

백엔드 개발 직무 지원자  
**KIM HEEJEONG**

**Contact**

010-5705-9594  
developer.heejeong@gmail.com

# Contents

1 자기 소개

2 직무 경험

3 핵심 역량

## 1. 자기소개

# 1분 자기소개

4년 차 Full-Stack 개발자 김희정입니다.

자체 시스템 개발 회사에서 **웹 솔루션 개발 직무**로 근무하며, **백오피스부터 하이브리드 웹앱, 배치 프로그램, 미들웨어** 등 다양한 업무를 경험하였습니다.

신입 때부터 **사수 없이 실무에 투입**되었고, **총 10개 이상의 프로젝트**를 소프트웨어 생명주기(SDLC) 전반에 걸쳐 **주도 개발** 하였습니다.

이를 통해 시스템의 전반적인 구조를 이해하고, 설계 및 개발하는 능력을 길러 **하나의 애플리케이션을 전반적으로 개발할 수 있는 능력**을 갖추게 되었습니다.

# 간략 요약

- 4년 차 Full-Stack 개발
- 자체시스템 개발 회사에서 웹 솔루션 개발 직무
- 다양한 업무 경험 (백오피스, 하이브리드 웹앱, 배치 프로그램, 미들웨어 등 개발)
- 신입 때부터 사수 없이 투입
- 총 10개 이상의 프로젝트 SDLC 전반에 걸쳐 주도 개발
- 하나의 애플리케이션을 전반적으로 개발할 수 있는 능력

## 1. 자기소개

# 저의 **이력**을 간략하게 소개하자면



### 학력

컴퓨터공학 학사  
국비교육 수료 (2개월)



### 경력

인턴 4개월  
나이콤 3년 5개월



### 자격증

정보처리기사 (2019)  
SQLD (2022)  
리눅스 마스터 2급 (2024)

## 1. 자기소개

# 제가 사용해본 기술 스택은 이렇습니다

### Frontend

HTML/CSS/JS

jQuery  
React (node.js)

### Protocol

HTTP  
Socket  
WebSocket  
ActiveMQ

### Backend

Java 1.8+

Spring Framework  
(+ Spring Boot)

Spring Security

MyBatis

Spring Data JPA  
(+ QueryDSL)

### DBMS

PostgreSQL  
MS-SQL  
...

MongoDB

### DVCS

Git  
( + GitHub )

SVN

### ETC

Kotlin  
Android Studio

Swift  
XCode

PHP 5.x

Python 3.8+  
Django  
Framework

## 2. 직무경험

### 직전회사 **나이콤 (NICOM)**

#### ➤ 나이콤(NICOM)은 어떤 회사입니까?

- 주 사업영역: 제조업, 도서관 솔루션
- 하드웨어(기구, 펌웨어)부터 소프트웨어(키오스크, 웹)까지 자체 시스템 제작

#### ➤ 어떤 제품을 판매합니까?

- 도서관에 있는 자동화 장비
- 판매 제품: 출입 통제 시스템(게이트, 생체인증), 대출/반납기, 좌석 발급기, 무인회원증 발급기, 안내이송로봇 등

#### ➤ 어떤 팀에 속해 있었습니까?

- SW 팀 입사 → 7명 (키오스크 4, 디자인 1, 웹 2)
- Web 팀 이동 → 8명 (팀장, 대리 2, 사원 5)

#### ➤ 어떤 솔루션을 개발하였습니까?

- 백오피스: 주 고객층(도서관 사서)이 장비를 효율적으로 관리할 수 있도록 설정 관리, 이력 및 통계 제공하는 백오피스
- API: 타사 혹은 내부 솔루션에서 연계하는 API
- 앱: 스마트도서관 앱 개발
- 미들웨어: 로봇과 타 프로그램을 연동하는 미들웨어 개발



**출입 게이트**

인증을 통해 이용자 식별 및  
**출입 통제 시스템**  
생체인증(지문, 지정맥, 얼굴)



