

## 소프트웨어공학 프로젝트 보고서

< Requirement Analysis & Design - 채용 정보 시스템 >

과목명 : 소프트웨어공학

분반 : 2분반 / 6조

제출일 : 2023년 5월 12일 금요일

팀원 : B711040 김종호

B782004 곽성재

B811160 임탁균

B935452 차민지

# 목차

1. INTRODUCTION	3
2. REQUIREMENT LIST	5
3. USE CASE DIAGRAM	6
4. USE CASE DESCRIPTION	7
5. SEQUENCE DIAGRAM & ANALYSIS CLASS DIAGRAM	11

## 1. Introduction

해당 보고서는 총 5개의 Document ( Requirement list, Use Case Diagram, Use Case Description, Communication Diagram, Analysis Class Diagram)를 종합하여 작성한 과제 2의 최종 보고서이다.

본 과제를 진행하면서 6조는 각 팀원이 3개의 작업을 맡아 진행하였다. 3개의 작업은 (1) 과제 1 작업의 수정, (2) Sequence Diagram 그리기 & Class Diagram 만들기, (3) Main Task (Sequence Diagram, Analysis Class Diagram, UML 작업, Final Report 작업) 로 구분된다. 특히 Sequence Diagram 그리기 작업의 경우, 과제 1에서의 수정 사항을 반영한 결과를 바탕으로 나온 17개의 Use Case를 최대한 System 단위의 일관성을 유지하여 분배하려 노력하였다. 그 결과 각 팀원이 본인이 맡은 Use Case에 대한 Sequence Diagram과 Class Diagram을 만드는 과정에서 높은 이해도를 바탕으로 작업을 수행할 수 있었다.

과제 1 작업의 수정은 과제 1에서 각 팀원이 맡은 역할에 해당하는 작업물을 수정하는 방식으로 진행하였다. 김종호 팀원은 Use Case Description 수정을, 차민지 팀원은 Requirement List 수정을, 곽성재 팀원과 임택균 팀원은 Use Case Diagram을 다시 그리는 작업을 진행하였다.

Use Case의 경우, 과제 1에서는 총 20개의 Use Case가 나왔지만, 본 과제의 경우 17개의 Use Case로 그 개수가 줄어 들었다. 따라서 Use Case를 분배하는 방식이 일부 변경되었다. “로그인, 로그아웃, 회원가입, 회원탈퇴”에 해당하는 Use Case는 ‘LogIn’ subsystem에 포함된다. 해당 파트의 Use Case에 대한 Sequence Diagram과 Class Diagram은 김종호 팀원이 전담하였다. “채용 정보 등록, 정보 수정, 정보 삭제, 채용 마감 처리, 채용 정보 리스트 조회”에 해당하는 Use Case는 ‘Recruitment\_Posting’ subsystem에 포함된다. 해당 파트의 Use Case에 대한 Sequence Diagram과 Class Diagram은 곽성재 팀원이 전담하였다. “지원 정보 조회, 지원 취소, 지원 정보 삭제, 채용 상세 페이지, 지원하기”에 해당하는 Use Case는 ‘Recruitment\_Apply’ subsystem에 포함된다. 해당 파트의 Use Case에 대한 Sequence Diagram과 Class Diagram은 임택균 팀원이 전담하였다. “지원 정보 통계, 마감 내역 조회, 채용 정보 검색”에 해당하는 Use Case는 나머지에 해당된다. Use Case의 공평한 분담을 위해 각각의 substem에서 하나씩의 Use Case를 차출하였다. 해당 파트의 Use Case에 대한 Sequence Diagram과 Class Diagram은 차민지 팀원이 전담하였다.

Main Task는 각 팀원이 과제 1에서 맡은 역할의 성격을 바탕으로 크게 4개의 파트로 구분하여 진행하였다. **Sequence Diagram 파트**의 역할은 각 팀원이 작성한 Sequence Diagram을 수합하고 전체 Sequence Diagram을 만들어 UML 파트를 맡은 팀원에게 전달하는 역할을 수행하였다. 해당 역할은 김종호 팀원이 전담하였다. **Analysis Class Diagram 파트**의 역할은 각 팀원이 작성한 Class Diagram을 수합하여 Analysis Class Diagram을 만들어 UML 파트를 맡은 팀원에게 전달하는 역할을 수행하였다. 해당 역할은 차민지 팀원이 전담하였다. **UML 파트**의 역할은 전달받은 Sequence Diagram과 Analysis Class Diagram을 Star UML 툴을 사용하여 만드는 작업을 수행하였다. 해당 역할은 임택균 팀원이 전담하였다. 마지막으로

**Final Report 파트**는 최종적으로 완성된 결과물(Requirement List, Use Case Diagram, Use Case Description, UML(Sequence Diagram & Analysis Class Diagram))을 수합하여 내용을 점검한 후 최종 Document를 작성하는 역할을 수행하였다. 해당 역할은 곽성재 팀원이 전담하였다.

각 팀원이 맡은 역할을 정리하면 다음과 같다.

**김종호** – Use Case Description / Sequence Diagram & Class Diagram : 로그인, 로그아웃, 회원가입, 회원탈퇴 / Main Task : Sequence Diagram

**곽성재** – Use Case Diagram / Sequence Diagram & Class Diagram : 채용 정보 등록, 정보 수정, 정보 삭제, 채용 마감 처리, 채용 정보 리스트 조회 / Main Task : Final Report

**임탁균** – Use Case Diagram / Sequence Diagram & Class Diagram : 지원 정보 조회, 지원 취소, 지원 정보 삭제, 채용 상세 페이지, 지원하기 / Main Task : Star UML

**차민지** – Requirement list / Sequence Diagram & Class Diagram : 지원 정보 통계, 마감 내역 조회, 채용 정보 검색 / Main Task : Analysis Class Diagram

Drawing 작업은 **StarUML v5.1.0**을 사용하여 진행하였다. 보고서 작성에 사용된 각각의 파일들은 **Data 폴더**에서 확인할 수 있다.

대면 회의는 홍익대학교 A동 카페(카페드림)에서 진행하였고, 비대면 회의는 구글미트를 활용하여 진행하였다. 회의 내용 정리 및 전반적인 의견 수렴은 노션페이지와 카카오톡을 활용하였다.

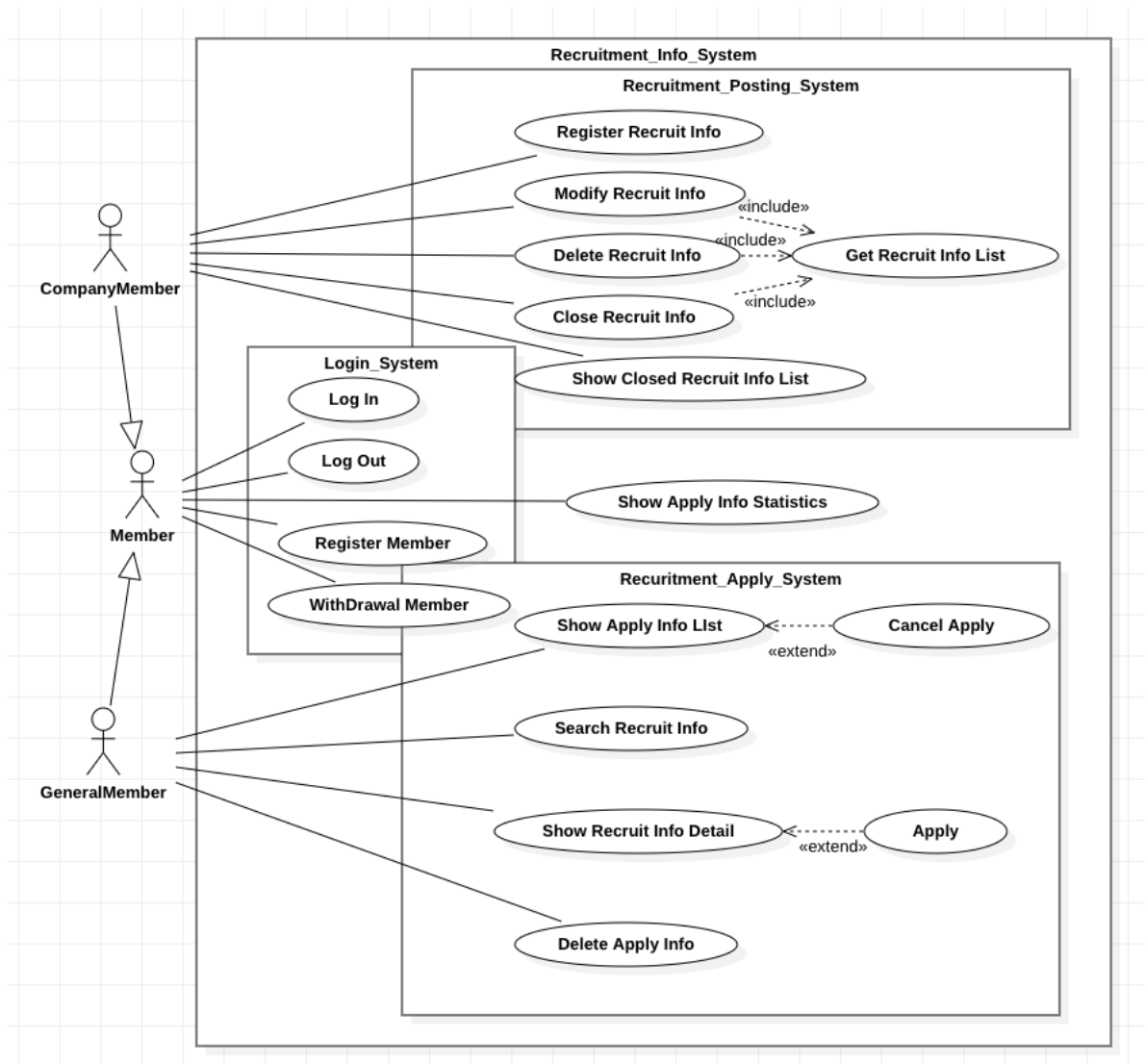
노션 페이지 링크 : <https://quasar-lumber-222.notion.site/2-6e6d5a928ba642bbb5adb9c21c9dfd8d>

