

소프트웨어공학

Checkbox ✓ 웹 앱

(2분반) 난쏘공 팀



목차

01. 프로젝트 계획 및 관리

02. 팀 구조

03. 요구사항 분석 결과

04. 설계 결과

05. 시연

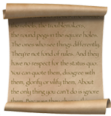
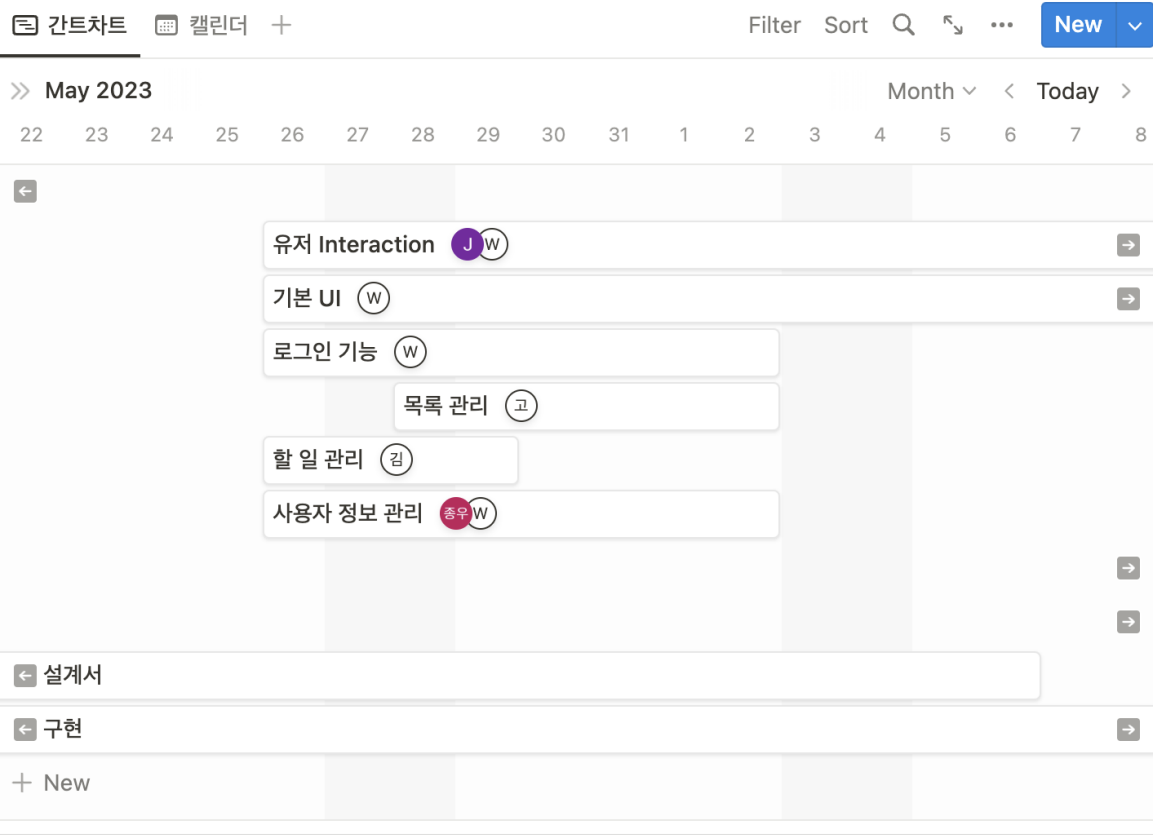
06. Lessons Learned

프로젝트 계획 및 관리



노선을 통한 프로젝트 계획 및 관리

GANTT CHART



회의록

회의록 작성 순서
김우석 → 고민영 → 김종우 → 김상호 → 김진호

회의록

Aa 이름	회의 유형	생성자
03월 31일 금요일	주간 회의	(W) WooSeok Kim
04월 07일 금요일	주간 회의	(W) WooSeok Kim
05월 12일 금요일	주간 회의	(고) 고민영
05월 17일 수요일	임시 회의	(종우) 종우 김
05월 19일 금요일	주간 회의	(김) 김상호
05월 26일 금요일	주간 회의	(J) 김진호
05월 29일 월요일	임시 회의	(W) WooSeok Kim
06월 02일 금요일	주간 회의	(고) 고민영
06월 06일 화요일	임시 회의	(종우) 종우 김

Microsoft 를 통한 문서 관리

The screenshot shows a Microsoft Word document titled '요구사항명세서_난쏘공_v1'. The document contains a table of contents and a list of requirements. The table of contents lists four scenarios (SN001 to SN004) with their corresponding page numbers. The requirements section includes a table with columns for '설명' (Description), '관련 액터' (Related Actor), '사전 조건' (Pre-conditions), and '사후 조건' (Post-conditions). The table lists various requirements for a system, including user login, password reset, and system status checks.

설명	관련 액터	사전 조건	사후 조건
회원이 특정 목록 내의 할 일(들) 중 완료한 할 일을 완료로 표시한다.	회원	회원은 UC002를 통해 로그인한 상태여야 한다.	회원은 UC008을 통해 특정 목록 내의 할 일을 추가한 상태여야 한다.
시스템은 회원이 체크박스를 통해 완료로 표시한 할 일은 회색으로 바꾼다.			
회원은 완료한 할 일의 체크박스를 체크하여 할 일을 완료로 표시한다.			
시스템은 회원이 선택한 할 일을 완료된 할 일로 처리한다.			
시스템은 회원이 선택한 할 일을 회색 글씨로 변경시킨다.			
회원이 선택한 할 일을 완료로 처리 못한 경우: - 시스템은 회원에게 해당 할 일을 완료 처리하는 것에 실패했다는 메시지를 출력한다. - 시스템은 회원이 완료 표시한 할 일을 미완료 상태로 놔둔다.			

3.2.2.15 UC015 할 일 완료로 취소한다.

설명	관련 액터	사전 조건
회원이 목록에서 완료 표시된 할 일을 취소하여 다시 할 일로 전환한다.	회원	회원은 UC002를 통해 로그인한 상태여야 한다.



노션을 통한 문서 관리



문서

API 명세

유용한 정보, 자료 공유

- Star UML
 - Checkbox_UML_v2.1.mdj 1617.9KB
 - Checkbox_UML_v2.0.mdj 1571.0KB
 - v2.0)
 - Checkbox_UML_v2.0.mdj 938.3KB
 - v2.3)
 - UseCaseDiagram_v2.3.mdj 278.4KB
 - v2.2)
 - UseCaseDiagram_v2.2.mdj 388.8KB
 - v2.1)
 - UseCaseDiagram_v2.mdj 375.7KB
 - v1)
 - UseCaseDiagram_v1.mdj 118.9KB

시스템 요구사항 명세서

v2.0~)

sangmyungkr-my.sharepoint.com

https://sangmyungkr-my.sharepoint.com/:w:/g/personal/201910939_sangmyung_kr/ESL97SPloJ1LnNN9Nu...

v1.6) 제출본

요구사항명세서_난쏘공_v1.6.docx 646.4KB

v1.2)

