

MZ세대 만족도 분석 프로젝트

프로젝트 수행 결과서

조민정



| 목차

01 프로젝트 기획서

- 기획 의도
- 개발 목표

02 개발 환경 및 사용 기술 경험

- 사용 기술 목록표 (환경/기술/도구)
- 사용 기술 경험

03 개발 스케줄표

- Project I
- Project II
- Project III

04 빅데이터 분석 결과서

- 데이터 수집
- 데이터 전처리 및 검증
- 모델링 및 분석 결과
- 주요 시각화

05 요구사항 정의서

06 화면 설계서

07 UML

- Usecase Diagram
- Sequence Diagram
- Class Diagram

08 ER Diagram

09 주요 서비스 기능 및 소스 코드

- 주요 서비스 기능
- 소스코드

10 Software Architecture

11 시연

12 향후 계획 및 프로젝트 수행 소감

MZ세대 만족도 분석 프로젝트

My Zeneration : 내가 속한 나의 MZ세대는 어떤 모습일까?

서비스 소개

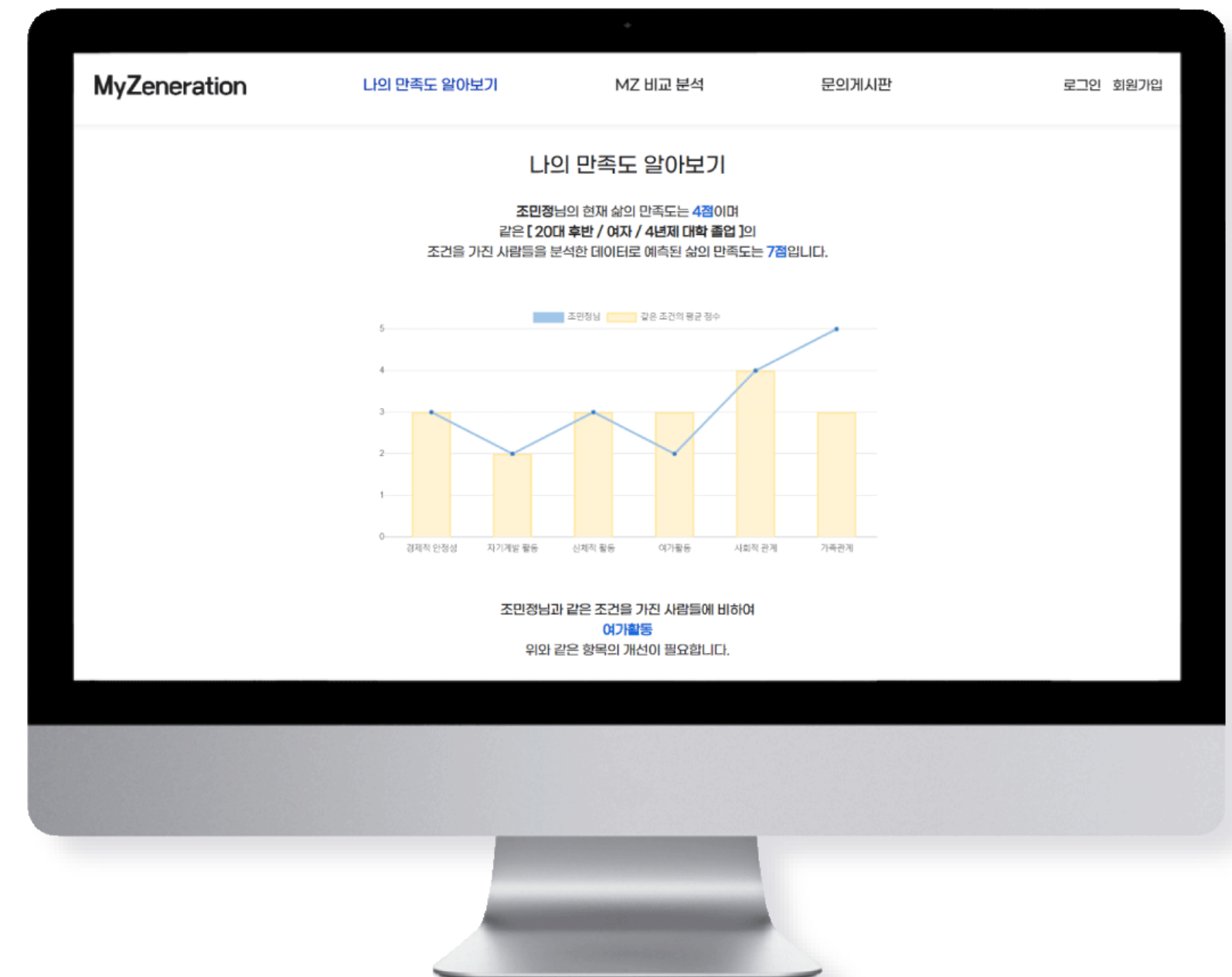
20·30대 MZ세대의 환경과 삶의 만족도를 분석하여
만족도 증대를 위한 방법을 추천하는 서비스

개발 기간

Project I (프로젝트 기획, 빅데이터 수집 및 분석, UI 개발)
2023.05.23 ~ 2023.06.15

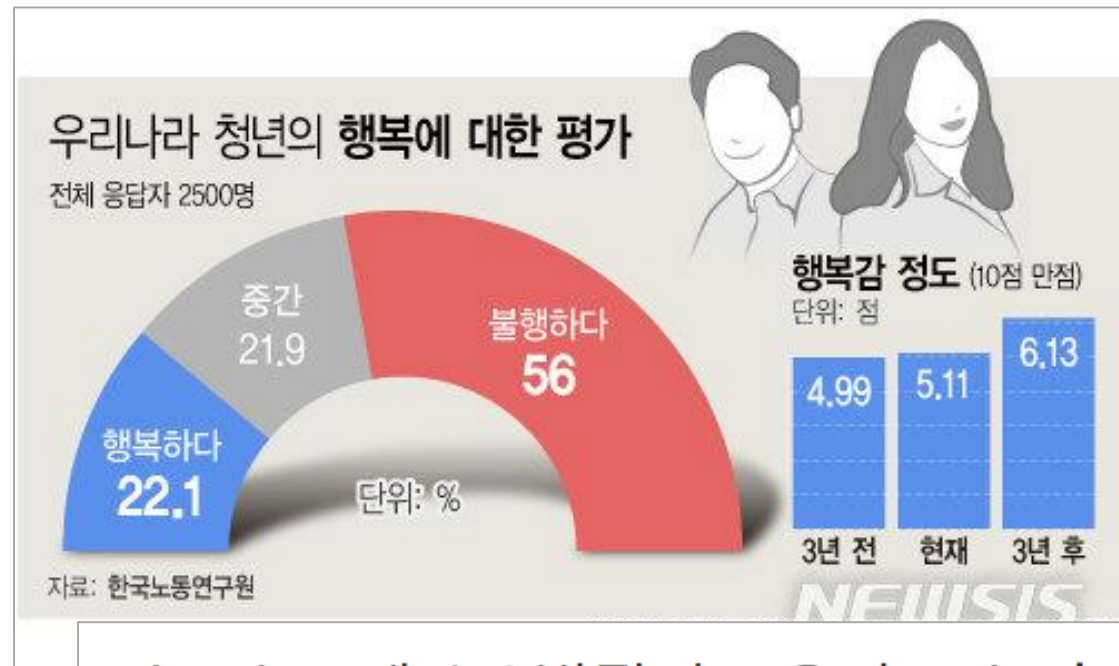
Project II (Servlet & JSP 프로그램 개발)
2023.07.13 ~ 2023.07.26

Project III (Spring Framework 프로그램 개발)
2023.08.07 ~ 2023.08.18



1. 프로젝트 기획서

기획 의도



인스타 볼 때마다 박탈감·우울감... 나도 '카페인 우울증'?

[기자수첩] 보여주기식 소비와 전시 당신은 행복한가?



#Flex #YOLO #해외여행 #결혼 #취업

다양한 소통 플랫폼의 존재감이 커져가는
현 시대를 살아가는 2·30대

MZ세대

SNS 세상 속 전시되어 있는 여러 사람들의
화려한 일상을 반복해서 접하게 되면서

타인에 대한 관심도가 자연스럽게 **증가**
자신과 끊임없이 비교하며 **상대적인 박탈감**을 느낌



다양한 요인에 의해 낮아지는 삶의 만족도

1. 프로젝트 기획서

기획 의도

한국보건사회연구원에서 주관하는 청년 삶 실태 조사

청년층의 **삶의 실태와 특성** (성별, 연령(20~39세), 학력, 사회적 관계, 자기계발 활동 횟수 등)과 **삶의 만족도**가 포함되어 있는 **빅데이터를 분석**

**삶의 만족도에 높은 영향력을
갖는 요인 파악**

개인마다 도출된 분석 결과를 기반으로
MZ세대의 삶의 만족도를 증진시키기 위한 방안을 제안하고자 한다.

1. 프로젝트 기획서

개발 목표

● 빅데이터 분석을 통한 회귀 모델 도출 (사용자 조건 - 삶의 만족도)

[설문지]



2022년 한국노동패널조사

이름	
일자리 번호	가

다음은 지난 조사 당시(____년 ____월 ____일)에 다니셨고 현재도 계속 다니시는 일자리(____)에 대한 질문입니다. 지금부터는 위에 적힌 일자리만 생각하시면서 응답하여 주십시오.

응답자 ID

2022년 청년 삶 실태조사

통계법 제33조(비밀의 보호)

국무조정실에서는 청년층의 삶의 실태와 특성, 욕구 및 인식을 파악하여 다양한 청년들의 삶의 질을 제고할 수 있는 정책수립의 기초자료를 마련하고자 「2022년 청년 삶 실태조사」를 실시하고 있습니다. 이번 조사는 청년기본법 제11조 (청년 실태조사 등)에 따라 실시되는 것으로 조사에 적극 협조해 주시기 바랍니다. 감사합니다.

* 응답 내용은 통계법 제33조(비밀의 보호) 및 제39조(벌칙)에 의해 철저히 보호되고 있습니다.

[조사기반 데이터]

	가구원수	성별	연령	지역구분	시도코드
1	4	2	2	1	11
2	3	2	2	1	11
3	1	2	3	1	11
4	3	1	3	1	23
5	4	2	1	1	11
6	1	2	2	1	11
~	~	~	~	~	..

