



OKPOS ASP Service Clone coding

프로젝트 진행기간 / 개발자
2022.03.08 ~ 03.28 / 노수빈

Overview

1. 기업분석

- 기획의도 : 지원업무 파악
- 구현 시스템 이해 및 개발 목표
- OKPOS ASP 관리자 페이지 구조도
- OKPOS ASP 일반 사용자 페이지 구조도

2. 기획

- 페이지 구조도 및 URL 설계
- 프로토타입 제작 : Figma
- ERD 설계 : ERDcolud
- 추진 일정

3. 구현

- 결과 이미지
- 느낀점



1. 기업분석

- 기획의도 : 지원업무 파악
- 구현 시스템 이해 및 개발 목표
- OKPOS ASP 관리자 페이지 구조도
- OKPOS ASP 일반 사용자 페이지 구조도

기획의도 : 지원업무 파악

• 담당업무인 'OKPOS 내 웹 프론트엔드' 조사

- OKPOS의 사업분야 중, 웹과 관련 있는 것은 ASP라고 판단
- 담당업무, 자격요건, 우대사항에 적합한 인재이길 보여주기 위한 프로젝트 기획

웹 프론트엔드 엔지니어	<p>담당업무</p> <ul style="list-style-type: none"> • OKPOS 내 웹 프론트엔드 개발 및 유지보수 • 포토샵, 스케치, Figma 기반의 웹 퍼블리싱 <p>자격요건</p> <ul style="list-style-type: none"> • HTML5, CSS, Javascript(ES6+)에 사용 경험 • 소프트웨어 관련 전공 혹은 그에 준하는 경력 보유자 <p>우대사항</p> <ul style="list-style-type: none"> • AWS 환경에서 서비스 운영 및 배포 경험자 • Reactjs, Vuejs 등 웹 프레임워크 사용 개발 경험 • Git, Git-flow에 대한 이해 및 소스코드 관리 경험자 • PO/기획/디자이너/개발 등 다양한 직군과의 협업 경험
--------------	--

사업분야
POS H/W
키오스크(KIOSK)
신용카드 단말기
ASP
SI 컨설팅
유지보수
부가서비스

(좌) 잡코리아 OKPOS 공고, (우) OKPOS 홈페이지 메뉴

• 사용기술 및 프로그램

- HTML5, JSP, CSS, JS, JQuery, Spring Boot, JDK11, MySQL8, Maven4.0
- Eclipse jee2021-06, VSCode, PhotoShop, Figma, ERDcloud, Git

구현 시스템 이해 및 개발 목표

• OKPOS ASP Service

- 매장 내에서 발생하는 OSPOS의 키오스크 또는 POS 작업이 인터넷으로 실시간 전송되어 해당 정보를 PC나 스마트폰을 이용해 매장 관련 정보를 조회/관리할 수 있는 서비스
- 매장 내에서 POS로만 조회할 수 있었던 기존 방식을 인터넷으로 확장시킨 것
- 구성 : 매장POS, ASP Server, ASP(WEBPOS)

• 주요기능

- 매장POS, WEBPOS, 본지점수발주, 회원관리

• 서비스 이용자 구분

- ASP 서비스 관리자
- 대리점 관리자
- 단일 매장 관리자

• 개발 목표

- 내부자료 접근 불가로, 주요 기능에 기반 일부 구현
- 부족한 부분은 타사이트 참고하여 구현



OKPOS ASP 대리점 관리 페이지 구조도

OKPOS

초기화면

시스템 관리

기초관리

기타관리

인트라넷

POS버전관리

본사마스터

SMS관리

기타

매뉴얼다운로드

공지사항

게시판

시스템전용게시판

대리점게시판

팝업게시판

본사정보관리

매장정보관리

매장정보조회

정보변경확인

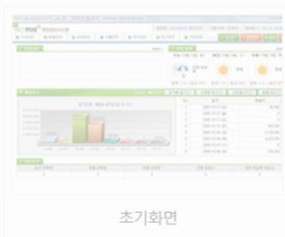
매장환경설정

• 분석자료

- 홈페이지에서 다운로드
(고객센터 - 자료실 - 165번 게시물 참고)

• 특징

- OKPOS의 이용 고객을 관리하는 페이지
- OKPOS 솔루션 매뉴얼/설치/버전관리
- 매장/본사 정보 관리



OKPOS ASP 단일 매장 관리 페이지 구조도

OKPOS 영업정보시스템

초기화면

기초관리

매출관리

회원관리

수불관리

재고관리

정산관리

기타관리

매장관리

출력물관리

상품관리

가격관리

결제수단

기타관리

키오스크관리

매장정보관리

사람정보관리

사람정보조회

패스워드변경

VAN/코너 정보

테이블속성

테이블관리

포스기능정의

상품기초정보등록

거래처등록

상품등록/조회

상품정보일괄변경

상품등록

판매터치키등록

주방프린터

이미지관리

다국어 관리

터치키관리

품질 일괄변경

• 분석자료

- 구글링 (okpos asp)

