisitChart(), creatTimeChart(), creatPriceChart()가 createAgeChart()의 자리에서 실행된다.

4. State machine diaram

해당 시스템의 흐름은 로그인 이전, 매니저 로그인 이후, 점주 로그인 이후의 세 가지로 구분된다. 각 경우에 따라 로그인 진행 화면, 매니저 메인 화면, 점주 메인 화면에서의 동작이 대부분 분리되어 이루어진다. 해당 챕터에서는 명확한 설명을 위해 이 세 가지 경우를 나누어, 각각에 대한 State machine diaram을 작성하였다.

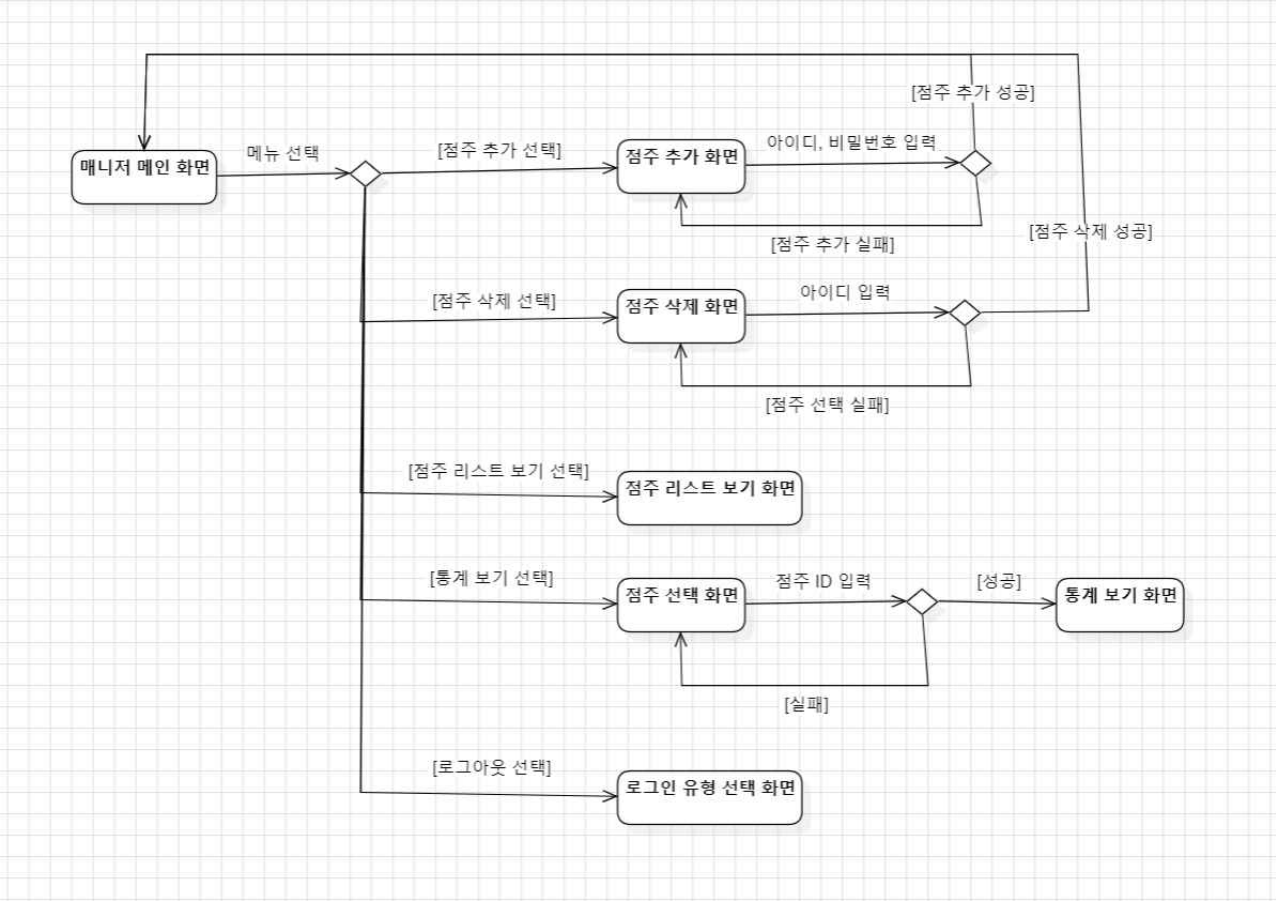
작성된 다이어그램들은 화면 간 전환 관계를 중심으로 표현했기에 각 화면 내에서의 세부적인 동작 방식을 모두 알 수는 없다. 만약 한 State 내에서의 구체적인 동작 방식을 알고 싶다면 Use case diagram이나 Sequence diagram 등 보고서 내 다른 다이어그램을 참고해 도움을 받을 수 있다.

