PROJECT DOCUMENT

System Model (Sequence Diagram) Document

제 WeekdaysIdea조

조원 : 201502094 이재호

201704146 박지은

지도교수: 원유재 (서명)

Document Revision History

REV#	DATE	AFFECTED SECTION	AUTHOR
1	2020/05/14	Usecase Diagram, Sequence Diagram 작성	이재호, 박지은
2	2020/05/16	Sequence Diagram 수정 및 설명 추가	이재호, 박지은

Table of Contents

List of Figure

1. Introduction

1.1. Objective

이 문서는 블록체인을 이용한 수목 진료 시스템의 시스템 모델(시퀀스 다이어그램)에 대한 내용을 기술하고 있다. 요구사항 명세 단계에서 작성한 유스케이스 다이어그램을 기반으로 각 유스케이스의 상세한 내부 동작 흐름을 시퀀스 다이어그램으로 모델링한다.

2. Use Case Diagram

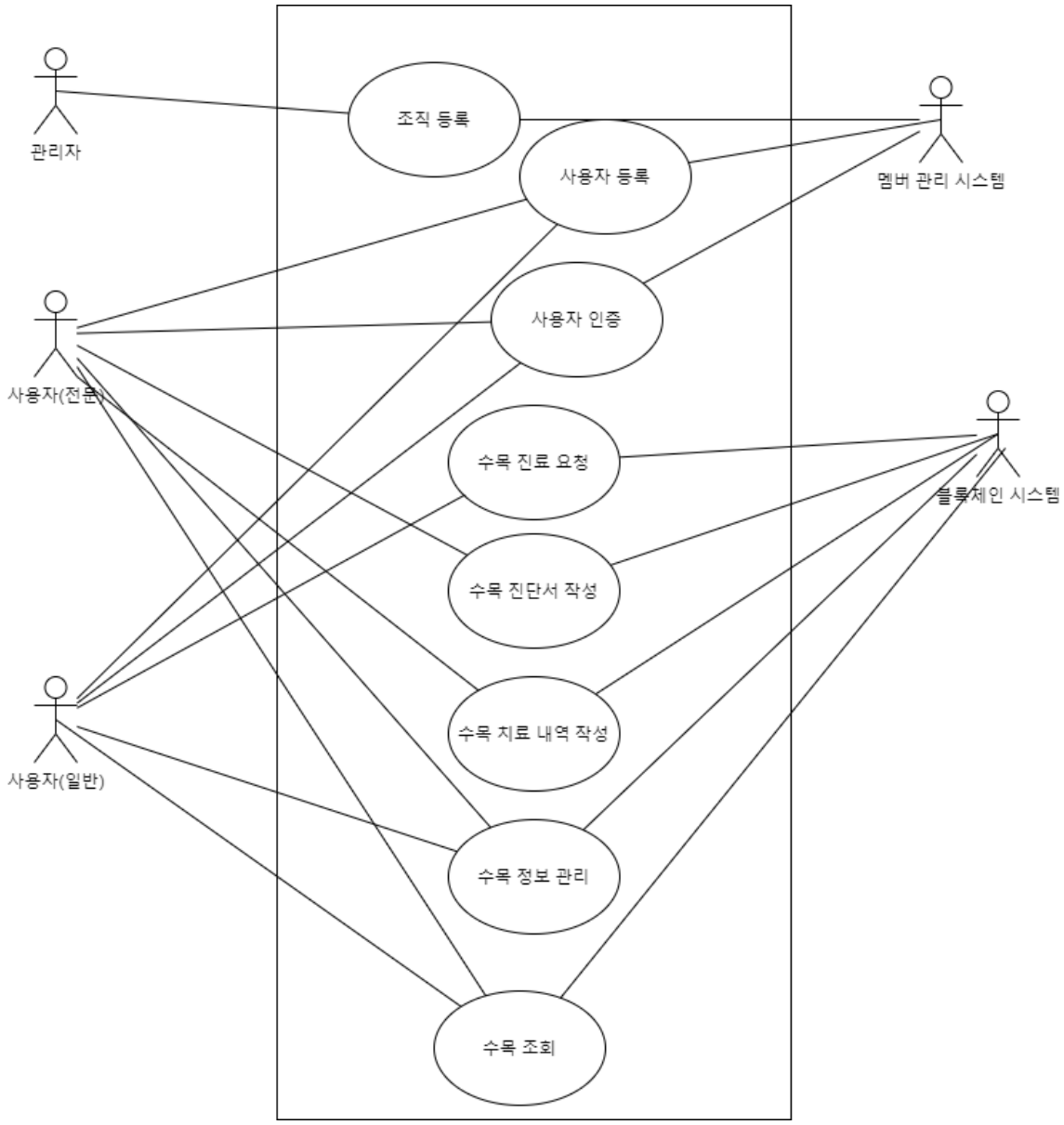

Figure 1 – Use Case Diagram

3. Sequence Diagram

3.1. 조직 등록

조직 등록은 Client 객체에 새로운 조직을 만들어 사용자를 관리하고 권한을 조절할 수 있도록 한다.

