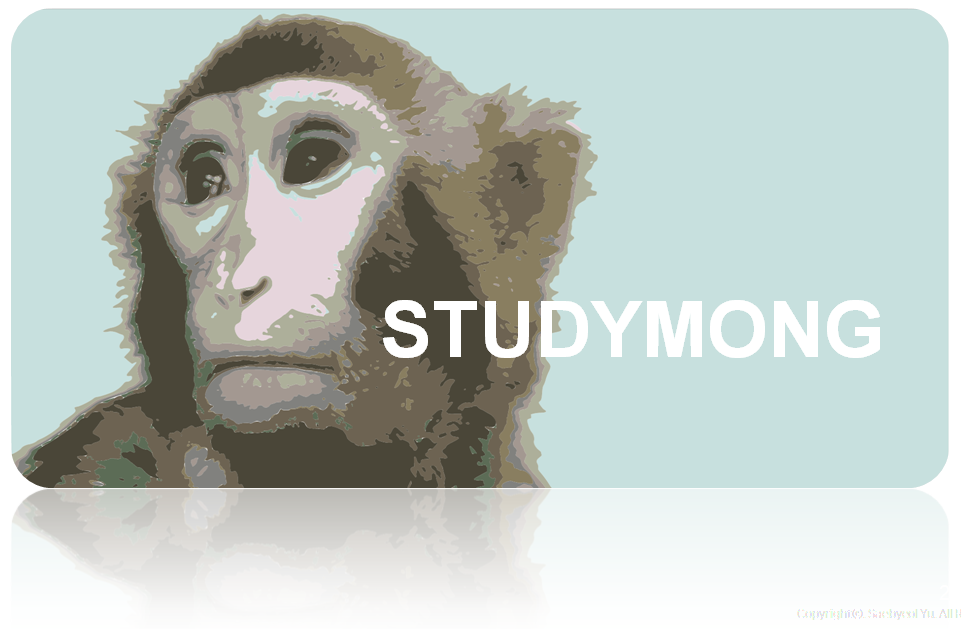
스터디 프로젝트 PortFolio

studyMong

2019.07.08.~2019.08.03.

팀장 : 손상준

팀원 : 강덕현,안성도,김민호,이태관

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | **목 차** |  |  |  |  |

Ⅰ. 프로젝트 소개 3

1. 프로젝트 주제 3

2. 개발환경 5

3. 역할분담 6

Ⅱ. 설계 준비 8

1. 요구사항 명세 8

2. 업무 일정표 12

3. USE - CASE 13

Ⅲ. 설계 14

1. 데이터베이스 설계 14

2. 클래스 설계 15

3. UI 설계 19

3. URL 설계 31

Ⅳ. 구현 33

1. UI 및 기능 구현 33

2. QA테스트 44

Ⅴ. 마치고 46

1. 프로젝트를 마치고 46

**Ⅰ. 프로젝트 소개**

**1-1 프로젝트 주제**

▶문제점 발견

1. 지역별 다양한 카테고리에 맞게 간편하게 스터디를 모집할 수 있는 사이트의 부재
2. 일정관리,출석관리,멤버관리를 편하게 할 수 있는 사이트의 필요성
3. 게시판개설 등 커뮤니티성 스터디 모집 사이트의 부재
   1. **프로젝트 주제**

▶프로젝트 소개



- 프로젝트 명 : Study Mong

- 프로젝트 설명 : 스터디 개설, 카테고리 지역별 검색, 커뮤니티기능, 스터디 관리기능을 가진 사이트 를 개발하고자 하였다. Java와 JSP,Spring Framework를 활용하여 프로젝트를 진행하였으며, 다섯명의 팀원이 Git 을 기반으로한 협업으로 개인 개발능력 향상 및 협업능력 향상을 목표로 한다.

★ 구현하고자 하는 핵심 기능

1. 스터디 즐겨찾기 및 즐겨찾기에 추가된 스터디의 게시글 메인 페이지 포스팅 기능

2. 스터디 개설 및 검색 기능

3. 게시판 글쓰기 및 댓글기능

4. 스터디 일정기능

5. 스터디 관리(멤버관리,출석관리,가입승인관리)기능

**1-2개발환경**



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **이름** | **java** | **servlet** | **Apache Tomcat** | **Oracle DB** | **Maven** |
| **버전** | 1.8.0v | 3.1v | 8.5v | 12.2.0.1v | 3.1v |
| **이름** | **Spring** | **eclipse** | **Bootstrap** | **Ajax** |  |
| **버전** | 4.1.0v | 4.11.0v | 4.1.4v |  |  |

