

3. Design

Fire Moment



Student No	21912113
Name	송재민
E-mail	ssong_1015@naver.com

[Revision history]

Revision date	Version #	Description	Author
06/09/2023	1.00	First Design	송재민

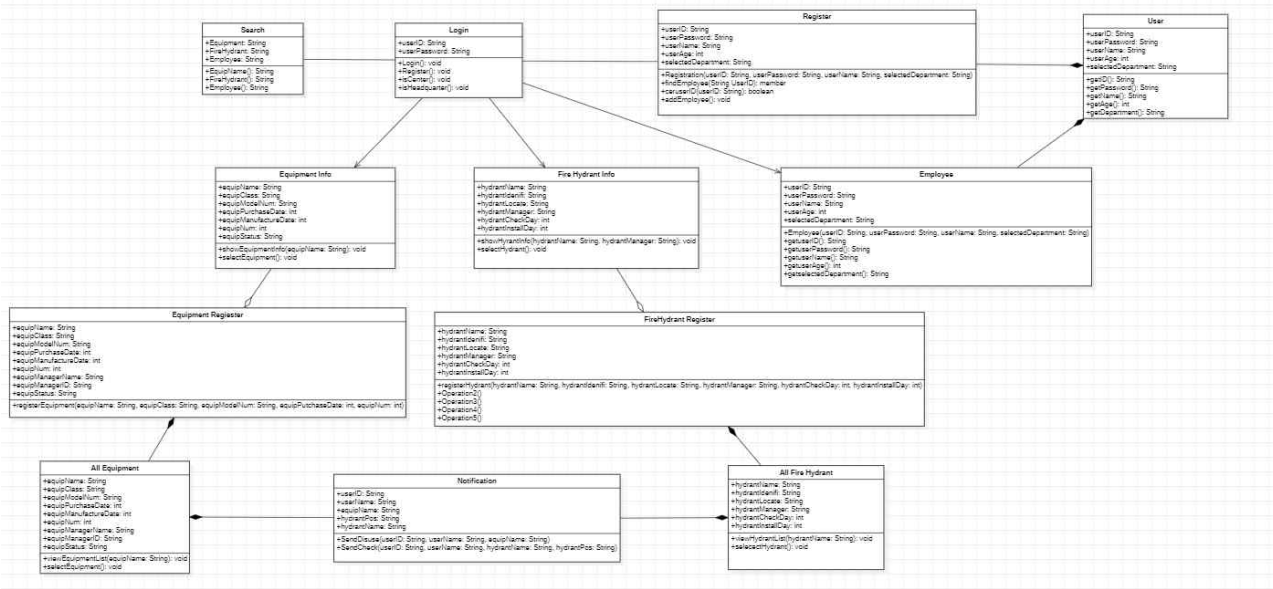
= Contents =

1. Introduction	1
2. Class diagram	2
3. Sequence diagram	15
4. State machine diagram	26
5. Implementation requirements	28
6. Glossary	28
7. References	28

1. Introduction

화재 진압은 소방관에게는 생명을 위협할 수 있는 일이며, 동시에 공공의 안전과 이익을 위한 필수적인 역할이다. 따라서 화재 현장에서는 소방 장비와 소화전을 주기적으로 관리하는 것이 매우 중요하다. 그러나 제대로 갖춰지지 않은 장비나 시설이 있다면 소방관의 생명이 위태롭게 될 수 있다. Fire Moment은 소방 장비의 안전 여부를 항상 확인하고 소화전에 빠르게 접근하여 사용할 수 있도록 도와주는 애플리케이션으로, 개인 소방 장비와 소화전의 정보를 데이터베이스에 저장하고 관리한다. 이를 통해 화재 발생 시 생명을 구하고 재산을 보호하는 데 도움을 줄 수 있다. 또한, 시간을 절약하고 오류를 최소화하여 소방관과 상황실 등의 비상 대응자들이 중요한 작업에 집중할 수 있도록 지원하는 것을 목표로 두고 있다. 이러한 목적을 이루기 위해, 이번 문서는 Analysis에 이은 Design 단계를 다룰 예정이며, Class diagram, Sequence diagram, State machine diagram을 통해 각 Diagram에 대한 설명 시각화하여 보여줄 예정이다.

2. Class diagram



1) Login

Login
+userID: String +userPassword: String
+Login(): void +Register(): void +isCenter(): void +isHeadquarter(): void

Attributes
userID:String userPassword:String
Methods
Login(): void Register(): void isCenter(): void isHeadquarter(): void

2) Register

Register
+userID: String +userPassword: String +userName: String +userAge: int +selectedDepartment: String
+Registration(userID: String, userPassword: String, userName: String, selectedDepartment: String) +findEmployee(String UserID): member +ceruserID(userID: String): boolean +addEmployee(): void

Attributes
userID: String userPassword: String userName: String userAge: int selectedDepartment: String
Methods
Registration(userID:String, userPassword:String, userName:String, selectedDepartment:String) findEmployee(String UserID): member ceruserID(userID:String): boolean addEmployee(): void