## 2. Requirement list

No.	Requirement	Use case
1	사용자의 회원 가입을 진행한다. 사용자의 기본정보 (회사 회원: 회사이름, 사업자번호, 사업분야, 주소, 전화번호 / 일반 회원: 이름, 주민번호, 주소, 전화번호)와 id/pw를 입력 받고 화면에 회원가입 완료창을 표시한다.	회원 가입
2	회원이 탈퇴 요청을 하면 탈퇴 처리한다.	회원 탈퇴
3	회원이 id, pw를 입력하면 로그인을 진행한다.	로그인
4	회원이 로그아웃 버튼을 누르면 로그아웃을 진행한다.	로그아웃
5	회사 회원에게 업무, 인원 수, 신청 마감일 등의 정보를 입력 받아 채용 정보를 등록한다.	채용 정보 등록
6	회사 회원에게 자신이 등록한 채용 정보 리스트를 제공한다.	채용 정보 리스트 조회
7	회사 회원이 자신이 등록한 채용 정보를 선택한 후 수정 사항을 입력하면 업데이트된 채용 정보를 화면에 표시한다.	채용 정보 수정
8	회사 회원이 자신이 등록한 채용 정보를 선택한 후 삭제 버튼을 누르면 해당 채용 정보를 삭제한다.	채용 정보 삭제
9	회사 회원이 등록한 채용 정보가 원하는 지원자 수에 도달한 경우, 회원에게 마감 의사를 입력 받아 조기 마감 처리한다.	채용 마감 처리
10	회사 회원에게 마감 상태인 채용 정보 내역(업무, 신청 마감일)을 마감일의 오름차순으로 정렬하여 제공한다.	마감 내역 조회
11	일반 회원에게 회사 이름 또는 업무 정보를 입력 받아 알맞은 채용 정보 리스트(회사 이름, 마감일 정보)를 회사이름의 오름차순으로 정렬하여 제공한다.	채용 정보 검색
12	일반 회원이 선택한 채용 정보의 상세한 정보(회사 이름, 업무, 인원수, 신청 마감일)를 제공한다.	채용 상세 페이지
13	일반 회원이 상세 페이지에서 지원 버튼을 누르면 즉시 지원 처리한다.	지원하기
14	일반 회원에게 자신의 지원 정보(회사이름, 업무, 인원수, 신청 마감일)를 회사이름으로 오름차순 정렬하여 제공한다.	지원 정보 조회
15	일반 회원이 마감일이 지나지 않은 지원 정보를 선택하여 지원 취소 버튼을 누르면 지원을 취소한다.	지원 취소
16	일반 회원이 마감 시간이 지난 지원 정보의 삭제 버튼을 누르면 해당 지원 정보를 삭제한다.	지원 정보 삭제
17	각 회원에게 통계 정보(회사 회원: 현재까지 등록한 채용 정보에 대해 마감시간 기준으로 업무별 지원자 수, 일반 회원: 본인이 지원한 정보에 대해 마감시간 기준으로 월별 지원 횟수)를 제공한다.	지원 정보 통계

기존 과제 1에서의 Use Case 20개에 대한 Requirement list에 대해 새로운 Requirement 내용을 반영하여 총 17개의 Use Case에 대한 Requirement list로 수정하였다. 본 문서는 가독성의 차원에서 한글로 Use Case의 이름을 기입하였지만, 이후 작업들은 Requirement list의 Use Case 이름을 영문으로 변환하여 진행하였다. Requirement list는 Functional Requirement에 포함해야 하는 'Process, Inputs, Outputs' 을 중심으로 작성하였다.

### 3. Use Case Diagram



< fig 1. Use Case Diagram >

Use Case Diagram은 위의 이미지와 같이 작성하였다.

'Member, CompanyMember, GeneralMember' 의 총 3개의 Actor를 설정하였다. Use Case는 Requirement list에서 작성한 내용을 바탕으로 17개의 Use Case를 설정하였다. System은 Supersystem인 'Requirement\_Info\_System'과 3개의 Subsystem 'Login, Recruitment\_Posting, Recruitment\_Apply' 로 설정하

였다. 위의 이미지는 StarUML을 사용하여 만들었다.

#### 4. Use Case Description

##### Use case : 회원 가입

Actor Action	System Response
1. None	2. 회사 회원, 일반 회원 선택 버튼을 표시한다.
3. 회원 종류를 선택한다.	4. 기본 정보 회사 회원: 회사이름, 사업자번호, 사업분야, 주소, 전화번호 등 일반 회원: 이름, 주민번호, 주소, 전화번호 등 전체 : ID/Password 입력 창을 표시한다.
5. 자신의 정보를 입력한뒤 가입 버튼을 누른다.	6. 회원 가입 완료 창을 표시한다.

##### Use case : 회원 탈퇴

Actor Action	System Response
1. None	2. 탈퇴 의사를 확인한다.
3. 탈퇴 의사를 표시한다.	4. 탈퇴 완료 창을 표시한다.

##### Use case : 로그인

Actor Action	System Response
1. None	2. 로그인 폼을 제공한다.
3. Actor가 자신의 ID,Password 를 입력한다.	4. 회원 정보에 맞는 화면을 제공한다.

##### Use case : 로그아웃

Actor Action	System Response
1. None	2. 로그아웃 확인 창을 표시한다.
3. Actor가 로그아웃 의사를 표시한다.	4. 로그아웃 처리된 화면을 표시한다.

### Use case : 채용 정보 등록

Actor Action	System Response
1. None	2. 채용 정보 등록 폼을 제공한다.
3. Actor가 업무, 인원 수, 신청 마감일등의 정보를 입력한다.	4. 등록된 채용 정보를 보여준다.

### Use case : 채용 정보 리스트 조회

Actor Action	System Response
1. None	2. Actor 가 등록한 채용 정보 리스트를 보여준다.
Extensions	Step 2 이후, Actor 는 채용 정보를 수정할 수 있다.
	Step 2 이후, Actor 는 채용 정보를 삭제할 수 있다.
	Step 2 이후, Actor 는 채용 정보를 마감 처리할 수 있다.

### Use case : 채용 정보 수정

Actor Action	System Response
1. None	2. Actor 가 등록한 채용 정보 리스트를 보여준다.
3. 수정할 채용 정보를 선택한다.	4. 선택한 채용 정보의 각 항목을 보여준다.
5. 수정할 정보를 입력 후 저장한다.	6. 수정된 채용 정보를 보여준다.

### Use case : 채용 정보 삭제

Actor Action	System Response
1. None	2. Actor 가 등록한 채용 정보 리스트를 보여준다.
3. 삭제할 채용 정보를 선택한다.	4. 삭제 여부를 확인 한다.
5. 삭제 여부를 선택한다.	6. 수정된 채용 정보 리스트를 보여준다.



### Use case : 채용 마감 처리

Actor Action	System Response
1. None	2. Actor 가 등록한 채용 정보 리스트를 보여준다.
3. 마감할 채용 정보를 선택한다.	4. 마감 확인 창을 표시한다.
5. 마감 여부를 선택한다.	6. 수정된 채용 정보 리스트를 보여준다.

### Use case : 마감 내역 조회

Actor Action	System Response
1. None	2. 마감된 채용 정보 리스트를 보여준다.

### Use case : 채용 정보 조회

Actor Action	System Response
1. None	2. 검색 창과 검색 옵션(회사이름 또는 업무)를 입력할 수 있는 창을 제공한다.
3. Actor 는 회사이름 혹은 업무 중 하나를 선택하여 검색한다.	4. 마감이 지나지 않은 채용 정보를 회사이름 오름차순으로 출력한다.
Extensions	Step 4 이후, 일반 회원은 선택한 채용 정보의 상세 정보를 출력할 수 있다.

### Use case : 채용 상세 페이지 제공

Actor Action	System Response
1. None	2. 채용의 상세한 정보(회사이름, 업무, 인원수, 신청 마감일) 을 출력한다.
Extensions	Step 2 이후, 일반 회원은 즉시 지원할 수 있다.

### Use case : 지원하기

Actor Action	System Response
1. None	2. 지원 확인 메시지를 출력한다.
3. 지원 여부를 선택한다.	4. 지원한 정보를 출력한다.

### Use case : 지원 정보 조회

Actor Action	System Response
1. None	2. 지원 정보 리스트를 제공한다.
Extensions	Step 2 이후 일반 회원은 지원을 취소할 수 있다.

### Use case : 지원 취소

Actor Action	System Response
1. None	2. 취소 여부를 확인한다.
3. 취소 여부를 선택한다.	4. 취소 완료 확인 창을 표시한다.

### Use case : 지원 정보 삭제

Actor Action	System Response
1. None	2. 삭제 여부를 확인한다.
3. 삭제 여부를 선택한다.	4. 삭제 완료 창을 표시한다.

