Frisbee Management

쾌적한 매장 관리를 위한 프로그램

**이주연**

목차

[1. 과제 개요 2](#_Toc58595967)

[1. 1 프로젝트 개요 2](#_Toc58595968)

[2. 프로젝트 배경 기술 3](#_Toc58595969)

[2. 1 배경 3](#_Toc58595970)

[2. 2 목적 및 필요성 3](#_Toc58595971)

[3. 요구사항 4](#_Toc58595972)

[3. 1 기능적 요구사항 4](#_Toc58595973)

[3. 2 비기능적 요구사항 4](#_Toc58595974)

[4. 과제 세부 내용 5](#_Toc58595975)

[4. 1 Use Case 5](#_Toc58595976)

[4. 2 Sequence Diagram 5](#_Toc58595977)

[4. 3 Class Diagram 7](#_Toc58595978)

[4. 4 ER Digaram 9](#_Toc58595979)

[5. 기대 효과 11](#_Toc58595980)

[5. 1 원활한 매장 관리 11](#_Toc58595981)

[5. 2 한 프로그램으로 모든 직원이 접속 11](#_Toc58595982)

[5. 3 매장 매출의 상승 11](#_Toc58595983)

[6. 기타 12](#_Toc58595984)

[6. 1 한계점 및 개선 사항 12](#_Toc58595985)

[6. 2 개발 환경 12](#_Toc58595986)

[7. INDEX 13](#_Toc58595987)

[Figure 13](#_Toc58595988)

[Table 13](#_Toc58595989)

[Reference 13](#_Toc58595990)

# 과제 개요

## 1 프로젝트 개요

|  |  |
| --- | --- |
|  | 한 매장을 관리하는 기본 기능을 추려 프로그램을 작성. |

매장 관리 프로그램은 한 매장을 관리하기 위해서라도 꼭 필요한 프로그램이다.

한 매장을 쾌적하게 관리하기 위해서는 일일이 수기로 작성하기보다는 데이터베이스의 힘을 빌려 방대한 양의 데이터를 손쉽게 정리하는 것이 편하다. 또한 데이터베이스를 직접 보는 것보다는 원하는 정보만 취합하여 꼭 필요한 정보만을 보는 것이 사용자에게 편리하다.

이에 본 프로젝트에서는 매장을 원활하게 관리할 수 있는 프로그램을 제안한다. 본 프로젝트는 가장 기본적인 기능을 구현하여 매장 관리를 손쉽게 하도록 추구한다.

프로젝트에서 요구되는 기능은 재고 관리, 매출 관리 즉, 매장의 월별 판매, 물품별 판매 내역이며 직원 추가나, 직원 근태 관리 등의 직원 관리이다. 또한 각 매장간의 의사소통 기능을 제공하여 재고 부족시 해당 물품의 재고가 있는 가장 가까운 거리의 매장을 보여 줘 그 매장에게 재고를 요청할 수 있게 한다.

# 프로젝트 배경 기술

## 1 배경

|  |  |
| --- | --- |
|  | 기계적인 작업을 자동화. |

매장 관리 프로그램은 어느 매장이든 필수적으로 요구되는 프로그램이다. 그러한 프로그램은 직원이 일일이 해야하는 작업을 최대한 자동화 시켜 준다.

본 프로젝트를 통해 매장 관리 중에 발생할 수 있는 귀찮은 반복 업무를 줄여주며 관리자나 직원이 보고 싶은 정보만을 추려 보이도록 한다. 이러한 프로그램을 하나의 전자기기 판매 매장을 기초로 하여 무료로 만들고 제공하는 것이 이 프로젝트의 목표이다.

## 2. 2 목적 및 필요성

|  |  |
| --- | --- |
|  | * 직원 관리 * 재고 관리 * 매출 관리 |

최종적으로 개발하고자 하는 프로그램은 각 매장의 관리자 및 직원이 더 수월하게 매장을 관리할 수 있게 하는 기능을 제공한다.

기존에도 매장 관리 프로그램은 이미 수도 없이 존재하지만 비싼 프로그램을 구매할 여력이 없는 사람들을 위해 아주 기본적인 기능을 제공하는 프로그램을 제작하였다. 개발 하고자 하는 프로그램은 이러한 문제점 또한 해결 해 줄 수 있을 것으로 보인다.

# 요구사항

## 1 기능적 요구사항

|  |  |
| --- | --- |
| 기능 | 요구사항 |
| 로그인 | 유저는 로그인을 통해 해당 프로그램을 사용 가능 |
| 회원가입 | 유저는 로그인 전 회원가입을 해야만 하며 직원으로 등록된 경우만 가능 |
| 공지 확인 | 데이터베이스에 등록되어있는 공지를 확인할 수 있음 |
| 공지 등록 | 권한이 있는 관리자만 데이터베이스에 공지 등록 가능 |
| 재고 확인 | 재고가 있는지 확인 가능 |
| 재고 정보 | 재고의 정확한 정보 확인 가능 |
| 매출 확인 | 연도별, 월별의 매출 현황 파악 가능 |
| 주문 정보 | 주문을 해야할 재고의 정보와 주문 요청 가능 |
| 직원 목록 | 해당 매장에서 근무하는 직원의 정보 확인 가능 |
| 본인 근태 확인 | 본인이 일한 근무 내용 확인 가능 |
| 본인 근태 등록 | 본인이 일한 근무 시간 등록 가능 |
| 직원 근태 확인 | 관리자의 권한으로 해당 매장의 직원 근태 확인 가능 |
| 직원 근태 등록 | 관리자의 권한으로 해당 매장의 직원 근태 추가 가능 |
| 직원 등록 | 관리자의 권한으로 직원 등록 가능 |