**1-3역할분담**

**- 구현 기능 :** 메인화면, 마이스터디, 즐겨찾기

스터디소식, 스터디룸

Ajax처리(댓글수정 ,즐겨찾기),

Interceptor를 활용한 페이지접근권한처리

**- 기타역할 :** UI프로토타이핑, 디자인, 테스트, 문서작성,

프로젝트통합, 클래스다이어그램, ERD,

USE-CASE다이어그램 작성

- **GitubBlog :** tkdwns1245.github.io

**- 구현 기능 :** 스터디공지 게시판, 자유게시판, 검색기능

게시글 작성, 수정,삭제, 댓글 작성, 수정, 삭제,

페이징처리

**- 구현 기능 :** 스터디 리스트, 스터디 만들기

스터디 자세히보기, 스터디 신청

스터디 리스트 파일업로드

**1-3역할분담**

**- 구현 기능 :** 일정관리 모듈 구현, 멤버관리 페이지 도움,

프로젝트 통합 도움

**- 기타 구현 :** 데이터베이스 설계, 요구사항 정리,

UI프로토타이핑

**- 구현 기능 :** 멤버관리,출석관리,가입신청서 관리

**- 기타 구현 :**

Ⅱ**. 설계준비**

2-1 요구사항 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **화면명** | **요구사항명** | **요구사항내용** | **날짜** | **진행사항** |
| 메인 | 우측메뉴바 | - 우측 상단 버튼 클릭 시 메뉴가 펼쳐진다.  - 로그인,스터디리스트,내스터디 버튼 | 2019-07-08 | 구현 |
| 메인 | 스터디 리스트 | - 최근 개설된 스터디 리스트 10개 게시  - 스터디 리스트에는 스터디명,가입자수,간단  소개글,개설일,스터디 별점 게시  - 스터디 사진이 있다면 사진 게시 (없다면  Default사진) | 2019-07-08 | 구현 |
| 메인 | 내 스터디 소식 | - 가장 최근에 게시된 스터디 글 10개 게시  - 스터디 글은 일정,게시판글 으로 구성 | 2019-07-08 | 구현 |
| 메인 | 인기스터디 | - 가장 인기있는 스터디 5개 게시(가입자수,  게시글 수 로변별) | 2019-07-08 | 구현 |
| 로그인 | 로그인기능 | -페이스북,네이버 아이디 연동하여 로그인 | 2019-07-08 | 구현 |
| 내 스터디 | 내스터디 확인기능 | - 내가 가입한 스터디 및 운영 스터디 리스트  - 즐겨찾기 기능 | 2019-07-08 | 구현 |
| 운영스터디 | 내가 운영중인 스터디 | - 내가 운영중인 스터디 정보  - 출력 |  | 구현 |

2-1 요구사항 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **화면명** | **요구사항명** | **요구사항내용** | **날짜** | **진행사항** |
| 스터디리스트 | 스터디리스트 | - 리스트에 스터디이름,카테고리,지역,소개글 출력 리스트형식 메인 스터디리스트형식과 동일 | 2019-07-08 | 구현 |
| 스터디리스트 | 검색 | - 카테고리별,지역별,연령별,제목,내용별 검색기능 | 2019-07-08 | 구현 |
| 스터디상세 | 스터디상세 | - 스터디 이름,개설자,지역,연령대,인원수,순위,카테고리 정보 출력  - 내용을 출력  - 탈퇴하기,입장 버튼 가입자만 활성화  - 가입하기 버튼 구현 미가입자만 활성화 | 2019-07-08 | 구현 |
| 스터디등록 | 스터디등록 | - 스터디 정보 입력 후 스터디 등록 | 2019-07-08 | 구현 |
| 스터디가입신청 | 가입신청 | - 가입신청서를 작성 후 스터디 가입 신청기능 | 2019-07-08 | 구현 |

2-1 요구사항 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **화면명** | **요구사항명** | **요구사항내용** | **날짜** | **진행사항** |
| 스터디룸 | 왼쪽 상단 메뉴바 | - 게시판,스터디정보,스터디일정,내스터디활동,스터디관리(스터디 개설자만 활성화) 탭 | 2019-07-08 | 구현 |
| 스터디룸 | 게시판 | - 최근게시글 10개 게시 | 2019-07-08 | 구현 |
| 스터디룸 | 공지사항 | - 최근공지사항 10개 게시 | 2019-07-08 | 구현 |
| 스터디룸 | 간단일정 | - 최근 추가된 일정 10개 게시 | 2019-07-08 | 구현 |
| 스터디룸 | 베스트멤버 | - 베스트멤버 순위 10명 지정 | 2019-07-08 | 구현 |
| 멤버관리 | 멤버정보 | - 멤버등급업,멤버강퇴,가입신청자관리, | 2019-07-08 | 구현 |
| 출석관리 | 출석관리 | - 일정 리스트로 일정검색  일정 캘린더로 일정선택 | 2019-07-08 | 구현 |

2-1 요구사항 명세

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **화면명** | **요구사항명** | **요구사항내용** | **날짜** | **진행사항** |
| 스터디일정 | 스터디일정 | - 스터디 관리자가에게는 스터디 일정을 추가하는 기능 추가  - 달력 에 스터디 일정을 표시 | 2019-07-08 | 구현 |
| 공지사항 게시판 | 공지사항게시판 | - 글쓰기 버튼 스터디 개설자만 활성화  - 댓글기능  - 검색기능  - 페이징처리 | 2019-07-08 | 구현 |
| 자유게시판 | 자유게시판 | -자유게시판 개설  -댓글기능  -검색기능  -페이징처리  -파일첨부 | 2019-07-08 | 구현 |

2-2 업무 일정표

▶ 업무 설명 :

- 업무 일정은 위 절차를 기본으로 하되 필요에 따라 유동적으로 한다.