### Use case : 지원 정보 통계

Actor Action	System Response
1. None	2. 각 회원에게 통계 정보를 제공한다. (회사 회원: 현재까지 등록한 채용 정보에 대해 마감시간 기준으로 업무별 지원자 수 일반 회원: 본인이 지원한 정보에 대해 마감시간 기준으로 월별,지원 횟수)

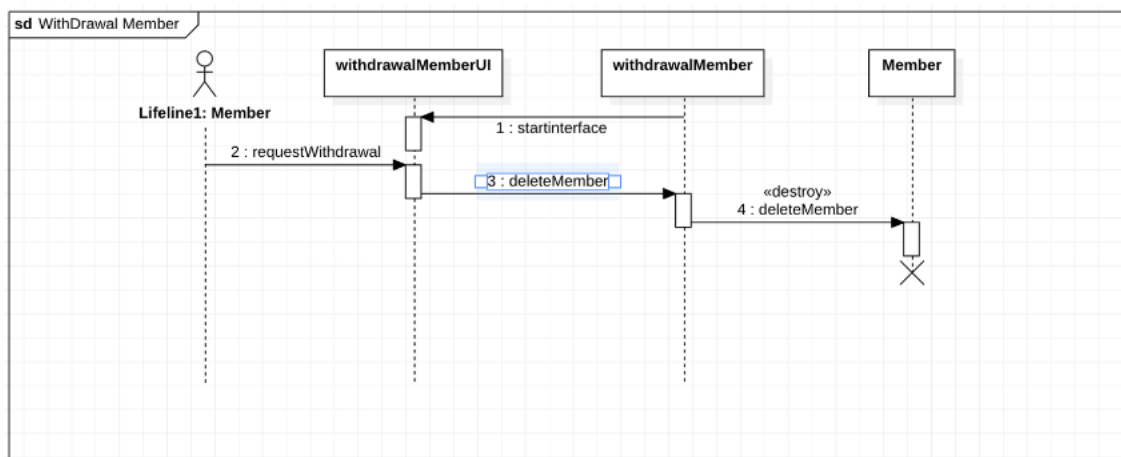
Use Case Description은 위와 같이 Step by Step Breakdown 방식으로 작성하였다.

Requirement list와 마찬가지로 기존의 형태를 유지하되, 새로운 명세서의 내용에 맞게 내용을 수정하여 총 17개의 Use Case에 대한 Description을 작성하였다. Requirement list의 경우와 마찬가지로 Use Case의 이름을 한글로 표기하였다.

## 5. Sequence Diagram & Analysis Class Diagram

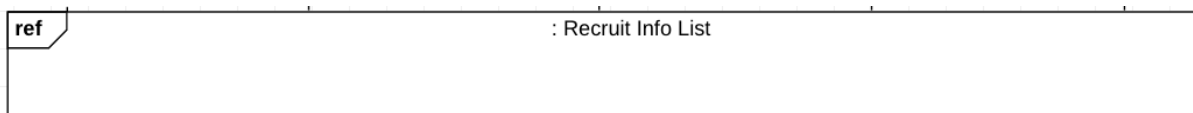
Sequence Diagram에 대한 이야기를 하기에 앞서 강의록의 이미지와 Star UML에서의 구현의 차이에 대한 이야기를 먼저 언급하겠다.

1. Actor의 굵은 바 또는 활성화를 의미하는 사각형(Message를 받기전에도 활성화 상태를 명시할 수 있는)을 Star UML에서는 지원하지 않는 것으로 보인다. 따라서 Star UML 작업에서 Actor 하단의 굵은 바는 life line으로 표기하였다.
2. StarUML 상에서 특수문자 시작을 막아놓아, 강의록과는 달리 Actor를 Actor : Member 형식으로 표기하였다. 마찬가지로 Boundary, Control, Entity Class도 :Client와 같은 형식이 아닌 Client 의 형식으로 표기하였다.
3. delete관련하여 소멸자 표기를 ~Member 와 같은 형식으로 표현하려고 하였다. 하지만 StarUML에서 특수문자로의 시작을 막아 놓았고, Member:~Member와 같은 형식으로도 진행해 보았지만 똑같이 입력 제한 현상이 발생하였다. 이와 같은 점을 고려하여 소멸자를 deleteMember와 같이 작성하였다.



< fig 2. 본 과제에서 사용한 소멸자 양식 >

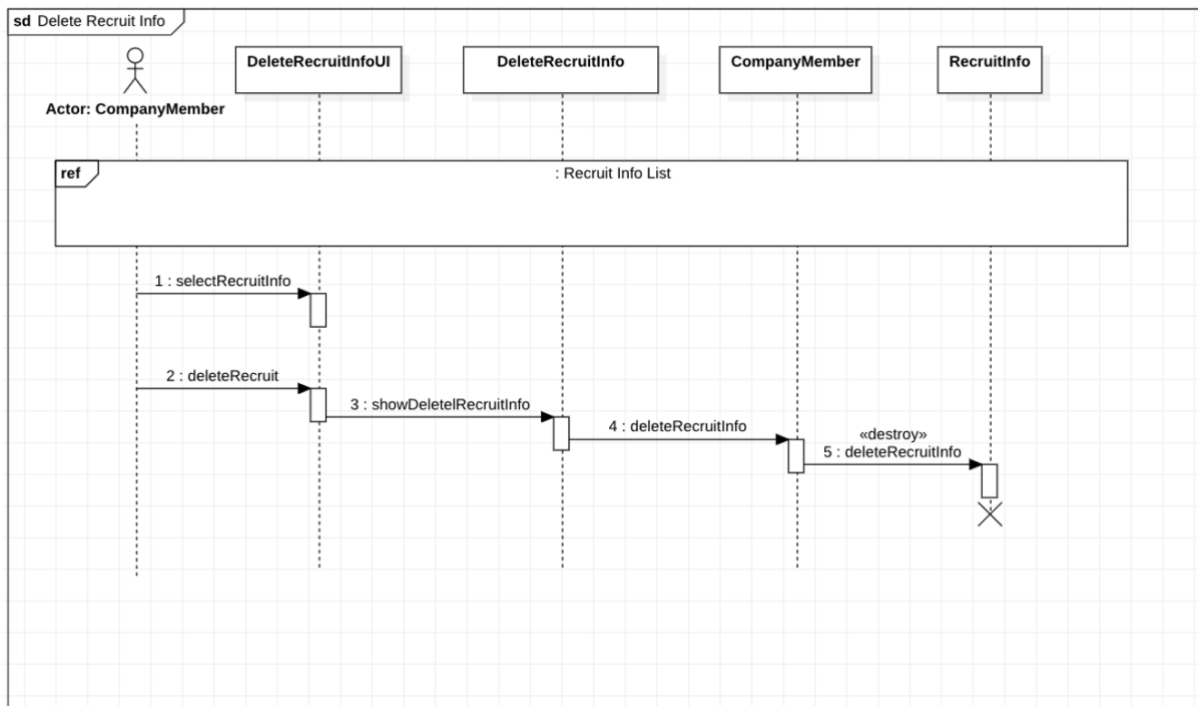
4. ref 의 경우 강의록은 Use Case의 이름을 적지만, Star UML은 다음의 형태로 리턴 값이 표기된다.



< fig 3. Ref 관련 이미지 >

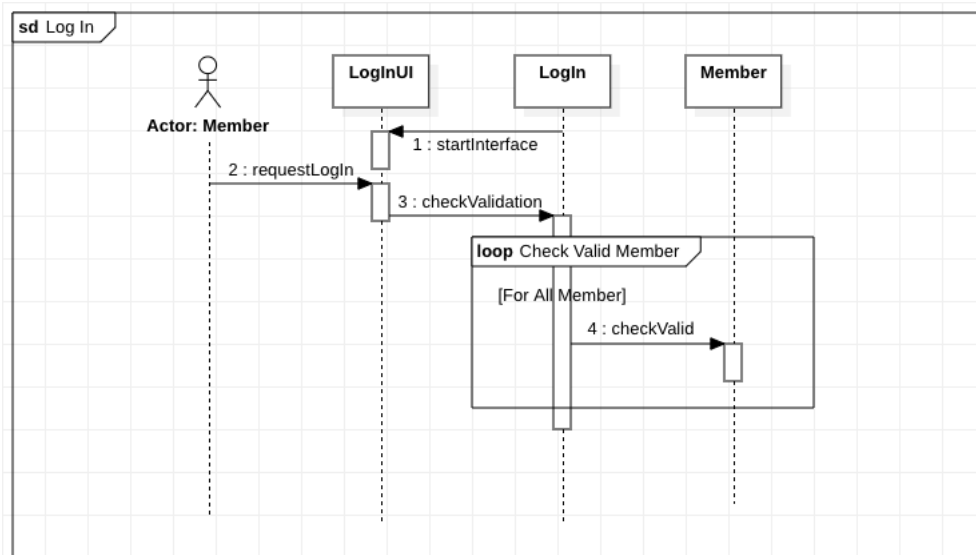
본 과제에서는 Star UML의 기능을 바탕으로 진행했기 때문에, Use Case의 이름이 아닌 리턴 값을 표기하는 방식으로 진행하였다.

5. 몇몇 Use Case의 경우 Figure 4. 와 같이 UI의 select함수를 호출하고 이후 작업을 바로 진행하는 로직으로 구성하였다. 이는 해당 Use Case들의 경우 첫 단계에서의 selec 함수는 단순히 체크 표시를 화면에 출력하는 작업을 수행하는 것이고, 이를 UI 단계에서 해결하도록 구성하였기 때문이다.



