



편리한 공장을 고민하다



식품 유통 및 사원 관리 시스템

FreshBox





TABLE OF CONTENTS

01 프로젝트 개요



02 기획 및 설계



03 주요 기능 소개



04 시연



05 협업 방식



06 테스트 및 오류 해결



07 결과 및 느낀 점



08 QnA





프로젝트 개요

개발 동기 (WHY?)

기존의 불편함 개선

- 기존 출퇴근 기록 방식(수기)의 불편함 개선
- 사용자의 편의를 위해 입출고 내역을 한 눈에 보고 관리할 수 있는 시스템 구축



개발 목표 (WHAT?)

사원 관리, 입출고 관리

- 회원가입, 로그인, 로그아웃 등 기본적인 회원정보 관리 기능
- 자사 공장에서 유통하는 상품 정보 관리 기능
- 입출고 내역을 보여주고, 사원의 Confirm을 받아 입출고 처리하는 기능



주요 기능 (HOW?)

- 회원가입, 로그인, 로그아웃
→ 사원과 관리자를 분리
- 상품 등록
- 입출고 내역 확인
- 사용자 편의성을 고려한 UXUI



사용자 (WHO?)

- (쇼핑몰/유통 회사 등의)
- 공장 운영자
 - 공정 관리 사원



기존 시스템을 개선하기 위해 공장 내 사원 관리와 입출고 관리를 지원하는 시스템



기획 및 설계 - User Interface



ID

PW

ID/PW찾기

회원가입

로그인

로그인 화면

DB와 연결

DB에 등록된 ID/PW에 한해서 로그인
승인하기

'ID/PW 찾기' 또는 '회원가입' 단계로 이
동 가능

로그인이 정상적으로 처리되어야만
Main 화면으로 이동 가능

모든 기능과 연동

회원 정보 출력



기획 및 설계 - User Interface

회원가입

아이디*	<input type="text"/>
비밀번호*	<input type="password"/>
비밀번호확인*	<input type="password"/>
이름*	<input type="text"/>
생년월일*	<input type="text"/>
휴대폰 번호*	<input type="text"/>
이메일*	<input type="text"/>
입사일(선택)	<input type="text"/>

