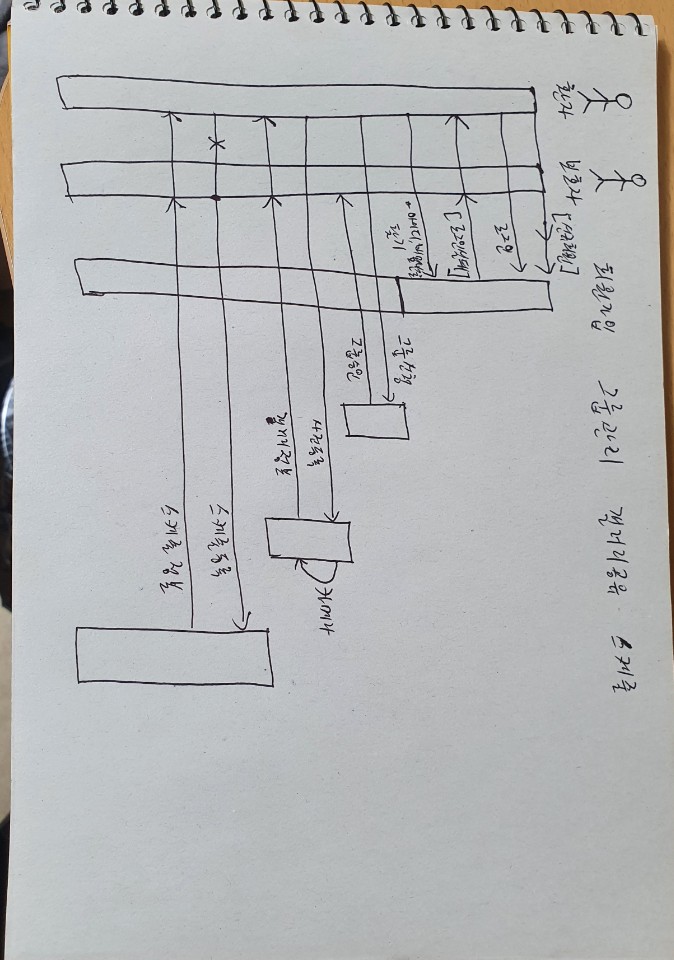
|  |  |  |
| --- | --- | --- |
| **소프트웨어설계 (팀활동)**  **Behavioral Modeling** | | |
|  | | |
| **제출팀** | 프로젝트명: | 치매 기억 저장소 |
| **2조** | 이름: | 유명음 |

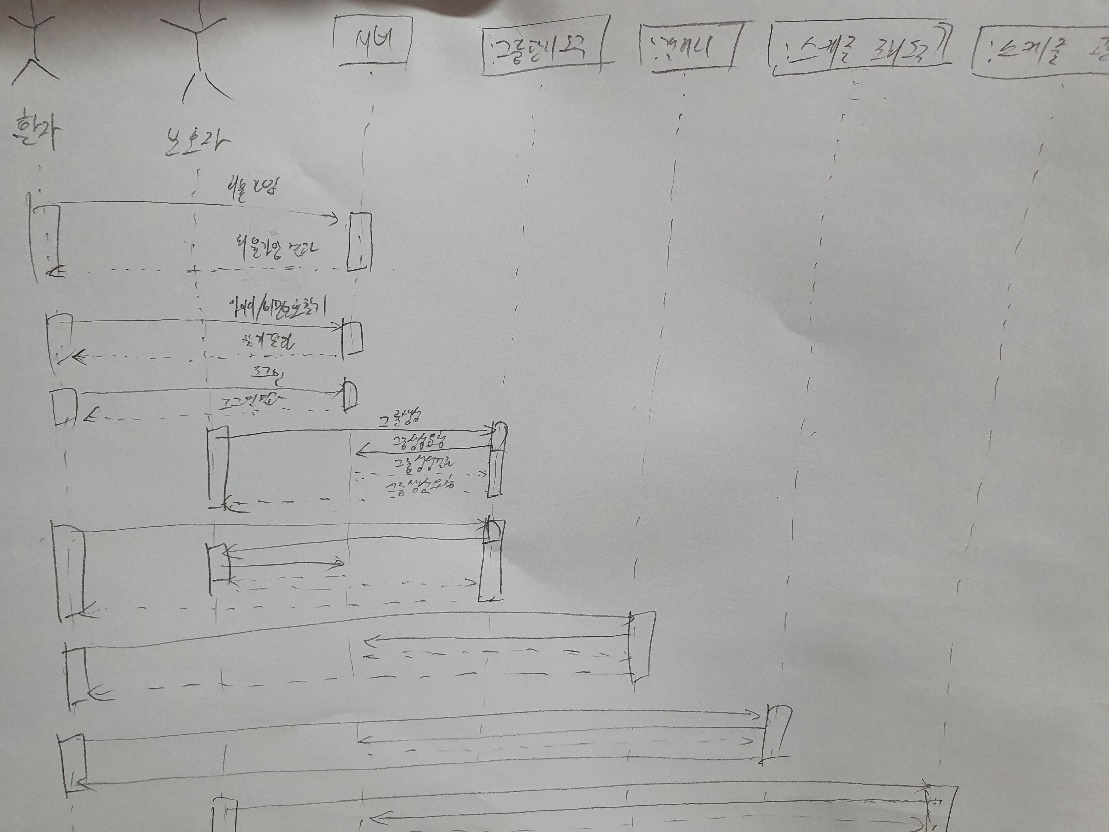
**1. Sequence Diagrams by the use-cases**

