HOT Delivery

JSL 인재개발원 26기

우마우마

2017. 12. 15 ~ 2018. 01. 16

**< 목 차 >**

**1. 기획안**

1.1 프로젝트 개요

1.2 구축 환경

1.3 사용 기술

1.4 팀원 구성 및 역할

1.5 개발 일정

**2. 요구사항 명세서**

2.1 관리자 페이지

2.2 사용자 페이지

**3. 프로그램 설계**

3.1 ER Sequence Diagram

3.2 Site Map

3.3 Table List

3.4 Table Structure

3.5 SQL Query

**4. UI 정의서**

4.1 메인 페이지

4.2 회원 페이지

4.3 점주 페이지

**5. 보완점 및 소감**

5.1 보완점

5.2 소감

**6. 참고자료**

6.1 벤치마킹

6.2 API

**1. 기획안**

**1.1 프로젝트 개요**

|  |  |
| --- | --- |
| **팀 명** | 우마우마 |
| **팀 원** | 정용정, 윤상호, 임채훈, 한은찬 |
| **프로젝트 이름** | HOT Delivery |
| **이용 고객** | 1. 배달 서비스를 희망하는 고객 2. 가게를 등록하고 싶은 고객 |
| **개발 기간** | 약 4주 (2017. 12. 15 ~ 2018. 01. 16) |
| **개발 이유** | 1. 재미있게 최대의 만족감을 얻을 수 있는 음식 주문을 위해  2. 실용적인 웹 개발  3. 어플리케이션 개발 가능 |

**1.2 구축환경**

|  |  |
| --- | --- |
| **OS** | Windows 7 64bit |
| **IDE** | Eclipes neon.3, Atom 1.18 |
| **WAS, WEB Server** | Apache Tomcat 7 |
| **Database** | Oracle 11g |

**1.3 사용기술**

|  |  |
| --- | --- |
| **프로그래밍 언어** | JAVA JDK 1.8.0  JSP + Servlet  JavaScript + jQuery |
| **Design Tool** | pixlr |

**1.4 팀원 구성 및 역할**

|  |  |
| --- | --- |
| **정용정** | 이상형월드컵, 관리 기능 및 로그인, 기타 DB 담당 |
| **윤상호** | 네이버 지도 오픈 api 및 결제 기능 담당 |
| **임채훈** | UI 개발 및 dto설계, 그래픽, 개발서, 발표자료 준비 |
| **한은찬** | 주문목록, 이상형월드컵 담당 |

**1.5 개발 일정**

|  |  |
| --- | --- |
| **1주차**  **(12.18` ~ 12.25)** | 아이디어 및 기안서 작성, DB설계 및 기능 구상 |
| **2주차**  **(12.26 ~ 01.01)** | 일반 페이지 UI 및 기능 개발 (공지사항)  사용자 페이지 UI 및 기능 개발 (회원가입, 로그인, 마이페이지)  사용자 페이지 핵심 기능 개발 (모임등록, 신청 및 관리) |
| **3주차**  **(01.02 ~ 01.08)** | 주문 페이지 기능 개발  점주 페이지 UI 및 기능 개발  프로젝트 디버깅 |
| **4주차**  **(01.09 ~ 01.17)** | 페이지 일본어 번역  최적화 및 관리, 보수  발표자료 제작 및 준비  모의 발표  최종 발표 |

**2. 요구사항 명세서**

**2.1 메인 페이지**

|  |  |  |
| --- | --- | --- |
| **기능** | **요구사항** | **상세 요구사항** |
| 로그인 | 로그인 체크 | 로그인 하려는 아이디를 찾고 비밀번호를 확인하여 로그인. |
| ID 찾기 | 회원가입시 등록한 통해 ID 를 찾는다. |
| 비밀번호 찾기 | 회원가입시 등록한 정보를 통하여 비밀번호를 재설정 할 수 있게 한다. |
| 회원가입 | 신규 등록 | 신규 회원의 경우 ID, 비밀번호, 이름, 연락처, 이메일을 입력 받도록 한다. |
| 아이디 중복체크 | 기존 아이디와 신규 등록하려는 아이디의 중복을 여부를 검사한다. |
| 페이스북 | 페이스북 연결 | 페이스북에 연결하여 이벤트를 확인할 수 있도록 한다. |
| 검색 | 음식이름 | 먹고 싶은 음식을 갖고있는 가게들을 출력한다. |
| 홈 | 홈 | 메인 페이지로 돌아갈 수 있도록 한다. |