4.1. 로그인 진행 화면



위의 그림은 로그인 이전까지의 단계를 표현한 State machine diaram이다. 고객 관리 시스템이 최초로 시작되면 사용자는 매니저 비밀번호 초기화 화면을 마주하게 된다. 사용자는 로그인 진행 화면의 단계들을 모두 밟은 후 점주 메인 화면 또는 매니저 메인 화면으로 이동한다.

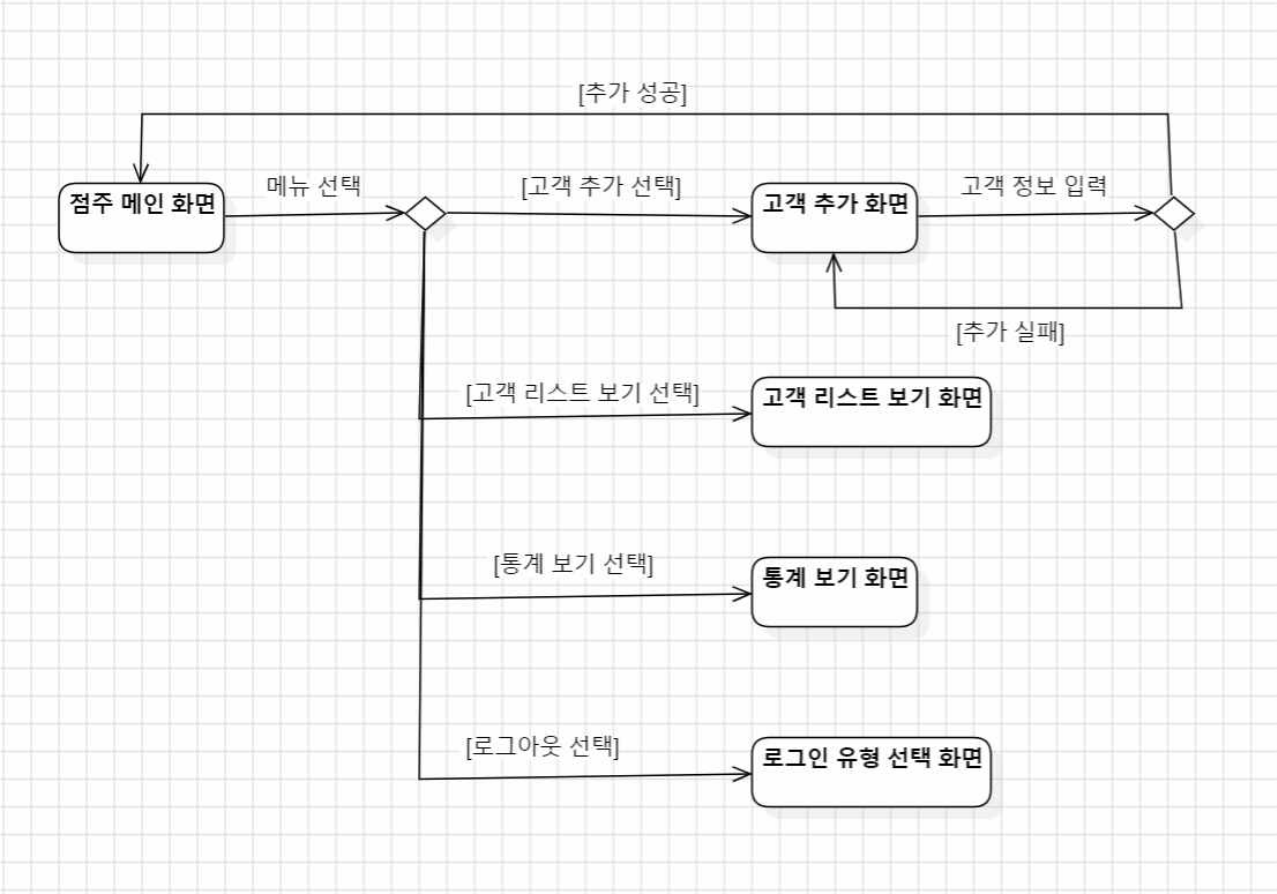
4.2. 매니저 메인 화면