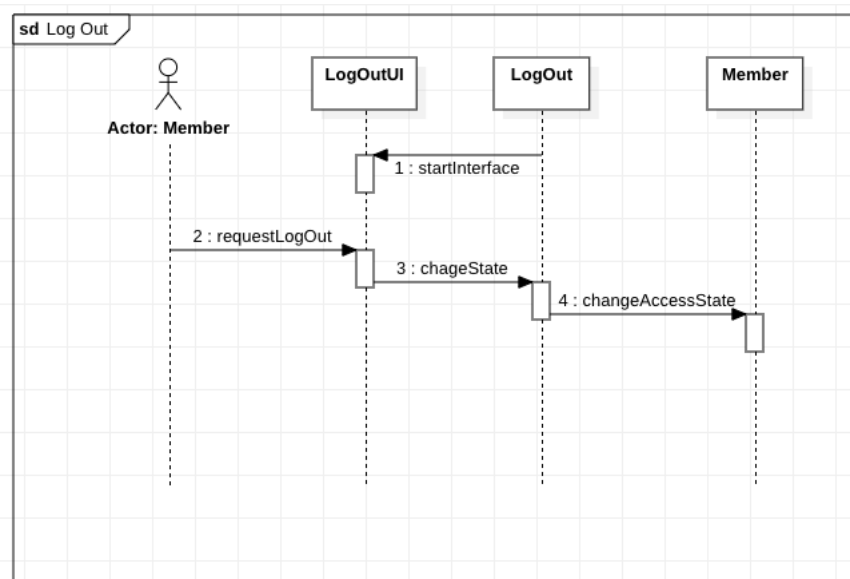
< fig 4. Select이후 바로 다음 action이 이어지는 Use Case의 예시 >

이후의 내용은 Sequence Diagram의 이미지를 나열하는 형식으로 진행된다. 설명이 필요한 지점이 있는 Sequence Diagram에 대해서만 설명을 기입하였다.



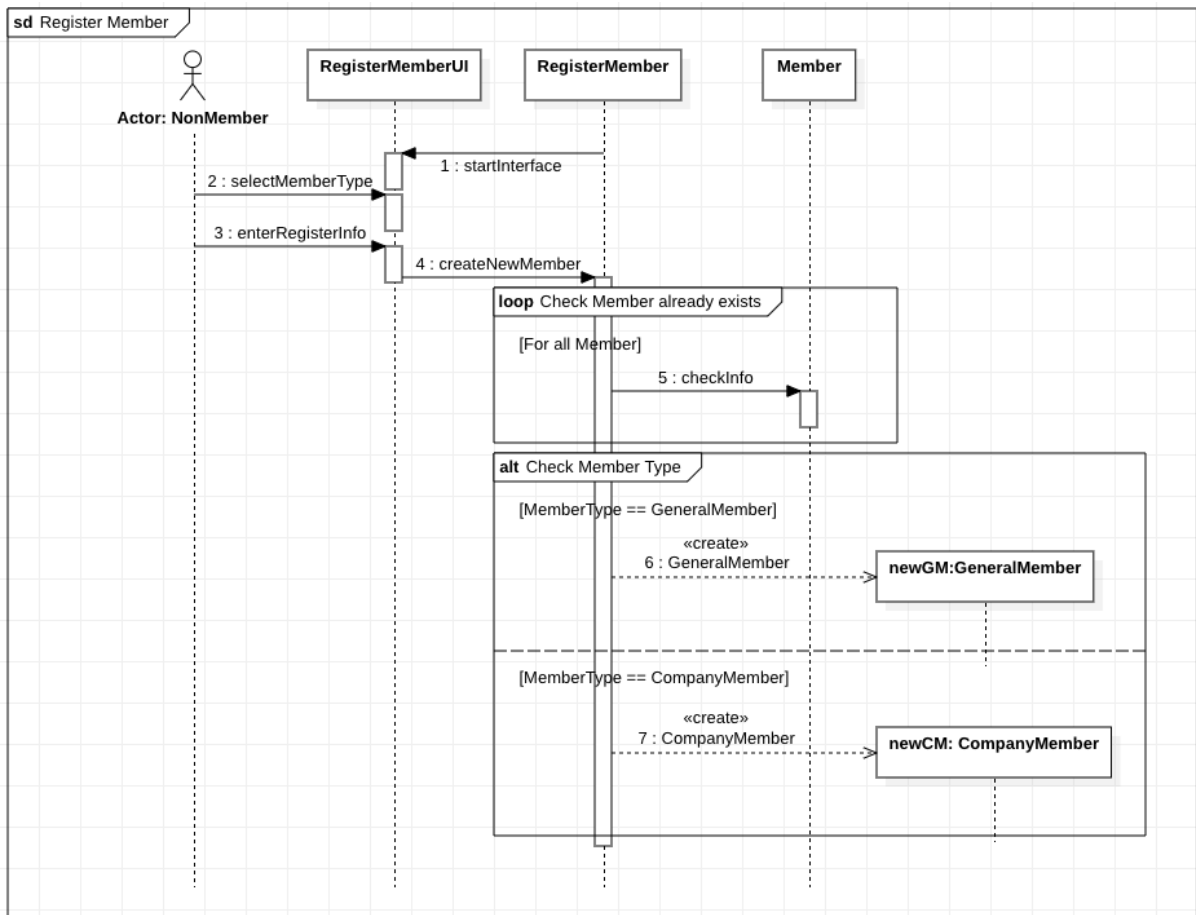
< fig 5. Use Case : Log In >

Log In의 경우 checkValid 함수를 호출하여 Member의 state 를 변경하는 방식으로 로그인 로직을 구성하였다. 이에 Member의 attribute에 state 를 추가하였다.



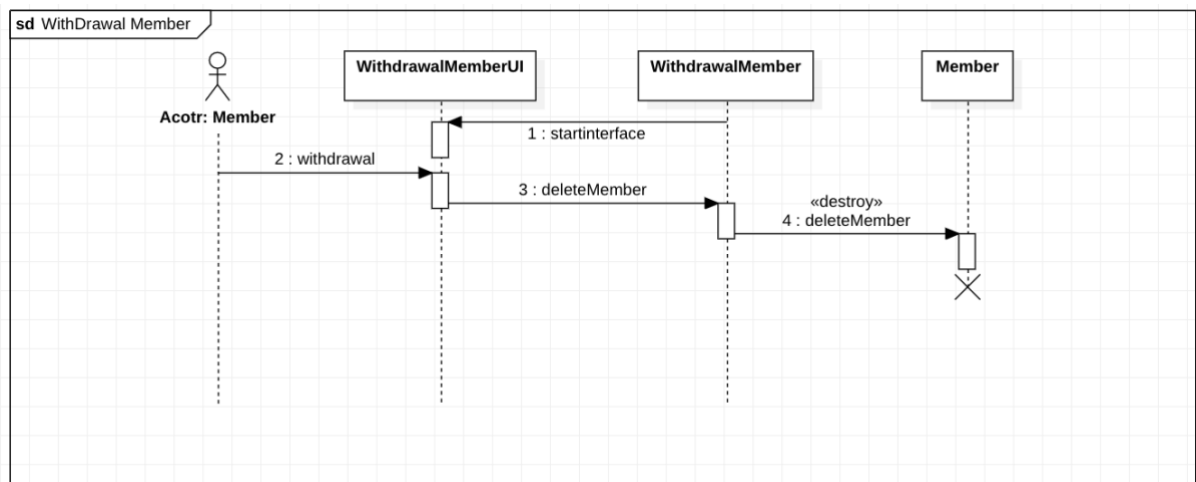
< fig 6. Use Case : Log Out >

Log In과 마찬가지로 Member의 state를 변경하여 로그아웃을 진행하는 로직으로 구성하였다.