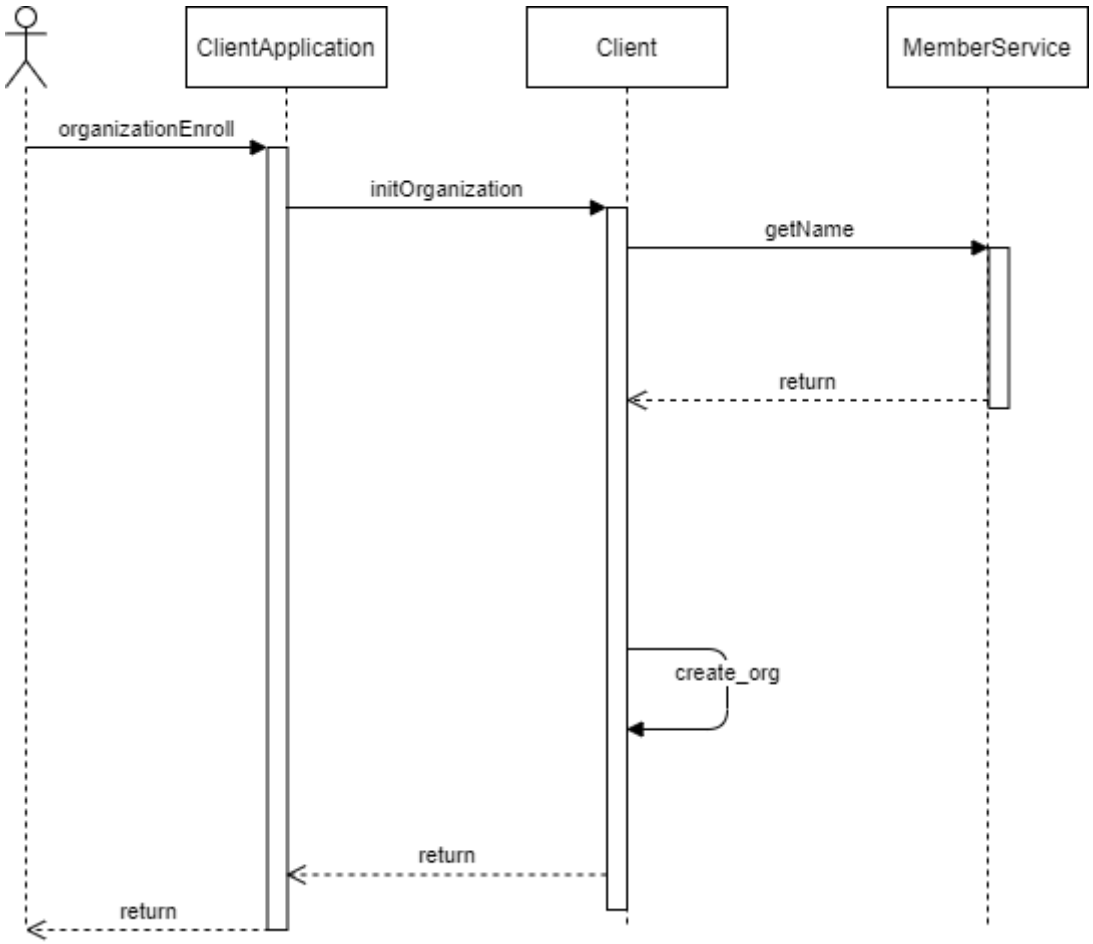

Figure 2 –조직 등록 Diagram

1. 사용자는 ClientApplication에 새로운 조직 생성을 요청한다.

1.1: ClientApplication은 Client에 신규 조직 생성을 호출한다.

1.1.1: Client는 MemberService의 이름을 받아온다. MemberService가 없다면 먼저 MemberService를 만든다.

1.1.2: Client는 받아온 이름에 해당하는 조직을 새로 만든다.

1.2: Client는 ClientApplication에 조직 등록이 완료됐음을 알린다.

2: ClientApplication은 사용자에게 조직 등록의 모든 과정이 완료됐음을 알린다.