**2.2 회원 페이지**

|  |  |  |
| --- | --- | --- |
| **기능** | **요구사항** | **상세 요구사항** |
| 음식월드컵 | 음식이름 | 먹고 싶은 음식을 선택하여 마지막에 선택된 음식을 팔고있는 가게들을 출력한다. |
| 마이페이지 | 개인정보 변경 | 회원의 개인정보를 수정할 수 있도록 한다. |
| 회원 탈퇴 | 회원탈퇴를 할 수 있도록 한다. |
| 배달지 설정 | 직접입력 | 어디서나 음식을 배달시킬 수 있으므로 주소를 입력 받아 배달지를 유동적으로 설정.  단 주소는 지번으로 해야 정확한 위치를 받아 마커를 찍을 수 있다. |
| 사용자주소 | 회원가입시 사용자의 주소로 배달지를 설정. |
| 지도 | 현재 위치와 가게의 위치를 표시 하도록 한다. |
| 음식 선택 | 가게 정보 | 가게와 등록되어 있는 음식의 정보를 보여주고 수량을 설정 |
| 음식 정보 | 구매할 음식, 수량을 설정할 수 있도록 한다. |
| 결제 | 결제 | 구매할 음식, 수량을 출력한다. |
| 전화번호 인증 | 핸드폰을 통해 본인인증을 하여 결제를 할 수 있도록 한다. |
| 이메일 체크 | 가입시 등록해 놓은 이메일과 비교하여 결제를 할 수 있도록 한다. |

**2.3 점주 페이지**

|  |  |  |
| --- | --- | --- |
| **기능** | **요구사항** | **상세 요구사항** |
| 마이페이지 | 메뉴 등록 | 메뉴를 등록 할 수 있도록 한다. |
| 메뉴 수정 | 메뉴를 수정 할 수 있도록 한다. |
| 가게 삭제 | 가게를 삭제 할 수 있도록 한다. |

**3. 프로그램 설계**

**3.1 Table List**

|  |  |
| --- | --- |
| **테이블명** | **설명** |
| deliveryMember | 회원 정보를 저장하는 테이블 |
| Primary Key : id |
| worldcup | 음식 카테고리를 저장하는 테이블 |
| . |
| store | 가게 정보를 저장하는 테이블 |
| Primary Key : storeNo |
| food | 음식 정보를 저장하는 테이블 |
| Foreign Key : storeNo |

**3.2 Table Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **테이블명** | deliveryMember | | | |
| **테이블역할** | 회원 정보를 저장하는 테이블 | | | |
| **필드역할** | **필드명** | **DATATYPE** | **NULL 허용** | **비고** |
| ID | id | VARCHAR2(20) |  | PK |
| 비밀번호 | password | VARCHAR2(15) |  |  |
| 이름 | name | VARCHAR2(50) |  |  |
| 돈 | money | VARCHAR2(10) |  | Default 0 |
| 이메일 | email | VARCHAR2(50) |  |  |
| 주소 | address | VARCHAR2(100) |  |  |
| 가게주소 | storeAddress | VARCHAR2(100) | null |  |
| 사용목적 | userkind | VARCHAR2(10) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **테이블명** | food | | | |
| **테이블역할** | 전체음식정보 | | | |
| **필드역할** | **필드명** | **DATATYPE** | **NULL 허용** | **비고** |
| 가게번호 | storeNo | NUMBER(10) | NOT NULL | References store  (storeNo) |
| 음식종류 | foodKind | NUMBER(5) | NOT NULL |  |
| 음식이름 | foodName | VARCHAR2(50) | NOT NULL |  |
| 가격 | price | VARCHAR2(5) | NOT NULL |  |
| 이미지 | imgUrl | VARCHAR2(100) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **테이블명** | store | | | |
| **테이블역할** | 메시지를 저장하는 테이블 | | | |
| **필드역할** | 필드명 | DATATYPE | NULL 허용 | 비고 |
| Sequence | StoreNo | NUMBER(10) |  | PK |
| 가게이름 | storeName | VARCHAR2(50) |  |  |
| 가게전화번호 | storePhone | VARCHAR2(15) |  |  |
| 주소 | address | VARCHAR(100) |  |  |
| 경도 | longitude | NUMBER(20,10) |  |  |
| 위도 | latitude | NUMBER(20,10) |  |  |
| 이미지 주소 | imgUrl | VARCHAR2(100) |  |  |
| 음식코드 | foodcode | VARCHAR2(10) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **테이블명** | worldCup | | | |
| **테이블역할** | 음식 정보를 저장하는 테이블 | | | |
| **필드역할** | **필드명** | **DATATYPE** | **NULL 허용** | **비고** |
| 음식 종류 | foodKind | VARCHAR2(20) | NOT NULL | P |
| 음식 번호 | foodNo | NUMBER(16) | NOT NULL |  |
| 이미지 URL | imgUrl | VARCHAR2(100) | NOT NULL |  |
| 음식코드 | foodcode | VARCHAR2(10) |  |  |

