HOT Delivery

JSL 人材開発印 26期

うまうま

2017. 12. 15 ~ 2018. 01. 16

**< 目次 >**

**1. 企画案**

1.1 プロジェクト概要

1.2 開発環境

1.3 使用技術

1.4 チームメンバーの構成と役割

1.5 開発日程

**2.　要求事項明細書**

2.1 メインページ

2.2 会員ページ

2.3 店主ページ

**3. プログラム設計**

3.1 ER Sequence Diagram

3.2 Site Map

3.3 Table List

3.4 Table Structure

3.5 SQL Query

**4. UI 定義書**

4.1 メインページ

4.2 会員ページ

4.3 店主ページ

**5. 補足点や感想**

5.1 補足点

5.2 感想

**6. 参考資料**

6.1 **ベンチマーキング**

6.2 API

**1. 企画案**

**1.1 プロジェクト概要**

|  |  |
| --- | --- |
| **チーム名** | うまうま |
| **メンバー** | ハンウンチャン、イムチェフン、ジョンヨンジョン、ユンサンホ |
| **プロジェクト名** | HOT Delivery |
| **使用お客層** | 1. 配達サービスを希望するお客 2. お店を登録したいお客 |
| **開発期間** | 約　2017.12.15~2018.01.16 |
| **開発理由** | 1. 面白くて最敵の満足感を得られるように食べ物の注文をするために  2. 実用的な開発  3.スマートフォンのアプリケーションを作ることができる |

**1.2 開発環境**

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

**1.3 使用技術**

|  |  |
| --- | --- |
| **プログラミング言語** | JAVA JDK 1.8.0  JSP + Servlet  JavaScript + jQuery |
| **DB ERD Tool** |  |
| **Design Tool** | pixlr |

**1.4 チームのメンバーと役割**

|  |  |
| --- | --- |
| **ジョンヨンジョン** | 理想型ワールドカップ,会員登録,　修正,　脱退機能およびDB担当 |
| **ユンサンホ** | NAVER地図オープンAPIおよび決済機能担当 |
| **イムチェフン** | UI開発およびDTO設計,　グラフィック,　開発書,　開発資料の準備 |
| **ハンウンチャン** | 注文目録,　理想型ワールドカップ担当 |

**1.5 開発日程**

|  |  |
| --- | --- |
| **１週目**  **(12.18` ~ 12.25)** | アイディアと起案書作成,　DB設計および機能構想 |
| **2 週目**  **(12.26 ~ 01.01)** | メインページUIと機能開発  会員ページのUIと機能開発　(会員登録, ログイン, マイページ) |
| **3 週目**  **(01.02 ~ 01.08)** | 注文ページ機能開発  会員ページの重要な機能開発　(フードワールドカップ)  プログラムのデバッグ |
| **4 週目**  **(01.09 ~ 01.17)** | ページを日本語で翻訳  最適化やメンテナンス  発表資料製作と準備  模擬発表  最後発表 |

**2. 要求事項明細書**

**2.1 メインページ**

|  |  |  |
| --- | --- | --- |
| **機能** | **要求事項** | **詳細要求事項** |
| ログイン | ログインチェック | ログインしようとするIDを探してパスワードを確認し、ログイン。 |
| ID検索 | 会員登録時に登録したIDを探す。 |
| パスワード検索 | 会員加入の時に登録されている情報を通じてパスワードを再設定することができるようにする。 |
| 会員加入 | 新規登録 | 新規会員の場合、ID、パスワード、氏名、連絡先、メールを入力してもらうようにする。 |
| ID重複チェック | 従来のIDと新規登録しようとするIDが重複するかどうかを検査する。 |
| フェイスブック | フェイスブックに行く | フェイスブックに連結してイベントを確認できるようにする。 |
| 検索 | 食べ物名 | 食べたい食べ物を売っている店を出力する。 |
| ホーム | ホーム | メインページに戻れるようにする。 |

