

## 객체지향설계 설계지침서 (전공설계)

## 1. 설계 주제

웹메일 시스템과 같은 네트워크 표준을 구현하는 웹 애플리케이션을 객체지향 분석/설계 방법을 적용하여 개발

## 2. 설계 내용

주어진 웹메일 시스템 프로젝트에 대한 유지보수 단계를 가정한다. 유지보수의 유형은 다음과 같다. 기존 코드의 오류를 수정하는 **교정(corrective)** 유지보수, 새로운 환경 변화에 적응하는 **적응(adaptive)** 유지보수, 새로운 기능 추가, 변경을 위한 **완전화(perfective)** 유지보수, 유지 보수성, 신뢰성 향상을 위한 구조 변경(리팩토링)과 같은 **예방(preventive)** 유지보수 등이 있다. 설계 프로젝트를 통하여 이러한 4가지 유형의 유지보수를 적용했음을 보여 줄 수 있어야 한다.

### 3. 설계 목적

웹메일 시스템과 같은 애플리케이션 개발을 통하여 웹 애플리케이션 제작 및 배포 전 과정에 대한 이해를 증진한다.

#### 4. 설계 구성 요소

| 목표/기준 설정 | 합 성 | 요구분석 | 구 현 | 시 험 | 평 가 | 결과 도출 | 기 타 |
|----------|-----|------|-----|-----|-----|-------|-----|
|          |     | √    | √   | √   |     |       |     |

|      |                                                                |
|------|----------------------------------------------------------------|
| 요구분석 | 문제 정의와 작업 분해를 통하여 시스템 범위를 이해하고 유지보수와 관련된 요구사항을 도출하고, 분석할 수 있다. |
| 구 현  | 정해진 일정에 맞추어 프로젝트를 진행하여 계획한 유지보수 유형을 기존 시스템에 구현할 수 있다.          |
| 시 험  | 교정/적응/완전화/예방 유지보수 적용 여부를 테스트를 통하여 보여줄 수 있다.                    |

## 5. 현실적 제한조건

| 경 제 | 환 경 | 사 회 | 윤 리 | 미 학 | 보건/안전 | 생산성/내구성 | 산업표준 | 기타 |
|-----|-----|-----|-----|-----|-------|---------|------|----|
|     |     |     |     |     |       | √       | √    | √  |

|      |                                                                                                                        |
|------|------------------------------------------------------------------------------------------------------------------------|
| 생산성  | 웹 응용 프로그램을 작성하기 위하여 모든 구성요소를 직접 프로그램하지 않고 레거시 시스템 또는 라이브러리를 활용하여 코딩의 효율성 및 테스트가 보다 간편해짐을 이해하고 활용할 수 있다.                |
| 산업표준 | 프로젝트를 수행하기 위하여 객체지향 분석, 설계의 기본 요소인 클래스 표기 방법인 UML 클래스 다이어그램을 활용할 수 있어야 하며, 적절한 SW 개발 방법론과 테스트, 형상 관리 기법을 적용할 수 있어야 한다. |
| 기타   | 개발 환경은 NetBeans IDE를 사용하여야 하며, 자바용 프로젝트 관리 도구인 maven을 사용하여 개발을 할 수 있다.                                                 |

## 5. 프로젝트 운영 방법

- 강의시간 이외의 시간을 이용하여 팀별로 프로젝트를 진행한다.
- 팀별 진행 상황을 파악할 수 있도록 최소한 주별 회의록을 작성하여야 한다. 회의록은 프로젝트 최종 발표시 제출하여야 한다.
- notion, slack 등과 같은 온라인 협업 도구를 사용하여 프로젝트 일정 관리, 할 일, 의사 소통을 해야 한다. (온라인 협업 도구에서 정리한 회의 내용을 주별 회의록에 복사)
- github.com 등의 서비스를 이용하여 git을 사용하여 소스 코드와 관련 산출물 모두에 대해 형상 관리를 해야 한다.

## 6. 팀 구성

- 팀 구성원은 4명 이하로 제한한다. 단, 담당 교수가 허락한 경우는 예외로 한다.
- 프로젝트 주제가 결정되면 구성원별 담당 영역을 나누어 프로젝트를 수행하고 프로젝트 결과 보고서에 구성원별 수행 결과를 명시하여야 한다.

