

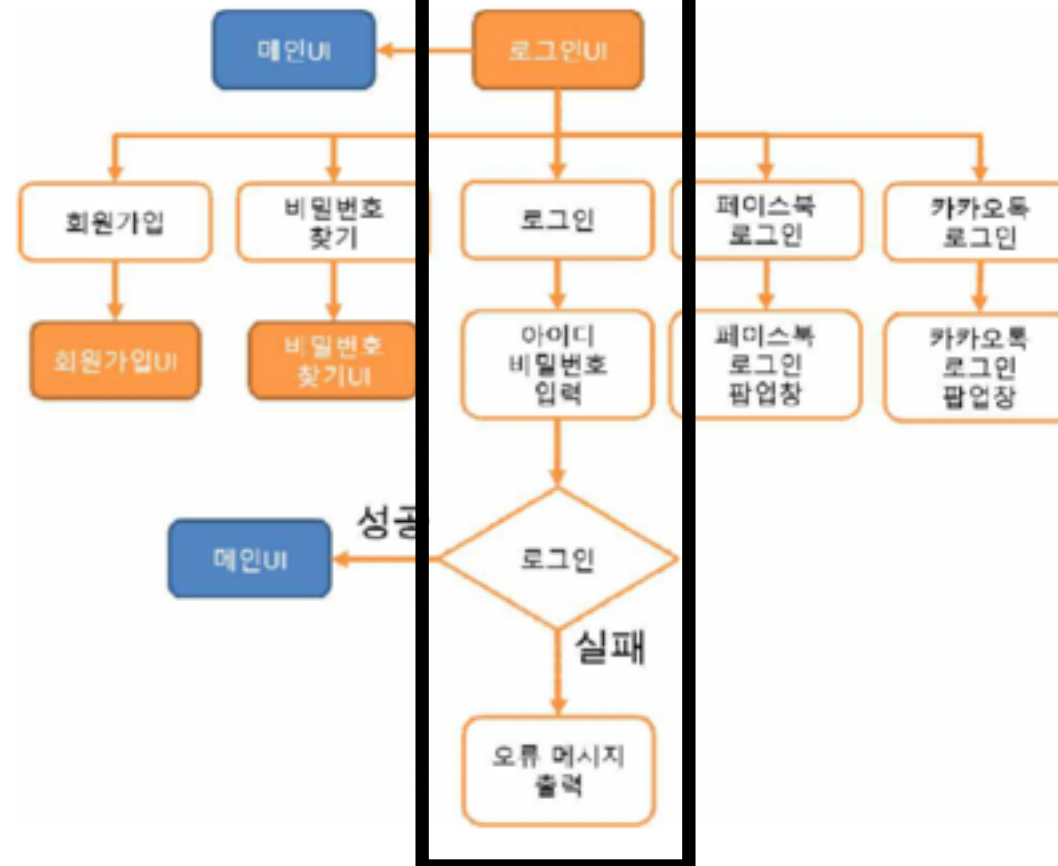
로그인UI

로그인기능 구현

기획서 - (2)