3) Search

Search
+Equipment: String +FireHydrant: String +Employee: String
+EquipName(): String +FireHydrant(): String +Employee(): String

Attributes
Equipment: String
FireHydrant: String
Employee: String
Methods
EquipName(): String
FireHydrant(): String
Employee(): String

4) Equipment Info

Equipment Info
+equipName: String +equipClass: String +equipModelNum: String +equipPurchaseDate: int +equipManufactureDate: int +equipNum: int +equipStatus: String
+showEquipmentInfo(equipName: String): void +selectEquipment(): void

Attributes
equipName: String equipClass: String equipModelNum: String equipPurchaseDate: int equipManufactureDate: int equipNum: int equipStatus: String
Methods
showEquipmentInfo(equipName:String): void selectEquipment(): void

5) Fire Hydrant Info

Fire Hydrant Info
+hydrantName: String +hydrantIdenifi: String +hydrantLocate: String +hydrantManager: String +hydrantCheckDay: int +hydrantInstallDay: int
+showHyrantInfo(hydrantName: String, hydrantManager: String): void +selectHydrant(): void

Attributes
hydrantName: String
hydrantIdenifi: String
hydrantLocate: String
hydrantManager: String
hydrantCheckDay: int
hydrantInstallDay: int
Methods
showHyrantInfo(hydrantName:String, hydrantManager:String): void
selectHydrant(): void

6) Employee

Employee
+userID: String +userPassword: String +userName: String +userAge: int +selectedDepartment: String
+Employee(userID: String, userPassword: String, userName: String, selectedDepartment: String) +getUserID(): String +getUserPassword(): String +getuserName(): String +getUserAge(): int +getselectedDepartment(): String

Attributes
userID: String userPassword: String userName: String userAge: int selectedDepartment: String
Methods
Employee(userID:String, userPassword:String, userName:String, selectedDepartment:String) getUserID(): String getUserPassword(): String getuserName(): String getUserAge(): int getselectedDepartment(): String

7) User

User
+userID: String +userPassword: String +userName: String +userAge: int +selectedDepartment: String
+getID(): String +getPassword(): String +getName(): String +getAge(): int +getDepartment(): String

Attributes
userID: String userPassword: String userName: String userAge: int selectedDepartment: String
Methods
getID(): String getPassword(): String getName(): String getAge(): int getDepartment(): String

8) Equipment Register

Equipment Register
+equipName: String +equipClass: String +equipModelNum: String +equipPurchaseDate: int +equipManufactureDate: int +equipNum: int +equipManagerName: String +equipManagerID: String +equipStatus: String
+registerEquipment(equipName: String, equipClass: String, equipModelNum: String, equipPurchaseDate: int, equipNum: int)

Attributes
equipName: String equipClass: String equipModelNum: String equipPurchaseDate: int equipManufactureDate: int equipNum: int equipManagerName: String equipManagerID: String equipStatus: String
Methods
registerEquipment(equipName:String, equipClass:String, equipModelNum:String, equipPurchaseDate:int, equipNum:int)

9) Fire Hydrant Register

FireHydrant Register
+hydrantName: String +hydrantIdenifi: String +hydrantLocate: String +hydrantManager: String +hydrantCheckDay: int +hydrantInstallDay: int +registerHydrant(hydrantName: String, hydrantIdenifi: String, hydrantLocate: String, hydrantManager: String, hydrantCheckDay: int, hydrantInstallDay: int) +Operation2() +Operation3() +Operation4() +Operation5()

Attributes
hydrantName: String hydrantIdenifi: String hydrantLocate: String hydrantManager: String hydrantCheckDay: int hydrantInstallDay: int
Methods
registerHydrant(hydrantName:String, hydrantIdenifi:String, hydrantLocate:String, hydrantManager:String, hydrantCheckDay:int, hydrantInstallDay:int)

10) All Equipment

All Equipment
+equipName: String +equipClass: String +equipModelNum: String +equipPurchaseDate: int +equipManufactureDate: int +equipNum: int +equipManagerName: String +equipManagerID: String +equipStatus: String
+viewEquipmentList(equipName: String): void +selectEquipment(): void

Attributes
equipName: String equipClass: String equipModelNum: String equipPurchaseDate: int equipManufactureDate: int equipNum: int equipManagerName: String equipManagerID: String equipStatus: String
Methods
viewEquipmentList(equipName:String): void selectEquipment(): void

11) Notification

Notification
+userID: String +userName: String +equipName: String +hydrantPos: String +hydrantName: String
+SendDisuse(userID: String, userName: String, equipName: String) +SendCheck(userID: String, userName: String, hydrantName: String, hydrantPos: String)

Attributes
userID: String userName: String equipName: String hydrantName: String hydrantPos: String
Methods
SendDisuse(userID:String, userName:String, equipName:String) SendCheck(userID:String, userName:String, hydrantName:String, hydrantPos:String)