로그인 이후 매니저 메인 화면에서는 모든 단계에 걸쳐 메뉴바가 항상 표시된다. 따라서 '메뉴 선택' 이벤트는 사용자의 현재 위치와 관계 없이 언제든 실행 가능하다. 다만 로그인 직후에는 시스템 기능을 사용하기 위해 반드시 메뉴 선택 이벤트를 수행해야 하므로, 해당 이벤트를 매니저 메인 화면과 연결해 두었다.

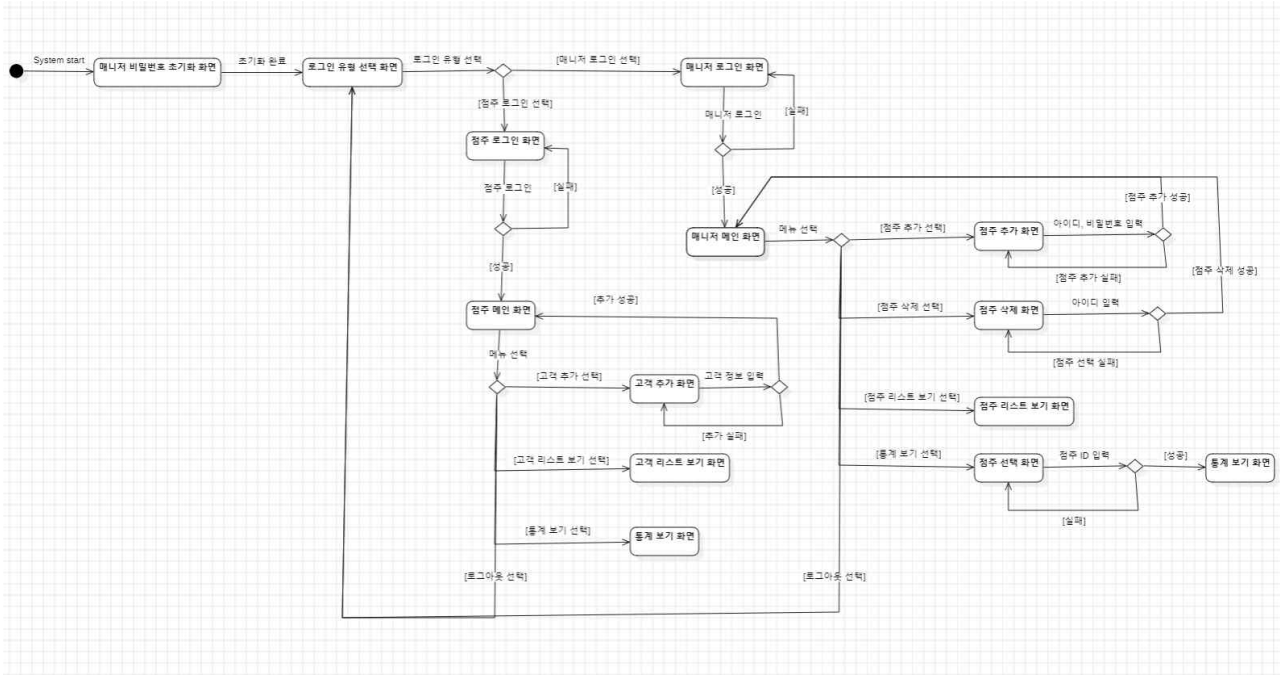
또한 '로그인 유형 선택 화면'은 앞서 설명한 4.1의 로그인 유형 선택 화면과 동일한 화면을 의미한다. 이에 대한 정확한 연결 관계는 4.4의 통합 다이어그램에서 확인할 수 있다.