## 2 비기능적 요구사항

|  |  |
| --- | --- |
| 기능 | 요구사항 |
| 직급 구분 | - 직원과 관리자의 구분이 존재  - 직원의 코드는 모두 AAA-11111의 형식을 따름 |
| 아이디 규칙 | 아이디는 겹치지 않아야 함 |
| 직관적인 GUI | 사용자가 직관적으로 이해할 수 있는 인터페이스를 요구 |

# 4. 과제 세부 내용

## 4. 1 Use Case

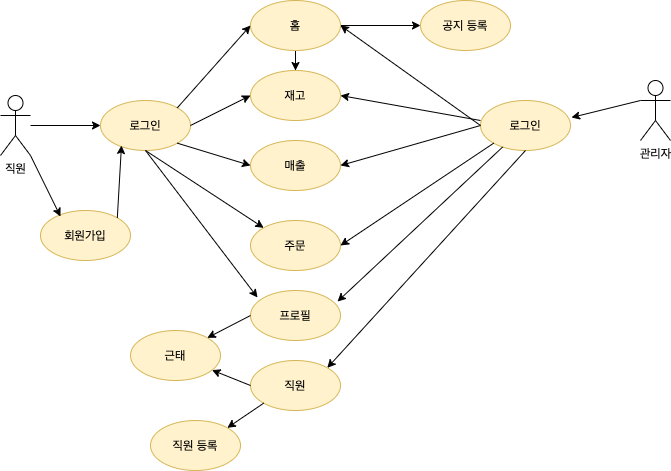


표 로그인 유스케이스

|  |  |
| --- | --- |
| 유스케이스명 | 로그인 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 관리자와 사용자가 프로그램에 접속하기 위해 로그인을 한다. |
| 사전 조건 | - 사용자는 이미 회원가입이 되어있어야 한다  - 사용자는 이미 직원으로 등록되어있어야 한다. |
| 사후 조건 | - 없음 |
| 기본 흐름 | 1. 사용자는 아이디와 비밀번호를 입력한다.  2. 시스템은 아이디와 비밀번호로 사용자의 존재 여부를 파악한다.  3. 존재하며 비밀번호가 일치할 경우 홈 유스케이스를 실행한다. |
| 대체 흐름 | 2.a 아이디가 없거나 비밀번호가 일치하지 않는 경우 경고 문구를 띄운다. |

표 회원가입 유스케이스

|  |  |
| --- | --- |
| 유스케이스명 | 회원가입 |
| 액터명 | 주 액터 : 사용자 |
| 개요 | 사용자가 로그인을 하기 위해 회원가입을 해야한다 |
| 사전 조건 | - 사용자는 이미 직원으로 등록되어있어야 한다.  - 사용자는 자신의 직원 코드를 알아야 한다.  - 아이디는 중복될 수 없다. |
| 사후 조건 | - 등록한 아이디와 비밀번호로 로그인을 할 수 있다. |
| 기본 흐름 | 1. 사용자는 직원 코드, 아이디, 비밀번호를 입력한다.  2. 시스템은 직원 코드와 아이디 존재 여부를 파악한다.  3. 직원 코드가 존재하며 아이디가 없을 경우 데이터베이스에 해당 정보를 기록한다. |
| 대체 흐름 | 2.a 아이디가 있을 경우  2.a.1 아이디가 존재한다는 경고 문구를 띄운다.  2.b 직원코드가 없을 경우  2.b.1 직원코드가 없다는 경고 문구를 띄운다.  2.c 직원코드가 있는데 아이디가 존재할 경우  2.c.1 이미 회원 가입이 되어있다는 경고 문구를 띄운다. |

|  |  |
| --- | --- |
| 유스케이스명 | 홈 |
| 액터명 | 주 액터 : 사용자 |
| 개요 | 사용자가 로그인을 한 후에 처음으로 보는 유스케이스 |
| 사전 조건 | - 사용자가 로그인을 성공적으로 했어야 한다  - 사용자의 직원 코드에 따라 홈 화면에서 보이는 메뉴의 개수가 다르다. |
| 사후 조건 | - 없음. |
| 기본 흐름 | 없음. |
| 대체 흐름 | 없음. |