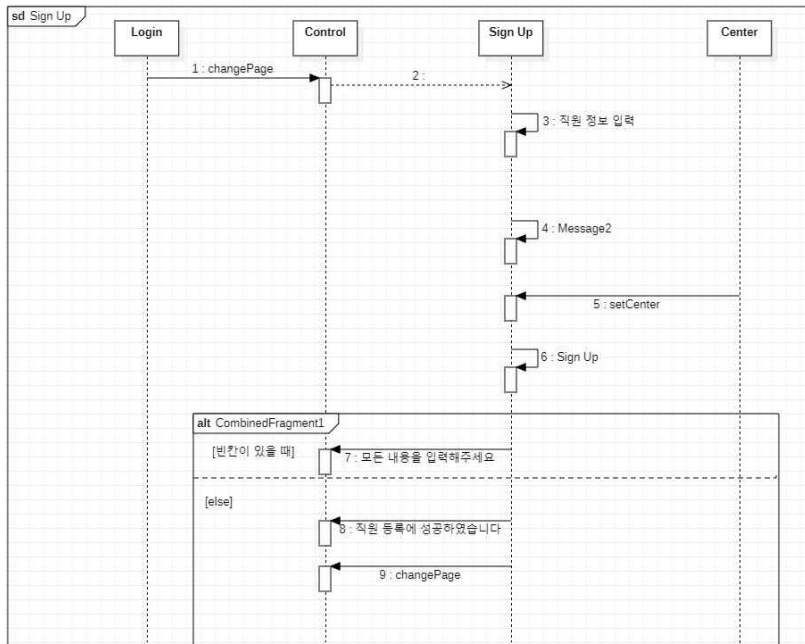
12) All Fire Hydrant

All Fire Hydrant
+hydrantName: String +hydrantIdenifi: String +hydrantLocate: String +hydrantManager: String +hydrantCheckDay: int +hydrantInstallDay: int
+viewHydrantList(hydrantName: String): void +selecectHydrant(): void

Attributes
hydrantName: String
hydrantIdenifi: String
hydrantLocate: String
hydrantManager: String
hydrantCheckDay: int
hydrantInstallDay: int
Methods
viewHydrantList(hydrantName:String): void
selecectHydrant(): void

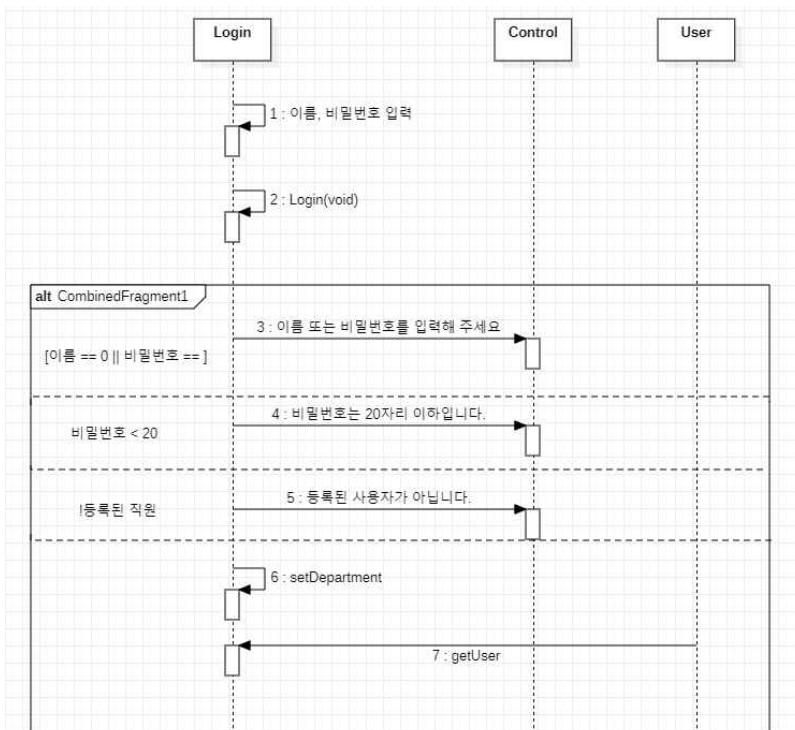
3. Sequence diagram

1) Sign up



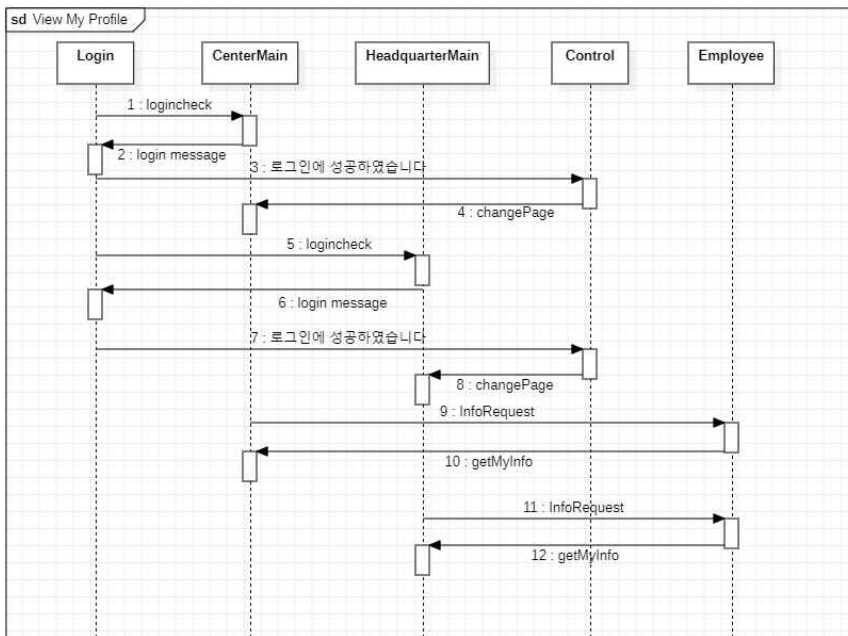
위의 그림은 회원가입 기능을 수행하는 과정을 나타낸 Sequence Diagram이다. 로그인 화면에서 직원 등록화면으로 넘어가서 기재해야하는 항목을 기재한 후 등록 버튼을 누르면 직원으로 등록이 성공했다는 메시지가 뜨면서 직원등록에 성공하게 된다.