(1) 초안 (신기철)

****

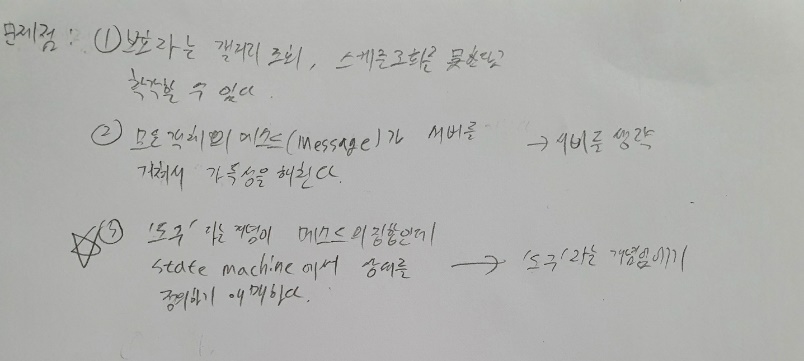
(2) 초안 수정 : 객체 개념 수정 (유명음)

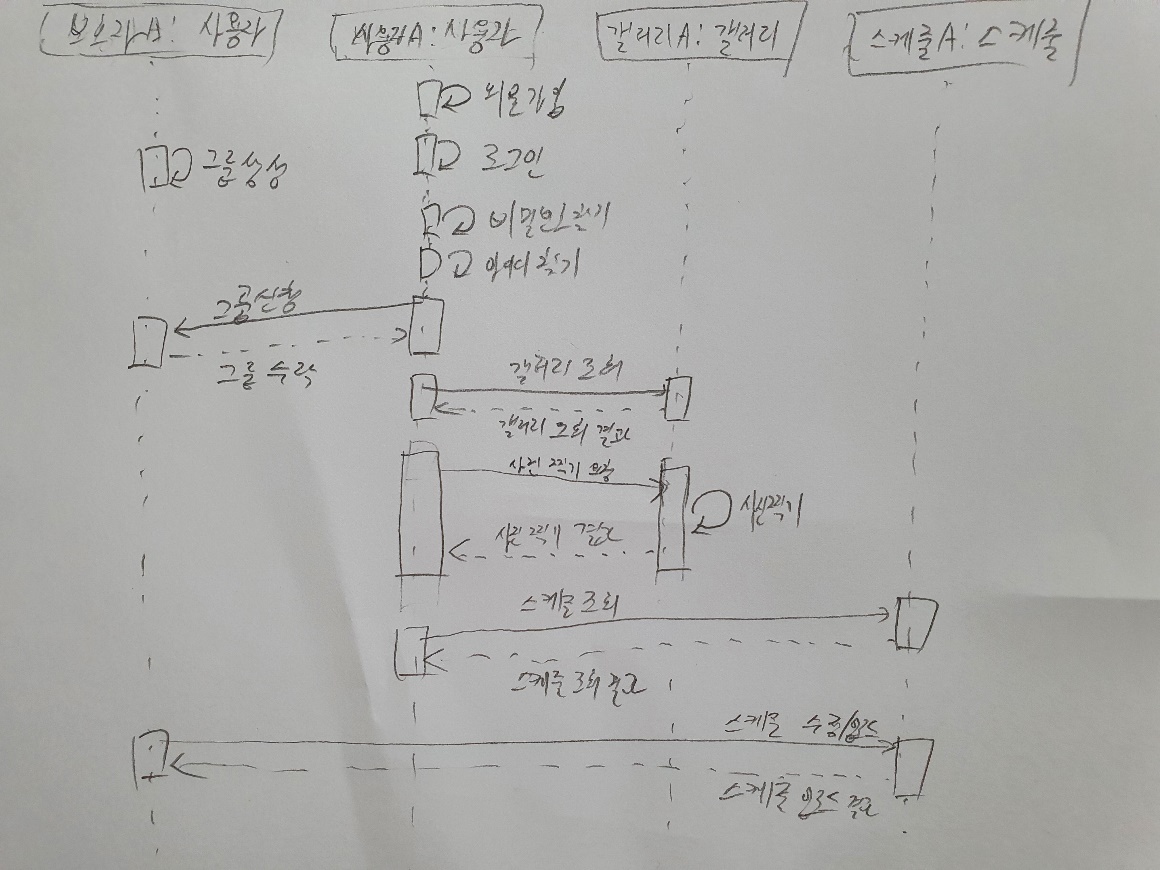
|  |  |
| --- | --- |