**3.3 SQL Query**

|  |  |  |
| --- | --- | --- |
| **Package** | **Delivery.member.dao** | **회원관리** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| inserMember(MemberDTO dto) | insert into deliveryMember(id,password,name,email,address,userkind) values(?,?,?,?,?,?) | 회원등록 |
| updateMember(MemberDTO dto) | update deliveryMember set password=?, email=?,address=? where id=? | 회원수정 |
| deleteMember(String id,String pwd) | delete from deliveryMember where id=? and password=? | 회원탈퇴 |
| idCheck(String id) | select id from deliveryMember where id=? | 아이디 중복검색 |
| idcheck(String id,String pwd) | select id,password from deliveryMember where id=? and password=? | 로그인 정보 확인 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.woldcup.dao.MakeRandomFoodDAO** | **음식월드컵** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| selectKind() | select \* form worldCup; | 음식정보 리스트에 저장 |
| selectFood(int num) | Select \* from worldCup where foodNo=? | 음식번호에 따른 음식이름과 이미지 Url 선택 |
| setDTO(ArrayList<FoodDTO>) | Select \* from worldCup where foodNo=? | 리스트에 따른 음식정보 선택 |
| setUrl(int[] idx) | Select \* from worldCup where foodNo=? | 인덱스 값에 따른 음식 정보 선택 |
| setUrl\_2(ArrayList<Integer>) | Select \* from worldCup where foodNo=? | 리스트에 따른 음식 정보 선택 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.food.dao.FoodDAO** | **음식정보** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| CountFood(String search) | Select count(\*) from food where foodName Like search | 검색한 음식의 총 갯수 |
| searchFood(String search, int count) | Select \* from food where foodName LIKE search | 검색한 음식의 정보 |
| searchAllStore(ArrayList<integer> storeNoList) | Select \* from store order by storeNo asc | 가게번호의 정보를 오름차순으로 정렬 |
| storeNoList(int num) | Select storeNo from food where foodkind = num | 가게번호가 같은 것들의 중복제거 |
| foodInfo() | Select \* from food | 음식의 정보 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.store.dao.StoreListDAO** | **가게정보** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| selectCount(int num) | select count(\*) store storeNo =? | 가게번호의 총 갯수 |
| storeList(ArrayList<integer> storeNoList) | Select \* from store | 가게의 정보 |
| storeNoList(int num) | Select storeNo from food where foodkind=num | 음식의 종류의 따른 가게번호 + 중복제거 |
| findFood(int storeNo) | Select \* from food where storeNo=storeNo | 가게번호의 음식 정보 |
| selectStoreInfo(int storeNo) | Select \* from store where storeNo=storeNo | 가게번호의 따른 가게정보 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.cart.dao.CartDAO** | **주문리스트** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| int getMoney(String id) | select money from deliveryMember where id=? | 현재 계정이 가지고 있는 금액을 가져오는 메소드 |
| void insertCart(String id, ArrayList<FoodDTO> foodList,String address) | insert into cart(id,foodName,quantity,address,summoney) values(?,?,?,?,?) | 결제가 되었을 경우 주문리스트에 넣는 메소드 |
| void updateMoney(String id, int money) | update deliveryMember set money=? where id=? | 금액충전 메소드 및 결제한 뒤 계산한 금액을 수정하는 메소드 |
| ArrayList<CartDTO> showList(MemberDTO dto) | select \* from cart where id=? | 주문페이지에서 주문리스트를 보여주기 위한 메소드 |

**4. UI 정의서**

**4.1 메인 페이지**

|  |  |
| --- | --- |
| **메인 페이지** | |
| **기능** | 초기 화면 |
| **URL** | main/Main.jsp |
| **PC 버전** |  |
| **기능 설명** | 상단 메뉴를 통해 다른 페이지로 접근이 가능하며  로그인을 한 경우 마이페이지로 접근이 가능하다.  음식의 전체적인 장르와 상세한 음식이름을 통해 식당 찾기가 가능하다.  간략한 소개와 배너가 보인다. |

|  |  |
| --- | --- |
| **로그인** | |
| **기능** | 로그인 |
| **URL** | login/login.jsp |
| **PC 버전** |  |
| **기능 설명** | 로그인 버튼을 눌렀을 시 이동되는 창  아아디, 비밀번호 체크를 통해 로그인이 가능하고 ID 및 비밀번호 찾기 가능. |