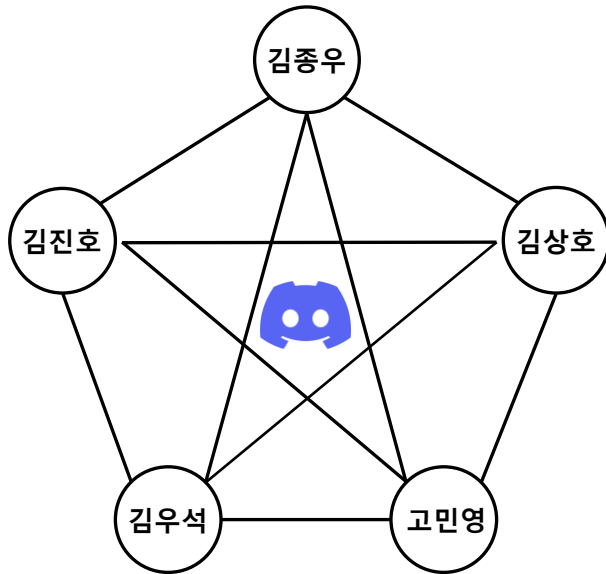
요구사항명세서_난쏘공_v1.2.docx 763.7KB

실시간 반영

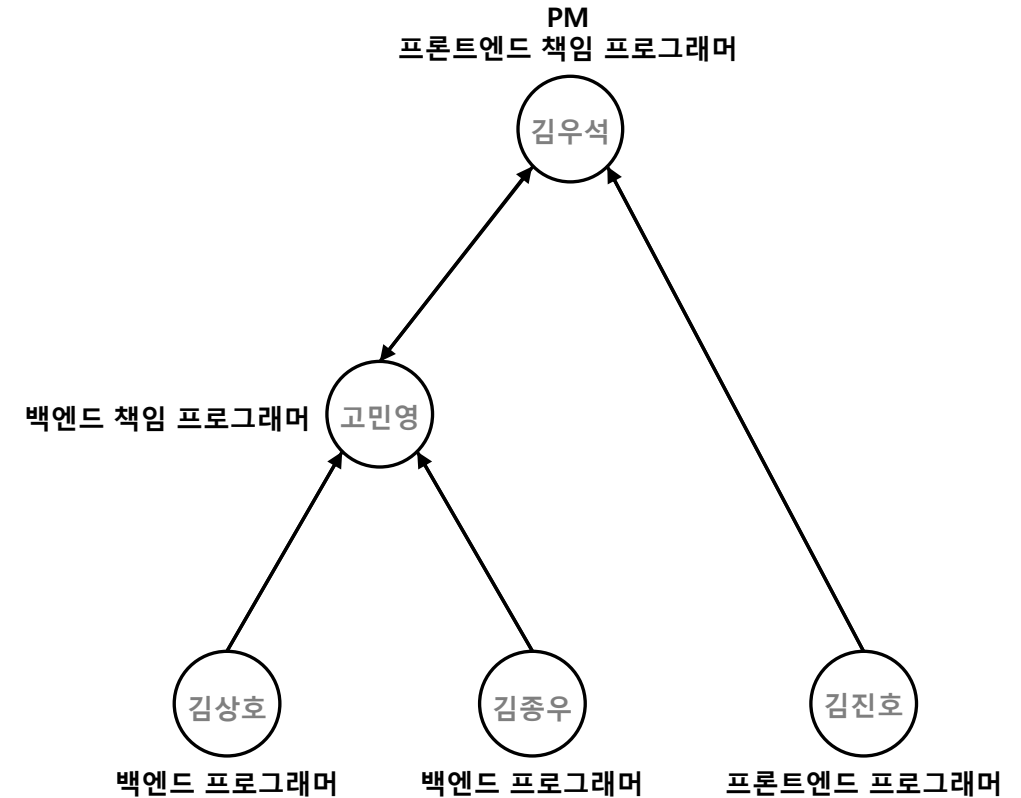
팀 구조

팀 구조

회의 진행 시 민주적 팀 구성 사용



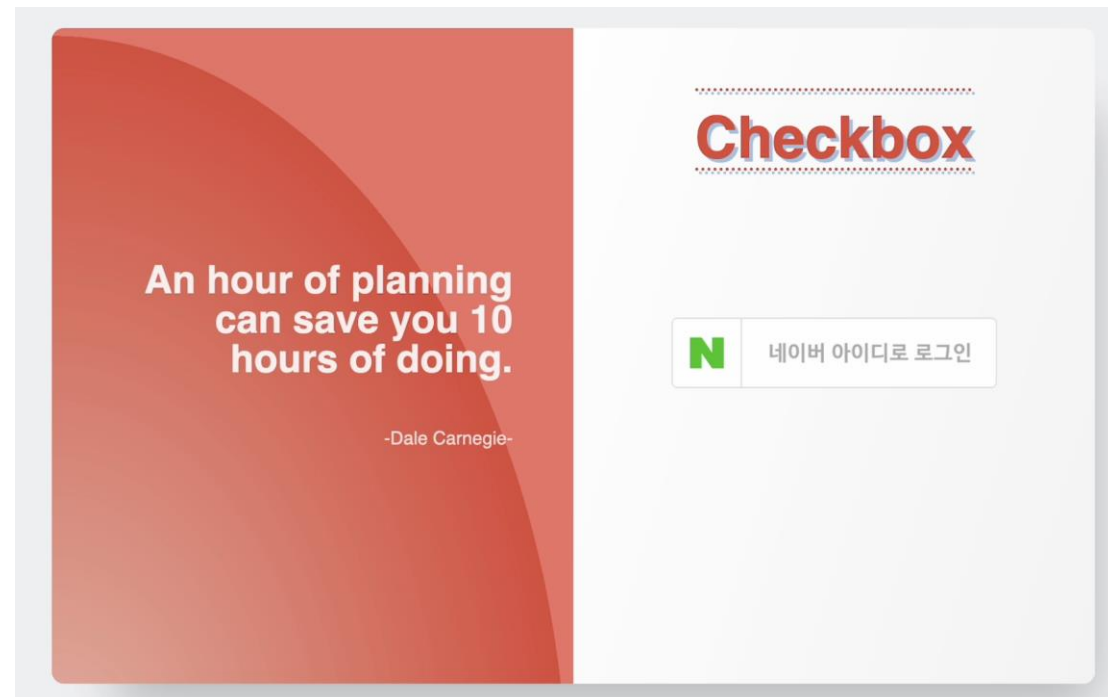
개발 진행 시 계층적 팀 구성 사용



요구사항 분석 결과


+ 고객의 **NAVER** 로그인 기능 요청


사용자별 목록 및 할 일 저장




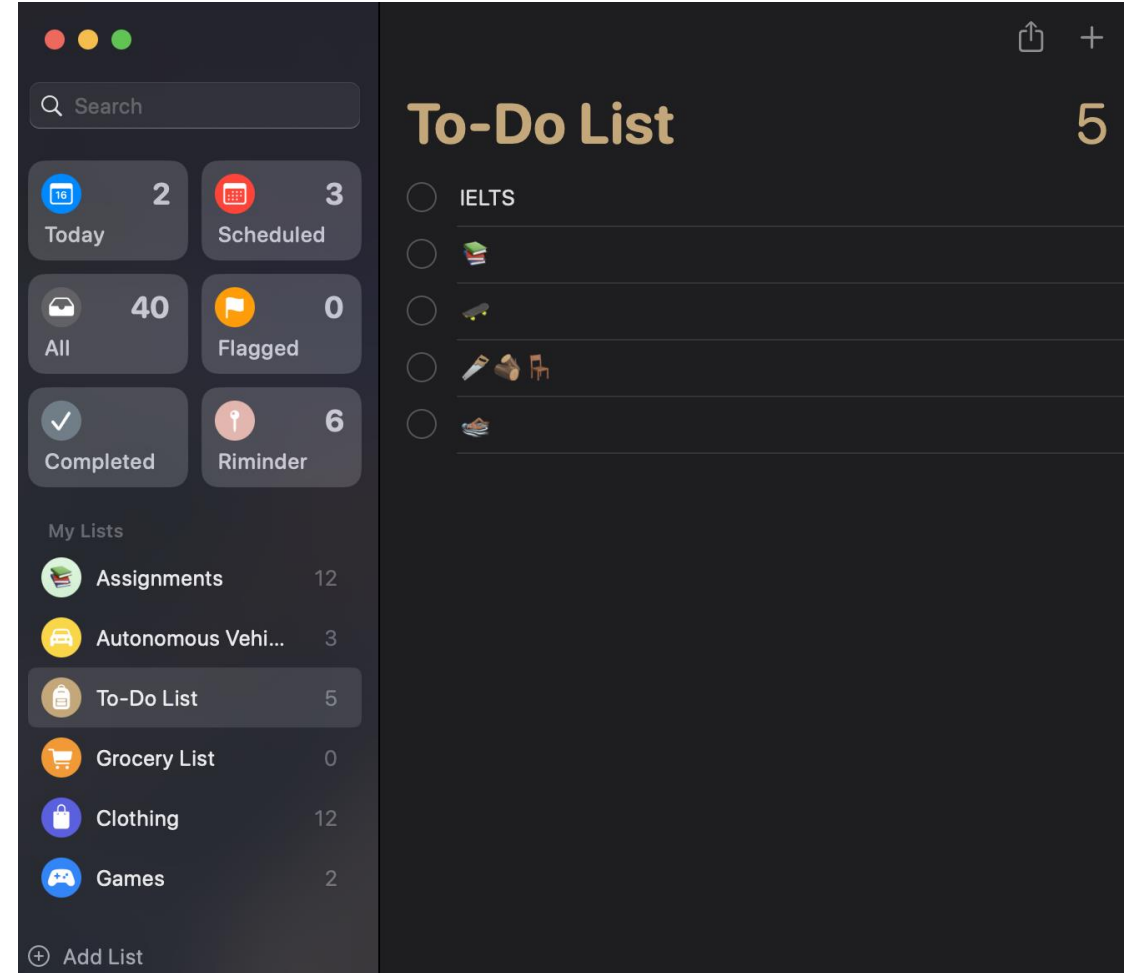
요구사항 분석 결과

? 고객이 Apple의  앱 벤치마킹 요청
Reminders

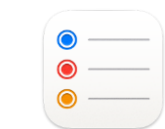
 목록별 할 일 정보 저장

 마감일 설정

 정렬 기능 및 완료된 할 일만 보기 옵션



01. 실제



앱 사용 Reminders

02. 실사용을 기반으로 프로젝트 개요 작성

1. 시스템 개요

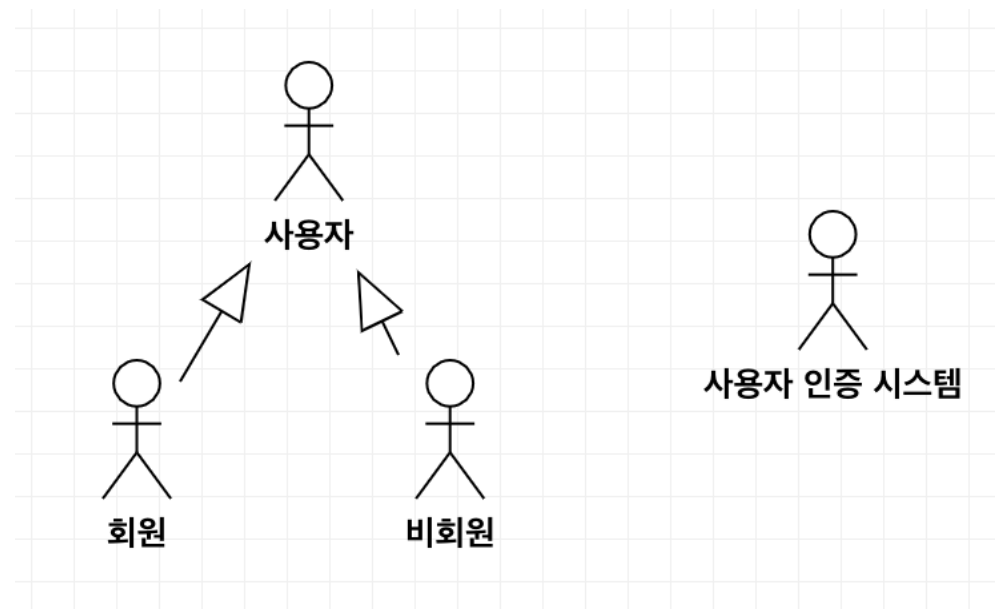
Check Box는 사용자의 할 일들 정리 및 관리를 도와주는 소프트웨어이다. 사용자는 할 일들을 추가하고 편집할 수 있으며, 완료된 할 일은 완료로 표시하고 현재까지 사용자가 한 일을 한눈에 확인할 수 있다. 또한 사용자에게 더 이상 필요없는 할 일은 삭제할 수 있다.

사용자는 목록 기능을 사용하여 할 일들을 체계화할 수 있다. 사용자의 요구에 따라 여러 개의 목록을 만들어 해당 목록에 할 일들을 추가하여 카테고리별로 정리할 수 있다. 사용자가 직접 지정한 목록으로 정리된 할 일들을 보여줌으로써 사용자는 할 일에 대한 생산성이 높아지는 효과를 얻을 수 있을 것이다.

또한 해당 애플리케이션은 할 일들을 추가할 때 언제까지 해당 일을 해야 하는지 알려주는 마감일 기능, 어느 일부부터 처리해야 하는지 알려주는 우선순위 기능, 그리고 마감일 전에 사용자를 상기시켜줄 수 있는 미리 알림 기능을 사용할 수 있다. 이러한 기능들로 인해 사용자는 보다 체계적이고 효과적으로 할 일들을 전문적으로 처리하는 것을 기대할 수 있다.

위의 기능들을 토대로 정렬 기능도 지원한다. 할 일을 추가할 때 설정한 우선순위 또는 마감일이 임박한 할 일들을 순서로 정렬이 가능하여 사용자의 요구에 맞춰 해당 애플리케이션은 유동적으로 변화가 가능하다.

03. 프로젝트 개요를 기반으로 액터 정의



요구사항 분석 결과 – 추출 과정

04. 프로젝트 개요 내용을 기반으로 기능 단위로 끊기

e. g., 사용자 관련 동작: 등록, 로그인

05. 고객 기능 요구사항 작성

3. 요구사항 분석

3.1 고객 기능 요구사항

구분	상세	비고
사용자 등록 및 로그인	(사용자 등록, 회원가입) 신규 사용자가 회원 가입을 하게 되면, 시스템은 회원 가입 절차를 수행하게 된다. (로그인) 회원 가입된 사용자가 로그인을 시도하면, 시스템은 로그인 절차를 수행하고 정상적으로 인증되면 홈 화면으로 이동한다.	회원이 입과 로그인 은 "네이버" 를 통해 진행한다.