3.2. 사용자 등록

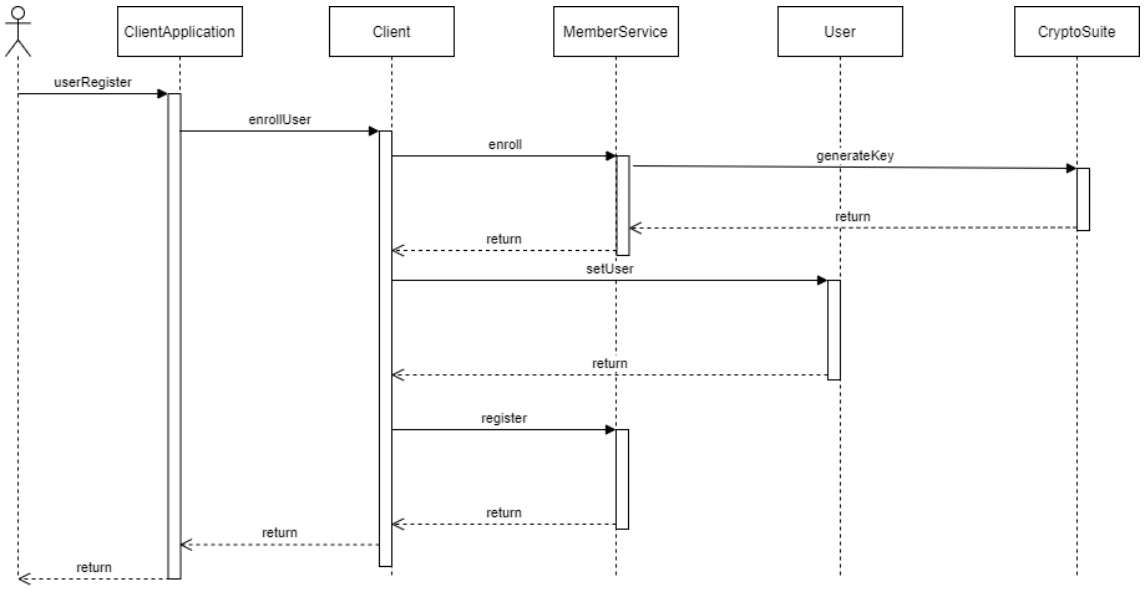

Figure 3–사용자 등록 Diagram

1. 사용자는 ClientApplication에 사용자 등록을 요청한다.

1.1: ClientApplication은 Client에 사용자 등록을 요청한다.

1.1.1: Client는 MemberService에 인증서 등록을 요청한다.

1.1.1.1: MemberService는 인증서 등록을 위해 CryptoSuite를 통해 키를 생성한다.

1.1.2: 등록할 사용자 정보를 위해 새로운 User를 만들어 정보와 역할을 설정한다.

1.1.3: 만들어진 User와 인증서를 연결한 뒤 MemberService에 등록한다.

1.2: Client가 사용자 등록이 완료되었음을 ClientApplication에 알린다.

2. 사용자 등록이 완료된다.

3.3. 사용자 인증

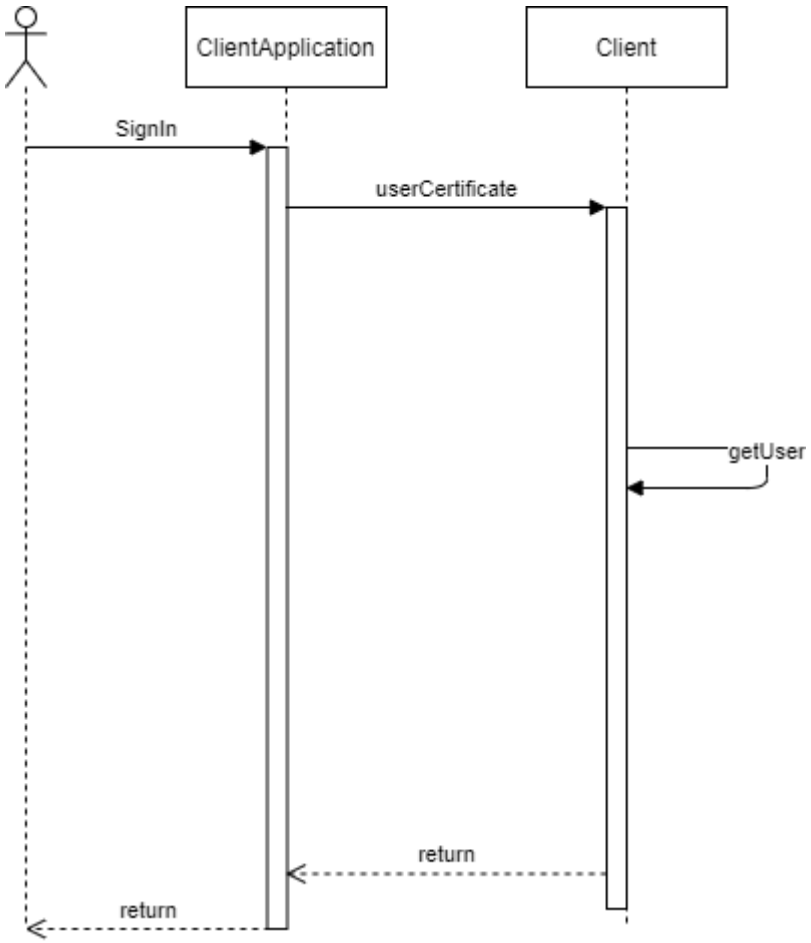

Figure 3–사용자 인증 Diagram

1. 사용자는 소유한 인증서를 바탕으로 ClientApplication에 인증을 요청한다.

    1.1: ClientApplication에서 Client에 사용자 인증을 요청한다.

        1.1.1: Client에 속한 Organization 정보를 확인해 사용자가 속해있는지 확인한다.