**좌석발급기**

무인 **좌석발급 시스템**.  
열람실 좌석 배치 및 현황 등



**스마트도서관**

사용자들이 인접하기 쉬운  
위치(지하철 역, 핫 플레이스)  
해 있는 **대출/반납 시스템**



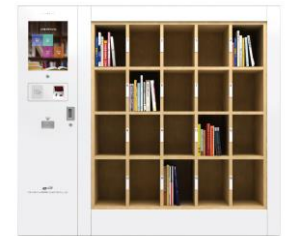
**무인 회원증 발급기**

도서관 회원증 발급을 이용  
자 스스로 비대면으로 발급



**안내 이송 로봇**

도서 정보 검색/위치 안내,  
대출/반납 등의 다양한 기능  
을 제공하는 로봇



**스마트 서가**

RFID 기술로 도서위치조회,  
자동반납 등 기능 제공

2. 직무경험

참여 프로젝트

프로젝트명	기간	인원	설명	담당역할
도서 위치 추적 시스템	2023.05 – 2023.10	SW 4人, 담당 1人	도서 위치 추적 시스템	관리자 프로그램 개발 (고도화)
안내 이송 로봇	2023.01 – 2023.04	SW 4人, 담당 1人	도서 길안내 및 대출, 반납 시스템	미들웨어 개발
스마트 서가 및 북박스 시스템	2022.12 – 2023.11	1人	무인 반납 시스템	관리자 프로그램 개발 (고도화)
좌석 관리 시스템	2022.08 – 2022.12	3人	무인 좌석 발급 시스템	관리자 프로그램 개발 (고도화)
생체 인증 솔루션	2022.07 – 2022.09	SW 2人, 담당 1人	출입 통제 단말기를 이용한 출입 관리 시스템 (GS 인증)	생체 인증 서버 API 개발
무인 회원증 발급 시스템	2022.05 – 2022.08	SW 2人, 담당 1人	무인 회원증 발급 시스템	관리자 프로그램 개발 (웹 접근성 인증)
초중고 도서 대출반납 시스템	2021.10 – 2021.12	SW 3人, 담당 1人	무인 도서 대출, 반납 시스템	관리자 프로그램 개발
스마트도서관	2021.01 – 2022.06	1人	무인 도서 대출, 반납 시스템	모바일 앱 및 부가 애플리케이션 개발 (고도화)
얼굴인식 출입 통제 시스템	2020.10 – 2020.11	SW 2人, 담당 1人	안면 인식을 통한 출입 관리 시스템	관리자 프로그램 개발

# 주요 프로젝트 소개



### Project 1

#### 스마트도서관 시스템 - 시스템 고도화

이용자들이 스마트도서관을 보다 편리하게 이용할 수 있도록 개발된 모바일 앱 및 부가 애플리케이션을 고도화 개발하였습니다.



### Project 2

#### 출입 통제 시스템(안면인식) - 백오피스 개발

안면인식 기능이 추가된 출입 통제 시스템에서 안면인식 장비와 연동하여 이용자들의 얼굴 데이터를 관리하는 관리자 프로그램을 개발하였습니다.



## 2. 직무경험

# 스마트도서관 시스템

### ※ Project

- 프로젝트명: 스마트도서관 시스템 고도화
- 인원: 1인
- 기간: 2021.01 – 2022.06

### ※ Tech Stack

- Language: Java, Kotlin, Swift
- Framework: Spring, Android, Xcode
- Database: MS-SQL