• 특징

- OKPOS 이용 고객의 POS 정보 관리
- 매장의 상품 판매와 관련된 정보 관리
- 매장의 매출/회원 등과 관련된 정보 관리



2. 기획

- 페이지 구조도 및 URL 설계
- 프로토타입 제작 : Figma
- ERD 설계 : ERDcolud
- 추진 일정

페이지 구조도 및 URL 설계



구현 페이지

- 1) 로그인 페이지
- 2) 대리점 관리 페이지 초기화면
- 3) 시스템 관리 – 본사마스터 – 매장환경설정
- 4) 단일 매장 관리 페이지 초기화면
- 5) 기초관리 – 상품관리 - 상품정보일괄변경

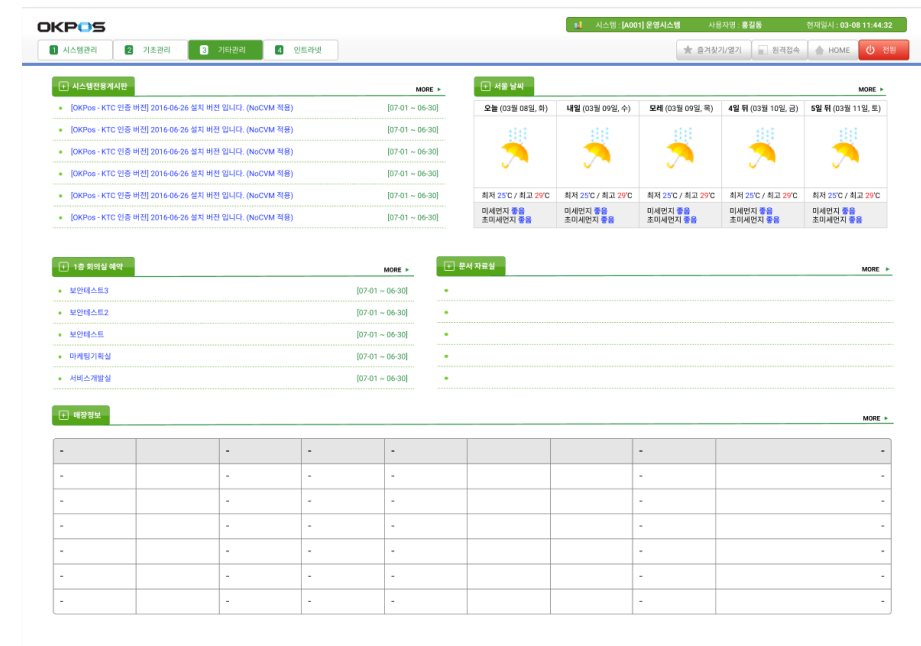
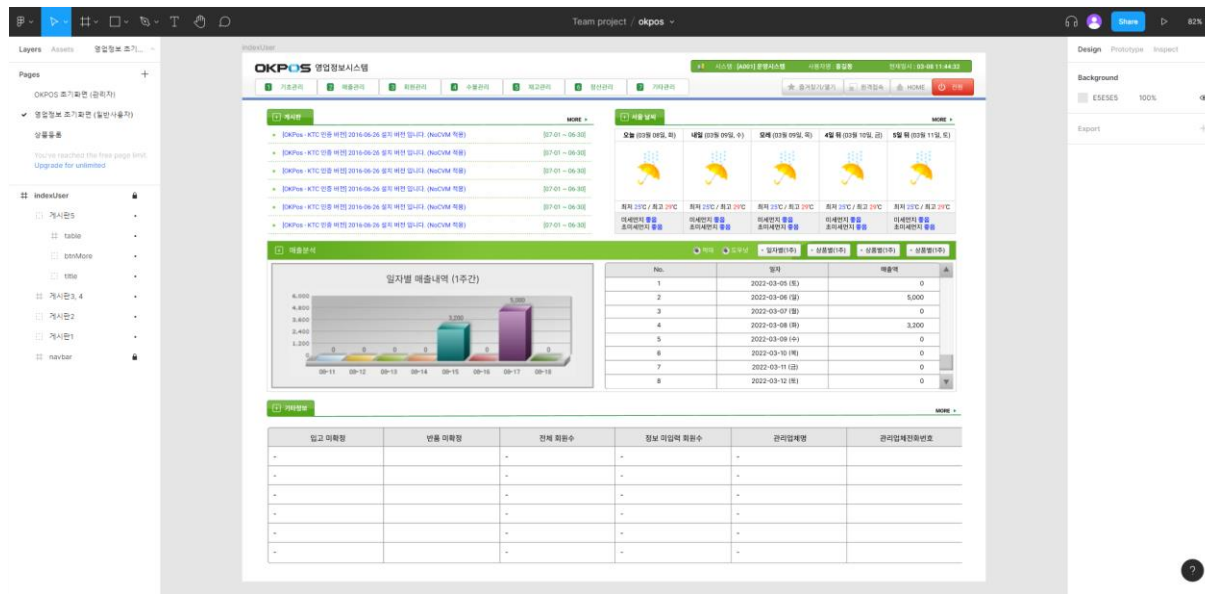
프로토타입 제작 : Figma

