

Warm Place

STUN(**S**trongest in the **U**niverse) 팀

CONTENTS

01

상황분석

02

서비스 구성

03

구현계획

04

프로토타입
및
코드 설명

05

비전

01 상황분석 - (1) 현 상황 및 시장분석

데이터로 본 코로나19 “소비자 심리(좌)” 및 “소비 트렌드(우)” 키워드



불안감, 장기화	감염증, 팬데믹	경기침체, 하락세
우울함, 무기력증	세계경제, 금융시장	직격탄



언택트, 비대면	홈코노미, 집콕족, 집밥, HMR(가정간편식), Home	
재택근무, 웹세미나	가속화, 성장세	포스트 코로나

코로나의 장기화에 심리적으로 우울함과 불안감을 느끼는 소비자들
사회적 거리두기로 인해 변하고 있는 소비 트렌드

01 상황분석 - (2) 서비스소개

<div>브랜드</div> <div>항목</div>	WARM PLACE	NAVER	<div>테이블링 TABLING</div> 	<div>다이닝코드</div>  <div>AR 커머셜 플랫폼, 와간다</div>
가게 정보 제공	O	O	O	O
예약 가능	O	O	O	O
웨이팅	O	X	O	X
인원 수 표기	O	X	X	X
부가 기능	주변 놀거리 및 가게 추천 포인트 제도	주변 가게 위치		포인트 제도