|  |  |
| --- | --- |
| 유스케이스명 | 공지 등록 |
| 액터명 | 주 액터 : 관리자 |
| 개요 | 관리자는 시스템에 중요한 공지를 등록할 수 있다. |
| 사전 조건 | - 사용자는 관리자의 직급을 가져야 한다. |
| 사후 조건 | - 데이터베이스에 등록된 공지는 홈 화면에서 열람이 되며 사후에 삭제가 가능하다. |
| 기본 흐름 | 1. 등록 버튼을 누른다.  2. 다이얼로그 창에 공지를 등록한다.  3. 데이터베이스를 통해 공지가 등록된다. |
| 대체 흐름 | 2.a 공지가 등록되지 않을 경우  2.a.1 공지가 등록되지 않았다는 경고창이 띄워진다. |

|  |  |
| --- | --- |
| 유스케이스명 | 재고 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 관리자와 사용자가 데이터베이스에서 추출된 재고 확인이 가능하다 |
| 사전 조건 | - 사용자는 이미 홈에 접속이 되어있어야 한다. |
| 사후 조건 | - 재고를 확인할 수 있다. |
| 기본 흐름 | 1. 사용자는 홈에서 재고 탭을 누른다.  2. 재고 화면에서 재고 표를 확인한 후 원하는 품목을 눌러 상품의 정보를 파악할 수 있다. |
| 대체 흐름 | - 없음 |

|  |  |
| --- | --- |
| 유스케이스명 | 매출 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 관리자와 사용자가 데이터베이스에서 추출된 매출 확인이 가능하다 |
| 사전 조건 | - 사용자는 이미 홈에 접속이 되어있어야 한다. |
| 사후 조건 | - 연도별, 월별 매출을 확인할 수 있다. |
| 기본 흐름 | 1. 사용자는 홈에서 매출 탭을 누른다.  2. 매출 화면에서 매출 표를 확인한다.  3. 원하는 연도, 월별 매출을 확인할 수 있다. |
| 대체 흐름 | - 없음. |

|  |  |
| --- | --- |
| 유스케이스명 | 주문 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 관리자와 사용자가 데이터베이스에서 부족한 상품을 추린 테이블을 확인 후 상품 주문이나 배송을 요청할 수 있다. |
| 사전 조건 | - 사용자는 이미 홈 화면에 접속이 되어있어야 한다. |
| 사후 조건 | - 없음 |
| 기본 흐름 | 1. 사용자는 홈에서 주문 탭을 누른다.  2. 재고가 부족한 항목들만이 올라온 테이블에서 원하는 항목을 누른다.  3. 재고가 있으며 가장 거리가 가까운 상점이 항목 정보와 함께 표시가 된다.  4. 주문이나 배송버튼을 눌러 원하는 항목과 물품을 주문한다. |
| 대체 흐름 | 3.a 원하는 재고의 가장 거리가 가까운 상점이 없을 경우  3.a.1 주문을 한다. |

|  |  |
| --- | --- |
| 유스케이스명 | 프로필 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 관리자와 사용자가 자신의 근무 상태를 파악한다. |
| 사전 조건 | - 사용자는 이미 홈에 들어가야 한다.  - 자신의 프로필에 사진이 등록되어 있어야 한다. |
| 사후 조건 | - 없음 |
| 기본 흐름 | 1. 사용자는 자신의 프로필 사진을 클릭한다.  2. 사용자의 정보가 있는 프로필이 보인다.  3. 자신의 근태 정보를 파악할 수 있으며 자신이 근태 상황을 추가할 수도 있다. |
| 대체 흐름 | - 없음 |

|  |  |
| --- | --- |
| 유스케이스명 | 직원 |
| 액터명 | 주 액터 : 관리자 |
| 개요 | 관리자만 사용 가능한 기능으로 직원을 관리할 수 있다. |
| 사전 조건 | - 관리자는 이미 홈에 접속해 있어야 한다. |
| 사후 조건 | - 없음. |
| 기본 흐름 | 1. 관리자는 직원 탭을 클릭한다.  2. 직원 목록을 볼 수 있으며 해당 직원을 클릭해 정보 또한 볼 수 있다. |
| 대체 흐름 | 없음 |

|  |  |
| --- | --- |
| 유스케이스명 | 직원 등록 |
| 액터명 | 주 액터 : 관리자 |
| 개요 | 관리자만 사용 가능한 기능으로 직원을 등록할 수 있다. |
| 사전 조건 | - 관리자는 이미 직원 탭에 들어와 있어야 한다. |
| 사후 조건 | - 데이터베이스와 화면에 있는 테이블이 업데이트 된다. |
| 기본 흐름 | 1. 관리자는 직원 등록 버튼을 클릭한다.  2. 직원 코드, 직원 이름, 시작 날짜 등을 입력해 직원을 등록한다. |
| 대체 흐름 | 2.a 등록에 실패한 경우  2.a.1 실패했다는 경고 문구가 뜬다. |

|  |  |
| --- | --- |
| 유스케이스명 | 근태 등록 |
| 액터명 | 주 액터 : 관리자, 사용자 |
| 개요 | 사용자가 근태를 등록할 수 있다. |
| 사전 조건 | - 사용자는 이미 근태 등록을 위한 패널에 있어야 한다. |
| 사후 조건 | - 데이터베이스와 화면에 있는 테이블이 업데이트 된다. |
| 기본 흐름 | 1. 사용자는 근태 등록 버튼을 누른다.  2. 근태 상황을 등록한다. |
| 대체 흐름 | 2.a 등록에 실패한 경우  2.a.1 실패했다는 경고 문구가 뜬다. |