|  |  |
| --- | --- |
| **로그인** | |
| **기능** | 아이디 찾기 |
| **URL** | login/login.jsp |
| **PC 버전** | C:\Users\JSLHRD\Desktop\이미지파일\아이디찾기.PNG |
| **기능 설명** | 아이디 찾기 버튼을 눌렀을 시 출력되는 팝업 창  이름과 이메일 체크를 통해 아이디 찾기가 가능. |

|  |  |
| --- | --- |
| **로그인** | |
| **기능** | 비밀번호 찾기 |
| **URL** | login/login.jsp |
| **PC 버전** | **C:\Users\JSLHRD\Desktop\이미지파일\비밀번호찾기.PNG** |
| **기능 설명** | 비밀번호 찾기 버튼을 눌렀을 시 출력되는 팝업 창  아이디와 이름과 이메일 체크를 통해 비밀번호 찾기가 가능. |

|  |  |
| --- | --- |
| **회원가입** | |
| **기능** | 회원가입 |
| **URL** | regist/UserCreate.jsp |
| **PC 버전** |  |
| **기능 설명** | 상단 메뉴의 회원가입 버튼을 클릭하면 회원가입 폼으로 이동  등록 버튼 클릭 시 가입이 되며 중복된 아이디가 있을 경우 아이디 부분만  다시작성 |

|  |  |
| --- | --- |
| **페이스북** | |
| **기능** | 페이스북 |
| **URL** | https://www.facebook.com/ |
| **PC 버전** |  |
| **기능 설명** | Facebook 버튼을 눌렀을 때 이동되는 창  이벤트 등 정보를 확인 할 수 있다. |

**4.2 회원 페이지**

|  |  |
| --- | --- |
| **음식 월드컵** | |
| **기능** | 음식 월드컵 |
| **URL** | worldcup/worldcup.jsp |
| **PC 버전** |  |
| **기능 설명** | 여러 음식 카테고리 중 이상형 월드컵처럼 음식을 선택함으로 주문 페이지로 이동 |

|  |  |
| --- | --- |
| **음식 월드컵** | |
| **기능** | 음식선택 페이지 |
| **URL** |  |
| **PC 버전** | C:\Users\JSLHRD\Desktop\KakaoTalk_20180115_015018742.png |
| **기능 설명** | 여러 음식 카테고리 중 이상형 월드컵처럼 음식을 선택함으로 가게로 이동 |

|  |  |
| --- | --- |
| **주문** | |
| **기능** | 음식 수량 선택 |
| **URL** | storelist/selectjsp |
| **PC 버전** | **C:\Users\JSLHRD\Desktop\이미지파일\음식수량.PNG** |

|  |  |
| --- | --- |
| **주문** | |
| **기능** | 배달지설정 |
| **URL** | storelist/select.jsp |
| **PC 버전** | **C:\Users\JSLHRD\Desktop\이미지파일\주소지.PNG** |
| **기능 설명** | 배달받을 주소를 설정하고 현위치와 가게위치를 지도에 표시 |

|  |  |
| --- | --- |
| **주문** | |
| **기능** | 결제 확인1 |
| **URL** | storelist/payment.jsp |
| **PC 버전** |  |
| **기능 설명** | 오른쪽의 삼각형 버튼을 누르면 주문할 음식과 총 결제 금액을 표시 |

|  |  |
| --- | --- |
| **주문** | |
| **기능** | 결제 확인2 |
| **URL** | storelist/payment.jsp |
| **PC 버전** |  |
| **기능 설명** | 이름과 이메일 총 결제 금액을 표시하고 전화번호를 통한 정보 확인 |

|  |  |
| --- | --- |
| **검색** | |
| **기능** | 검색한 음식의 가게 |
| **URL** | liststore/liststore.jsp |
| **PC 버전** |  |
| **기능 설명** | 메인 페이지의 검색 기능을 이용해 검색한 음식이름을 가지고 있는 가게들의 위치와 이름, 주소, 전화번호를 표시한다. |

|  |  |
| --- | --- |
| **마이 페이지** | |
| **기능** | 초기 화면 |
| **URL** | mypage/mypage.jsp |
| **PC 버전** |  |
| **기능 설명** | 회원 로그인에 성공했을 경우 접근할 수 있는 페이지 |

|  |  |
| --- | --- |
| **마이 페이지** | |
| **기능** | 회원정보 수정 |
| **URL** | mypage/mypage.jsp |
| **PC 버전** |  |
| 기능 설명 | 마이페이지의 회원정보 수정 버튼을 눌러 회원정보를 수정할 수 있다. |