2) Login



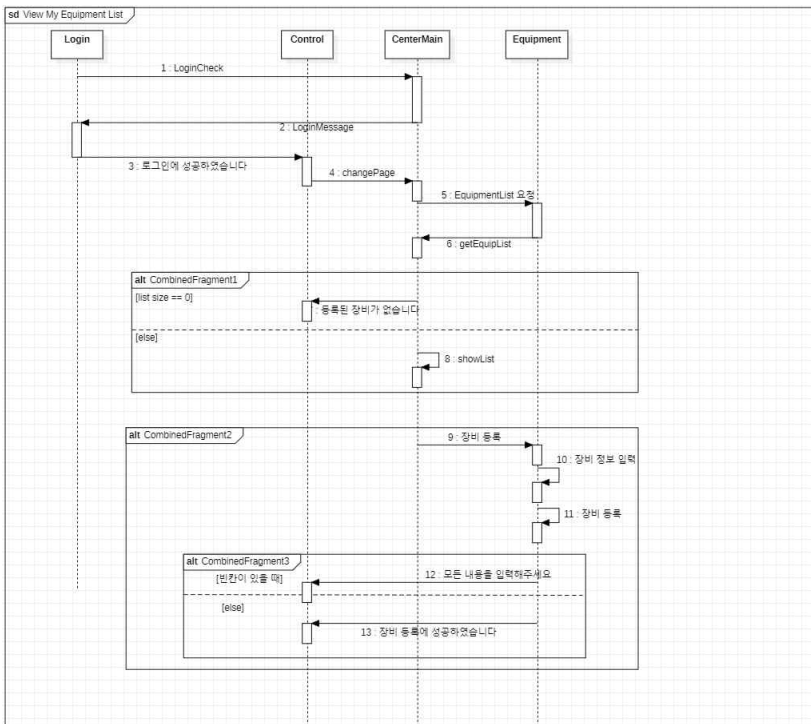
프로그램을 시작하면 나오는 로그인 화면이다. 등록된 직원이라면 자신의 이름과 비밀번호를 입력하여 로그인 버튼을 누르면 자신의 부서에 맞는 메인 화면이 나오게 된다. 만약 이름이나 비밀번호를 잘못 기입했다면 다시 입력하라는 메시지가 뜬다.

3) View my profile



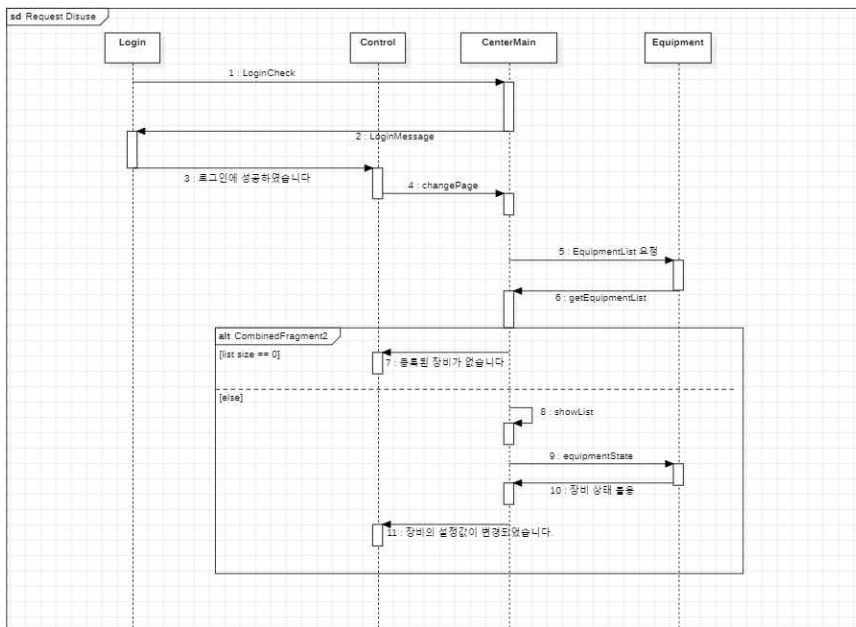
로그인이 성공하고 뜨는 메인화면에서 들어갈 수 있는 나의 프로필 화면이다. Employee에서 자신의 ID와 맞는 값을 가져와서 자신의 정보를 화면에 띄운다.