## 7. 진행 일정 (10주)

- 4주 - 팀 구성
- 6주 - 프로젝트 계획서 제출
- 7주 - 프로젝트 회의록 확인
- 매주 - 프로젝트 중간 점검 (일부 조, 선택적)
- (11주 - 프로젝트 회의록 확인)
- 13주 - 프로젝트 발표 및 결과보고서 제출

## 8. 프로젝트 평가 방법

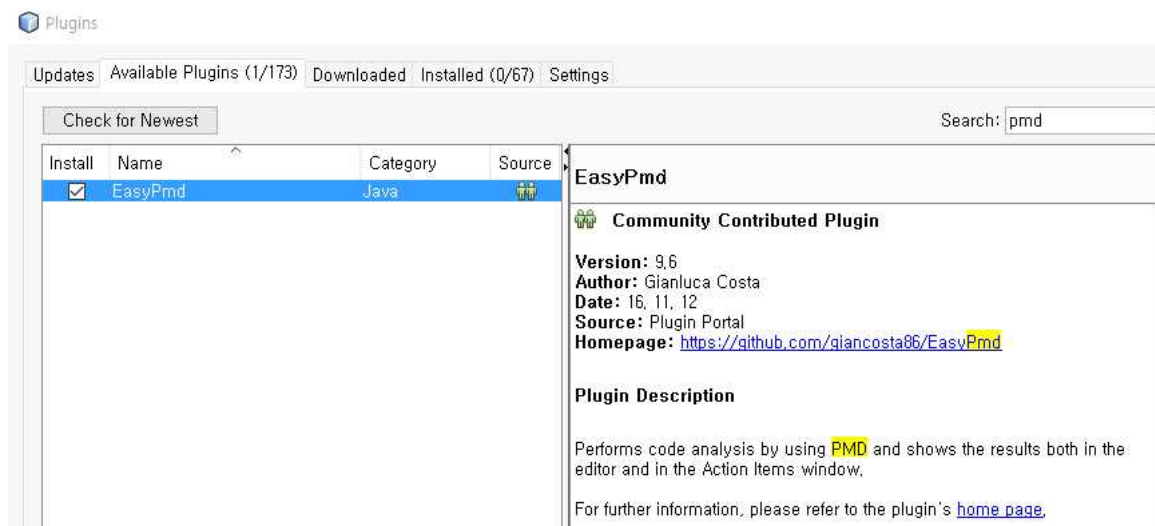
- 프로젝트 결과보고서에는 설계 제한 요소에 언급된 요소가 모두 포함되어야 하며, 설계 제한 요소 간 관계를 쉽게 알 수 있도록 보고서에 기술하여야 한다.
- 위에 기술한 내용에 근거하여 팀별로 평가한다.
- 또한 자신을 제외한 다른 팀원에 대한 팀 공헌도 평가를 통하여 최종 평가시 반영하도록 한다.

## 9. 최종 제출 자료(평가 사항)

아래 기술한 자료를 프로젝트 최종 발표일에 제출하여야 하며, 또한 동일 자료를 CD ROM에 저장하여 같이 제출하여야 한다. (주의: 컴퓨터 바이러스에 감염된 자료를 제출하는 경우 감점 요인이 됨.)

- 프로젝트 계획서
- 회의록 (주별, 임시 등)
- 프로젝트 최종 보고서
- 프로젝트 발표 자료 (파워포인트로 작성하여야 하며, 25장 이내로 프로젝트 관련 내용을 발표하여야 함.)
- 프로젝트 관련 소스 코드 : 프로젝트 열기를 통하여 내용을 확인할 수 있어야 함.