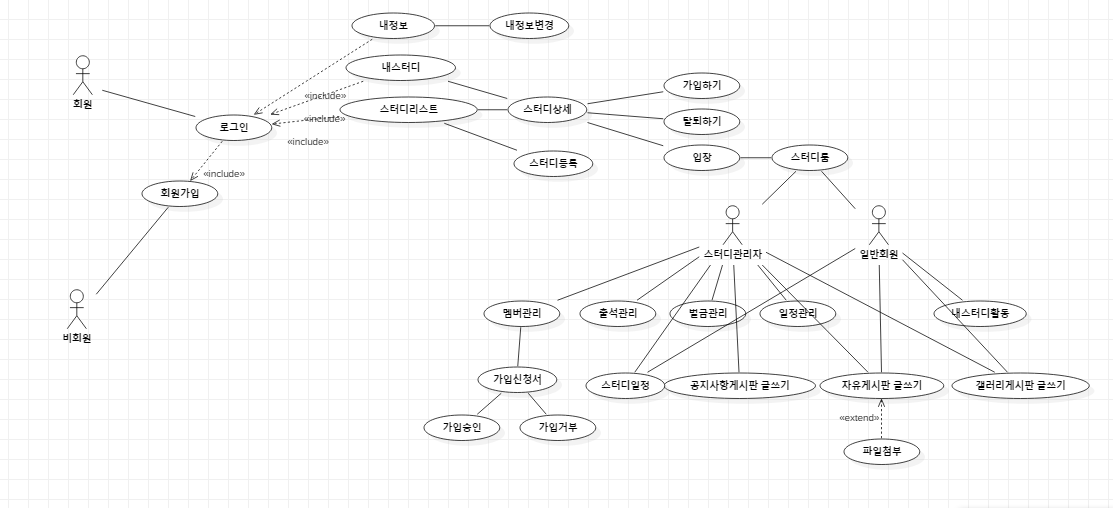
- 각 팀원마다 업무 절차 진행 속도가 다를 것으로 예상 되 먼저 이전 절차가 끝날 경

우 다음 절차 업무를 진행한다.

- 핵심 테스트는 구현 기간에 마무리하고 마지막 테스트 기간에 통합 테스트를 진행한다. 통합 과정에서 다른 팀원의 코드도 모두 이해하는 것을 목표로 한다.

- 프로젝트 통합은 기능작업 인원 당사자와 팀장이 함께 진행한다.

