

GREEN COMPUTER ACADEMY

Board Game Café Management System

보드게임 카페 정산 시스템

이민영

2020-12-15

Contents

1. Introduction	4
1.1. Objectives.....	4
1.2. Project Overview.....	4
1.3. UML	4
A. Use Case Diagram	5
B. Class Diagram	5
C. Sequence Diagram	5
1.4. ER Diagram.....	5
1.5. Development Tools.....	5
2. Database Design.....	6
2.1. ER Diagram.....	6
2.2. Relational Schema.....	7
3. System Architecture	8
3.1. Use Case Diagram.....	8
3.1.1. Diagram.....	8
3.1.2. Description.....	9
3.2. Sequence Diagram	15
3.2.1. 직원 추가.....	15
3.2.2. 직원 권한 변경	16
3.2.3. 직원 삭제.....	17
3.2.4. 메뉴 추가.....	18

3.2.5. 메뉴 삭제	19
3.2.6. 정산	20
3.2.7. 정산 데이터 수정	21
3.2.8. 정산 데이터 삭제	22
3.3. Class Diagram	23
3.3.1. Overall	23
3.3.2. MainScreen	23
3.3.3. CustomerInsert	24
3.3.4. 직원과 관련된 Class Diagram	25
3.3.5. 메뉴와 관련된 Class Diagram	26
3.3.6. 시간당 요금과 관련된 Class Diagram	29
4. Result Screen	32
4.1. MainScreen	32
4.2. CustomerInsert	33
4.3. EmpInsert	34
4.4. AuthChange	35
4.5. EmpDelete	36
4.6. MenuManagement	36
4.7. MenuOrder	37
4.8. SalesTableMenu	38
4.9. HourlyCostChange	39
4.10. CalculateTime	39
4.11. SalesTableTime	40

5. Index.....	42
5.1. Table Index	42
5.2. Diagram Index	42
5.3. Figure Index	43

1. Introduction

1.1. Objectives

이 챕터에서는 본 프로젝트의 개요를 간략하게 소개하고, 사용된 다이어그램들과 도구를 정의할 것이다.

1.2. Project Overview

보드게임은 한국에서는 조선시대, 1980년대부터 일반적인 놀이 문화였으나, PC의 빠른 보급 및 온라인 게임 문화의 성장과 함께 보드게임 카페를 통해 공급이 상승하였다. 보드게임 산업은 가족문화, 아이들의 교육 및 놀이 문화, 재미있는 건전한 게임 문화라는 인식과 함께 매년 10-20% 씩 성장하는 산업이다¹.

Statista에서 글로벌 보드게임 시장가치를 분석한 자료에 따르면, 글로벌 보드게임 업계가 5년동안 약 2.5배 성장해 2021년에는 시장가치가 81억달러에 달할 것이라 한다. 그런데 2019년에 발표한 자료에 따르면 2019년을 기준으로 이미 시장가치가 85억달러에 해당하였고 2022년에는 110억 달러를 돌파할 것으로 예측하고 있다². 즉, 한국뿐만 아니라 전 세계적으로 보드게임의 전망과 성장속도에 대해 긍정적으로 평가하고 있음을 알 수 있다.

보드게임을 즐기고 싶지만 값비싼 보드게임 가격에 여러 종류를 사는 것이 망설여지는 사람들이 결국 찾을 곳은 보드게임 카페이다. 본 프로젝트에서는 보드게임 산업의 성장에 따른 보드게임 카페 이용률 증가를 토대로 보드게임 카페 자영업자들에게 도움이 되고자 보드게임 카페 정산 시스템을 만들었다.

1.3. UML

UML이란 Unified Modeling Language의 약어로서, 시스템 개발 과정에서 개발자간의 의사소통이 원활하게 이루어지도록 표준화한 대표적인 객체지향 모델링 언어이다.

이 프로젝트에서는 UML 중 Use Case Diagram, Class Diagram, Sequence Diagram을 묘사하였다.

¹ 사단법인 한국보드게임산업협회 <http://www.boardgame.or.kr/association/boardgame>

² Global board games market value <https://www.statista.com/statistics/829285/global-board-games-market-value/> 2019-04-15

A. Use Case Diagram

Use Case Diagram는 행위 다이어그램에 속하는 다이어그램이다. 사용자와 시스템 사이의 상호작용을 보여준다.

B. Class Diagram

Class Diagram은 구조 다이어그램에 속하면서 시스템을 구성하는 클래스들간의 관계를 보여준다.

C. Sequence Diagram

Sequence Diagram은 행위 다이어그램 중 상호작용 다이어그램에 속하는데, 시간 흐름에 따른 객체 간 상호작용을 표현한다.

1.4. ER Diagram

ER Diagram이란 Entity Relationship Model을 표현하는 다이어그램이다. 현실세계의 요구사항들로부터 데이터베이스를 설계하는 과정에서 활용된다.

1.5. Development Tools

Eclipse: [Eclipse 4.16.0 \(2020-06\)](#)

WindowBuilder: [Eclipse WindowBuilder 1.8.1](#)

Object Aid: [Eclipse ObjectAid 1.2.4 \(2019-12-31\)](#)

Java: [Java SE Development Kit 8u271](#)

Oracle: [Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production](#)

DBeaver: [DBeaver Community Edition 7.2.4.202011011603](#)

Diagram: [Diagrams.net](#)

Github: https://github.com/YeongL/JAVA_CLASS_2020/tree/main/library/BoardGameCafe

2. Database Design

데이터베이스를 묘사하기 위해 ER Diagram과 Relational Schema를 정의하였다.

2.1. ER Diagram

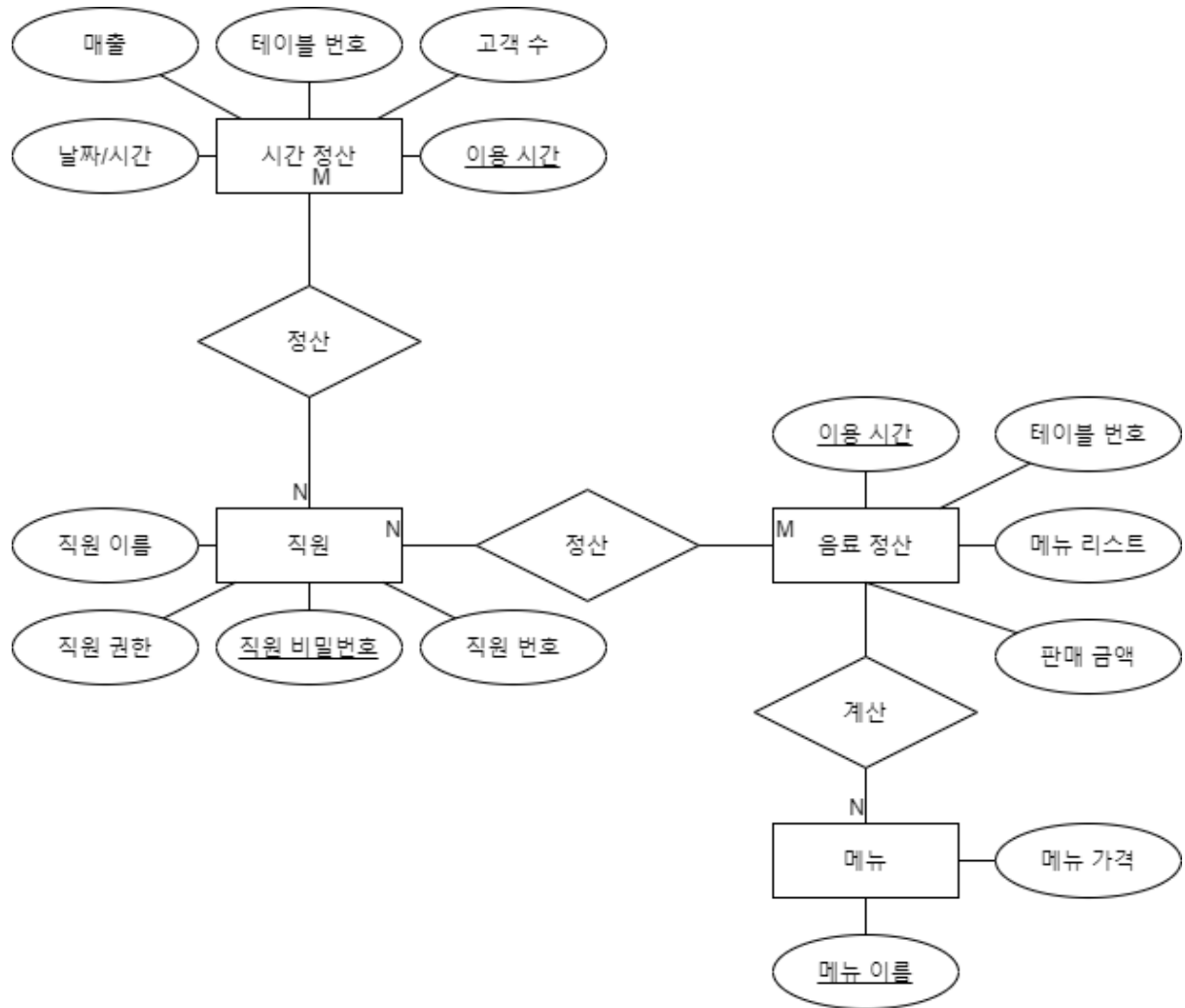


Diagram 1: ER Diagram

프로젝트 내에서 사용된 데이터베이스 객체들 간의 관계를 ER 다이어그램으로 표현하였다.

직원 객체의 직원 비밀번호를 시간 정산 객체와 음료 정산 객체가 각각 Foreign KEY로 상속한다.

음료 정산 객체의 메뉴 리스트 안에 주문한 음료들의 리스트가 나열된다. 직접적으로 연결되어 있지는 않지만 추후 묘사될 클래스 다이어그램에서 두 객체 간 관계를 확인할 수 있다.

2.2. Relational Schema

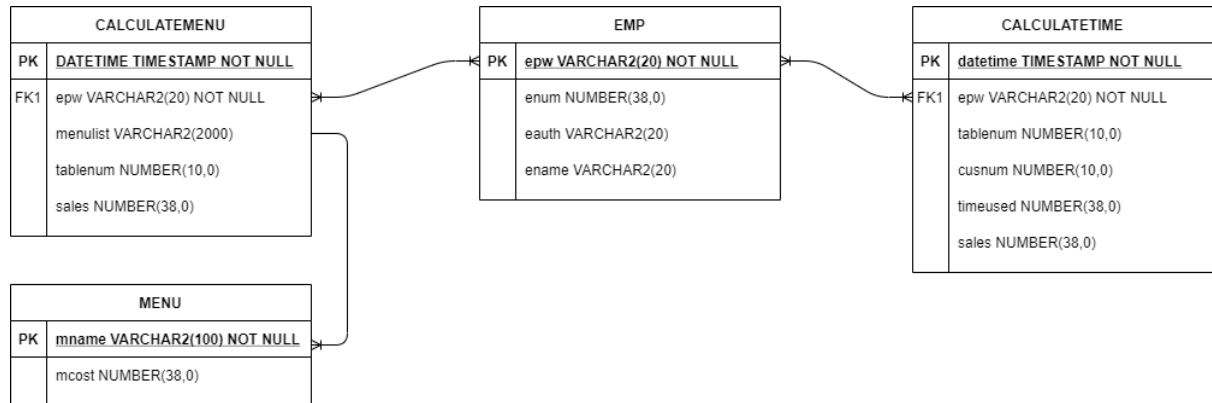


Diagram 2: Relational Schema

객체들을 테이블로 변환한 후 본 프로젝트에서 사용할 데이터베이스에 작성하였다.

3. System Architecture

3.1. Use Case Diagram

3.1.1. Diagram

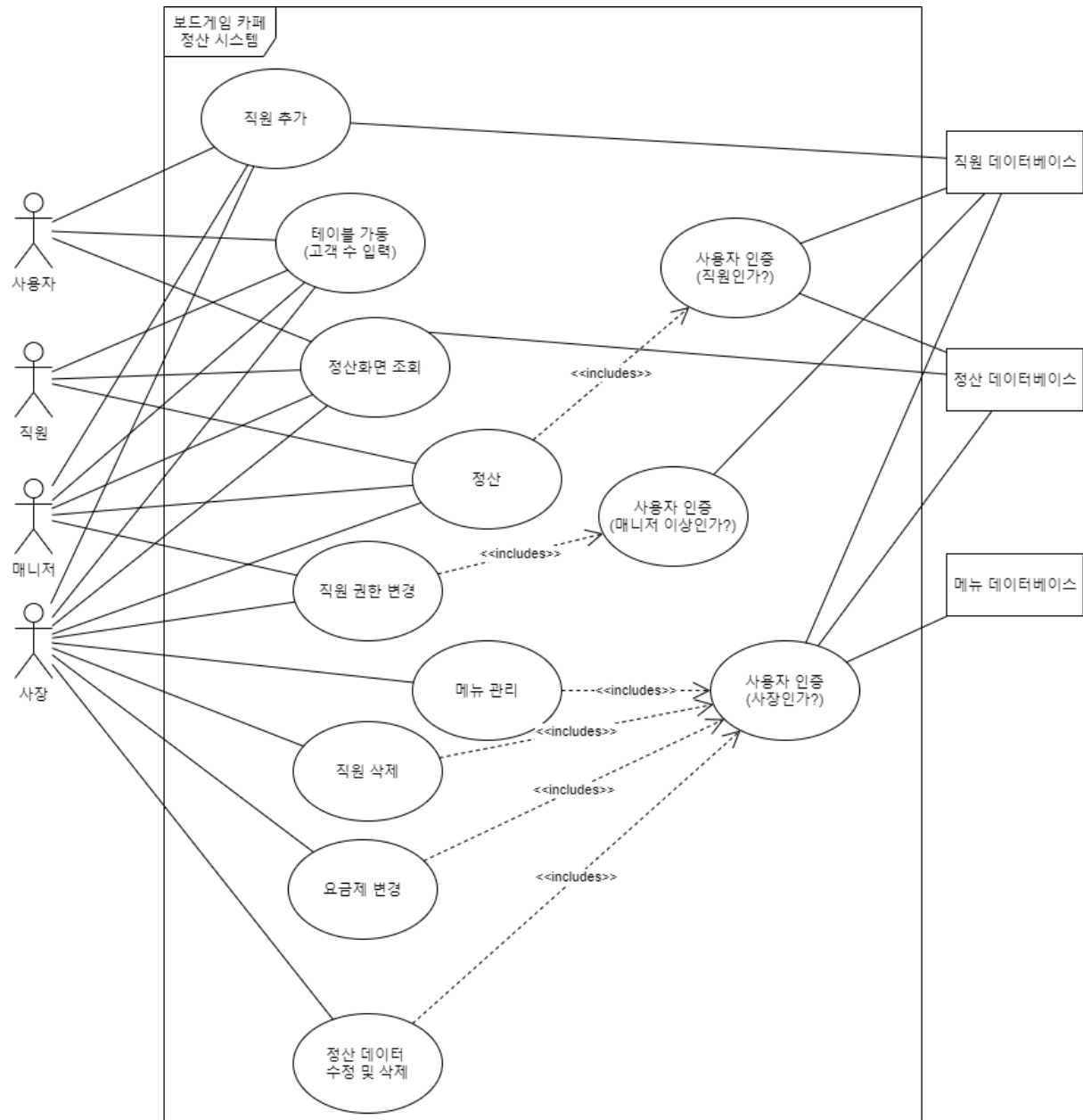


Diagram 3: Use Case Diagram

본 프로젝트의 사용자는 사장, 매니저, 직원으로, 외부 인원이 프로그램을 건드려도 데이터베이스 시스템에서 사장이 언제든지 삭제할 수 있게 만들었다.

