**응용 아키텍처 설계서**

버전: V3

작성일: 2019-11-19

작성자: 장성원, 정수환, 김가영, 이소연, 김건호

목차

[1. 개요 4](#_Toc25086592)

[2. 논리적 아키텍처 명세 5](#_Toc25086593)

[2.1. 논리적 컴포넌트 모델 5](#_Toc25086594)

[2.1.1. 컴포넌트 다이어그램 5](#_Toc25086595)

[2.1.2. 컴포넌트 목록 6](#_Toc25086596)

[2.2. 인터페이스 상세 명세 7](#_Toc25086597)

[2.2.1. UserMgt 7](#_Toc25086598)

[2.2.2. MessSI 7](#_Toc25086599)

[2.2.3. MainBoardMgt 8](#_Toc25086600)

[2.2.4. MinPostitSI 8](#_Toc25086601)

[2.2.5. CommandMgt 8](#_Toc25086602)

[2.3. 설계 클래스 명세 9](#_Toc25086603)

[2.3.1. UserMgtBean 9](#_Toc25086604)

[2.3.2. MessageBean 11](#_Toc25086605)

[2.3.3. MainBoardBean 13](#_Toc25086606)

[2.3.1. MindPostitBean 15](#_Toc25086607)

[2.3.2. CommandBean 17](#_Toc25086608)

[3. 유스케이스 실현 명세 18](#_Toc25086609)

[3.1.1. 사용자 로그인 관리 18](#_Toc25086610)

[3.1.2. 랜덤 쪽지 19](#_Toc25086611)

[3.1.3. 익명 게시판 20](#_Toc25086612)

[3.1.4. 사용자 정보 관리 21](#_Toc25086613)

[3.1.5. 마인드 포스트 잇 22](#_Toc25086614)

[3.1.6. 댓글 관리 22](#_Toc25086615)

[4. 물리적 아키텍처 명세 24](#_Toc25086616)

[4.1. 물리적 컴포넌트 명세 24](#_Toc25086617)

[4.2. 물리적 컴포넌트 의존 명세 27](#_Toc25086618)

[4.3. 물리적 컴포넌트 배치 명세 28](#_Toc25086619)

# 개요

*이 문서는 해당 소프트웨어 아키텍처의 범위를 구체화하기 위한 문서이다. 이전 문서에서 작성된 내용을 참조하여 소프트웨어 아키텍처에서 제공해야 하는 설계 및 개발 지침을 정의하기 위해 고려되어야 하는 업무 범위와 솔루션 범위를 파악하기 위해 작성한다.*

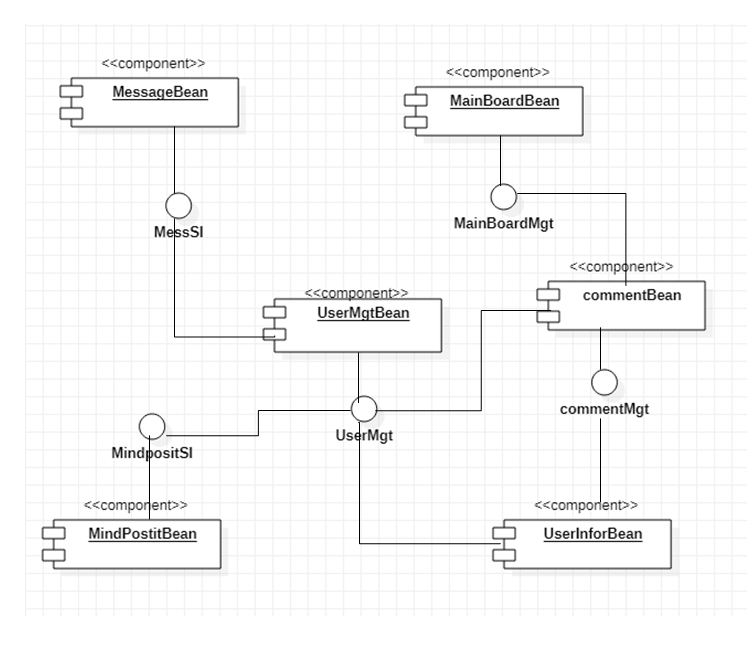
*또한 기술들에 대한 구조를 설계하고 개발하기 위해 필요한 지침사항들을 작성한다. 이는 연계 유형을 개발하는 경우 그 규모와 영향을 고려하여 구현 단계에서 수행을 원활하게 할 수 있도록 도와준다.*