| 기존 문제점 | 해결 방법 |
| ‘회원가입’, ‘그룹관리’는 객체가 아니다. | 각각 ‘서버’와 ‘그룹관리 도구’로 대체한다. 마찬가지로 갤러리와 스케줄도 대체한다. |

****

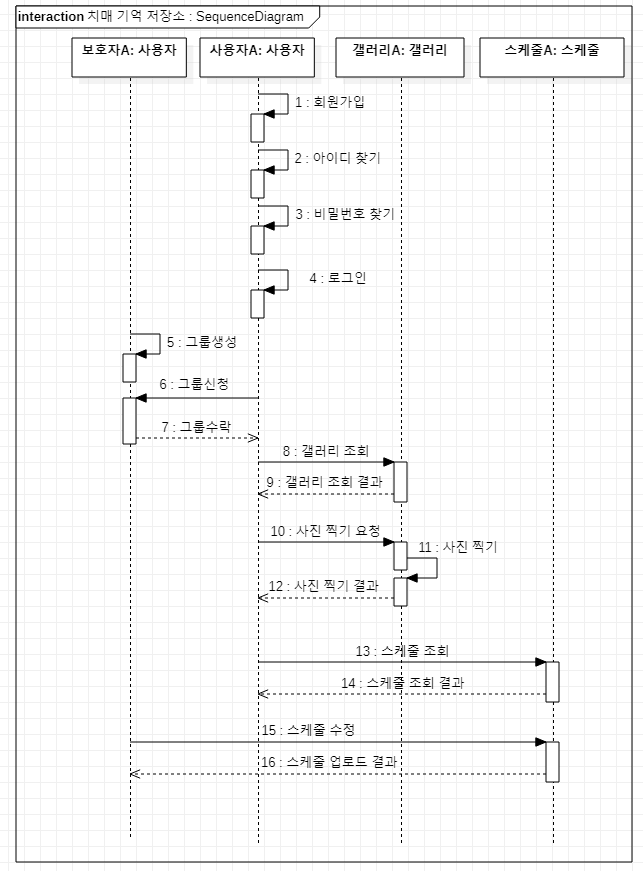
(3) 토의 (조원 전체)