## 4. 2 Sequence Diagram

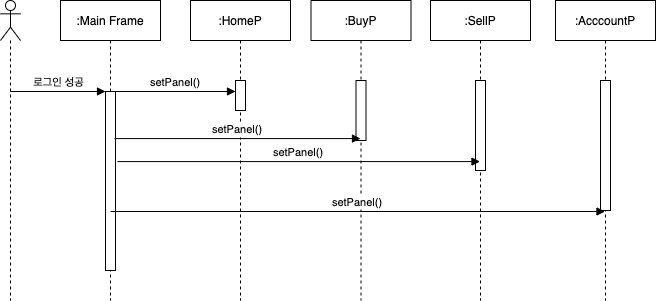
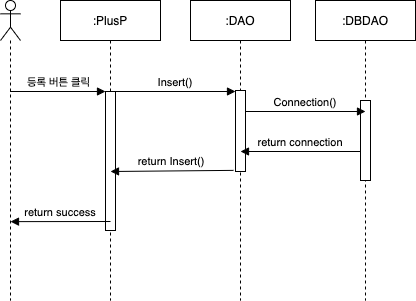
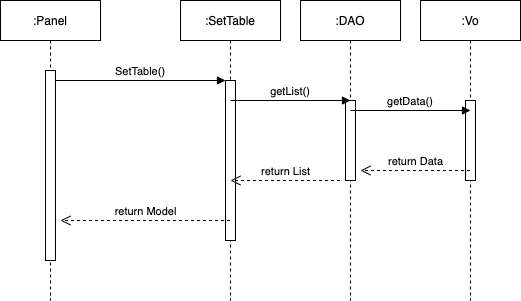