*기능적 비기능적 요구사항에 맞추어 컴포넌트를 도출하고, 컴포넌트들 간의 관계 및 컴포넌트의 내부 구조에 대한 정의를 한다. 기능적 요구사항에서는 시스템의 기능 구조적인 관점으로 아키텍처를 보는 것으로 각각의 기능 단위를 컴포넌트로 식별하고 이들 컴포넌트간의 관계를 표현하는 컴포넌트 다이어그램 형태로 표현했다. 또한 소프트웨어적인 아키텍처는 소프트웨어를 구성하는 클래스 요소와 이들간의 관계를 규정한다.*

# 논리적 아키텍처 명세

## 논리적 컴포넌트 모델

### 컴포넌트 다이어그램



### 컴포넌트 목록

|  |  |  |
| --- | --- | --- |
| 컴포넌트 | 명세 | |
| UserMgtBean | 설명 | 사용자의 로그인/로그아웃, 사용자 관리 기능을 제공. |
| 제공인터페이스 | UserMgt |
| 요구인터페이스 | 없음 |
| MessageBean | 설명 | 랜덤으로 한 사용자에게 쪽지를 전송하고 그에 따른 답장을 할 수 있는 기능을 제공. |
| 제공인터페이스 | MessSI |
| 요구인터페이스 | UserMgt |
| MainBoardBean | 설명 | 자신의 고민글등을 올리고 수정, 삭제할 수 있는 기능을 제공. |
| 제공인터페이스 | MainBoardMgt |
| 요구인터페이스 | UserMgt |
| UserInfoBean | 설명 | 사용자들이 회원가입시 내용을 저장하고, 오플리케이션에서 하는 모든 내용을 저장. |
| 제공인터페이스 | UserMgt |
| 요구인터페이스 | 없음 |
| MindPostitBean | 설명 | 배너의 주제에 대한 의견을 포스트잇 형태로 올리고 내릴 수 있는 기능을 제공. |
| 제공인터페이스 | MinPostitSI |
| 요구인터페이스 | userMgt |
| CommandBean | 설명 | 사용자들이 포스트 한 내용에 대한 댓글을 달고 추천을 눌러 인기글을 선정하는 기능을 제공. |
| 제공인터페이스 | CommandMgt |
| 요구인터페이스 | UserMgt |

## 인터페이스 상세 명세

### UserMgt

|  |  |
| --- | --- |
| 연산 | 설명 |
| void registerUser(UserInfo info)  throws DuplicatedUserIDException | 회원가입 시 실행되는 메소드이며 회원가입 시 입력한 정보를 저장하고 중복 아이디에 대한 예외 발생. |
| Void storeAddress(UserInfo info) | 회원가입시 입력한 정보를 데이터 베이스에 저장. |

### MessSI

|  |  |
| --- | --- |
| 연산 | 설명 |
| Struct Message send(struct massege m) | 사용자가 작성한 메시지를 message구조체 형식으로 전달. |
| Struct Message receive(struct Message r) | 전달된 메시지를 받아서 사용자 중 랜덤으로 한명에게 메시지 전달. |

### MainBoardMgt

|  |  |
| --- | --- |
| 연산 | 설명 |
| Void postBoard(struct Post p) | 사용자가 작성한 게시물을 mainboard에 게시. |
| Void deleteBoard(struct Post p) | 사용자가 삭제를 실행 시 게시물을 mainboard에서 삭제 |