4) View my equipment list & registration



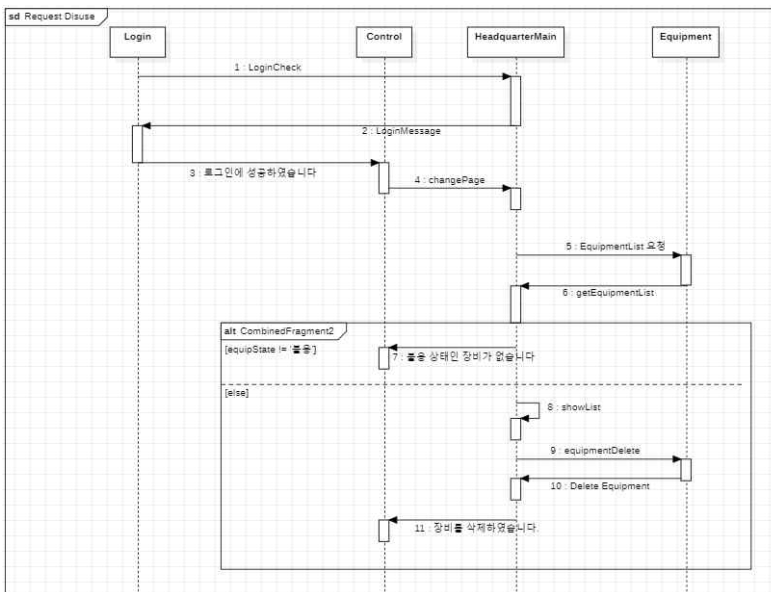
메인화면에서 확인 할 수 있는 나의 장비에 대한 Sequence Diagram이다. 자신이 등록한 장비가 화면에 띄워지게 되며, 원하는 장비를 선택할 경우, 해당 장비의 상세 정보가 띄워지게 된다. 또한 새로운 장비를 등록하고자 하는 경우 장비 등록 버튼을 눌러 해당하는 양식을 채우고 등록 버튼을 누른다면 장비 등록에 성공하였다는 알림이 뜨면서 등록이 완료된다.

5) Request disuse



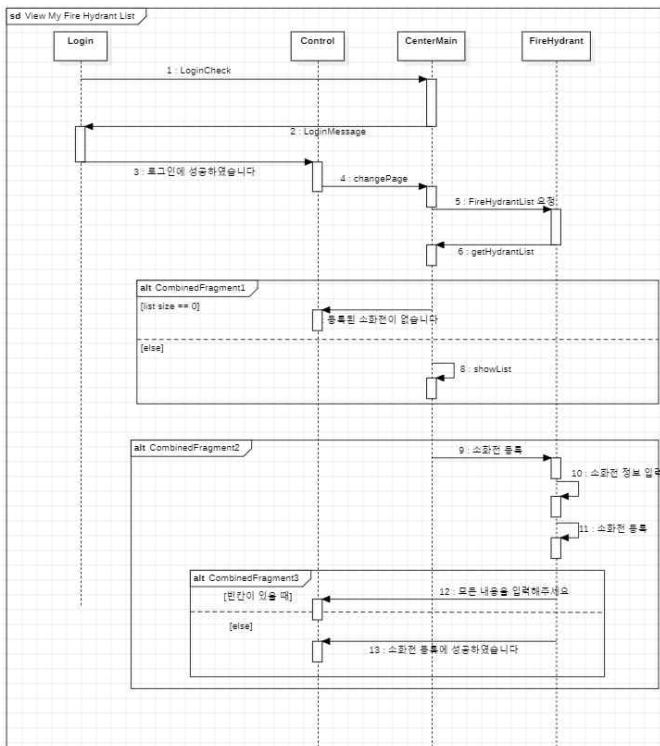
나의 장비 목록에서 할 수 있는 불용 신청에 대한 Sequence Diagram이다. 자신의 소
방 장비를 불용 신청해야 하는 경우, 해당하는 장비의 세부 사항에 진입하여 불용신청을
할 수 있다.