여러 개의 할 일 목록을 만드는 기능	사용자가 할 일들을 분류할 수 있는 목록을 만든다. 여기서 목록이란, 여러 개의 할 일들을 카테고리화시킬 수 있는 상위 폴더의 개념을 말한다.	
각 목록에 작업을 추가하는 기능	사용자가 만들어 둔 목록 안에 포함된 할 일을 추가하여 마감일, 우선순위, 그리고 미리 알림을 설정한다. 마감일, 우선순위, 그리고 미리 알림에 대한 설정 유무는 선택사항이다.	
작업을 완료로 표시하는 기능	시스템에 등록된 사용자가 자신의 할 일 목록 중에 완료한 작업을 완료로 표시한다.	
작업 삭제 기능	사용자가 자신이 추가했던 작업에 대해 삭제를 요청하면 해당 작업을 삭제 시킨다.	
작업 편집 기능	사용자가 자신이 추가했던 작업에 대해 세부내용, 마감일, 우선순위, 그리고 미리 알림 설정여부에 관한 정보를 각각 수정할 수 있다.	
완료된 작업을 보는 옵션	사용자는 목록 내에서 여러 할 일들 가운데 완료된 작업만 보는 옵션을 통해 완료된 할 일들만 출력된다.	
마감일 또는 우선순위에 따라 작업을 정렬하는 옵션	사용자가 할 일 정렬 옵션으로 마감일과 우선순위 중 한 가지를 선택하면, 선택한 옵션으로 작업을 정렬한다.	

06. 고객 기능 요구사항을 행동 단위로 끊기

e. g., 사용자 등록을 한다. 로그인을 한다.

07. 객체지향 분석을 통한 요구사항 분석

유스케이스의 실체화 및 구축

08. 요구사항 명세

ID	유스케이스 명	설명	우선 순위
UC001	사용자 등록을 한다.	사용자별 정보를 저장하기 위해 네이버 사용자 등록을 한다.	0 상
UC002	로그인을 한다.	사용자가 서비스를 이용하기 위해서 로그인을 한다.	1 상
UC003	로그아웃을 한다.	사용자가 로그아웃을 한다.	1 우
UC004	목록을 추가한다.	사용자의 할 일을 담은 목록을 추가한다.	2 상
UC005	목록을 삭제한다.	사용자가 목록을 삭제한다.	3 우
UC006	목록을 편집한다.	사용자가 목록을 편집하여 목록의 이름을 변경한다.	3 우
UC007	할 일을 조회한다.	사용자가 특정 목록을 선택하여 해당 목록 내의 할 일들을 조회하도록 한다.	2 우
UC008	할 일을 추가한다.	목록안에 사용자의 할 일을 추가한다.	2 중
UC009	미리 알림을 설정한다.	사용자가 추가할 할 일에 미리 알림 시간을 설정한다.	3 중
UC010	마감일을 설정한다.	사용자가 추가할 할 일의 마감일을 설정한다.	3 중
UC011	우선순위를 설정한다.	사용자가 추가할 할 일의 우선순위를 설정한다.	3 진
UC012	할 일을 삭제한다.	사용자가 선택한 할 일을 삭제한다.	3 진
UC013	할 일을 편집한다.	사용자가 기존 할 일을 편집한다.	3 진
UC014	할 일 완료 표시를 한다.	사용자가 특정 목록 내의 할 일(들) 중 완료한 할 일을 완료로 표시한다.	2 우
UC015	할 일 완료를 취소한다.	사용자가 목록에서 완료 표시된 할 일을 취소하여 다시 할 일로 전환한다.	2 우
UC016	완료된 할 일만 조회한다.	사용자가 목록 내에서 현재까지 완료된 할 일만을 조회한다.	4 우
UC017	마감일에 따라 할 일을 정렬한다.	사용자가 할 일을 마감일에 따라 정렬한다.	4 민
UC018	우선순위에 따라 할 일을 정렬한다.	사용자가 정한 우선순위에 따라 할 일을 정렬한다.	4 민