장점

- 다수의 사용자의 동시 편집 가능
- 다른 GUI Tool(kakao oven, powerpoint)에 비해 직관적인 구성으로, 첫 사용임에도 단시간에 숙련도 향상 가능

단점

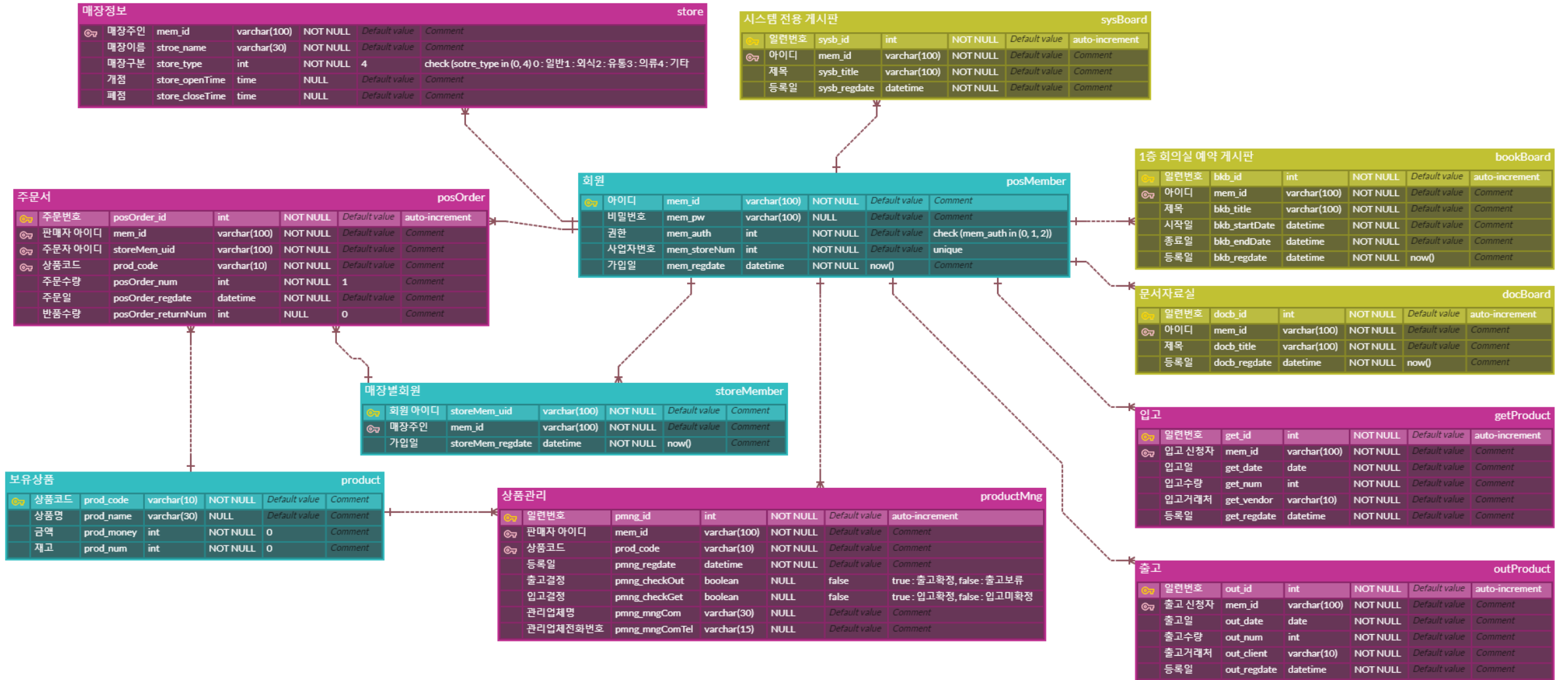
- table 제작이 불편함 (플러그인을 이용해도 복잡한 데이터는 다루기 힘들)
- 무료버전은 세 페이지만 만들 수 있음



(좌) 일반사용자 초기 화면 프로토타입, (우) 관리자 초기 화면 프로토타입

[Figma로 만든 프로토타입] https://www.figma.com/team_invite/redeem/snr9kBj6qz9P2w3lhQVTpZ

ERD 설계 : ERDcloud



/ 추진 일정

구분	추진내용	추진일정				
		3/8	3/9	3/10	3/11	3/12 ~ 3/28
계획	기업 분석 및 프로젝트 주제 선정					
분석	ASP 서비스 시장 분석 OKPOS의 'KIOSK 통합 설치 매뉴얼(ver.20210217)' 분석					
설계	Figma를 이용하여 OKPOS WebPOS 기반의 Prototype 구현					
	사이트 이용에 필요한 최소한의 ERD 설계					
개발	개발 환경(Eclipse) 세팅, DB DDL Spring MVC Design Pattern를 따르는 DAO, DTO 작성					
	Web Frontend(jsp, css, js, jquery) 개발					
	최소한의 작동을 위한 Web Backend 개발					
	디버깅					