2-3 USE-CASE 다이어그램

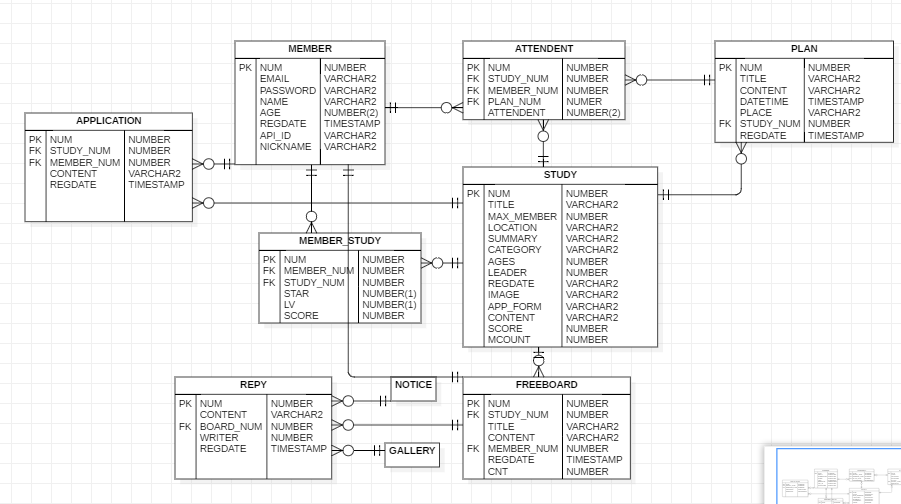


**Ⅲ. 설계**

3-1 데이터베이스 설계

▶ ERD

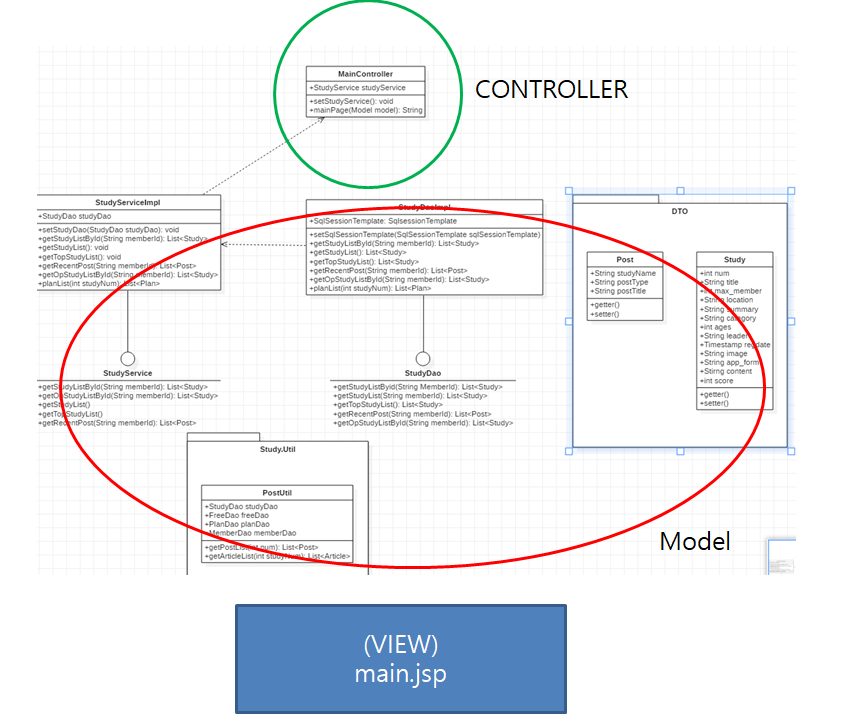
- 담당자 : 손상준



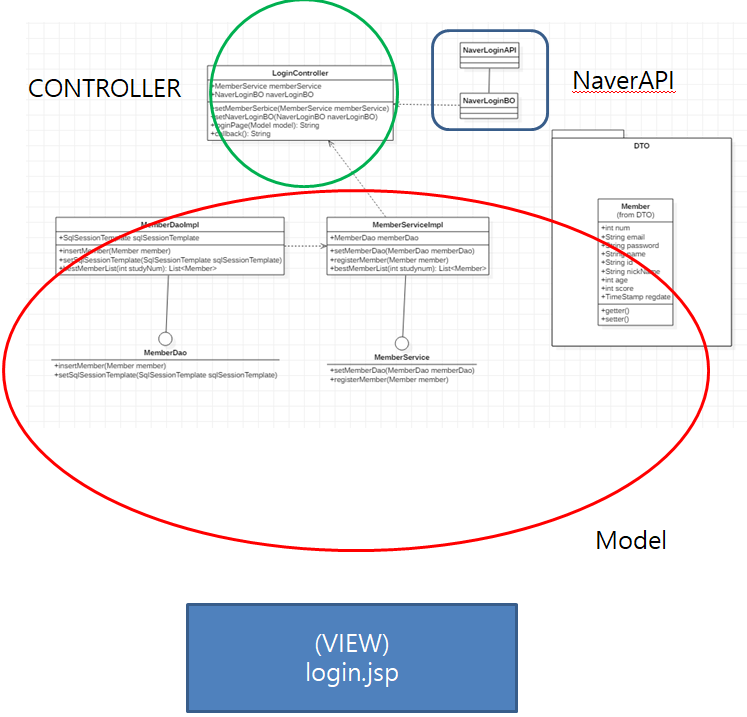
※ 자세한 DBquery  
 <https://github.com/tkdwns1245/studyTeamProject>