요구사항 분석 결과 - 추출 과정

09. 유스케이스명을



앱에서 실행해보기

Reminders

10. 전체적인 흐름을 요구사항 상세기술서에 작성

3.2.2.1 UC001: 사용자 등록을 한다.



설명	비회원이 회원가입을 진행한다.	
관련 액터	사용자, 사용자 인증 시스템	
사전 조건	비회원은 연동하고자 하는 계정이 아직 Check Box 와 연동되지 않은 상태여야 한다.	
사후 조건	시스템은 비회원을 초기 상태로 이동시킨다.	
기본 흐름	B01	비회원이 회원가입을 시도한다.
	B02	연동하고자 하는 ID, p/w를 입력한다.
	B03	약관사항에 동의한다.
	B04	시스템은 회원가입이 성공했음을 알린다.
대안 흐름	A01	UC002 B01에서 회원가입을 시도한다
예외 흐름	E01	연동하려는 ID가 이미 연동 완료 된 경우 (회원이 가입시도) -1. 시스템은 이미 회원가입이 완료된 계정이라는 메시지를 출력한다. -2. 시스템은 회원을 UC002를 실행한다.
	E02	약관사항에 동의하지 않은 경우 -1. 시스템은 회원가입에 실패했다는 메시지를 출력한다. -2. 시스템은 비회원을 사전조건으로 돌려놓는다.
	E03	회원가입 하려는 ID의 연동이 실패한 경우 -1. 시스템은 연동 실패 했다는 메시지를 출력한다. -2. 시스템은 비회원을 사전조건으로 돌려놓는다.
시나리오	SN001	B01→B02→B03→B04
	SN002	B01→B02→E01
	SN003	B01→B02→E03
	SN004	B01→B02→B03→E02
	SN005	A01→B02→B03

설계 결과

Server - Backend



명확한 Class 개념 존재



기존 클래스 추출법 사용

Client - Frontend



명확한 Class 개념이 존재하지 않음



기능/객체 단위로 임의로 클래스 다이어그램 작성

01. 실제



Reminders

앱 사용으로부터 추출된 개요 사용

02. 개요 내용으로부터 명사 추출법을 사용

단어	개수
<u>tasks</u>	<u>10</u>
<u>users</u>	<u>10</u>
deadlines	5
<u>lists</u>	<u>3</u>
organize	2
preferences	2
sort	2
completed	2
(생략)	

03. 각 단어가 포함된 유스케이스 조사

e. g., UC008: 할 일을 추가한다,
UC013: 할 일을 편집한다, *etc*

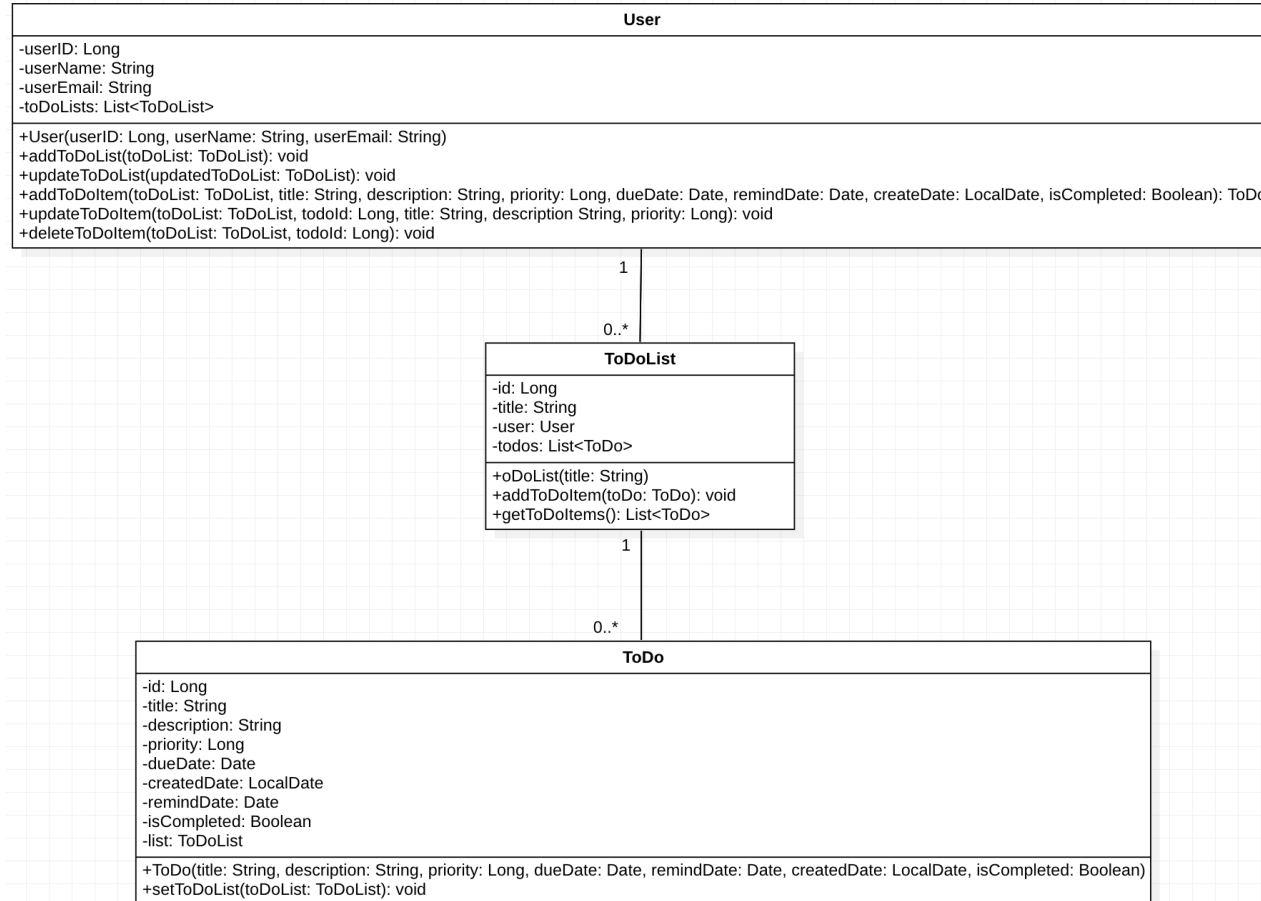
04. 해당 유스케이스로부터 명사 추출법 사용

단어	개수
<u>user</u>	<u>13</u>
<u>todo</u>	<u>12</u>
system	4
<u>list</u>	<u>2</u>
<u>title</u>	<u>2</u>
<u>description</u>	<u>2</u>
(생략)	

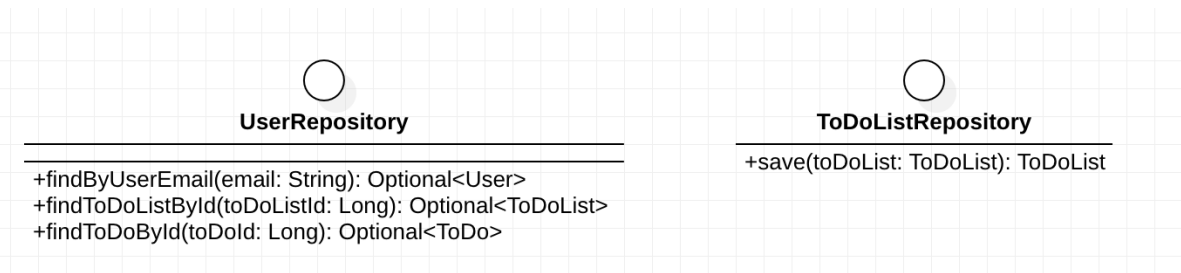
<“할 일을 추가한다” 중 단어 개수>

ToDo
-id: Long -title: String -description: String -priority: Long -dueDate: Date -createdDate: LocalDate -remindDate: Date -isCompleted: Boolean -list: ToDoList
+ToDo() +setToDoList(todoList: ToDoList): void