### MinPostitSI

|  |  |
| --- | --- |
| 연산 | 설명 |
| Struct Potstit post(sturct Postit pi) | 사용자가 작성한 글을 포스트잇 형태로 게시. |
| Void banner(String subject) | 화면 상단에 일정시간마다 주제를 게시 |
| Voide deleteAll() | 배너의 주제가 바뀔 때 모든 포스트들을 삭제. |
| Void deletePostit(struct Postit pi) | 사용자가 삭제 버튼을 클릭시 본인이 작성한 포스트들을 삭제. |

### CommandMgt

|  |  |
| --- | --- |
| 연산 | 설명 |
| struct Message write(String cmd) | 작성된 게시글들에 댓글을 작성. |
| Void recommend() | 게시글의 추천버튼을 눌러 추천수로 인기글을 선정. |

## 설계 클래스 명세

|  |
| --- |
| * *논리적 컴포넌트 이외에 도출된 클래스를 명세 한다.* * *논리적 컴포넌트의 인터페이스 연산에서 사용된 인자 타입과 반환 값 타입, 예외 클래스 등을 기술한다.* * *JPAEntity 클래스 등과 같이 논리적 컴포넌트들이 공유하는 클래스를 기술한다.* * *구체적인 작성 방법은 분석명세서의 2장. 시스템 구성을 참고한다.* |

### UserMgtBean

#### 설계 클래스 개요

|  |  |
| --- | --- |
| 클래스 명 | 설명 |
| Private info | 개인 정보를 관리 |
| log | 회원가입 로그인을 할 수 있고 앱을 나갈 수 있음. |
| Message mange | 그동안 활동했던 로그들을 기록 및 관리 |

#### 설계 클래스 명세: Private info

서비스를 이용하기 전 서비스의 필요한 간단한 정보를 입력하여 이용 권한을 가진다.

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| Int sex | 남자인지 여자인지 구별 |
| String birthdate | 생일 |
| String ID | 회원 이름 |
| String MyBoard(struct Board \*b) | 활동했던 게시물 기록 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Private info() | 서비스 이용 권한 절차를 거침 |
| logout() | 로그아웃을 할 수 있음 |
| Message mange () | 개인 쪽지를 관리할 수 있음. |

#### 설계 클래스 명세: log

회원가입 로그인을 할 수 있고 앱을 나갈 수 있음.

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| Int login | Login을 누르면 true 값 반환 |
| Int sex | 남자인지 여자인지 구별 |
| String birthdate | 간단한 개인정보입력 |
| String ID | 회원 이름 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| login() | 서비스 이용 권한 절차를 거침 |
| sign\_up() | 회원 가입을 할 수 있음. |

#### 분석 클래스 명세: Message manage

쪽지 관리를 할 수 있는 클래스

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String message | 쪽지 내용 |
| String date | 쪽지 가 주고 간 날짜 |
| Int delete | 쪽지 삭제 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| message(int sender, string mes) | 그동안의 주고 받은 쪽지를 볼 수 있음 |
| send() | 쪽지 보내기 |

### MessageBean

#### 설계 클래스 개요

|  |  |
| --- | --- |
| 클래스 명 | 설명 |
| Send\_M | 쪽지를 보낼 수 있는 클래스 |
| Show\_M | 그동안의 쪽지 활동들을 볼 수 있는 클래스 |

#### 설계 클래스 명세: Send\_M

랜덤 쪽지를 보내고 싶을 때 보낼 수 있는 클래스

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String message | 메시지 내용 |
| Boolean send | Send를 누르면 true로 메시지가 보내진다. |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Send() | 상대방에게 쪽지를 보낼 때 쓰는 연산 |
| Write() | 상대방에게 보낼 쪽지를 쓰기위한 연산 |

#### 설계 클래스 명세: Show\_M

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String message | 쪽지 내용 |
| String date | 쪽지 가 주고 간 날짜 |
| Boolean delete | 쪽지 삭제 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Show() | 작성한 쪽지를 열람 |
| Delete() | 작성한 쪽지를 삭제 |