3.1.2. Description

A. 직원 추가 Description

System	Board Game Café Settlement System
Use Case	직원 추가
Actors	사용자, 직원 DB
Description	정상 흐름: 1. 사용자가 직원의 이름, 고유 번호, 권한을 입력한다. 2. 사용자가 입력 버튼을 누른다. 3. 직원 데이터베이스에 새로운 직원의 정보가 등록된다. 대안 흐름: 1. 사용자가 정상흐름 2에서 취소 버튼이나 닫기 버튼을 누르는 경우, 직원 추가 창이 닫힌다. 예외 흐름: 1. 정상 흐름 1에서 고유 번호가 중복되는 값이거나, 숫자가 아닌 경우, 정상흐름 2를 시행하면 오류 메시지를 표시한다.
Stimulus	사용자가 직원 추가 메뉴를 누른다.
Response	직원 추가 창이 뜬다.
Comments	직원의 고유 번호가 Primary key이기 때문에 중복되는 값이면 오류를 표출하게 하였다.

Table 1: 직원 추가 Table

B. 테이블 가동 Description

System	Board Game Café Settlement System
Use Case	테이블 가동
Actors	사용자
Description	정상 흐름: 1. 사용자가 숫자를 입력한다. 2. 테이블에 시간이 흘러간다. 3. 테이블에 입력한 숫자가 표시된다. 4. 테이블에 할당된 클래스에 숫자가 입력된다. 대안 흐름: 1. 정상 흐름 1에서 사용자가 숫자를 입력하지 않고 X버튼을 눌러 창을 닫아버려도 정상흐름 2가 가동되므로 따로 취소 메뉴를 눌러줘야 한다. 예외 흐름: 1. 사용자가 정상 흐름 1에서 숫자가 아닌 다른 값을 입력하면 숫자를 입력하라는 오류 메시지를 띄운다.

Stimulus	메인 화면의 테이블에 할당된 시작 메뉴를 사용자가 누른다.
Response	숫자 입력 화면이 뜬다.
Comments	직접적으로 데이터베이스에 영향을 주지 않는 단계이므로 액터를 사용자 하나로 정의하였다.

Table 2: 테이블 가동 Table

C. 정산화면 조회 Description

System	Board Game Café Settlement System
Use Case	정산화면 조회
Actors	사용자, 정산 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 1. 사용자가 정산화면 조회 메뉴를 누른다. 2. 정산 데이터베이스에서 정산 정보를 불러온다. 3. 정산화면 조회 화면에 정산 데이터가 표시된다.
Stimulus	사용자가 정산화면 조회 메뉴를 누른다.
Response	정산화면 조회 화면이 뜬다.
Comments	사용자에 관계없이 정산 데이터베이스에 있는 정보를 조회할 수 있다.

Table 3: 정산화면 조회 Table

D. 정산 Description

System	Board Game Café Settlement System
Use Case	정산
Actors	사용자(직원, 매니저, 사장), 직원 DB, 정산 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 1. 사용자가 고유 번호를 입력한다. 2. 사용자가 정산 버튼을 누른다. 3. 직원 데이터베이스에 고유 번호가 있는지 확인한다. 4. 정산 정보가 정산 데이터베이스에 저장된다. <p>대안 흐름:</p> <ol style="list-style-type: none"> 1. 정상 흐름 2에서 사용자가 취소 버튼을 눌렀을 때, 정산 화면이 닫힌다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 1. 정상 흐름 1에서 사용자가 고유 번호 자리에 숫자를 입력하지 않으면 숫자가 아니라는 오류 메시지가 표시된다. 2. 정상 흐름 3에서 사용자가 입력한 숫자가 직원 데이터베이스에 없을 시 권한을 다시 입력하라는 오류 메시지가 표시된다.
Stimulus	사용자가 정산 메뉴를 누른다.

Response	정산 화면이 뜬다.
Comments	이 정산은 음료 계산이 비 포함된 시간당 요금만 계산하는 정산이기 때문에 메뉴 데이터베이스를 포함하지 않았다.

Table 4: 정산 Description

E. 직원 권한 변경 Description

System	Board Game Café Settlement System
Use Case	직원 권한 변경
Actors	사용자(사장, 매니저), 직원 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 1. 직원 데이터베이스에서 직원 정보를 불러와서 직원 권한 변경 화면에 표시한다. 2. 사용자가 직원을 선택한다. 3. 사용자가 변경할 권한을 선택한다. 4. 사용자가 고유 번호를 입력한다. 5. 사용자가 변경 버튼을 누른다. 6. 직원 데이터베이스에서 사장 또는 매니저의 고유 번호가 맞는지 확인한다. 7. 직원 데이터베이스에서 선택된 직원의 권한을 변경한다. <p>대안 흐름:</p> <ol style="list-style-type: none"> 1. 정상흐름 4에서 사용자가 변경이 아닌 취소 버튼을 눌렀을 시, 직원 권한 변경 화면이 닫힌다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 1. 정상흐름 5에서 사용자가 입력한 번호가 사장 또는 매니저의 고유 번호가 아닐 시, 권한이 부족하다는 오류 메시지를 띄운다.
Stimulus	사용자가 직원 권한 변경 메뉴를 누른다.
Response	직원 권한 변경 화면이 뜬다.
Comments	사장 또는 매니저가 아니면 다른 직원의 권한을 변경할 수 없게 해 놓았다.

Table 5: 직원 권한 변경 Description

F. 메뉴 관리 Description

System	Board Game Café Settlement System
Use Case	메뉴 관리
Actors	사장, 직원 DB, 메뉴 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 1. 사용자가 메뉴 관리 메뉴를 누른다. 2. 메뉴 데이터베이스에서 메뉴 데이터를 불러와서 메뉴 관리 화면에 띄

	<p>운다.</p> <ol style="list-style-type: none"> 3. 사용자가 메뉴 추가 버튼을 누른다. 4. 사용자가 데이터와 고유 권한을 입력한다. 5. 사용자가 추가 버튼을 누른다. 6. 직원 데이터베이스에서 고유 권한이 사장의 권한인지 확인한다. 7. 메뉴 데이터베이스에 추가한 메뉴가 추가된다. 8. 메뉴 관리 화면에 추가한 메뉴가 추가된다. <p>대안 흐름:</p> <ol style="list-style-type: none"> 1. 정상흐름 3에서 메뉴 추가 버튼이 아닌 메뉴 삭제 메뉴를 누르면 고유 권한을 입력하는 화면이 표시된다. 2. 고유 권한을 입력하고 삭제 버튼을 누른다. 3. 직원 데이터베이스에서 입력한 고유 권한이 사장의 권한인지 확인한다. 4. 메뉴 데이터베이스에서 선택한 메뉴가 삭제된다. 5. 메뉴 관리 화면에서 선택한 메뉴가 삭제된다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 1. 정상흐름 6과 대안흐름 3에서 입력한 권한이 사장의 권한이 아닐 시, 권한이 부족하다는 메시지가 표시된다. 2. 정상흐름 4에서 입력한 데이터에 중복되는 값이 있을 시 값이 중복된다는 메시지를 표시한다. 3. 정상흐름 4에서 가격 부분에 숫자가 아닌 값을 입력 시, 숫자를 입력하라는 오류 메시지를 표시한다.
Stimulus	사장이 메뉴 관리 메뉴를 누른다.
Response	메뉴 관리 화면이 표시된다.
Comments	메뉴 관리 화면에서는 메뉴 추가 및 삭제가 가능하다.

Table 6: 메뉴 관리 Description

G. 직원 삭제 Description

System	Board Game Café Settlement System
Use Case	직원 삭제
Actors	사장, 직원 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 1. 직원 데이터베이스에서 직원 정보를 불러와서 직원 삭제 화면에 표시한다. 2. 사장이 직원을 선택한다. 3. 사장이 고유 번호를 입력한다. 4. 사장이 삭제 버튼을 누른다. 5. 직원 데이터베이스에서 사장의 고유 번호가 맞는지 확인한다. 6. 직원 데이터베이스에서 선택된 직원을 삭제한다.

	<p>대안 흐름:</p> <ol style="list-style-type: none"> 정상흐름 4에서 사장이 삭제가 아닌 취소 버튼을 눌렀을 시, 직원 삭제 화면이 닫힌다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 정상흐름 5에서 사용자가 입력한 번호가 사장의 고유 번호가 아닐 시, 권한이 부족하다는 오류 메시지를 띄운다.
Stimulus	사장이 직원 삭제 메뉴를 누른다.
Response	직원 삭제 화면이 표시된다.
Comments	직원 삭제는 사장만 가능하게 해 놓았다.

Table 7: 직원 삭제 Description

H. 요금제 변경 Description

System	Board Game Café Settlement System
Use Case	요금제 변경
Actors	사용자(사장), 직원 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 요금제 변경 화면에 현재 요금제가 표시된다. 사용자가 변경할 요금을 입력한다. 사용자가 고유 권한을 입력한다. 사용자가 변경 버튼을 누른다. 직원 데이터베이스에서 사용자가 입력한 권한이 사장의 권한인지 확인한다. 요금제 변수에 할당된 값이 사용자가 입력한 값으로 바뀐다. <p>대안 흐름:</p> <ol style="list-style-type: none"> 정상흐름 4에서 사용자가 변경이 아닌 취소 버튼을 누를 시 요금제 변경 화면이 닫힌다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 정상흐름 2에서 사용자가 변경할 요금 입력 창에 숫자가 아닌 다른 값을 입력하고 변경 버튼을 눌렀을 시 숫자가 아니라는 오류 메시지를 띄운다. 정상흐름 5에서 사용자가 입력한 권한이 사장의 권한이 아닐 시 권한이 부족하다는 오류 메시지를 띄운다.
Stimulus	사용자가 요금제 변경 메뉴를 클릭한다.
Response	요금제 변경 화면이 표시된다.
Comments	시간당 요금제 변수는 메인 화면에 public으로 정의되어 있다.

Table 8: 요금제 변경 Description

I. 정산 데이터 수정 및 삭제 Description

System	Board Game Café Settlement System
Use Case	정산 데이터 수정 및 삭제
Actors	사용자(사장), 직원 DB, 정산 DB
Description	<p>정상 흐름:</p> <ol style="list-style-type: none"> 수정 화면 <ol style="list-style-type: none"> 사용자가 정산 수정 화면에서 변경할 값들을 입력하고 고유 권한을 입력한다. 사용자가 수정 버튼을 누른다. 직원 데이터베이스에서 사용자가 입력한 고유 권한이 사장인지 확인한다. 정산 데이터베이스에 있는 값들이 수정된다. 정산 조회 화면에 있는 해당 값이 수정된다. 삭제 화면 <ol style="list-style-type: none"> 사용자가 삭제 화면에서 고유 권한을 입력한다. 사용자가 삭제 버튼을 누른다. 직원 데이터베이스에서 사용자가 입력한 고유 권한이 사장인지 확인한다. 정산 데이터베이스에 있는 해당 값이 삭제된다. 정산 조회 화면에 있는 해당 값이 삭제된다. <p>대안 흐름:</p> <ol style="list-style-type: none"> 정상 흐름 - 수정 화면 2와 정상 흐름 - 삭제 화면 2에서 취소 버튼을 누를 시, 해당 창이 닫힌다. <p>예외 흐름:</p> <ol style="list-style-type: none"> 정상 흐름 - 수정 화면 1에서 사용자가 옳지 못한 데이터를 입력하면 잘못된 데이터 입력이라는 오류 메시지를 표시한다. 정상 흐름 - 수정 화면 3과 정상 흐름 - 삭제 화면 3에서 고유 권한이 사장이 아닐 시 권한이 부족하다는 오류 메시지를 화면에 띄운다.
Stimulus	정산 조회 화면에서 사용자가 수정 또는 삭제 버튼을 누른다.
Response	정산 데이터 수정 / 삭제 화면이 표시된다.
Comments	메뉴 추가 및 삭제와 비슷한 흐름으로 진행된다.