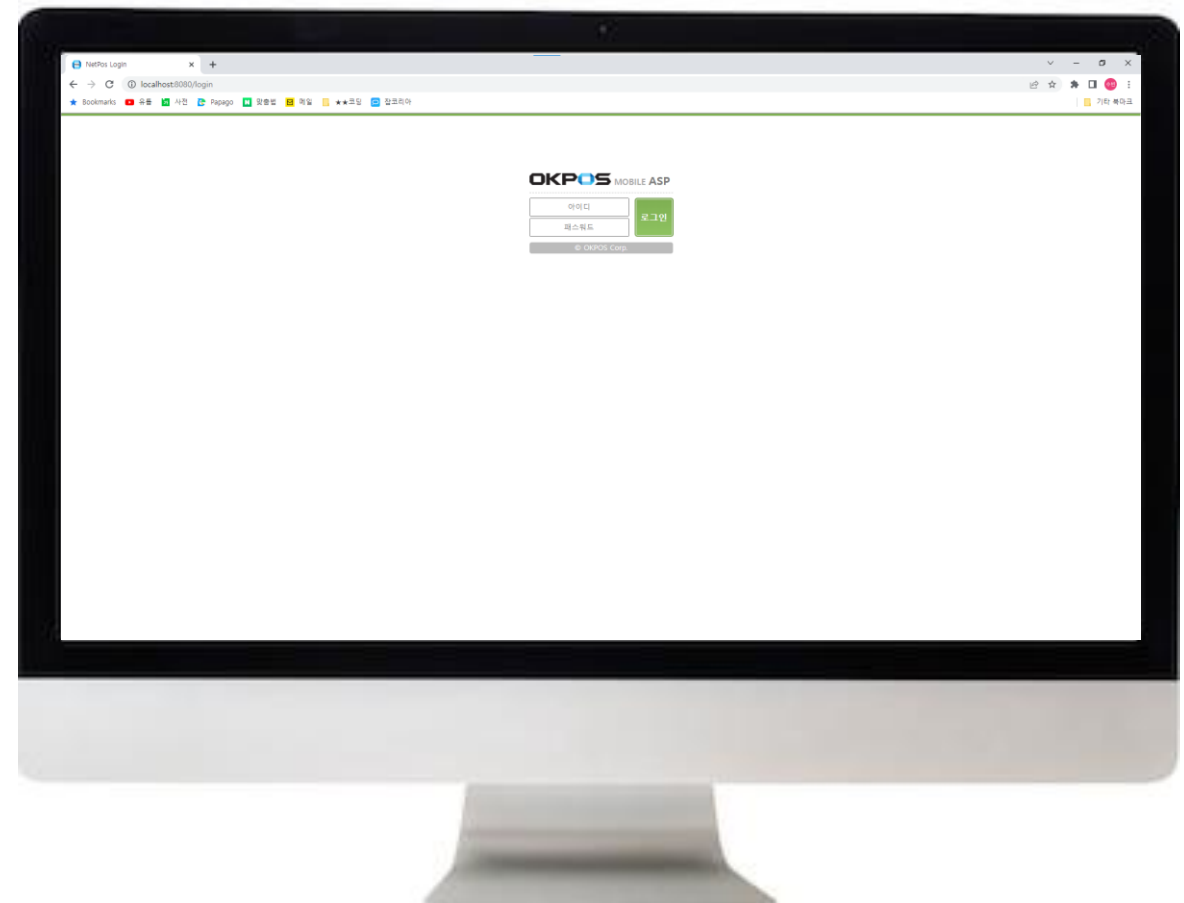
3. 구현

- 결과 이미지
- 느낀점

결과 이미지

로그인 페이지

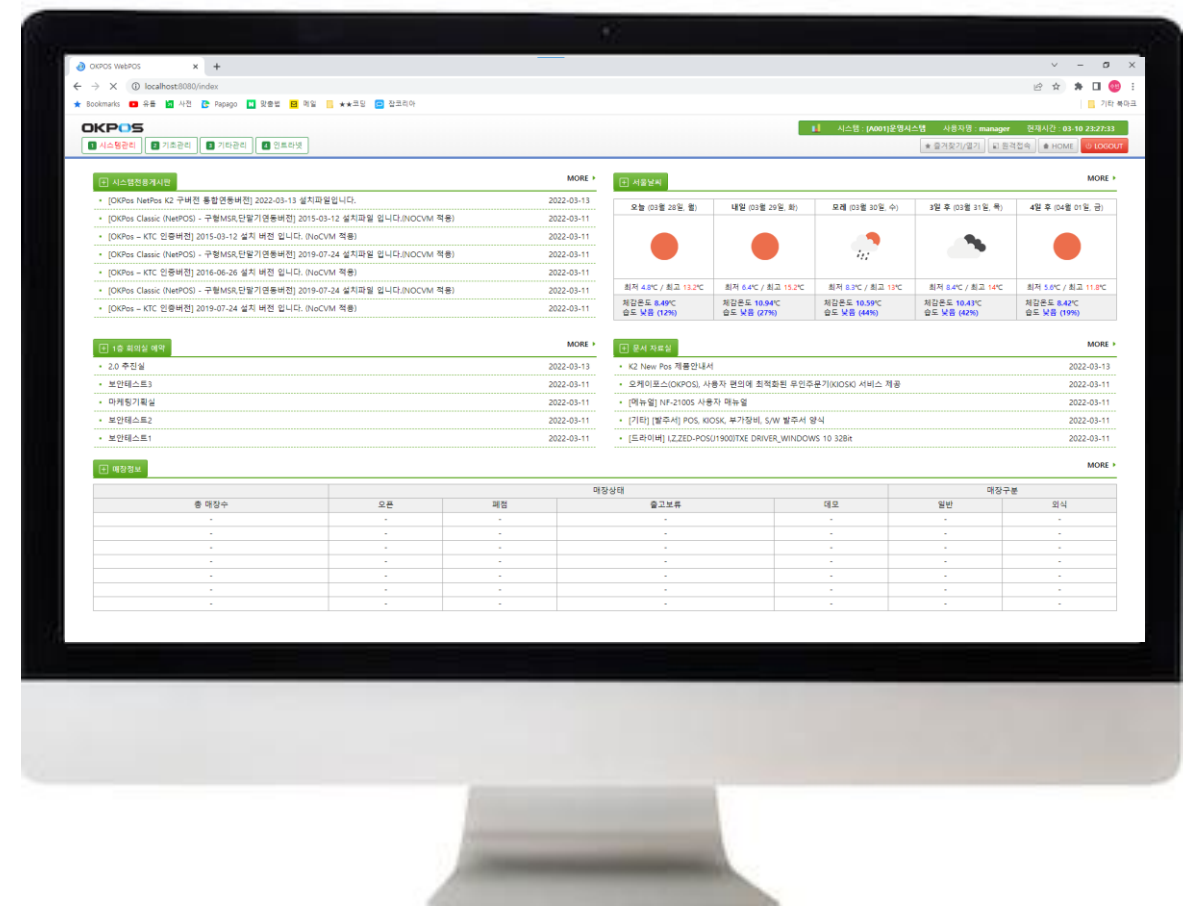
- 특정 페이지의 이용에 적합한 사용 권한을 가졌는지 확인하기 위한 인증 절차