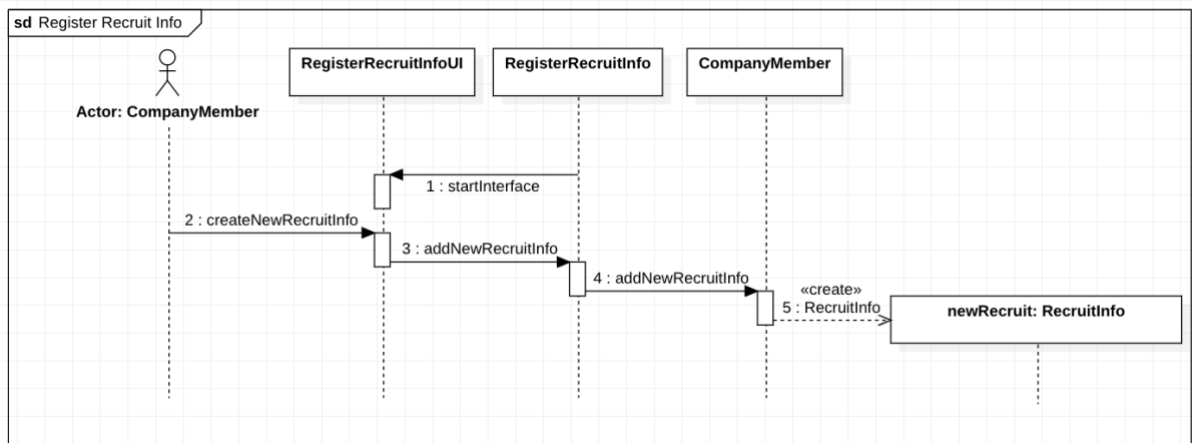


< fig 7. Use Case : Register Member >

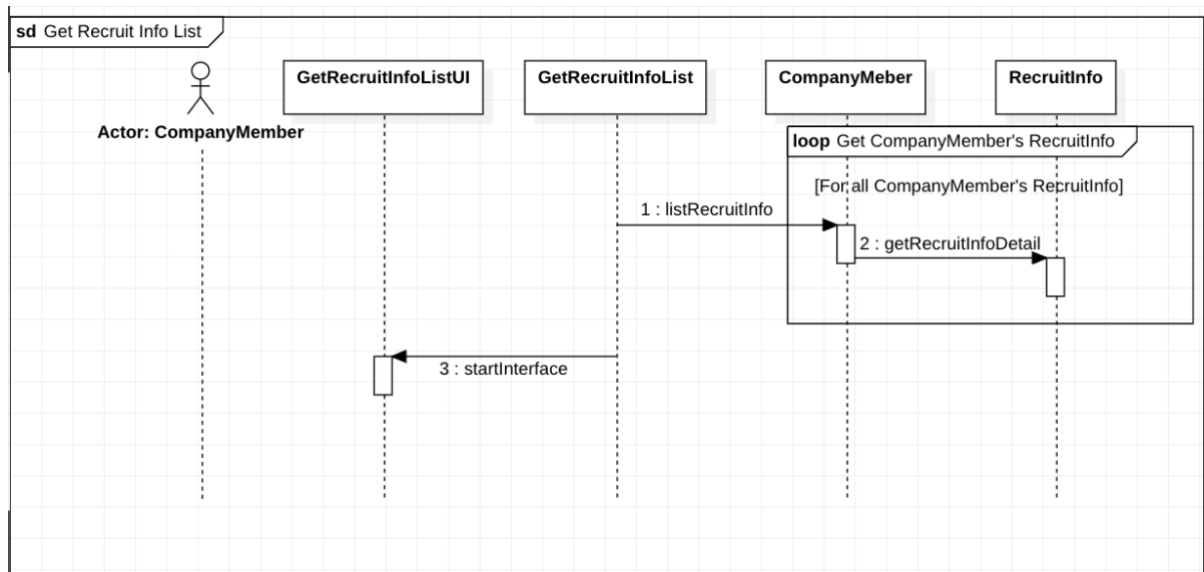
컨트롤이 Member의 checkInfo함수를 호출하면 계정정보가 일치하는지 확인하고, 만약 일치 하는 정보가 없다면 checkInfo함수 내에서 newRequest를 만드는 방식으로 로직을 구성하였다.



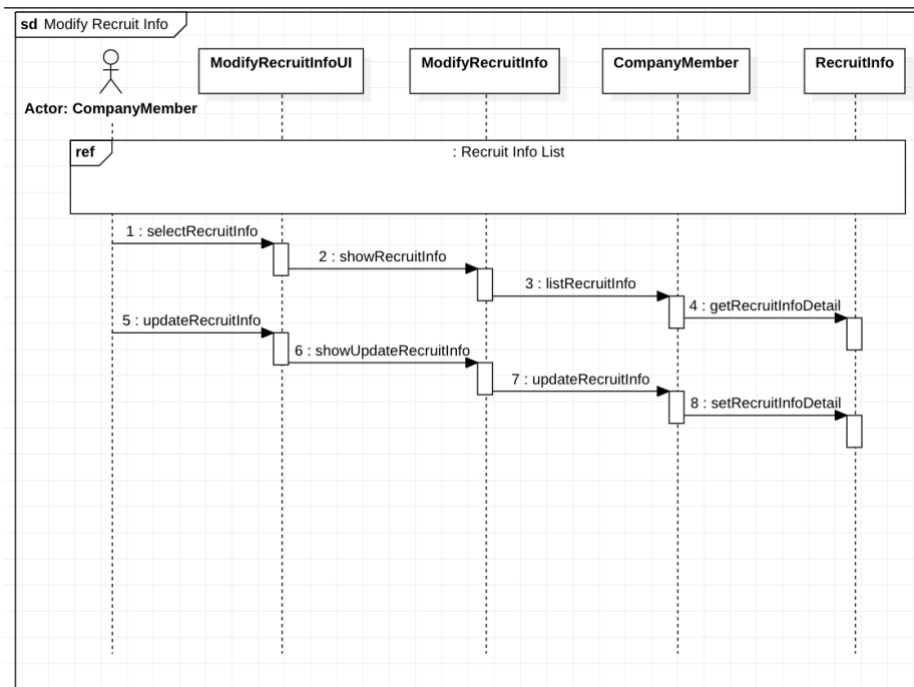
< fig 8. Use Case : Withdrawal Member >



< fig 9. Use Case : Register Recruit Info >

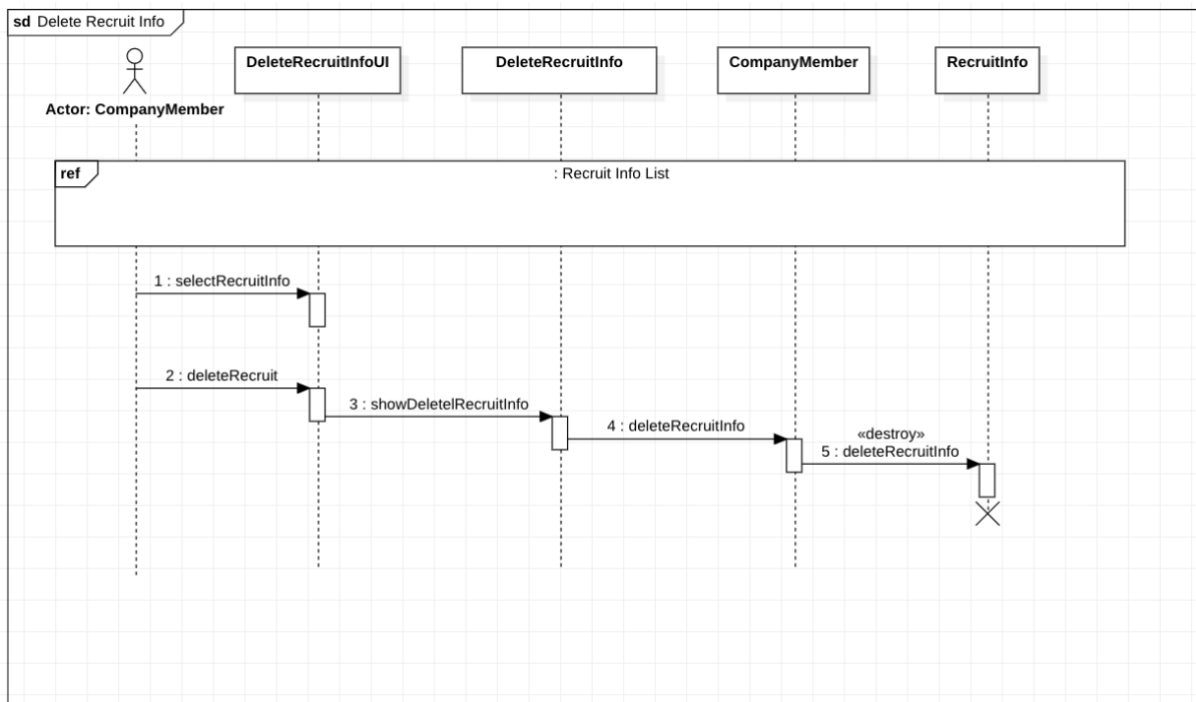


< fig 10. Use Case : Get Recruit Info List >



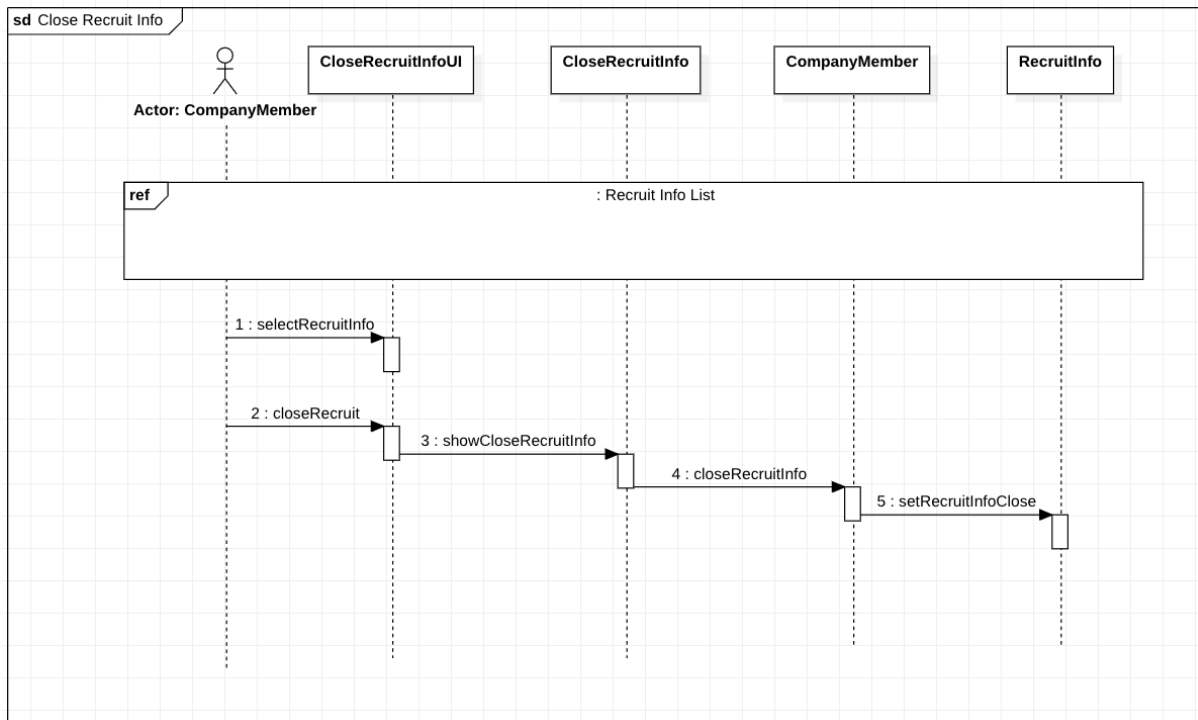
< fig 11. Use Case : Modify Recruit Info >

Ref 의 경우 Use Case : Get Recruit Info List 를 의미하지만, Star UML의 특성을 따라 리턴값인 Recruit Info List로 표기하였다. 이후 Sequence Diagram에서도 위의 규칙을 준수하고 있다.