6) View disuse request



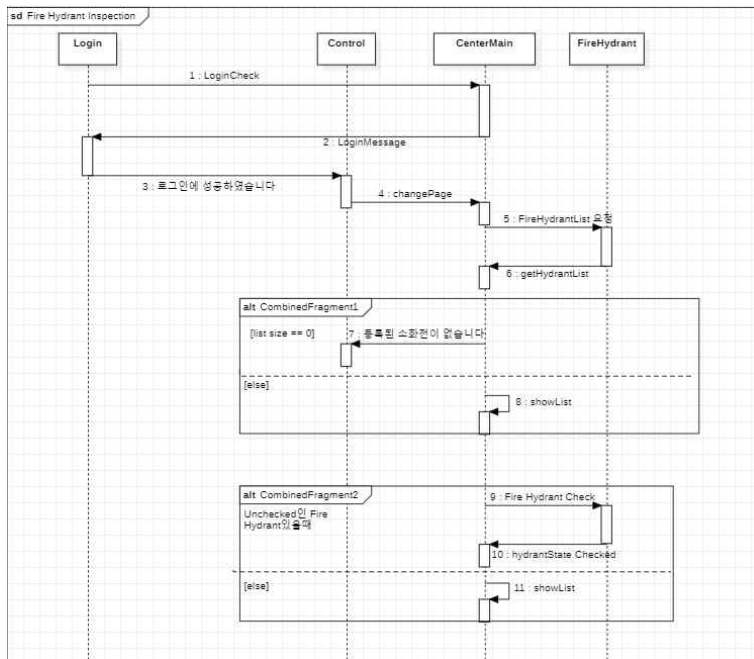
본서에서 센터에서 신청한 불용신청 된 장비 목록을 확인할 수 있는 Sequence Diagram이다. 본서에서는 서버에 등록된 모든 장비를 확인할 수 있으며, 불용신청 된 장비를 확인하고 해당 장비를 처리하기 위해 서버에서 삭제 할 수 있다.

7) View my fire hydrant list Fire hydrant registration



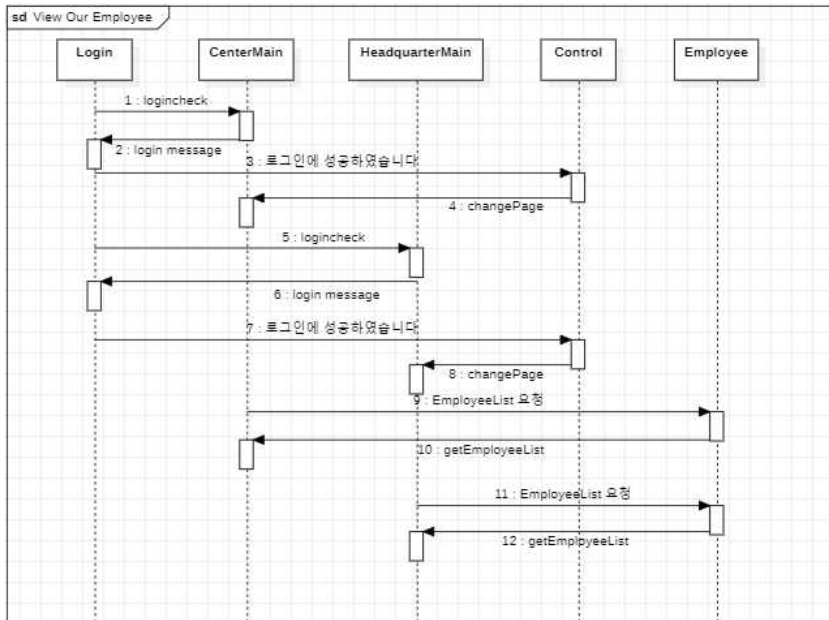
로그인에 성공한 후 센터에서 자신의 소화전 목록을 확인하고 등록할 수 있는 Sequence Diagram이다. 여기서는 자신의 관할인 소화전을 확인할 수 있으며, 소화전을 새로 등록하고자 하는 경우 소화전 등록에 필요한 정보를 작성하여 등록 버튼을 누르면 소화전 등록에 성공 하였다는 알림이 뜨면서 해당 소화전이 서버에 등록되게 된다.

8) Fire hydrant inspection



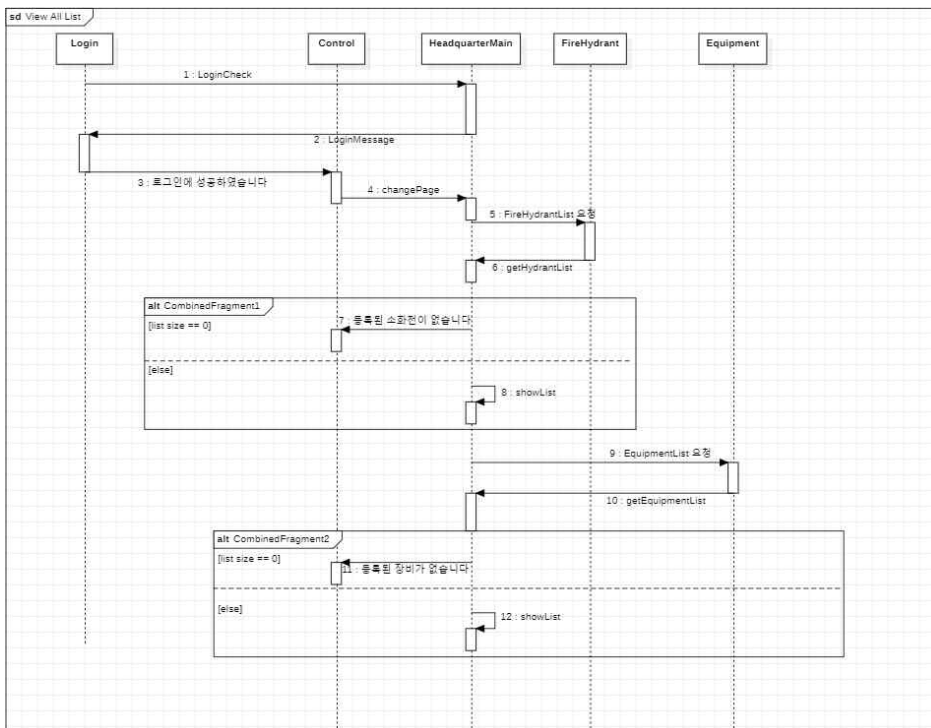
자신의 등록한 소화전을 점검하고 이를 확인하기 위한 Sequence Diagram이다. 사용자는 로그인을 하여 센터의 메인화면에 진입한 후 자신의 소화전 목록을 확인하고 난 후, 자신이 점검한 소화전을 선택하여 상태를 점검으로 변경하면 된다.