**2.2 会員ページ**

|  |  |  |
| --- | --- | --- |
| **機能** | **要求事項** | **詳しい 要求事項** |
| フードワールドカップ | 食べ物の名前 | 食べたい食べ物を選択して最後に選択された食べ物を売っている店を出力する |
| マイページ | 個人情報変更 | 会員の個人情報を修正できるようにする |
| 会員脱退 | 会員退会をできるようにする |
| 届け先設定 | 直接入力 | どこでも料理を配達させることができるので住所を入力を受けて配達を流動的に設定 |
| 使用者住所 | 会員加入の際に使用者の住所に配達するかを設定 |
| 地図 | 現在の位置と店の位置を表示 |
| 食べ物選択 | 店の情報 | 店と登録されている食べ物の情報を示し、数量を設定 |
| 食べ物情報 | 購買する食べ物、数量を設定できるようにする |
| 決済 | 決済 | 購買する食べ物、数量を出力 |
| 電話番号の認証 | 携帯を通じて本人認証をして決済をできるようにする |
| メールチェック | 加入時に登録しておいた電子メールと比較して決済をできるようにする |

**2.3 店主ページ**

|  |  |  |
| --- | --- | --- |
| **機能** | **要求事項** | **詳しい 要求事項** |
| マイページ | メニュー登録および修正 | メニューを登録、 修正できるようにする。 |
| 店の削除 | お店を削除できるようにする。 |

**3. プログラム設計**

**3.1 Table List**

|  |  |
| --- | --- |
| **テーブル名** | **説明** |
| deliveryMember | 会員情報を保存するテーブル |
| Primary Key : id |
| worldcup | 食べ物カテゴリーを保存するテーブル |
| . |
| store | お店の情報を保存するテーブル |
| Primary Key : storeNo |
| food | 食べ物の情報を保存するテーブル |
| Foreign Key : storeNo |

**3.2 Table Structure**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **テーブル名** | deliveryMember | | | |
| **テーブルの役割** | 会員情報を保存するテーブル | | | |
| **フィールドの役割** | **フィールド名** | **DATATYPE** | **NULL 許容** | **備考** |
| ID | id | VARCHAR2(20) |  | PK |
| パスワード | password | VARCHAR2(15) |  |  |
| お名前 | name | VARCHAR2(50) |  |  |
| お金 | money | VARCHAR2(10) |  | Default 0 |
| メール | email | VARCHAR2(50) |  |  |
| 住所 | address | VARCHAR2(100) |  |  |
| お店の住所 | storeAddress | VARCHAR2(100) | null |  |
| 使用目的 | userkind | VARCHAR2(10) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **テーブル名** | food | | | |
| **テーブルの役割** | 食べ物の情報を保存するテーブル | | | |
| **フィールドの役割** | **フィールド名** | **DATATYPE** | **NULL 許容** | **備考** |
| お店の番号 | storeNo | NUMBER(10) | NOT NULL | References store  (storeNo) |
| 食べ物の種類 | foodKind | NUMBER(5) | NOT NULL |  |
| 食べ物の名前 | foodName | VARCHAR2(50) | NOT NULL |  |
| 価格 | price | VARCHAR2(5) | NOT NULL |  |
| イメージ | imgUrl | VARCHAR2(100) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **テーブル名** | store | | | |
| **テーブルの役割** | お店の情報を保存するテーブル | | | |
| **フィールドの役割** | **フィールドの役割** | DATATYPE | NULL **許容** | **備考** |
| Sequence | StoreNo | NUMBER(10) |  | PK |
| 店の 名前 | storeName | VARCHAR2(50) |  |  |
| 店の電話番号 | storePhone | VARCHAR2(15) |  |  |
| 住所 | address | VARCHAR(100) |  |  |
| 硬度 | longitude | NUMBER(20,10) |  |  |
| 緯度 | latitude | NUMBER(20,10) |  |  |
| イメージの住所 | imgUrl | VARCHAR2(100) |  |  |
| 食べ物コード | foodcode | VARCHAR2(10) |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **テーブル名** | worldCup | | | |
| **テーブルの役割** | 食べ物の情報を保存するテーブル | | | |
| **フィールドの役割** | **フィールドの役割** | **DATATYPE** | **NULL 許容** | **備考** |
| 食べ物の種類 | foodKind | VARCHAR2(20) | NOT NULL | P |
| 食べ物番号 | foodNo | NUMBER(16) | NOT NULL |  |
| イメージURL | imgUrl | VARCHAR2(100) | NOT NULL |  |
| 食べ物コード | foodcode | VARCHAR2(10) |  |  |