## 10. 교정 유지보수를 위한 도구: EasyPMD, FindBugs 플러그인



Plugins

Updates

Available Plugins (173)

Downloaded

Installed (0/67)

Settings

Check for Newest

| Install                             | Name                             | Category       | Sour... |
|-------------------------------------|----------------------------------|----------------|---------|
| <input type="checkbox"/>            | DeltaHex Editor                  | Editing        |         |
| <input type="checkbox"/>            | DisplayDependencies              | Tools          |         |
| <input type="checkbox"/>            | Display readme files in proje... | Tools          |         |
| <input type="checkbox"/>            | Doctrine2 Framework              | PHP            |         |
| <input type="checkbox"/>            | DukeScript Project Wizard        | HTML5          |         |
| <input checked="" type="checkbox"/> | EasyPmd                          | Java           |         |
| <input type="checkbox"/>            | Eclipse Java Code Formatter...   | Editing        |         |
| <input type="checkbox"/>            | Eclipse Java Code Formatter...   | Editing        |         |
| <input type="checkbox"/>            | Eclipse Java Code Formatter...   | Editing        |         |
| <input type="checkbox"/>            | Editor zoom                      | Editing        |         |
| <input type="checkbox"/>            | EOF Line Feed                    | Editing        |         |
| <input type="checkbox"/>            | Explore from here                | Infrastructure |         |
| <input type="checkbox"/>            | Fair Minesweeper!                | Games          |         |
| <input checked="" type="checkbox"/> | FindBugs Integration             | Java           |         |
| <input type="checkbox"/>            | FreeMarkerPlugin                 | Languages S... |         |
| <input type="checkbox"/>            | FXML 2 JAVA Convector            | JavaFX 2       |         |
| <input type="checkbox"/>            | Gdbserver                        | C/C++          |         |
| <input type="checkbox"/>            | gitignore.io                     | Versioning     |         |
| <input type="checkbox"/>            | Git toolbar                      | Versioning     |         |
| <input type="checkbox"/>            | Gluon Plugin                     | JavaFX 2       |         |
| <input type="checkbox"/>            | Gradle Support                   | Gradle         |         |

FindBugs Integration

Certified Plugin

Version: 1.32

Date: 16, 9, 30

Source: Certified Plugins

Homepage: <http://www.netbeans.org/>

Plugin Description

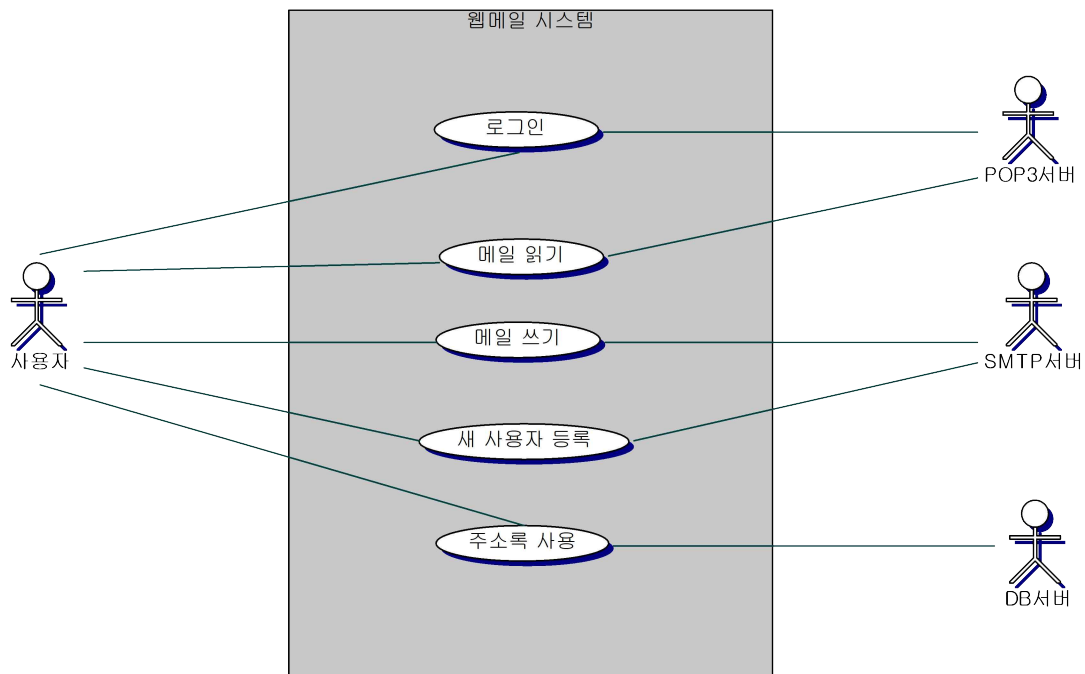
FindBugs Integration

Install

2 plugins selected, 8MB

# 웹메일 시스템

## 1. 웹메일 시스템 use case diagram



### 1.1 로그인

액터: 사용자, POP3서버

목적: 웹메일 시스템에 로그인한다.