회원가입

취소

회원가입 화면

상세한 유효성 검사

각 칸에 디테일한 유효성 검사가 시행됨

- 아이디 중복 검사
- 비밀번호 자릿수 및 특수문자 조합
- 생년월일 정상범위 검사 등

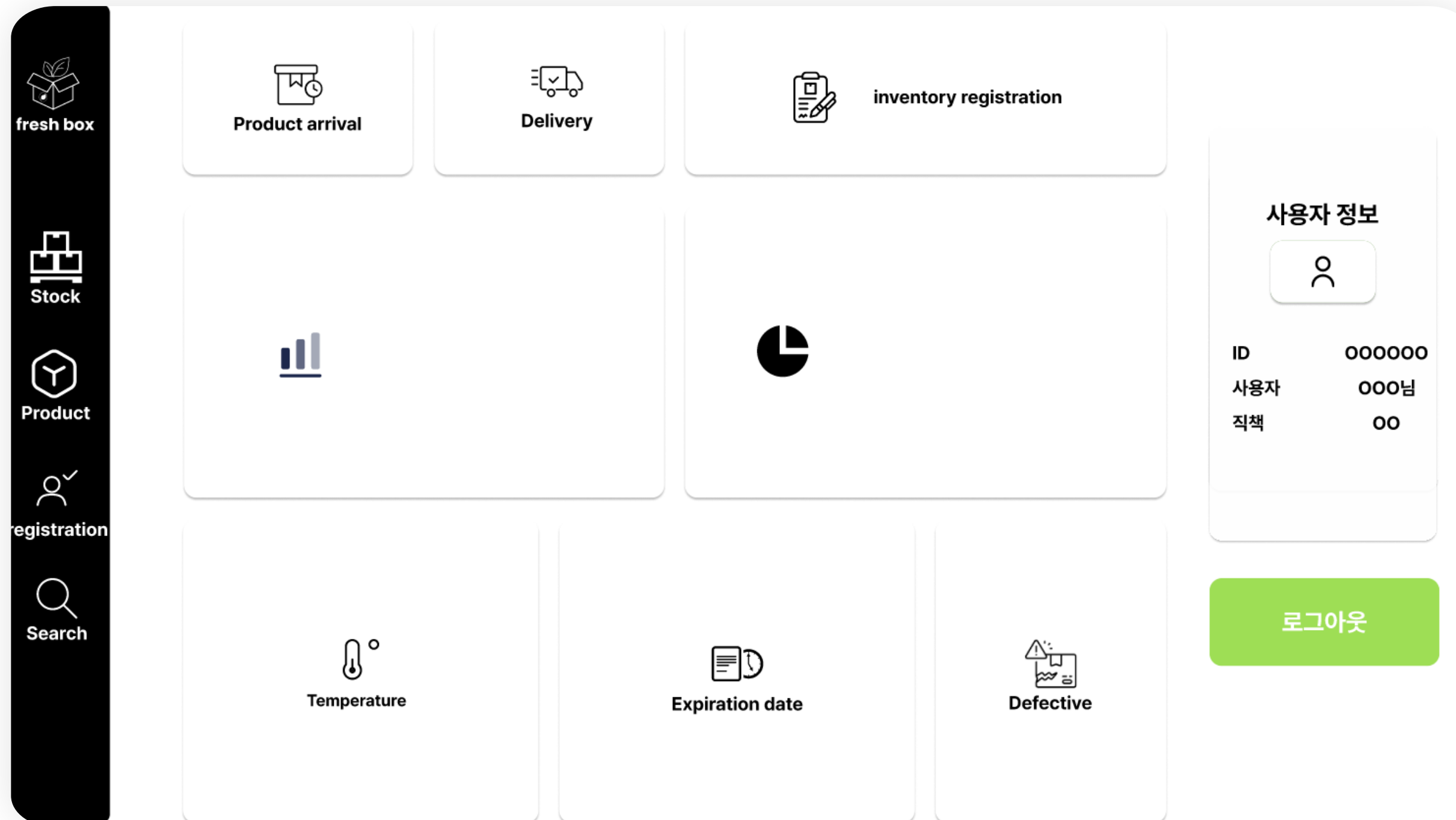
각 칸의 유효성 검사가 통과됐을 경우에
만 회원가입 진행됨

유효성 검사

DB에 신규 회원 등록



기획 및 설계 - User Interface



MAIN 화면

모든 기능과 연동

한 Window 화면에 여러 UserControl 화면을 연동하도록 기획했습니다.

클릭 시 해당하는 View가 화면에 출력됩니다.

우측 화면에는 현재 로그인한 회원 정보가 출력됩니다.

[모든 기능과 연동](#)[회원 정보 출력](#)



기획 및 설계 - User Interface

Product registration

수기로 입력해주세요

상품명을 입력해주세요

재고를 입력해주세요

- ☒ 냉동
- ☐ 냉장
- ☐ 실온

품목을 선택 해주세요 ▼

등록



fresh box

상품 등록 화면

유효성 검사 후 등록

각 칸에 유효성 검사가 시행됨

- 상품명 중복 검사
- 바코드 자릿수 검사 등

각 칸의 유효성 검사가 통과됐을 경우에만 상품 등록 처리됨

유효성 검사

DB에 새 상품 등록



기획 및 설계 - User Interface

상품명

수량

입고

입고 주문 ID

상품명

입고 수량

창고

입고 요청 내역 화면

입고 요청 내역 출력

공장 측에서 주문한 입고 요청 건에 대한 목록이 출력됨

입고 요청 사항과 실제 입고된 내용이 같은지 확인 후 입고 처리

주문 내역과 입고 정보 불일치 시 반송 처리 가능

유효성 검사

입고 처리



기획 및 설계 - MVVM 구조도

```
FreshBox/
├─ Assets/
├─ Converter/
├─ Database/
├─ DTOs/
├─ Enums/
├─ Interfaces/
├─ ★ Models/           ← Model
├─ Repository/
├─ Services/
├─ ★ ViewModels/       ← ViewModel
├─ ★ Views/            ← View
└─ Dependencies/       (Visual Studio에서 자동 생성되는 의존성)
```

MVVM 구조도

폴더	역할
Models	데이터 구조 정의
ViewModels	UI와 Model을 연결하는 로직
Views	XAML UI 파일들
Repository / Services	DB 및 비즈니스 로직
DTOs / Enums / Interfaces	데이터 전달 및 상호작용 정의
Converter	바인딩 값 변환