3-2 클래스 설계

▶ 메인페이지

- 담당자 : 손상준

3-2 클래스 설계

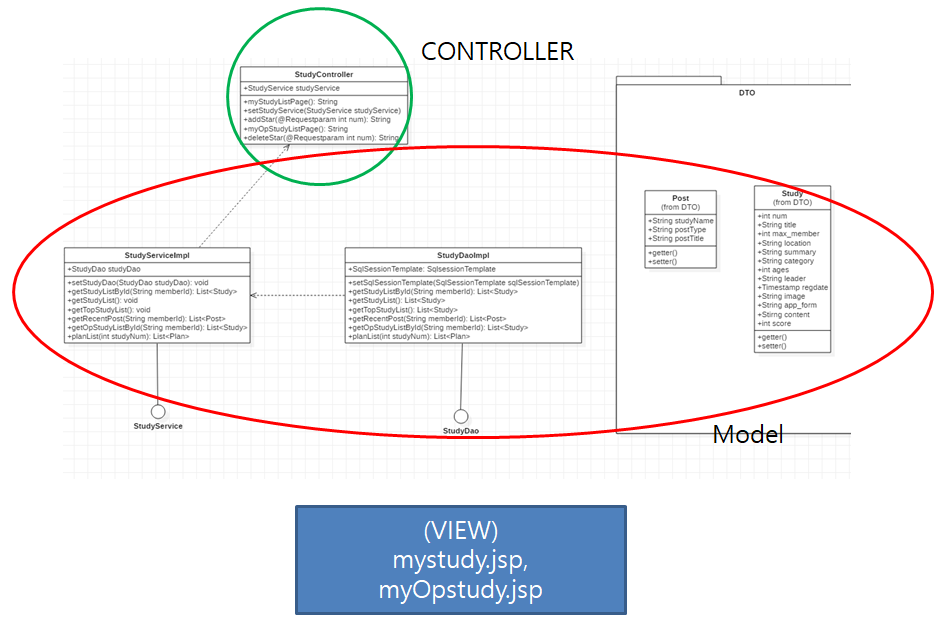
▶ 로그인화면

- 담당자 : 손상준

3-2 클래스 설계

▶ My스터디, My운영스터디

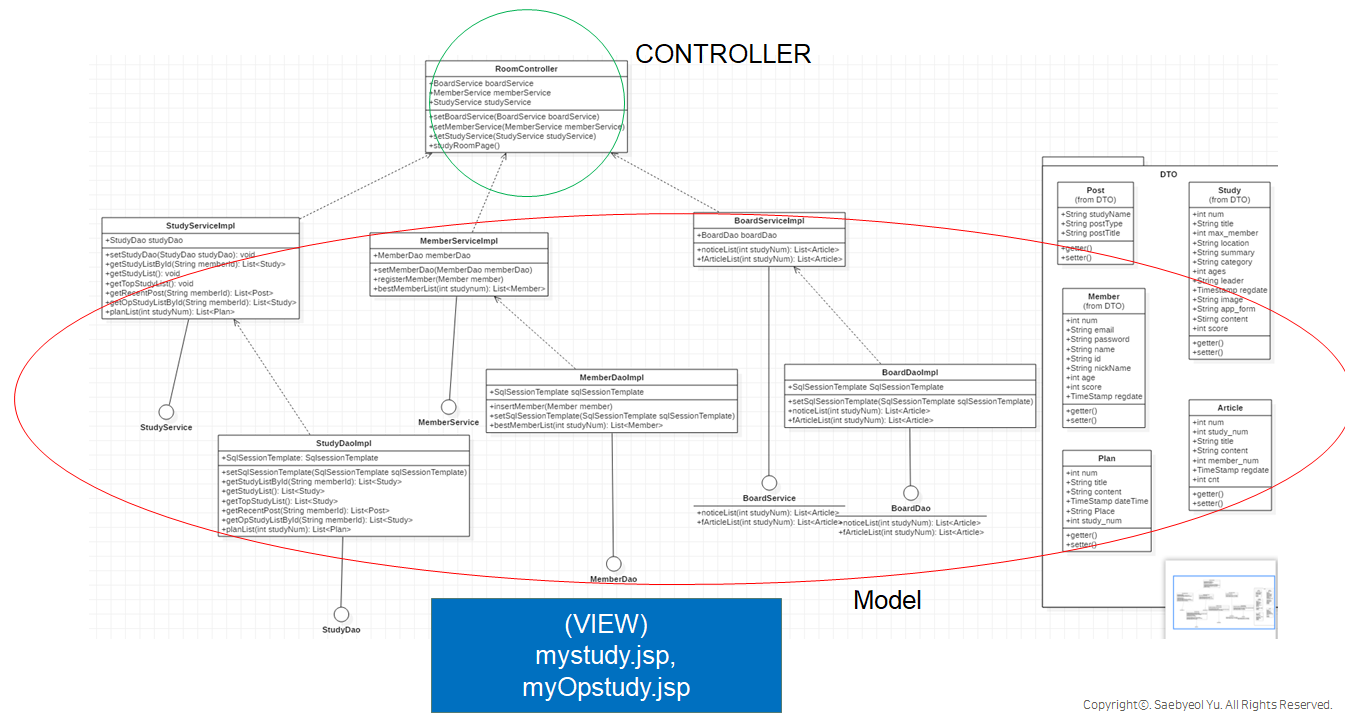
- 담당자 : 손상준



3-2 클래스 설계

▶ 스터디룸페이지

- 담당자 : 손상준



※ 자세한 설계 및 소스코드주소  
<https://github.com/tkdwns1245/studyTeamProject>

3-3 UI 설계

◆ 담당자 : 손상준

▶ 메인화면

3-3 UI 설계

▶ 로그인화면

3-3 UI 설계

▶ My스터디

3-3 UI 설계

▶ 스터디리스트

3-3 UI 설계

▶ 스터디상세

3-3 UI 설계

▶ 스터디등록

3-3 UI 설계

▶ 스터디 가입신청서

3-3 UI 설계

▶ 스터디룸화면

3-3 UI 설계

▶ 멤버관리

3-3 UI 설계

▶ 출석관리

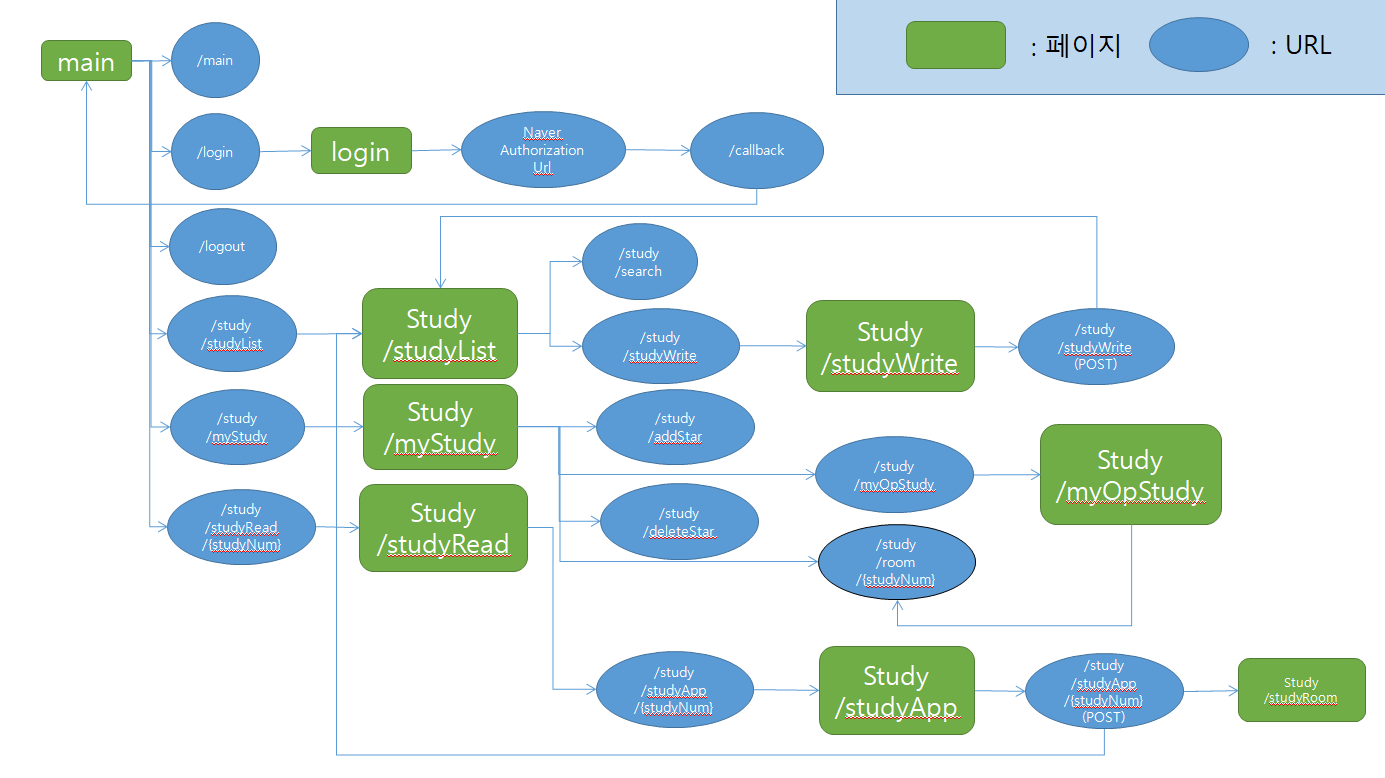
3-3 UI 설계

▶ 스터디일정

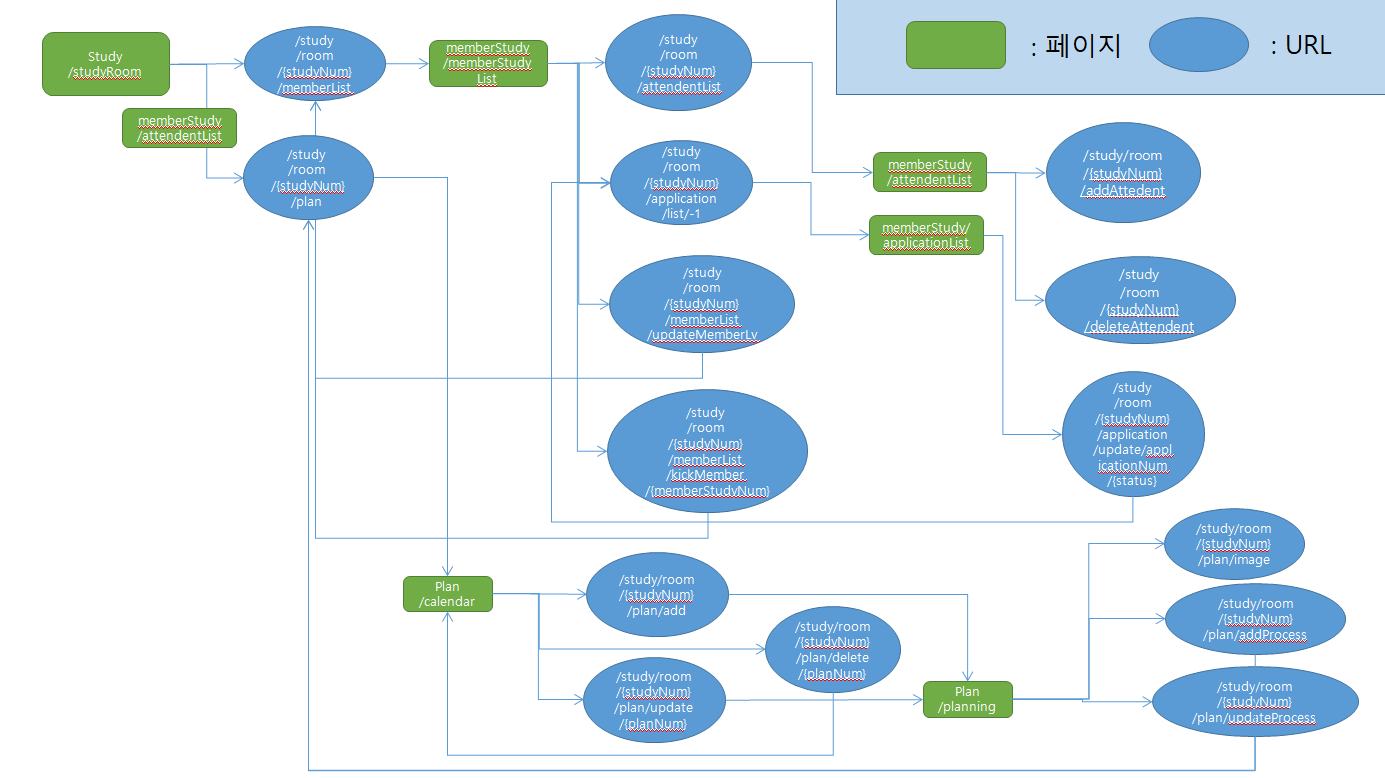
3-3 UI 설계

▶ 자유, 공지사항 게시판

3-3 URL 설계

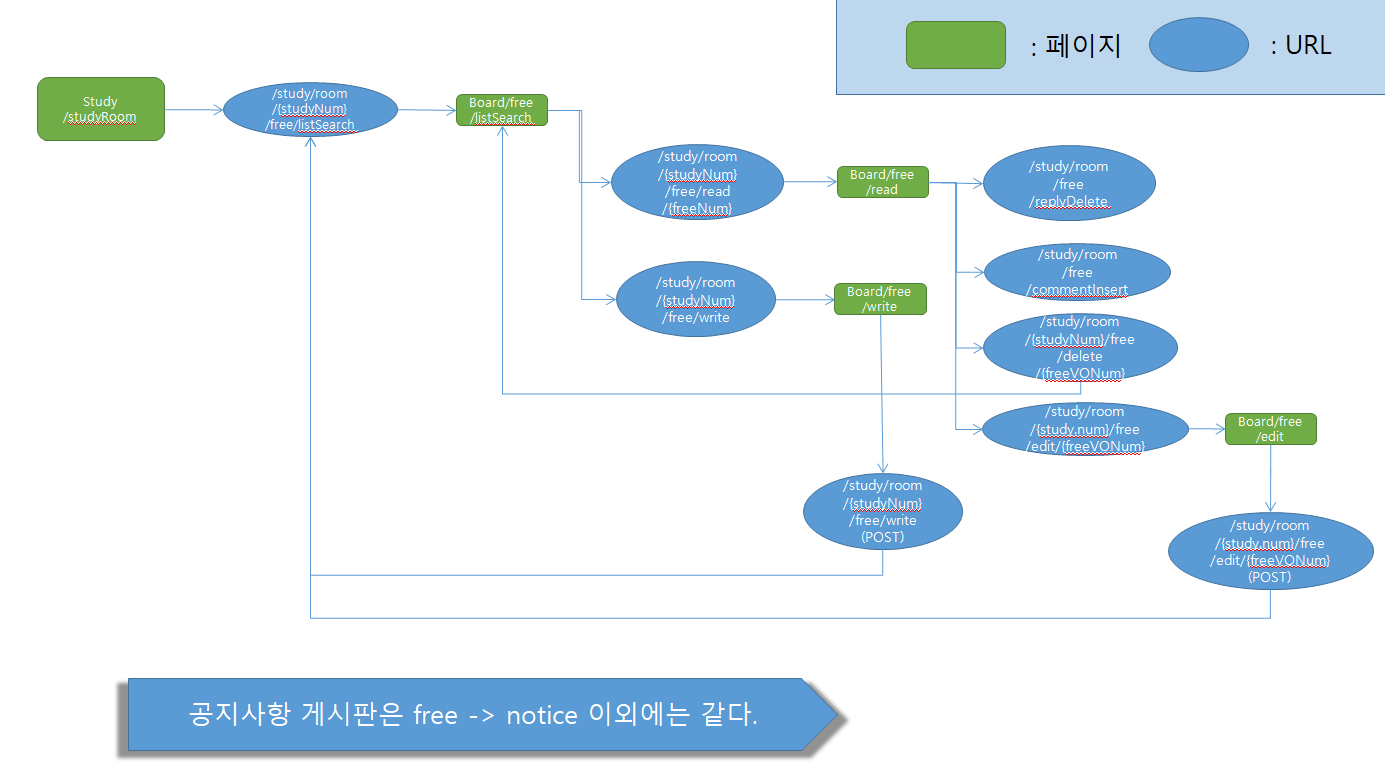
▶메인페이지,마이스터디페이지

3-3 URL 설계

▶스터디룸,일정관리,멤버관리,출석관리

가입신청서 관리

3-3 URL 설계

▶ 자유게시판,공지게시판

Ⅳ. **구현**

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 메인페이지(담당자 : 손상준) |
| 구현 UI |  |
| 구현 내용 | - 상단 바 로그인 버튼  - 로그인  • 네이버 아이디로 로그인API(네아로)를 활용하여 구현  - 상단 바 메뉴 기능(항목 : 스터디리스트,내스터디) |
| 구현 클래스 | - MainController - NaverLoginBO  - LoginController  - NaverLoginAPI |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 메인페이지 내용(담당자 : 손상준) |
| 구현 UI |  |
| 구현 내용 | 1. 스터디리스트 : 생성된 스터디 리스트 출력 및 더보기 버튼 클릭 시 스터디 리스트 페이지로 이동  2. 인기스터디리스트 : 스터디의 멤버수, 게시글 에 따라 점수가 부여되어 이를 기준으로 스터디 리스트 출력  • 점수 는 해당 스터디의 멤버수와 게시글 수에 따라서 설정되며 DB단에서 Query로 처리  3. 최근스터디소식 : 로그인 시에만 활성화 되는 리스트로 즐겨찾기에 추가된 스터디의 일정과 게시판 소식을 게시일 순으로 출력  • PostUtil 클래스를 만들어 Article과 Plan을 Post 형식으로 변환한 뒤 일정순으로 정렬한 리스트를 View에 전달 |