개요:

입력된 사용자 정보를 이용하여 사용자 인증을 한다.

유형: 기본

이벤트 순서

- 1) 사용자가 userid와 password를 입력한다.
- 2) 시스템은 입력된 userid와 password를 사용하여 POP3 서버 인증을 한다.

- 3) userid와 password가 일치하면 메뉴화면을 보여준다.
- 4) userid와 password가 불일치하면 에러화면을 보여준다.

참고

메뉴화면: 메일읽기, 메일쓰기, 주소록관리 메뉴 (새사용자 등록 메뉴도 있으면 좋을 것 같음.)

에러화면: 로그인화면으로 돌아갈 수 있도록 URL을 연결시켜 주도록 한다.

## 1.2 메일 읽기

\* 액터: 사용자, POP3서버

\* 목적: 수신 메일 읽기

\* 개요: 사용자가 메뉴화면에서 메일읽기를 선택하면 수신된 메일 목록을 보여준다. 사용자가 원하는 메일을 선택하면 해당 메일을 POP3서버로부터 읽어와서 보여준다.

\* 유형: 기본

\* 이벤트 순서

- 1) 로그인시 입력한 userid와 password를 이용하여 POP3 서버에서 메일 목록중 최근 k개만 가져와서 화면에 보여준다. (default:k = 10) 메일 목록 아래에는 메일 페이지 목록을 보여준다.
- 2) 사용자가 현재 화면에 보여진 메일 목록중에서 특정 메일을 선택하면 POP3 서버에 접속하여 해당 메일을 가져와서 보여준다.
- 3) 사용자가 페이지 목록중 특정 페이지를 선택하면 해당 페이지에 해당하는 메일 목록을 k개 가져와서 화면에 보여준다.

참고.

- 메일 목록: (메일수신번호, 송신주소, 제목, 수신시간)
- 페이지 목록: 1 ~ ceil(n/k)

### 1.3 메일 쓰기

\* 액터: 사용자, SMTP 서버

\* 목적: 작성한 메일 보내기

\* 개요: 사용자가 메뉴화면에서 메일쓰기를 선택하면 메일수신자, 제목, 본문 내용과 관련된 화면을 보여주고, 모두 작성한 후 메일을 송신한다.

\* 유형: 기본

\* 이벤트 순서

1) 사용자가 메일수신자, 제목, 본문 내용을 입력한다.

2) 송신 버튼을 누르면 SMTP 서버에 접속하여 메일을 송신한다.

\* 대체 이벤트

2) 메일 송신 실패시 메일 송신 실패 화면을 보여준다.

### 1.4 주소록

\* 액터: 사용자, DB서버

\* 목적: 사용자 주소록 관리

\* 개요: 사용자의 주소록 등록과 현재까지의 주소록 목록을 볼 수 있다.

\* 유형: 기본

\* 이벤트 순서

1) 사용자가 메뉴화면에서 주소록을 선택하면 사용자 등록 메뉴와 함께 현재까지 입력된 사용자 주소록을 보여준다.

2) 주소록 등록을 원하면 사용자 이름, 이메일주소, 전화번호 등의 관련정보를 입력한 후, 등록 버튼을 누른다.