**3.3 SQL Query**

|  |  |  |
| --- | --- | --- |
| **Package** | **Delivery.member.dao** | **会員管理** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| inserMember(MemberDTO dto) | Insert　into　deliveryMember(id,password,name,email,address,userkind) values(?,?,?,?,?,?) | 会員登録 |
| updateMember(MemberDTO dto) | update deliveryMember set password=?, email=?,address=? where id=? | 会員修正 |
| deleteMember(String id,String pwd) | delete from deliveryMember where id=? and password=? | 会員脱退 |
| idCheck(String id) | select id from deliveryMember where id=? | ID重複検索 |
| idcheck(String id,String pwd) | select id,password from deliveryMember where id=? and password=? | ログイン情報確認 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.woldcup.dao.MakeRandomFoodDAO** | **フードワールドカップ** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| selectKind() | select \* form worldCup; | 食べ物の情報リストに保存 |
| selectFood(int num) | Select \* from worldCup where foodNo=? | フード番号による食べ物の名前とイメージUrl選択 |
| setDTO(ArrayList<FoodDTO>) | Select \* from worldCup where foodNo=? | リストによる食べ物情報選択 |
| setUrl(int[] idx) | Select \* from worldCup where foodNo=? | インデックス値による食べ物情報選択 |
| setUrl\_2(ArrayList<Integer>) | Select \* from worldCup where foodNo=? | リストによる食べ物情報選択 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.food.dao.FoodDAO** | 食べ物情報 |
|  |  |  |
| **Method** | **SQL** | **Description** |
| CountFood(String search) | Select count(\*) from food where foodName Like search | 検索した食べ物の数 |
| searchFood(String search, int count) | Select \* from food where foodName LIKE search | 検索した食べ物の情報 |
| searchAllStore(ArrayList<integer> storeNoList) | Select \* from store order by storeNo asc | お店の番号の情報を昇順に整列 |
| storeNoList(int num) | Select storeNo from food where foodkind = num | お店の番号が同じものの重複を削除 |
| foodInfo() | Select \* from food | 食べ物の情報 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.store.dao.StoreListDAO** | **お店情報** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| selectCount(int num) | select count(\*) store storeNo =? | お店の番号の数 |
| storeList(ArrayList<integer> storeNoList) | Select \* from store | お店の情報 |
| storeNoList(int num) | Select storeNo from food where foodkind=num | 食べ物の種類のよった店番号+重複除去 |
| findFood(int storeNo) | Select \* from food where storeNo=storeNo | 店の番号のグルメ情報 |
| selectStoreInfo(int storeNo) | Select \* from store where storeNo=storeNo | 店の番号のよるお店の情報 |

|  |  |  |
| --- | --- | --- |
| **Class** | **Delivery.cart.dao.CartDAO** | **注文リスト** |
|  |  |  |
| **Method** | **SQL** | **Description** |
| int getMoney(String id) | select money from deliveryMember where id=? | 現在持っている金額を返還する |
| void insertCart(String id, ArrayList<FoodDTO> foodList,String address) | insert into cart(id,foodName,quantity,address,summoney) values(?,?,?,?,?) | 決済ができた場合、注文リストに入れる |
| void updateMoney(String id, int money) | update deliveryMember set money=? where id=? | 金額チャージ、決済した後決済した金額を修正する |
| ArrayList<CartDTO> showList(MemberDTO dto) | select \* from cart where id=? | 注文リストを見せる |

**4. UI定義書**

**4.1 メインページ**

|  |  |
| --- | --- |
| **メインページ** | |
| **機能** | 初期画面 |
| **URL** | main/Main.jsp |
| **PCバージョン** |  |
| **機能説明** | 上のメニューを通じて他のページでアクセスが可能で、  ログインした場合、マイページにアクセスすることができる。  食べ物の全体的なジャンルと詳細な食べ物の名前を通じてお店を探せる。  簡略な紹介やバナーが見える。 |

|  |  |
| --- | --- |
| **ログイン** | |
| **機能** | ログイン |
| **URL** | login/login.jsp |
| **PCバージョン** |  |
| **機能説明** | ログインボタンを押した際移動されるウィンドウ  ID、パスワードのチェックを通じてログインが可能で、IDおよびパスワード忘れた場合探すことができる。 |