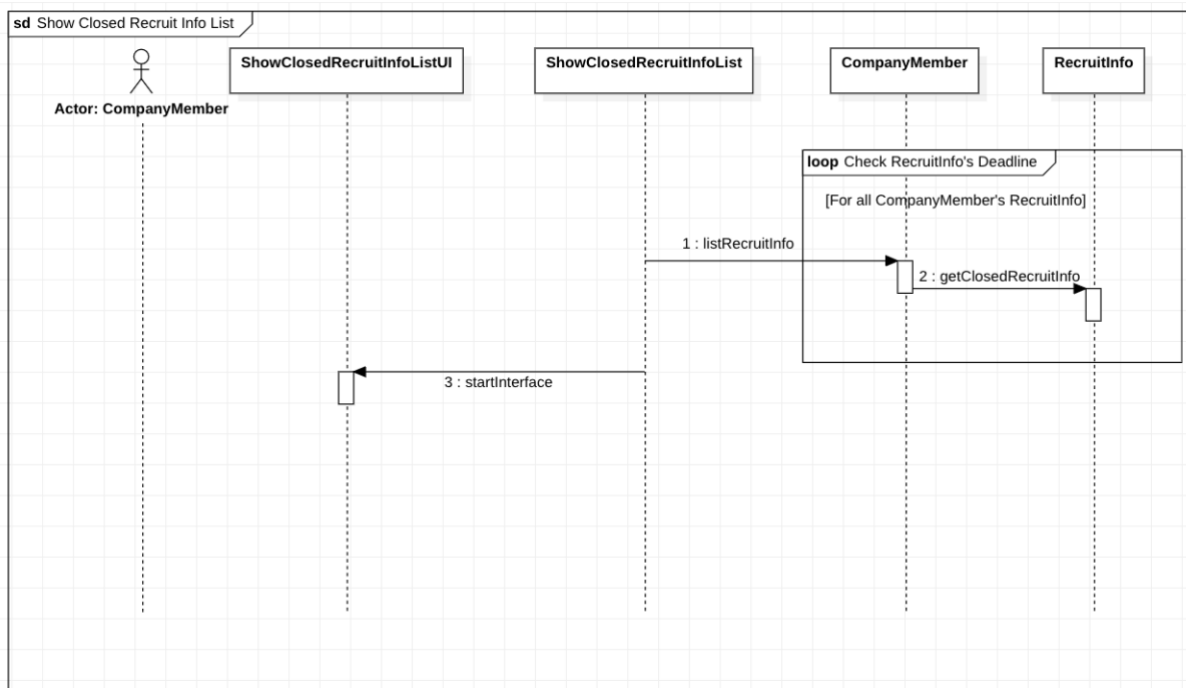


< fig 12. Use Case : Delete Recruit Info >

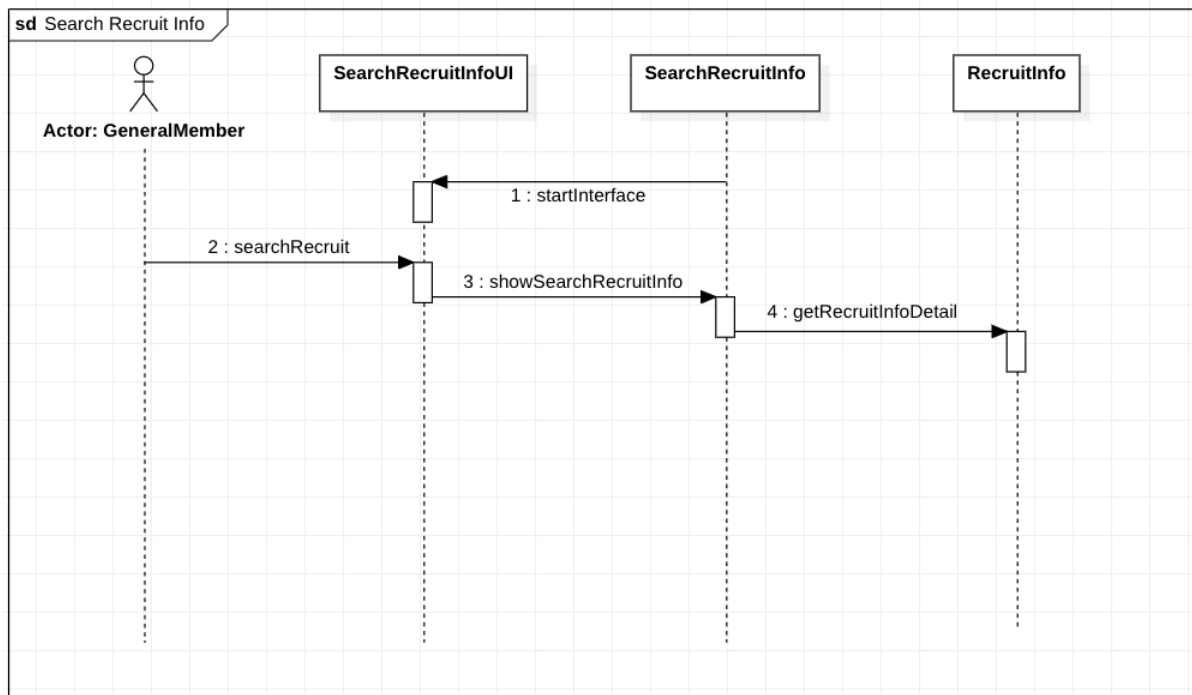




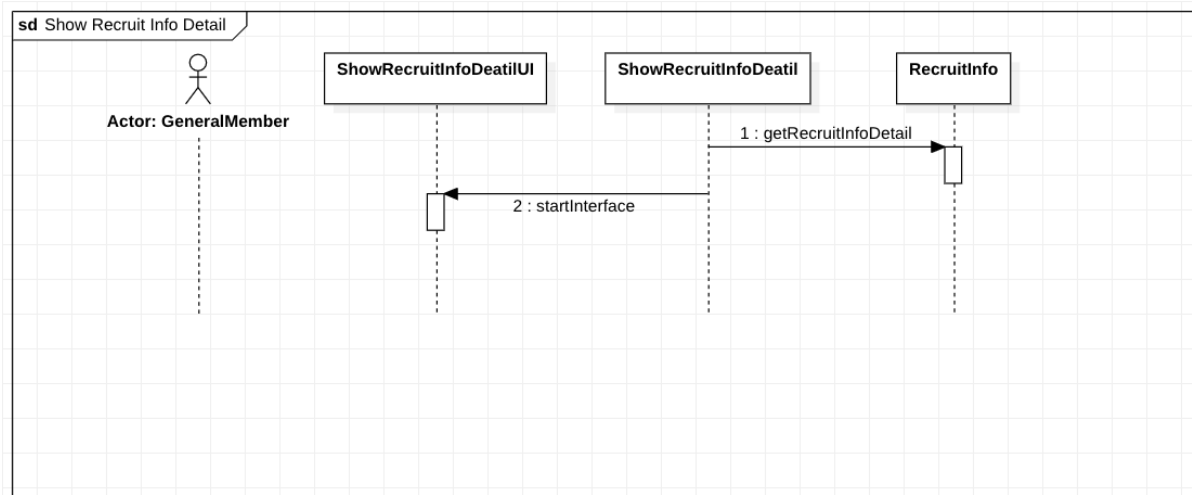
< fig 13. Use Case : Close Recruit Info >