| 구현 클래스 | - MainController, PostUtil |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 스터디리스트(담당자:이태관) |
| 구현 UI |  |
| 구현 내용 | - 스터디 검색기능: 장소별, 카테고리별, 스터디이름으로 검색기능  • 점수 는 해당 Location, Category, Keyword별 7가지 모든 경우의 수에 해당하는 Service의메소드를만들어View에 스터디 리스트를 전달  - 스터디리스트 페이징처리  • StudyPageMaker의 calcData( ) 메서드에서 페이지 카운팅 처리  • StudyPageMaker의 makeQuery( ) 메서드에서 해당 페이지 링크주소 리턴 |
| 구현 클래스 | - StudyController  - StudyPageMaker |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 스터디개설(담당자:이태관) |
| 구현 UI |  |
| 구현 내용 | - 스터디개설기능: 양식에 따라 입력하여 스터디 개설  - 에러처리 : 필수항목 미입력시 에러메시지 출력  • 필수항목 미입력시 javascript로 해당 항목을 체크하여 에러메시지 팝업 |
| 구현 클래스 | - StudyController |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 마이스터디(담당자:손상준) |
| 구현 UI |  |
| 구현 내용 | - 즐겨찾기 기능 : 가입된 스터디 중 즐겨찾기 버튼을 클릭하여 해당 스터디 즐겨찾기에 추가  • ajax로 비동기식 화면처리  - 스터디 룸 페이지 이동 : 스터디 리스트의 타이틀 클릭하여 스터디 룸 페이지로 이동 |
| 구현 클래스 | - StudyController |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 스터디룸(담당자:손상준) |
| 구현 UI |  |
| 구현 내용 | - 상단 메뉴바(항목 : 스터디 관리, 스터디 일정, 스터디 정보, 게시판, 공지사항)  - 스터디룸 내용(게시판,공지사항,일정,베스트멤버) |
| 구현 클래스 | - StudyController |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 자유,공지사항게시판(담당자:김민호) |
| 구현 UI |  |
| 구현 내용 | - 공지사항 게시판 : 글쓰기 버튼은 운영자만 활성화  - 공통 : 글쓰기,글수정,글삭제, 페이징처리, 검색기능 구현  • PageMaker의 calcData( ) 메서드에서 페이지 카운팅 처리  • PageMaker의 makeQuery( ) 메서드에서 해당 페이지 링크주소 리턴 |