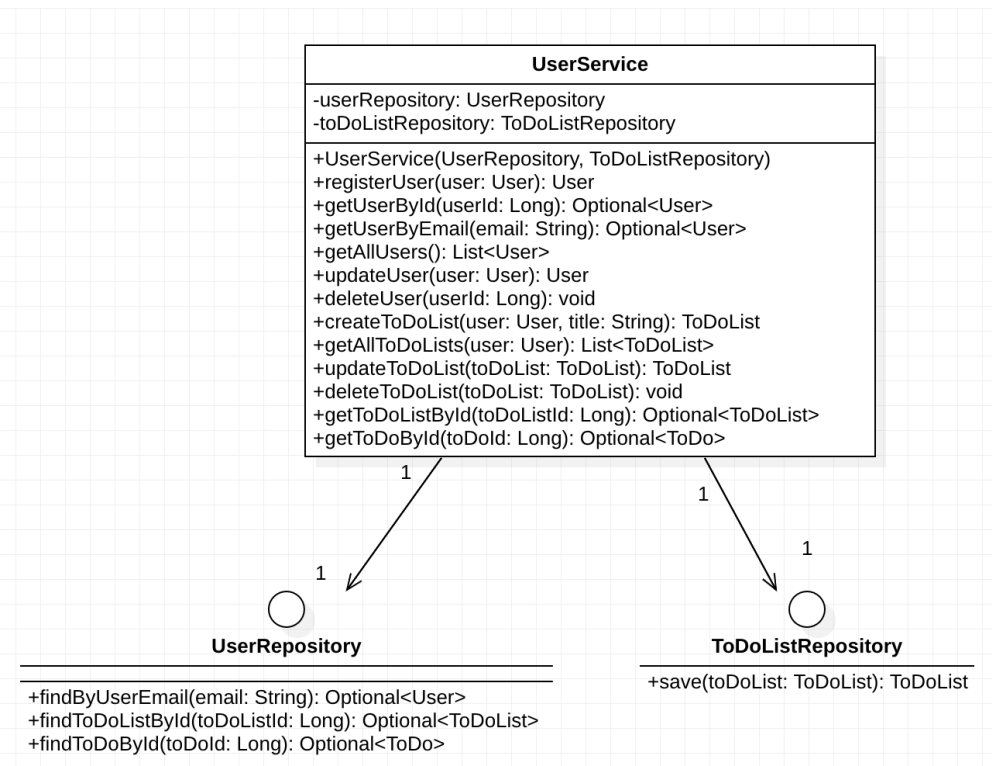
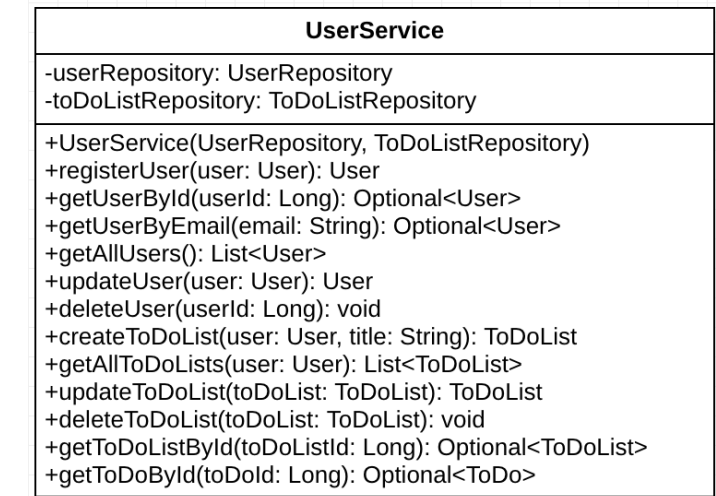
05. 클래스에 대한 관련 동작 (CRUD 함수) 기입



01. 항목에 대한 제어 이전에 필요한 함수를 인터페이스에 정의



02. 인터페이스에 대한 제어



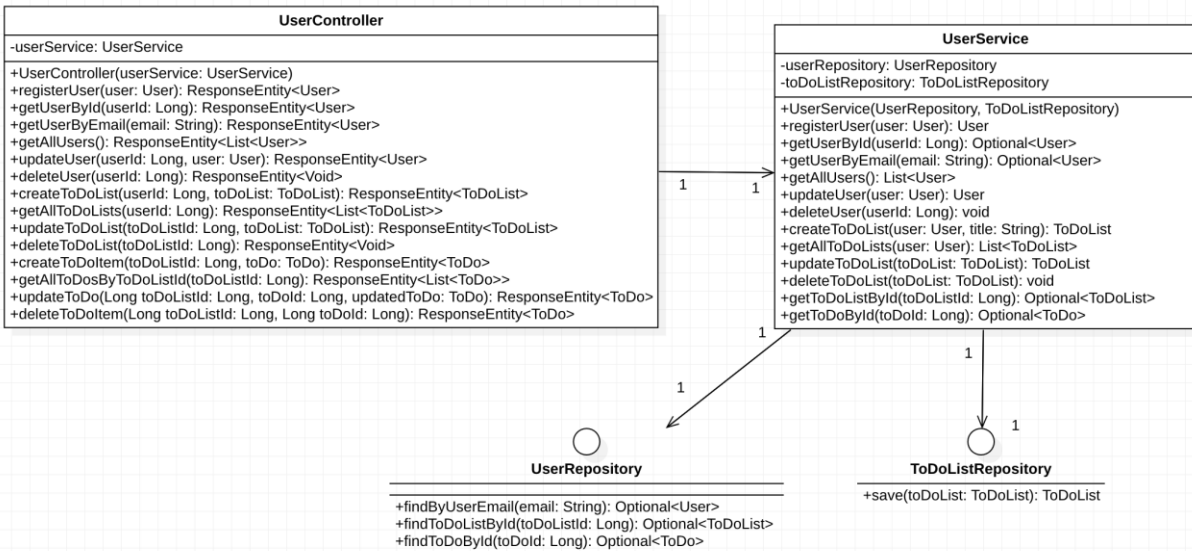
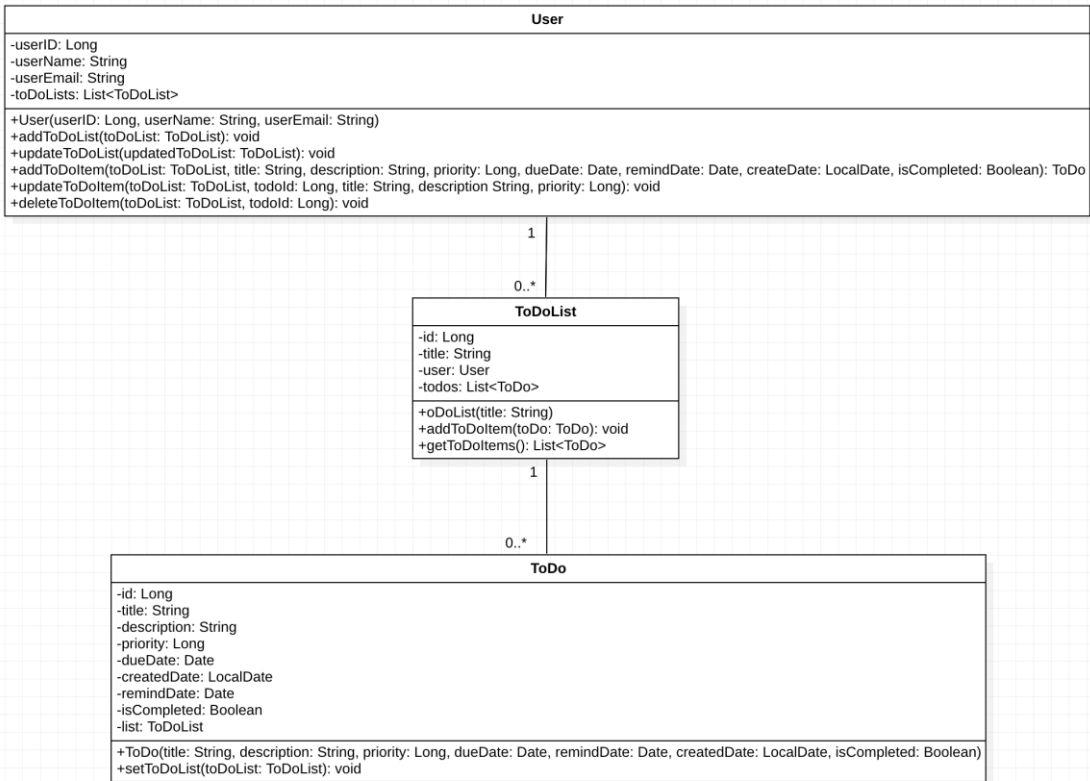
01. 유스케이스 단위의 기능을 처리할 클래스 인식

UserController

02. 유스케이스 단위의 기능을 처리할 함수 기입

UserController
-userService: UserService
+UserController(userService: UserService) +registerUser(user: User): ResponseEntity<User> +getUserById(userId: Long): ResponseEntity<User> +getUserByEmail(email: String): ResponseEntity<User> +getAllUsers(): ResponseEntity<List<User>> +updateUser(userId: Long, user: User): ResponseEntity<User> +deleteUser(userId: Long): ResponseEntity<Void> +createToDoList(userId: Long, toDoList: ToDoList): ResponseEntity<ToDoList> +getAllToDoLists(userId: Long): ResponseEntity<List<ToDoList>> +updateToDoList(toDoListId: Long, toDoList: ToDoList): ResponseEntity<ToDoList> +deleteToDoList(toDoListId: Long): ResponseEntity<Void> +createToDoItem(toDoListId: Long, toDo: ToDo): ResponseEntity<ToDo> +getAllToDosByToDoListId(toDoListId: Long): ResponseEntity<List<ToDo>> +updateToDo(Long toDoListId: Long, toDoId: Long, updatedToDo: ToDo): ResponseEntity<ToDo> +deleteToDoItem(Long toDoListId: Long, Long toDoId: Long): ResponseEntity<ToDo>

01. 클래스 연관관계를 고려하여 통합

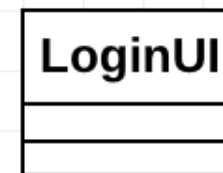
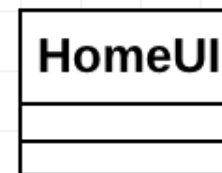