Table 9: 정산 데이터 수정 및 삭제 Description

3.2. Sequence Diagram

3.2.1. 직원 추가

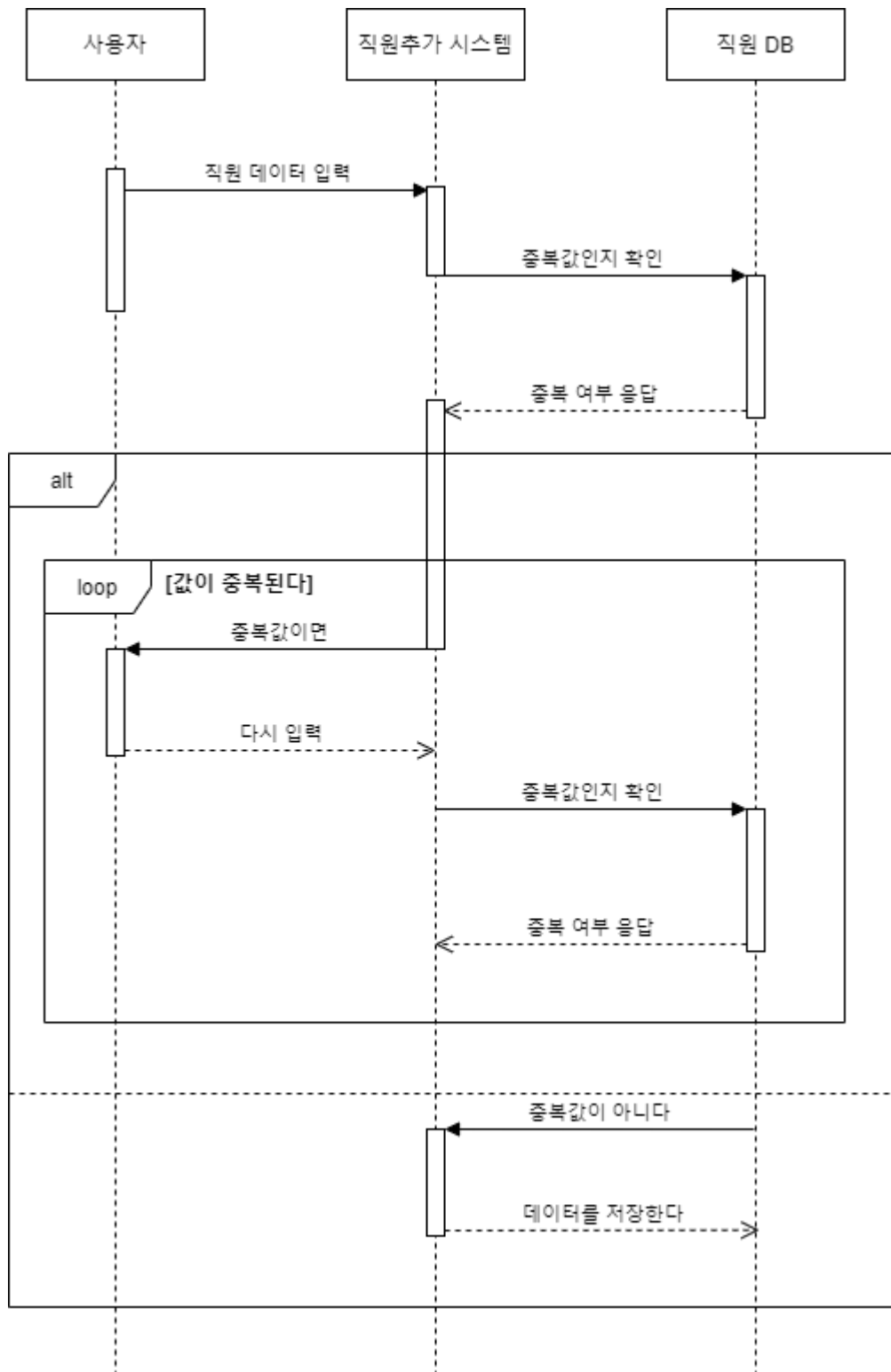


Diagram 4: 직원 추가 Sequence Diagram

3.2.2. 직원 권한 변경

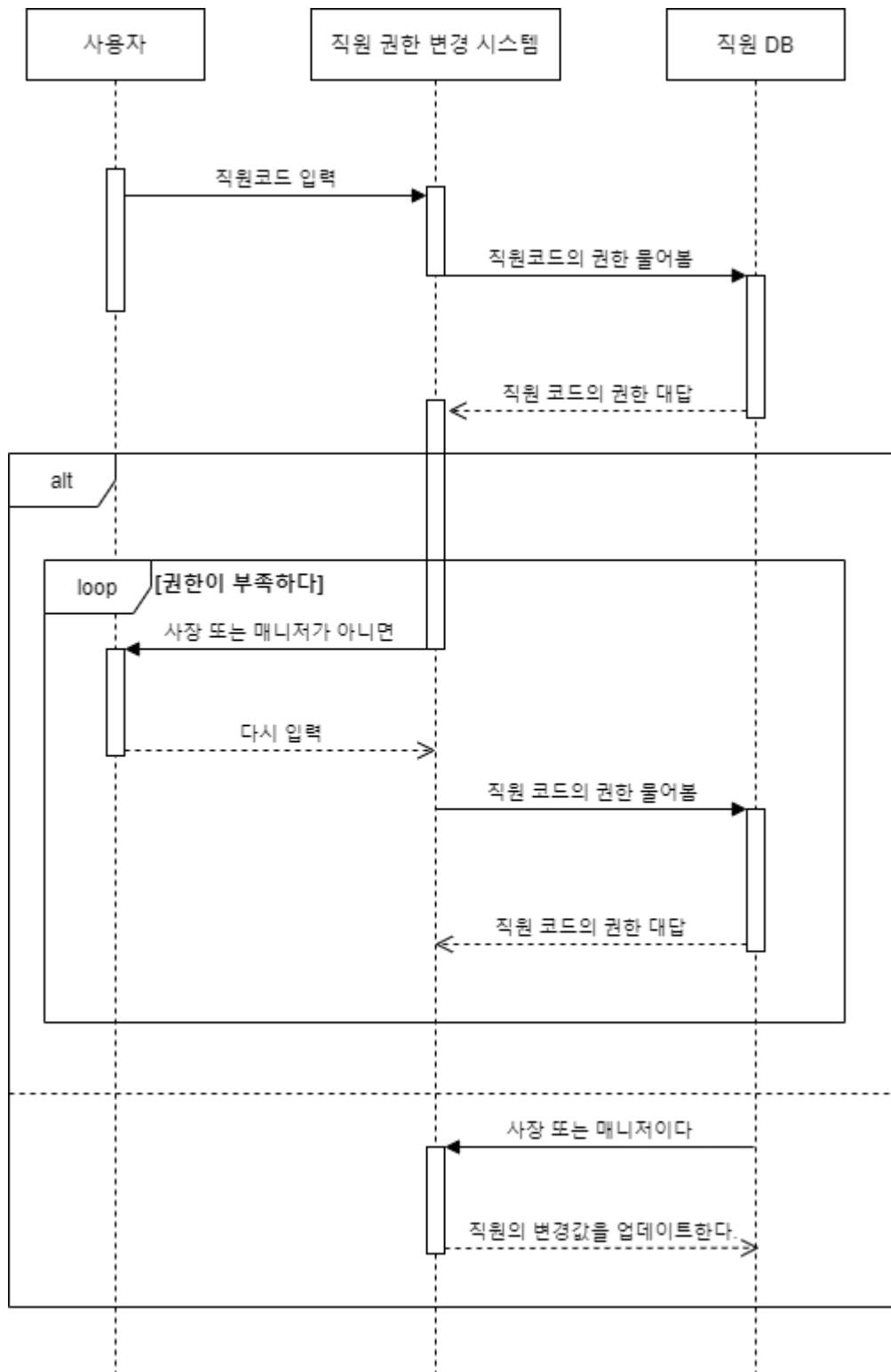


Diagram 5: 직원 권한 변경 Sequence Diagram

3.2.3. 직원 삭제

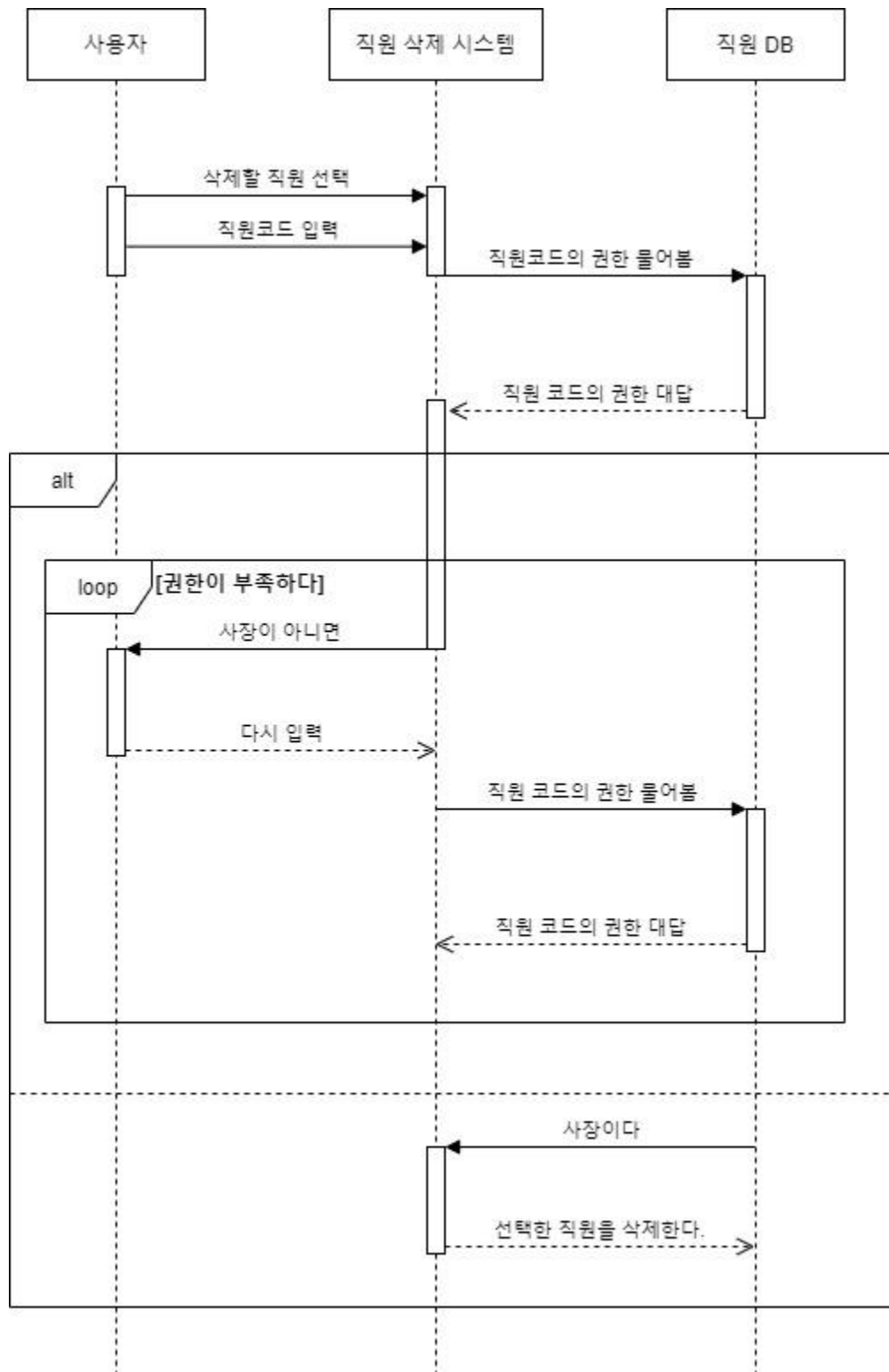


Diagram 6: 직원 삭제 Sequence Diagram

3.2.4. 메뉴 추가

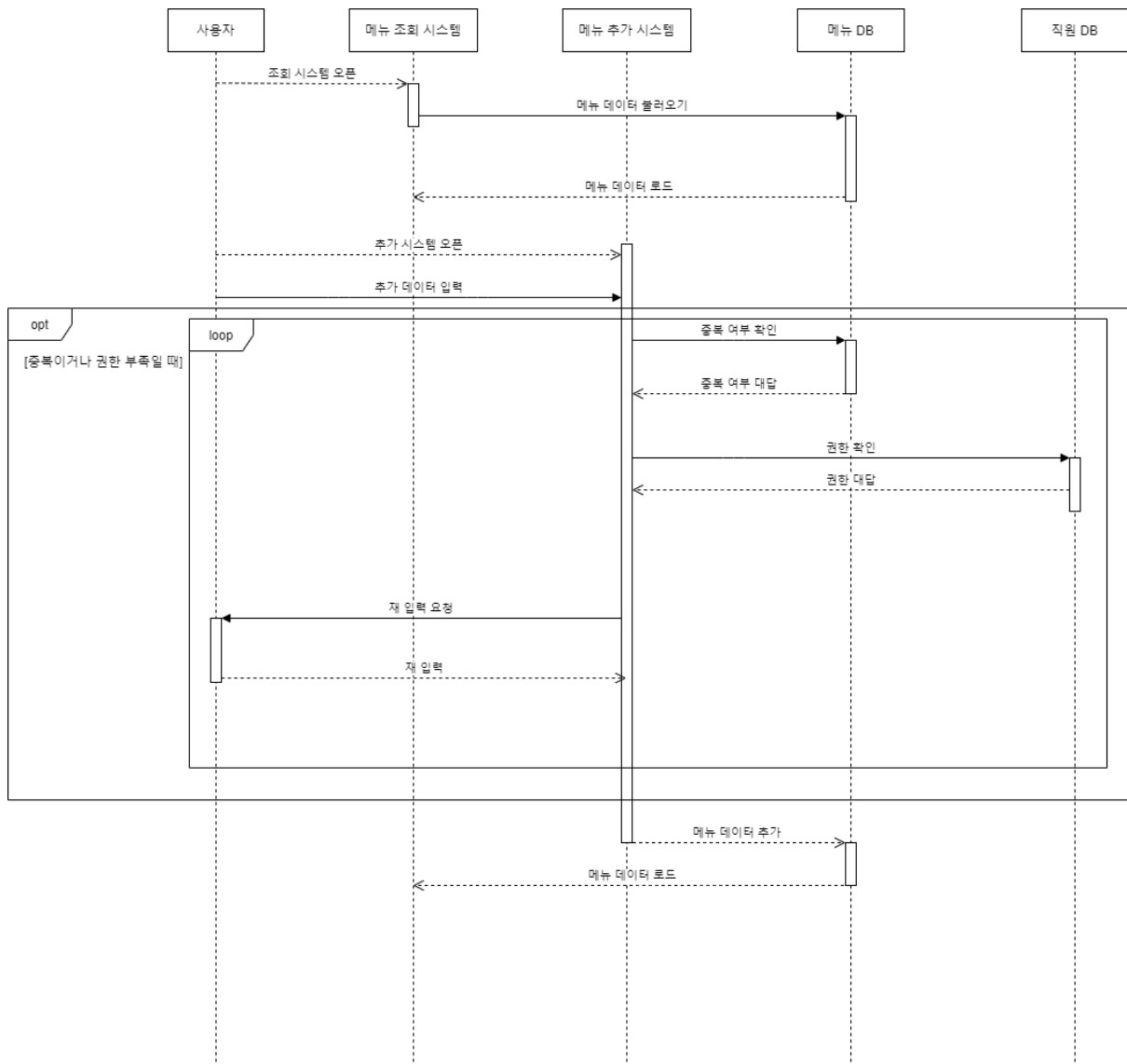


Diagram 7: 메뉴 추가 Sequence Diagram

메뉴 추가에 관한 Sequence Diagram을 표현해보았다. 사용자가 메뉴 조회 시스템을 오픈한 후 조회 시스템은 메뉴 데이터베이스에서 메뉴들의 데이터를 불러와서 조회 시스템에 표시한다. 메뉴 조회 시스템에서 사용자가 메뉴 추가를 누르면 메뉴 추가 시스템이 열린다. 추가할 데이터를 입력하면 중복 여부와 권한을 각각 확인한다. 권한이 부족하거나 데이터가 중복이면 메뉴 데이터를 데이터베이스에 추가할 수 없다.