9) View our employee



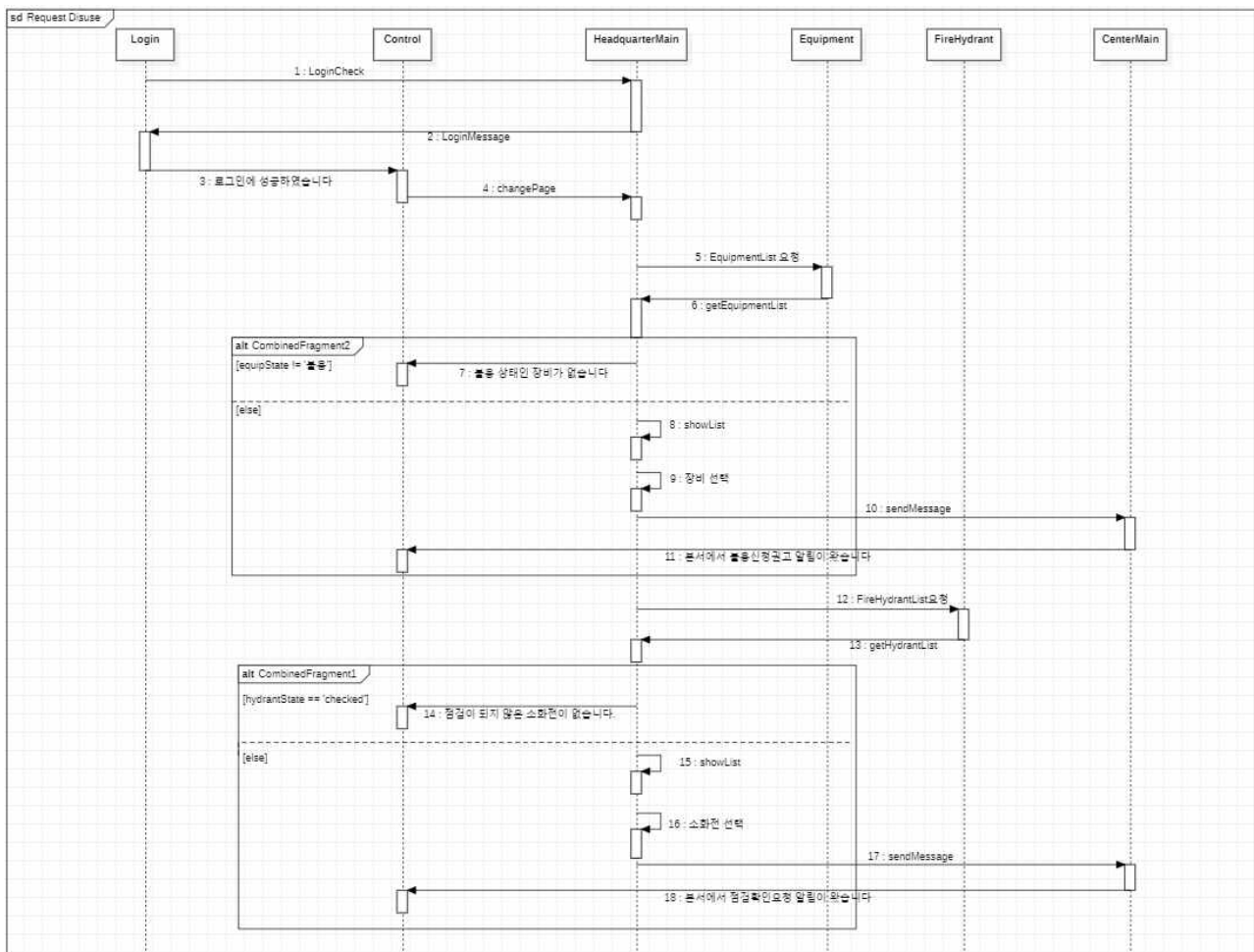
센터와 본부에서 둘 다 접근 가능한 직원 찾기에 대한 Sequence Diagram이다. Employee에서 서버에 등록되어있는 직원의 정보를 모두 보여준다.