한국보건사회연구원에서 주관하는 청년 삶 실태 조사 데이터와 한국노동연구원에서 1만 3000여 가구를 추적 조사한 데이터
R 프로그램으로 데이터 분석을 통해 삶의 특성과 삶의 만족도 간의 **회귀 모델을 얻음**



사용자의 삶의 만족도에 **가장 큰 영향을 미치는 환경 조건 상위 3개를 파악하여 개선 방안으로 추천**

개발 목표

1 개인의 만족도 분석

사용자의 정보 (연령, 성별, 최종학력 등) 와
삶의 실태 (경제적 안정성, 주기적 운동 횟수, 사회적 관계 등),
본인이 느끼는 삶의 만족도를 입력

입력 정보와 같은 연령, 성별, 학력을 가진 사람들을 추출하여
데이터 분석 후 회귀 모델에 적용해 산출된 삶의 만족도와
실제 사용자가 입력한 삶의 만족도를 비교

&

사용자의 현재 환경에서
만족도에 높은 영향을 미치는 요인 상위 3가지를 도출하여
삶의 만족도 증진을 위한 방안을 제안

삶의 만족도?

#25살
#남자
#대학교졸업
#경기도거주



여가 활동?

자기계발?

사회관계?

2 비교 기준에 따른 MZ세대의 통계

성별, 연령, 학력, 거주지역, 취업여부, 삶의 만족도
위의 항목 중 비교해 보고자 하는

항목을 선택



선택된 항목을 기준으로
만족도(삶, 여가생활, 직무), 취업률, 월 임금, 근무시간 등과 같은

통계를 그래프로 보여줌

2. 개발 환경 및 사용 기술 경험

사용 기술 목록표

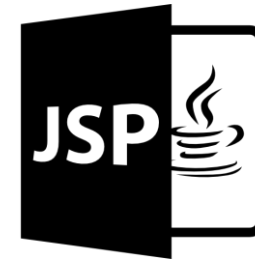
기술



[Java]
1.8.0_271



[R]
3.6.3



[JSP]
2.3



[HTML5, CSS3, Javascript]

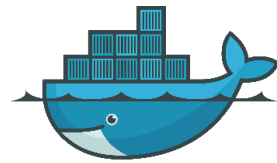
서버



Apache
Tomcat
9.0.78



[AWS EC2]



docker
[docker]

데이터베이스



[Oracle Database]
11.2.0.2.0

개발 환경



[Spring Framework]
4.3.9



[Spring Boot]
4.7.0



[eGovFrame]
3.9.0

eclipse



[Eclipse]
Photon Release (4.8.0)



[VS Code]

2. 개발 환경 및 사용 기술 경험

사용 기술 경험

Java

- Java 1.8을 기반으로 미니 프로젝트 구현
- 주소록 관리 프로그램
 - 연락처 저장, 검색, 수정, 삭제, 출력
 - File 클래스를 이용하여 파일로 저장
 - 자바 컬렉션 프레임워크 사용
- Single Linked List 구현
 - 노드 삽입
 - 노드 삭제
 - 저장되어 있는 노드 역순으로 바꾸기
- 선택 정렬 (Selection Sort) 알고리즘
- 은행 계좌 관리 프로그램
- 개인 프로젝트
 - R 프로그램 연동

Spring Framework

- SLF4J를 이용한 Logging
- Spring IoC
- Spring DI
 - XML을 이용한 DI
 - Annotation을 이용한 DI
- Spring AOP
 - XML기반의 POJO 클래스 이용한 AOP 구현
 - AspectJ의 @Aspect annotation 기반 AOP 구현
- Spring JDBC
 - JDBC 설정, 트랜잭션 처리
 - 연결 정보 암호화 (Jasypt)
- Spring MVC Pattern
 - 회원 관리 프로그램 실습
 - 게시판 프로그램 실습
- MyBatis
 - Mapper XML

Spring Boot

- Lombok을 이용한 데이터 모델 핸들링
- 폼 데이터 유효성 검증
 - Validator, ValidationUtils, initBinder, Valid annotation
- JDBC Template

R

- 공공 API를 이용한 데이터 분석 실습
- 데이터 전처리
 - 결측값, 특이값 처리
- 분석
 - 단순선형 회귀분석
 - 다중선형 회귀분석
 - 로지스틱 회귀분석
- 시각화
 - ggplot2 패키지를 이용한 시각화
- 프로젝트
 - 다항 로지스틱 회귀 분석을 실시하여 회귀 모델을 이용한 예측값 도출

DataBase

- SQL과 DBMS
- 관계 데이터 모델
 - 무결성 제약조건
 - 관계대수
- SQL 내장함수
- 부속질의
 - 중첩질의
 - 스칼라 부속질의
 - 인라인 뷰
- 인덱스
 - 오라클 B-tree 인덱스
- PL/SQL
- ER 모델
- 정규화 (Nomalization)
- 데이터 백업

Python

- 파이썬 함수 정의
- 람다(Lambda) 함수
- 파이썬으로 텍스트 파일 처리
- 표준 모듈과 외부 모듈 활용
- Matplotlib을 이용한 시각화

2. 개발 환경 및 사용 기술 경험

사용 기술 경험

Javascript

- 이벤트 핸들러와 리스너 활용
- DOM트리와 노드에 대한 이해
- Ajax를 이용하여 클라이언트에서 서버에 데이터를 비동기식으로 요청
- chart.js를 이용해 그래프 출력
- 회원가입 페이지 사용자 입력 유효성 검사

JSP

- MZ세대 만족도 분석 서비스 (Project II, III) 웹 사이트 구현
- JSTL을 이용한 데이터 처리 및 출력

HTML5/CSS3/Bootstrap

- MZ세대 만족도 분석 서비스 (Project I) UI 구현
- 반응형 페이지 제작
- Bootstrap을 이용한 게시판 화면 제작

AWS EC2

- AWS EC2 환경 setup
- AWS EC2 환경에 Ubuntu 설치
- puTTY를 이용한 인스턴스 접속
- FileZila를 이용하여 AWS EC2 인스턴스에 파일 전송

Linux

- Ubuntu 기반 Linux 개발 환경 구축 실습
- 기본 명령어 이해 및 실습
 - 디렉토리 관련 명령 실습
 - 파일 관련 명령 실습

React

- SPA (Single Page Aechitecture)에 대한 이해
- CRA를 통한 리액트 프로젝트 세팅
- Jsx를 사용하여 코드 작성
- 클래스형 컴포넌트
 - state
 - Life cycle method
- 함수형 컴포넌트
 - Hooks를 이용한 Data Fetching

MSA

- 클라우드 서비스에 대한 이해
- Micro Service Architecture에 대한 이해
 - CI/CD
 - RESTful API
- MSA 설계에 대한 이해
 - 도메인 주도 설계 (DDD)
 - 단일 책임 원칙
 - 가벼운 통신 프로토콜
 - 독립된 데이터 저장소
- MSA 표준 구성 요소

Docker

ELK

Git

- 프로젝트 관련 소스코드 및 파일 관리를 위한 Git 명령어 사용
- Git을 이용한 협업

개발 스케줄표

프로젝트	업무 분장	이름	번호	세부항목	시작일	종료일	작업기간 (공휴일 제외)
Project I	기획	공통	1	아이디어 구상	2023-05-23	2023-05-25	3
			2	기획서 초안작성	2023-05-25	2023-05-26	2
			3	기획서 중간발표	2023-05-30	2023-05-30	1
		이준형	4	기획서/요구사항 정의서 작성	2023-05-31	2023-06-03	4
		윤희선	5	와이어프레임, 화면설계, 명세서 작성	2023-06-05	2023-06-09	5
		공통	6	1차 프로젝트 결과 발표	2023-06-23	2023-06-23	1
	빅데이터분석	공통	7	데이터 수집	2023-05-29	2023-06-03	6
		조민정	8	데이터분석 및 전처리	2023-06-07	2023-06-15	9
		이준형	9	데이터 시각화	2023-06-15	2023-06-17	3
		조민정	10	분석보고서 작성	2023-06-19	2023-06-22	4
		조민정	11	데이터 베이스 설계	2023-06-17	2023-06-18	1
	UI	조민정	12	메인 화면 개발	2023-06-15	2023-06-16	2
		윤희선	13	개인 만족도 분석 기능 화면 개발 - 나의 만족도 알아보기	2023-06-11	2023-06-14	4
		윤희선	14	조건 별 만족도 비교 분석 기능 화면 개발 - MZ 비교 분석	2023-06-15	2023-06-18	4
		이준형	15	문의하기 게시판 화면 개발	2023-06-12	2023-06-15	4

개발 스케줄표

프로젝트	업무 분장	이름	번호	세부항목	시작일	종료일	작업기간 (공휴일 제외)
Project II	Servlet/JSP Programming	조민정	17	UML 작성	2023-07-13	2023-07-14	2
			18	개인 만족도 분석 - 사용자 입력 정보 관리 (회원/비회원) 구분 기능 구현	2023-07-12	2023-07-12	1
			19	개인 만족도 분석 - 사용자 조건에 따른 예상 만족도와 실제 만족도 비교 기능 구현	2023-07-15	2023-07-15	1
			20	개인 만족도 분석 - 삶의 만족도 향상 방안 추천 기능 구현	2023-07-17	2023-07-18	2
			21	기준 별 통계분석 - 선택 기준에 따른 통계 분석 기능 구현	2023-07-19	2023-07-19	1
			22	기준 별 통계분석 - 분석 결과 시각화 기능 구현	2023-07-20	2023-07-21	2
			24	회원관리 (가입, 아이디/비밀번호 찾기, 탈퇴) 기능 구현	2023-07-22	2023-07-23	2
			25	로그인 기능 구현	2023-07-23	2023-07-23	1
			26	회원 정보 조회 및 수정 기능 구현	2023-07-24	2023-07-25	2
			27	개발 결과물 발표	2023-07-26	2023-07-26	1
Project III	Spring Framework Programming	조민정	28	Spring Framework 개발 환경 세팅	2023-08-07	2023-08-07	1
			29	Bean 설정	2023-08-08	2023-08-08	1
			30	DI 주입	2023-08-09	2023-08-09	1
			31	Spring JDBC 적용	2023-08-10	2023-08-10	1
			32	Spring MVC 프로젝트 개발	2023-08-11	2023-08-25	14
			33	프로젝트 수행 결과서 작성	2023-08-28	2023-09-01	5
			34	개발 결과물 발표	2023-09-05	2023-09-05	1