3.1 로그인UI 상세 시스템 흐름도와 화면UI 그리고 기능설명

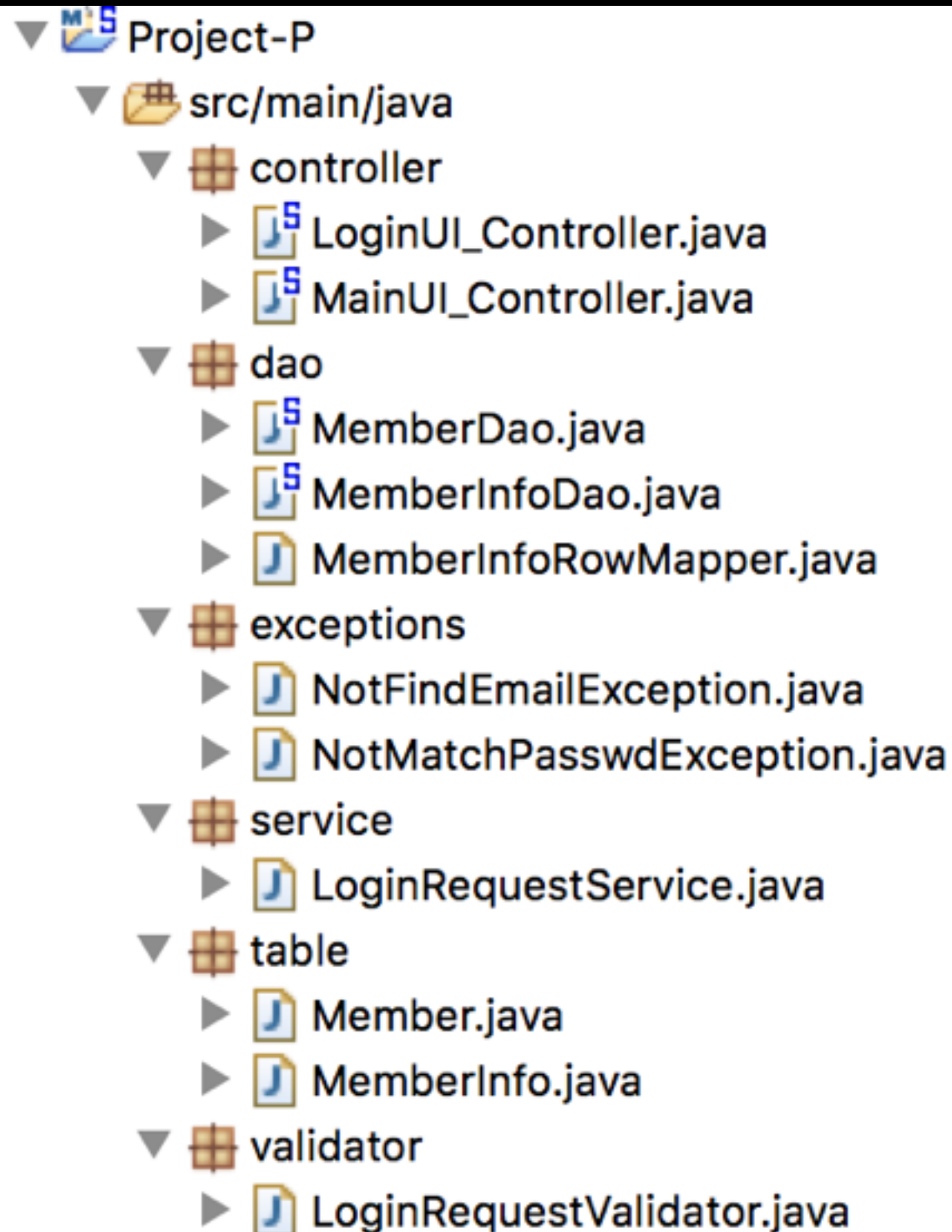
3.1.1 로그인UI 상세 시스템 흐름도



3.1.2 로그인UI 기능 설명

- 로그인UI에서 상단의 Pikicast버튼을 누르면 메인UI로 이동한다.
- 로그인UI에서 회원가입 텍스트링크를 누르면 회원가입UI로 이동한다.
- 로그인UI에서 비밀번호찾기 텍스트링크를 누르면 비밀번호찾기UI로 이동한다.
- 로그인UI에서 아이디, 비밀번호를 입력하고 로그인버튼을 누르면 로그인한다.
- 로그인 시 로그인이 성공한다면 메인UI로 이동한다.
- 로그인 시 아이디가 틀리다면 "가입한 사용자가 아니에요"라는 메시지를 출력한다.
- 로그인 시 비밀번호가 틀리다면 "잘못된 아이디 또는 비밀번호를 입력하셨습니다"라는 메시지를 출력한다.
- 로그인 시 탈퇴한 회원의 아이디, 비밀번호를 입력하면 "이 분은 이미 피키캐스트를 떠나셨어요"라는 메시지를 출력한다.