01 상황분석 - (3) 자사 SWOT분석

S

Strength

디테일한 정보제공Ex) 매장 안 손님 수, 자기 웨이팅 번호,
유사한 키워드를 가진 가게 추천 등**O**

Opportunity

상권 활성화**W**

Weakness

인지도, 사용자 부족

SNS

포인트

T

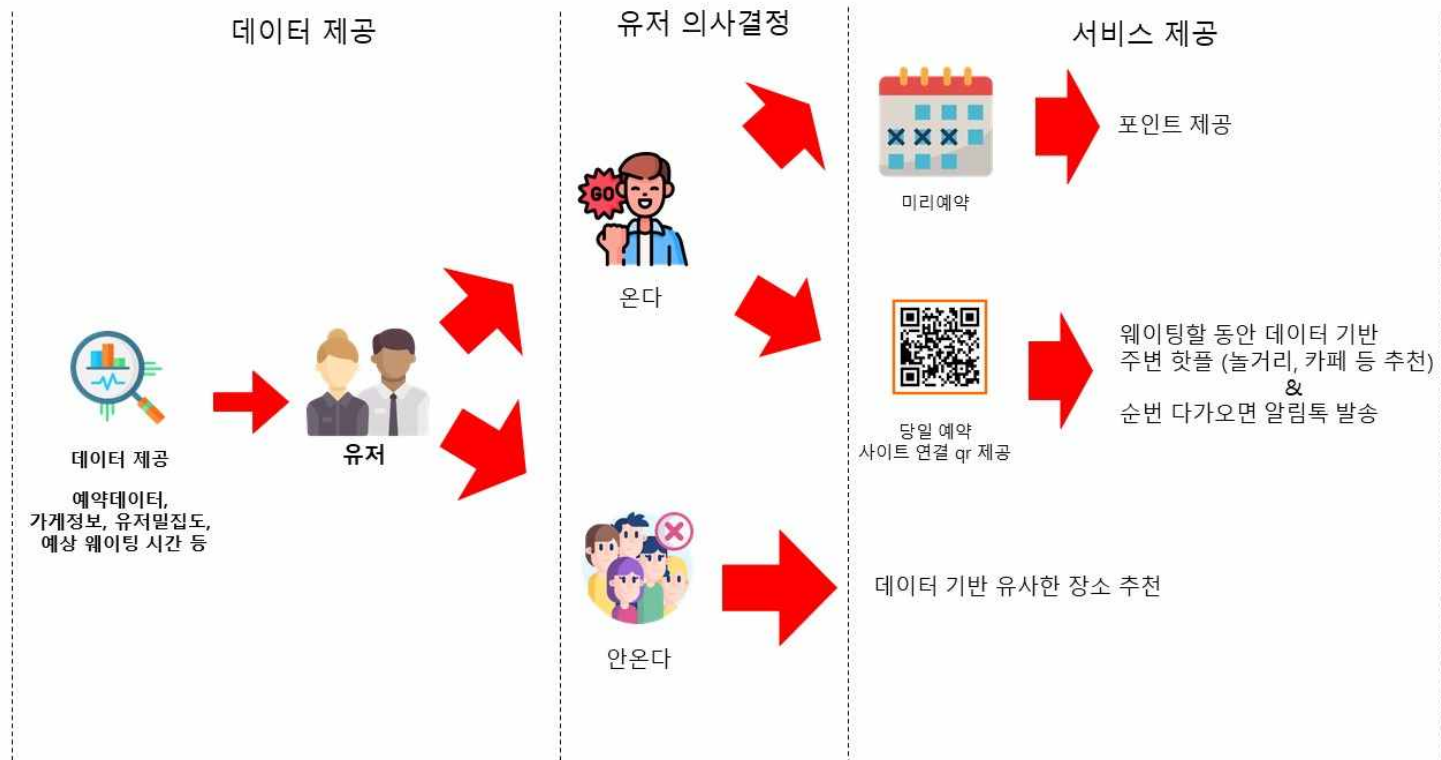
Threat

타 업체의 모방

Ex) 네이버, 카카오

차별화된 디테일한 정보를 제공하는 강점은 강조하고 인지도가 부족한 약점을 보완하는데 중점을 둬

02 서비스 구성도



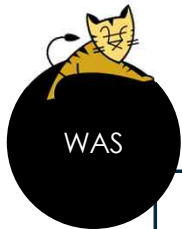
03 구현계획 - (1)개발일정

팀원 \ 일 정	1차 시모델	2차 원본	단위 테스트	통합 테스트
박정민(PM)	팀 구성 및 비즈니스 모델 설계 모델링	비즈니스 모델 설계 및 점검	단위 테스트 시나리오 작성	최종 점검
조용현(기획)	아이디어 기획 및 설계 모델링	아이디어 기획 및 개발	테스트 시나리오 검토, 기능 확인	
온창민(개발)	기획 및 설계 모델링	아이디어 기획 및 DB 구현 및 설계	모듈 확인 오류 수정	
황민영(개발)	기획 및 설계 모델링	아이디어 기획 및 DB 구현 및 설계	모듈 확인 오류 수정	
이재혁(디자인)	앱 기능 및 UI 모델링	앱 기능 및 UI 구현	테스트 시나리오 검토, UI 기능 확인	

03 구현계획 - (2) 소프트웨어 아키텍처



03 구현계획 - (3) 웹 어플리케이션 서버 & 데이터베이스



-가게 추천을 위한 로직

- 1) 데이터베이스에 저장된 가게테이블에서 가게좌표를 받아와서 사용자 거리의 일정범위 내 (예 100M이내)에 있는 가게를 가져온다.
- 2) 데이터베이스에 저장된 소비자가 원하는 키워드를 가진 가게 리스트를 가져온다.
- 3) JAVA 코드를 활용해 FOR반복문으로 받아온 객체를 비교하여 일정범위 내에 있는 가게 중 소비자가 원하는 키워드를 가진 가게들을 추려 웹 서버에 넘겨준다.

STORE_T						
가게 테이블						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓			가게ID	STORE_ID	INT
		✓		가게이름	STORE_NAME	VARCHAR(45)
		✓		가게좌표	STORE_GPS	VARCHAR(45)
		✓		가게대기번호	STORE_WNUM	VARCHAR(45)

KEYWORD_ID_T						
키워드 테이블						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓			키워드ID	KEYWORD_ID	INT
		✓		가게ID	STORE_ID	INT

KEYWORD_INFO_T						
Table logical name						
PK	AI	FK	Null	Logical Name	Name	Type
✓	✓			키워드정보	KEYWORD_INFO	VARCHAR(45)
		✓		키워드ID	KEYWORD_ID	INT



DB
AWS

가게 좌표를 구글지도나 네이버지도 API를 통해 데이터 수집,
한 가게는 여러 개의 키워드를 참조

04 서비스 디자인 - (1) 메인

Warm Place

홈

로그인 회원가입 예약하기 맛집후기 마이페이지

메인 기능들

가고싶은 장소를 미리 보고
싶었던 적 있지않나요??