|  |  |
| --- | --- |
| **ログイン** | |
| **機能** | IDを検索 |
| **URL** | login/login.jsp |
| **PCバージョン** | **C:\Users\JSLHRD\Desktop\이미지파일\아이디찾기.PNG** |
| **機能説明** | ID探すボタンを押した場合、出力されるポップアップウィンドウ  名前とメールチェックを通じてIDを探すことができる。 |

|  |  |
| --- | --- |
| **ログイン** | |
| **機能** | パスワードを検索 |
| **URL** | login/login.jsp |
| **PCバージョン** | **C:\Users\JSLHRD\Desktop\이미지파일\비밀번호찾기.PNG** |
| **機能説明** | 探すのボタンを押した場合、出力されるポップアップウィンドウ  IDと名前とメールチェックを通じてパスワードを探すことができる。 |

|  |  |
| --- | --- |
| **会員登録** | |
| **機能** | 会員登録 |
| **URL** | regist/UserCreate.jsp |
| **PCバージョン** |  |
| **機能説明** | メインページの会員登録ボタンを押すと、会員登録ページに移動  登録ボタンを押すと、登録ができ、重複されたIDがある場合、IDだけ再び作成すると、登録ができる。 |

|  |  |
| --- | --- |
| **フェイスブック** | |
| **機能** | フェイスブック |
| **URL** | https://www.facebook.com/ |
| **PC バージョン** |  |
| **機能説明** | Facebookボタンを押した時移動されるページ  イベントなどの情報を確認することができる。 |

**4.2 会員ページ**

|  |  |
| --- | --- |
| **フードワールドカップ** | |
| **機能** | フードワールドカップ |
| **URL** | worldcup/worldcup.jsp |
| **PCバージョン** |  |
| **機能説明** | メインページのフードワールドカップのボタンを押すと出力されるページ  ゲームスタートボタンを押すと選択できるページが出力される。 |

|  |  |
| --- | --- |
| **フードワールドカップ** | |
| **機能** | 食べ物選択ページ |
| **URL** |  |
| **PCバージョン** | C:\Users\JSLHRD\Desktop\KakaoTalk_20180115_015018742.png |
| **機能説明** | 様々な料理カテゴリーの中で理想のワールドカップゲームのように食べ物を選択することで注文ページに移動 |

|  |  |
| --- | --- |
| **注文** | |
| **機能** | 食べ物数量を選択 |
| **URL** | order/order.jsp |
| **PCバージョン** | **C:\Users\JSLHRD\Desktop\이미지파일\음식수량.PNG** |

|  |  |
| --- | --- |
| **注文** | |
| **機能** | 届け先を設定 |
| **URL** | order/order.jsp |
| **PCバージョン** | **C:\Users\JSLHRD\Desktop\이미지파일\주소지.PNG** |
| **機能説明** | 届け先住所を設定して現在の位置と店の位置を地図に表示 |

|  |  |
| --- | --- |
| **注文** | |
| **機能** | 決済確認 1 |
| **URL** | order/order.jsp |
| **PCバージョン** |  |
| **機能説明** | 右の三角の様子のボタンを押すと注文しようとする金額を表示 |

|  |  |
| --- | --- |
| **注文** | |
| **機能** | 決済確認 2 |
| **URL** | storelist/payment.jsp |
| **PC バージョン** |  |
| **機能説明** | 名前とメールアドレス、支払い金額を表示して電話番号を通じた情報確認 |

|  |  |
| --- | --- |
| **検索** | |
| **機能** | 検索した料理店 |
| **URL** | liststore/liststore.jsp |
| **PC バージョン** |  |
| **機能説明** | メインページの検索機能を利用して検索した料理の名前を売っているお店の場所と名前、住所、電話番号を表示する。 |

|  |  |
| --- | --- |
| **マイページ** | |
| **機能** | 初期画面 |
| **URL** | mypage/mypage.jsp |
| **PC バージョン** |  |
| **機能説明** | 会員ログインに成功した場合には、アクセスできるページ |