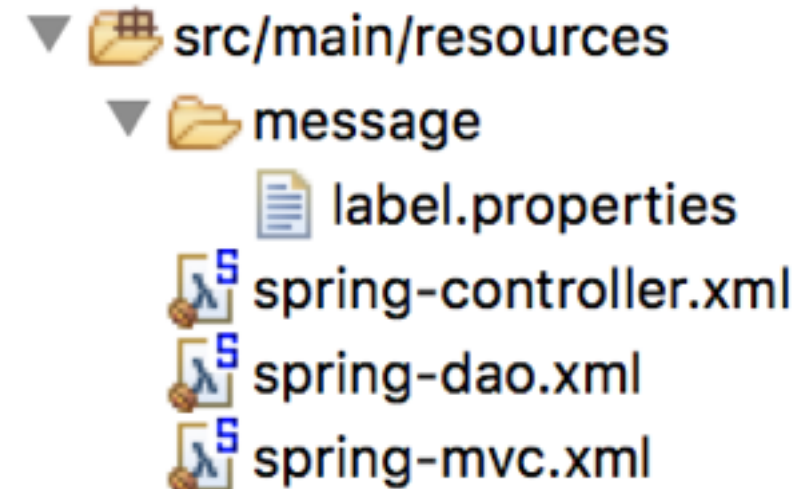
프로젝트 패키지



```
▼ Project-P
  ▼ src/main/java
    ▼ controller
      ▶ LoginUI_Controller.java
      ▶ MainUI_Controller.java
    ▼ dao
      ▶ MemberDao.java
      ▶ MemberInfoDao.java
      ▶ MemberInfoRowMapper.java
    ▼ exceptions
      ▶ NotFindEmailException.java
      ▶ NotMatchPasswdException.java
    ▼ service
      ▶ LoginRequestService.java
    ▼ table
      ▶ Member.java
      ▶ MemberInfo.java
    ▼ validator
      ▶ LoginRequestValidator.java
```

Project-P

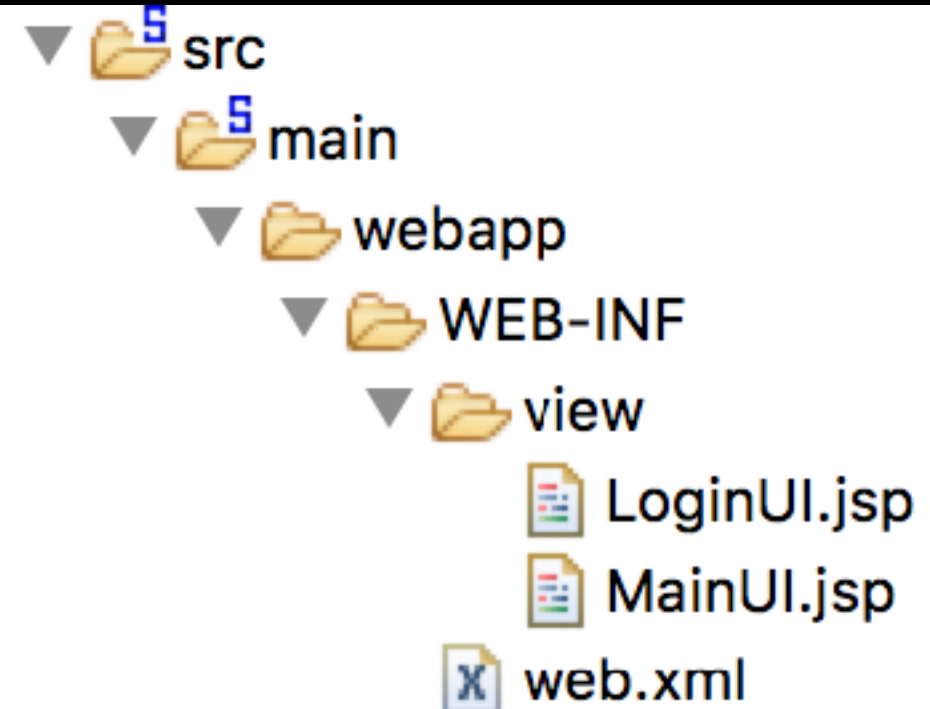
- src/main/java
 - controller
 - LoginUI_Controller.java
 - MainUI_Controller.java
 - dao
 - MemberDao.java
 - MemberInfoDao.java
 - MemberInfoRowMapper.java
 - exceptions
 - NotFindEmailException.java
 - NotMatchPasswdException.java
 - service
 - LoginRequestService.java
 - table
 - Member.java
 - MemberInfo.java
 - validator
 - LoginRequestValidator.java



```
▼ src/main/resources
  ▼ message
    label.properties
    spring-controller.xml
    spring-dao.xml
    spring-mvc.xml
```