3.2.5. 메뉴 삭제

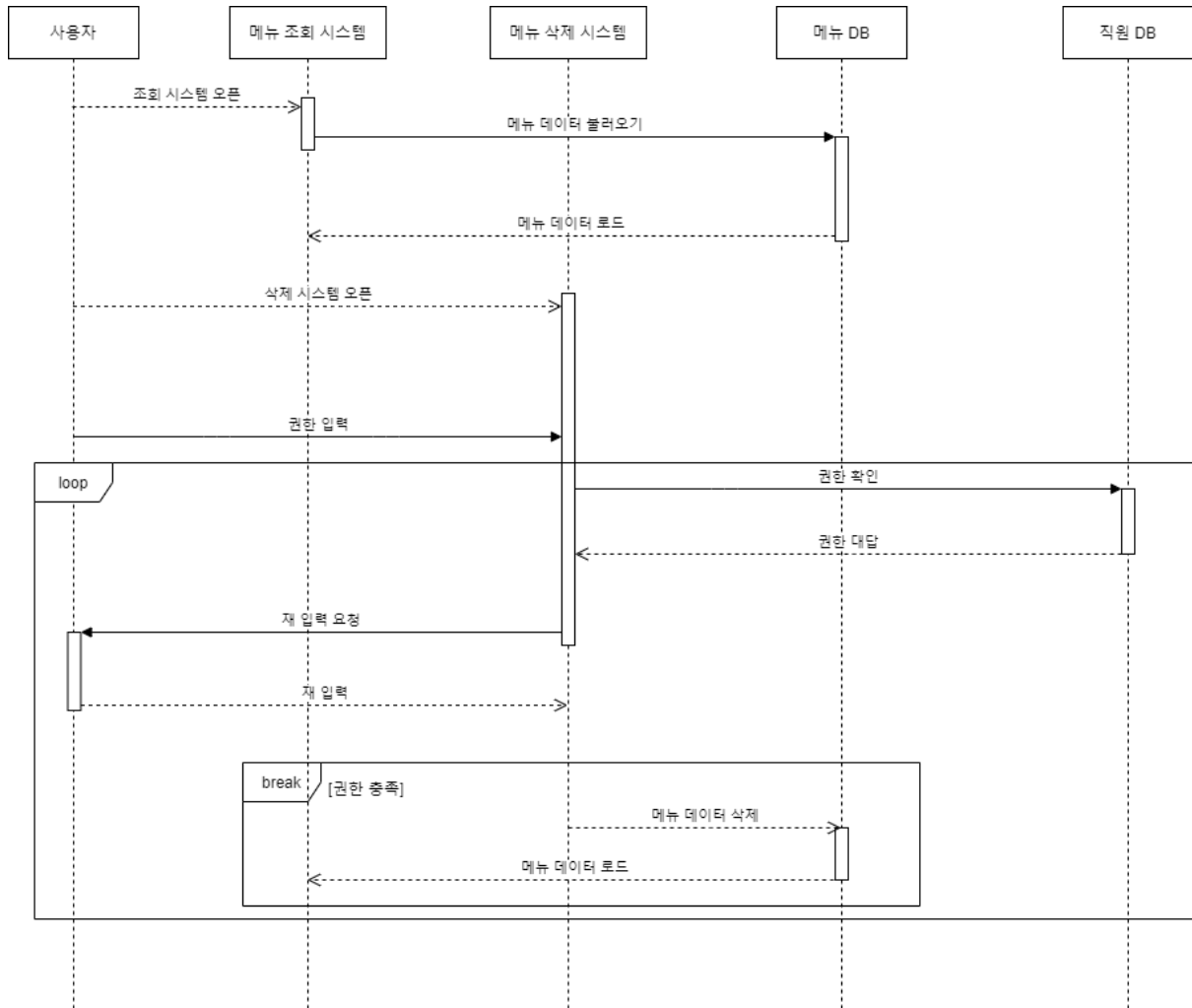


Diagram 8: 메뉴 삭제 Sequence Diagram

메뉴 삭제도 메뉴 추가와 비슷하게 이루어진다. 다만 차이점은 메뉴 삭제는 중복되는 값을 확인하는 과정이 생략되었다.

3.2.6. 정산

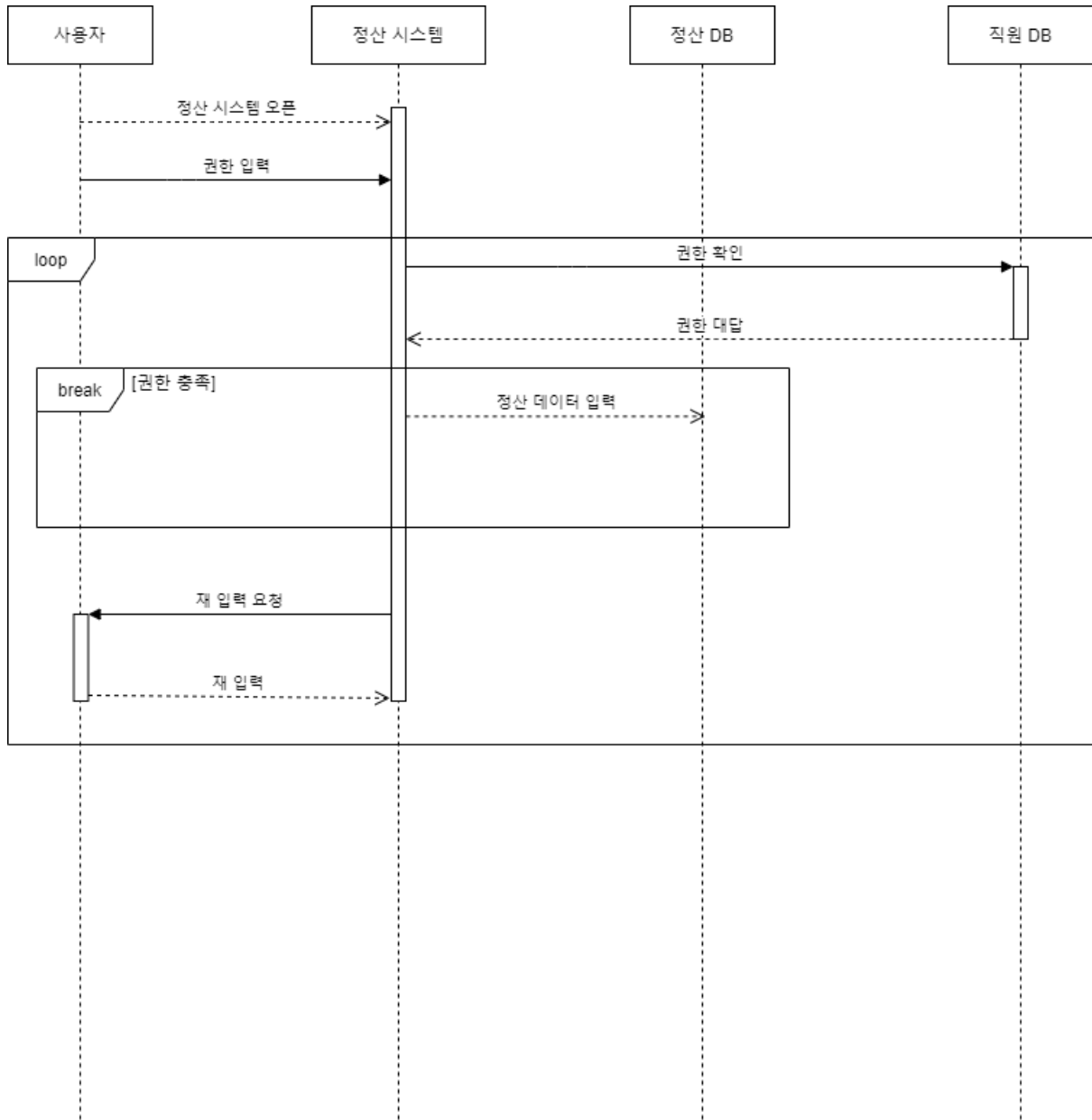


Diagram 9: 정산 Sequence Diagram

정산 시스템도 메뉴 추가와 비슷한 과정으로 진행된다. 권한을 충족하면 정산 데이터베이스에 값을 저장하고 충족하지 못하면 저장하지 않는다.

3.2.7. 정산 데이터 수정

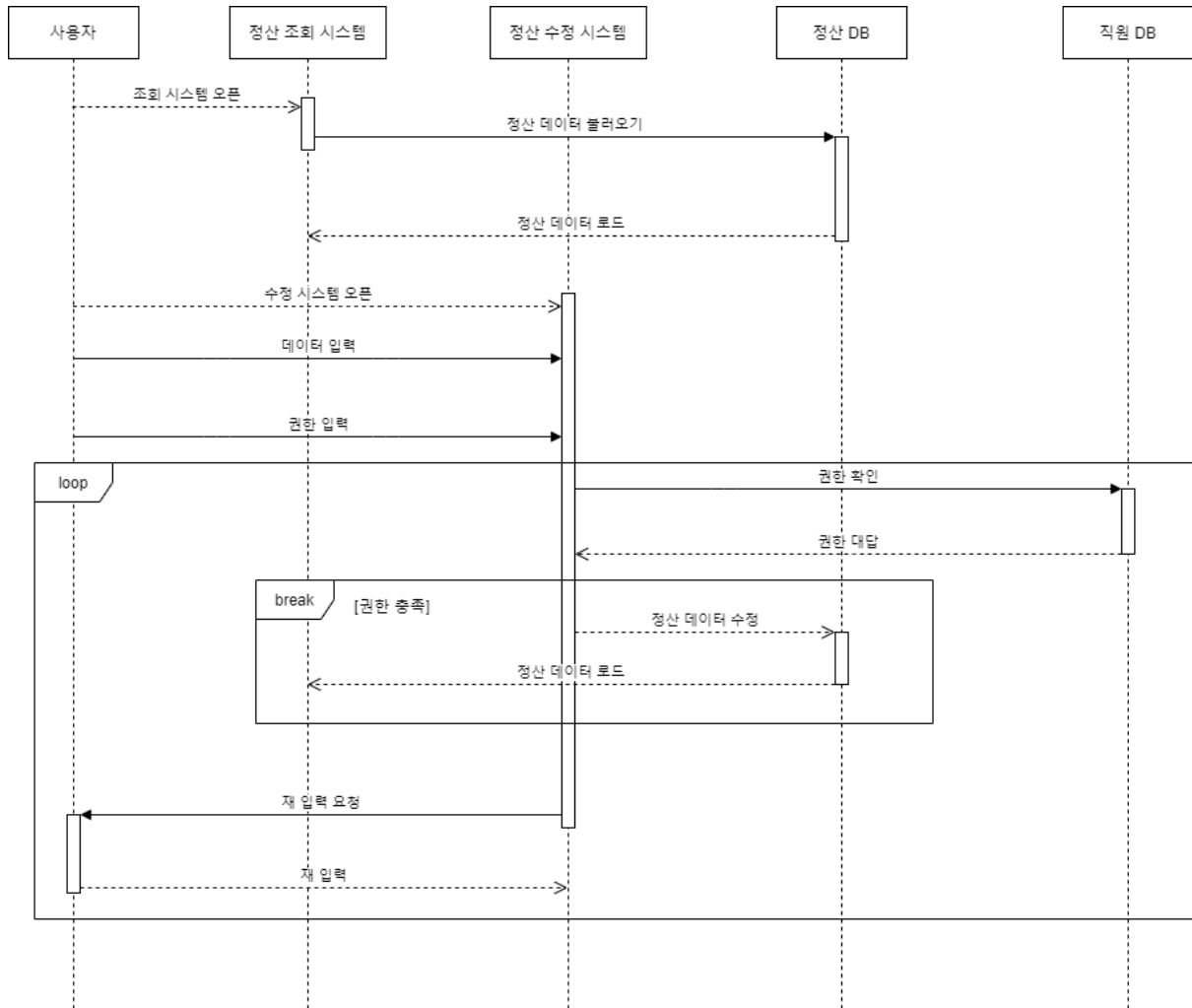


Diagram 10: 정산 데이터 수정 Sequence Diagram

정산 데이터 수정은 정산 과정과 비슷하지만 중간에 데이터 입력 과정이 추가되었다.