CHAPTER 02 🔍

기획 및 설계 - Entity Relation Diagram

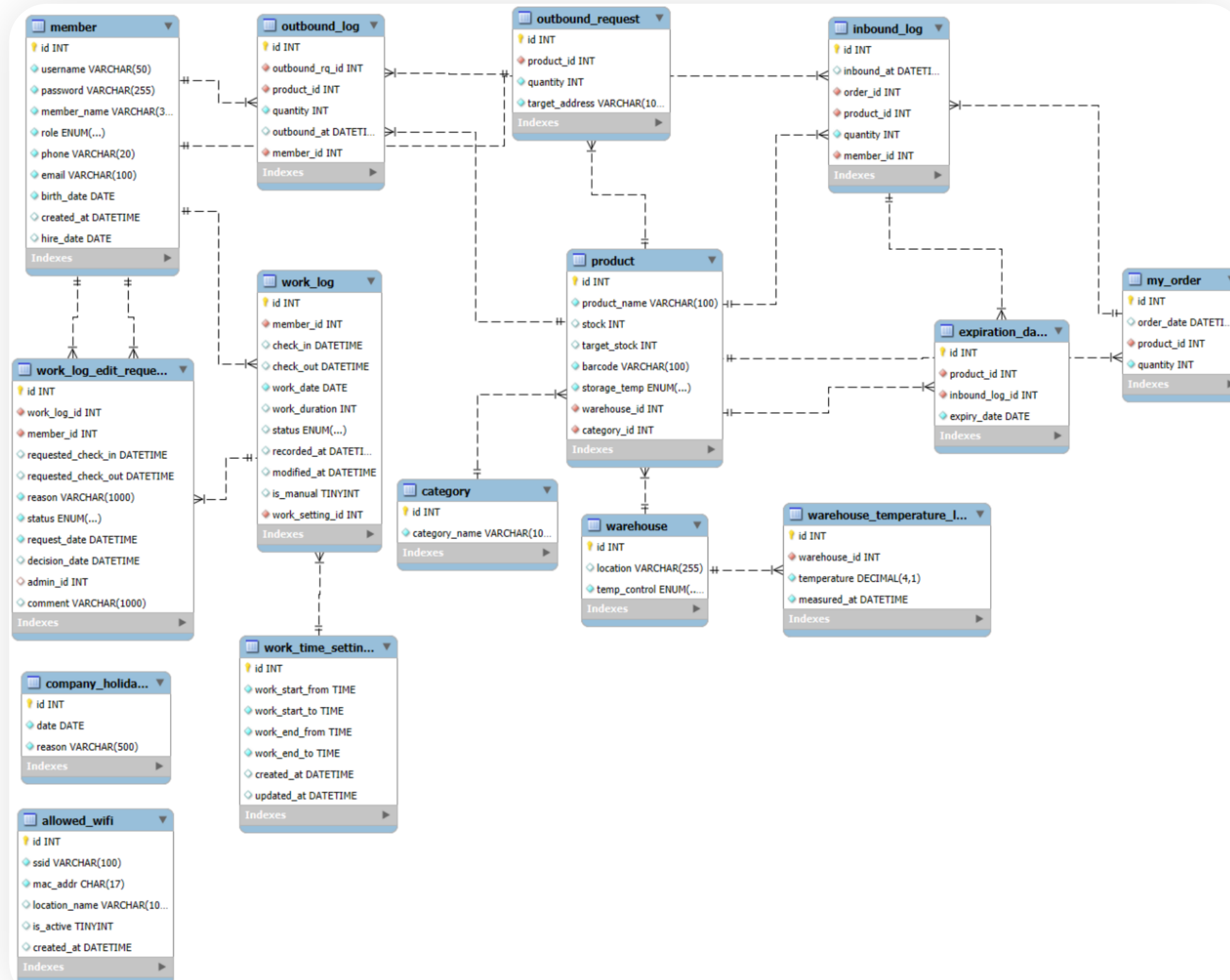
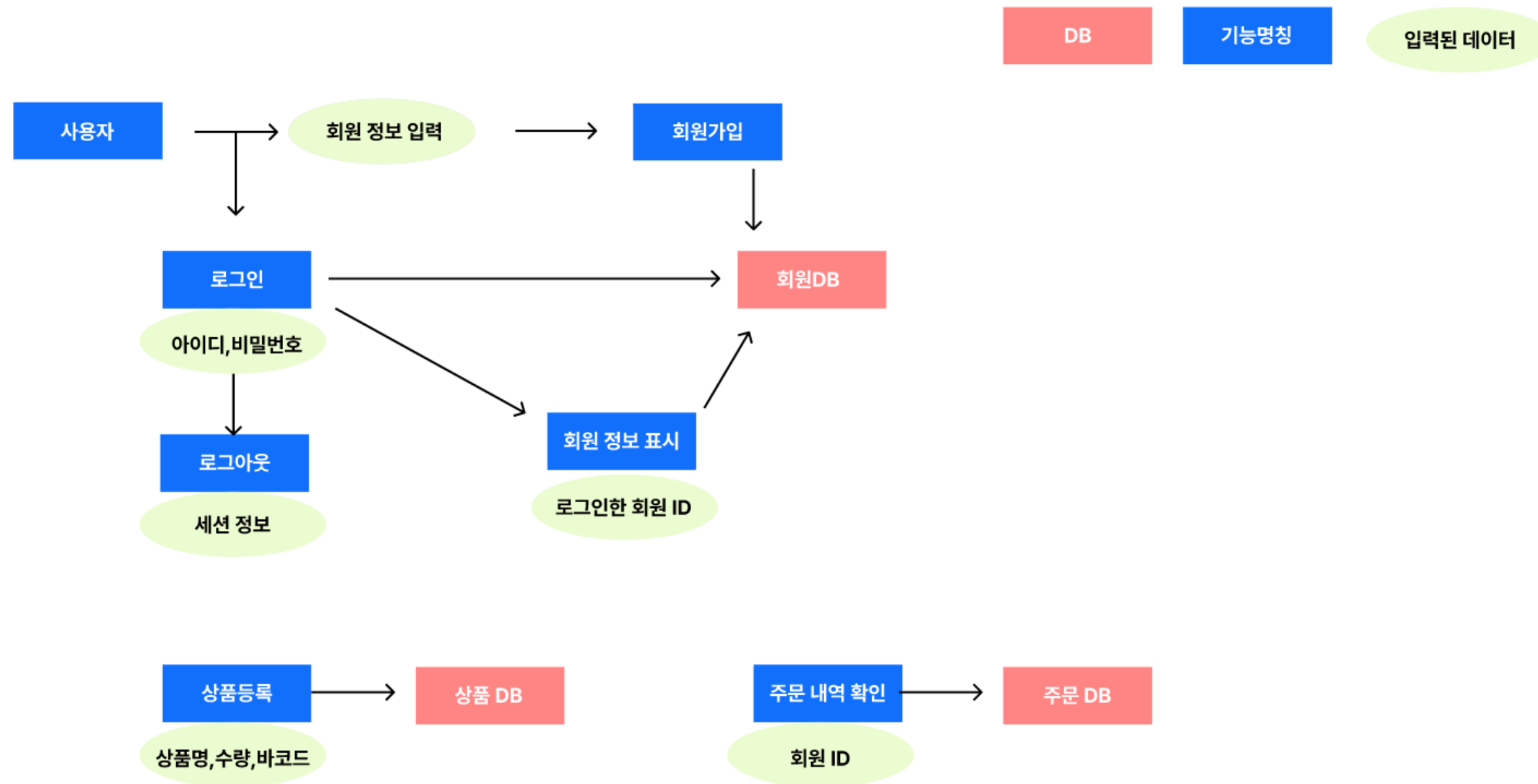