그림 . 로그인이 성공한 경우 시퀀스



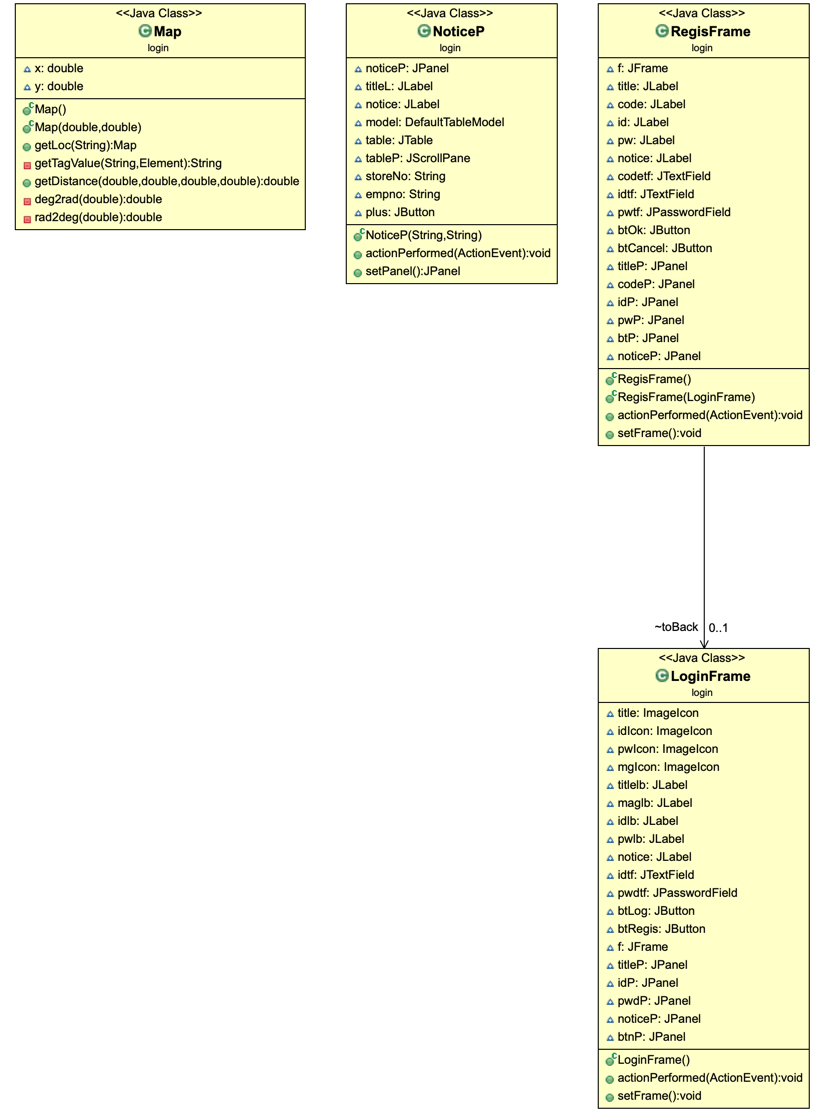
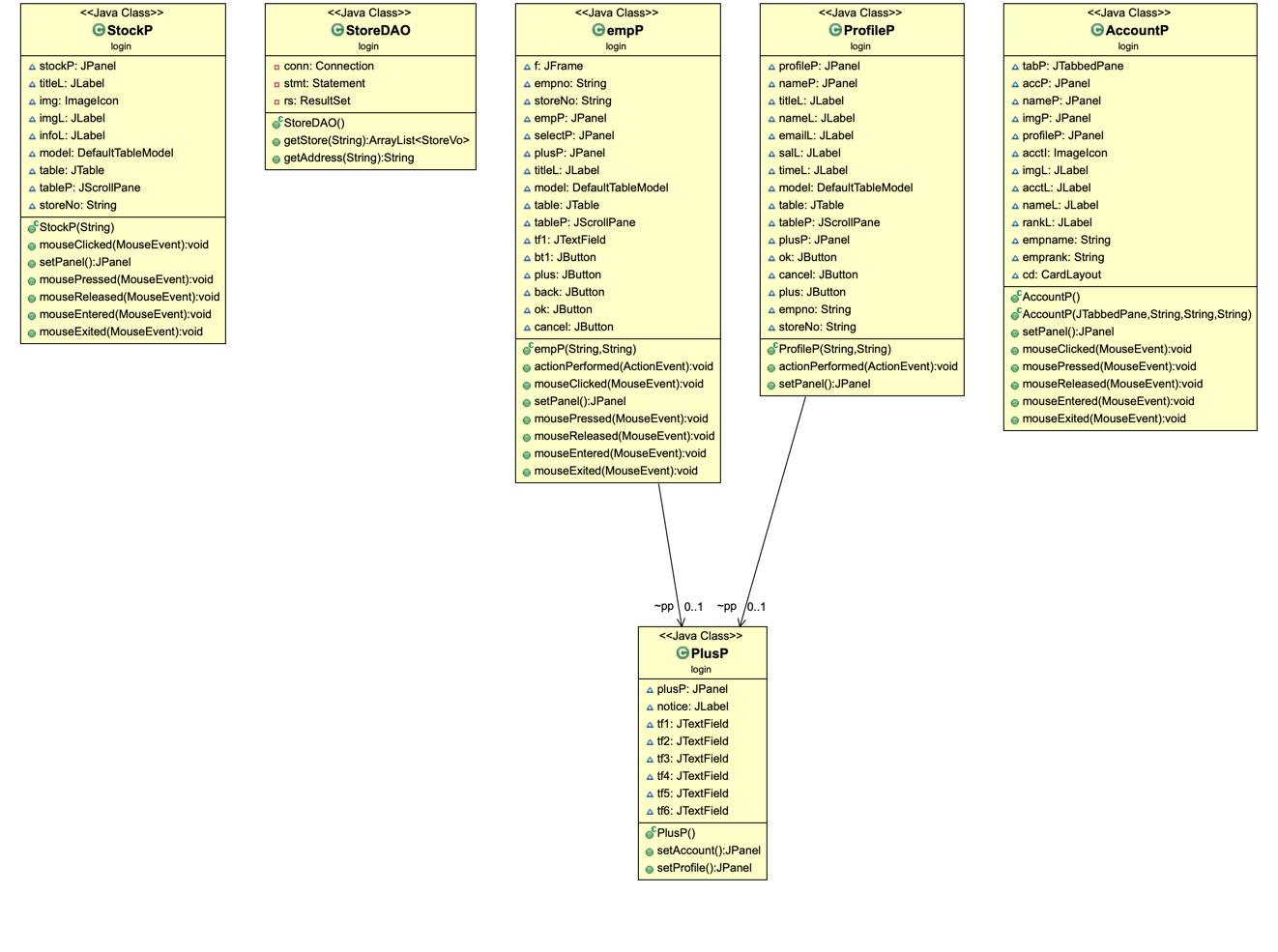