### ※ Applications

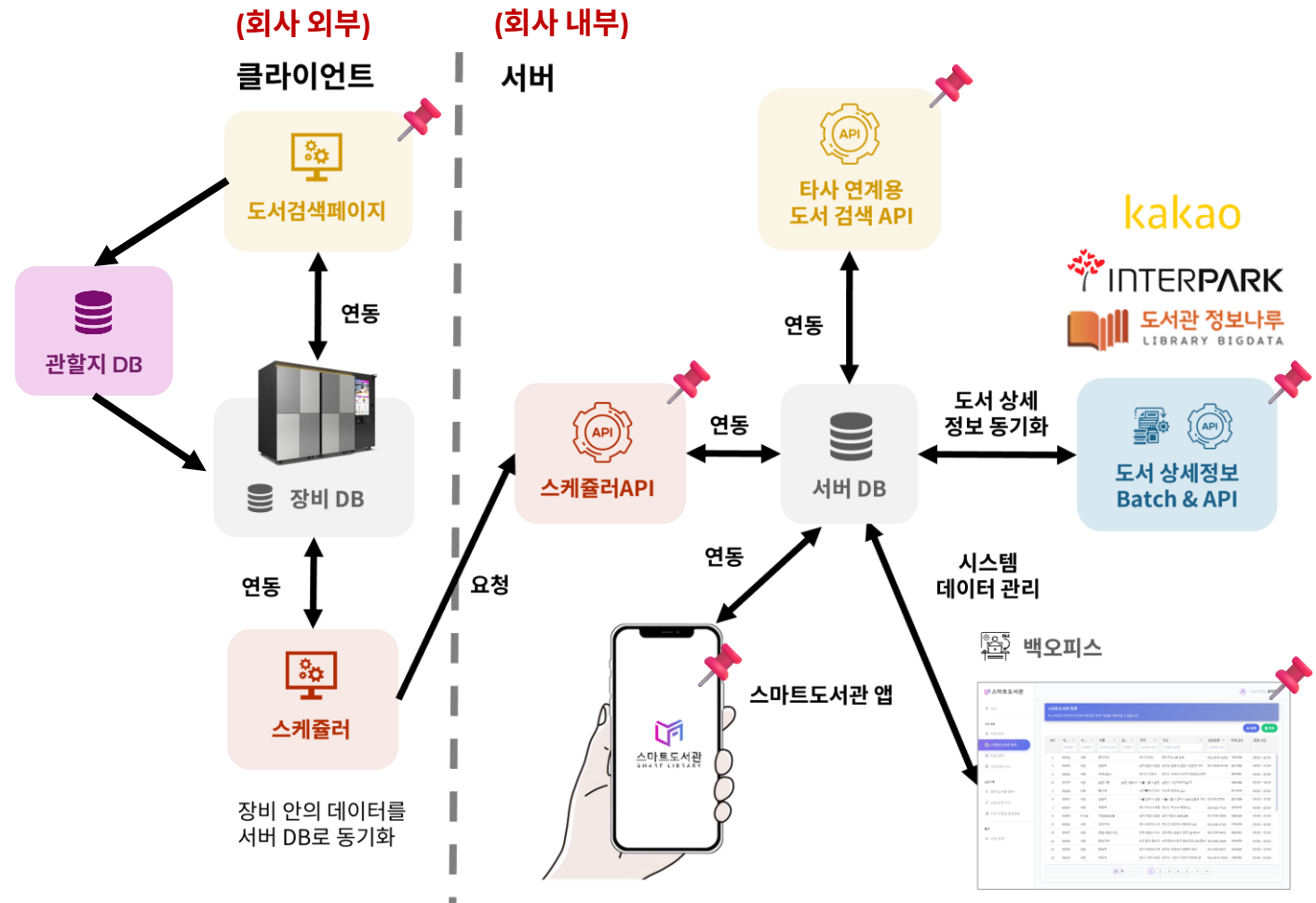
[클라이언트]

- 도서 검색페이지 (타사 제공용) - 고도화

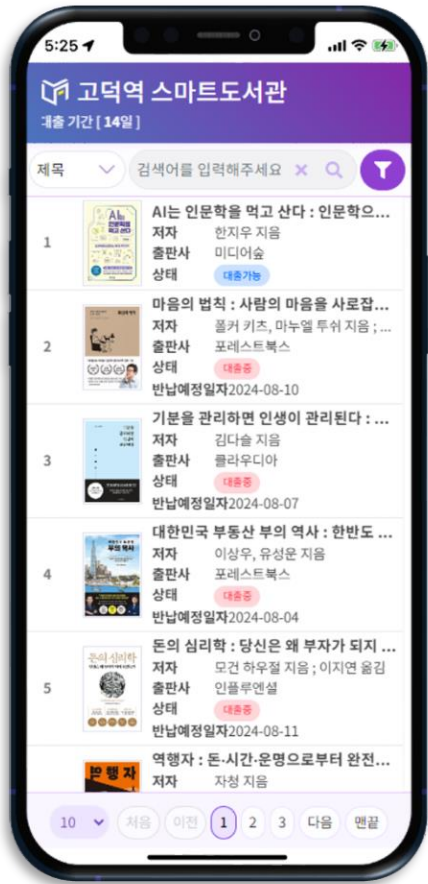
[서버]

- 스마트도서관 웹앱(Android/iOS) - 고도화
- 스마트도서관 관리자페이지 - 고도화
- 스마트도서관 스케줄러 API - 고도화
- 도서 상세정보 Batch & API - 신규
- 도서 검색 API (타사 연계용) - 신규

# 시스템 구조



# 도서 검색페이지 고도화



검색 페이지 모바일 화면

S

## Situation

도서 검색페이지는 스마트도서관의 도서 목록을 조회하는 애플리케이션으로, 배포 사이트마다 제공됩니다. 스마트도서관은 주로 도서관 관할지로 묶여 있습니다. 예시) 강동구 통합 도서관 → 고덕역, 중앙보훈병원역

관할지마다 물리적으로 분리된 다수의 스마트도서관 도서 목록 조회할 수 있어야 합니다.