|  |  |
| --- | --- |
| **회원 페이지** | |
| **기능** | 회원탈퇴 |
| **URL** | mypage/mypage.jsp |
| **PC 버전** |  |
| **기능 설명** | 마이 페이지의 회원 탈퇴 버튼을 눌러 회원탈퇴가 가능하다. |

|  |  |
| --- | --- |
| **회원페이지** | |
| **기능** | 금액충전 |
| **URL** | mypage/mypage.jsp |
| **PC 버전** |  |
| **機能説明** | 회원페이지의 금액 충천 버튼을 누르면 나타난다.  충전 버튼을 누르면 금액이 충천된다. |

|  |  |
| --- | --- |
| **주문 리스트** | |
| **기능** | 주문 리스트 |
| **URL** | cart/Cart.jsp |
| **PC 버전** |  |
| **기능 설명** | 메인페이지의 주문 리스트 버튼을 누르면 주문된 음식과 주소 등 주문 정보를 보여준다. |

**4.3 점주 페이지**

|  |  |
| --- | --- |
| **회원 페이지** | |
| **기능** | 가게 메뉴 등록 |
| **URL** | mypage/mypage.jsp |
| **PC 버전** |  |
| **기능 설명** | 점주 로그인에 성공했을 경우 메뉴 등록 및 수정을 위한 페이지 (구현예정) |

**5. 보완점 및 소감**

**5.1 보완점**

|  |  |
| --- | --- |
| **분류** | **상세 내용** |
| **UI** | 페이지 최적화  모바일 환경에서의 가독성 향상 |
| **기능** | 관리자 기능 구현  점주의 가게 등록 및 수정 기능 구현 |
| **성능** | 다양한 상황에서의 NullPointerException  반복되는 로직의 클래스/모듈화  유효성검사 정확도 향상 |
| **보안** | Command Pattern을 활용한 사용자 임의의 QueryString방지  다수 접속에 따른 DB Connection관리  SQL injection 예방 |

**5.2 소감**

|  |  |
| --- | --- |
| **정용정** | 이번 프로젝트를 하면서 많은 점을 느꼈습니다. 멤버가 서로 커뮤니케이션을 통해 각자 맡은 부분을 잘 수행한 것 같아서 좋았습니다. 리더의 역할이 중요하다는 걸 느끼게 되었고 팀원 간의 커뮤니케이션, 역할분담이 어렵다는 것을 실감 하였습니다. 결과적으로 멤버들에게 배운 것도 많이 있어서 만족적인 팀 프로젝트 였습니다. |
| **윤상호** | 처음 쓰는 오픈 API를 사용해봤는데 그걸 사용해 적용함으로써 재미를 느꼈고 각자 맡은 임무를 다들 잘해줘서 고맙게 생각하고 있습니다. 결과적으로 많이 배웠고 깨달은 것도 많은 경험이었습니다. 다음에 기회가 된다면 이 프로젝트를 더 확장해서 많은 기능을 추가한 홈페이지를 만들고 싶습니다. |
| **한은찬** | 처음 하는 웹프로젝트에서 지금까지 배웠던 것들을 최대한 활용해서 진행해 보았지만 아직 부족한 점이 많다는 것을 깨달았습니다. 하지만 팀원과의 의사소통이 잘되었고, 각자의 역할에서 충실하게 임했다고 생각되어 팀원 간의 유대도 깊어지고 더욱 더 발전된 저를 볼 수 있는 기회가 된 것 같아서 좋았습니다. |
| **임채훈** | 이번 팀 프로젝트에서 주 담당부분이 처음 접해보는 프론트 부분이라 시작 했을 때에는 많은 어려움이 있었지만 한 페이지 만들 때마다 디자인이 원하는 모양대로 변하는 것을 느낄 수 있어서 포기하지 않고 버틸 수 있었습니다. 힘들었지만 그 덕에 요즘 요구하는 풀 스택 엔지니어에 한걸음 다가간 것 같아서 좋았습니다. |

**６. 참고자료**

**6.1 벤치마킹**

|  |  |
| --- | --- |
| **분류** | **내용** |
| **홈페이지** | Yogiyo |
| **개요** | 배달 서비스 |
| **URL** | https://www.yogiyo.co.kr/ |
| **참고내용** | UI |
| **이미지** |  |

**6.2 API**

|  |  |
| --- | --- |
| **Naver Map API** | |
| **기능** | Javascript를 활용하여 주소값을 입력해 지도를 웹페이지에 나타내는 서비스 |
| **홈페이지** | https://developers.naver.com/docs/map/javascriptv3/ |
| **사용한 부분** | 현재 위치와 가게의 위치를 지도로 표시하는 기능 |
| **이미지** |  |