### MainBoardBean

#### 설계 클래스 개요

|  |  |
| --- | --- |
| 클래스 명 | 설명 |
| uplaod | 게시물을 쓰기 한 클래스 |
| show | 사람들이 업로드한 게시물을 볼 수 있는 클래스 |
| modified | 자신이 올린 게시물을 수정할 수 있는 클래스 |
| delete | 자신의 게시물을 삭제할 수 있는 클래스 |

#### 설계 클래스 명세: upload

게시물을 쓰기 한 클래스

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String worry | 고민 내용 |
| Int ID | 고민자의 ID |
| String date | 고민 게시물을 올린 날짜 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Upload() | 고민글을 완성하고 고민을 업로드 하기위한 메소드 |
| W\_write() | 고민을 쓰기위한 연산 |

#### 설계 클래스 명세: show

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| Struct other\_worry(struct string Id, string m) | 상대방의 고민 게시물 내용 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Like() | 좋아요 버튼을 누름 |
| Detail() | 한 고민자의 내용을 자세히 봄 |
| Comment() | 고민자 게시물에 댓글을 달 수 있음 |

#### 설계 클래스 명세: modified

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String worry | 고민 수정된 내용 |
| Int ID | 고민자의 ID |
| String date | 고민 게시물을 수정 날짜 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Modified() | 수정을 누르면 수정 가능 |

#### 설계 클래스 명세: delete

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| Boolean delete | 수정 확인을 받기위함 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| delete () | 삭제를 원하면 삭제를 가능하게 해주는 메소드 |

### MindPostitBean

#### 설계 클래스 개요

|  |  |
| --- | --- |
| 클래스 명 | 설명 |
| Show\_postit | 마인드 포스트의 질문과 포스트를 볼 수 있음 |
| answer | 질문의 포스트 잇을 할 수 있는 클래스 |

#### 설계 클래스 명세: Show\_postit

쪽지을 쓰기 한 클래스

#### 속성 개요

|  |  |
| --- | --- |
| 속성 | 설명 |
| String question | 마인드 포스트의 질문 |
| Struct post(struct string id, string c) | 고민자들의 답변 |
| Boolean upload | 포스트 잇을 하기위한 버튼 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Show\_post() | 붙은 포스트를 볼 수 있는 연산 |