10) View total equipment list and fire hydrant list

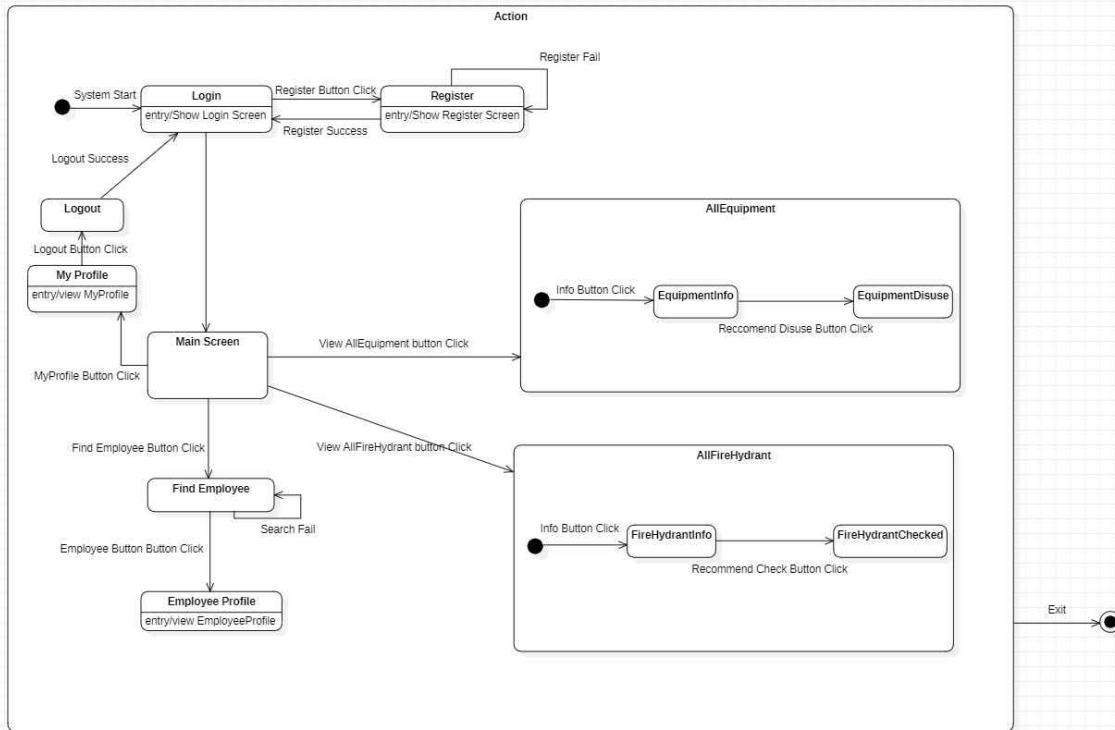


본부의 직원이 로그인한 후, 서버에 등록된 장비의 전체 목록이나 소화전의 전체 목록을 확인할 수 있는 Sequence Diagram이다. 만약 서버에 등록된 장비나 소화전이 존재하지 않을 경우, 각각 ‘등록된 장비가 없습니다’, ‘등록된 소화전이 없습니다’라는 알림이 뜨게 된다.

11) Recommend a request for equipment disuse hydrant inspection



본부의 직원이 로그인한 후, 전체 소화전 목록이나 전체 장비 목록에 진입하였을 때, 해당 소화전과 장비에서 점검이 필요하거나 불용신청이 필요한 소화전과 장비가 있을 경우 본부에서 센터로 소화전의 점검이 필요하다는 알림과, 장비의 불용신청을 권고하는 알림을 보내는 동작을 하게 된다.



본부 소속의 직원일 경우, 나의 프로필, 전체 장비 목록, 전체 소화전 목록, 직원 찾기를 실행할 수 있으며, 각각 자신의 정보, 서버에 저장된 전체 장비의 목록, 서버에 저장된 전체 소화전의 목록, 전체 직원 등을 확인할 수 있다. 전체 장비 목록을 실행하면 저장되어있는 장비가 모두 뜨게 되고, 불용이 필요한 장비의 경우 해당 장비를 가진 직원에게 불용 신청 권고 알림을 보낼 수 있다. 전체 소화전 목록을 실행하면 서버에 저장된 소화전의 목록이 뜨게 되고, 점검이 필요한 소화전이 생길 경우 관할 소방관에게 점검 요청 알림을 보낼 수 있다.

5. Implementation requirements

H/W Requirements
(1) Android based Machine
S/W Requirements
(1) Android 7.0+
(2) SDK 24+

6. Glossary

클래스 다이어그램 (Class Diagram)
객체 지향 프로그래밍에서 클래스들과 그들 사이의 관계를 시각적으로 표현하는 도구
시퀀스 다이어그램 (Sequence Diagram)
객체 간 상호작용을 시간의 순서에 따라 표현하는 도구
데이터베이스 (Database)
테이블로 구성되며, 테이블 간의 관계를 정의하여 데이터의 모음을 저장하고 관리하는 시스템

7. References

[#1 안드로이드 앱 만들기]

<https://zzimo.tistory.com/51>

[데이터베이스와 테이블 생성]

<https://junyoung-developer.tistory.com/130>