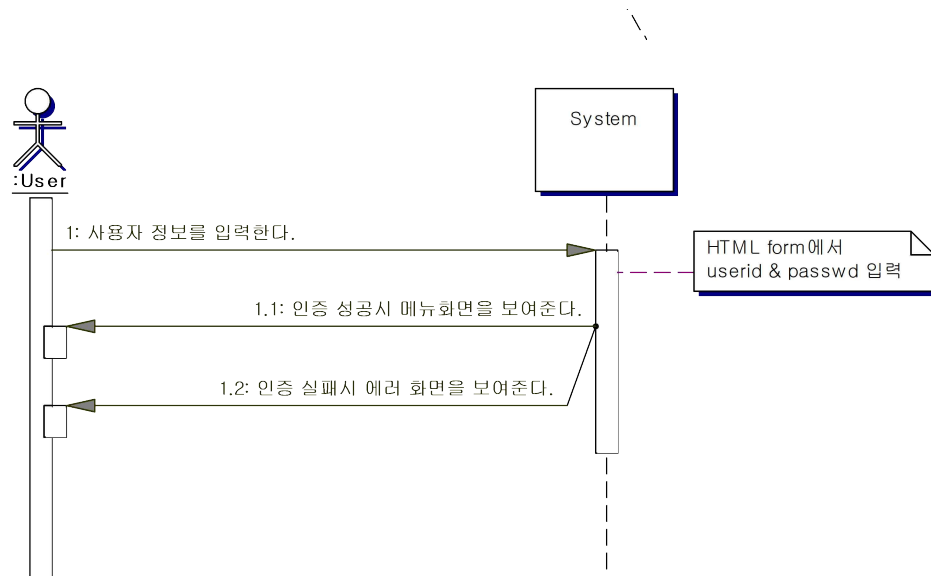
3) 등록 성공시 화면이 갱신되면서 주소록에 방금 입력한 주소록 정보가 보인다.

\* 대체 이벤트

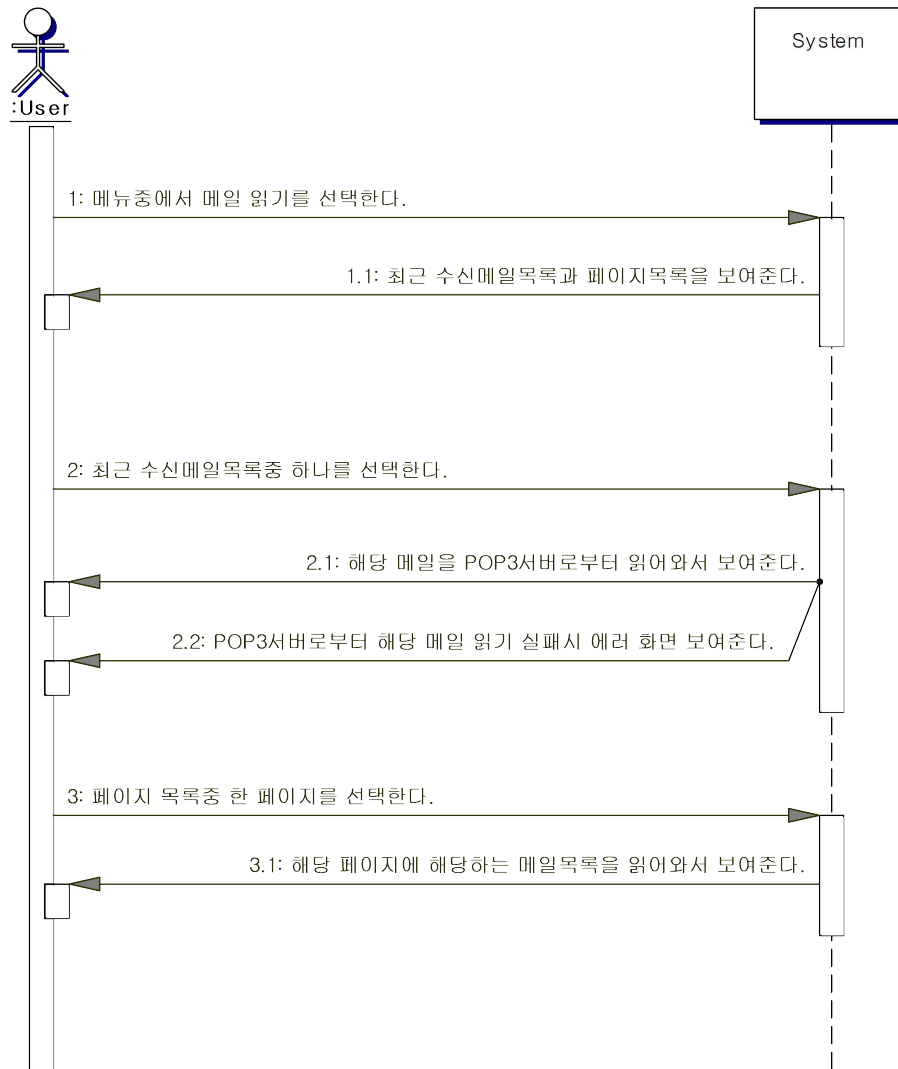
3) 등록 실패시 실패 이유를 설명하는 에러 화면을 보여준다.