- ID/PW 중 하나라도 틀리면 아무런 경고도 뜨지 않은 채, /login?error로 uri 변경



결과 이미지

• (MANAGER 권한) index 페이지

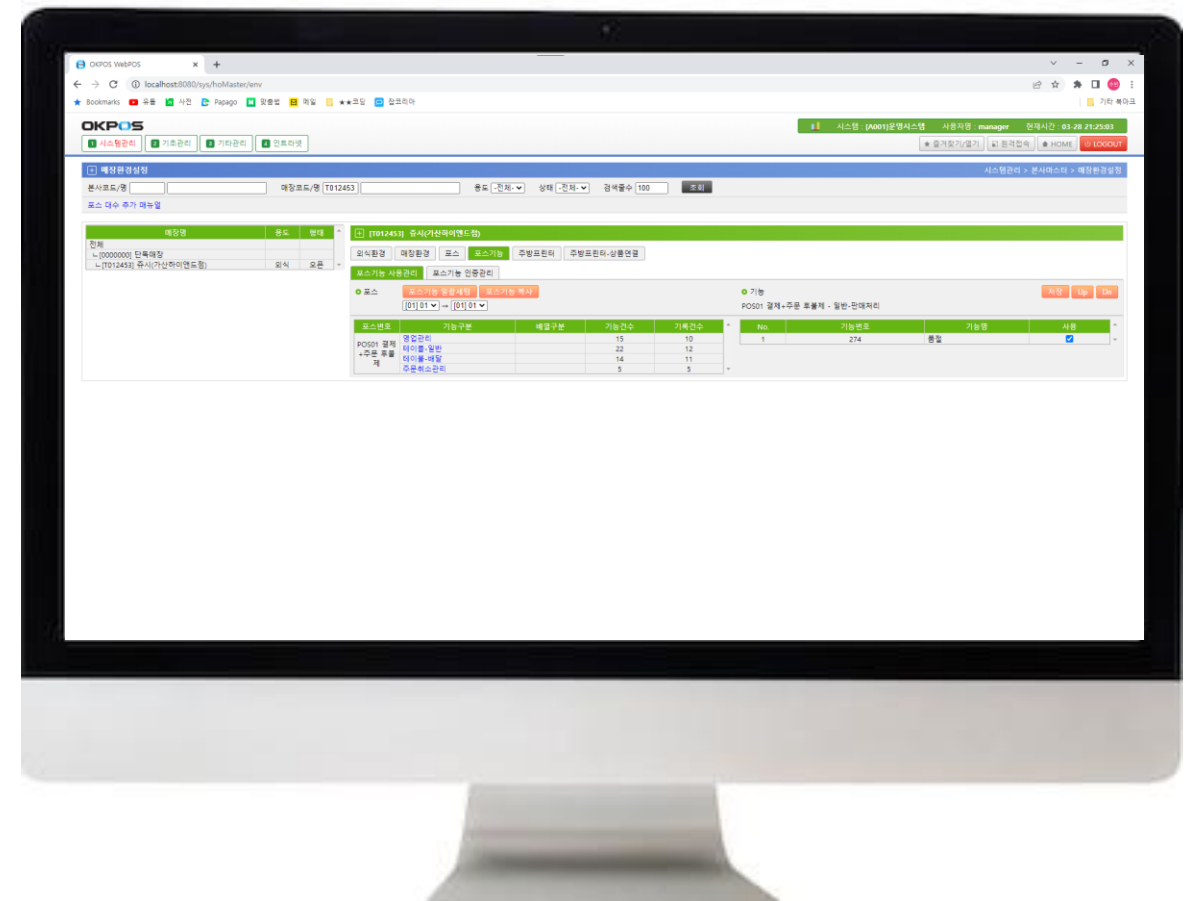
- MANAGER 권한으로 로그인하면 연결되는 초기화면
- 구성요소 : 게시판(3), 날씨API(1), 매장정보(1)