- 관할지마다 스마트도서관이 같은 네트워크 위치 → 검색페이지,
- 다른 네트워크 혹은 홈페이지에 융합 → 검색 API

T

## Task

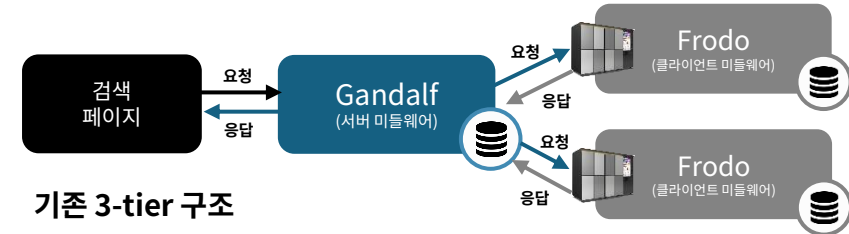
기존 프로그램에서는 이 문제를 해결하기 위해 각 계층(Server: 관할지, Client: 스마트도서관)에 미들웨어를 두어 각각 Socket 통신 → 이로 인한 다수 문제 발생

### 1. 관리의 어려움

유지보수: 3개 프로그램  
배포: 2 + N (장비개수)

### 2. 구조적 문제

소켓 통신 구조로 실패율 상승  
및 속도 증가 등 구조적 문제



기존 3-tier 구조

A

## Action

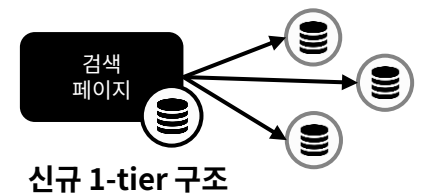
- 원인/구조 분석: 각 장비의 DB에 접근하기 위해 계층에 둔 미들웨어를 걷어낼 수 있을까?
- 리스크 분석: 미들웨어를 걷어내면 부수 효과는 없을까?
- 해결 방안 모색: 다중 DB 연결 고안 → Spring의 RoutingDataSource를 이용하여 구현

R

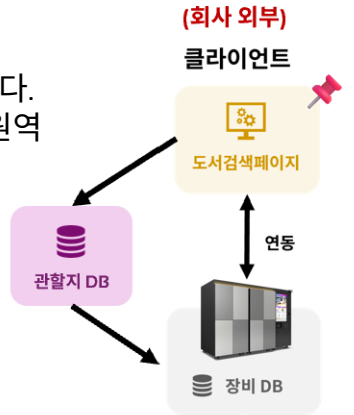
## Result

3-tier 시스템을 1-tier로 구조를 개선하여 유지보수 및 성능 향상시켰습니다.

- 유지보수 프로젝트 3개 → 1개로 개선
- 배포 개수: 2 + N (장비개수) 개 → 1개로 개선
- 검색 속도 개선: 5 vu/s p99 7.5s → 100 vu/s p99 93ms (약 60초간 부하 테스트, 도구: Artillery)