    1.2 ClientApplication에 사용자 확인이 완료되었음을 알린다.

2. 사용자 인증이 완료된다.

3.4. 수목 진료 요청

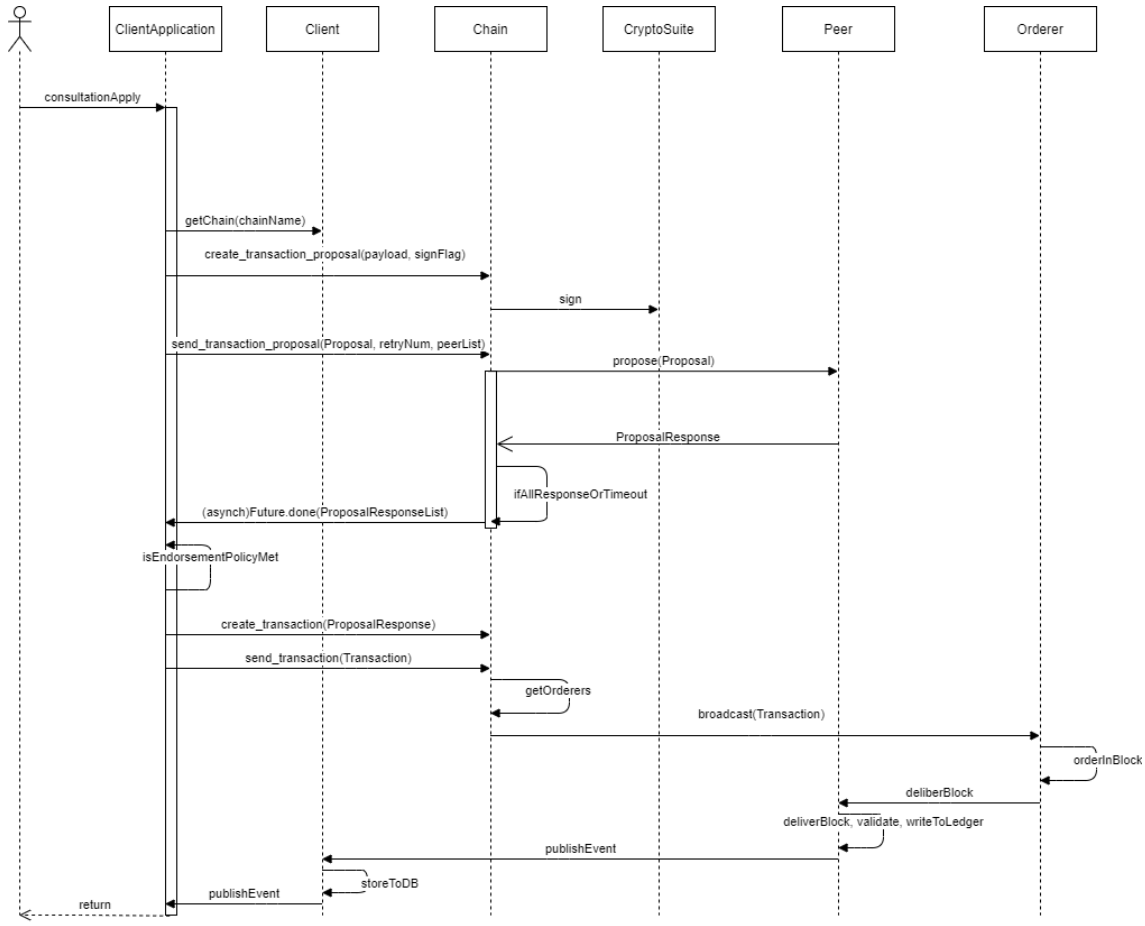

Figure 4 수목 진료 요청Diagram

1. 사용자가 ClientApplication으로 수목 진료 요청을 한다.

    1.1: 수목 진료 요청을 위한 체인의 이름을 확인한다.

    1.2: 확인한 체인으로 트랙잭션 요청을 생성한다.

    1.3: 체인에 속한 Peer들에게 트랜잭션 요청을 보낸다.

        1.3.1: Peer들은 받은 트랜잭션 요청이 유효한지 확인한 뒤 Response를 보낸다.

        1.3.2: Peer들의 Response를 확인하고 ClientApplication에게 전달한다.

    1.4: ClientApplication은 Response를 확인해 Endorsement Policy에 준하는 Reponse가 왔는지 검토한다.

1.5: 검토가 완료되면 proposal과 reponse, 그리고 체인코드를 담아 transaction을 만들어 체인으로 보낸다.

1.6 체인에 해당하는 Orderer를 확인해 transaction을 전달한다.

1.7 Orderer가 트랜잭션을 받아 시간 순으로 정렬해 블록을 생성한다.

1.8 Committing Peer가 트랜잭션을 검증하고 커밋해 유효한지 확인하고, transaction에 valid/invliad 값이 태그된다.

1.9 이벤트를 발생시켜 검증을 마친 블록을 DB에 저장한다.

2. 과정이 마무리되어 이벤트가 발생하고 ClientApplication에게 알려 트랜잭션 완료를 알린다.