src/main/resources

- message
 - label.properties
 - spring-controller.xml
 - spring-dao.xml
 - spring-mvc.xml



```
▼ src
  ▼ main
    ▼ webapp
      ▼ WEB-INF
        ▼ view
          LoginUI.jsp
          MainUI.jsp
          web.xml
```

src

- main
 - webapp
 - WEB-INF
 - view
 - LoginUI.jsp
 - MainUI.jsp
 - web.xml

로그인UI-메인UI

LoginUI_Controller.java MainUI.jsp LoginUI.jsp Project-P Login UI

http://localhost:8080/Project-P/login/main

Logo_Main

이메일 :

비밀번호 :

로그인

페이스북 로그인

카카오톡 로그인

회원가입

비밀번호 찾기

로그인UI-LoginUI_Controller

```
LoginUI_Controller.java
1 package controller;
2
3 import org.springframework.stereotype.Controller;
4
14 @Controller
15 @RequestMapping("/login")
16 public class LoginUI_Controller {
17     private MemberInfoDao memberInfoDao;
18
19
20     public LoginUI_Controller(MemberInfoDao memberInfoDao){
21         this.memberInfoDao = memberInfoDao;
22     }
23
24     @RequestMapping("main")
25     public String showLoginUI(Model model) {
26         model.addAttribute("memberInfo", new MemberInfo());
27         return "LoginUI";
28     } //end login(Model model)
29
30     @RequestMapping("loginRequest")
31     public String loginRequest(MemberInfo memberInfo, Errors errors, Model model) {
32         new LoginRequestValidator().validate(memberInfo, errors);
33         if(errors.hasErrors())
34             return "LoginUI";
35
36         String email = memberInfo.getEmail();
37         String passwd = memberInfo.getPasswd();
38         try{
39             LoginRequestService.authenticate(memberInfoDao, email, passwd);
40             //변수명을 Login_Email_Text -> MainUI에서 받는 변수명으로 변경해야함.
41             model.addAttribute("Login_Email_Text", email);
42             return "MainUI";
43         } catch (NotFoundEmailException e){
44             e.printStackTrace();
45             return "LoginUI";
46         } catch (NotMatchPasswdException e){
47             e.printStackTrace();
48             return "LoginUI";
49         } //end try
50     } //end loginRequest(MemberInfo memberInfo, Errors errors)
51 } //end class LoginUI_Controller
```

로그인UI-LoginUI_Controller

```
LoginUI_Controller.java
30 @RequestMapping("loginRequest")
31 public String loginRequest(MemberInfo memberInfo, Errors errors, Model model) {
32     new LoginRequestValidator().validate(memberInfo, errors);
33     if(errors.hasErrors())
34         return "LoginUI";
35
36     String email = memberInfo.getEmail();
37     String passwd = memberInfo.getPasswd();
38     try{
39         LoginRequestService.authenticate(memberInfoDao, email, passwd);
40         //변수명을 Login_Email_Text -> MainUI에서 받는 변수명으로 변경해야함.
41         model.addAttribute("Login_Email_Text", email);
42         return "MainUI";
43     }catch(NotFindEmailException e){
44         e.printStackTrace();
45         return "LoginUI";
46     }catch(NotMatchPasswdException e){
47         e.printStackTrace();
48         return "LoginUI";
49     }//end try
50 }//end loginRequest(MemberInfo memberInfo, Errors errors)
51 }//end class LoginUI_Controller
```

- 로그인UI에서 아이디, 비밀번호를 입력하고 로그인버튼을 누르면 로그인한다.
- 로그인 시 로그인이 성공한다면 메인UI로 이동한다.
- 로그인 시 아이디가 틀리다면 "가입한 사용자가 아니에요"라는 메시지를 출력한다.
- 로그인 시 비밀번호가 틀리다면 "잘못된 아이디 또는 비밀번호를 입력하셨습니다"라는 메시지를 출력한다.