#### 설계 클래스 명세: answer

게시물을 쓰기 한 클래스

#### 속성 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Int ID() | 고민자의 ID |
| String answer | 사용자의 포스트 내용 |

#### 연산 개요

|  |  |
| --- | --- |
| 연산 | 설명 |
| Post\_it() | 포스트 잇을 할 수 있는 메소드 |

### CommandBean

#### 설계 클래스 개요

|  |  |
| --- | --- |
| 클래스 명 | 설명 |
| Command | 해당 게시글에 답글을 달 수 있음 |

#### 설계 클래스 명세: Show\_postit

쪽지을 쓰기 한 클래스

#### 속성 개요

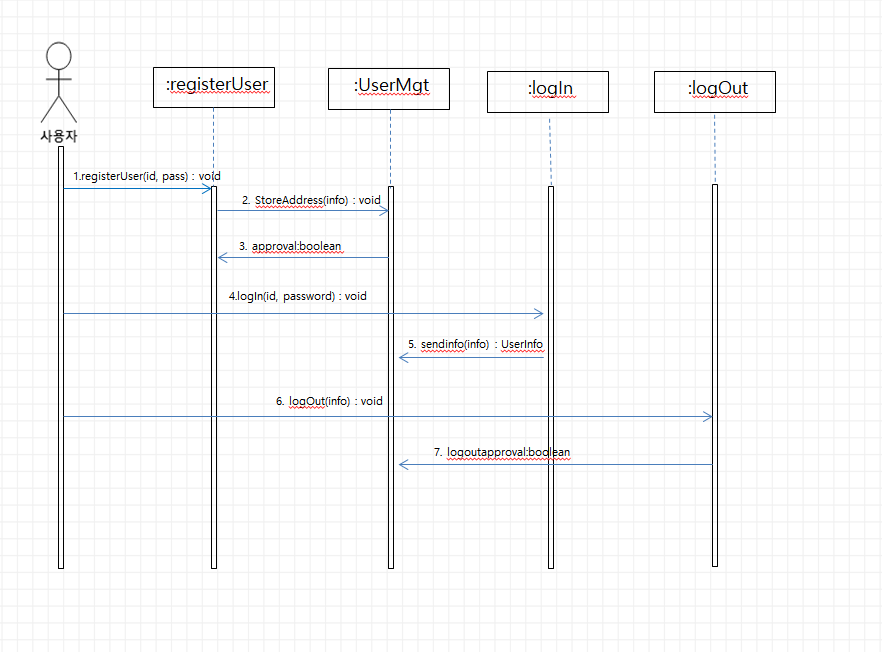
|  |  |
| --- | --- |
| 속성 | 설명 |
| String command | 댓글 내용 |
| Int ID | 댓글을 단 사용자의 ID |
| Boolean up\_c | 댓글달기 최종 버튼 |

#### 연산 개요

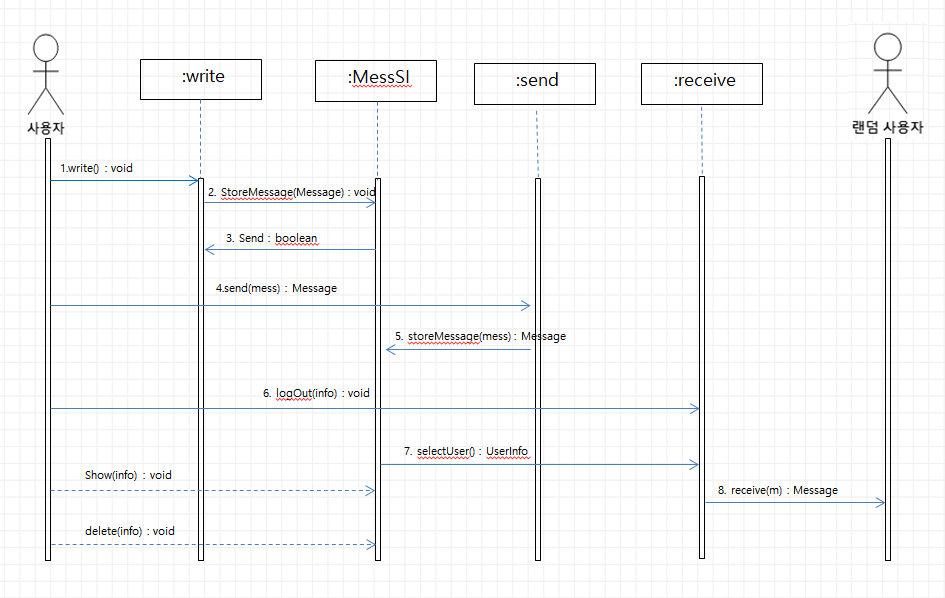
|  |  |
| --- | --- |
| 연산 | 설명 |
| Command() | 댓글을 달수있는 메소드 |