3.5. 수목 진료 진단서 등록

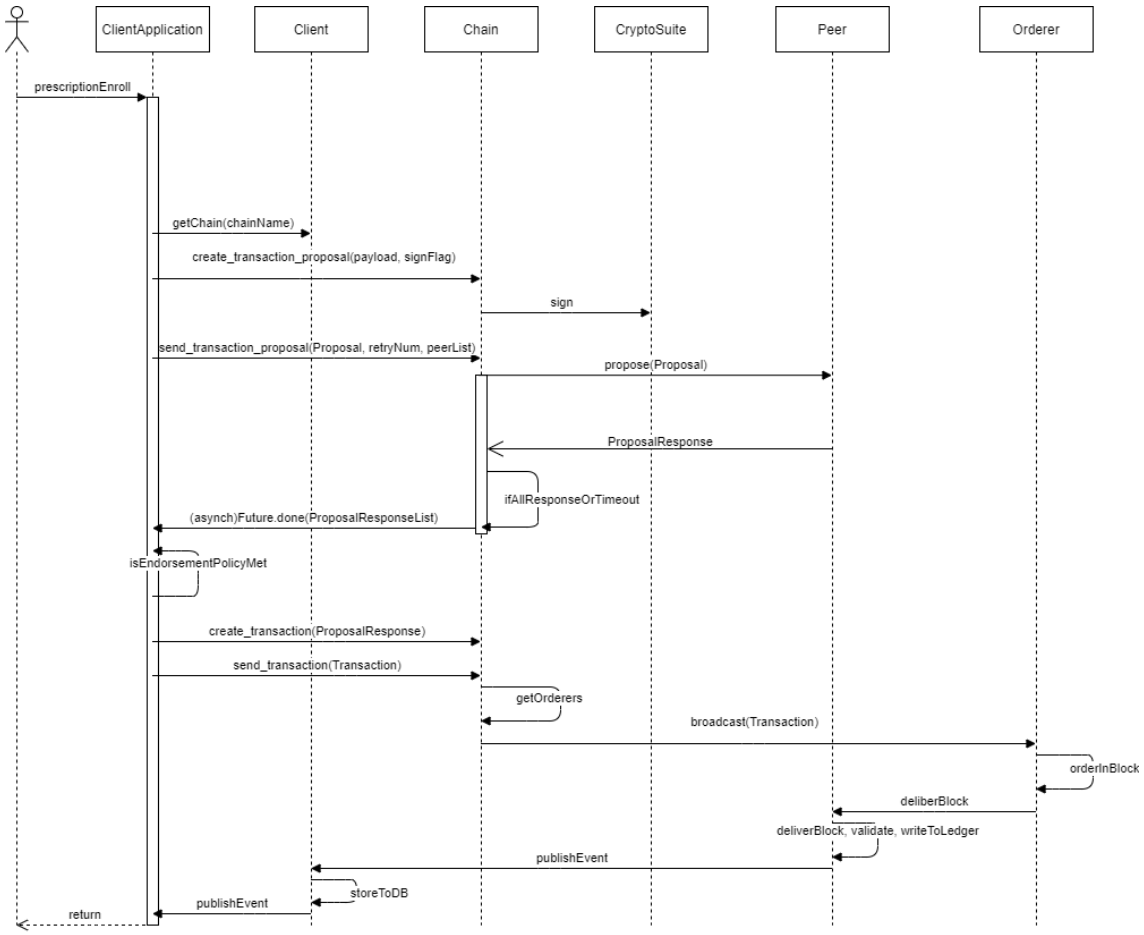

Figure 5–수목 진료 진단서 등록 Diagram

1.사용자가 ClientApplication으로 수목 진료 진단서 등록 요청을 한다.

1.1: 수목 진료 진단서 등록을 위한 체인의 이름을 확인한다.

1.2: 확인한 체인으로 트랙잭션 요청을 생성한다.

1.3: 체인에 속한 Peer들에게 트랜잭션 요청을 보낸다.

1.3.1: Peer들은 받은 트랜잭션 요청이 유효한지 확인한 뒤 Response를 보낸다.

1.3.2: Peer들의 Response를 확인하고 ClientApplication에게 전달한다.

1.4: ClientApplication은 Response를 확인해 Endorsement Policy에 준하는 Reponse가 왔는지 검토한다.

1.5: 검토가 완료되면 proposal과 reponse, 그리고 체인코드를 담아 transaction을 만들어 체인으로 보낸다.

1.6 체인에 해당하는 Orderer를 확인해 transaction을 전달한다.

1.7 Orderer가 트랜잭션을 받아 시간 순으로 정렬해 블록을 생성한다.

1.8 Committing Peer가 트랜잭션을 검증하고 커밋해 유효한지 확인하고, transaction에 valid/invliad 값이 태그된다.

1.9 이벤트를 발생시켜 검증을 마친 블록을 DB에 저장한다.

2. 과정이 마무리되어 이벤트가 발생하고 ClientApplication에게 알려 트랜잭션 완료를 알린다.