로그인 실패 - 존재하지 않는 email을 입력하거나 비밀번호가 틀렸을 경우

```
1 package service;
2
3 import dao.MemberInfoDao;
4
5
6
7
8 public class LoginRequestService {
9
10     public static void authenticate(MemberInfoDao memberInfoDao, String email, String passwd){
11         MemberInfo memberInfo = memberInfoDao.selectByEmail(email);
12
13         if(memberInfo == null)
14             throw new NotFindEmailException("존재하지 않는 email입니다.");
15         else if(!memberInfo.getPasswd().equals(passwd))
16             throw new NotMatchPasswdException("아이디 또는 비밀번호가 일치하지 않습니다.");
17     } //end authenticate(MemberInfoDao memberInfoDao, String email, String password)
18
19 } //end class LoginRequestService
```

- 로그인 시 아이디가 틀리다면 "가입한 사용자가 아니에요"라는 메시지를 출력한다.
- 로그인 시 비밀번호가 틀리다면 "잘못된 아이디 또는 비밀번호를 입력하셨습니다"라는 메시지를 출력한다.

로그인 실패 - 존재하지 않는 email을 입력하거나 비밀번호가 가 틀렸을 경우

LoginUI_Controller.java MainUI.jsp LoginUI.jsp

http://localhost:8080/Project-P/login/loginRequest

Logo_Main

이메일 :

비밀번호 :

로그인

페이스북 로그인

카카오 로그인

회원가입

비밀번호 찾기

Markers Problems Servers Console Spring Explorer Boot Dashboard Progress

Tomcat v8.0 Server at localhost (1) [Apache Tomcat] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (2016. 7. 23. 오후 1:10:10)

```
exceptions.NotFoundEmailException: 존재하지 않는 email입니다.  
    at service.LoginRequestService.authenticate(LoginRequestService.java:14)  
    at controller.LoginUI_Controller.loginRequest(LoginUI_Controller.java:39)  
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)  
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)  
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)  
    at java.lang.reflect.Method.invoke(Method.java:497)  
정보: Server startup in 3655 ms  
exceptions.NotMatchPasswdException: 아이디 또는 비밀번호가 일치하지 않습니다.  
    at service.LoginRequestService.authenticate(LoginRequestService.java:16)  
    at controller.LoginUI_Controller.loginRequest(LoginUI_Controller.java:39)
```

```
1 package exceptions;  
2  
3 public class NotFoundEmailException extends RuntimeException {  
4  
5     /**  
6      *  
7      */  
8     private static final long serialVersionUID = 1L;  
9  
10    public NotFoundEmailException(String message){  
11        super(message);  
12    }  
13 }
```

```
1 package exceptions;  
2  
3 public class NotMatchPasswdException extends RuntimeException {  
4  
5     /**  
6      *  
7      */  
8     private static final long serialVersionUID = 1L;  
9  
10    public NotMatchPasswdException(String message){  
11        super(message);  
12    }  
13 }
```


로그인UI-MemberInfoDao

```
MemberInfoDao.java
1 package dao;
2
3 import java.util.List;
4
5
6
7
8
9
10 public class MemberInfoDao {
11     private JdbcTemplate jdbcTemplate;
12
13     public MemberInfoDao(DataSource dataSource) {
14         this.jdbcTemplate = new JdbcTemplate(dataSource);
15     } // end MemberInfoDao(DataSource dataSource)
16
17     public void insert(MemberInfo memberInfo) {
18
19     }
20
21     public List<MemberInfo> selectAll() {
22         String sql = "select * from DvelopDB.memberInfo";
23
24         List<MemberInfo> results = jdbcTemplate.query(sql, new MemberInfoRowMapper());
25
26         return results;
27     } // end selectAll()
28
29     public MemberInfo selectByEmail(String email) {
30         String sql = "select * from memberInfo where email = ?";
31
32         List<MemberInfo> result = jdbcTemplate.query(sql, new MemberInfoRowMapper(), email);
33
34         return result.isEmpty() ? null : result.get(0);
35     } // end selectByEmail(String email)
36 } // end class MemberInfoDao
```