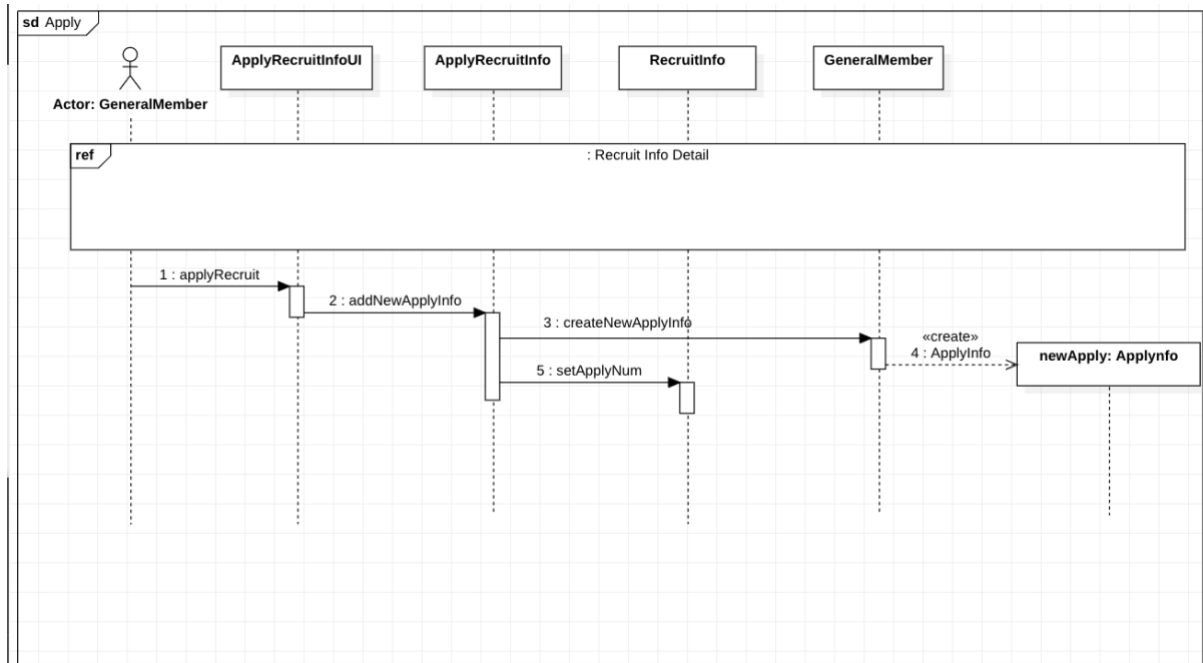
< fig 14. Use Case : Show Closed Recruit Info List >



< fig 15. Use Case : Search Recruit Info >

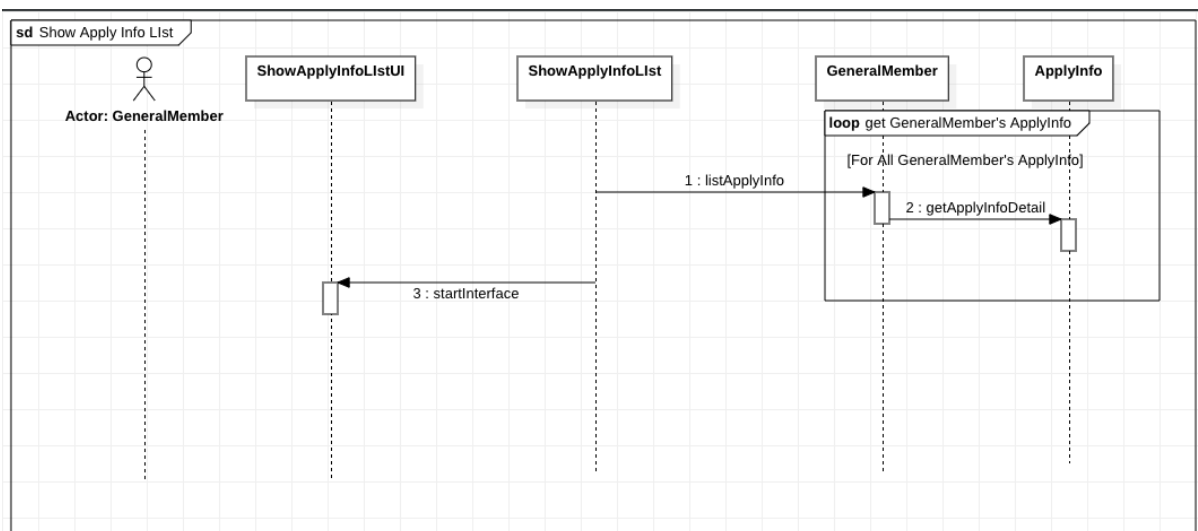


< fig 16. Use Case : Show Recruit Info Detail >

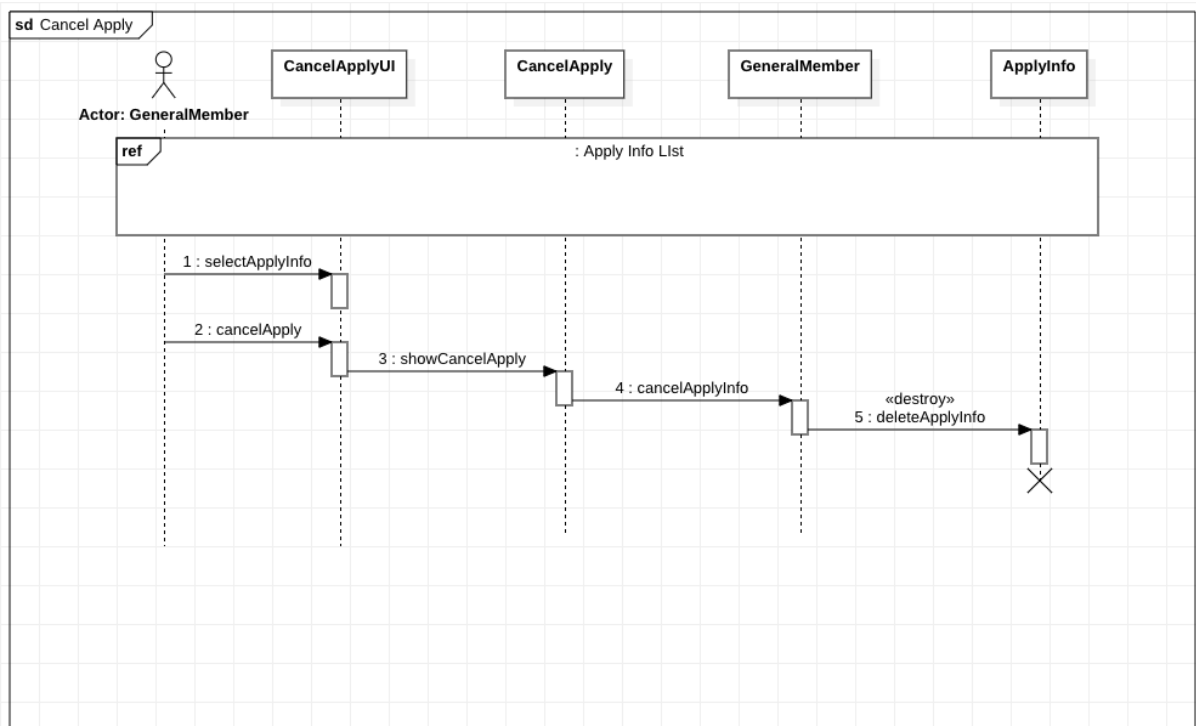


< fig 17. Use Case : Apply >

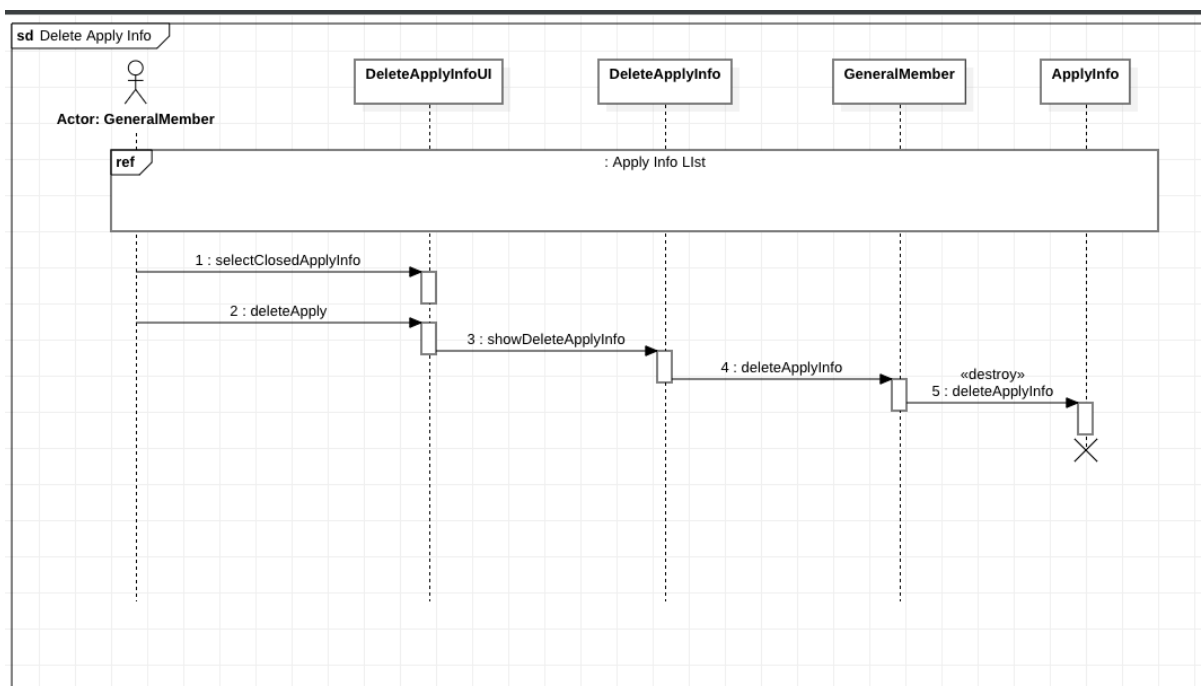
Ref는 Use Case : Show Recruit Info Detail 의 리턴 값을 표시하였다. Apply 이후에 RecruitInfo의 지원자 수를 1 늘리기 위해 setApplyNum 함수를 호출하였다. 따라서 RecruitInfo의 attribute에 numOfApplicants를 포함하였다.