신규 1-tier 구조



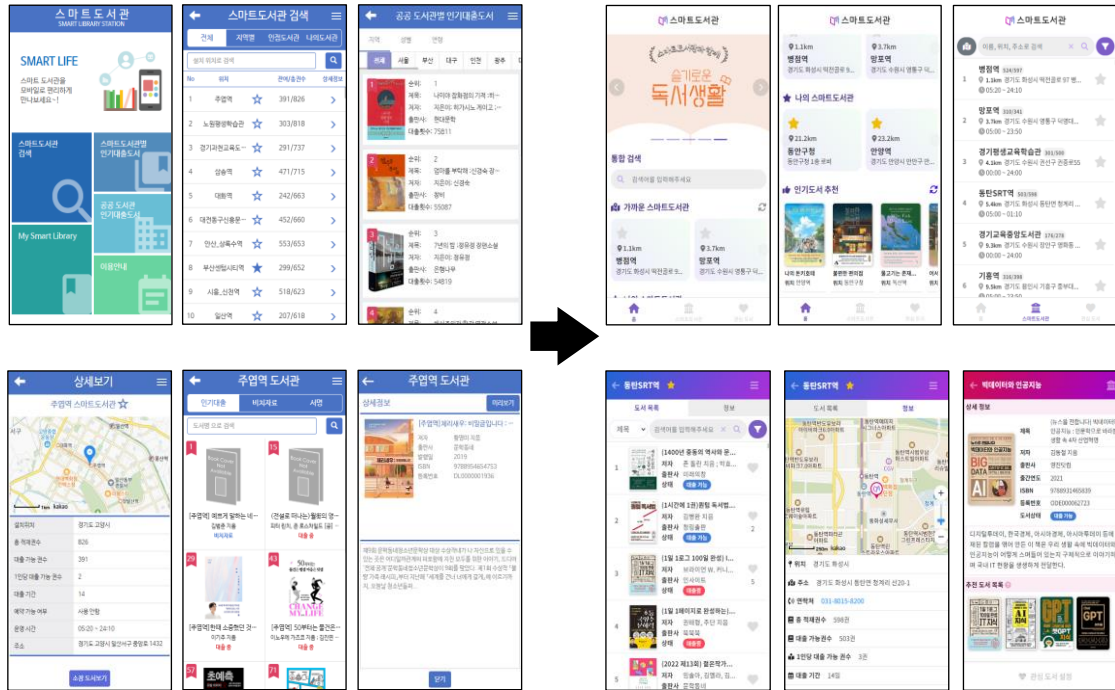
# 스마트 도서관 앱 고도화

AS-IS

TO-BE

AS-IS

TO-BE



불필요한 화면 분리 및  
멀티 Depts로 불편한 UI/UX

직관적이고 일관된 UI/UX

앱의 취지와 동떨어진  
공공도서관 인기 대출도서 메뉴

해당 메뉴의 기능을  
도서 추천 기능으로 융합

기존 개발자의 퇴사로  
유지보수가 어려운 기술 스택  
(node.js, Angular, go)

주로 사용하는 기술 스택으로 전환  
(Java, Spring)

WebView 로딩 지연 시 빈 화면

Splash Screen 구현  
WebView 로드 시 전환

App 최적화 X

Full-Screen, Swiper 제스처,  
알림 창 등 Native 최적화

UI/UX 개선 예시. 특정 스마트도서관의 도서 목록을 보고 싶다면?

- 기존: 메인 → 스마트도서관 검색 메뉴 클릭 → 스마트도서관 클릭 → 소장도서 보러 가기 클릭 (총 4 depts)
- 현황: 스마트도서관 탭 > 스마트도서관 클릭 (총 2depts)

# 출입통제 시스템

## ※ Project

- **프로젝트명:** 얼굴인식 출입통제 시스템
- **설명:** 출입 게이트에 이용자 얼굴 인식 기능을 추가한 시스템
- **인원:** 1人
- **담당 역할:** 백오피스(관리자 프로그램)
- **기간:** 2020.10 – 2020.11 (**고도화** 2023.09 – 2023.10)