5. 요구사항 정의서

요구사항 정의서 (1/2)

목 차

1. 기능적 요구사항	2
1.1 메인페이지 기능	2
1.1.1 메뉴 이동 기능	2
1.1.2 텍스트 애니메이션 기능	2
1.2 개인 만족도 분석 기능.....	2
1.2.1 회원/비회원 선택 기능	2
1.2.2 정보 입력 기능	3
1.2.3 삶의 만족도 분석 결과 조회 기능	3
1.2.4 삶의 만족도 향상 방안 추천 기능	3
1.3 요인 간 비교 분석 기능	3
1.3.1 비교 항목 선택 기능	3
1.3.2 비교 분석 결과 조회 기능	4
1.4 문의 게시판 기능.....	4
1.4.1 문의 게시판 글 작성 기능	4
1.4.2 답변 작성 기능	4
1.4.3 문의 글 조회 기능.....	4
1.5 회원 관리 기능	4
1.5.1 회원 조회 기능	4
1.5.2 마이페이지 기능.....	5
1.5.3 회원 가입 기능	5

1. 기능적 요구사항

1.1 메인 페이지 기능

1.1.1 메뉴 이동 기능

- 상단에 메뉴 바를 고정시켜 각 메뉴 클릭 시 해당 메뉴 링크로 페이지를 이동할 수 있어야 한다.
- 회원 메뉴의 경우 로그인 전에는 로그인, 회원가입 / 로그인 후에는 마이 페이지, 로그아웃 버튼을 노출시킬 수 있어야 한다.
- 메인 페이지가 아닌 각각의 메뉴로 페이지가 이동되었을 때 활성화 된 메뉴를 구분할 수 있도록 표시할 수 있어야 한다.

1.1.2 텍스트 애니메이션 기능

- 서비스의 목표와 관련된 키워드를 텍스트 애니메이션으로 제작하여 서비스가 추구하고자 하는 방향을 제시할 수 있어야 한다.
- 메인 페이지 하단의 버튼을 클릭하면 ‘나의 만족도 알아보기’ 페이지로 이동할 수 있어야 한다.

1.2 개인 만족도 분석 기능

1.2.1 회원/비회원 선택 기능

- 기능 실행을 위해서 몇 가지 정보를 입력해야 한다는 것을 사용자가 알 수 있어야 한다.
- 기능을 실행하기 전 회원으로 진행할 것인지 비회원으로 진행할 것인지 선택할 수 있어야 한다. 단, 이미 접속이 되어있는 회원에게는 선택 팝업을 노출시키지 않는다.
- 비회원으로 진행을 선택할 경우 페이지를 나가게 되면 입력된 정보는 저장되지 않으며 분석 결과 또한 초기화 된 다는 것을 알 수 있어야 한다.
- 회원으로 진행을 선택할 경우 접속이 되어 있지 않은 사용자는 로그인 페이지로 이동하여 접속을 할 수 있어야 한다.

- 회원으로 진행을 선택할 경우 입력된 정보는 회원 정보에 저장된다는 것을 알 수 있어야 하고, 회원 정보 수정 페이지에서 수정을 할 수 있어야 한다.

- 팝업 닫기 버튼을 누르면 메인페이지로 이동할 수 있어야 한다.

1.2.2 정보 입력 기능

- 사용자는 본인의 이름, 성별, 생년월일, 학력, 연봉을 입력할 수 있어야 한다.
- 사용자는 경제적 안정성, 가족관계, 사회적 관계를 5점 척도로 선택할 수 있어야 한다.
 - 1점 (좋지 않음), 2점 (대체로 좋지 않음), 3점 (보통), 4점 (대체로 좋음), 5점 (좋음)
- 사용자는 한 달을 단위로 자기계발 활동, 신체적 활동, 여가 활동의 횟수를 선택할 수 있어야 한다.
 - 1점 (5회 이하), 2점 (6~10회), 3점 (11~15회), 4점 (16~20회), 5점 (21회 이상)
- 사용자는 현재 본인의 삶의 만족도를 10점 척도로 선택할 수 있어야 한다.
- 입력 항목은 모두 필수로 입력되어야 하고, 입력되지 않은 항목은 필수 입력 사항임을 표시할 수 있어야 한다.
- 모든 정보 입력 후 만족도 분석 결과 보기 버튼을 누를 시 입력된 정보를 토대로 기능을 실행할 수 있어야 한다.
- 팝업 닫기 버튼을 누르면 메인페이지로 이동할 수 있어야 한다.

1.2.3 삶의 만족도 분석 결과 조회 기능

- 사용자가 입력한 정보와 만족도 점수를 표시할 수 있어야 한다.
- 빅데이터 분석으로 얻은 회귀 모델에 만족도를 제외한 사용자의 조건을 대입하여 같은 조건의 사람들의 만족도를 도출하고, 그 결과를 표시할 수 있어야 한다.

1.2.4 삶의 만족도 증가 방안 추천 기능

- 사용자의 삶의 만족도에 영향을 미치는 요인과 각 요인 별 가중치를 부여하여 그 수치와 항목별 차이를 시각적으로 표시할 수 있어야 한다.
- 사용자의 삶의 만족도에 영향을 미쳤던 요인 중 상위 3개를 순위 형태로 나열할 수 있어야 한다.

5. 요구사항 정의서

요구사항 정의서 (2/2)

1.3 요인 간 비교 분석 기능

1.3.1 비교 항목 선택 기능

- 사용자는 비교해보고자 하는 요인을 한 가지만 선택할 수 있어야 한다.
- 선택 가능한 요인을 좌측 사이드에 고정시켜 표시할 수 있어야 한다.
- 요인은 성별(남, 여), 연령(20대 초반, 20대 중반, 20대 후반, 30대 초반, 30대 중반, 30대 후반), 거주지역(수도권, 강원, 충청, 영남, 호남, 제주), 학력(고등학교 졸업 미만, 고등학교 졸업, 대학교 재학 또는 중퇴, 전문대학교 졸업, 4년제 대학교 졸업 이상), 취업 여부(취업, 미 취업), 삶의 만족도 총 6개로 표시할 수 있어야 한다.
- 항목을 클릭 시 바로 적용된 그래프가 표시되어야 한다.

1.3.2 비교 분석 결과 조회 기능

- 선택한 요인을 비교 항목으로 적용해 분석한 결과를 시각화하여 그래프로 표현할 수 있어야 한다.
- 삶의 만족도 / 여가생활 만족도 / 직무 만족도 / 취업률 / 고용형태(정규직 / 비정규직) / 월평균 임금 / 주 평균 근로시간 / 혼인율 총 8개의 항목을 그래프로 나타낼 수 있어야 한다.
 - 삶의 만족도: line 그래프
 - 여가생활 만족도: bar 그래프
 - 직무 만족도: bar 그래프
 - 취업률: bar 그래프
 - 고용형태: line 그래프
 - 월 평균 임금: bar 그래프
 - 주 평균 근로시간: bar 그래프
 - 혼인율: pie 그래프
- 선택 기준에 따라 결과 그래프를 유동적으로 보여줄 수 있어야 한다.
 - 선택 기준으로 취업 여부를 선택할 경우: 결과 그래프로 직무 만족도, 취업률, 고용형태, 월 평균 임금, 주 평균 근로시간은 표시하지 않는다.
 - 선택 기준으로 삶의 만족도를 선택할 경우 : 결과 그래프로 삶의 만족도는 표시하지 않는다.
- 분석 결과를 수치화하여 이해하기 쉽도록 나타낼 수 있어야 한다.

1.5 회원 관리 기능

1.5.1 회원 가입 기능

- 사용자가 회원가입을 하고자 할 때 본 서비스가 요구하는 몇 가지 정보를 입력하여 가입을 할 수 있어야 한다.

1.5.2 회원 접속 기능

- 기존 회원들은 아이디, 비밀번호를 입력하여 접속을 할 수 있어야 한다.
- 사용자가 아이디를 잊은 경우 정보를 입력하여 아이디를 찾을 수 있어야 한다.
- 사용자가 비밀번호를 잊은 경우 정보를 입력하여 비밀번호를 재설정할 수 있어야 한다.

1.5.3 회원 정보 수정 기능

- 사용자가 회원 정보를 수정하고자 할 때 수정할 수 있어야 한다.

1.5.4 회원 탈퇴 기능

- 사용자는 회원 탈퇴를 하고자 할 때 회원 탈퇴를 할 수 있어야 한다.

화면 설계서

1) 나의 만족도 알아보기

1-1) 사용자 정보 입력 팝업

만족도 분석을 위해 필요한 정보를 사용자에게 입력 받기 위한 팝업 창입니다.
비회원도 진행이 가능하며, 로그인 후 최초 정보 입력 시 해당 정보는 저장되어 재입력을 하지 않고 바로 서비스 이용이 가능함.