TABLE 목록

- member
- category
- warehouse
- product
- work_log
- my_order
- inbound_log
- expiration_date
- outbound_request
- outbound_log
- warehouse_temperature_log
- work_time_settings
- work_log_edit_requests
- company_holidays
- allowed_wifi



기획 및 설계 - Data Flow Diagram





프로젝트 주요 기능

POINT. 01

회원 가입

- 중복 검사
 - 유효성 검사
 - 비밀번호 암호화
- DB로 INSERT

아이디*

ItsMe

✕ 6~12자의 영문 대소문자와 숫자만 사용할 수 있습니다.

아이디*

Mari1004

☒ 사용할 수 있는 ID입니다.

비밀번호*

●●●●●●

비밀번호는 10자 이상 20자 이하로 입력해주세요.

비밀번호*

●●●●●●●●●●

영문, 숫자, 특수문자 중 최소 2가지 조합이 필요합니다.

비밀번호*

●●●●●●●●●●

☒ 사용 가능한 비밀번호입니다.

비밀번호 확인*

●●●●●●●●

비밀번호가 일치하지 않습니다.

생년월일*

2

공백을 포함할 수 없습니다.

생년월일*

생년월일을 입력해주세요. (예시) 19980801

생년월일*

199

년월일 형식의 8자리로 입력해주세요. (예시)

생년월일*

19950492

유효하지 않은 생년월일입니다.