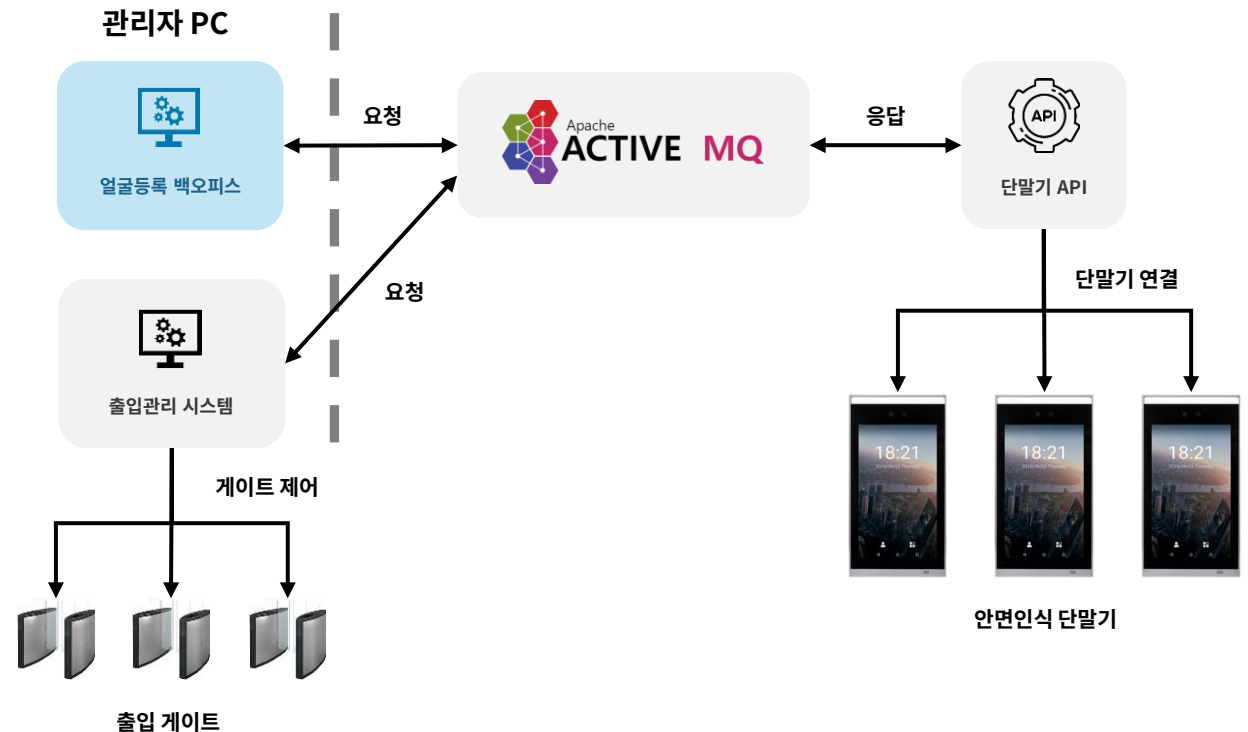
## ※ Tech Stack

- Language: Java 1.8
- Framework: Spring Boot 2.x, Spring Security 5.x
- Message Broker: **ActiveMQ**
- Template: Thymeleaf
- Persistence: JPA, QueryDSL
- DB: PostgreSQL

## ※ Process

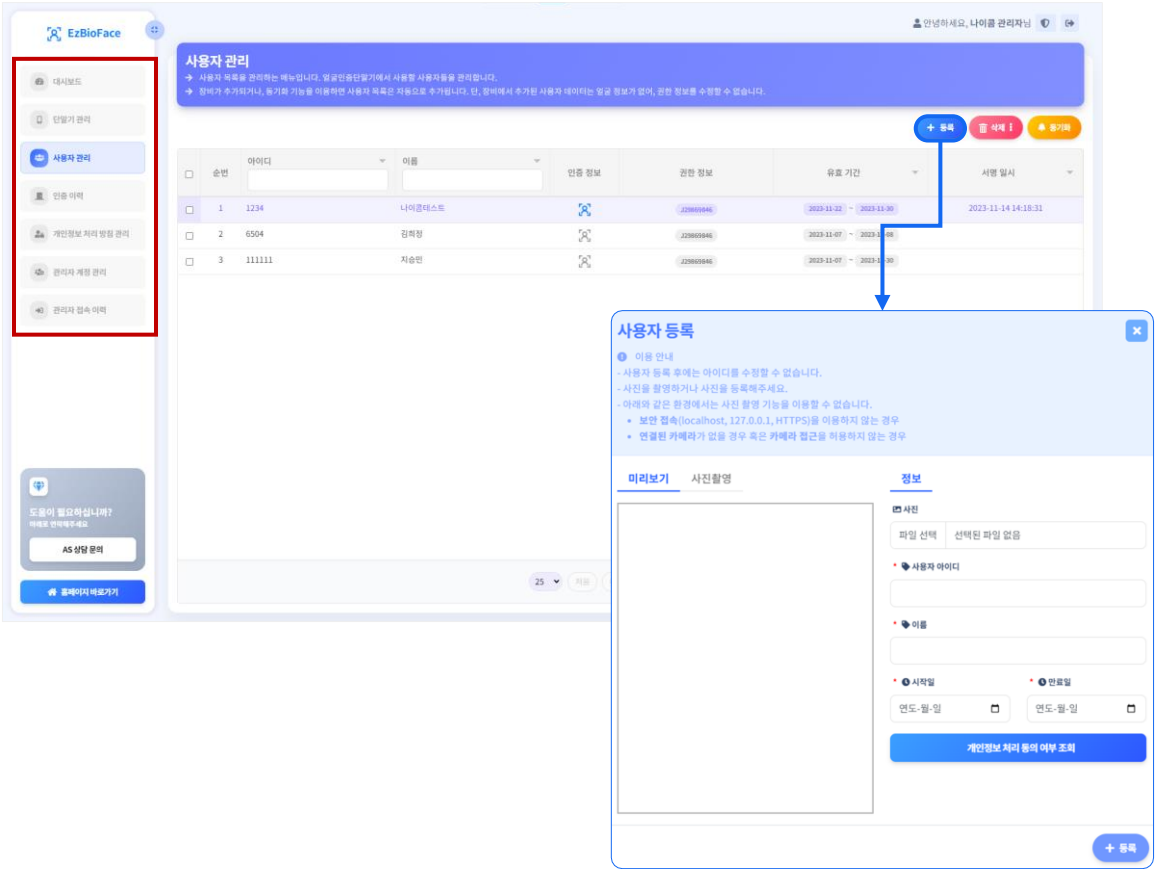
1. 관리자 페이지에서 **사용자 얼굴 정보 관리** (사진 업로드/웹캠 촬영)
2. **ActiveMQ**로 단말기 API와 통신하여 데이터 동기화(조회, 추가, 수정, 삭제)
3. 단말기에서 **얼굴 인증** 시 출입관리 시스템에서 모니터링 및 **게이트 제어**