|  |  |
| --- | --- |
| 기존 문제점 | 해결 방법 |
| 모든 객체의 메소드(Message)가 서버를 거쳐서 가독성이 떨어진다. | 서버를 생략해서 가독성을 높인다. |
| 객체 이름 양식이 규칙과 맞지 않다. | ‘객체 이름:클래스 이름’ 형식으로 변경한다. 단, 하나밖에 없는(static한) 객체는 객체 이름을 생략한다. |
| 환자에만 로그인, 갤러리 조회 등이 연결되어 있어서 보호자에는 기능이 없다고 생각할 수 있다. | 환자를 사용자로 바꿔서 보호자도 해당기능이 있음을 명시한다. 더불어 위치도 조정해서 최대한 메시지 화살표의 길이를 줄인다. |
| 도구가 많은데, state machine 토의 중 이 도구의 상태를 정의하기 애매했다. | ‘도구’의 개념을 없애고 좀 더 객체에 가깝게 표현한다.  ex) 스케줄 조회 도구 + 스케줄 관리 도구 = 스케줄 |





(4) 최종본 (UML)

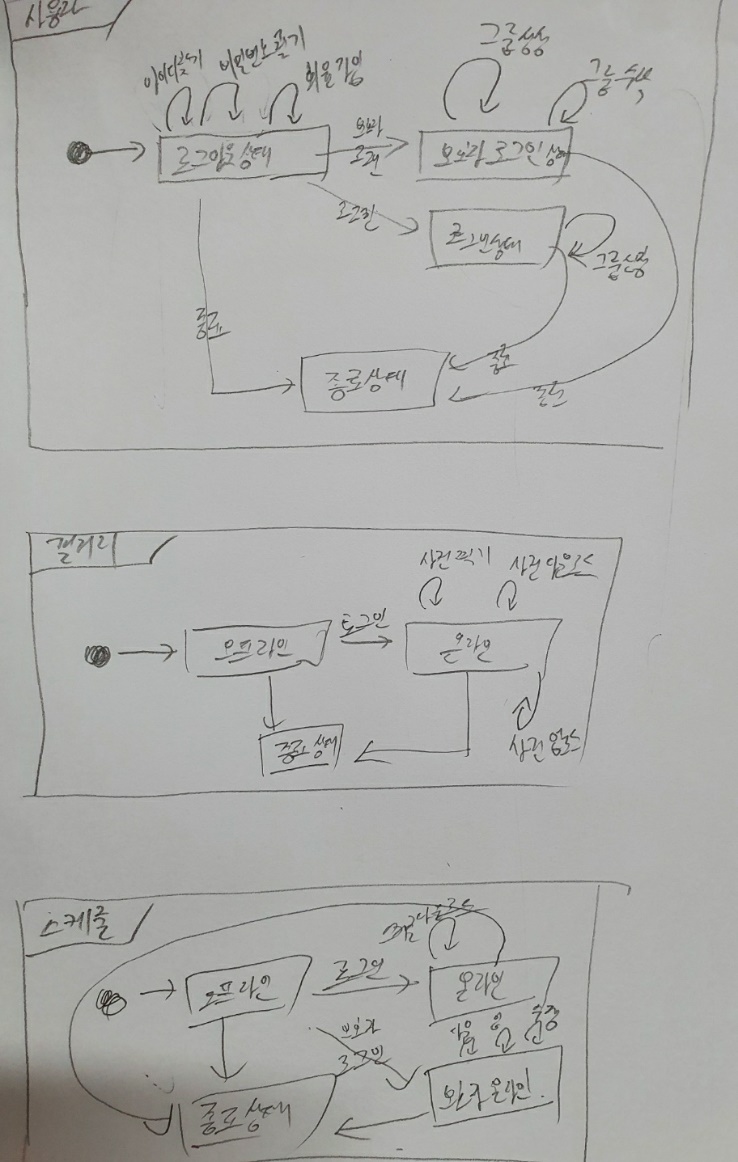
****

**2. Search & Draw the object’s states (State Diagram)**

(1) sequence diagram 최종본을 기반으로 토의를 거쳐 state diagram 생성

※ 토의 내용

* 서버를 생략하였으므로 회원가입, 그룹생성 등의 transition은 self-message(자기 방향으로 가는 화살표)와 같이 표현한다.
* 권한 차이로 생기는 state는 로그인에서 분기를 나눈다.
* 모든 객체는 앱 시작에 생성, 앱 종료에 소멸되므로 이를 표현한다. (모든 상태에서 종료 상태로 가는 transition 표시)



(2) 최종본 (UML)