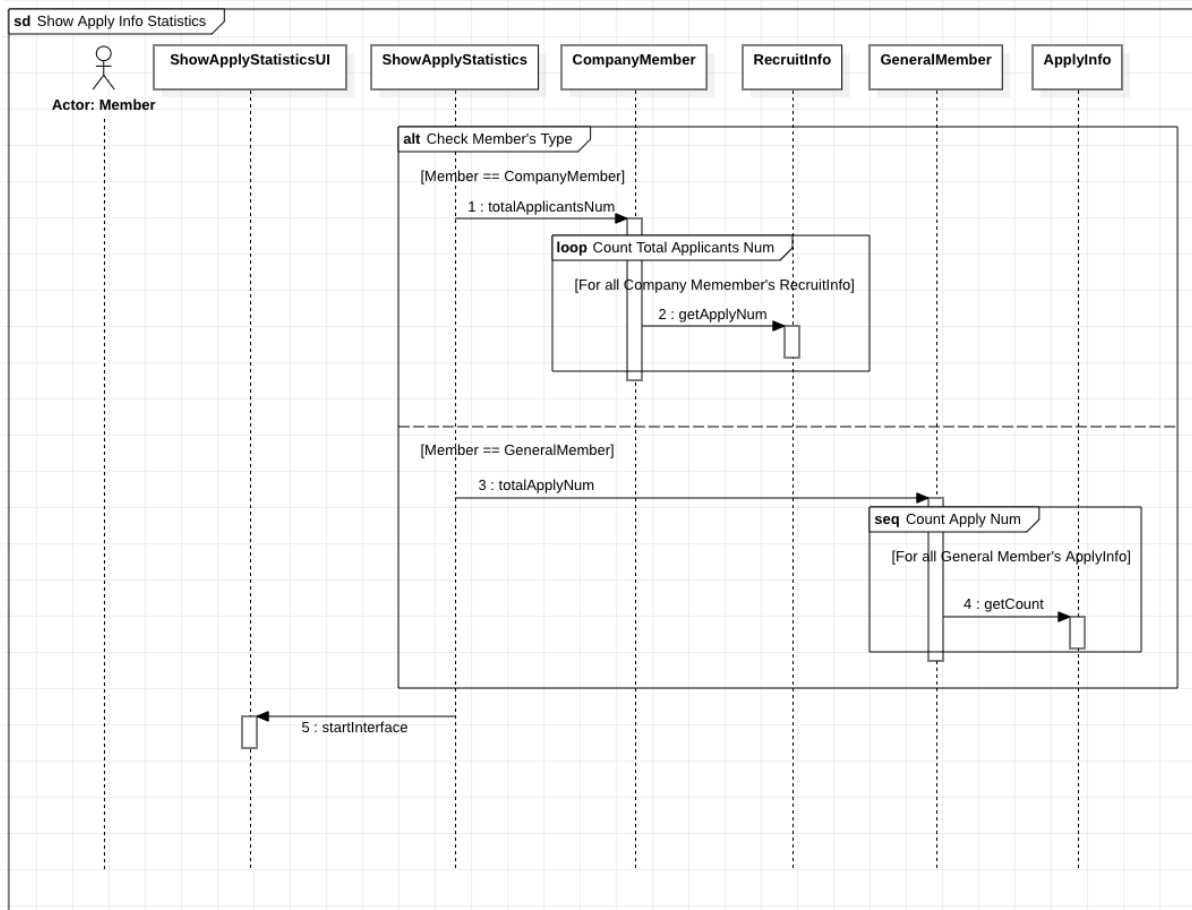
< fig 18. Use Case : Show Apply Info List >



< fig 19. Use Case : Cancel Apply >



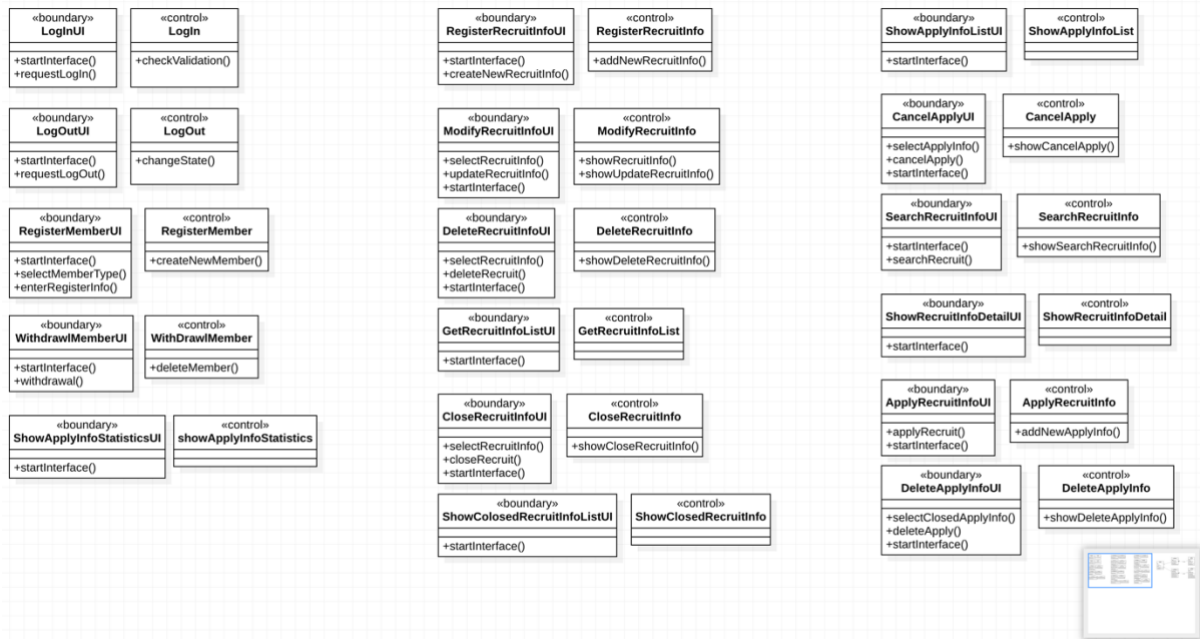
< fig 20. Use Case : Delete Apply Info >



< fig 21. Use Case : Show Apply Info Statistics >

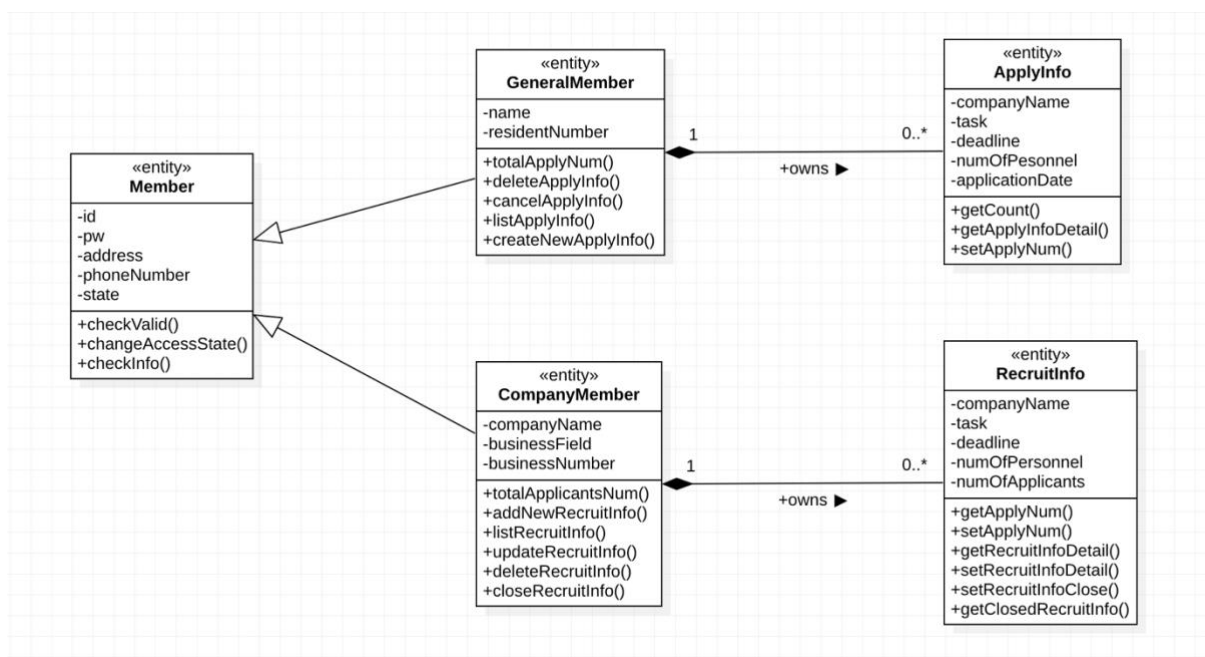
회사 회원의 경우 마감 시간 기준 업무별 지원자 수를 출력해야 한다. 따라서 월별로 정렬된 채용정보의 마감 시간을 확인하고, 해당하는 채용정보의 numOfApplicant 를 불러오는 getApplyNum 함수를 호출하는 로직으로 구성하였다.

일반 회원의 경우 마감 시간 기준 월별 지원 횟수를 출력해야 한다. 따라서 월별로 정렬된 지원정보의 마감 시간을 확인하고, 개수를 count하는 getCount 함수를 사용하는 로직으로 구성하였다.



< fig 22. Boundary Class와 Control Class >

위의 이미지는 모든 Use Case에 대한 Boundary Class와 Control Class를 나타낸 이미지이다.



< fig 23. Generalization과 Association >

위의 이미지는 Class 사이의 Generalization과 Association을 표현한 이미지이다.