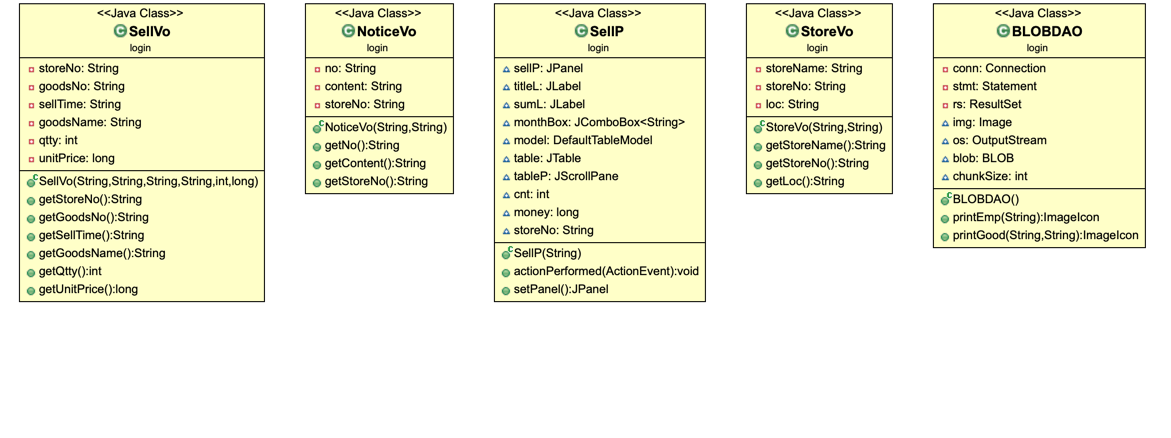
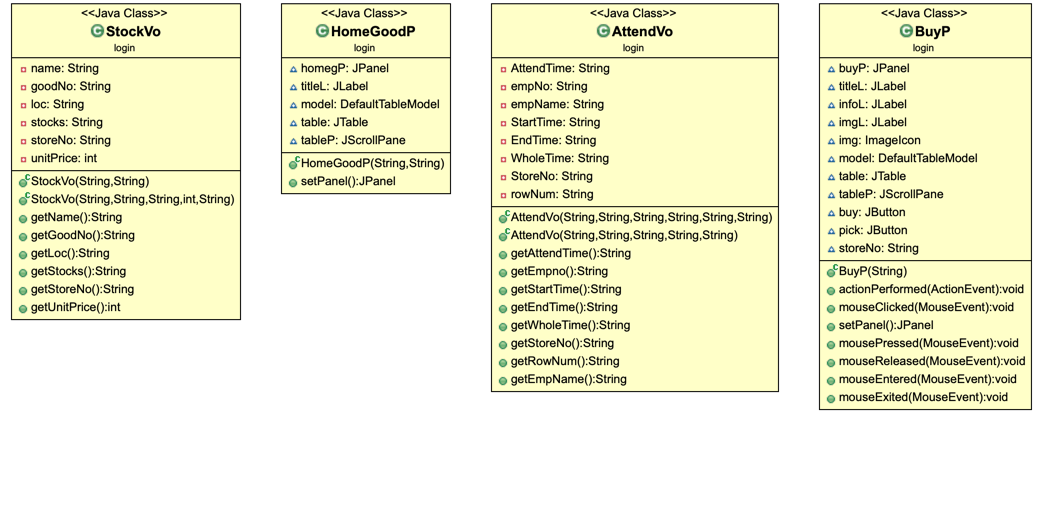
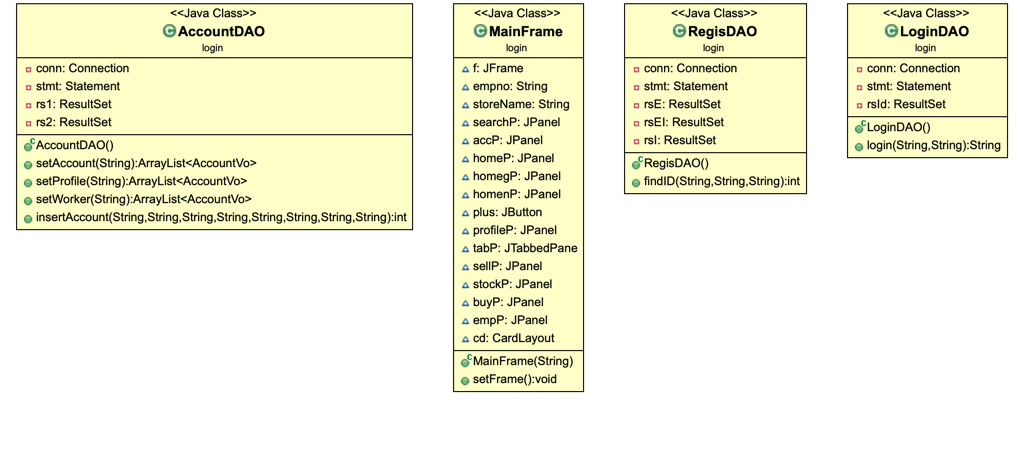