|  |  |
| --- | --- |
| **マイページ** | |
| **機能** | 会員情報修正 |
| **URL** | mypage/mypage.jsp |
| **PC バージョン** |  |
| 機能説明 | マイページの会員情報修正ボタンを押すと会員情報を修正することができる。 |

|  |  |
| --- | --- |
| **会員ページ** | |
| **機能** | 会員脱退 |
| **URL** | mypage/mypage.jsp |
| **PC バージョン** |  |
| **機能説明** | マイページの会員脱退のボタンを押すと会員脱退ができる。 |

|  |  |
| --- | --- |
| **会員ページ** | |
| **機能** | 金額チャージ |
| **URL** | mypage/mypage.jsp |
| **PC バージョン** |  |
| **機能説明** | 会員ページで金額チャージメニューを押すと表れる  チャージボタンを押すと金額がチャージされる |

|  |  |
| --- | --- |
| **注文リスト** | |
| **機能** | 注文リスト |
| **URL** | cart/Cart.jsp |
| **PC バージョン** |  |
| **機能説明** | メインページの注文リストのボタンを押すと注文した食べ物や住所など注文情報を示される。 |

**4.3 店主ページ**

|  |  |
| --- | --- |
| **会員ページ** | |
| **機能** | 店のメニュー登録 |
| **URL** | mypage/mypage.jsp |
| **PC バージョン** |  |
| **機能説明** | 店主のログインに成功した場合メニュー登録および修正のためのページ(開発予定) |

**5.　補足点や感想**

**5.1 補足点**

|  |  |
| --- | --- |
| **分類** | **詳細内容** |
| **UI** | ページ最適化  モバイル環境での可読性向上 |
| **機能** | 決済機能実現  店のメニュー登録および修正機能実現 |
| **性能** | 多様な状況でのNullPointerException  繰り返されるロジックのクラス/モジュール化  有効性検査精度向上 |
| **保安** | Command Patternを活用したユーザ任意のQueryString防止  多数接続によるDB Connection管理  SQL injection予防 |

**5.2 感想**

|  |  |
| --- | --- |
| **ジョンヨンジョン** | 今回のプロジェクトをしながら様々なことを感じました。メンバーがお互いにコミュニケーションを通じて各自が担当する部分をうまく遂行したようでよかったです。リーダーの役割が重要だということを感じるようになり、チーム間のコミュニケーション、役割分担が難しいということを実感しました。 結果的にメンバーたちからもたくさん学びましたし、満足できるなチームプロジェクトでした。 |
| **ユンサンホ** | 初めてオープンAPIを使用してみましたが、それを使用して機能を適用することにより、面白さを感じ、各自の役割をみんな優しくしてくれてありがたく思っています。結果的に多くのことを学び、様々なことを気付いた経験でした。 次に機会があれば、またこのプロジェクトを利用してもっと機能を追加したホームページを作りたいです。 |
| **ハンウンチャン** | 初めてのウェブプロジェクトで今まで習ったことを最大限活用して進行してみましたが、まだ実力が不足な点が多いということを気づきました。しかし、チーム員とのコミュニケーションもうまくできましたし、各自の役割で忠実に臨んだと考えられてチーム員たちの繋がりも深まり、さらにプロジェクトが終わった頃には自らが発展されたようなことを感じられて良かったです。 |
| **イムチェフン** | 今回のチームプロジェクトで主な担当部分が初めて接するフロント部分で始めた時には多くの壁がありましたが、一ページ作るたびにデザインが願う様子に変わるのが感じられて最後までやり通すことができました。大変でしたが、そのおかげで最近IT市場で要求するフルスタックエンジニアに一歩近づいたようでよかったです. |

**６. 参考資料**

**6.1 ベンチマーキング**

|  |  |
| --- | --- |
| **分類** | **内容** |
| **ホームページ** | Yogiyo |
| **概要** | 出前サービス |
| **URL** | https://www.yogiyo.co.kr/ |
| **参考内容** | UI |
| **イメージ** |  |

**6.2 API**

|  |  |
| --- | --- |
| **Naver Map API** | |
| **機能** | Javascriptを活用して住所を入力して地図をウエブページに出力するサービス |
| **ホームページ** | https://developers.naver.com/docs/map/javascriptv3/ |
| **使用機能** | 現在の位置とお店の位置を表紙する機能 |
| **イメージ** |  |