프로젝트명	MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	나의 만족도 알아보기 → 정보 입력 팝업	페이지 #	3
화면경로	나의 만족도 알아보기				

MZ세대 분석서비스

www.mygeneration.com

로그인 마이페이지 회원가입

나의 정보 입력

이름

1입력

5이름을 입력해 주세요

2학력

3경제적 안정성

4사회적 관계

5가족 관계

6

No.	Description
1	이름, 성별(남,여), 생년월일 - 필수 입력 * 회원 최초 진입 시 회원 정보 출력 (수정 불가)
3	- 모든 사항 필수 선택 - 학력 선택 (고등학교 졸업 미만/고등학교 졸업/전문 대학 재학/전문대학 졸업/대학교 재학/대학교 졸업/대학원 졸업 이상) - 경제적 안정성, 사회적 관계, 가족 관계 : 5점 척도로 선택
4	필수 항목 입력하지 않은 경우 빨간색 텍스트로 노출
5	닫기 버튼 클릭 시 메인 화면으로 이동

MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	나의 만족도 알아보기 → 정보 입력 팝업	페이지 #	4
나의 만족도 알아보기				

서비스

www.mygeneration.com

로그인 마이페이지 회원가입

1자기계발 활동

2현재 삶의 만족도를 10점 척도로 입력해 주세요

3

No.	Description
1	- 모든 사항 필수 선택 - 각 활동은 한 달을 기준으로 1점 (5회 이하) , 2점 (6~10회) , 3점 (11~15회) , 4점 (16~20회) , 5점 (21회 이상) 횟수로 선택 - 자기 계발을 위한 활동 - 신체적 활동 - 여가 활동
2	사용자의 현재 삶의 만족도 10점 척도로 선택 - 필수 선택
3	- 모든 정보 입력 되었을 경우 정보 저장하여 기능 실행 - 모든 정보 입력되지 않은 경우 입력이 필요한 사항 메시지 노출시켜서 알림
4	필수 항목 입력하지 않은 경우 빨간색 텍스트로 노출

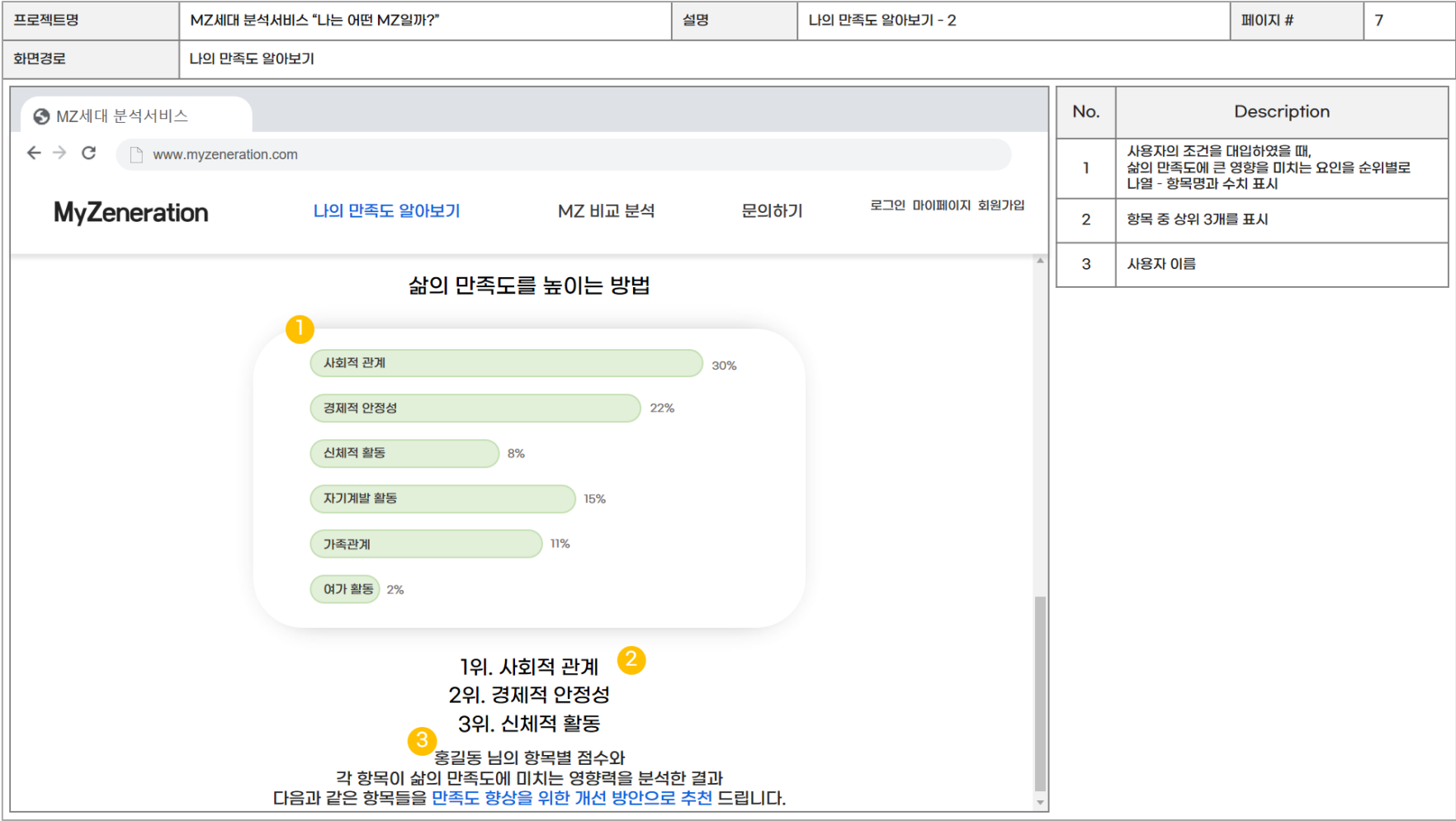
1) 나의 만족도 알아보기

1-2) 만족도 분석 결과 화면

사용자가 입력한 삶의 만족도 점수와
입력된 6가지 항목의 점수를 대입하여 예측한 삶의 만족도를 비교하여
사용자가 한 눈에 알아보기 쉽도록 개선이 필요한 항목을 시각화하여 나타냄



총 6개의 항목이 사용자의 삶의 만족도에 영향을 미치는 정도를 보여주고,
상위 3개의 항목을 만족도 향상을 위한 개선 방안으로 추천



2) MZ세대 통계 분석

사용자가 비교하고자 하는 항목을 선택하면
해당 항목을 기준으로 통계 분석한 결과를 시각화 하여 수치와 함께 확인할 수 있음

프로젝트명	MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	MZ비교분석 페이지 - 1	페이지 #	9
화면경로	MZ비교분석				

MZ세대 분석서비스		No.	Description
← → ↺ www.mygeneration.com			
MyZeneration			
나의 만족도 알아보기			
MZ 비교 분석			
문의하기			
로그인 마이페이지 회원가입			
비교 분석 하고자 하는 항목을 선택해 주세요		1	6개의 선택사항 중 1개의 항목만 선택 가능 1) 성별 (남성/여성) 2) 연령 (20대초/중/후, 30대초/중/후) 3) 학력 (중학교 졸업 이하/고등학교 졸업/전문대학 졸업/4년제 대학 졸업 이상) 4) 거주지역 (수도권/강원/충청/호남/영남/제주) 5) 취업여부 (취업/미취업) 6) 삶의 만족도 항목 클릭 시 바로 그래프 노출
성별		2	조회 버튼 클릭 시 선택한 항목을 기준으로 비교 분석 그래프 노출
연령		3	8개 항목을 그래프로 출력 1) 삶의 만족도 2) 여가생활 만족도 3) 직무만족도 4) 취업률 5) 고용형태(정규직 / 비정규직) 6) 월평균 임금 7) 주 평균 근로시간 8) 혼인율
학력			
거주지역			
취업여부			삶의 만족도 - Line 그래프 - 기준별 비중이 높은 항목과 수치 출력
삶의 만족도		4	각 항목마다 비교 결과를 수치화해서 문자로 출력

프로젝트명	MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	MZ비교분석 페이지 - 2	페이지 #	10
화면경로	MZ비교분석				

MZ세대 분석서비스		No.	Description
← → ↺ www.mygeneration.com			
MyZeneration			
나의 만족도 알아보기			
MZ 비교 분석			
문의하기			
로그인 마이페이지 회원가입			
비교 분석 하고자 하는 항목을 선택해 주세요		1	여가생활 만족도 단위 : %
성별		2	직무 만족도 단위 : %
연령			
학력			
거주지역			
취업여부			
삶의 만족도			

프로젝트명	MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	MZ비교분석 페이지 - 4	페이지 #	12
화면경로	MZ비교분석				

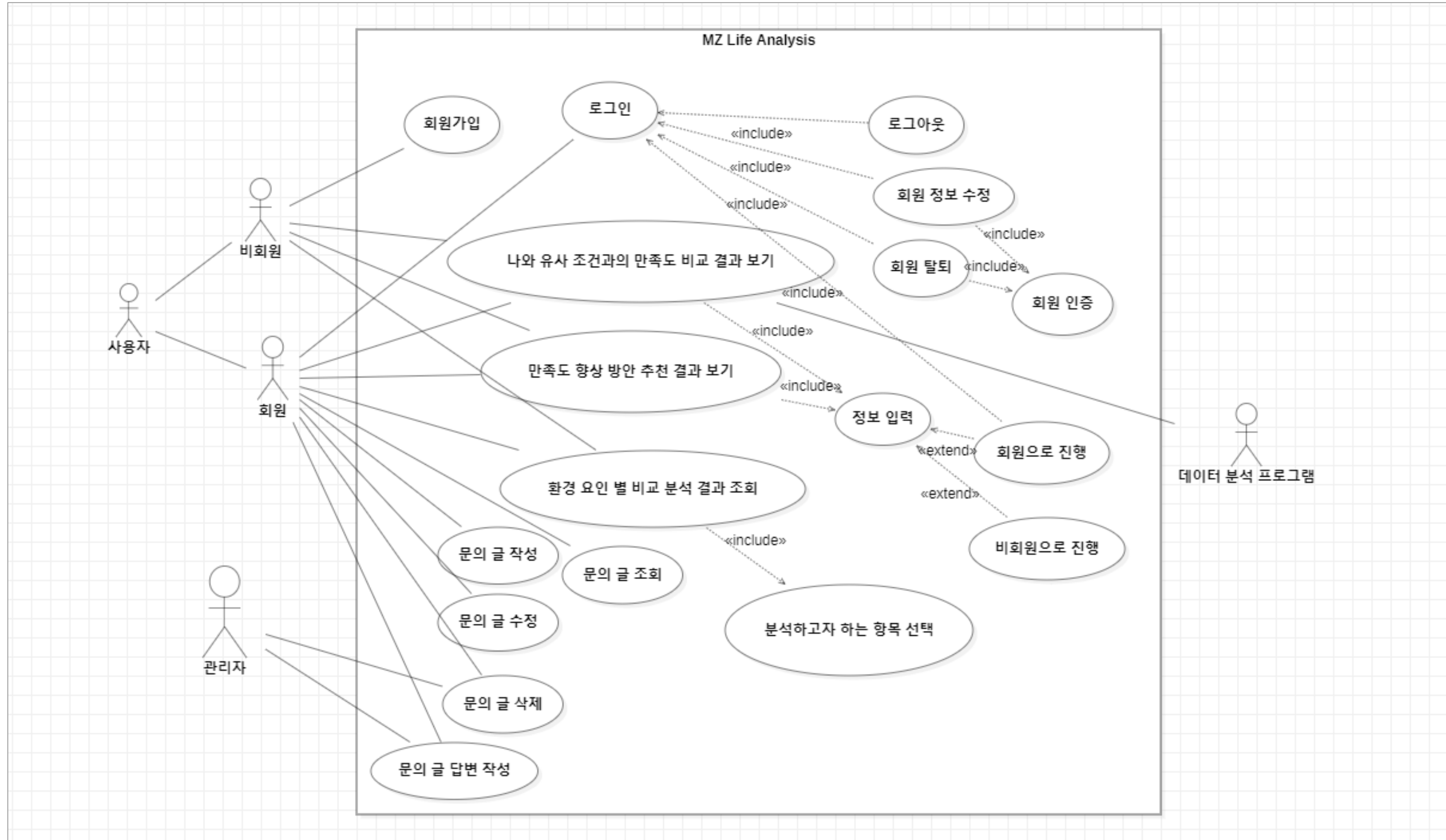
MZ세대 분석서비스		No.	Description
← → ↺ www.mygeneration.com			
MyZeneration			
나의 만족도 알아보기			
MZ 비교 분석			
문의하기			
로그인 마이페이지 회원가입			
비교 분석 하고자 하는 항목을 선택해 주세요		1	월 평균 임금 결과 출력 - Bar 그래프 - 평균 수치
성별		2	주 평균 근로시간 결과 출력 - Bar 그래프 - 평균 수치
연령			
학력			
거주지역			
취업여부			
삶의 만족도			