로그인UI-MemberInfoRowMapper

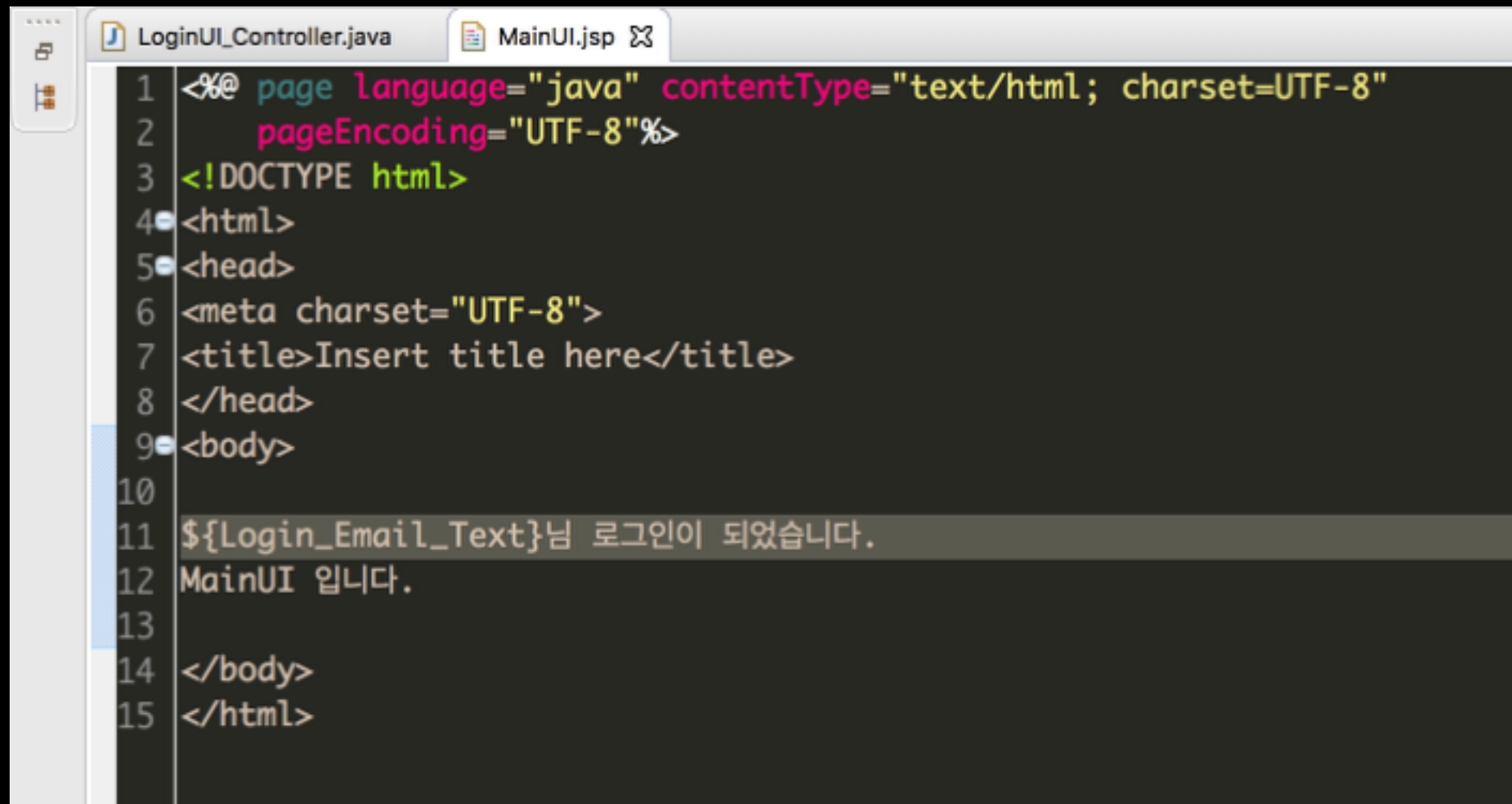
MemberInfoRowMapper.java

```
1 package dao;
2
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5
6 import org.springframework.jdbc.core.RowMapper;
7
8 import table.MemberInfo;
9
10 public class MemberInfoRowMapper implements RowMapper<MemberInfo> {
11     @Override
12     public MemberInfo mapRow(ResultSet rs, int rowNum) throws SQLException {
13         // TODO Auto-generated method stub
14         MemberInfo memberInfo = new MemberInfo();
15         memberInfo.setEmail(rs.getString("email"));
16         memberInfo.setPasswd(rs.getString("passwd"));
17         memberInfo.setBirth_date(rs.getString("birth_date"));
18         memberInfo.setReg_date(rs.getString("reg_date"));
19         memberInfo.setSex(rs.getString("sex"));
20         return memberInfo;
21     }
22 }
```