코드와의 일치율: **100%**

01. 실제

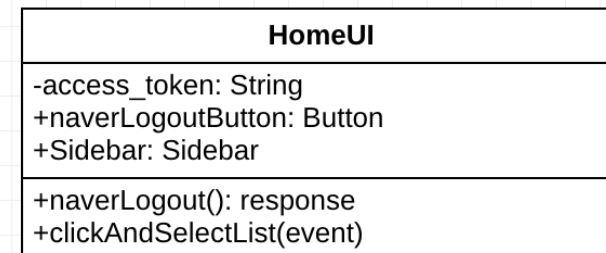
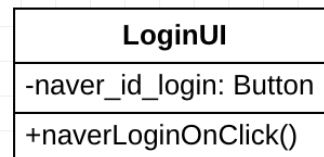


앱 사용을 통해 화면 추출
Reminders



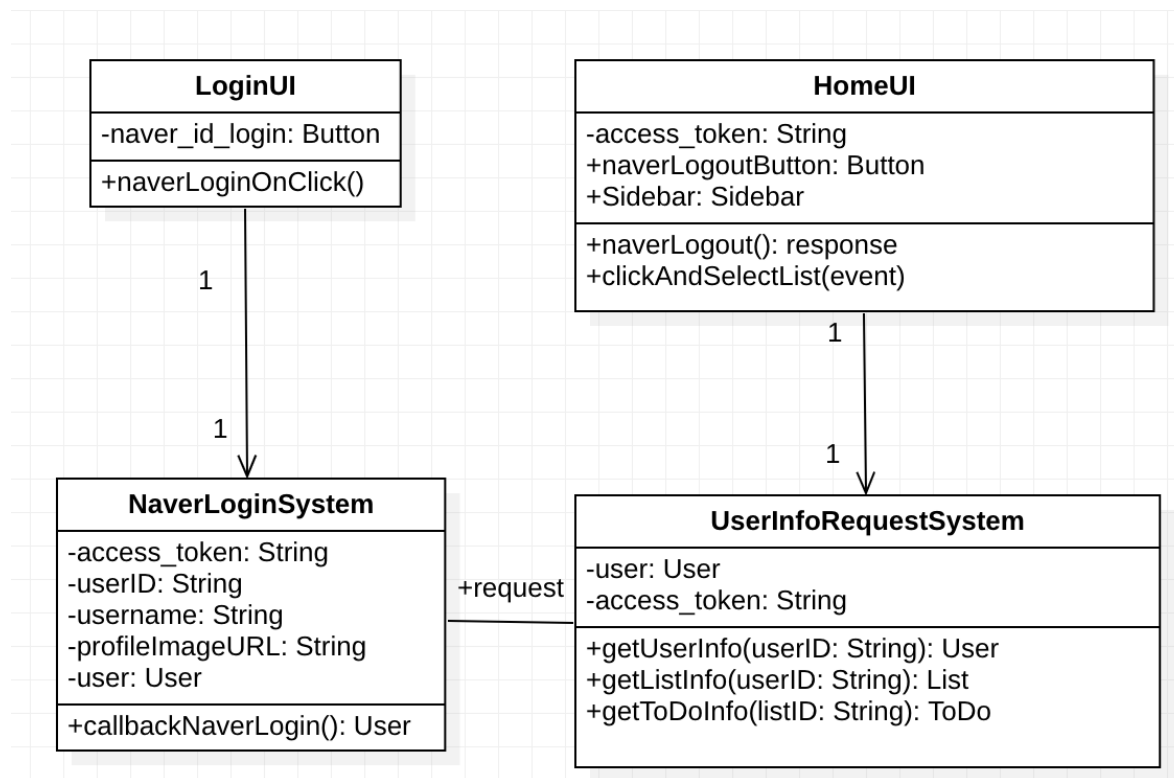
02. 창 이동이 필요한 유스케이스 참조

03. 화면에 기능이 들어간 버튼 추가

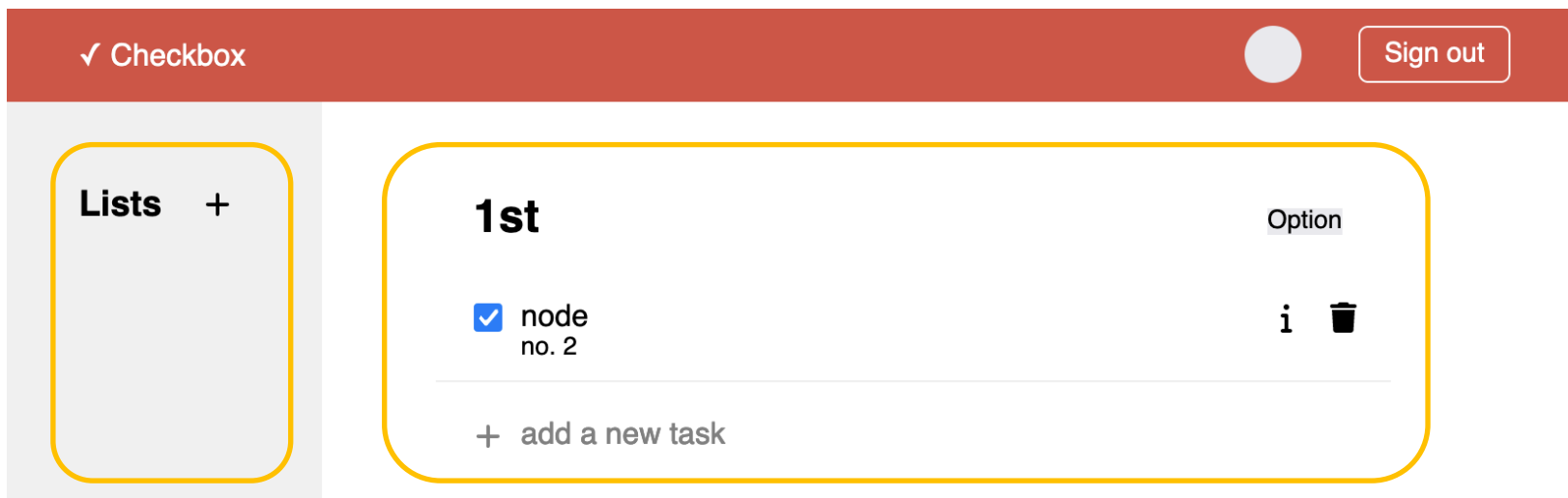


01. 경계 클래스간의 연관관계 고려

02. 둘 사이의 연관관계 고려



01. 화면 어디에 목록, 그리고 할 일 정보가 들어갈지 인식, 그리고 화면 분할



<프로토타입으로 찾기>

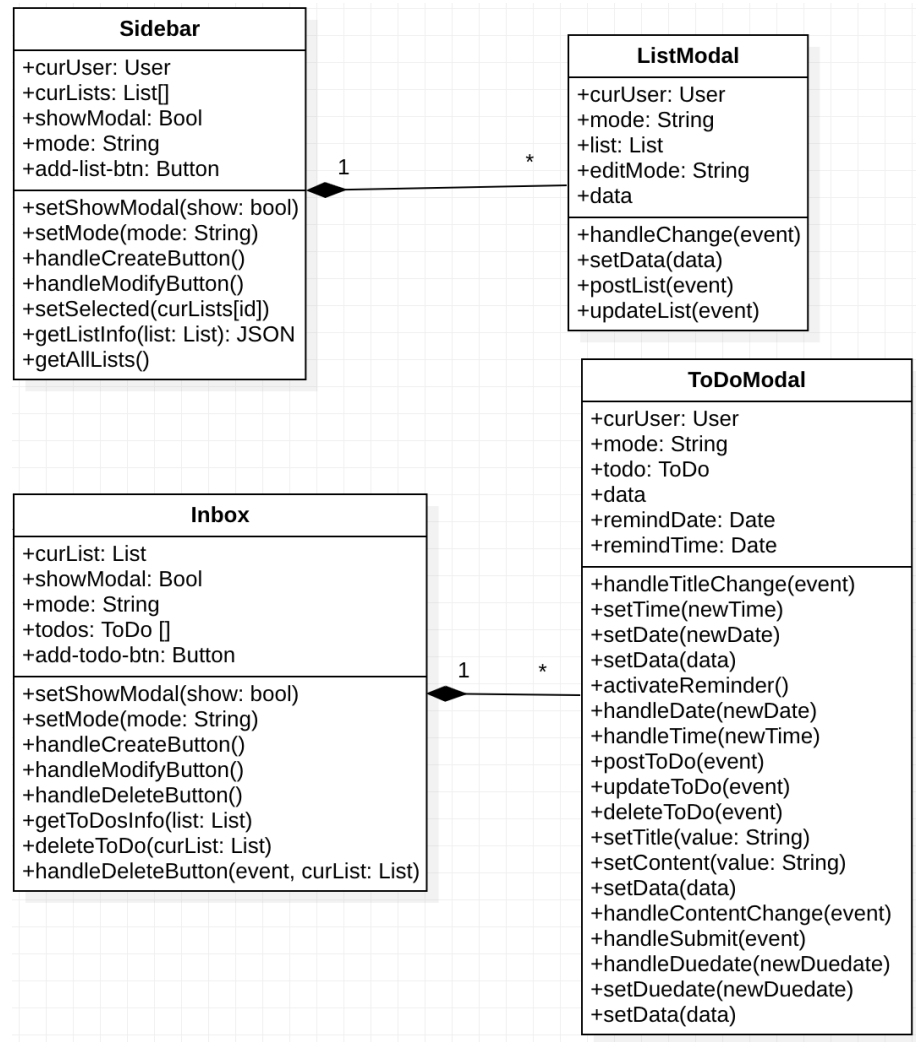
02. 분할된 부분 추출

i. e., Sidebar and Inbox

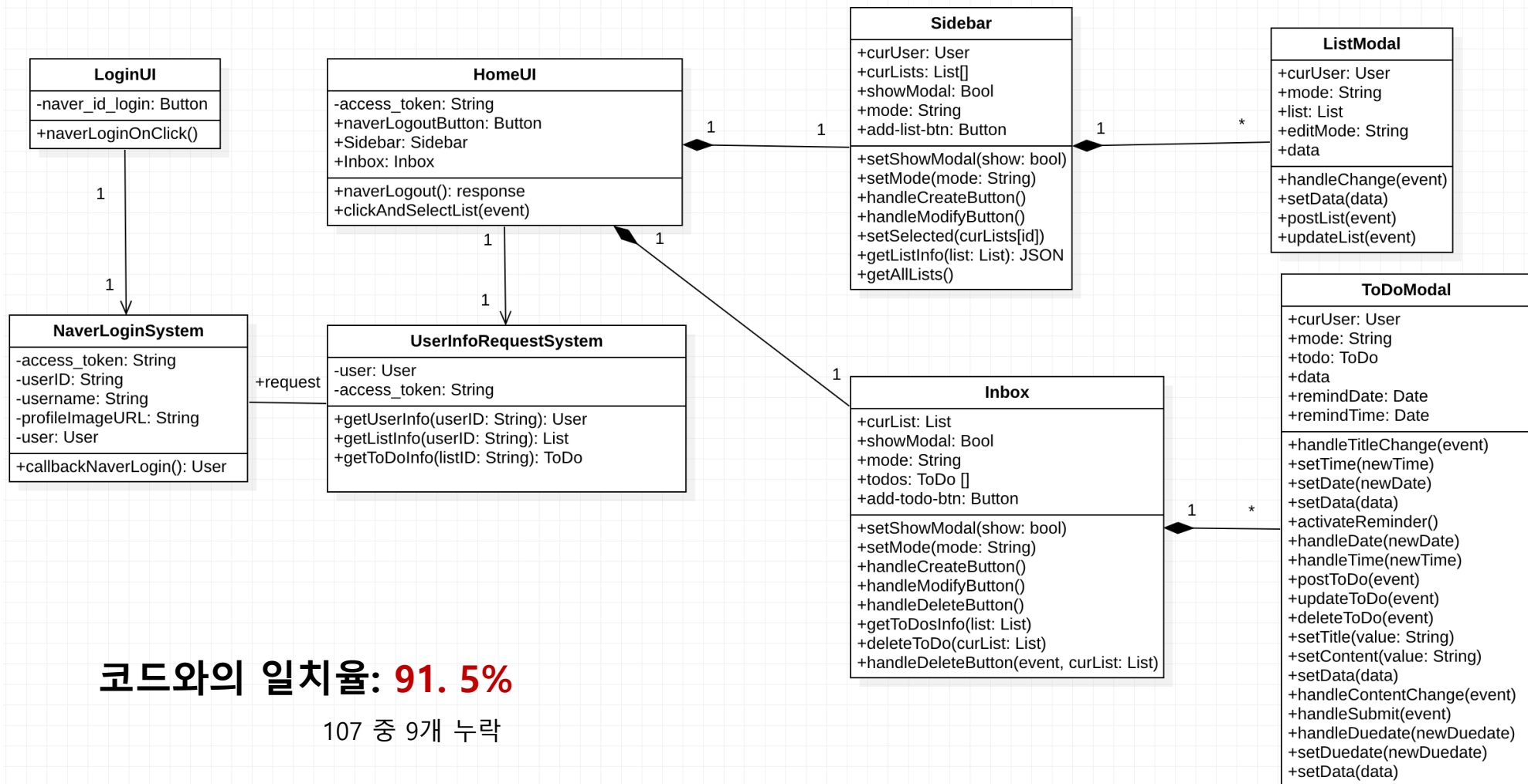
03. Sidebar와 Inbox의 일을 처리해줄 클래스 추출

i. e., ListModal and ToDoModal

04. 프로토타입 및 의 확인을 통해 멤버 추가 Reminders



설계 결과 - 추출



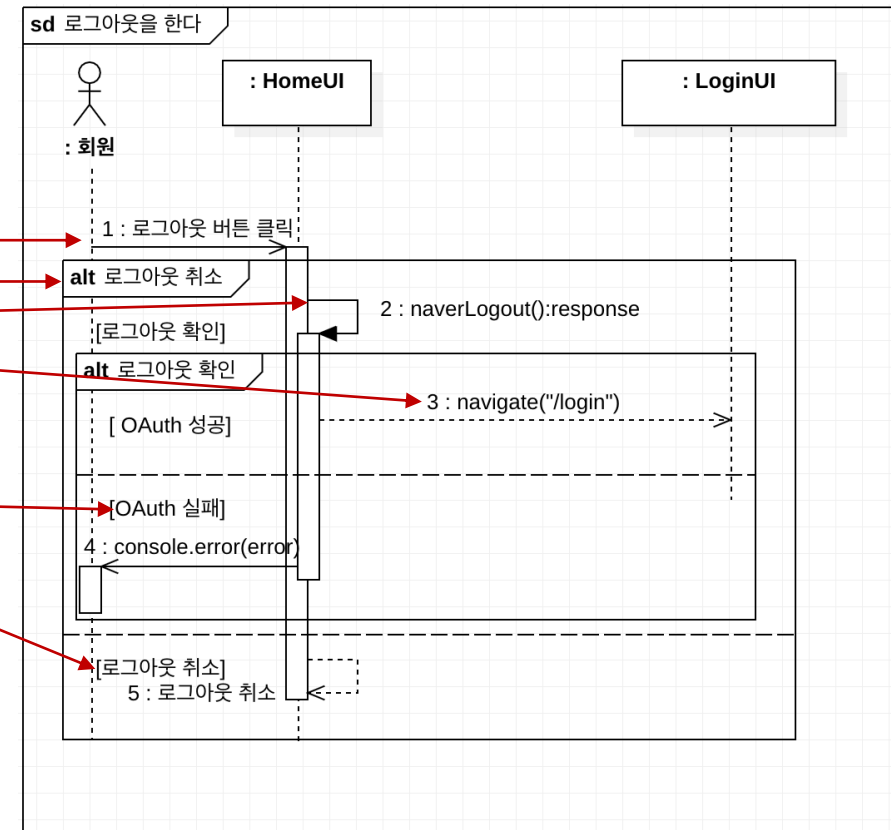
설계 결과 - 시퀀스 다이어그램

01. 작성하고자 하는 유스케이스를 준비

3.2.2.3 UC003: 로그아웃을 한다.

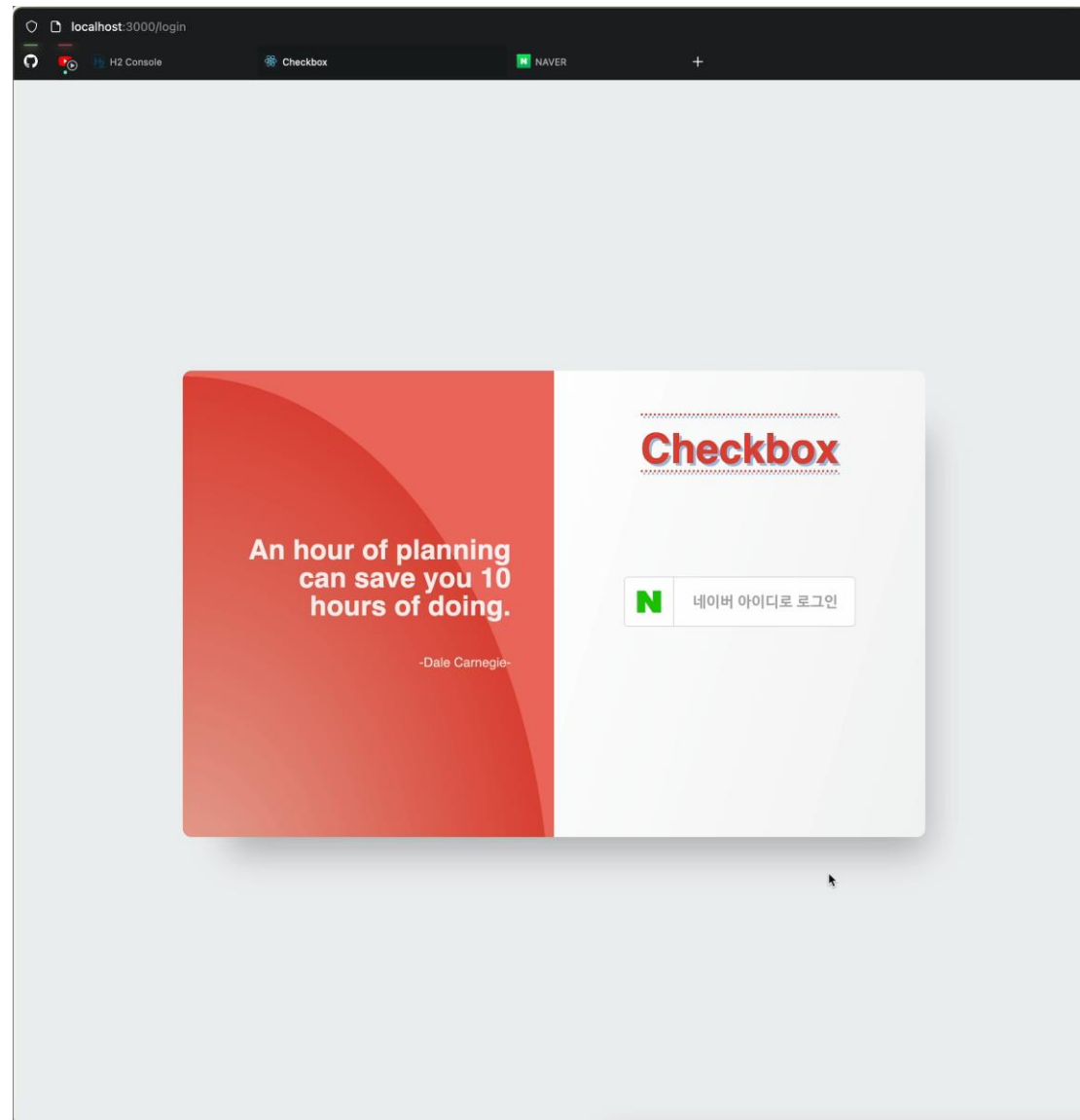
설명	회원이 로그아웃을 한다.
관련 액터	회원, 사용자 인증 시스템
사전 조건	회원은 UC002를 통해 로그인한 상태여야 한다.