휴대폰 번호*

010-

휴대폰 번호는 숫자만 입력해주세요. 하이픈(-), 공백은

회원가입

아이디*

Mari1004

☒ 사용할 수 있는 ID입니다.



CHAPTER 03 🔍

프로젝트 주요 기능

POINT. 02

로그인 & 로그아웃

- 아이디 & 비밀번호 확인
- 로그인에 성공하면
홈화면으로 전환
- 홈화면에 현재 로그인한
사용자 정보 표시
- 로그아웃 클릭하면
로그인 화면으로 전환



ID Mari1004

PW ●●●●●●●●

ID/PW 찾기

회원가입

로그인

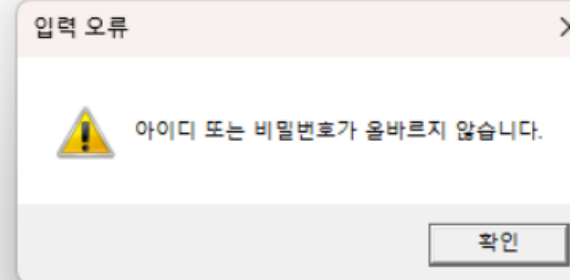
ID hellobox

PW ●●●●●●●●

ID/PW 찾기

회원가입

로그인



사용자 정보



ID Mari1004

사용자 최마리

직책 Employee

로그아웃



CHAPTER 03



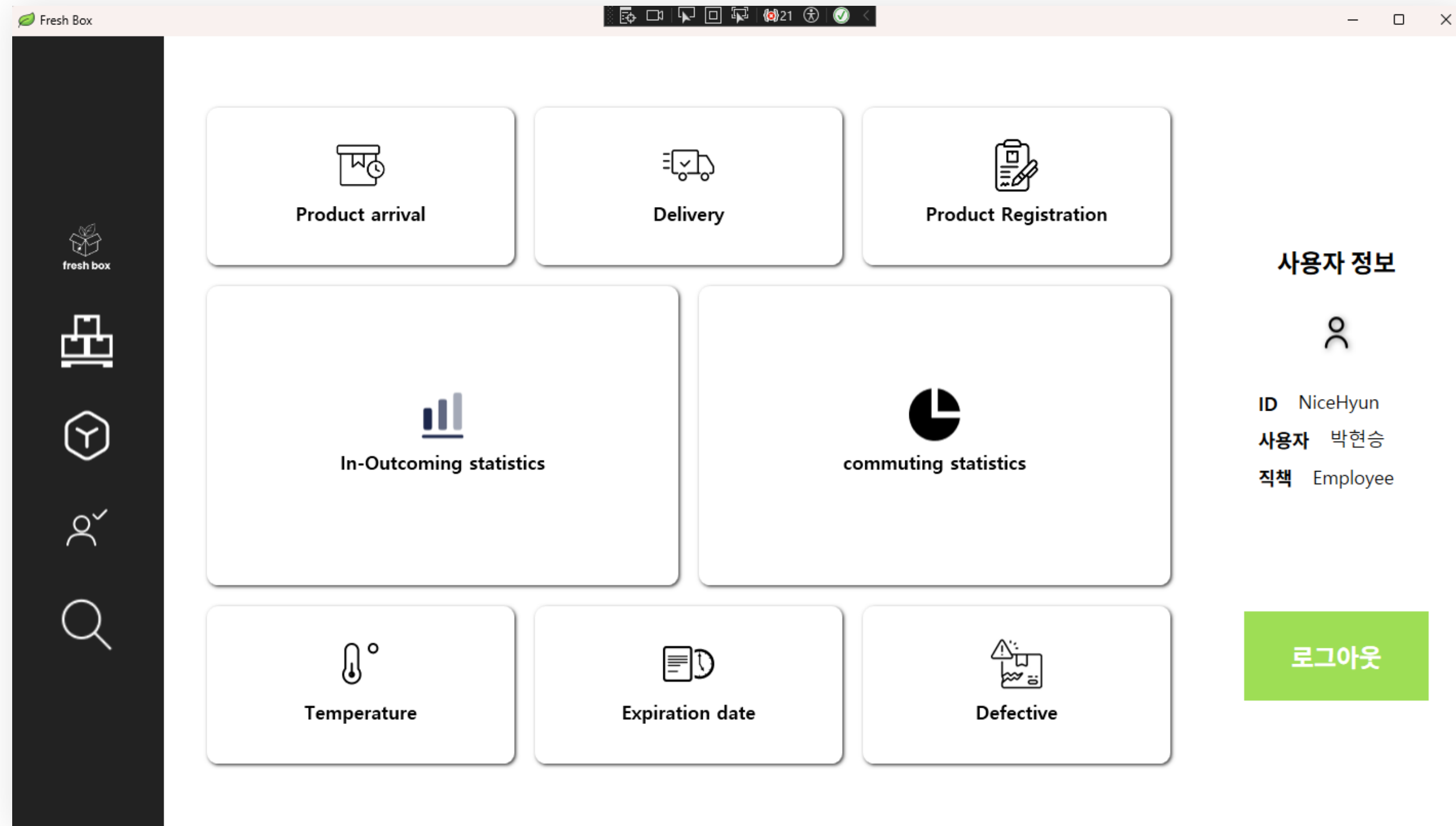
프로젝트 주요 기능

POINT. 03

메인 화면의 화면 전환

- 단일 MainWindow에서 콘텐츠를 동적으로 전환하는 페이지 전환 방식
- 구현 방식

1. INavigationService 인터페이스 정의
2. NavigationService 클래스 구현
3. Singleton으로 관리





프로젝트 주요 기능

POINT. 04

상품 등록

- DB의 카테고리 정보 화면에 바인딩
- 사용자가 입력한 상품 정보의 유효성 체크
- 상품 등록



Product registration

상품명

새우깡

목표 재고량

450

상품 바코드

a12324533

보관 장소

- ☒ 실온
☐ 냉장
☐ 동동

품목

과자

등록





CHAPTER 03



프로젝트 주요 기능

POINT. 05

입고 관리

- 공장 측에서 발주한 주문 내역 화면에 표시
- 직원이 선택한 주문과 입력한 입고 내용을 비교
 - 입고 내용이 다르면 반품 처리
 - 일치하면 입고 처리

Fresh Box

상품명: 수량: 입고 ↺

Id	Orderdate	ProductId	Quantity
1	1/5/2025 2:39:18 AM	9	46
2	4/25/2025 1:39:38 AM	29	94