코로나 시대에도 안전하게 놀러갑니다!!

실시간 확인

예약하기

예약하기



실시간 장소확인

장소를 입력해주세요.

Call to action →

04 서비스 디자인 - (2) 회원가입

Warm Place

[로그인](#) [회원가입](#) [예약하기](#) [맛집후기](#) [마이페이지](#)

Warm Place에 오신 것을 환영합니다!

회원가입을 시작하겠습니다.

회원가입[회원가입](#)[다시입력](#)**정보입력****입력 초기화**

04 서비스 디자인 - (3) 예약하기



위치
표시

04 서비스 디자인 - (4) 맛집후기

Warm Place

[로그아웃](#) [회원가입](#) [예약하기](#) [맛집후기](#) [마이페이지](#)

번호	제목	작성자	작성일
1	맛...있습니다...	onebin	2021-08-12 09시36분
2	맛있어용	love_nh	2021-08-12 09시45분
3	예약을 쉽게 할 수 있어요	gg	2021-08-12 10시39분
4	스시 맛집이네용	stun	2021-08-12 11시22분
5	1	greatpark	2021-08-12 13시08분

글

다음

다음페이지

글쓰기

글쓰기 버튼

04 코드 - (1) 로그인액션/ 로그아웃 액션

```
request.setCharacterEncoding("utf-8");

// 현재 세션 상태를 체크한다
String userID = null;
if(session.getAttribute("userID") != null){
    userID = (String)session.getAttribute("userID");
}
// 이미 로그인했으면 다시 로그인을 할 수 없게 한다
if(userID != null){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('이미 로그인에 되어 있습니다.');");
    script.println("location.href='../St_Join/main.jsp'");
    script.println("</script>");
}
st_UserDAO userDAO = new st_UserDAO();
st_UserDTO User = new st_UserDTO();

int result = userDAO.login(user.getUserID(),user.getUserPassword());
if(result == 1){
    // 로그인에 성공하면 세션을 부여
    session.setAttribute("userID",user.getUserID());
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('로그인 성공.');");
    script.println("location.href='../St_Join/main.jsp'");
    script.println("</script>");
}else if(result == 0){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('비밀번호가 틀립니다.');");
    script.println("history.back()");
    script.println("</script>");
}else if(result == -1){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('존재하지 않는 아이디입니다.');");
    script.println("history.back()");
    script.println("</script>");
}else if(result == -2){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('데이터베이스 오류입니다.');");
    script.println("history.back()");
    script.println("</script>");
}
}
```

```
<title>Worm Place_로그아웃액션</title>
</head>
<body>
    <%
        session.invalidate();
    %>
    <script>
        alert('로그아웃 되었습니다.')
        location.href='../St_Join/main.jsp';
    </script>
</body>
```

로그인 액션에서 로그인 확인/아이디,비밀번호 확인

로그아웃 에서 세션 무효화 후 메인페이지로 연결

04 코드 - (2) 회원가입

```
//하나라도 빈칸있는 경우
if(NewUser.getUserID().equals("") || NewUser.getUserPassword().equals("") || NewUser.getName().equals("") || NewUser.getEmail().equals("") || NewUser.getNumber().equals(""))
{
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('다시 확인해주세요.');"");
    script.println("</script>");
    script.close();
    return;
}
```

빈칸 있을 경우 확인하는 알람창

```
//ID와 비밀번호가 모두 있는 경우
st_UserDAO userDAO = new st_UserDAO();
int result = userDAO.join(NewUser.getUserID(), NewUser.getUserPassword(), NewUser.getName(), NewUser.getEmail(), NewUser.getNumber());
if(result == -1){
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('이미 존재하는 아이디입니다');"");
    script.println("history.back();"");
    script.println("</script>");
}
else{
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('회원가입 성공!!!');"");
    script.println("location.href='../St_Join/main.jsp');"");
    script.println("</script>");
    script.close();
    return;
}
```