| 구현 클래스 | - FreeController,NoticeController,PageMaker |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 게시판 글쓰기(담당자:김민호) |
| 구현 UI |  |
| 구현 내용 | - SUMMER NOTE를 활용하여 게시판 구현  • 이미지 업로드 Ajax로 비동기식 화면처리  - 필수내용 미 입력시 에러메시지 출력  • validator 클래스를 활용하여 back-end 단에서 처리 |
| 구현 클래스 | - FreeController,NoticeController,ArticleValidator |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 게시판 댓글수정 및 삭제(담당자:손상준) |
| 구현 UI |  |
| 구현 내용 | - 댓글 수정 및 삭제 버튼  • Ajax로 비동기식 화면처리 |
| 구현 클래스 | - FreeController,NoticeController,ArticleValidator |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 스터디 일정(담당자:강덕현) |
| 구현 UI |  |
| 구현 내용 | - 캘린더 기능(추가된 일정 표시)  • calendar 클래스를 활용하여 달력 구현 하였으며 CalendarUtil에서 년,월에 해당하는 일수를 받아온다.,  - 일정 추가기능(캘린더에서 ‘일’을 클릭한 후 하단의 추가하기 버튼을 클릭하여 일정을 추가) |
| 구현 클래스 | - PlanController,CalendarUtil |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 스터디 일정추가(담당자:강덕현) |
| 구현 UI |  |
| 구현 내용 | - SUMMER NOTE를 활용하여 일정 추가 기능 구현  • 이미지 업로드 Ajax로 비동기식 화면처리  - 필수 내용 미 입력시 에러 문자 출력  • validator 클래스를 활용하여 Back-end단에서 에러처리 |
| 구현 클래스 | - PlanController  - PlanAddRequestValidator  - PlanUpdateRequestValidator |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 상세 일정(담당자:강덕현) |
| 구현 UI |  |
| 구현 내용 | - 캘린더 날짜에 추가된 일정을 클릭시 상세 일정 페이지 출력 |
| 구현 클래스 | - PlanController |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 멤버 관리(담당자:안성도)(도움: 강덕현) |
| 구현 UI |  |
| 구현 내용 | - 룸페이지의 상단메뉴의 스터디관리 항목을 클릭하여 이동  - 운영진은 멤버 등급 조정 가능  - 운영진은 멤버 강퇴버튼을 사용 가능 |
| 구현 클래스 | - MemberStudyController |