3.6. 수목 치료 내역 작성

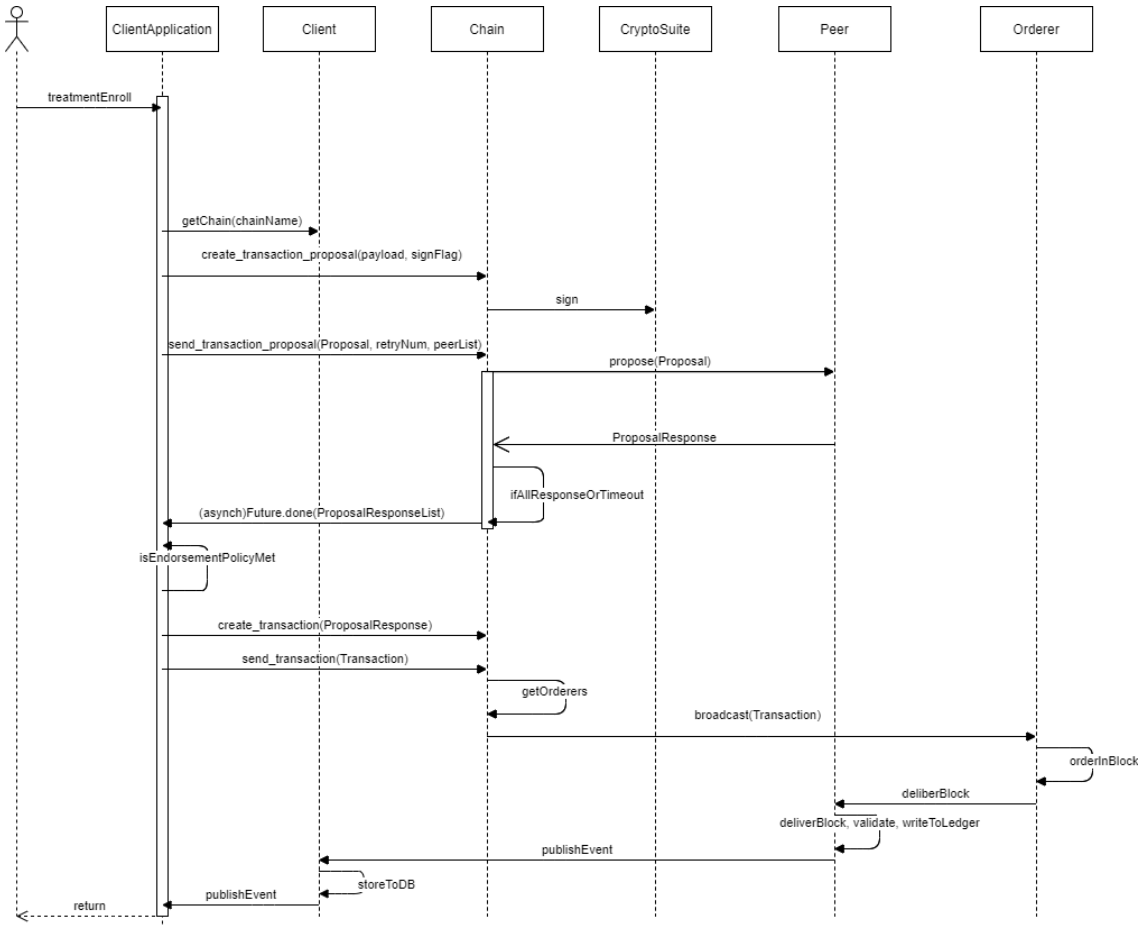

Figure 6–수목 치료 내역 작성 Diagram

1.사용자가 ClientApplication으로 수목 치료 내역 작성 후 등록 요청을 한다.

1.1: 수목 진료 치료 내역 등록을 위한 체인의 이름을 확인한다.

1.2: 확인한 체인으로 트랙잭션 요청을 생성한다.

1.3: 체인에 속한 Peer들에게 트랜잭션 요청을 보낸다.

1.3.1: Peer들은 받은 트랜잭션 요청이 유효한지 확인한 뒤 Response를 보낸다.

1.3.2: Peer들의 Response를 확인하고 ClientApplication에게 전달한다.

1.4: ClientApplication은 Response를 확인해 Endorsement Policy에 준하는 Reponse가 왔는지 검토한다.

1.5: 검토가 완료되면 proposal과 reponse, 그리고 체인코드를 담아 transaction을 만들어 체인으로 보낸다.

1.6 체인에 해당하는 Orderer를 확인해 transaction을 전달한다.

1.7 Orderer가 트랜잭션을 받아 시간 순으로 정렬해 블록을 생성한다.

1.8 Committing Peer가 트랜잭션을 검증하고 커밋해 유효한지 확인하고, transaction에 valid/invliad 값이 태그된다.

1.9 이벤트를 발생시켜 검증을 마친 블록을 DB에 저장한다.