아이디 중복 확인

정상적 작동시
회원가입 성공

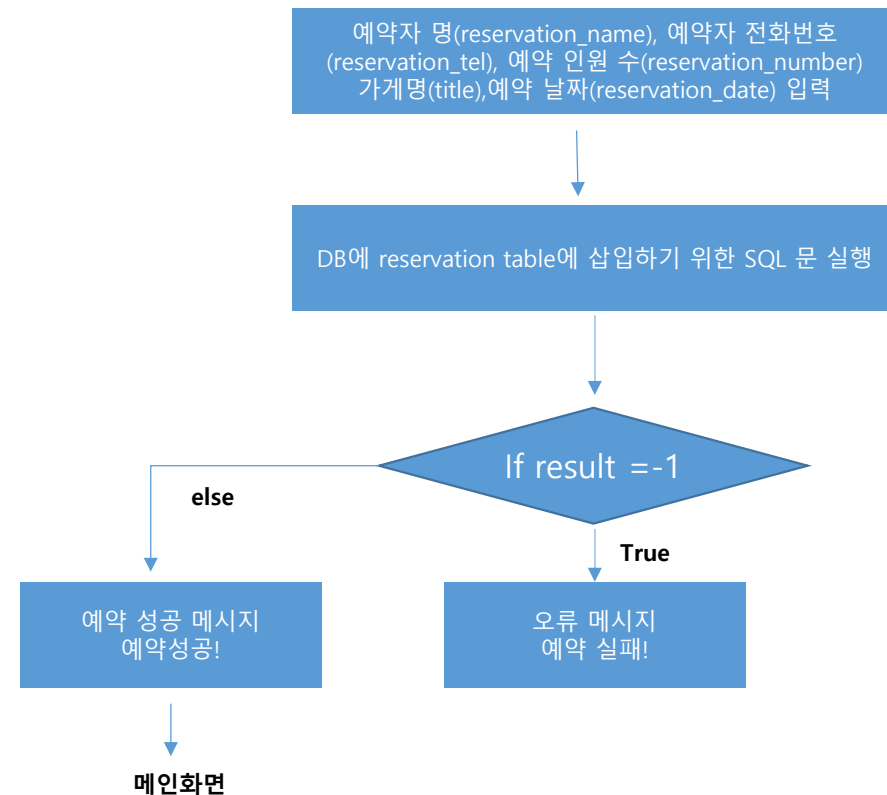
04 코드 - (3) 예약하기

```

public int insert_reservation(ReservationDTO
reservation) {
    String sql = "INSERT INTO
reservation(user_name, user_tel, number,
store_name, reservation_date) VALUES(?,?,?,?,?)";
    try {
        st_UserDAO user = new st_UserDAO();
        out.println(request.getParameter("title"));
        ReservationDTO reservDTO = new
ReservationDTO(request.getParameter("reservation_name"),
request.getParameter("reservation_tel"),
request.getParameter("reservation_number"),
request.getParameter("title"),
request.getParameter("reservation_date"));
        int result = user.insert_reservation(reservDTO);
        if (result == -1) {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('예약 실패')");
            script.println("history.back()");
            script.println("</script>");
        }
        else {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('예약 성공!!!')");
            script.println("location.href='../St_Join/main.jsp'");
            script.println("</script>");
            script.close();
            return;
        }
    }
}

```

예약하기 화면



04 코드 - (4)_1 맛집후기_표 첫 행

```
<!-- 게시판 메인 페이지 영역 시작 -->
<div class="container px-5 my-5 px-5 " style="width:1000px;">
  <div class="row">
    <table class="table table-striped" style="text-align: center; border: 1px solid #dddddd">
      <thead>
        <tr>
          <th style="background-color: #eeeeee; text-align: center;">번호</th>
          <th style="background-color: #eeeeee; text-align: center;">제목</th>
          <th style="background-color: #eeeeee; text-align: center;">작성자</th>
          <th style="background-color: #eeeeee; text-align: center;">작성일</th>
        </tr>
      </thead>
      <tbody>
```