결과 이미지

• (MANAGER 권한) 매장환경설정 페이지

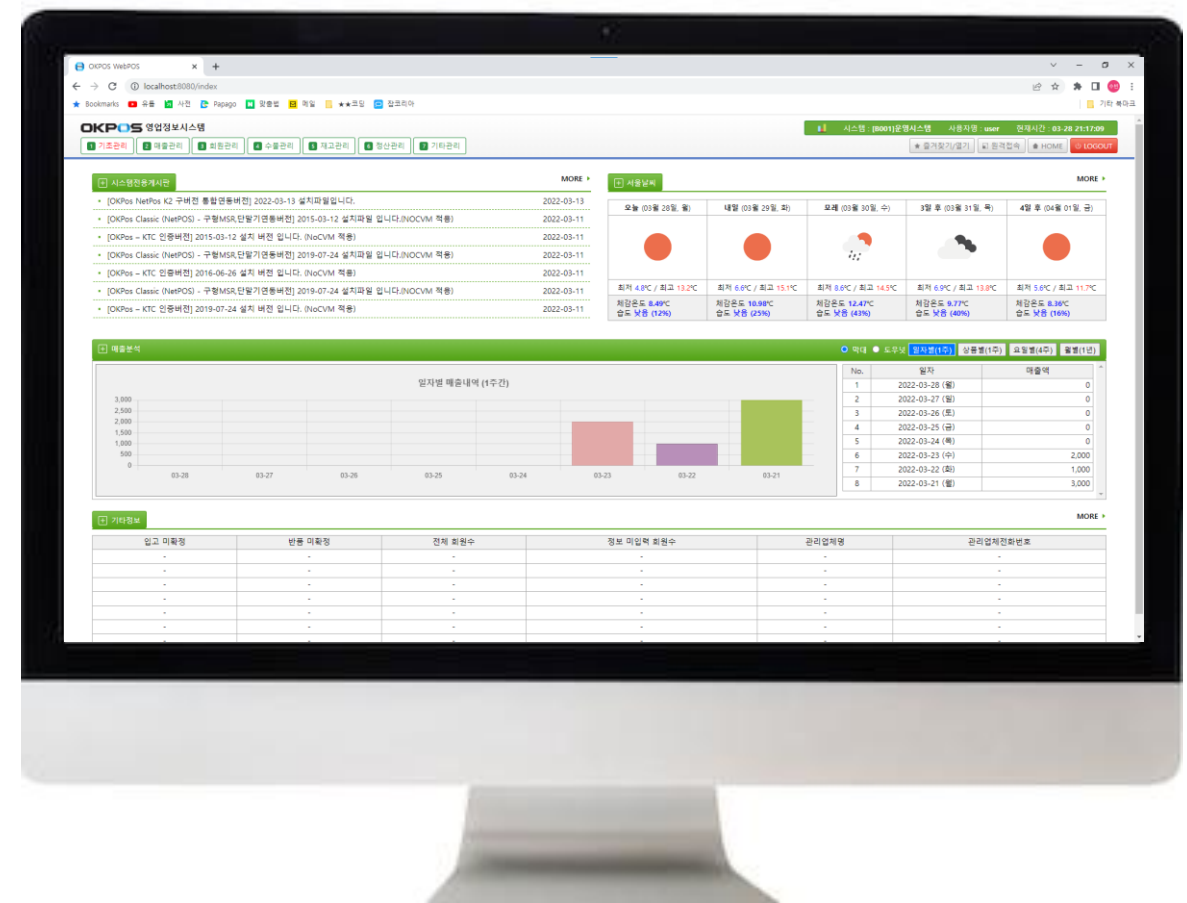
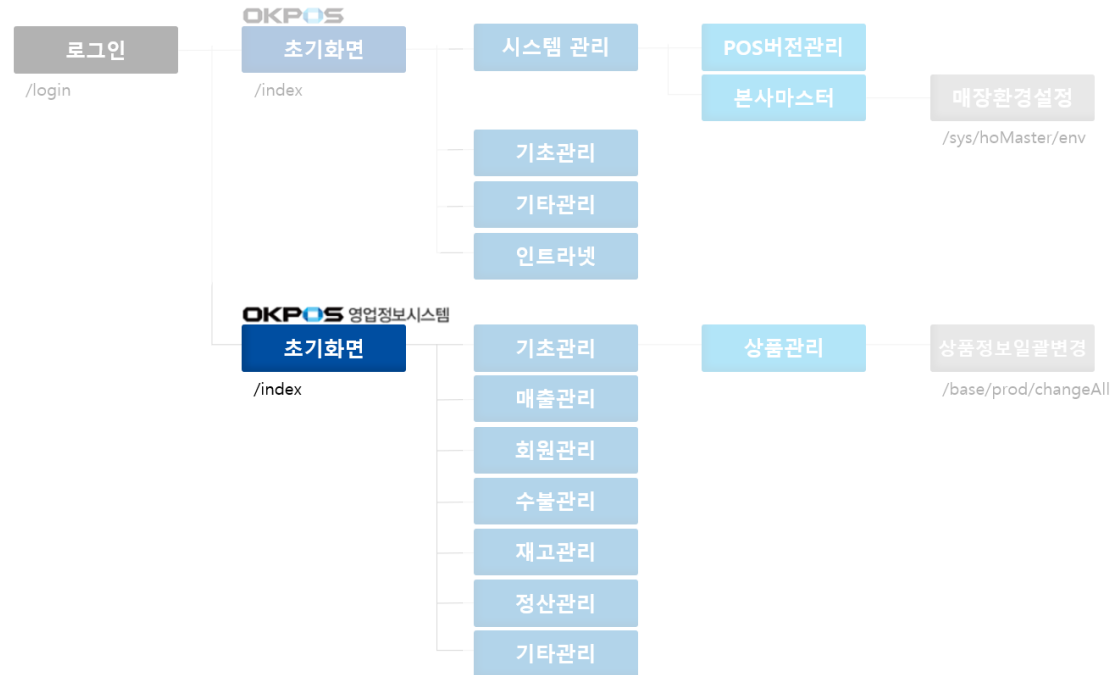
- 시스템 관리의 서브 페이지
- 구성요소 : 게시판(1), 날씨API(1), 매출분석(1), 기타(1)