프로젝트명	MZ세대 분석서비스 "나는 어떤 MZ일까?"	설명	MZ비교분석 페이지 - 5	페이지 #	13
화면경로	MZ비교분석				

MZ세대 분석서비스		No.	Description
← → ↺ www.mygeneration.com			
MyZeneration			
나의 만족도 알아보기			
MZ 비교 분석			
문의하기			
로그인 마이페이지 회원가입			
비교 분석 하고자 하는 항목을 선택해 주세요		1	취업률 분석 결과 출력 - Pie 그래프 - 많은 비율을 차지하는 항목과 수치
성별			
연령			
학력			
거주지역			
취업여부			
삶의 만족도			

7. UML

Usecase Diagram



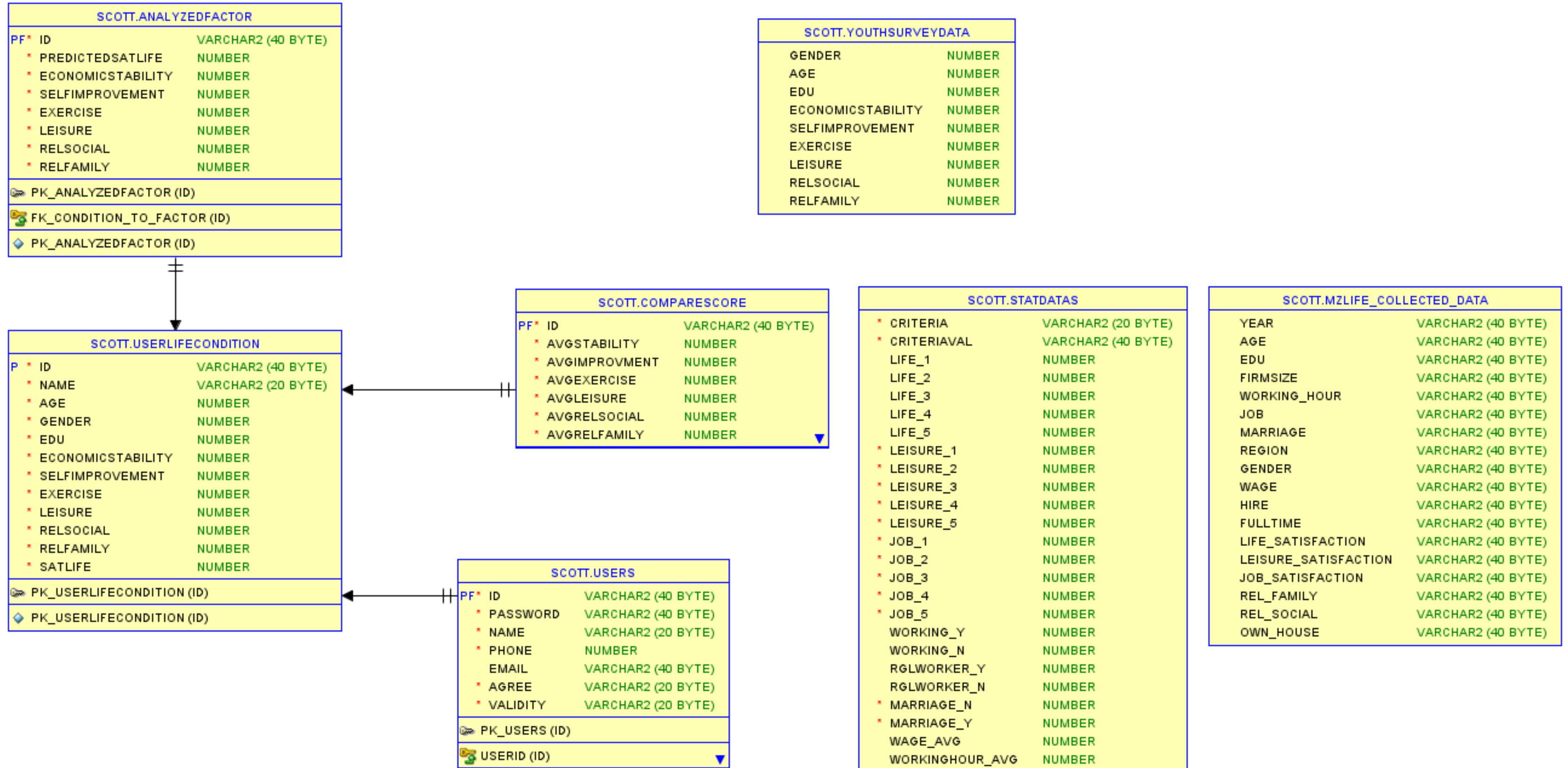
7. UML

Class Diagram



8. ERD

ER Diagram



9. 주요 서비스 기능

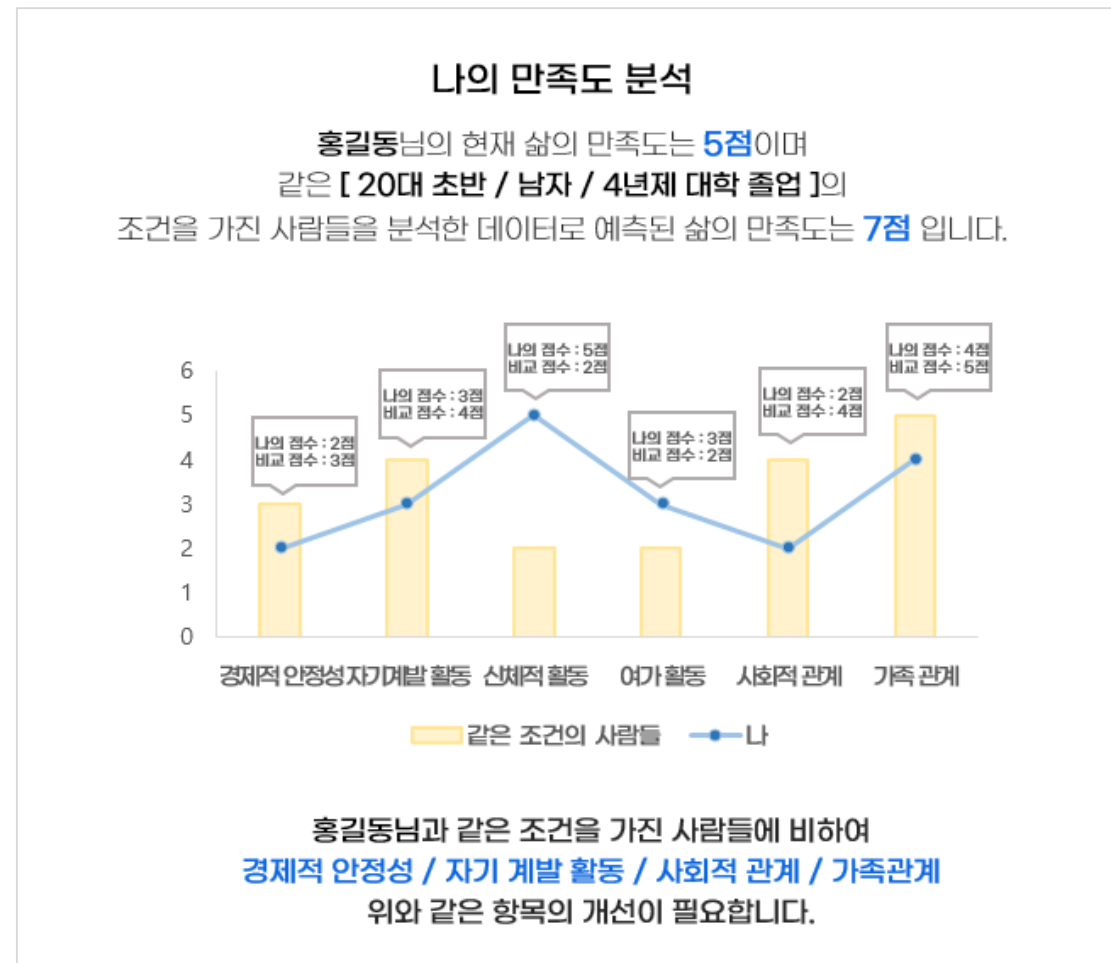
1) 사용자 만족도 분석 및 증가 방법 추천

[통계청]에서 제공하는 청년삶실태조사의
각 청년의 개인별 조건과 삶의 만족도가 포함되어 있는 데이터를 분석하여

사용자가 입력한 조건(연령, 성별, 학력)이 같은 사람들을 추출하여
회귀 분석을 통해 사용자의 항목별 점수를 대입해 예측한 삶의 만족도와
만족도에 영향을 미치는 항목의 평균 점수를 비교