1. 맛집 후기 표 첫 행

부트스트랩을 사용하여 기본적으로(아무 후기 글이 없을 때) 후기 글에 나올 열 지정

04 코드 - (4)_2 맛집후기_표 내용

```

<tbody>
<%
    BbsDAO bbsDAO = new BbsDAO(); // 인스턴스 생성
    ArrayList<Bbs> list = bbsDAO.getList(pageNumber);
    for(int i = 0; i < list.size(); i++){
%>

        <td><%= list.get(i).getBbsID() %></td> --> --%>
    <tr>
        <td><%= (i+1) %></td>
        <!-- 게시물 제목을 누르면 해당 글을 볼 수 있도록 링크를 걸어둔다 -->
        <td><a href="view.jsp?bbsID=<%= list.get(i).getBbsID() %>">
            <%= list.get(i).getBbsTitle() %></a></td>
        <td><%= list.get(i).getUserID() %></td>
        <td><%= list.get(i).getBbsDate().substring(0, 11) + list.get(i).getBbsDate().substring(11, 13) + "시"
            + list.get(i).getBbsDate().substring(14, 16) + "분" %></td>
    </tr>
<%
    }
%>
</tbody>

```

2. 맛집 후기 표 내용 부분

게시판 DAO 인스턴스를 생성하고 저장된 DB에서 각각 제목,작성자ID, 작성 시간을 가져옴
 첫번째 td태그에는 글에 번호를 부여하는 부분이고
 두번째 td태그에는 view.jsp 링크를 걸어서 후기 제목을 클릭하면 내용을 볼 수 있는 페이지로 이동

04 코드 - (4)_3 맛집후기_수정 및 삭제

```
<%  
// 현재 세션 상태를 체크한다  
String userID = null;  
if (session.getAttribute("userID") != null) {  
    userID = (String) session.getAttribute("userID");  
}  
%>
```

세션 확인

작성자 본인이면 글을
수정하거나 삭제 가능.
로그인 세션을 통해 확인 후
DB에서 userID를 대조하여 일치 시,
수정과 삭제 링크가 있는 버튼을 사용가능.

```
<!-- 해당 글의 작성자가 본인이라면 수정과 삭제가 가능하도록 코드 추가 -->  
<%  
if (userID != null && userID.equals(bbs.getUserID())) {  
%>  
<a href="update.jsp?bbsID=<%=bbsID%>" class="btn btn-primary">수정</a>  
<a href="deleteAction.jsp?bbsID=<%=bbsID%>" class="btn btn-primary">삭제</a>  
<%  
}  
%>
```

04 코드 - (4)_4 맛집후기_쓰기

```
<%
// 현재 세션 상태를 체크한다
String userID = null;
if (session.getAttribute("userID") != null) {
    userID = (String) session.getAttribute("userID");
}
%>
```

```
<%
if (userID == null) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('로그인을 하세요')");
    script.println("location.href='../St_board/login.jsp'");
    script.println("</script>");
} else {
    %>
```

```
<!-- 게시판 글쓰기 양식 영역 시작 -->
<div class="container">
    <div class="row">
        <form method="post" action="writeAction.jsp">
            <table class="table table-striped" style="text-align: center; border: 1px solid #dddddd">
                <thead>
                    <tr>
                        <th colspan="2" style="background-color: #eeeeee; text-align: center;">게시판 글쓰기 양식</th>
                    </tr>
                </thead>
                <tbody>
                    <tr>
                        <td><input type="text" class="form-control" placeholder="글 제목" name="bbsTitle" maxlength="50"></td>
                    </tr>
                    <tr>
                        <td><textarea class="form-control" placeholder="글 내용" name="bbsContent" maxlength="2048" style="height: 350px;"></td>
                    </tr>
                </tbody>
            </table>
            <!-- 글쓰기 버튼 생성 -->
            <input type="submit" class="btn btn-primary pull-right" value="글쓰기">
        </form>
    </div>
</div>
<!-- 게시판 글쓰기 양식 영역 끝 -->
```

마찬가지로 로그인 세션을 활용,
로그인을 한 사람만 후기를 작성.
후기 글 제목,내용을 작성 한 후
post 방식으로 writeAction.jsp로 넘김.