2. 과정이 마무리되어 이벤트가 발생하고 ClientApplication에게 알려 트랜잭션 완료를 알린다.

3.7. 수목 정보 관리(등록/수정/삭제)

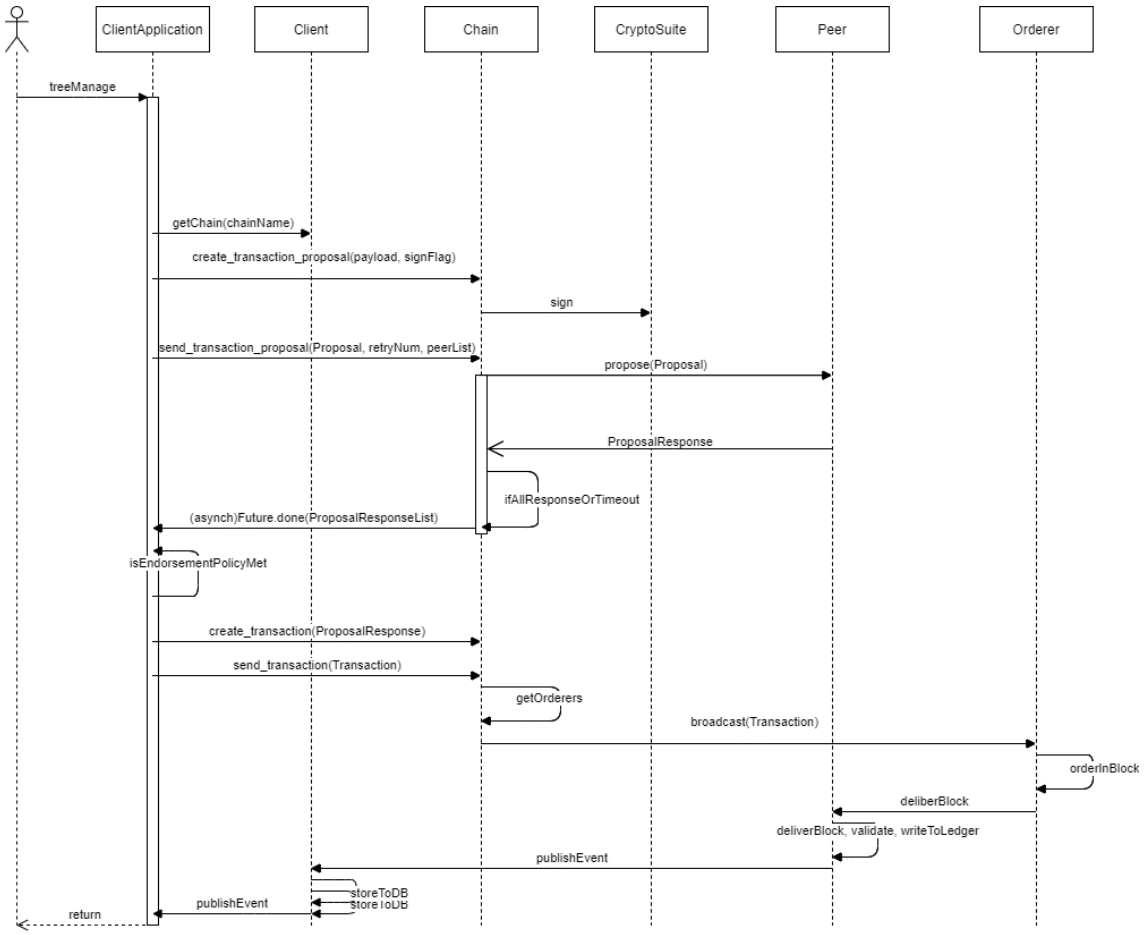

Figure 7–수목 정보 관리 Diagram

1.사용자가 ClientApplication으로 수목 정보 관리 요청을 한다.

1.1: 수목 정보 관리를 위한 체인의 이름을 확인한다.

1.2: 확인한 체인으로 트랙잭션 요청을 생성한다.

1.3: 체인에 속한 Peer들에게 트랜잭션 요청을 보낸다.

1.3.1: Peer들은 받은 트랜잭션 요청이 유효한지 확인한 뒤 Response를 보낸다.

1.3.2: Peer들의 Response를 확인하고 ClientApplication에게 전달한다.

1.4: ClientApplication은 Response를 확인해 Endorsement Policy에 준하는 Reponse가 왔는지 검토한다.

1.5: 검토가 완료되면 proposal과 reponse, 그리고 체인코드를 담아 transaction을 만들어 체인으로 보낸다.

1.6 체인에 해당하는 Orderer를 확인해 transaction을 전달한다.

1.7 Orderer가 트랜잭션을 받아 시간 순으로 정렬해 블록을 생성한다.

1.8 Committing Peer가 트랜잭션을 검증하고 커밋해 유효한지 확인하고, transaction에 valid/invliad 값이 태그된다.

1.9 이벤트를 발생시켜 검증을 마친 블록을 DB에 저장한다.