4.3. 점주 메인 화면



점주 메인 화면 역시 매니저 메인 화면과 동일하게 모든 단계에 걸쳐 메뉴바가 항상 표시된다. 사용자는 해당 메뉴바를 통해 언제든 메뉴 선택 이벤트를 실행할 수 있다.

4.4 통합 다이어그램



로그인 진행 화면, 매니저 메인 화면, 점주 메인 화면을 표현한 세 개의 다이어그램을 통합한 State machine diaram이다. 세 다이어그램을 통합하며 매니저 메인 화면과 점주 메인 화면에서 발생하는 '로그아웃 선택' 이벤트가 로그인 진행 화면의 '로그인 유형 선택 화면'과 연결되어 있음을 확인할 수 있다.

덧붙여 다이어그램을 살펴보면 알 수 있듯이 해당 시스템 내에는 별도의 종료 단계가 존재하지 않는다. 사용자는 모든 단계에서 언제든 X 버튼을 눌러 원하는 시점에 시스템을 종료할 수 있다.

5. Implementation requirements

1) 해당 프로그램은 Java 23 버전을 기준으로 개발된다.

2) GUI 화면 구성은 Swing 라이브러리를 사용하여 구현한다.

3) 고객 통계 데이터는 JFreeChart 라이브러리를 이용해 그래프 형태로 시각화한다.

6. Glossary

Term	Description
Manager	여러 명의 Shop owner를 관리하는 관리자. 프랜차이즈 매장의 경우, 본사에 해당한다.
Shop owner	자신의 가게 안에서 물건을 판매하고 방문한 손님에 대한 정보를 기록한다. 모든 Shop owner는 Manager의 아래에 존재한다.
Customer	Shop owner의 가게에 와 물건을 구매하는 구매자.
JDK	JDK(Java Development Kit)는 Java 프로그램을 개발하기 위한 소프트웨어 개발 환경이다. Java 프로그램을 작성, 컴파일, 실행하고 디버깅하는 데 필요한 도구, 실행 환경, 라이브러리 및 API를 포함하고 있다. 이 프로젝트에서는 많은 버전 중 JDK 21을 사용한다.
Swing	Java에서 GUI(그래픽 사용자 인터페이스)를 개발하기 위해 사용되는 표준 라이브러리이다. AWT(Abstract Window Toolkit)를 기반으로 하되 더 많은 기능과 유연성을 제공한다.
JFreeChart	Java 애플리케이션에서 전문적인 품질의 차트를 쉽게 생성할 수 있게 해주는 오픈 소스 차트 라이브러리이다. 다양한 종류의 차트(막대, 파이, 선, 산점도 등)를 지원하며 Swing과의 통합이 쉽다는 장점이 있다.

7. References

1) Icons and Images

https://www.flaticon.com/

2) Swing

https://docs.oracle.com/en/java/javase/17/docs/api/java.desktop/javax/swing/package-summary.html