3.2.8. 정산 데이터 삭제

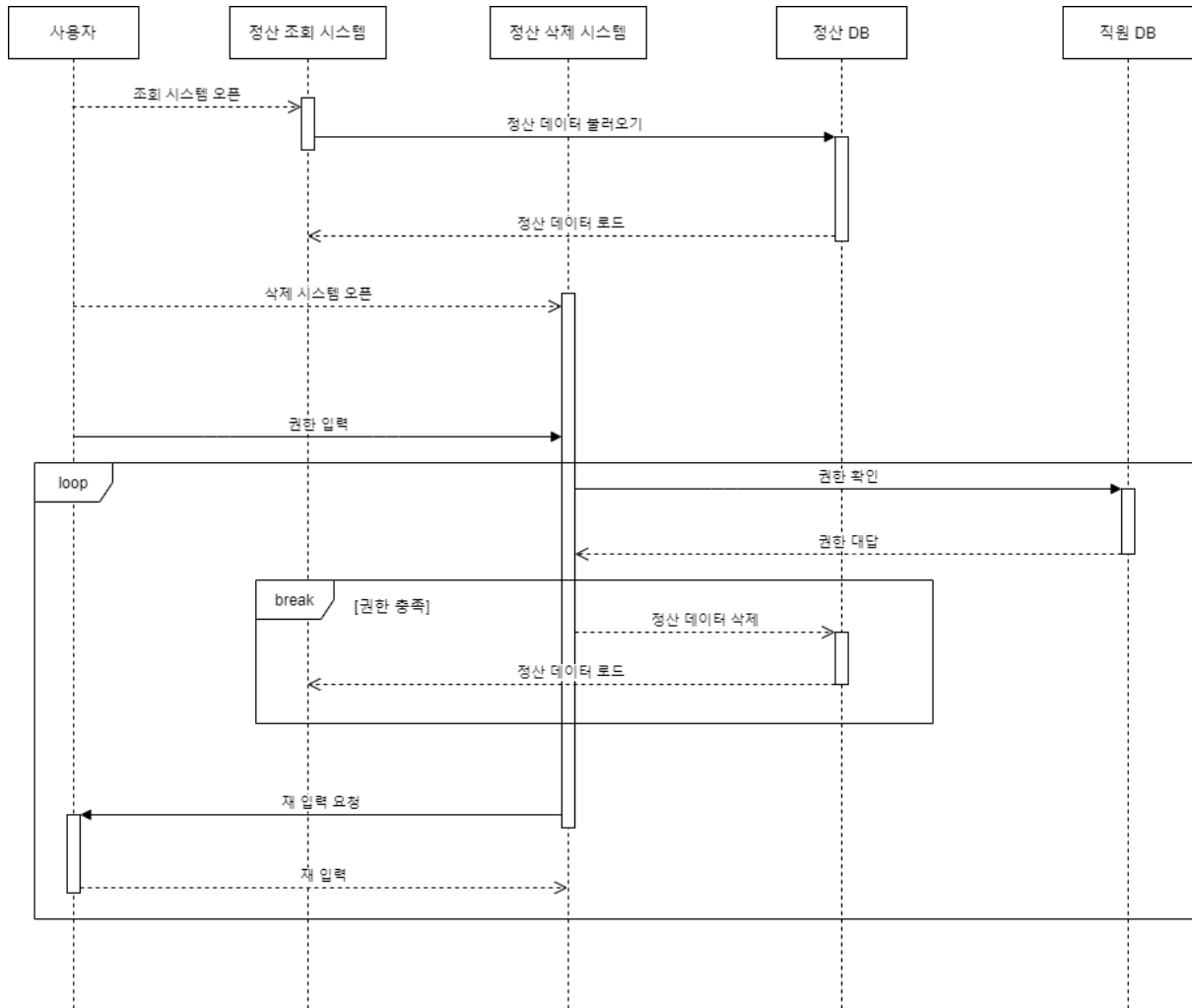


Diagram 11: 정산 데이터 삭제 Sequence Diagram

정산 데이터 삭제 과정은 메뉴 삭제 과정과 유사하다. 조회 시스템에서 선택한 정산 데이터를 삭제하려 하면 삭제 시스템이 열린다. 입력한 권한이 충족되면 데이터베이스에서 해당하는 데이터를 삭제할 수 있다.

3.3. Class Diagram

3.3.1. Overall

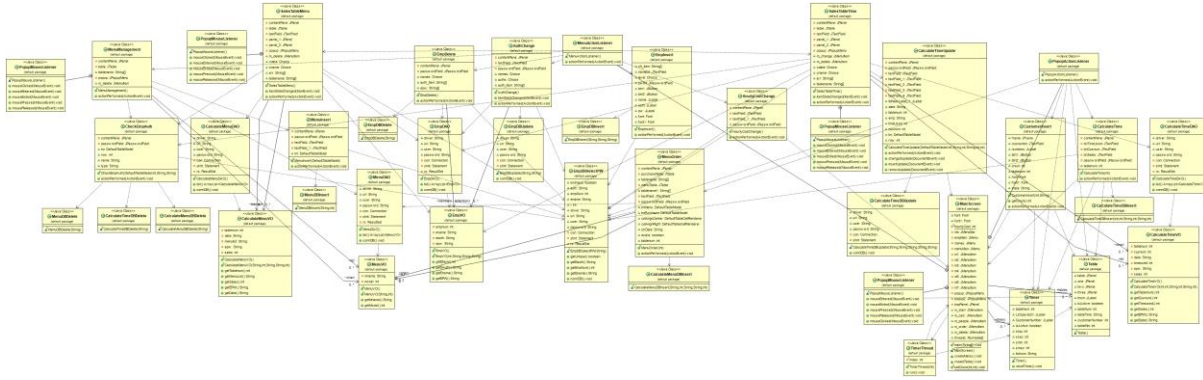


Diagram 12: Overall Class Diagram

제작한 프로그램에서 사용한 모든 클래스들에 대한 클래스 간의 관계를 다이어그램으로 나타내보았다.

3.3.2. MainScreen

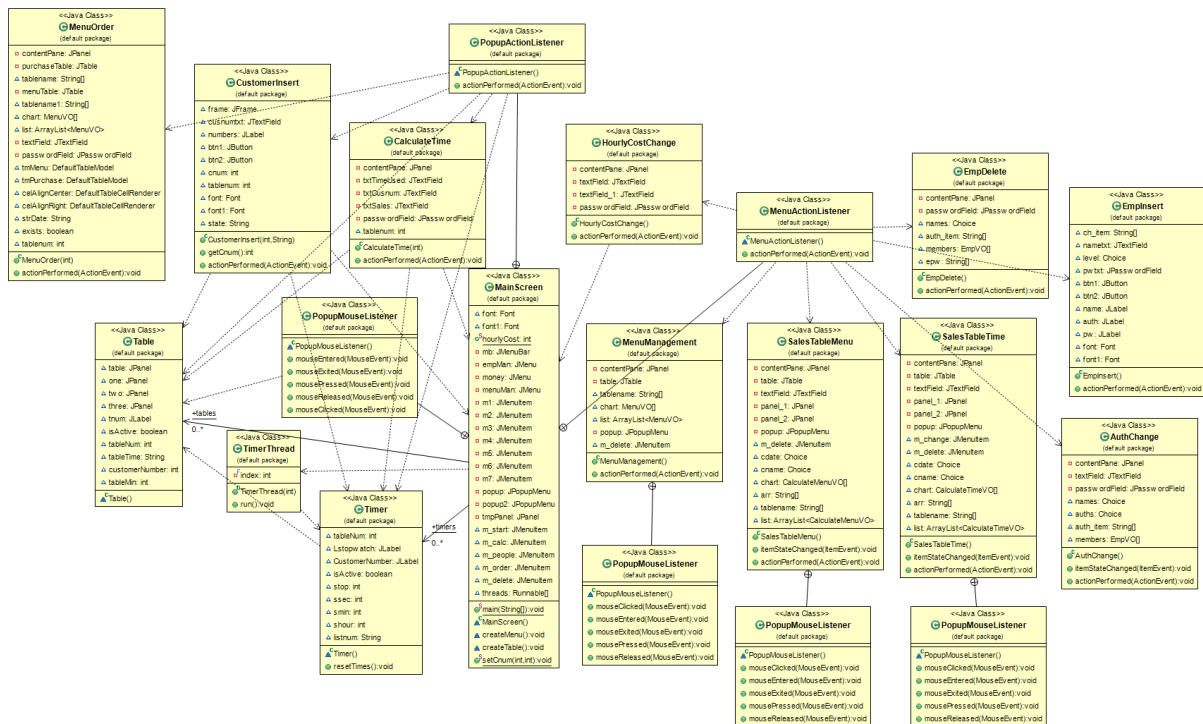


Diagram 13: MainScreen Class Diagram

MainScreen Class는 Timer, TimerThread, Table class를 사용하며 MenuActionListener를 통해 EmpInsert, AuthChange, EmpDelete, SalesTableMenu, MenuManagement, HourlyCostChange,

SalesTableTime Class를 사용하고 있고, PopupActionListener를 통해 MenuOrder, CustomerInsert, CalculateTime Class를 사용하고 있다.

3.3.3. CustomerInsert

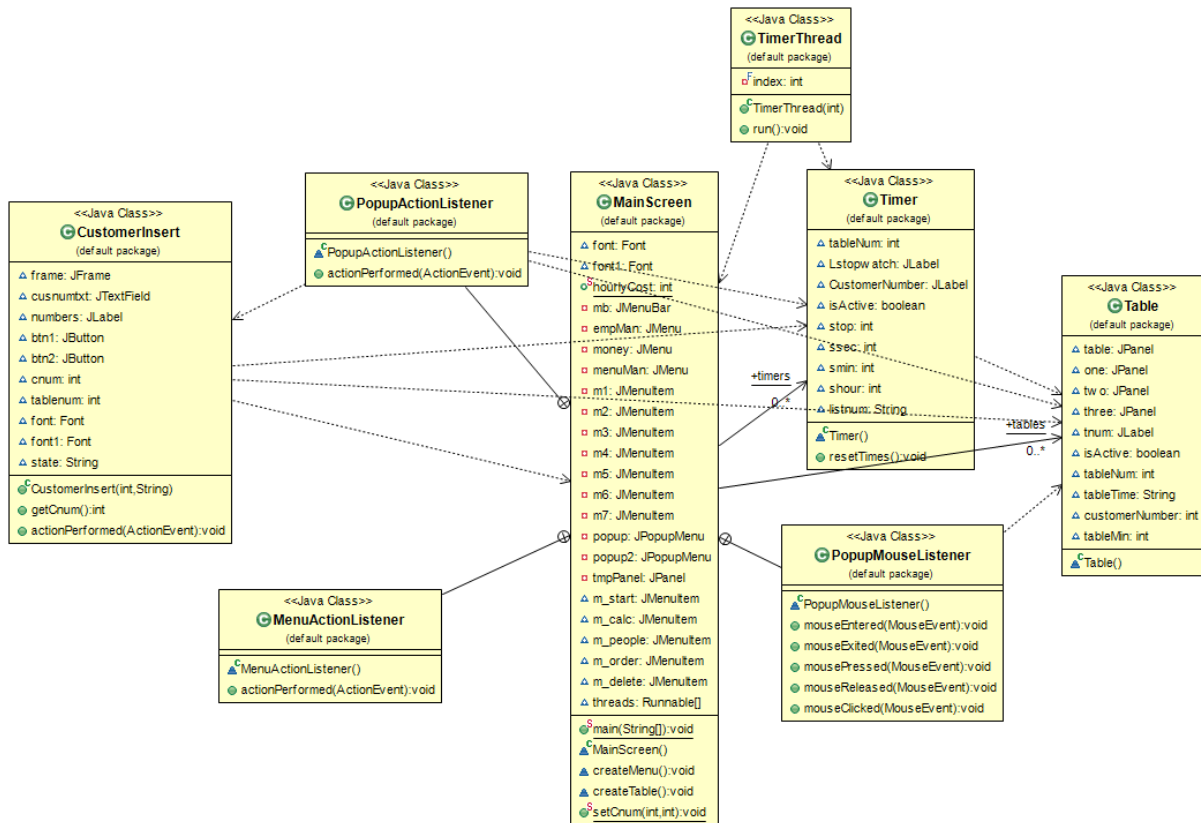


Diagram 14: CustomerInsert Class Diagram

테이블에 앉은 고객의 수를 입력하는 CustomerInsert Class는 제대로 된 값을 입력하면 그 값에 해당하는 Table Class의 값을 변화시킨다. 동시에 Timer Class를 통해 TimerThread를 발동시켜서 각 테이블당 흘러간 시간을 표시한다.

3.3.4. 직원과 관련된 Class Diagram

A. EmplInsert Class Diagram

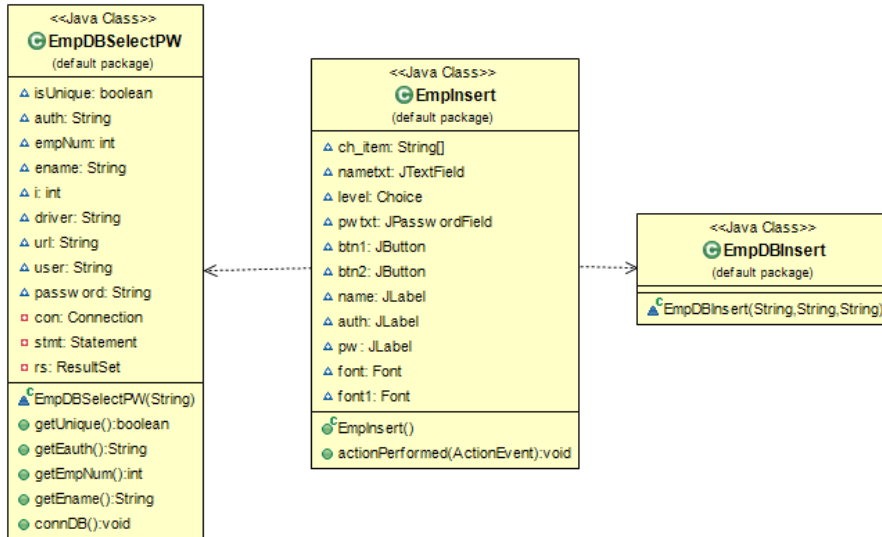


Diagram 15: EmplInsert Class Diagram

EmplInsert Class는 읽어온 값을 데이터베이스에 저장하는 EmpDBInsert Class와 중복되는 값을 확인하는 EmpDBSelectPW Class를 사용하고 있다.

B. AuthChange Class Diagram

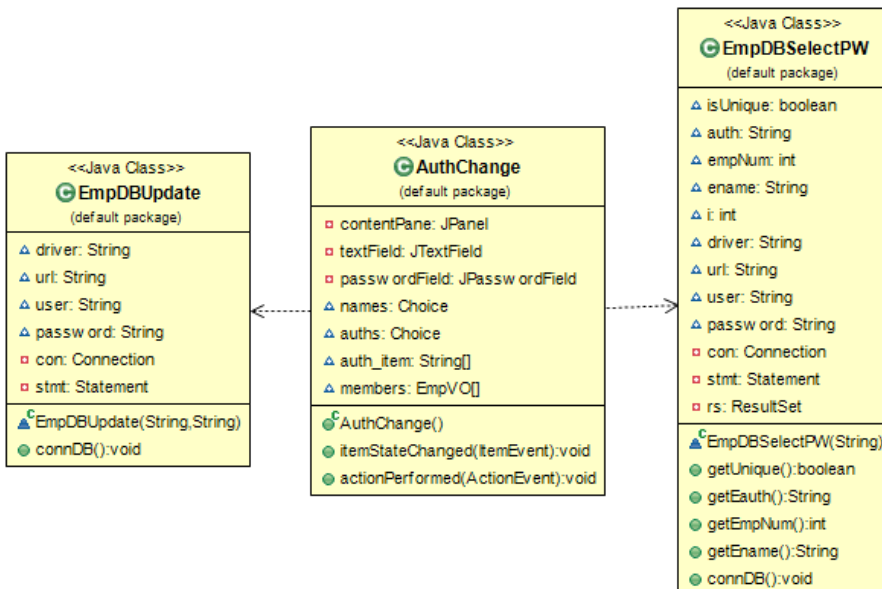


Diagram 16: AuthChange Class Diagram

직원의 권한을 변경하는 AuthChange Class는 데이터베이스에 저장된 직원의 값을 변경하는

EmpDBUpdate Class와 직원 고유 번호로 권한을 확인하는 EmpDBSelectPW Class를 사용하고 있다.