사후 조건	시스템은 회원을 사전조건으로 이동시킨다.
기본 흐름	B01 회원이 로그아웃을 한다.
	B02 시스템은 회원에게 로그아웃을 할 것인지 물어본다.
	B03 시스템은 회원을 로그아웃시킨다.
대안 흐름	A01 회원이 로그아웃을 취소한 경우: - 시스템은 회원을 사전 조건으로 되돌린다.
예외 흐름	E01 회원 정보를 불러오는 데에 실패한 경우: -1. 시스템은 회원 정보를 불러오는 것에 실패했다는 메시지를 출력한다. -2. 시스템은 회원을 사전 조건으로 돌아간다.
시나리오	SN001 B01→B02→B03
	SN002 B01→B02→A01
	SN003 B01→B02→E01

02. 유스케이스 시나리오를 옮기기



시연

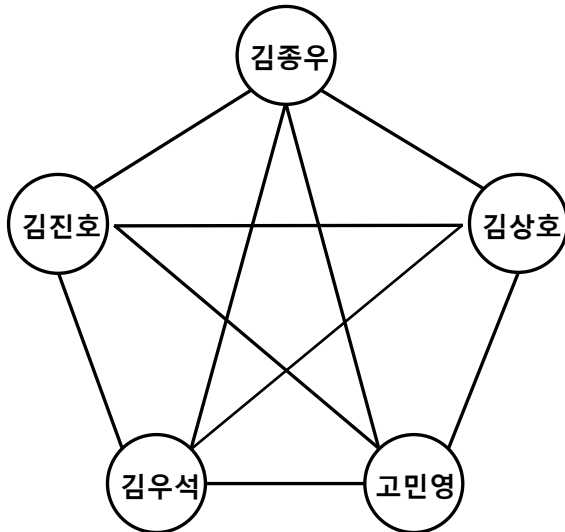
시연



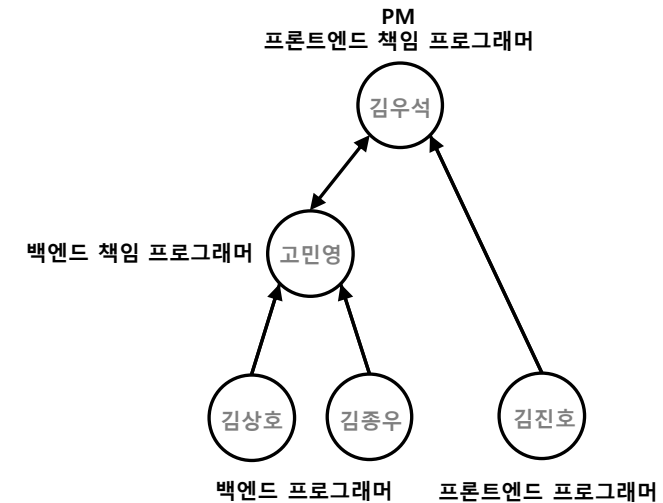
Lessons Learned

팀 구성 민주적 팀 구성의 단점

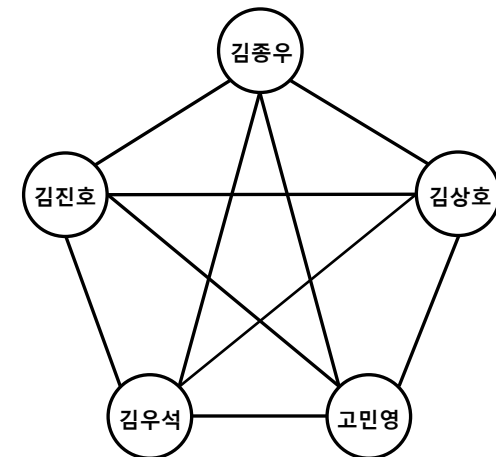
- 프로젝트 초반에 민주적 팀 구성 사용
 - 장점: 더 넓은 관점으로 문제 해결 가능
 - 단점: 의사소통의 폭발 발생



개발 시, 계층적 팀 구성 사용



회의 시, 민주적 팀 구성 사용



김우석 느낀 점

PM 팀원들의 역량에 대한 이해 부족

- 개인별 역량이 모두 다름
- 개개인에 대한 역량에 대한 이해가 부족했음
 - 계획보다 일정이 밀리게 되었음
 - 계획해두었던 파이프라이닝에 실패



PM은 팀원들에 대한 이해도가 높아야 함

김우석 느낀 점

설계 설계서의 역효과

- 팀원 모두가 처음 접해보는 설계서 작성법에 미숙함
- 설계서는 개발을 수월하게 해주는 도구
 - 오히려 역효과 발생