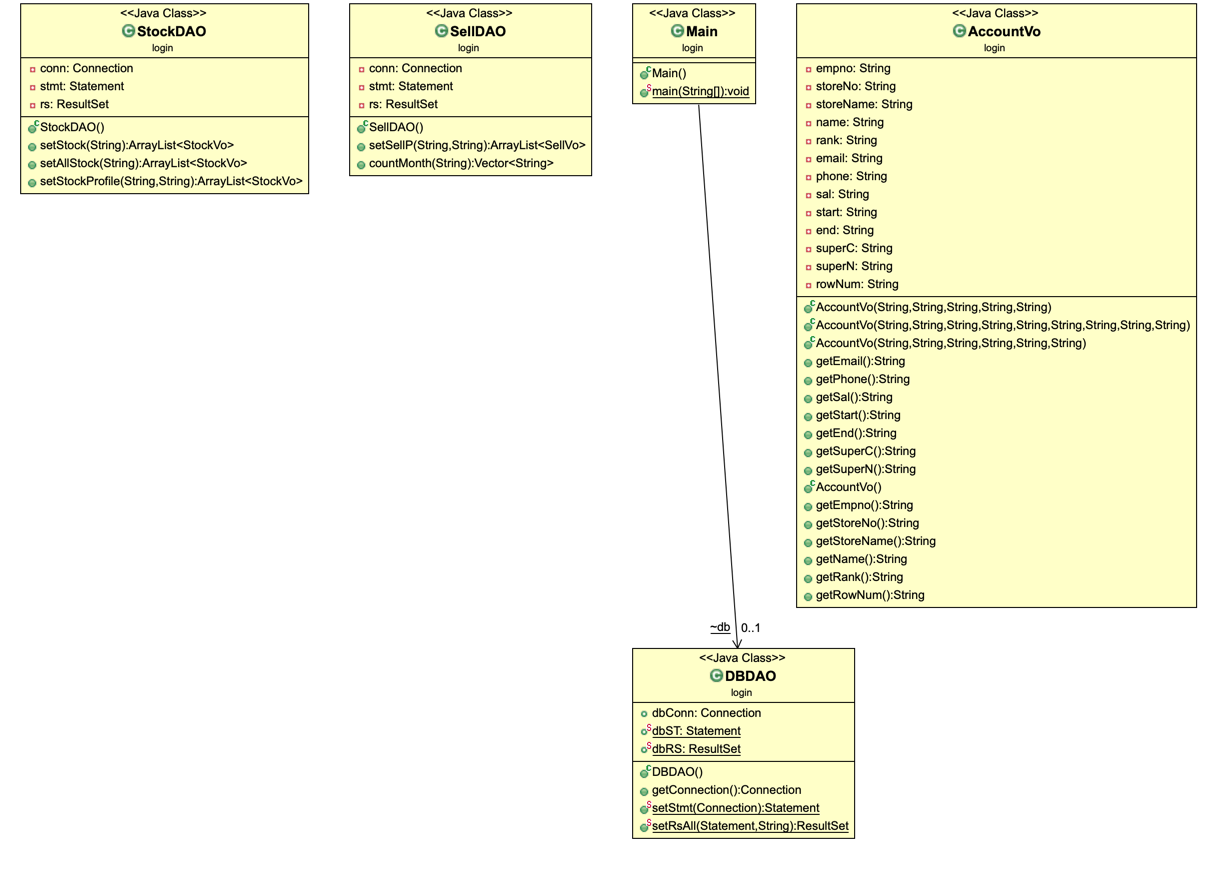
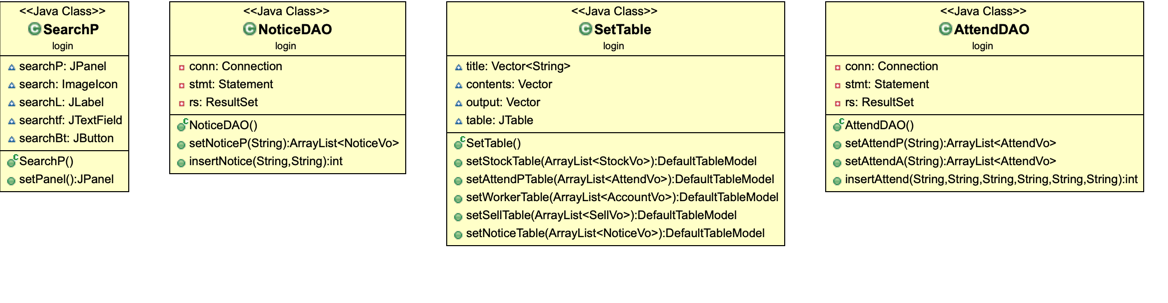
2. 등록하는 시퀀스