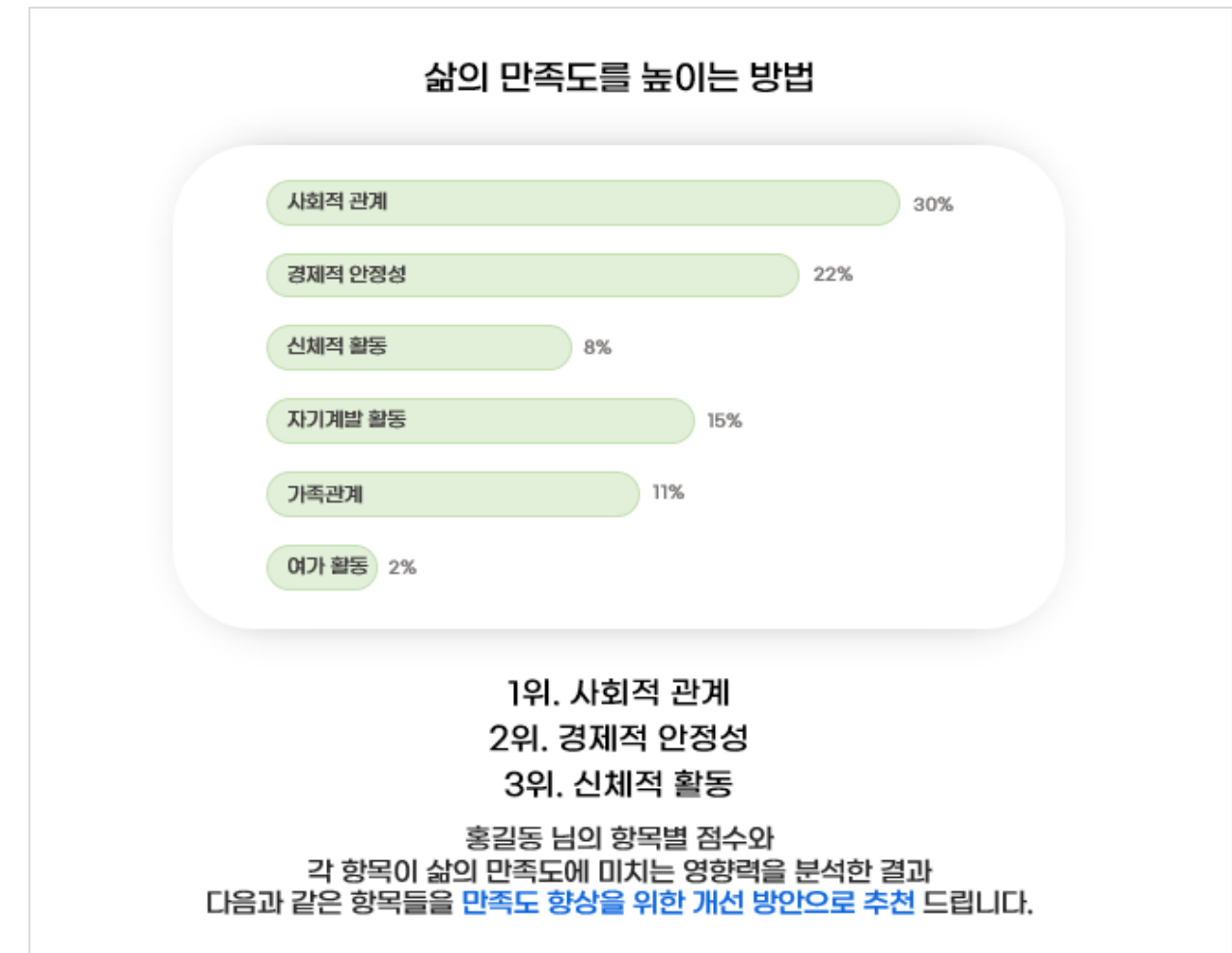


비교 대상에 비해 점수가 낮은 항목을
개선이 필요한 항목으로 표시



사용자의 조건과 항목별 점수를 분석하여
각 항목이 만족도 증가에 영향을 미치는 정도를 표시

가장 많은 영향을 미치는 항목 상위 3개를
삶의 만족도 증대를 위한 개선 방법으로 추천



9. 주요 서비스 기능

2) 선택 기준 별 통계 분석

성별, 연령, 학력, 거주지역, 취업여부, 삶의 만족도
총 6개의 항목 중
사용자가 비교해 보고자 하는 기준을 선택하면

만족도(삶, 여가생활, 직무)
취업률
고용형태
월 평균 임금
주 평균 근무시간
혼인율

8가지의 주제를 통계 분석한 결과를
다양한 그래프로 시각화 하여 나타내고,
가장 많은 비중을 차지하는 항목을 수치로 나타냄

비교분석 하고자 하는 항목을 선택해 주세요.

성별

연령

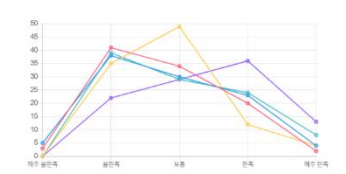
학력

거주지역

취업여부

삶의 만족도

삶의 만족도



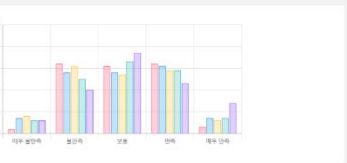
고등학교 졸업 미만: 불만족 41%
고등학교 졸업: 불만족 38%
대학교 재학/종퇴: 보통 49%
전문대학교 졸업: 불만족 39%
4년제 대학교 졸업 이상: 만족 36%

여가생활 만족도



고등학교 졸업 미만: 보통 42%
고등학교 졸업: 보통 36%
대학교 재학/종퇴: 보통 35%
전문대학교 졸업: 보통 31%
4년제 대학교 졸업 이상: 보통 31%

직무 만족도



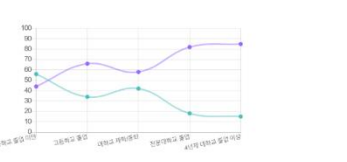
고등학교 졸업 미만: 불만족 32%
고등학교 졸업: 만족 31%
대학교 재학/종퇴: 불만족 31%
전문대학교 졸업: 보통 33%
4년제 대학교 졸업 이상: 보통 37%

취업률



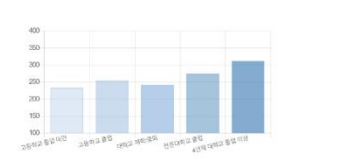
고등학교 졸업 미만: 미취업 68%
고등학교 졸업: 미취업 51%
대학교 재학/종퇴: 미취업 53%
전문대학교 졸업: 취업 59%
4년제 대학교 졸업 이상: 취업 59%

고용형태



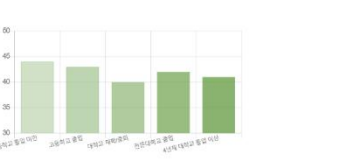
고등학교 졸업 미만: 비정규직 56%
고등학교 졸업: 정규직 66%
대학교 재학/종퇴: 정규직 58%
전문대학교 졸업: 정규직 82%
4년제 대학교 졸업 이상: 정규직 85%

월 평균 임금



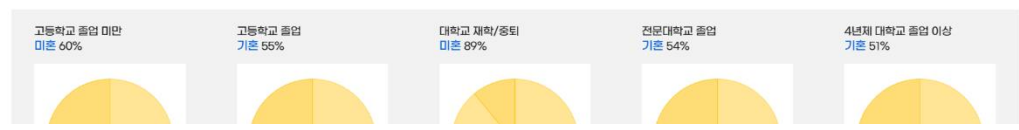
고등학교 졸업 미만: 평균 233원
고등학교 졸업: 평균 255원
대학교 재학/종퇴: 평균 242원
전문대학교 졸업: 평균 275원
4년제 대학교 졸업 이상: 평균 312원

주 평균 근무시간



고등학교 졸업 미만: 평균 44시간
고등학교 졸업: 평균 43시간
대학교 재학/종퇴: 평균 40시간
전문대학교 졸업: 평균 42시간
4년제 대학교 졸업 이상: 평균 41시간

혼인율



9. 소스코드

[Spring] Spring MVC 프로젝트

Controller

- └ com.mzlife.app.sts.controller
 - └ LifeSatController.java : 주요 서비스 기능 요청 처리
- └ com.mzlife.app.member.controller
 - └ MemberController.java : 회원 관리 기능 요청 처리