04 코드 - (4)_4 맛집후기_쓰기액션1

```
// 로그인한 한 사람만 글을 쓸 수 있도록 코드를 수정한다
if (userID == null) {
    PrintWriter script = response.getWriter();
    script.println("<script>");
    script.println("alert('로그인을 하세요')");
    script.println("location.href='login.jsp'");
    script.println("</script>");
} else {
    // 입력이 안 된 부분이 있는지 체크한다
    if (bbs.getBbsTitle() == null || bbs.getBbsContent() == null) {
        PrintWriter script = response.getWriter();
        script.println("<script>");
        script.println("alert('입력이 안 된 사항이 있습니다')");
        script.println("history.back()");
        script.println("</script>");
    } else {
        // 정상적으로 입력이 되었다면 글쓰기 로직을 수행한다
        BbsDAO bbsDAO = new BbsDAO();
        int result = bbsDAO.write(bbs.getBbsTitle(), userID, bbs.getBbsContent());
        // 데이터베이스 오류인 경우
        if (result == -1) {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글쓰기에 실패했습니다')");
            script.println("history.back()");
            script.println("</script>");
            // 글쓰기가 정상적으로 실행되면 알림창을 띄우고 게시판 메인으로 이동한다
        } else {
            PrintWriter script = response.getWriter();
            script.println("<script>");
            script.println("alert('글쓰기 성공')");
            script.println("location.href='bbs.jsp'");
            script.println("</script>");
        }
    }
}
```

writeAction.jsp

로그인이 되어있다면 입력 사항을 체크
한 후 이상이 없다면

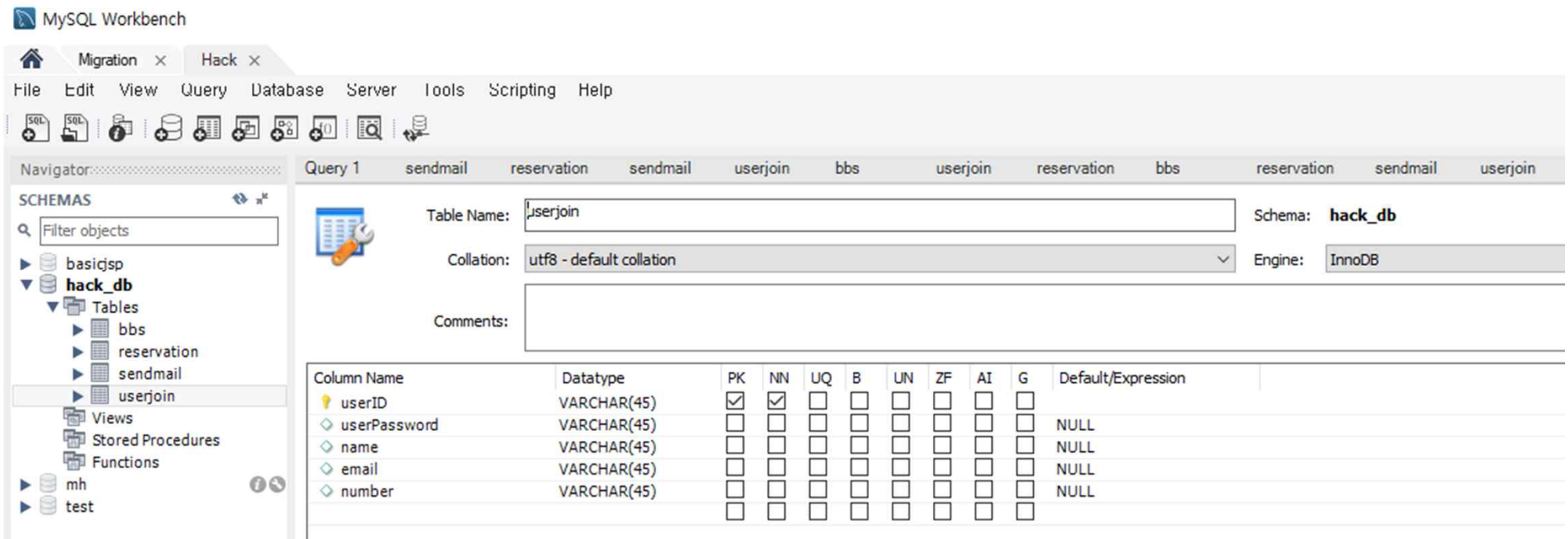
후기의 제목과 작성자의 userID, 내용을
bbsDAO 객체를 생성.