2. 과정이 마무리되어 이벤트가 발생하고 ClientApplication에게 알려 트랜잭션 완료를 알린다.

3.8. 수목 조회

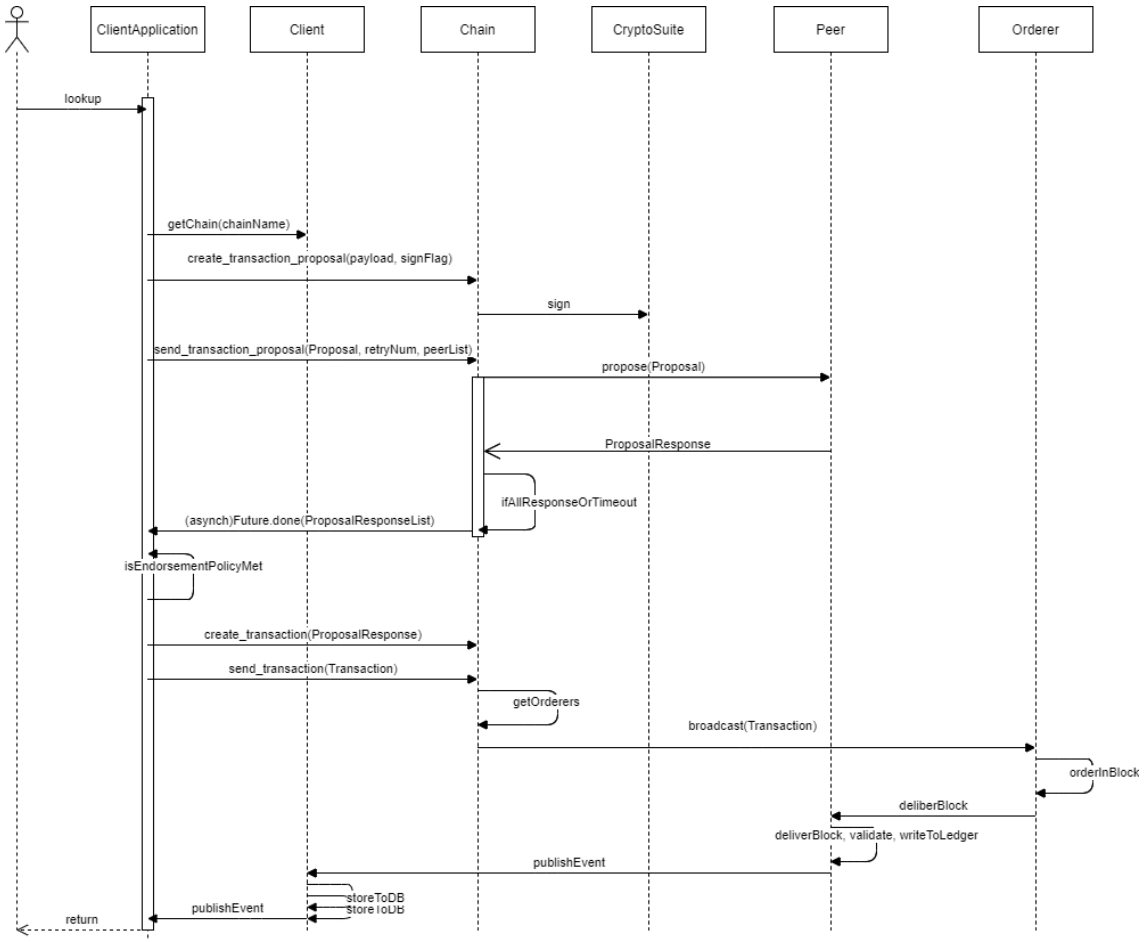

Figure 8–수목 조회 Diagram

1.사용자가 ClientApplication으로 수목 조회 요청을 한다.

1.1: 수목 조회를 위한 체인의 이름을 확인한다.

1.2: 확인한 체인으로 트랙잭션 요청을 생성한다.

1.3: 체인에 속한 Peer들에게 트랜잭션 요청을 보낸다.

1.3.1: Peer들은 받은 트랜잭션 요청이 유효한지 확인한 뒤 Response를 보낸다.

1.3.2: Peer들의 Response를 확인하고 ClientApplication에게 전달한다.

1.4: ClientApplication은 Response를 확인해 Endorsement Policy에 준하는 Reponse가 왔는지 검토한다.

1.5: 검토가 완료되면 proposal과 reponse, 그리고 체인코드를 담아 transaction을 만들어 체인으로 보낸다.

1.6 체인에 해당하는 Orderer를 확인해 transaction을 전달한다.

1.7 Orderer가 트랜잭션을 받아 시간 순으로 정렬해 블록을 생성한다.

1.8 Committing Peer가 트랜잭션을 검증하고 커밋해 유효한지 확인하고, transaction에 valid/invliad 값이 태그된다.

1.9 이벤트를 발생시켜 검증을 마친 블록을 DB에 저장한다.

2. 과정이 마무리되어 이벤트가 발생하고 ClientApplication에게 알려 트랜잭션 완료를 알린다.