## 2. 시스템 순차도

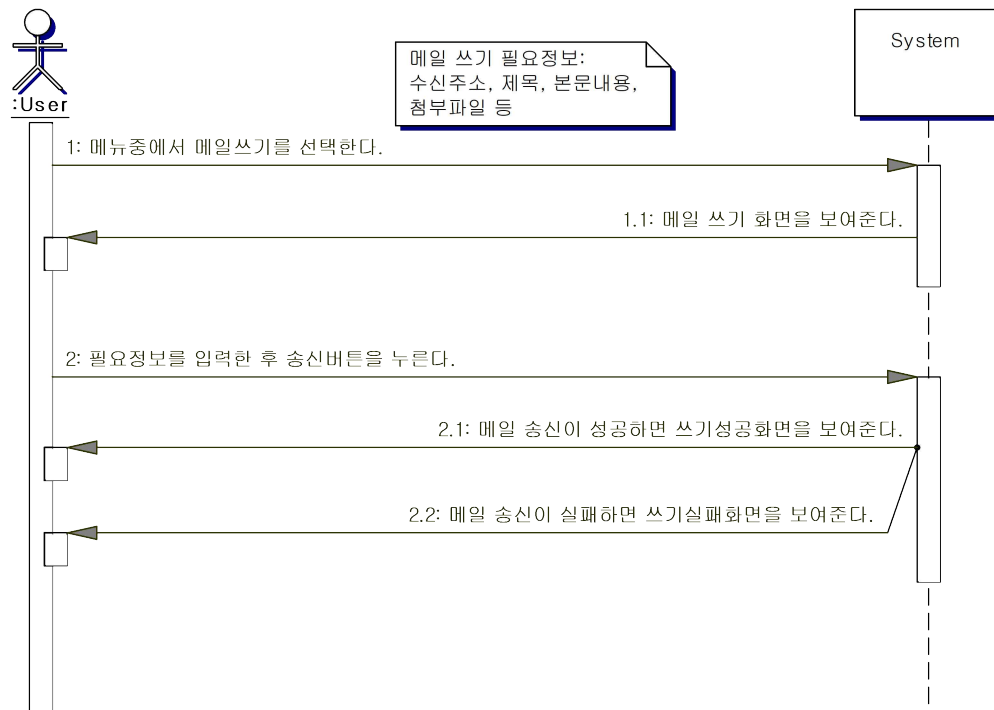
### 2.1 로그인



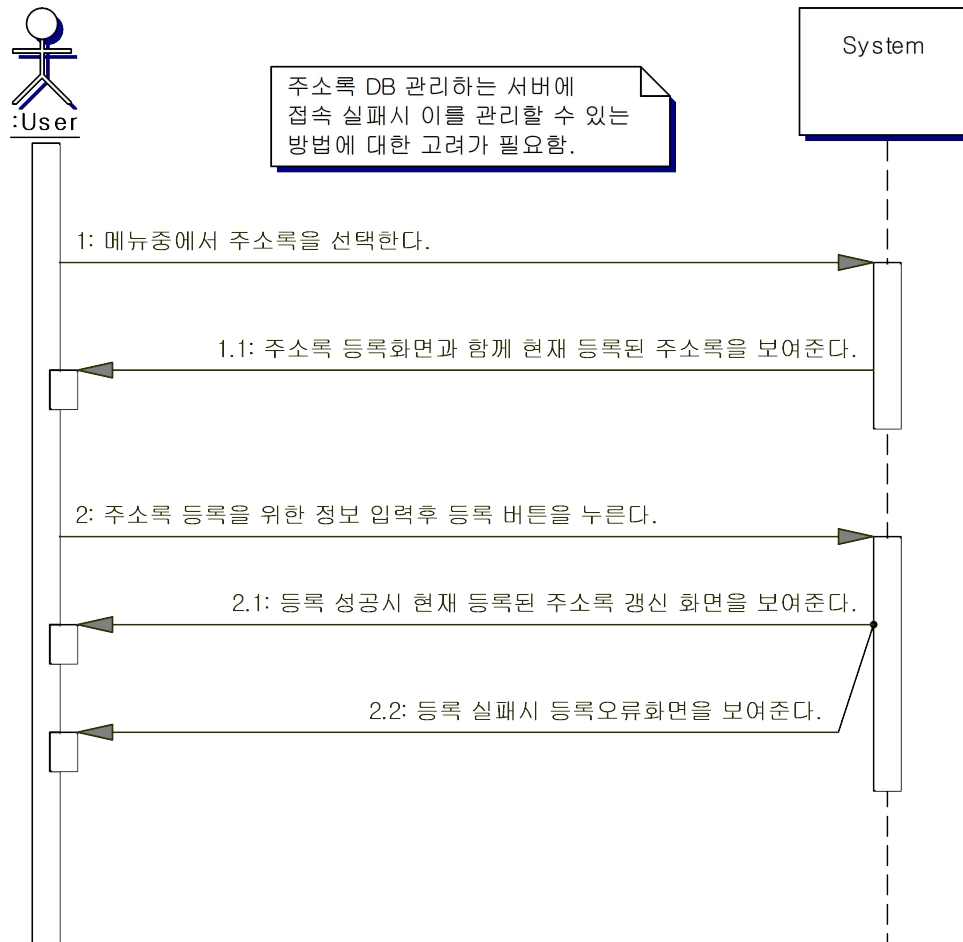
## 2.2 메일 읽기



## 2.3 메일 쓰기



## 2.4 주소록



### 3. 환경 설정

#### 3.1 라이브러리

<Apache Commons fileUpload>

- 파일 업로드 라이브러리
- 홈페이지: <http://commons.apache.org/fileupload/>
- 주의사항: 아파치 Commons IO 필요함

<Apache Commons IO>

- IO 기능 개발에 사용되는 유틸리티
- 홈페이지: <http://commons.apache.org/io/>

<JavaMail API 1.4.1>

- 자바 메일 관련 API
- 홈페이지: <http://java.sun.com/products/javamail/index.jsp>

#### 3.2 메일 서버 설정

<index.jsp>

<%

if (session.isNew()) {

    session.setAttribute("host", "[seps2.deu.ac.kr](http://seps2.deu.ac.kr)"); // should be modified if you change the POP3 server

    session.setAttribute("debug", "false");

    session.setAttribute("pageno", "1");

    session.setMaxInactiveInterval( session.getMaxInactiveInterval()\*2 );

}

%>

### 3.2 다운로드 경로 설정

<MailHandleBean.java>

```
public class MailHandleBean {  
  
    // LJM : 다운로드할 파일의 임시 저장 디렉토리  
    // 파일 다운로드후 파일은 DownloadServlet에서 삭제됨.  
    // 동시에 여러 사용자가 동일한 파일명을 가지는 파일 다운로드시 문제 발생되므로  
    // 각 사용자별로 이 디렉토리 밑에 새로 디렉토리를 생성해서 관리할 필요있음.  
    final String downloadTempDir = "/var/spool/webmail/download/";  
    //final String downloadTempDir = "F:/temp/download/";  
}
```

### 3.3 업로드 경로 설정

```
public class smtpServlet extends HttpServlet {  
  
    // uploadTempDir과 uploadTargetDir를 동일 디렉토리로 하도록 권장함.  
    final private String uploadTempDir = "C:/temp/upload/";  
    final private String uploadTargetDir = "C:/temp/upload/";  
  
    //final private String uploadTempDir = "/var/spool/webmail/";  
    //final private String uploadTargetDir = "/var/spool/webmail/";  
}
```