 - 설계에 너무 많은 시간을 사용하게 됨
 - 설계와 개발을 병행하게 됨



충분한 경험을 통해 학습

설계서 작성법에 대한 충분한 숙지 필요

김우석 느낀 점

설계와 구현의 연결

- 구체적인 설계과정이 어려웠음
 - 설계 기반으로 구현을 하고 나니 설계과정의 중요성을 알게 됨
- 팀프로젝트 과정을 통해 나의 역량 부족함을 느낌
- 팀원의 역량을 파악하는 게 중요함을 느낌
- 이러한 과정이 전체적으로 힘들면서도 재밌었음
 - 이번 기회를 통해 방학을 활용해 나의 역량을 끌어올리고 싶은 의지가 커짐

고민영 느낀 점

소프트웨어공학의 첫 적용

- 평소에는 ER 다이어그램, 와이어프레임과 같은 단순한 설계만 진행
- 처음으로 소프트웨어공학을 적용해서 산출물을 만드는 과정이 유익했음

김상호 느낀 점

소프트웨어공학의 첫 적용

- 프로젝트의 전체적인 흐름을 (요구사항 분석, 설계, 구현, 유지보수 등) 그대로 따라가면서 실무적인 흐름들을 읽을 수 있었음
- 설계를 한 부분을 실제로 구현하는 데에 있어 부족함을 느낌
- 학습과 경험의 중요함을 알게 됨
- 협업과 의사소통 능력 향상의 필요성을 느낌

김종우 느낀 점

개발 프로세스의 중요성

- 명세서와 다이어그램을 통해 목표를 명확히 이해하고 구현 방향을 정함으로써 개발 중의 시작 단축과 혼동/오류를 최소화 할 수 있었다.
- 현업에서 개발 능력 못지않게 Document 작성 과정 또한 중요하다고 느꼈다.

김진호 느낀 점

감사합니다

학번	이름	역할
201910939	김우석	조장, PM, 프론트엔드 개발
201910920	고민영	백엔드 개발
201910931	김상호	백엔드 개발
201910941	김종우	백엔드 개발
201910953	김진호	프론트엔드 개발