로그인UI-MemberInfo

```
MemberInfo.java
1 package table;
2
3 public class MemberInfo {
4
5     String email;
6     String passwd;
7     String reg_date;
8     String birth_date;
9     String sex;
10
11     public String getEmail() {
12         return email;
13     }
14
15     public void setEmail(String email) {
16         this.email = email;
17     }
18
19     public String getPasswd() {
20         return passwd;
21     }
22
23     public void setPasswd(String passwd) {
24         this.passwd = passwd;
25     }
26
27     public String getReg_date() {
28         return reg_date;
29     }
30
31     public void setReg_date(String reg_date) {
32         this.reg_date = reg_date;
33     }
34 }
```

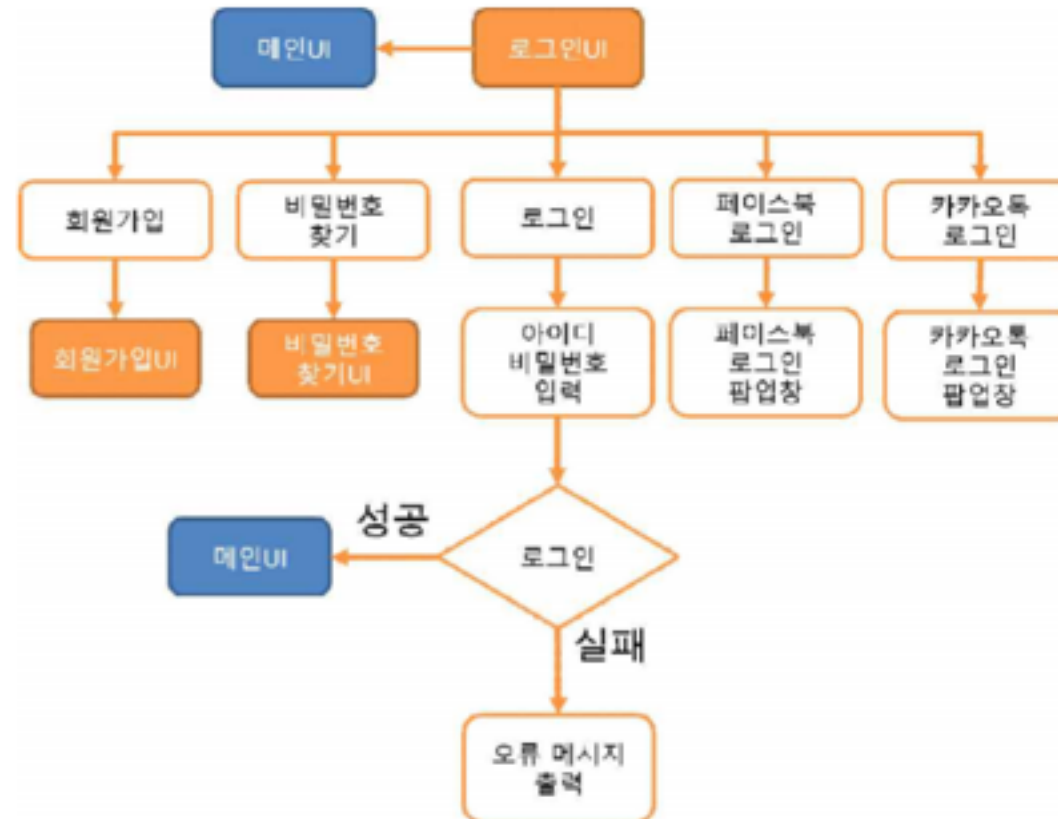
로그인 성공



로그인 실패 - 이메일, 비밀번호를 입력하지 않았을 경우

3.1 로그인UI 상세 시스템 흐름도와 화면UI 그리고 기능설명

3.1.1 로그인UI 상세 시스템 흐름도



3.1.2 로그인UI 기능 설명

- 로그인UI에서 상단의 Pikicast버튼을 누르면 메인UI로 이동한다.
- 로그인UI에서 회원가입 텍스트링크를 누르면 회원가입UI로 이동한다.
- 로그인UI에서 비밀번호찾기 텍스트링크를 누르면 비밀번호찾기UI로 이동한다.
- 로그인UI에서 아이디, 비밀번호를 입력하고 로그인버튼을 누르면 로그인한다.
- 로그인 시 로그인이 성공한다면 메인UI로 이동한다.
- 로그인 시 아이디가 틀리다면 "가입한 사용자가 아니에요"라는 메시지를 출력한다.
- 로그인 시 비밀번호가 틀리다면 "잘못된 아이디 또는 비밀번호를 입력하셨습니다"라는 메시지를 출력한다.
- 로그인 시 탈퇴한 회원의 아이디, 비밀번호를 입력하면 "이 분은 이미 피키캐스트를 떠나셨어요"라는 메시지를 출력한다.

로그인 실패 - 이메일, 비밀번호를 입력하지 않았을 경우

LoginUI_Controller.java MainUI.jsp LoginUI.jsp Project-P Login UI

http://localhost:8080/Project-P/login/loginRequest

Logo_Main

이메일 :

필수항목입니다.

비밀번호 :

필수항목입니다.

로그인

페이스북 로그인

카카오 로그인

회원가입

비밀번호 찾기

```
10 <body>
11
12 <form action="/Project-P/main/showMain" method="post">