C. EmpDelete Class Diagram

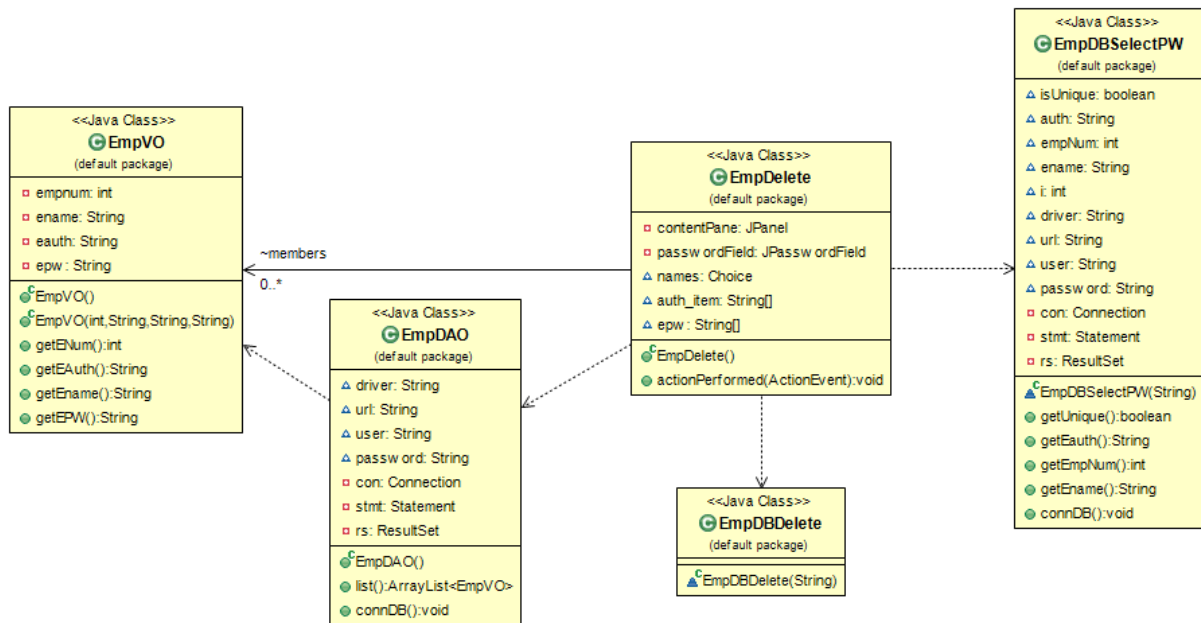


Diagram 17: EmpDelete Class Diagram

3.3.5. 메뉴와 관련된 Class Diagram

A. MenuManagement Class Diagram

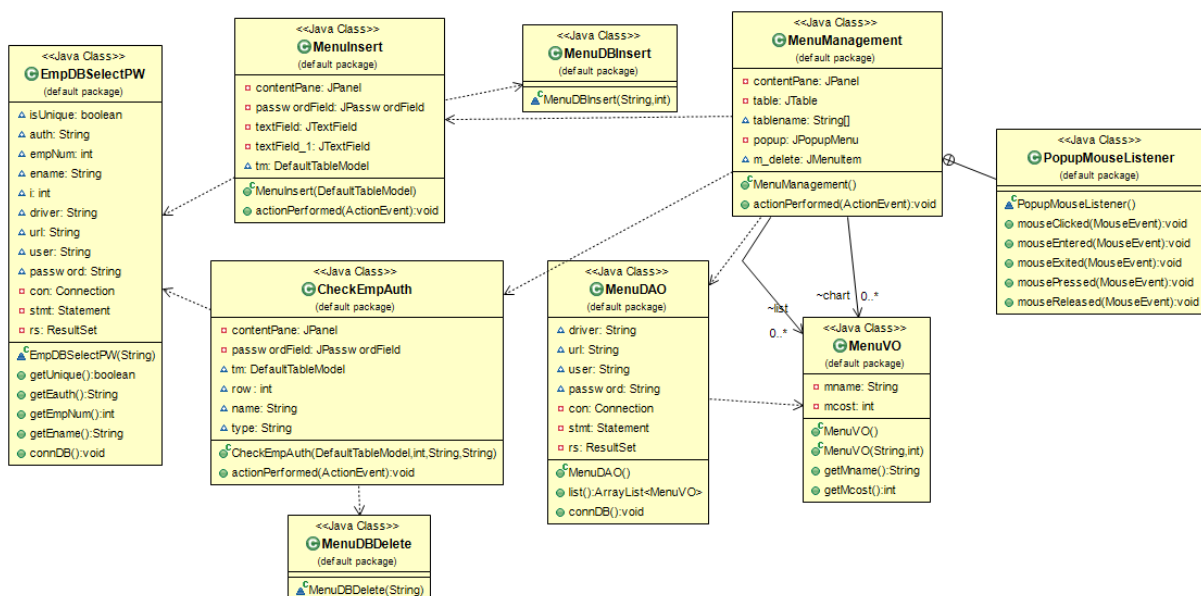


Diagram 18: MenuManagement Class Diagram

메뉴(음료)를 추가 및 삭제할 수 있는 클래스인 MenuManagement Class이다. MenuDAO를 통해 MENU 데이터베이스에 있는 값들을 MenuVO에 집어넣어서 테이블 형태로 표시한다. Menu Insertion과 Deletion이 가능하며, MenuInsert Class를 열어서 입력할 권한이 되는지 EmpDBSelectPW Class로 확인한 후 권한이 되면 MenuDBInsert Class를 통해 MENU DB에 값을 넣는다. Deletion은 CheckEmpAuth Class에서 진행되는데, 마찬가지로 EmpDBSelectPW Class를 통해 권한이 되는지 확인한 후 권한이 되면 MenuDBDelete Class를 통해 MENU DB에서 값을 삭제할 수 있다.

B. MenuOrder Class Diagram

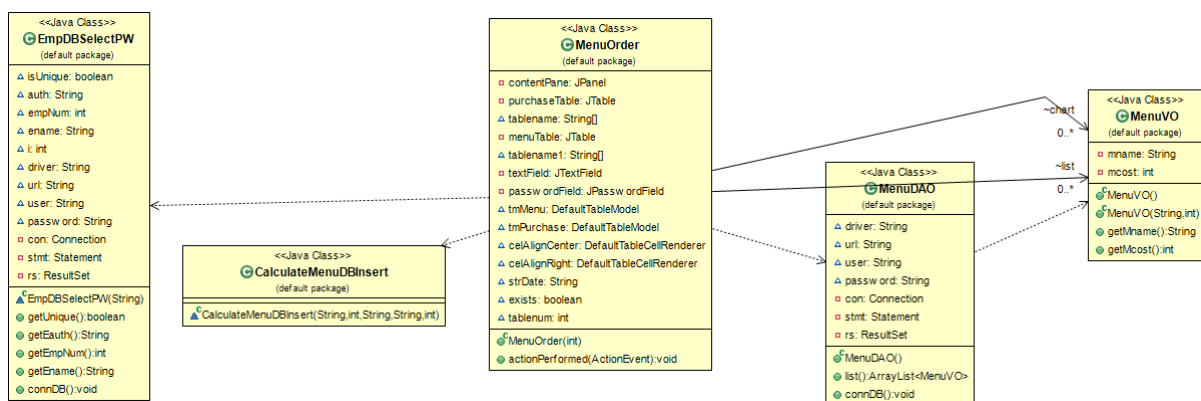


Diagram 19: MenuOrder Class Diagram

TimerThread가 돌아가고 있는 테이블에 한정해서 MenuOrder Class를 사용할 수 있다. MenuOrder Class에서는 MenuDAO 클래스를 통해 MENU DB에서 음료 데이터를 가져와서 MenuVO Class에 저장한다. EmpDBSelectPW Class를 통해 직원이 맞는지 확인한 후 CalculateMenuDBInsert Class를 통해 CALCULATEMENU 데이터베이스에 값을 집어넣는다.

C. SalesTableMenu Class Diagram

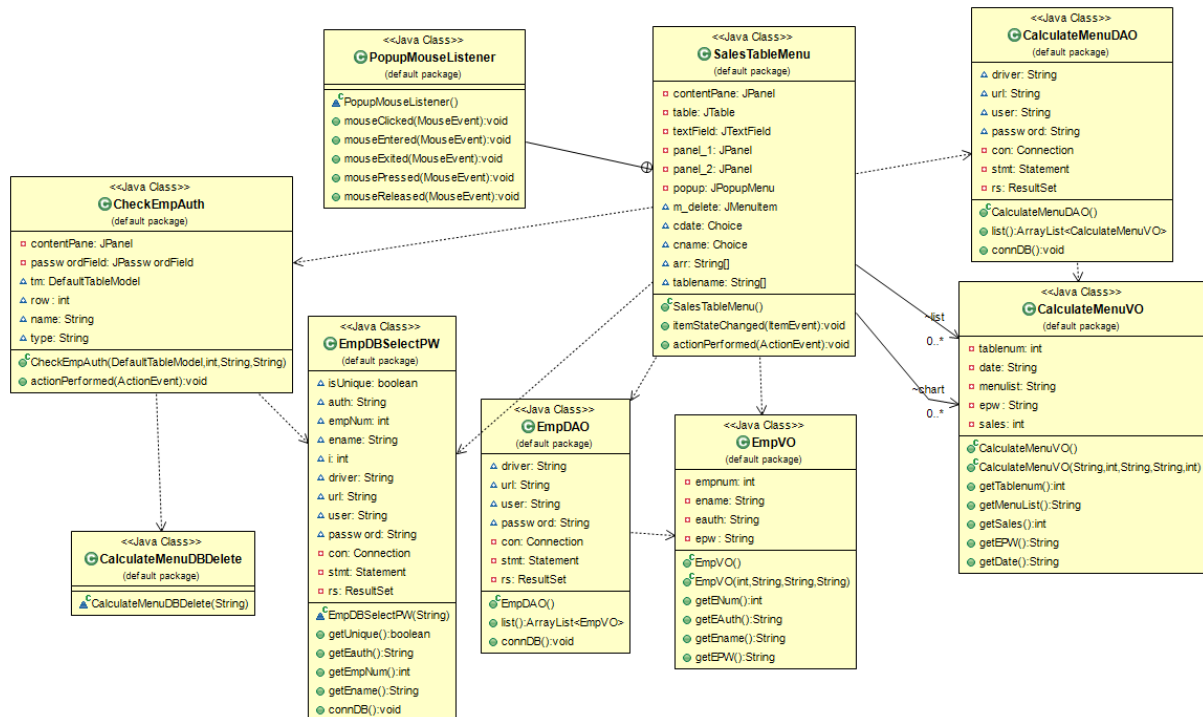


Diagram 20: SalesTableMenu Class Diagram

음료와 관련된 정산 데이터를 조회 또는 삭제할 수 있는 SalesTableMenu Class이다. CalculateMenuDAO와 CalculateMenuVO Class를 통해 CALCULATEMENU DB에 있는 데이터들을 화면 위 테이블에 띄울 수 있다. 날짜 별 정산 기록을 보기 위해 CalculateMenuVO Class를 활용했고, 직원 별 정산 기록을 보기 위해 EmpDAO Class와 EmpVO Class를 활용하였다. 정산 정보 삭제를 위해 CheckEmpAuth Class를 사용했는데, 이 클래스에서 EmpDBSelectPW Class를 통해 권한 정보를 얻어와서 입력한 값의 권한이 충분하면 CalculateMenuDBDelete Class를 발동시켜서 CALCULATEMENU DB에서 해당하는 열의 값을 삭제할 수 있다.

3.3.6. 시간당 요금과 관련된 Class Diagram

A. HourlyCostChange Class Diagram

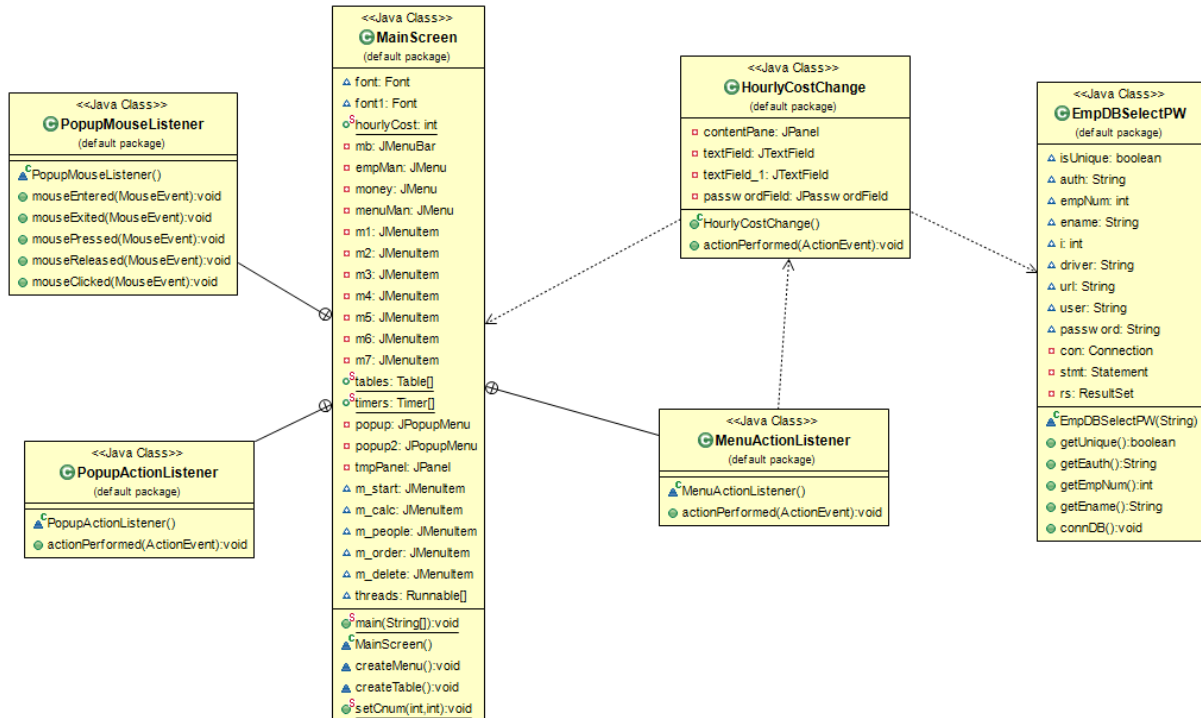


Diagram 21: HourlyCostChange Class Diagram

시간당 요금제를 변경할 수 있는 HourlyCostChange Class이다. EmpDBSelectPW를 통해 적절한 권한을 갖고 있는지 확인한 후 권한이 적절하면 MainScreen Class 안에 있는 hourlyCost라는 int 값을 변경한다.

