**Education**

|  |  |  |  |
| --- | --- | --- | --- |
| **기간** | **과정명** | **시수(H)** | **교육기관** |
| 2019.08-2020.02 | 자바기반 웹 개발자 양성 | 720h | kg아이티뱅크 |
|  |  |  |  |

**Education**

|  |  |
| --- | --- |
| **과정명** | **교육 내용** |
| 자바기반 웹 개발자 양성 | HTTP 프로토콜을 커뮤니케이션 매체로 이용한 데이터의 비즈니스 로직을 전반적으로 익히며, 스프링 프레임워크를 이용한 웹서비스 구현하였음. 또한 MyBatis를 이용한 웹과 DB의 연동방법 터득하였고 웹 개발 및 유지 보수에 필요한 기술 등을 터득하였음. |

**Technology**

|  |  |
| --- | --- |
| **기술분류** | **보유기술** |
| **Language** | **[JAVA]**  - 자바 문법 습득  - JDBC 개념 이해 및 데이터베이스와 연동  - 이벤트의 개념과 이벤트 처리 방식의 이해 및 활용 가능  - OOP(객체-지향) 프로그래밍의 개념 습득(추상화, 캡슐화, 상속, 다형성)  - 객체지향을 이용한 프로그래밍 기법 가능  - 기초적인 알고리즘 코드 작성 가능(ex>선택정렬, 삽입정렬 등등..)  **[Python](JAVA와의 차이점을 알고 싶어 공부했기에 초심자 수준)**  - 기본적인 문법과 JAVA와의 차이점 습득  - 간단한 계산기 구현 가능  **[JSP]**  - Java Bean을 활용, Set/Get 메소드를 이용해 DB와 연동하여 DB 입출력  - Model1 / Model2 기반 개발  - 상용화된 여러가지 기능(회원가입, 로그인, id기억, 로그인 유지, 파일 업로드, 게시글, 댓글 등등..)구현 가능 |
| **DBMS** | **[Oracle 11g]**  - DB 설치 및 기본 SQL 작성  - DB 모델링 기법 및 정규화 |
| **WAS** | **[Tomcat 8.5]**  - 서버 구축 작업 및 사용 가능 |
| **TOOL** | **[IDE / DB Tool]**  - Eclipse를 이용한 자바 코드 작성 |
| **API** | - 카카오 open API를 이용하여 웹 페이지에 지도 구현  - script안의 open API에서 제공하는 기능이 속한 메서드 수정과 간단한 메서드 생성을 통하여 위치 좌표, 주소, 가게 이름 등을 원하는 방법으로 이용 |
| **Framework** | **[myBATIS]**  - MVC 패턴 및 myBATIS 연동 가능  - XML을 통해 SQL statement에 매핑 가능  **[Spring]**  - 어노테이션 기반 Controller 구현 가능  - Spring MVC패턴 이용 WebApp 제작 |

**Project (프로젝트)**

**Project Info**

|  |  |
| --- | --- |
| **프로젝트 명** | 위치기반 맛집 웹 서비스 |
| **개발 기간** | 2020.01-2020.02(3주) |
| **참여인원** | 6명 |
| **담당업무** | 조원 데이터베이스 설계, 커뮤니티 기능 구현 |

**Use Technology**

|  |  |
| --- | --- |
| **개발 환경** | JAVA, Eclipse |
| **사용 도구** | STS, Oracle11g, Apache tomcat 8.5, MyBatis |
| **사용 기술** | Spring MVC Model2, open API(카카오 developer), ajax |

**Introduction**

|  |
| --- |
|  |

**개발 전 계획**

* 메인 페이지를 제외한 모든 서비스에 접근 시, 로그인(회원가입)을 강제적으로 요구하게 한다.

(세션에 저장된 아이디값의 null여부를 체크하는 인터셉터를 이용)

* 지도 API를 이용, 지정한 위치를 중심으로 하여 나타나는 맛집 리스트 및 상세정보(지도상의 맛집 클릭시 해당 가게의 이름을 가진 커뮤니티 게시글과 베스트 맛집의 정보를 보여줌)는 메인페이지에서 나타나게 한다(상세정보는 시간부족으로 인하여 구현하지 못함).
* 회원가입 시 일반회원, 사업자회원을 구분하여 가입. 사업자회원은 회원정보 수정을 통하여 자신의 맛집을 등록할 수 있도록 한다.