3	1/27/2025 1:43:48 AM	16	73
4	3/17/2025 9:13:37 AM	6	100
5	2/3/2025 6:28:44 AM	22	27
6	3/27/2025 3:02:06 PM	25	8
7	4/20/2025 4:49:18 PM	26	87
8	1/11/2025 10:02:26 AM	6	21
9	3/14/2025 7:51:03 PM	11	68
10	1/17/2025 8:29:38 PM	9	16
11	5/6/2025 1:31:26 AM	20	57
12	4/25/2025 5:16:57 AM	22	23
13	3/20/2025 4:18:51 PM	1	61
14	3/6/2025 11:00:49 PM	22	53
15	2/15/2025 6:18:14 AM	29	73
16	4/9/2025 9:50:11 AM	28	66
17	3/30/2025 10:53:58 AM	30	40
18	1/8/2025 2:14:50 PM	21	46
19	4/4/2025 4:41:51 AM	13	85
20	5/1/2025 9:25:23 AM	9	20
21	1/20/2025 6:26:58 PM	18	89
22	2/22/2025 4:11:41 AM	27	59
23	4/9/2025 1:07:20 PM		
24	4/6/2025 4:52:12 AM		
25	4/17/2025 10:52:39 P		
26	5/6/2025 5:51:35 AM		
27	3/18/2025 12:35:46 P		
28	4/2/2025 6:50:33 PM		
29	2/21/2025 12:25:12 P		
30	2/5/2025 3:36:24 AM		

상품명: 사과 수량: 입고 ↺

Id	Orderdate	ProductId	Quantity
1	1/5/2025 2:39:18 AM	9	46
2	4/25/2025 1:39:38 AM	29	94
3	1/27/2025 1:43:48 AM	16	73
4	3/17/2025 9:13:37 AM	6	100
5	2/3/2025 6:28:44 AM	22	27
6	3/27/2025 3:02:06 PM	25	8
7	4/20/2025 4:49:18 PM	26	87

상품 불일치

상품이 다릅니다! 그대로 입고 처리하시려면 Yes를, 반품 처리하시려면 No를 클릭하세요.