B. CalculateTime Class Diagram

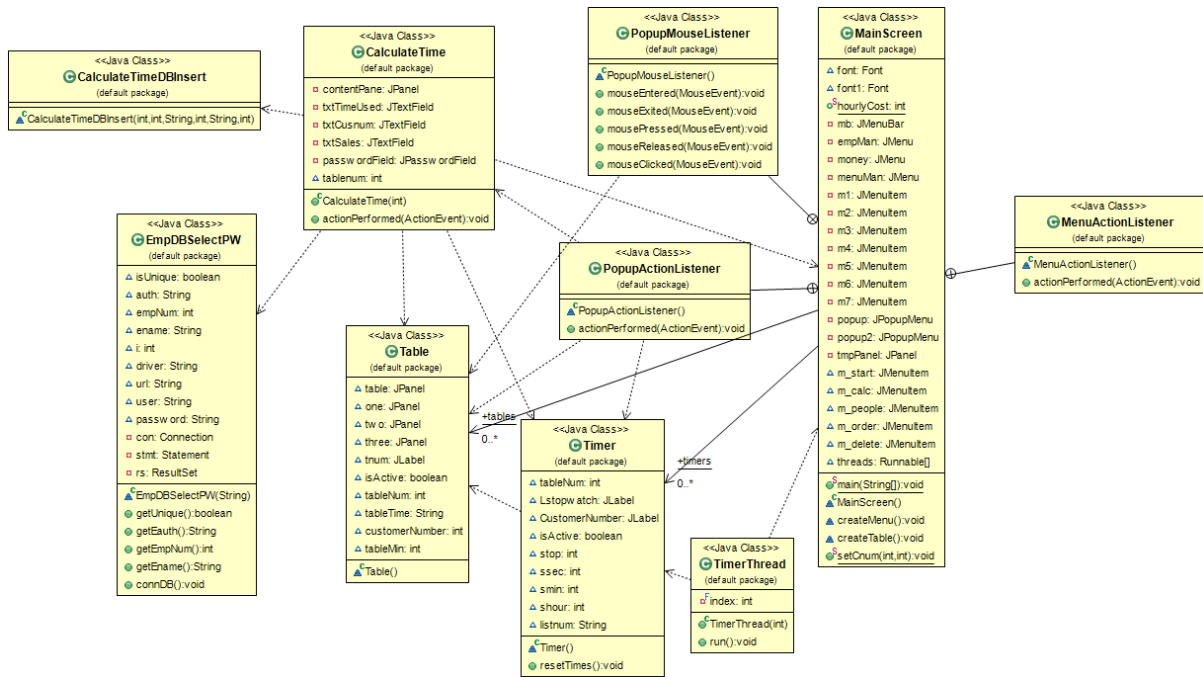


Diagram 22: CalculateTime Class Diagram

시간당 요금을 CALCULATETIME DB에 집어넣는 Class이다. Table과 Timer Class를 통해 현재 테이블 번호, 고객 수, 이용 시간을 받아와서 표시하고, MainScreen의 hourlyCost 변수를 받아와서 현재 사용 요금을 표시한다. EmpDBSelectPW Class를 이용하여 데이터베이스에 값을 넣으려는 사람이 직원이 맞는지 확인한 후 직원이 맞다면 CalculateTimeDBInsert Class를 통해 CALCULATETIME DB에 값을 집어넣는다.

C. SalesTableTime Class Diagram

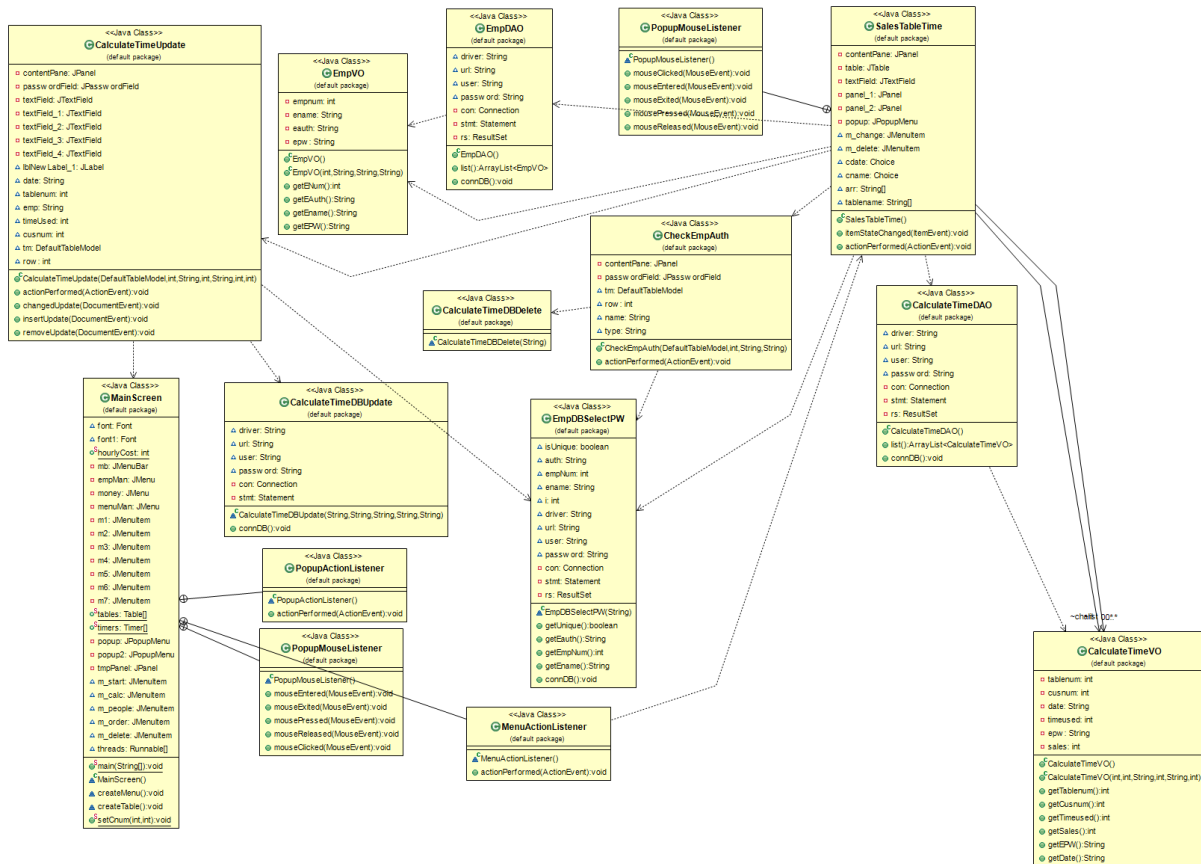


Diagram 23: SalesTableTime Class Diagram

CALCULATETIME DB를 조회, 수정, 삭제할 수 있는 Class이다. 날짜 별 조회를 위해 CalculateTimeVO Class를 사용하였고, 직원 별 조회를 위해 EmpDAO와 EmpVO Class를 사용하였다. 데이터를 수정하기 위해서 CalculateTimeUpdate Class를 사용하는데, 이 Class에서는 MainScreen Class에 있는 hourlyCost 변수를 사용하며, EmpDBSelectPW Class를 이용해 정산 정보 변경 권한이 있는지 확인한다. 권한이 충족되면 CalculateTimeDBUpdate Class를 통해 CALCULATETIME DB에 있는 정보를 수정할 수 있다. 삭제는 CheckEmpAuth Class를 통해 진행되는데, EmpDBSelectPW Class를 통해 삭제할 권한이 충족되는지 확인한 후 권한을 충족하면 CalculateTimeDBDelete Class를 통해 CALCULATETIME DB에 있는 정보를 삭제한다.

4. Result Screen

4.1. MainScreen

A. Intended

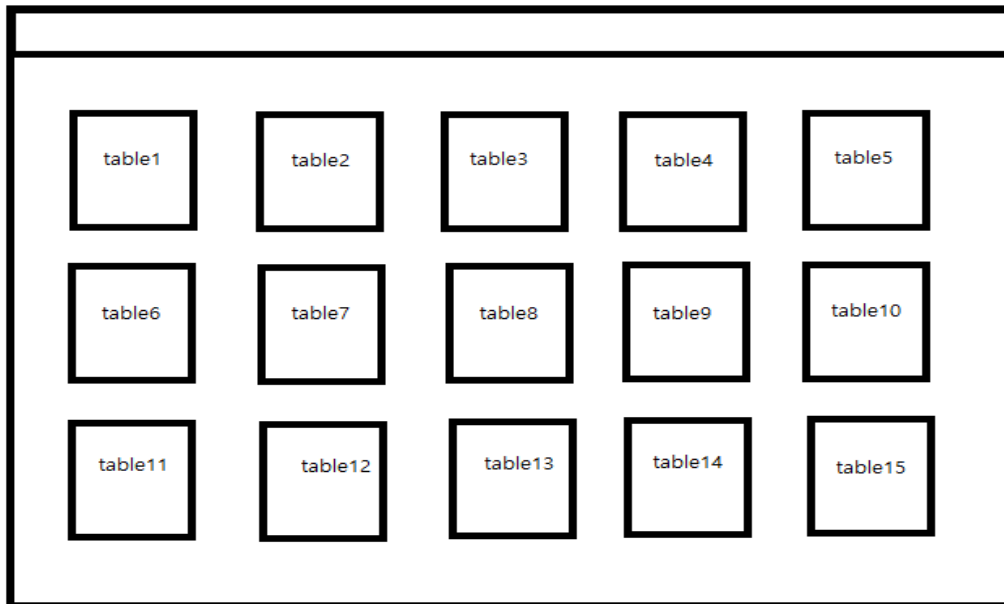


Figure 1: Intended MainScreen

B. Actual



Figure 2: Actual MainScreen

4.2. CustomerInsert

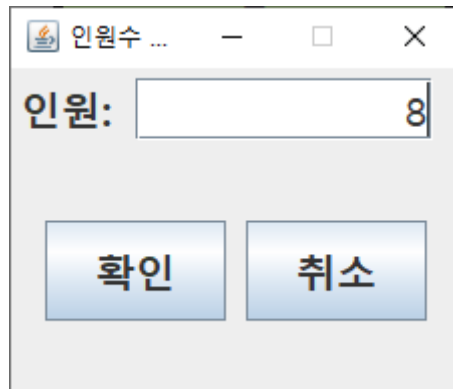
A. Intended



A diagram of the intended CustomerInsert dialog box. It is a rectangular box with a thick black border. At the top, it contains the text "인원수 입력" (Input Number of People). Below this, there is a label "인원:" followed by a textfield placeholder "textfield". At the bottom, there are two buttons: "확인" (Confirm) on the left and "취소" (Cancel) on the right, both enclosed in rectangular boxes.

Figure 3: Intended CustomerInsert

B. Actual



A screenshot of the actual CustomerInsert dialog box. The window title bar shows a small icon, the text "인원수 ...", and standard minimize, maximize, and close buttons. The main content area has a light gray background. It features a label "인원:" followed by a text input field containing the number "8". Below the input field are two buttons with a blue gradient: "확인" (Confirm) on the left and "취소" (Cancel) on the right.

Figure 4: Actual CustomerInsert

4.3. EmplInsert

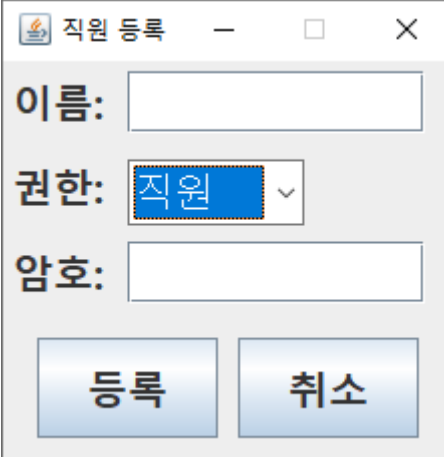
A. Intended



A diagram of the intended EmplInsert form. It is a vertical rectangle with a black border. Inside, the text is as follows: '등록' at the top, followed by '이름: textfield', '권한:choice', '비밀번호:textfield' with 'setechochar' indented below it, and '확인버튼 취소버튼' at the bottom.

Figure 5: Intended EmplInsert

B. Actual

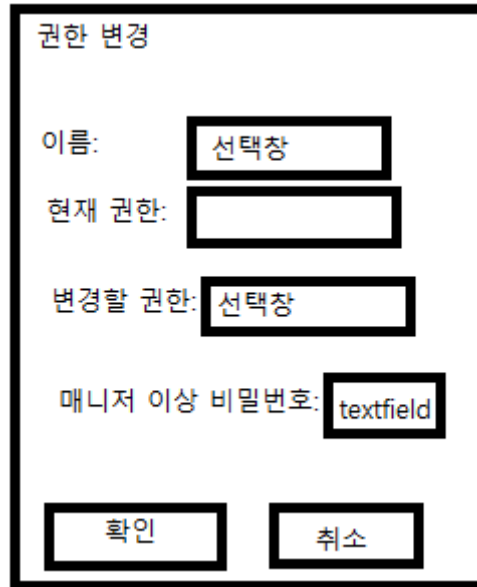


A screenshot of the actual EmplInsert form window. The title bar reads '직원 등록' (Employee Registration). The form contains three input fields: '이름:' (Name) with a text box, '권한:' (Authority) with a dropdown menu showing '직원' (Employee), and '암호:' (Password) with a text box. At the bottom are two buttons: '등록' (Register) and '취소' (Cancel).