일반 회원은 아이디를 제외한 정보만 수정가능하게 만든다(아이디 수정은 곧 회원가입과 같다고 판단).

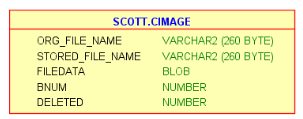
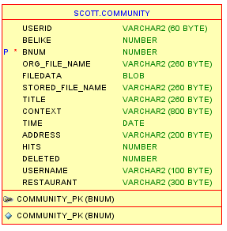
* 일반 회원은 커뮤니티 서비스를 이용할 수 있으며, 게시글 작성시 제목, 사진, 본문을 작성하고 가게이름을 검색하여 지도API에서 위치와 주소를 따온다(시간 부족으로 사진 업로드 개수는 한 장으로 제한, 관리자 또한 게시글 작성이 가능).
* 베스트 맛집에서 맛집을 클릭 시, 예약(당일 제외)과 번호표 발급(당일)이 가능. 번호표의 경우 해당 가게의 사업자 회원이 카운트를 지울 수 있도록 구현(시간부족으로 인한 미구현).
* 관리자 계정(admin)은 모든 정보(등록된 맛집, 게시글, 예약, 회원리스트)에 접근 가능. 수정 및 삭제가 가능
* 모든 기능과 DB table 생성 시 자신이 맡은 기능을 정확히 명시하여 변수와 클래스 명, 테이블 명이 중복되지 않도록 한다(소통의 오류로 인해 로그인 구현자와 커뮤니티 구현자만 명시).

**Develop Detail**

|  |
| --- |
|  |

**DB설계(본인이 맡은 기능과 그에 연관된 테이블에 대한 설명만 있음)**

* **커뮤니티 글 작성, 수정, 삭제 시 이용할 테이블**

****

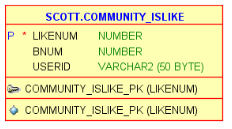
- bnum(Board Number)을 주 키(primary key)로 지정하여 게시글의 중복을 피함(시퀀스를 이용하여 데이터 1열 추가 시 bnum이 1씩 증가).

- 게시글 삭제 후 다시 작성 시 bnum의 중복이 발생하는 것을 방지하기 위하여 삭제 시 deleted컬럼의 값이 1로 바뀌게 된다( + 게시글 삭제 시 데이터베이스에서 사라지는 것은 보안의 문제도 있다고 생각).

- 사진은 불러올 시 BLOB형식으로 불러온다.

- CIMAGE 테이블은 검색 기능을 구현할 조원에게 제공할 정보를 보내주기 위해 생성함. 게시글 작성 시 해당 DB에 겹치는 정보를 동시에 적재한다(소통의 오류로 사용되지 않음 – 현재 커뮤니티 게시글은 베스트 맛집과 메인페이지에서 검색해도 나오지 않음).

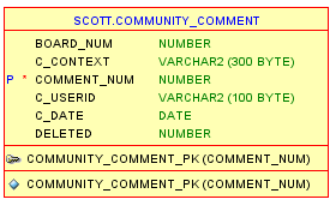
* **좋아요( + 취소) 기능 테이블**



* 들어간 게시글의 번호를 bnum에, 로그인 된(세션에 저장된)아이디를 userid에 넣어 좋아요를 클릭시 데이터가 생성됨.
* 현재 LIKENUM컬럼은 무의미함.

(마무리 단계에서 효율성을 위해 좋아요 와 좋아요 취소(한번 더 클릭)를 deleted라는 컬럼에 0과 1로 구분하고 LIKENUM을 주 키로 두어 중복을 피하는 쪽으로 수정하려 했으나, 시간 부족으로 LIKENUM컬럼은 만들었지만 사용하지 않고 좋아요 시 데이터 생성, 취소 시 삭제하게끔 구현)

* **댓글 작성, 수정, 삭제 기능 테이블**



* 댓글 작성 시 게시글 작성과 비슷하되, 하나의 게시글에 접근 시 해당 게시글의 번호(bnum)을 네비게이터로 삼아 리스트를 불러오게끔 설계