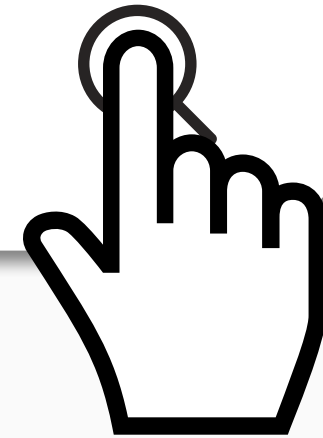
예(Y) 아니오(N)



CHAPTER 04 🔍



시연하러 가기





협업 방식



GitFlow 브랜치 전략

main ← 최종 제품 (배포용, 항상 안정적)
└─ develop ← 개발 통합본
 └─ feature/{기능이름} ← 기능별 작업 브랜치



Pull Request 활용

1. 개인 feature branch에서 자유롭게 작업
→ develop으로 PR 요청
2. develop이 업데이트될 때마다 팀에 전달
→ 각자 pull 받아 개인 브랜치에 merge



협업 방식

역할 분담

박현승	발표, 디자인(로고 제작, 와이어프레임), 화면 설계, [Front-end] UXUI 디자인
서예경	리더, 프로젝트 생성 / 버전 관리(GitHub) / 발표 자료 PPT 작성, [Back-end] 상품 & 재고 관리
최마리	산출물 문서 작성 및 관리 (Notion, Readme.md, 일정 정리, 요구 사항 정의서 등), [Back-end] 회원 & 출퇴근 관리

개발 일정

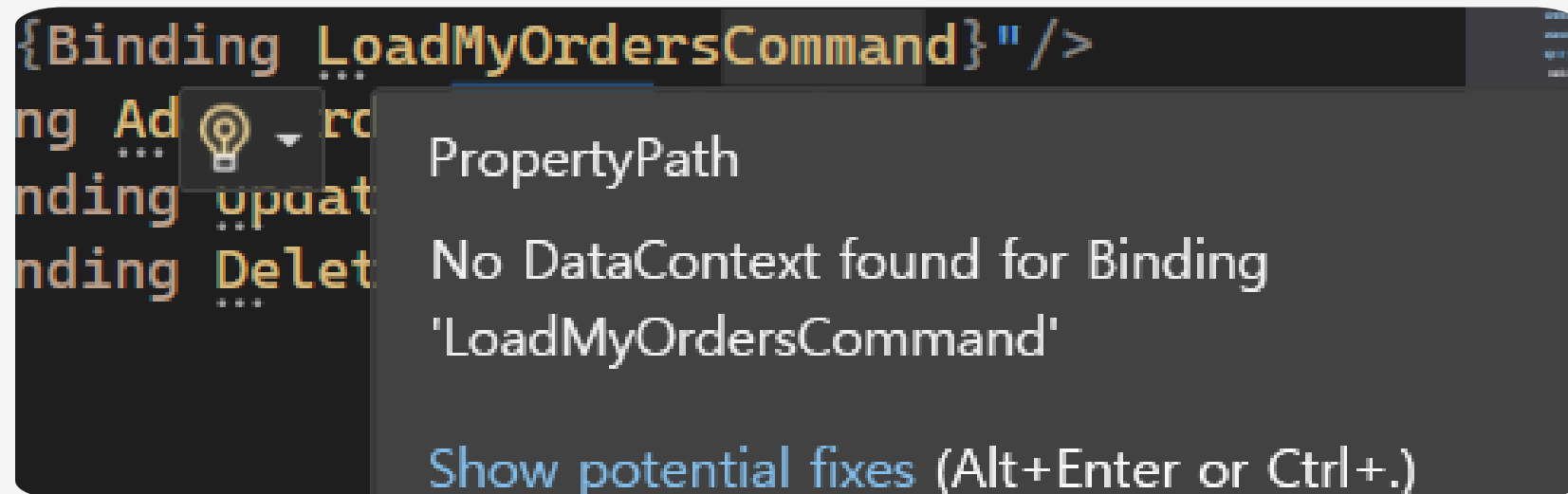
18	19	20	21	22	23	24 프로젝트 시작 - 프로젝트 설계
25	26 기능 구현	27	28 초기 점검	29	30	31
01	02	메모: 5/26(월) : DB Table 설계, MVVM 패턴 적용 조언 받기, 설계한 폴더 구조로 프로젝트 생성 뒤 GitHub 원격 저장소 공유 5/28(수) : 초기 점검 회의				

2025		6월				
일요일	월요일	화요일	수요일	목요일	금요일	토요일
01	02 중간 점검	03	04	05 기능 완료	06 TEST 및 디버깅	07 문서 정리 - PPT 제작 - README 작성
08 발표 준비	09 마감(발표)	10	11	12	13	14