Figure 6: Actual EmplInsert

4.4. AuthChange

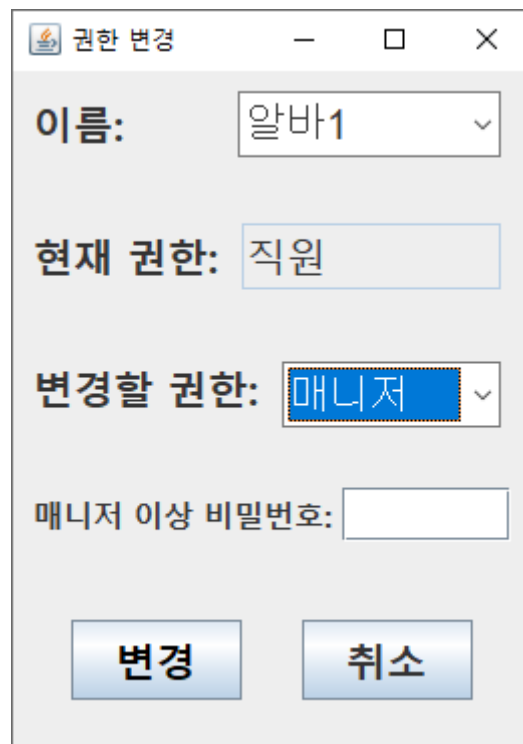
A. Intended



A diagram of a form titled "권한 변경" (Change Authority). It contains four input fields: "이름:" (Name) with a "선택창" (Dropdown) placeholder, "현재 권한:" (Current Authority) with an empty text box, "변경할 권한:" (Authority to Change) with a "선택창" (Dropdown) placeholder, and "매니저 이상 비밀번호:" (Manager or higher password) with a "textfield" placeholder. At the bottom are two buttons: "확인" (Confirm) and "취소" (Cancel).

Figure 7: Intended AuthChange

B. Actual

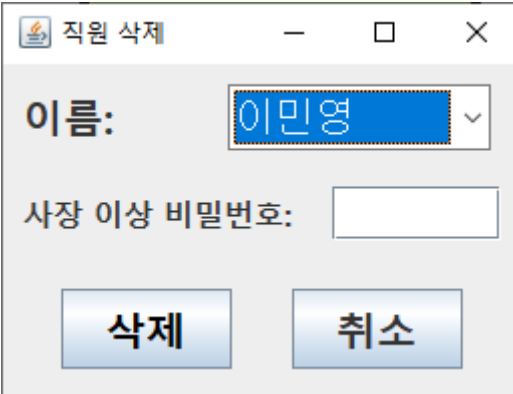


A screenshot of a web application window titled "권한 변경" (Change Authority). The form contains: "이름:" (Name) with a dropdown menu showing "알바1"; "현재 권한:" (Current Authority) with a text box containing "직원"; "변경할 권한:" (Authority to Change) with a dropdown menu showing "매니저"; and "매니저 이상 비밀번호:" (Manager or higher password) with an empty text box. At the bottom are two buttons: "변경" (Change) and "취소" (Cancel).

Figure 8: Actual AuthChange

4.5. EmpDelete

Actual



직원 삭제

이름: 이민영

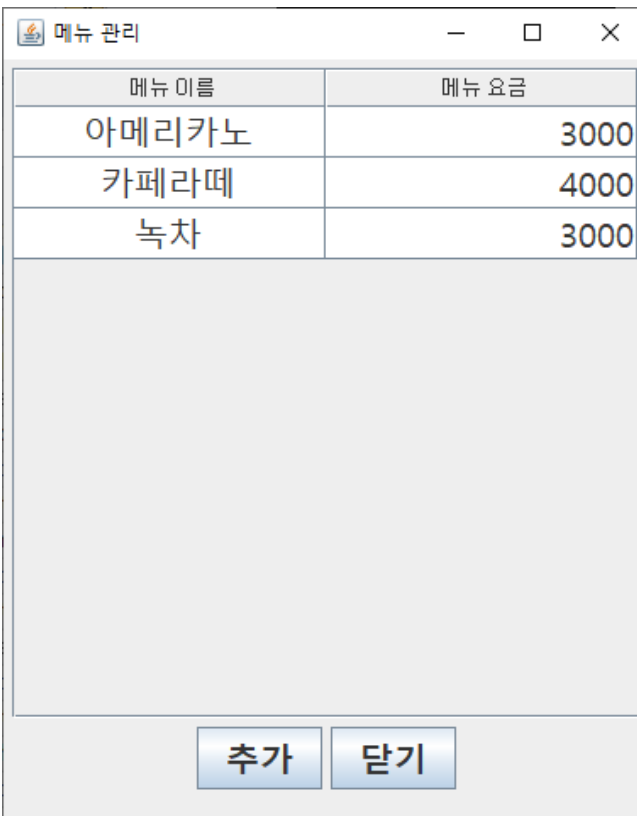
사장 이상 비밀번호:

삭제 취소

Figure 9: Actual EmpDelete

4.6. MenuManagement

Actual



메뉴 이름	메뉴 요금
아메리카노	3000
카페라떼	4000
녹차	3000

추가 닫기

Figure 10: Actual MenuManagement

4.7. MenuOrder

Actual

음료 주문

2020-12-11 09:39:58

Table # 1

메뉴 이름	가격
아메리카노	3000
카페라떼	4000
녹차	3000

추가

제거

메뉴 이름	갯수	합계 가격
아메리카노	2	6000
카페라떼	1	4000
녹차	3	9000

총 요금:

19000

권한 입력:

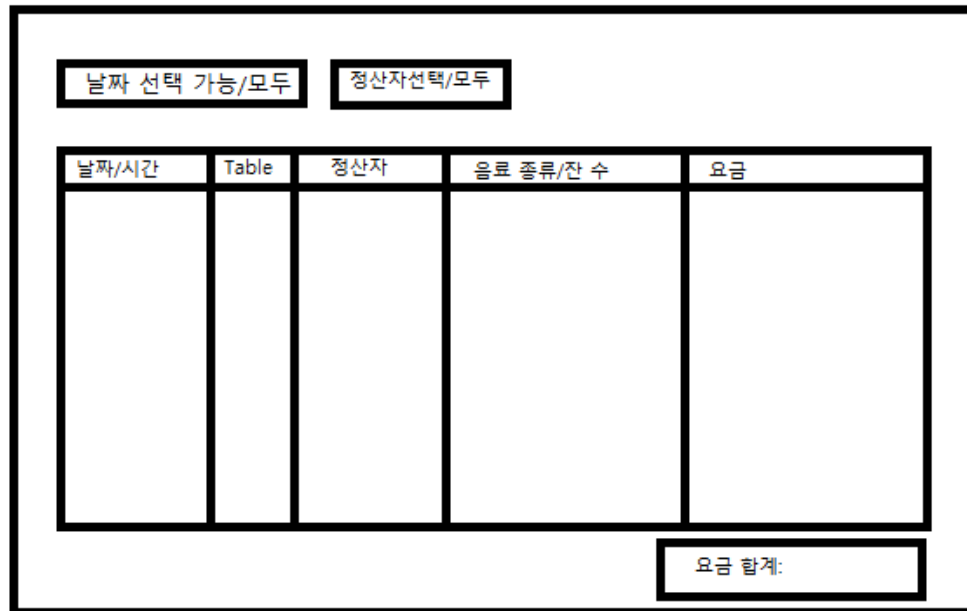
결제

취소

Figure 11: Actual MenuOrder

4.8. SalesTableMenu

A. Intended



The intended UI mockup for SalesTableMenu consists of a main container with a black border. At the top, there are two buttons: '날짜 선택 가능/모두' (Date Selection Available/All) and '정산자선택/모두' (Settlement Selection/All). Below these buttons is a table with five columns: '날짜/시간' (Date/Time), 'Table', '정산자' (Settlement), '음료 종류/잔 수' (Beverage Type/Quantity), and '요금' (Fee). The table body is currently empty. At the bottom right of the container, there is a button labeled '요금 합계:' (Total Fee:).

Figure 12: Intended SalesTableMenu

B. Actual



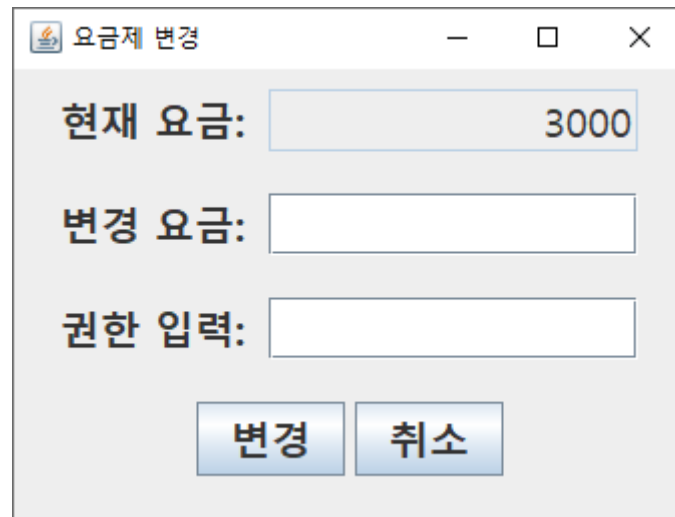
The actual UI screenshot shows a window titled '음료 요금 정산' (Beverage Fee Settlement). It features two dropdown menus: '날짜:' (Date) set to '모두' (All) and '이름:' (Name) set to '모두' (All). Below the dropdowns is a table with the same structure as the intended design. The table contains six rows of data. At the bottom right, there is a label '요금 합계:' (Total Fee:) followed by a text box displaying '104000'.

날짜/시간	Table	정산자	음료 종류/잔 수	요금
2020-11-26 17:06:29	0	이민영	아메리카노-3잔 / 카...	17000
2020-11-30 09:43:44	7	이민영	아메리카노-2잔 / 녹...	13000
2020-12-04 13:58:52	5	이민영	녹차-2잔 / 카페라떼-...	26000
2020-12-04 13:59:15	7	기린	카페라떼-2잔 / 아메...	17000
2020-12-04 13:59:32	11	알바1	카페라떼-3잔 /	12000
2020-12-11 09:39:58	1	이민영	아메리카노-2잔 / 카...	19000

Figure 13: Actual SalesTableMenu

4.9. HourlyCostChange

Actual



요금제 변경

현재 요금: 3000

변경 요금:

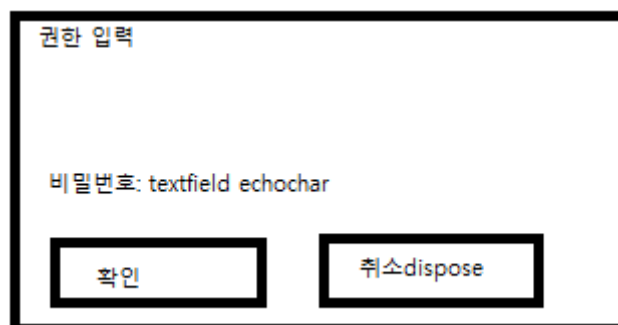
권한 입력:

변경 취소

Figure 14: Actual HourlyCostChange

4.10. CalculateTime

A. Intended



권한 입력

비밀번호: textfield echochar

확인 취소dispose

Figure 15: Intended CalculateTime

B. Actual

정산

이용 시간: 00:08:52

사용 인원: 8 명

합계 요금: 3200 원

권한 입력: ****

결제

취소

Figure 16: Actual CalculateTime

4.11. SalesTableTime

A. Intended

날짜 선택 가능/모두

정산자선택/모두

날짜/시간	Table	정산자	시간(분)	인원수	요금

요금 합계:

Figure 17: Intended SalesTableTime

B. Actual

시간당 요금 정산

날짜: 이름:

날짜/시간	Table	정산자	시간(분)	인원수	요금
2020-11-20 15:44:14	5	기린	0	2	0
2020-11-20 15:44:17	2	기린	0	2	0
2020-11-20 15:44:20	1	기린	0	2	0
2020-11-20 15:44:23	0	기린	0	2	0
2020-11-20 16:31:42	0	알바1	1	8	400
2020-11-20 16:33:31	7	이민영	20	8	8000
2020-11-20 16:49:11	0	기린	60	5	15000
2020-11-30 09:48:23	7	이민영	14	8	5600
2020-12-04 16:24:06	5	이민영	143	8	57200
2020-12-04 16:24:11	7	기린	143	7	50050
2020-12-04 16:24:17	11	알바1	143	3	21450
2020-12-11 09:43:53	1	기린	8	8	3200

요금 합계: 216200

Figure 18: Actual SalesTableTime

5. Index

5.1. Table Index

Table 1: 직원 추가 Table.....	9
Table 2: 테이블 가동 Table.....	10
Table 3: 정산화면 조회 Table	10
Table 4: 정산 Description	11
Table 5: 직원 권한 변경 Description.....	11
Table 6: 메뉴 관리 Description.....	12
Table 7: 직원 삭제 Description.....	13
Table 8: 요금제 변경 Description	13
Table 9: 정산 데이터 수정 및 삭제 Description.....	14

5.2. Diagram Index

Diagram 1: ER Diagram	6
Diagram 2: Relational Schema.....	7
Diagram 3: Use Case Diagram	8
Diagram 4: 직원 추가 Sequence Diagram.....	15
Diagram 5: 직원 권한 변경 Sequence Diagram	16
Diagram 6: 직원 삭제 Sequence Diagram.....	17
Diagram 7: 메뉴 추가 Sequence Diagram.....	18
Diagram 8: 메뉴 삭제 Sequence Diagram.....	19
Diagram 9: 정산 Sequence Diagram	20
Diagram 10: 정산 데이터 수정 Sequence Diagram.....	21
Diagram 11: 정산 데이터 삭제 Sequence Diagram.....	22

Diagram 12: Overall Class Diagram	23
Diagram 13: MainScreen Class Diagram.....	23
Diagram 14: CustomerInsert Class Diagram	24
Diagram 15: EmplInsert Class Diagram	25
Diagram 16: AuthChange Class Diagram	25
Diagram 17: EmpDelete Class Diagram	26
Diagram 18: MenuManagement Class Diagram	26
Diagram 19: MenuOrder Class Diagram.....	27
Diagram 20: SalesTableMenu Class Diagram.....	28
Diagram 21: HourlyCostChange Class Diagram	29
Diagram 22: CalculateTime Class Diagram	30
Diagram 23: SalesTableTime Class Diagram.....	31

5.3. Figure Index

Figure 1: Intended MainScreen	32
Figure 2: Actual MainScreen.....	32
Figure 3: Intended CustomerInsert.....	33
Figure 4: Actual CustomerInsert	33
Figure 5: Intended EmplInsert.....	34
Figure 6: Actual EmplInsert	34
Figure 7: Intended AuthChange.....	35
Figure 8: Actual AuthChange	35
Figure 9: Actual EmpDelete.....	36
Figure 10: Actual MenuManagement.....	36

Figure 11: Actual MenuOrder.....	37
Figure 12: Intended SalesTableMenu.....	38
Figure 13: Actual SalesTableMenu	38
Figure 14: Actual HourlyCostChange.....	39
Figure 15: Intended CalculateTime	39
Figure 16: Actual CalculateTime.....	40
Figure 17: Intended SalesTableTime.....	40
Figure 18: Actual SalesTableTime	41