# 유스케이스 실현 명세

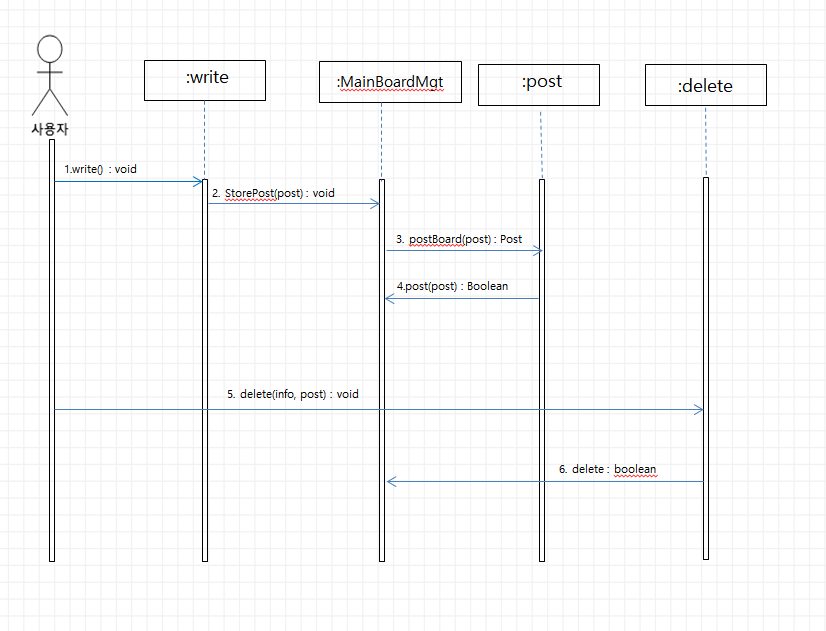
### 사용자 로그인 관리



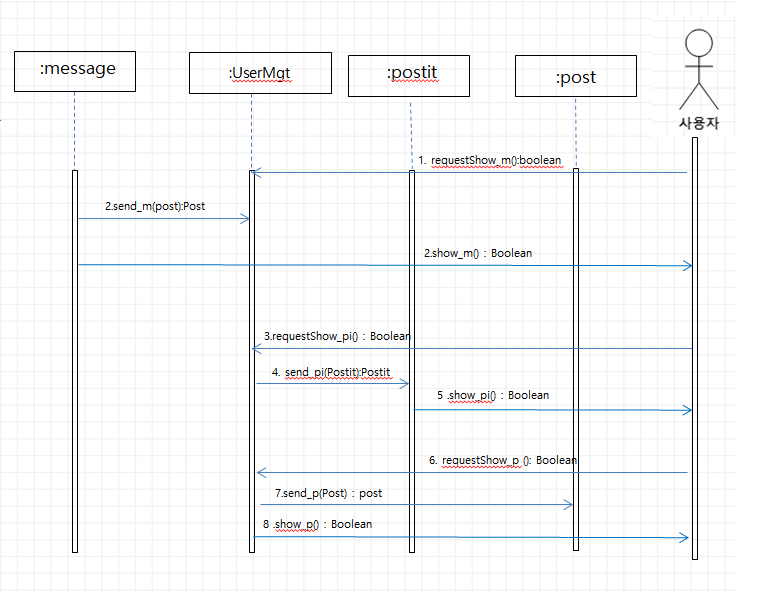
### 랜덤 쪽지



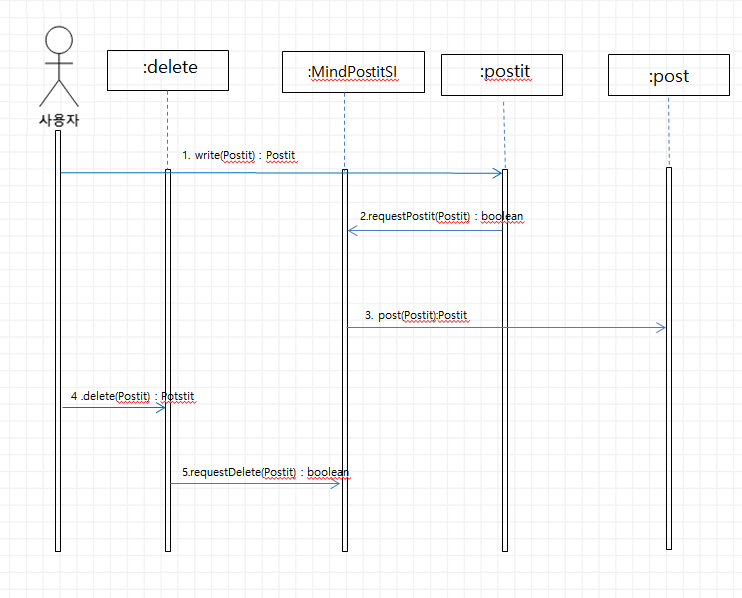
### 익명 게시판



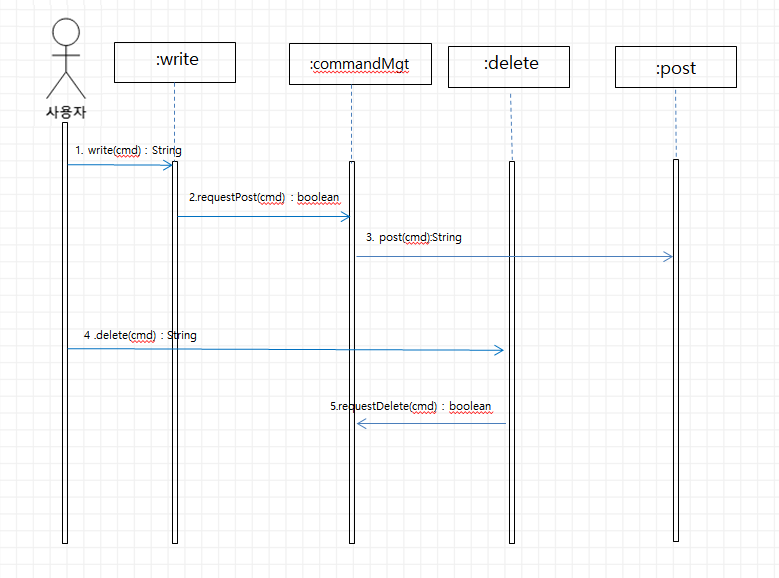
### 사용자 정보 관리



### 마인드 포스트 잇



### 댓글 관리

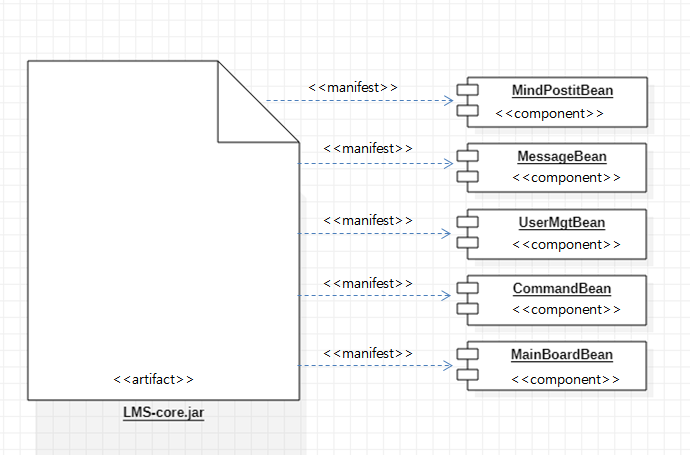


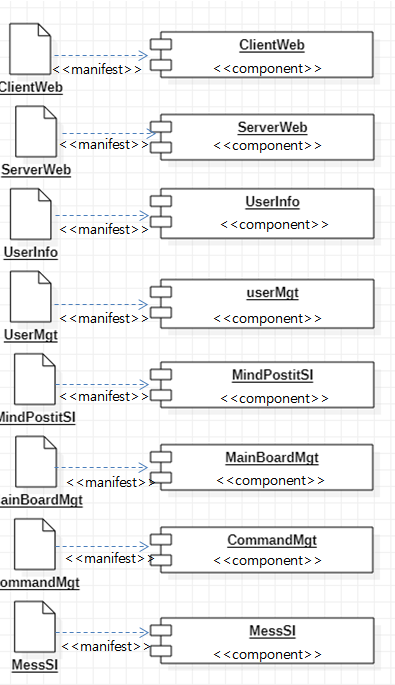
# 물리적 아키텍처 명세

## 물리적 컴포넌트 명세

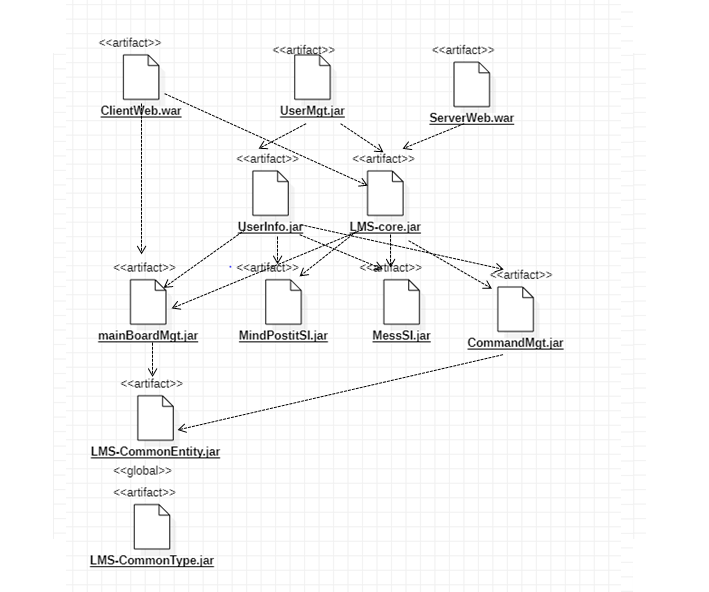
|  |  |  |
| --- | --- | --- |
| 물리적 컴포넌트 | 포함된 논리적 컴포넌트 | 핵심 기능 |
| LMS-CommonType.jar | common:type 패키지 | 공통 데이터 클래스 |
| common::exception 패키지 | 예외 클래스 |