3. 패널이 테이블 셋하는 시퀀스

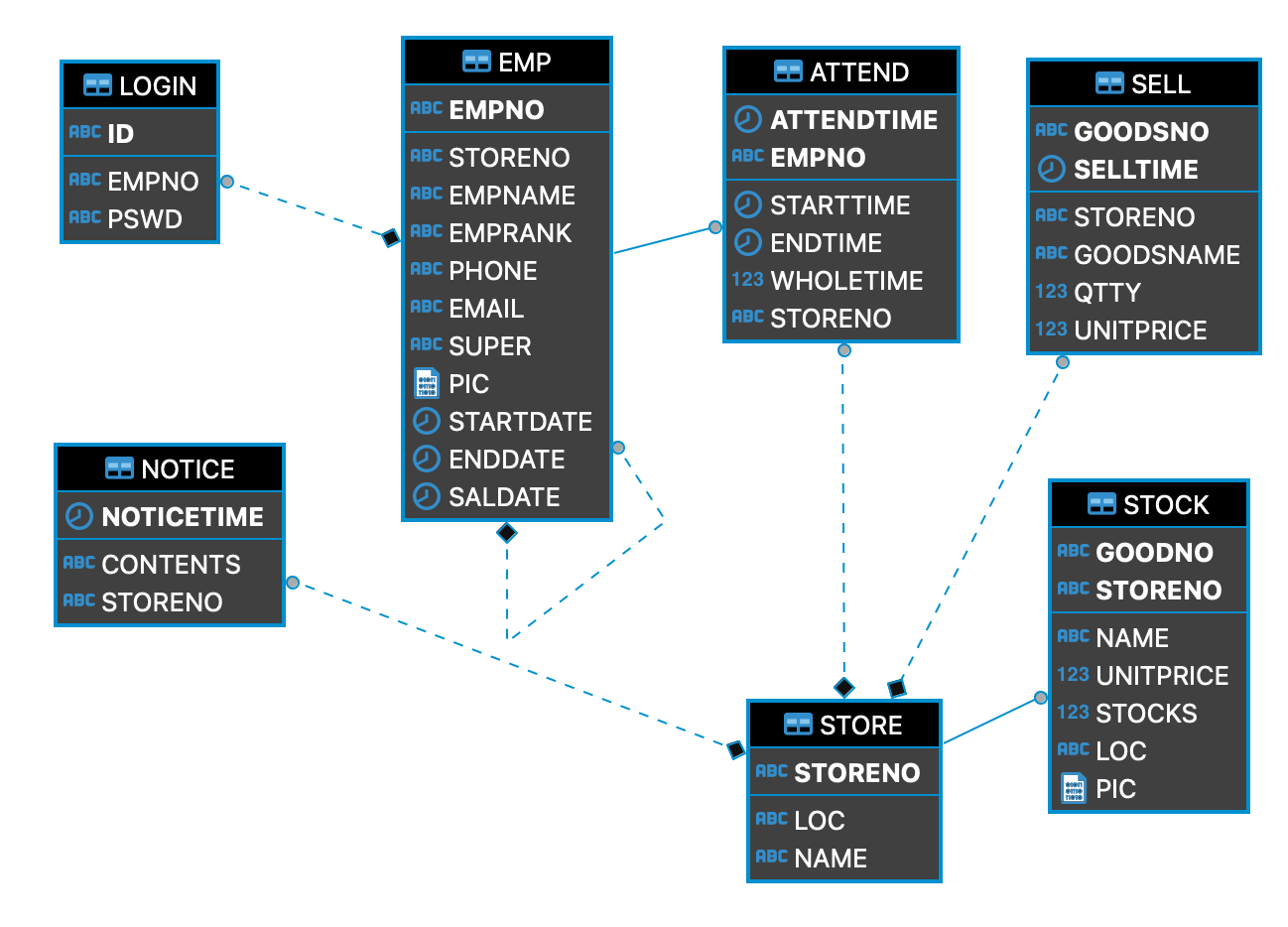
## 3 Class Diagram

|  |  |
| --- | --- |
|  | 클래스 다이어그램은 시간에 따라 변하지 않는 시스템의 정적인 명을 보여주는 구조 다이어그램이며 시스템을 구성하는 클래스들 사이의 관계를 표현한다. |



1. Frame, Panel의 분리
   1. 로그인 프레임, 회원 가입 프레임, 홈을 뜻하는 메인 프레임을 제외하고는 전부 패널에 패널을 추가하는 식으로 구현.
   2. 프레임에 패널을 추가하고 패널은 다른 클래스에서 메소드의 리턴값으로 넘겨줘 하나씩 생성
2. DAO, Vo의 분리
   1. DB에 접속하는 DAO와 각 값을 ArrayList로 담아 줄 Vo를 생성하여 체계적으로 프로그램을 구성
3. API를 JSON파싱으로 거리 불러들이는 Map 클래스 구축
   1. 받아들여온 정보를 바탕으로 현재 상점과의 거리를 잰다.

## 4. 4 ER Digaram



- LOGIN : 직원의 아이디 및 비밀번호를 담고 있는 테이블

- EMP : 직원 정보를 담고 있는 테이블

- ATTEND : 직원의 근태 정보를 담고 있는 테이블

- SELL : 매출 정보를 담고 있는 테이블

- STOCK : 재고 정보를 담고 있는 테이블

- STORE : 가게 정보를 담고 있는 테이블

- NOTICE : 공지를 담고 있는 테이블

# 5. 기대 효과

## 5. 1 원활한 매장 관리

프로그램을 사용하며 직원이나, 매출, 재고를 더 빠르게 파악할 수 있을 것이다.

## 5. 2 한 프로그램으로 모든 직원이 접속

직급이 다르더라도 접속할 수 있는 메뉴나 기능을 다르게 하여 한 프로그램으로도 각기 다른 직원이 허용된 매장 관리를 할 수 있다.

## 5. 3 매장 매출의 상승

원활한 매장 관리로 인하여 매장 관리가 용이해지고 그로 인해 매장의 매출이 상승될 수 있는 가능성이 있다.

# 6. 기타

## 6. 1 한계점 및 개선 사항

서버로 관리하는 것이 아니라 한 컴퓨터의 데이테베이스를 기반으로 재고 및 직원을 관리하는 형식으로 만들어진 프로그램이기 때문에 전국의 매장 관리 및 상호 교환이 원활하지는 않다. 백엔드를 연결하여 보완 가능하다.

또한 본 프로그램을 사용하는 매장에 근태를 확인하는 다른 단말기나 프로세스가 있다면 연동하여 근태를 직접 추가하지 않을 수 있을 것이다.

## 6. 2 개발 환경

* 운영체제 : JRE가 설치된 운영체제
* 언어 : java, sql
* 데이터베이스 : Oracle 11g
* 소스코드 작성 도구 : Eclipse EE
* 데이터베이스 관리 도구 : DBeaver