# 시스템 구조



# 출입통제 관리자 프로그램

## 주요 화면



## 기능

### ※ 총 7개의 메뉴

대시보드, 사용자 관리, 장비 관리, 개인정보처리방침 관리, 관리자 관리, 관리자 접속 이력

### ※ 주요 기능

- 사용자 관리: 사용자 정보 관리.  
(+ 웹캠 기능을 이용하여 이용자 **얼굴 촬영**)
- 장비 연동: **ActiveMQ**로 얼굴인식 단말기에 이용자 정보 관리

### ※ 부가 기능

- 대시보드: 장비 상태, 인증 통계, 실시간 이력 등
- 장비 관리: 얼굴 인증 단말기 관리 기능 ⇒ **단말기 상태(ON/OFF) 추적**
- 인증 이력: **ActiveMQ**에서 **사용자 인증 이벤트** 발생시 이벤트를 기록하여 이력 및 대시보드 통계 데이터로 활용
- 개인정보 처리 방침 관리: 개인정보 처리 방침 내용 관리 (에디터)
- 관리자 관리: 시스템을 사용할 관리자 계정 관리
- 관리자 접속 이력 관리: 관리자가 접속한 이력 조회

# 출입통제 관리자 프로그램

## 고도화 과정

### ※ 초기

- 요구사항: 이용자 관리의 불편함 해소
- 목적: 빠른 시간 안에 제공
- ➔ 이용자 관리 기능만 제공

### ※ 요구사항 증가

초기 이후 약 3년 만에 발생한 시스템 셋업  
➔ 기존 프로그램 기능 추가 요구.

- 로그인 기능 추가
- 개인정보처리방침 동의 관리

### ※ 고도화의 늪

기존 프로젝트에 기능을 붙이는 것이 빠를까,  
신규 프로젝트를 구성하는 게 빠를까?

- 기술 스택의 변화
- 신입이 만든 알아보기 힘든 코드
- ➔ 신규 프로젝트를 구성하는 게 빠름!