결과 이미지

• (USER 권한) index 페이지

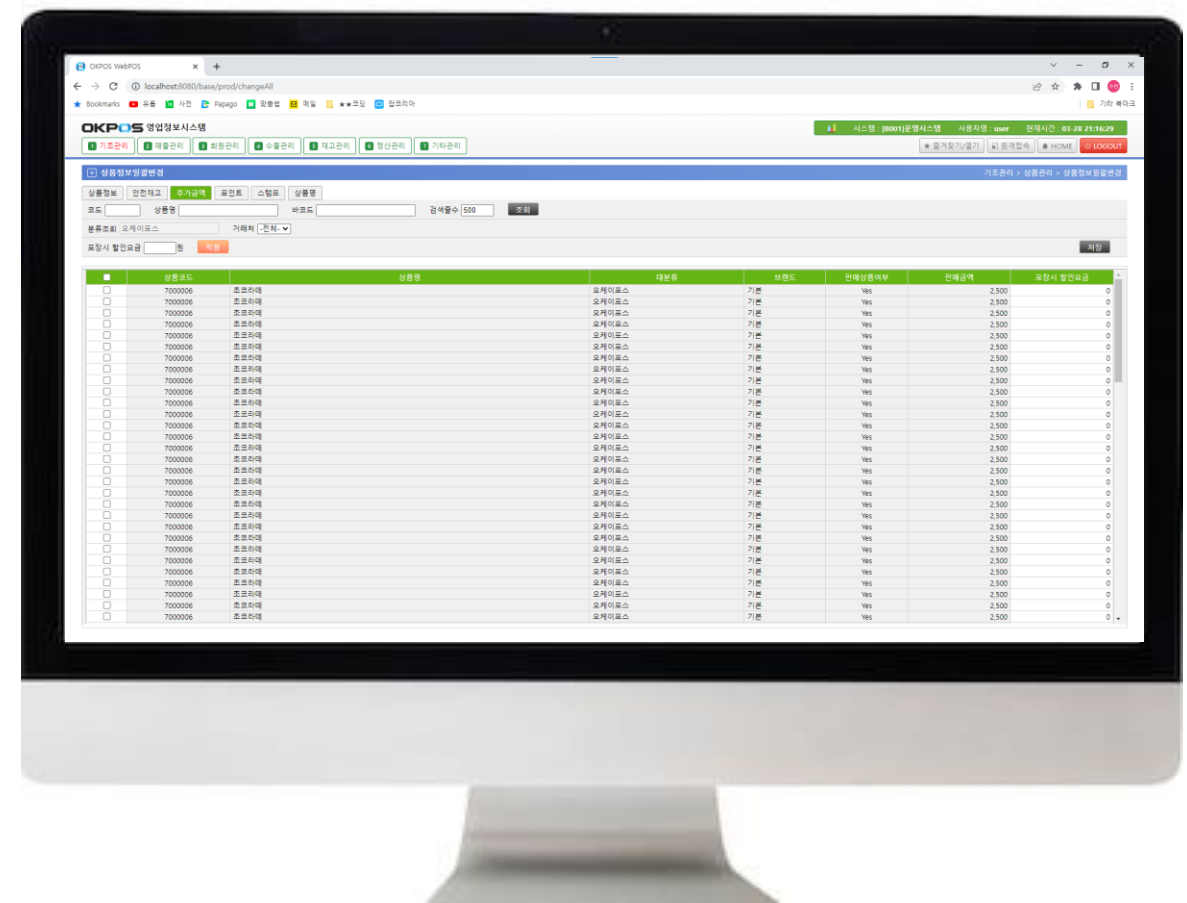
- USER 권한으로 로그인하면 연결되는 초기화면
- 구성요소 : 게시판(1), 날씨API(1), 매출분석(1), 기타(1)