트러블 슈팅 #1

DataContext 바인딩 오류 - 참조 실패



- 버튼 클릭 이벤트에 RelayCommand 바인딩 실패
- 에러 메시지: 메서드를 찾을 수 없음

원인

ViewModel을 DataContext에 지정하지 않음

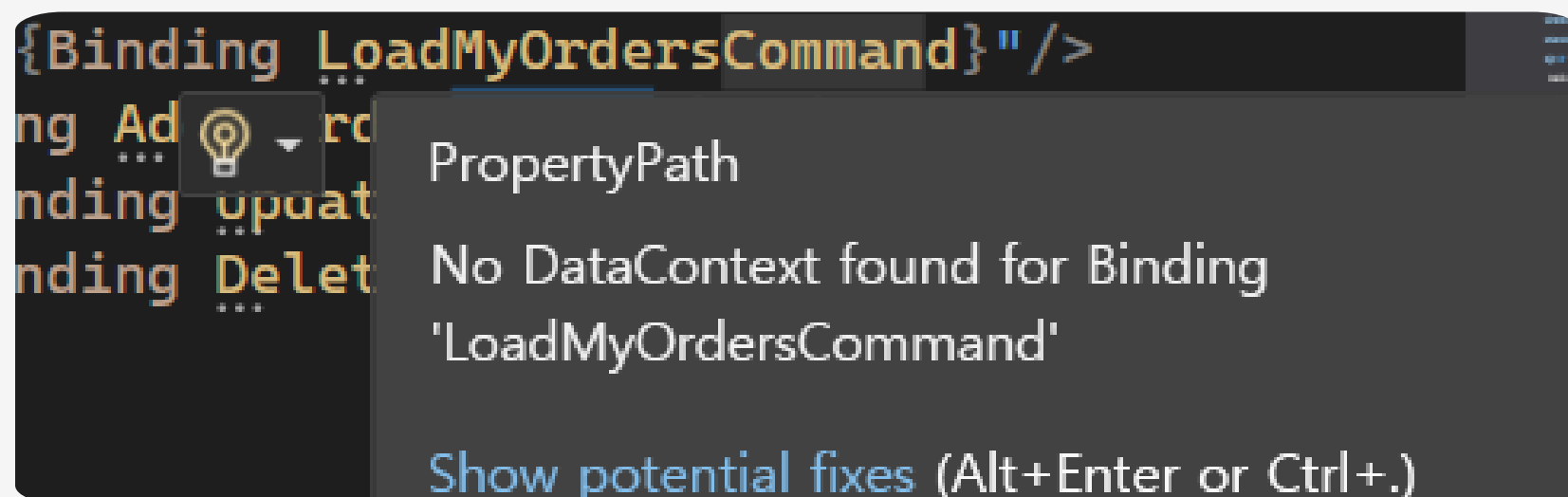
해결 방법

다음과 같이 xaml.cs와 xaml 상단에 ViewModel 파일 명시적으로 선언



트러블 슈팅 #2

PasswordBox 바인딩 이슈





결과 및 느낀점

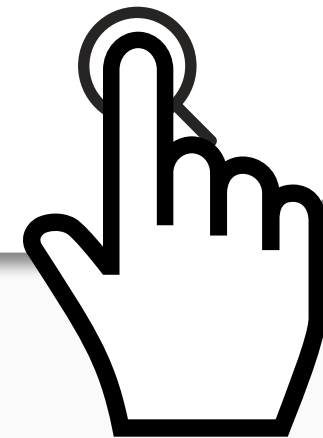
이름	어려웠던 점 및 해결 방법	개선점 및 느낀점
박현승	기본 코딩 개념 부족, 직접 구현 미흡 → 지속적 학습 필요	단기간 구현의 한계 경험, 팀원 간 소통과 체력 분배 중요성 체감
서예경	MySQL 초기 연결 문제 → 재설치 및 재연결로 해결 MVVM 구조화 어려움 → 예시 코드 참고하여 공부 DataContext 참조 문제 → 클래스 접근자 변경 및 클린 솔루션 시도로 해결	기능 집중 위한 시간 확보 필요, 리더로서 역할 및 목표 설정 중요성 인지
최마리	MySQL 연동 초기 시간 소모, 동기 메서드 위주 구현이 아쉬움 → 코드 구조가 잡힌 이후부터는 순조롭게 진행	의존성 주입 적용 필요, 비동기 메서드로 변경, 개발 기간 부족 아쉬움



CHAPTER 08 



Q&A





감사합니다 :))