## 주요 Point

### 1. 사용자 관리 프로세스 개선

- 초기: 동기적 방식
- ➔ 잦은 조회 실패
- 고도화: 비동기 및 이벤트 기반 처리
- ➔ 조회 실패 문제와 UX 개선

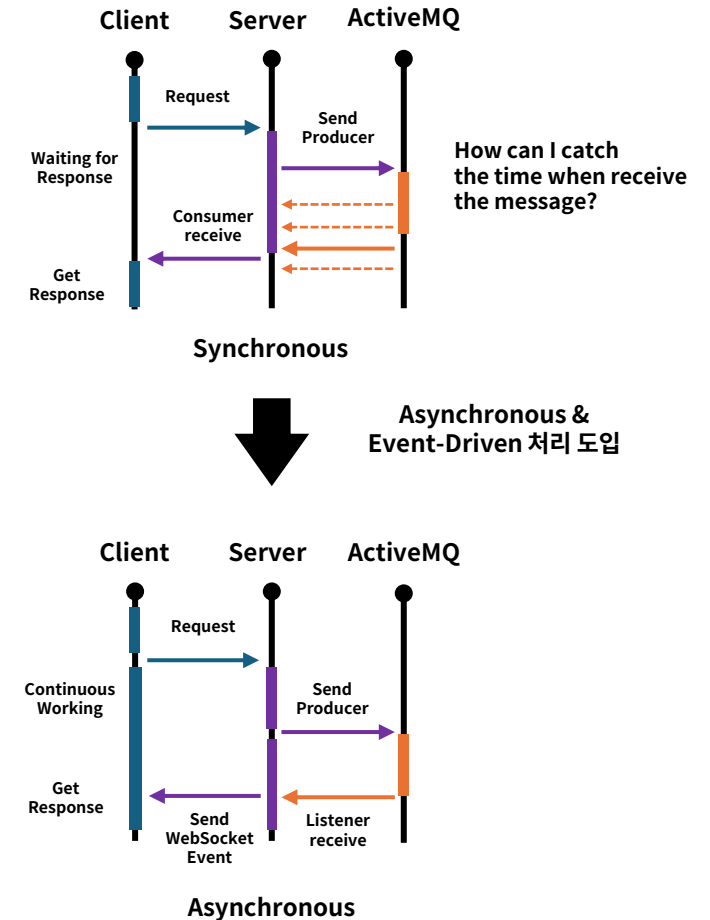
### 2. 기술 스택의 변화

- 초기: Java 1.8, Spring MVC 5.x, MyBatis
- 고도화: Java 1.8, Spring Boot 2.x, JPA (+ QueryDSL)

### 3. 기능 증가

더욱 풍부해진 기능으로 사용성 증가

- 초기: 사용자 관리
- 고도화: 사용자 관리, 장비 관리, 관리자 관리 (로그인), 개인정보처리방침 관리/동의, 대시보드, 인증 이력 등



## 저의 핵심역량은

1

SDLC 전반에 걸친  
애플리케이션 개발

2

원활한  
커뮤니케이션 능력

3

지속적인 신기술  
학습, 도입, 공유

# 핵심 역량

1

## 전반적인 애플리케이션 개발 능력

소프트웨어 생명주기(SDLC) 전반에 걸친 프로젝트 경험을 10회 이상 거듭함으로써 하나의 애플리케이션을 전반적으로 개발할 수 있는 능력을 갖추었습니다.

- 요구사항을 통해 Figma로 화면/기능 설계
- ERD와 ORM(JPA)를 이용한 DB 및 객체 모델링
- Spring Project 세팅부터 API 개발, 빌드/배포
- 퍼블리싱부터 프론트엔드 개발

2

## 원활한 커뮤니케이션 능력

팀 내 타 파트(키오스크, 디자인)와 협업하고, 파트 내 프로젝트의 A-Z까지 주도한 경험이 있습니다.

협업 경험과 다양한 직군(기획자, 디자이너, 프론트엔드, 백엔드 등)을 소화해낸 경험을 바탕으로 성공적인 협업을 이끌어 낼 수 있습니다.

3

## 지속적인 신기술 학습, 도입, 공유

초기에는 기존 코드 기반으로 Spring MVC, MyBatis, JSP의 웹 애플리케이션 개발 능력 향상에 집중하였습니다.

어느정도 기술 스택이 익숙해졌을 때에는 리스크를 줄이기 위해 가장 작은 프로젝트부터 신기술 적용을 위해 노력했습니다.

- 기존 Session 기반 인증 → Spring Security 도입
- Spring Legacy → Boot 전환  
Template Engine: JSP → Thymeleaf
- MyBatis → JPA (+ QueryDSL) 전환  
프로그래밍 스타일 전격 변화 (Map → 객체)

지속해서 신기술을 도입하면서 필연적으로 코드 스타일이 혁신적으로 변하게 되었고 새로운 인사이트를 얻을 수 있었습니다.

얻은 인사이트를 주제로 블로그에 포스팅함으로써 지식 정제와 인사이트 공유의 이점을 얻을 수 있었습니다.

➔ 신입으로 입사하여 발전할 수 있었던 이유



Thank you for reading

**감사합니다**

**Contact**

010-5705-9594  
developer.heejeong@gmail.com