13   <input type="submit" name="go_Main" value="Logo_Main"> <br>
14 </form>
15
16 <form:form action="loginRequest" commandName="memberInfo">
17   <label>이메일 : </label> <form:input path="email"/><br>
18   <form:errors path="email"/><br><!-- 에러메세지를 출력하는 form:errors 태그 -->
19   <label>비밀번호 : </label> <form:password path="passwd"/><br>
20   <form:errors path="passwd"/><br><!-- 에러메세지를 출력하는 form:errors 태그 -->
21   <input type="submit" value="로그인">
22 </form:form>
23
```

```
1 package validator;
2
3 import org.springframework.validation.Errors;
4
5
6
7
8 public class LoginRequestValidator implements Validator {
9
10  @Override
11  public boolean supports(Class<?> arg0) {
12    // TODO Auto-generated method stub
13    return LoginRequestValidator.class.isAssignableFrom(arg0);
14  } // supports(Class<?> arg0)
15
16  @Override
17  public void validate(Object target, Errors errors) {
18    // TODO Auto-generated method stub
19    // 검사대상객체
20    MemberInfo memberInfo = (MemberInfo) target;
21
22    // email이 비어있다면 검사대상객체의 email프로퍼티의 에러코드로 required를 추가
23    if (memberInfo.getEmail() == null || memberInfo.getEmail().trim().isEmpty())
24      errors.rejectValue("email", "required");
25    // password가 비어있다면 검사대상객체의 password프로퍼티의 에러코드로 required를 추가
26    if (memberInfo.getPasswd() == null || memberInfo.getPasswd().trim().isEmpty())
27      errors.rejectValue("passwd", "required");
28  } // validate(Object target, Errors errors)
29 } // class LoginRequestValidator
```

```
label.properties
1 required=필수항목입니다.
```