View

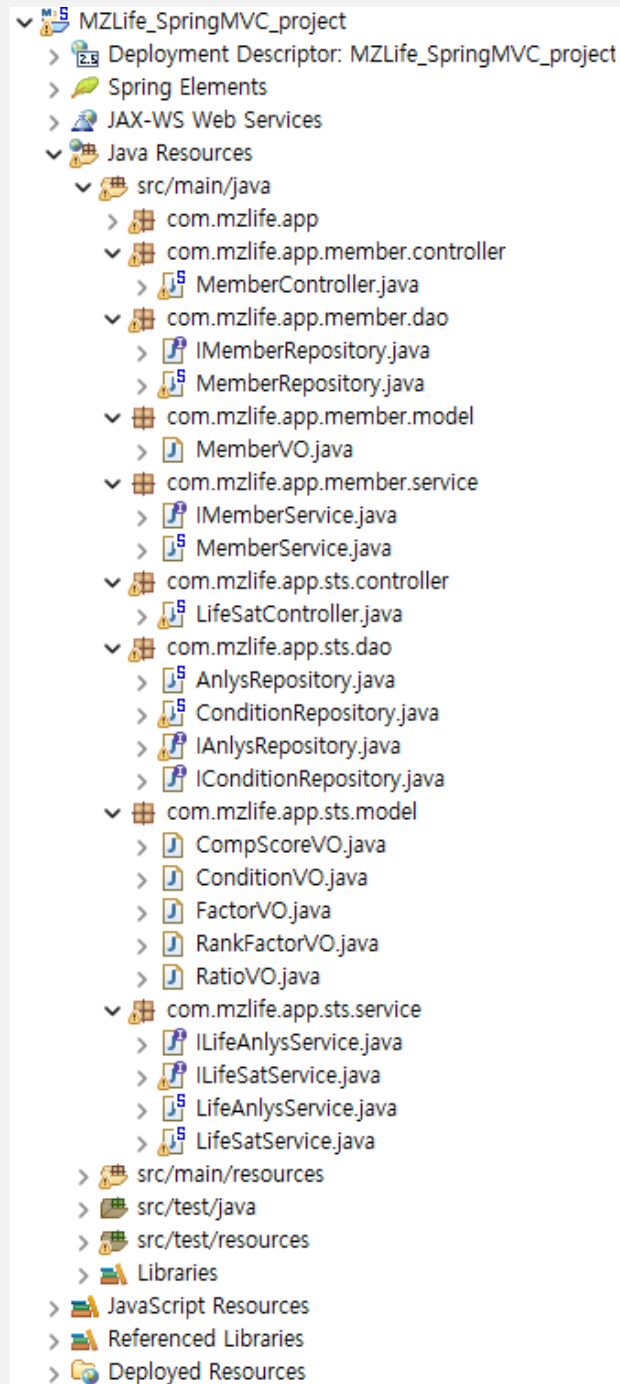
views

- └ anlys (주요 서비스)
 - └ mz_anlysis.jsp : 기준 별 통계 분석 기능 화면
 - └ mz_satisfaction_popupinfo.jsp : 개인 만족도 분석 기능 정보 입력 팝업
 - └ mz_satisfaction.jsp : 개인 만족도 분석 기능 화면
 - └ result_graphs.jsp : 기준 별 통계 분석 기능 화면 - 결과 그래프 영역
- └ member (회원 관리)
 - └ login.jsp : 로그인 화면
 - └ join.jsp : 회원가입 화면
 - └ find_id.jsp : 아이디 찾기 화면
 - └ find_pw_change.jsp : 비밀번호 변경 화면
 - └ edit_info.jsp : 회원 정보 수정 화면
 - ...
- └ main.jsp
- └ header.jsp

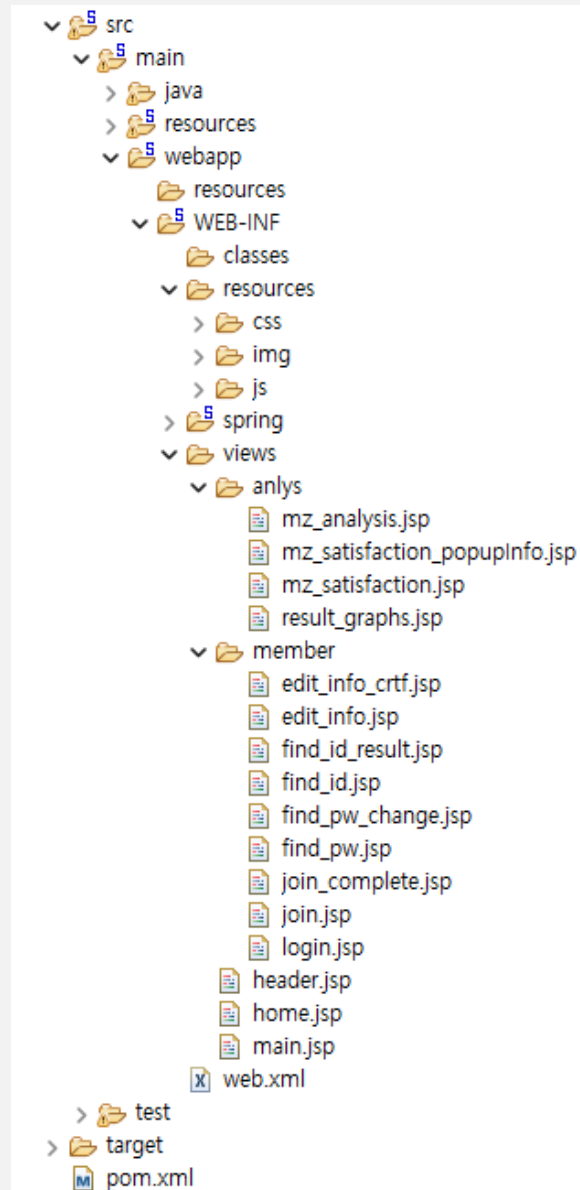
Model

src/main/java

- └ com.mzlife.app.sts.dao
 - └ IAnlysRepository.java (interface)
 - └ AnlysRepository.java
 - └ IConditionRepository.java (interface)
 - └ ConditionRepository.java
- └ com.mzlife.app.sts.model
 - └ CompScoreVO.java
 - └ ConditionVO.java
 - └ FactorVO.java
 - └ RatioVO.java
 - └ RankFactorVO.java
- └ com.mzlife.app.sts.service
 - └ ILifeAnlysService.java (interface)
 - └ LifeAnlysService.java
 - └ ILifeSatService.java (interface)
 - └ LifeSatService.java
- └ com.mzlife.app.member.dao
 - └ IMemberRepository.java (interface)
 - └ MemberRepository.java
- └ com.mzlife.app.member.model
 - └ MemberVO.java
- └ com.mzlife.app.member.service
 - └ IMemberService.java (interface)
 - └ MemberService.java



[Controller, Model]



[View]

[Controller]

@Controller

@RequestMapping(value="/anlys") // 주요 서비스 관련 요청은 모두 LifeSatController로 들어오도록 함
public class LifeSatController extends HttpServlet {

@Autowired

ILifeSatService satService; // 개인 만족도 분석

@Autowired

ILifeAnlysService anlysService; // mz 통계 분석

@RequestMapping(value="/satLife")

public String checkHistory(ConditionVO cond, RedirectAttributes redirectAttr, HttpSession session, HttpServletRequest request) {

session = request.getSession(); // session으로 로그인 여부 판단

String id = (String)session.getAttribute("id");

// 회원인지 확인, 회원이면 전에 입력한 정보가 있는지 확인

if(id!=null) {

ConditionVO condition = satService.getUserCond(id);

if(condition == null) { return "anlys/mz_satisfaction"; } // 로그인 했지만 저장된 입력 정보가 없음

if(condition != null) { request.setAttribute("conditions", condition); return "redirect:/anlys/conversion"; } // 로그인 했고, 저장된 정보가 있음

}

return "anlys/mz_satisfaction"; // 비로그인 상태

}

...

@RequestMapping(value="/factor")

public String getFactor(HttpServletRequest request, Model model) {

...

CompScoreVO compare = satService.getCompareScore(user.getId()); // 비교군의 평균점수

FactorVO factor = satService.getFactor(user.getId()); // 각 요인별 만족도에 영향을 주는 정도

RankFactorVO ranking = satService.sortFactor(factor); // 영향도 상위 3개 선별

model.addAttribute("user", user);

model.addAttribute("avg", compare);

model.addAttribute("factor", factor);

model.addAttribute("ranking", ranking);

return "anlys/mz_satisfaction";

}

@RequestMapping(value="/compare")

public @ResponseBody List<RatioVO> getRatio(@RequestParam(value="comparison") String ct) {

List<RatioVO> ratio = anlysService.getRatio(ct);

return ratio;

}

}

@RequestMapping

: 요청 url을 mapping 하여 필요한 서비스 매칭

: Class level에 요청 mapping을 설정

@Autowired

: 요청 처리를 위해 필요한 service → annotation을 이용한 DI

command 객체를 이용해 Parameter 처리

HttpSession을 이용하여 서비스 실행 전 로그인 상태 판단

view의 이름을 지정하여 데이터 전송

@ResponseBody

: 객체를 Json 형태로 변환

@RequestParam

: HTTP 요청으로 전달된 파라미터의 값을 받음

[Model] Repository

```
@Autowired
JdbcTemplate jdbcTemplate;

private class CondMapper implements RowMapper<ConditionVO> {
    @Override
    public ConditionVO mapRow(ResultSet rs, int count) throws SQLException {
        ConditionVO cond = new ConditionVO();
        cond.setId(rs.getString("id"));
        cond.setName(rs.getString("name"));
        ...
        return cond;
    }
}
```

Jdbc Template 사용하여 DB 접근

JdbcTemplate의
RowMapper 인터페이스를 통해
VO 객체로 반환

```
@Override
public ConditionVO getUserCond(String userId) {
    String sql = "select id, name, gender, birth, edu, "
        + "economicStability,selfImprovement, "
        + "exercise, leisure, relfamily, relsocial, satLife"
        + " from userlifecondition where id=?";

    try {
        return jdbcTemplate.queryForObject(sql, new CondMapper(),userId);
    } catch (IncorrectResultSizeDataAccessException error) {
        return null;
    }
}
```

쿼리문을 입력하여
DB에서 원하는 데이터 끌고 오기

JDBC queryForObject의 실행 결과가 없거나
두개 이상의 행이 있을 경우를 대비해서 예외 처리

[View]

```
$("#buttonSelect").click(function(){
$.ajax({
    type:'GET',
    url: 'compare.do?comparison='+$('input:radio[name=comparison]:checked').val(),
    dataType: "json",
    success:function(response){
        for (var i = 0; i < response.length; i++) {
            var graphJobId = "satJob"+(i+1);
            var graphJobBox = document.getElementById(graphJobId).getContext('2d');
            var ChartJob = new Chart(graphJobBox, {
                type: 'bar',
                data: {
                    labels: labelSat,
                    datasets: satJobDataset
                },
                options: graphOption(0,50,10,'%')
            });
        }
    }
});
... }}
```

Ajax로 요청을 보내 받은 데이터를 chart js를 이용하여 시각화 표현

[Model] VO

```
public class ConditionVO {
    private String id;
    private String name;
    private String gender;

    public String getId() {
        return id;
    }
    public void setId(String id) {
        this.id = id;
    }
    ...
}
```

DB에서 받아온 데이터를 담기 위한 객체

[Model] Service

```
@Service
public class LifeSatService implements ILifeSatService {

    @Autowired
    IConditionRepository condRepository;

    @Override
    public String insertCondition(ConditionVO cond) {
        String userId = condRepository.insertCondition(cond);
        return userId;
    }
    @Override
    public void predictLifeSat(String userId) {
        condRepository.predictLifeSat(userId);
    }
    ...
}
```