4-1 UI 및 기능구현

|  |  |
| --- | --- |
| 구현 페이지 | 출석 관리(담당자:안성도)(도움: 강덕현) |
| 구현 UI |  |
| 구현 내용 | - 룸페이지의 상단메뉴의 스터디관리 항목을 클릭하여 이동  - 캘린더의 일정을 클릭하여 출석여부 체크 기능 |
| 구현 클래스 | - MemberStudyController |

4-1 UI 및 기능구현

▶가입신청서 관리

|  |  |
| --- | --- |
| 구현 페이지 | 가입신청서 관리(담당자:안성도)(도움: 강덕현) |
| 구현 UI |  |
| 구현 내용 | - 운영진은 가입신청서을 수락 및 거절 가능 |
| 구현 클래스 | - MemberStudyController |

4-2 QA테스트

|  |  |  |  |
| --- | --- | --- | --- |
| **페이지이름** | **구현내용** | **테스트내용** | **테스트결과** |
| 메인 | 로그인 | 네이버아이디 로그인 연동 | 문제없음 |
| 메인 | 스터디최근소식 | 즐겨찾기에 추가된 스터디의 게시판,일정 포스팅 | 문제없음 |
| 메인 | 인기스터디리스트 | 스터디리스트 점수별 인기스터디리스트 포스팅 | 문제없음 |
| 스터디리스트 | 스터디검색 | 조건별 검색기능 | 문제없음 |
| 스터디리스트 | 스터디만들기 에러처리 | 내용이 없을시 에러메시지 발생 | 문제없음 |
| 마이스터디 | 즐겨찾기 | 즐겨찾기 버튼을 클릭 시 즐겨찾기 추가 | 문제없음 |
| 마이스터디 | 룸페이지 이동 | 타이틀 클릭 시 룸페이지로 이동 | 문제없음 |
| 스터디룸 | 게시판,공지게시판,일정게시판 | 게시글 포스팅 | 문제없음 |

4-2 QA테스트

|  |  |  |  |
| --- | --- | --- | --- |
| **페이지이름** | **구현내용** | **테스트내용** | **테스트결과** |
| 자유게시판 | 게시판 글쓰기,수정,삭제 | 글쓰기 에러처리,목록이동 | 목록이동에 에러발생글쓰기 에러처리 미구현 (처리완료) |
| 자유게시판 | 검색기능 | 조건별 검색기능 | 문제없음 |
| 자유게시판 | 댓글기능 | 댓글 추가 및 삭제 | 문제없음 |
| 공지사항 게시판 | 게시판 글쓰기,수정,삭제 | 에러처리,목록이동,운영자만 글쓰기 버튼 활성화 | 목록이동에 에러발생, 글쓰기 에러처리 미구현 (처리완료) |
| 공지사항 게시판 | 검색기능 | 조건별 검색기능 | 문제없음 |
| 공지사항 게시판 | 댓글기능 | 댓글 추가 및 삭제 | 문제없음 |
| 스터디일정 | 일정추가 | 일정 추가 기능에러처리 및 캘린더에 추가된 일정 확인 | 문제없음 |

Ⅴ. **마치고**

5-1 어려웠던 점



▶ 어려웠던점

1. **통일성 있고 완성된 설계의 어려움**

- 팀원 개개인이 기능별로 설계를 하고 이를  
 통합하는 과정에서 엇갈린 설계로 인한 어  
 려움이 있었다.

1. **프로젝트 통합의 어려움**

- 처음 Git으로 통합하는 과정에서 생긴 conflict를 처리하는 데 까다로웠다.

- Git을 활용하는 방법이 생소하여 실수로 프로젝트가 날라가는 등의 문제들이 발생하였다. 이를 해결하기위해 Git이외에도 백업의 필요성을 느꼇다.

- 필요한 기능들이 필요한 시기에 적절히 완성되지 않아 테스트 하기 위  
해서 다른 사람이 기능을 완성하기를 기다려야 하는 번거로움이 생겼다.

감사합니다

GithubBlog : tkdwns1245.github.io