결과 이미지

• (USER 권한) 상품정보일괄변경 페이지

- 기초관리의 서브페이지
- 구성요소 : 게시판(1), 날씨API(1), 매출분석(1), 기타(1)



/ 느낀점

• 기획 단계에서 개발 범위를 명확히 정하지 않으면, 개발 단계에서 진행이 힘들

- 지금까지 개발한 웹 사이트는 페이지가 복잡하지 않았음
- 기획 단계가 불분명해도 개발할 페이지에 어떤 테이블의 어떤 칼럼이 필요하고, 해당 페이지에 어떤 요소를 왜 배치해야 하는지에 대한 파악을 기억력에 의존해도 문제 없었음
- OKPOS의 ASP의 일부만 구현하면서 개발 범위를 애매하게 잡으니, 불필요한 테이블 생성, 필수 테이블 또는 칼럼의 부재 등의 문제로 **DDL 설계를 뒤엎는 일**이 빈번했음
- URI 설계를 하지 않고 개발하니, Controller에서 Forwarding 할 때 실제 파일 경로와 Mapping된 URI를 매치시키지 못해 404 에러를 자주 직면함
- 페이지 구조도를 작성하지 않고 초기화면을 구축하자, 빠진 내용이 없는지 확인하는 과정에서 딜레이가 많이 생겼음
- 위와 같은 이유로, 기획 단계에서 개발 범위를 명확히 정하지 않으니 개발의 **진행상황 확인 및 오류 체크를 위한 점검표를 만들에 어려움**을 느낌
- 초기 계획은 기획을 하루 안에 끝내는 것이었으나 결국, 기획 내용을 명확히 정의하기 위해 3일을 추가로 투자했음
- 기획에서 개발 범위를 명확히 정한다고 해서, 설계를 잘하는 것은 아니었음.
 - 1) ex1 - PK를 주면 안 되는 컬럼이 많았는데 이를 수정하면 FK 제약조건에 위반되는 컬럼이 생기며 관련 테이블까지 수정하는 게 어려웠음
 - 2) ex2 - 서브 페이지 구현에서 필요한 데이터가 무엇이고, 그 데이터는 어떤 테이블의 컬럼으로 위치시킬지 정해야 됐는데 테이블 설계가 꼬이면서 DB Anomaly 발생. 이 때문에 관련 테이블을 전부 삭제하고, back-end는 구현하지 못 함.

느낀점

• 의지 대상의 부재로, 더 많은 공부를 하게 됨

- 팀프로젝트에서는 개인의 역량이 부족해도 다른 팀원이 부족한 부분을 메꿔주니, 하루를 투자해도 해결되지 않는 문제는 다른 팀원의 도움을 받아 해결하게 됨
- 그 중 하나가 Spring Security였는데, 이번 프로젝트에서 로그인 프로세스를 완성하기 위해 3일 소요 (실 소요시간 36~40시간 추정)
- 개발 시간은 팀 프로젝트의 2~3배가 필요했지만, 개발 역량 향상에 많은 도움이 되었음
- 지금껏 개발자 역량 중, 가장 부족한 부문이 정보 검색 능력(단순 코드복사가 아닌, 문제 상황에 대한 해결 방안 이해)이었는데 이번 프로젝트를 통해 많이 개선됨
- **개선 전 : 문제 상황에 대한 단순 코드복사**
 - 1) ex1 - spring login으로 검색하여 나온 코드 복사 붙여넣기한 뒤, 코드 실행 순서를 따라가며 전반적인 흐름만 파악
 - 2) ex2 - 검색되지 않는 문제는 해결하지 못하는 경우가 빈번함
- **개선 후 : 문제 상황의 원인을 기반으로 관련 이슈의 개념을 찾아보며 개발 코드에 적용 가능한 형태로 바꿈**
 - 1) ex1 - spring security의 개념을 관련 클래스/인터페이스(WebSecurityConfigurer, UserDetails, UserDetailsService, ...)로 이해(어떤 역할을 하는지 관련 메서드(HttpSecurity.loginProcessingUrl(), ...)를 검색해가며 이해)한 뒤, 검색 정보를 필요한 기능과 불필요한 기능으로 나눠가며 개발
 - 2) ex2 - 검색되지 않는 문제가 있다면 원인 파악부터 다시 진행함으로써, 검색 키워드를 넓히거나 새로운 문제 방향을 찾아내어 해결. (이 경우는 대부분 해당 개념의 일부를 이해하지 못하고 넘어가면서 발생한 문제였기 때문에 원인 파악을 하며 어떤 개념을 놓치고 넘어간 것인지 알게 됨)