필요한 데이터를
Repository에 접근해 가져오거나
controller에서 처리할 요청에 따라
수행해야 하는 서비스 로직

```
<table class="table-input">
    <tr>
        <td>연령</td>
        <td class="age">${user.birth}</td>
    </tr>
    <tr>
        <td>성별</td>
        <td class="gender">${user.gender}</td>
    </tr>
    <tr>
        <td>학력</td>
        <td class="edu">${user.edu}</td>
    </tr>
</table>
```

JSTL과 EL 표현식으로 서버에서 받은 데이터 출력

[View]

9. 소스코드

[R] 데이터 처리 및 분석

- 활용 데이터

한국노동연구원에서 진행한 한국노동패널조사 데이터(약 13,000개) + 통계청에서 제공하는 청년삶실태조사 데이터(약 15,000개)를 처리

- 빅데이터 전처리 : 결측 값(Na) 제거

```
# csv 파일 데이터 프레임에 넣기
labor_data <- read.csv("C:/project_MyZeneration/data_laborPanel.csv", na.strings=c(""), encoding = "UTF-8", fileEncoding = "UTF-8")

# complete.cases() : 결측 값이 존재하지 않는 행만 treat_data에 저장
treat.data <- labor_data[complete.cases(labor_data), ]
```

- 다중선형회귀 분석

```
# 사용자와 같은 연령, 성별, 학력을 가진 행만 추출
subset.df <- treat.data[treat.data $age == user_age & treat.data $gender == user_gender & treat.data $edu == user_edu ,]

# lm() : 종속변수(삶의 만족도)와 독립변수(경제적 안정성, 자기계발 활동, 신체적 활동, 여가 활동, 사회적 관계, 가족관계) 사이의 선형 회귀 분석 진행
life <- lm(satLife ~ economicStability + selfImprovement + exercise + leisure + relSocial + relFamily , subset.df)

# predict() : 회귀 모델에 사용자의 데이터를 대입해 예측 삶의 만족도 산출
predicted.life <- predict(life, newdata=user.data)
```

9. 소스코드

[Java – R] Java 프로그램에서 R 프로그램 연동

```
public void predictLifeSat(String userId) {
    // R 연동해서 회귀모델에 사용자 조건을 대입해 조건에 예상되는 만족도 산출

    RConnection rConn = null;

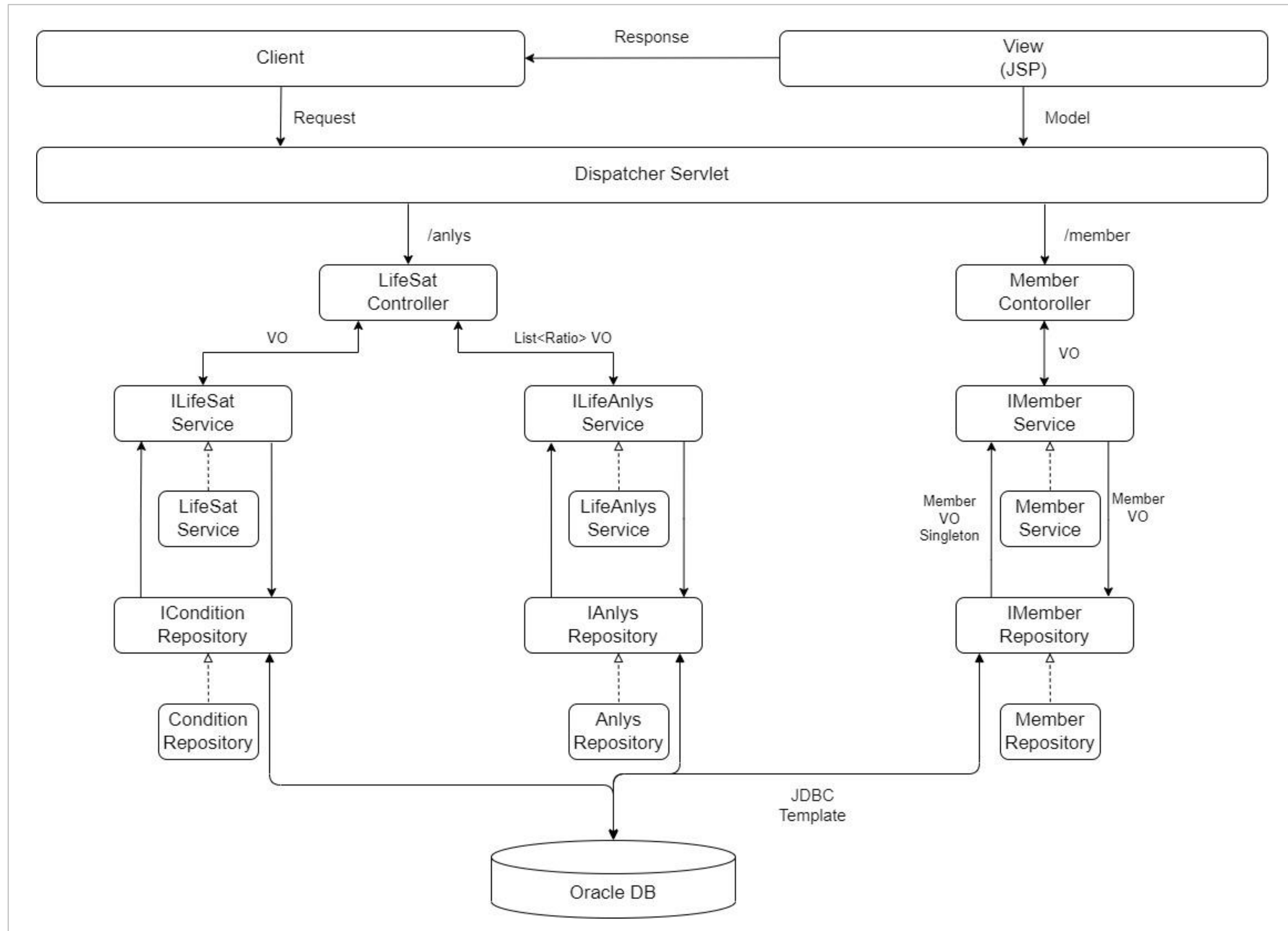
    try {
        // R 연동을 위해 Rserve 라이브러리 실행
        Process process = new ProcessBuilder("C:/Program Files/R/R-3.6.3/library/Rserve/libs/x64/Rserve.exe").start();

        // R과 연결
        rConn = new RConnection(); // R Connection
        rConn.eval("source('C:/workspace/mzlife_satisfaction.R')"); // R 소스 파일
        rConn.voidEval("library(Rserve)");
        rConn.voidEval("Rserve()");
        // RJDBC 연결
        rConn.voidEval("library(RJDBC)");
        rConn.voidEval("jdbcDriver <- JDBC(driverClass = \"oracle.jdbc.driver.OracleDriver\", classPath = \"C:/oraclexe/app/oracle/product/11.2.0/server/jdbc/lib/ojdbc6.jar\")");
        rConn.voidEval("con <- dbConnect(jdbcDriver, \"jdbc:oracle:thin:@localhost:1521:xe\", \"scott\", \"tiger\")");

        // R에 사용자 id 전달해서 삶의 만족도 예측하는 함수로 예측값 반환하여 DB에 업데이트
        rConn.voidEval("calcLifeSat('\" + userId + \"')");
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (rConn != null) {
            rConn.close();
        }
    }
}
```

10. Software Architecture

Software Architecture



11. 시연

프로젝트 시연

향후 계획 및 프로젝트 소감

- 향후 계획

JWT 토큰 기반 로그인 방식 구현

기존의 로그인 방식은 session 기반 로그인 방식으로 메모리 내부에 저장되기 때문에 저장 공간의 용량을 차지하게 된다. 이를 보완하기 위해 메모리 저장공간 확보가 필요 없는 JWT를 사용하고자 한다.

또한 서버를 확장하게 될 경우 세션은 따로 분산시키는 시스템을 설계해야 하지만 토큰 기반 인증은 그러한 과정이 필요하지 않기 때문에 확장성을 위하여 변경할 계획이 있다.

Spring Security를 이용한 인증/인가 구현

다양한 보안 기능을 제공하고 있는 Spring Security를 이용한 인증과 인가 처리를 통해 어플리케이션의 보안을 강화하고자 적용 계획이 있다.

- 프로젝트 수행 소감

프로젝트 기획의 중요성

막연한 **기획 아이디어와 실질적인 개발의 설계를**
접목 시키는 과정은 쉽지 않았습니다.

이를 해결하기 위해
기획서, 요구사항 명세서, 화면 설계서 등 **문서**
작성을 통해 개발을 위한 기획 단계를
구체화시켜갔습니다.

초기에 기획한 대로 개발을 진행하는 것 또한
어렵다는 것을 깨닫게 되었고, 이를 통해 **탄탄한**
기획의 중요성을 알게 되었습니다.

안정적인 시스템의 중요성

프로그램을 개발하면서 가장 크게 깨달았던 것은
개발자가 예상하지 못한 사용자의 흐름은 항상
존재한다는 것이었다.

스스로 프로그램을 테스트 하면서 완성되었다고
생각하였는데 예상치 못한 경우에서 에러가
발생하였고, 이를 해결하기 위해 정말 많은
테스트를 거치며 하나하나 해결해 나갔다.

이를 통해 **프로그램 개발만큼 중요한 것이**
체계적인 테스트라는 것을 깨닫게 되었고,
사용자의 흐름이 나의 예상을 벗어날 수 있다는
점을 항상 염두에 두고 있어야 한다는 것을 배울
수 있었습니다.

빅데이터 수집 및 처리의 어려움

빅데이터 분석 프로젝트를 진행하면서
가장 어려웠던 것은 원하는 데이터를 수집하는 것,
그 **데이터 분석의 결과가 생각한 대로 나오지**
않는 것이었습니다.

충분한 시간을 들여 데이터를 확보하는 데에
노력하였고, 결측값과 특이값을 처리하며
유의미한 분석 결과를 얻기 위해 노력하였습니다.

많은 양의 데이터 분석을 진행하면서 그 분석
결과에서 특정한 추세를 발견하게 되거나
예측하는 과정에서 빅데이터의 경이로움과
새로운 기술을 익히는 즐거움을 깨달았습니다.

감사합니다