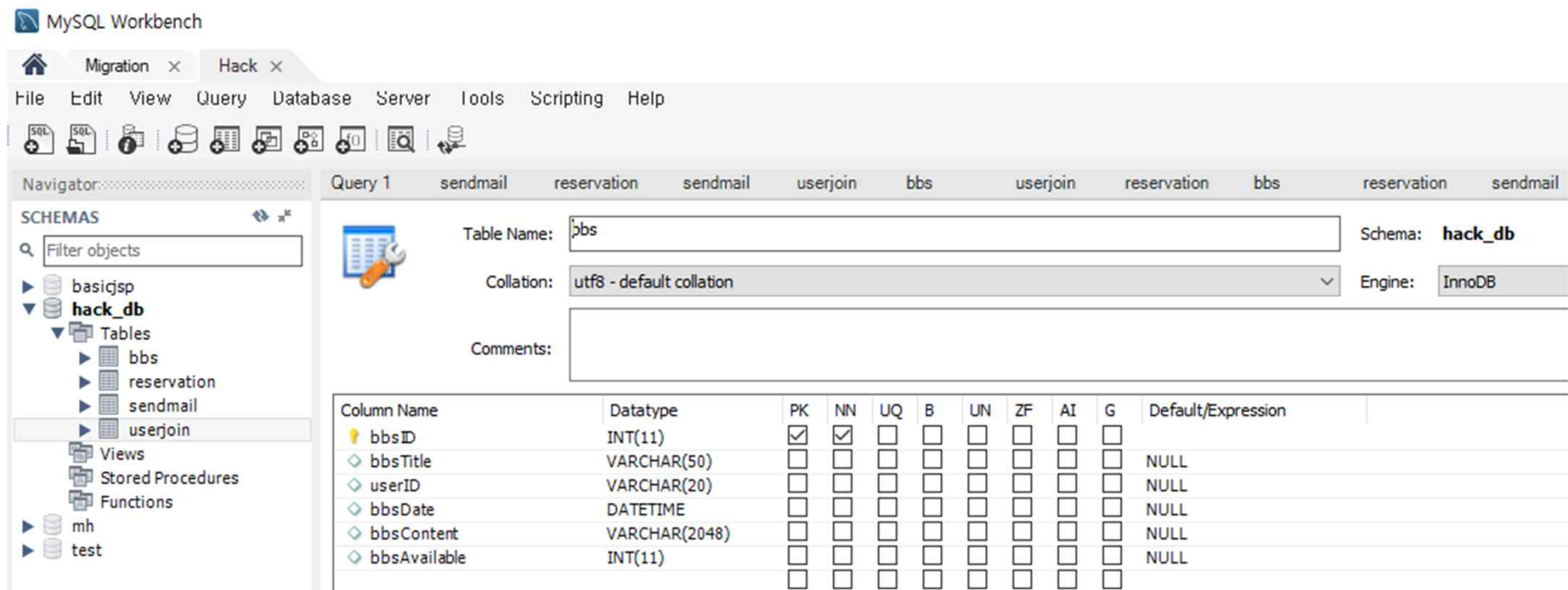
04 코드 - (4)_4 맛집후기_쓰기액션2

```
// 글쓰기 메소드
public int write(String bbsTitle, String userID, String bbsContent) {
    String sql = "insert into bbs values(?, ?, ?, ?, ?, ?)";
    try {
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, getNext());
        pstmt.setString(2, bbsTitle);
        pstmt.setString(3, userID);
        pstmt.setString(4, getDate());
        pstmt.setString(5, bbsContent);
        pstmt.setInt(6, 1); // 글의 유효번호
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1; // 데이터베이스 오류
}
```

bbs.DAO 객체의 write 메소드를 통해 전달받은 제목, userID, 내용을 sql문을 통해 DB로 최종 전달.
(getNext(),getDate()함수는 bbs.DAO안에서 생성)

맛집 후기 게시물 수정하기 및 삭제하기(update.jsp, updateAction.jsp, deleteAction)기능 또한 게시물 쓰기와 같은 과정을 진행





[illegible]

Warm Place

로그인 회원가입 예약하기 맛집후기 마이페이지

가고싶은 장소를 미리 보고 싶었던 적 있지않나요??