| LMS-CommonEntity.jar | common:entity 패키지 | <<JPAEntity>> 클래스 |
| ClientWeb.war | ClientWeb | 사용자용  웹 페이지 |
| ServerWeb.war | ServerWeb | 관리자 용  웹 페이지 |
| LMS-core.jar | MessageBean | 랜덤 쪽지 기능 |
| UserMgtBean | 로그인관리  개인정보관리 |
| MindPostitBean | 마인드 포스트 잇  기능 |
| CommandBean | 댓글 작성  인기도 측정 |
| MainBoardBean | 익명 게시판  작성 관리 기능 |
| UserInfo.jar | UserInfoBean | 사용자의 행동 관리 |
| UserMgt.jar | UserMgt | 사용자 관리  인터페이스 |
| MindPostitSI.jar | MindPostitSI | 마인드 포스트 잇 관리 인터페이스 |
| MessSI.jar | MessSI | 랜덤 쪽지 관리  인터페이스 |
| MainBoardMgt.jar | MainBoardMgt | 익명 게시판 관리 인터페이스 |
| CommandMgt.jar | CommandMgt | 댓글 관리  인터페이스 |

|  |
| --- |
| *또는 아래와 같이 배치 다이어그램으로 표현할 수도 있다.* |



ㄴ

## 물리적 컴포넌트 의존 명세



## 물리적 컴포넌트 배치 명세

|  |  |  |
| --- | --- | --- |
| 물리적 컴포넌트 | 노드 | 수행환경 |
| LMS-commonType.jar | 프로그램관리서버 | JRE 1.6 |
| ClientWeb.war | 프로그램관리서버 | GlassFish V2::웹컨테이너 |
| ServerWeb.war | 프로그램관리서버 |
| UserMgt.jar | 프로그램관리서버 |
| mainBoardMgt.jar | 프로그램관리서버 |
| CommandMgt.jar | 프로그램관리서버 |
| MindPostitSI.jar | 프로그램관리서버 | GlassFish V2::EJB컨테이너 |
| MessSI.jar | 프로그램관리서버 |
| UserInfo.jar | 프로그램관리서버 |
| LMS-core.jar | 프로그램관리서버 |