코로나 시대에도 안전하게 놀러갑시다!!

실시간 확인

예약하기



실시간 장소확인

장소를 입력해주세요.

Call to action →

비디오 녹화 - 시작

시연영상

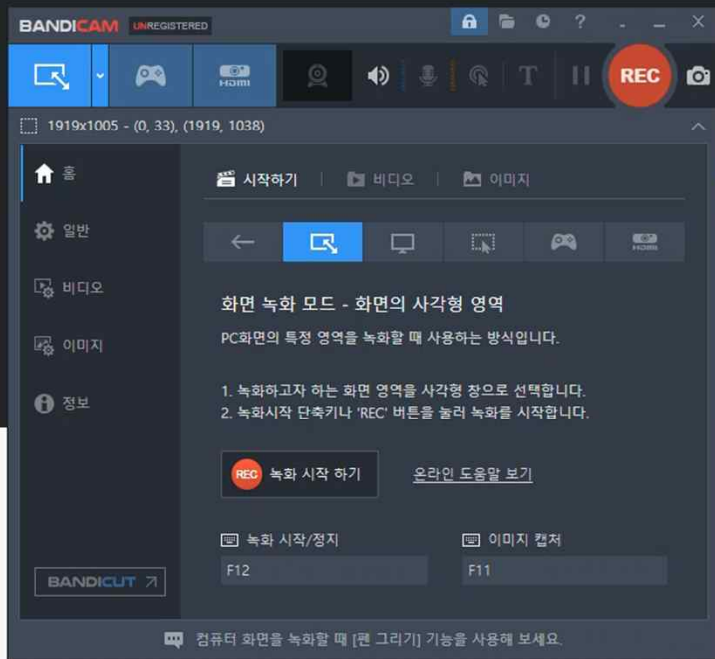
rojects/St_Join/main.jsp

www.BANDICAM.com



Warm Place

[로그인](#) [회원가입](#) [예약하기](#) [맛집후기](#) [마이페이지](#)



은 장소를 미리 보고 던 적 있지않나요??

르나 시대에도 안전하게 놀러갑니다!!

[실시간 확인](#)

[예약하기](#)



실시간 장소확인

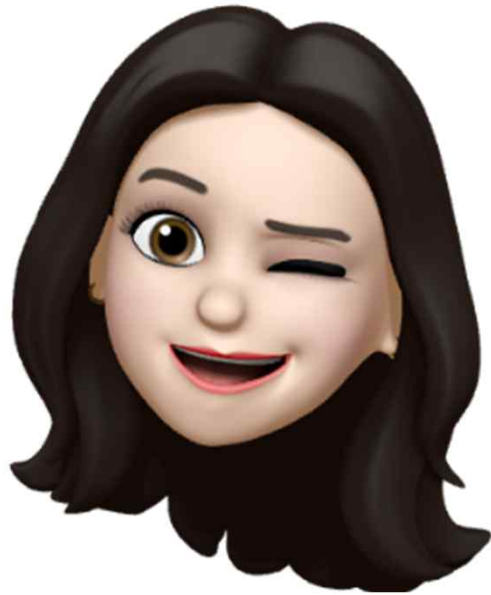
장소를 입력해주세요.

[Call to action →](#)

05 비전 - 향후 계획

- UI 개선
- QR 코드 및 안심번호 데이터 활용 가능 시 , 확장된 기능 추가
- 코드 간결화를 통해 성능향상
- 데이터베이스 기술 향상으로 성능 개선

Contact ME



박정민(Jaunnnngs)

E-mail: jaunnnngs@naver.com

Github: <https://github.com/jaunnnngs>