

개인 프로젝트 발표를 시작하겠습니다. 목차 인데요, 시스템 선정에 대한것을 말씀 드리고 인셉션 페이지 엘라보레이션 페이지 순으로 발표하겠습니다.

저는 무인 경비 시스템을 선정했습니다. 신원 확인 시스템에 가까운 시스템인데요, 이 무인 경비 시스템을 간단한 프로토타입 까지 구현함을 목표로 잡았습니다.

먼저 기능적 요구사항과 비 기능적 요구사항 입니다. 해당 시스템의 가장 중요한 기능으로, 출입 기능, 등록 / 수정기능, 폐쇄 / 개방 기능이 있다고 생각했고 해당 시스템은 비기능적인 사항으로 편의성과 보안성을 가져야 한다고 생각했습니다.

(디스크립션 문서 펴고)

보시고 계신것은 유스케이스 디스크립션중 하나인 출입에 대한 디스크립션입니다. 위와 같은 형식으로 총 다섯가지 유스케이스를 작성하였고, 우선은 도서관에 대한 시스템을 예시로 작성 해봤습니다.

(다시 돌아와서)

작성한 유스케이스를 토대로 그린 유스케이스 다이어그램입니다. Access를 확장하는 두가지의 유스케이스와 권한 수정 그리고 폐쇄 / 개방의 유스케이스를 볼 수 있습니다.

시스템 시퀀스 다이어그램입니다. 네가지의 상황을 그려보았는데, 시스템이 이런식으로 돌아갈 것 같다 라고 제가 생각하고 그린 것입니다. 보시면 출입과 관련한 부분에서는 문을 열수 있는 상태인지 확인하는 함수와 문을 여는 함수가 필요하며 권한 수정 유스케이스에서는 정보를 갱신하는 함수, 전체적으로는 사용자의 신원을 확인하는 함수가 필요함을 볼 수 있습니다.

위 작업을 바탕으로 제가 필요하다고 생각하는 주요한 시스템 오퍼레이터 다섯가지 입니다.

보시는 두 다이어그램은 출입과 관련된 시퀀스 다이어그램입니다. 보시면 두 다이어그램은 동작이 매우 비슷한데요, 문을 여는 요청을 받으면 액세스 컨트롤이 도어 컨트롤의 isOpen을 통해 폐쇄 상태인지 아닌지를 파악한 후 상태에 따라 문을 열게 합니다. 하지만 들어가는 상황의 경우 해당 출입자의 신원을 확인해야 하므로 받은 정보를 가지고 신원을 찾고 받은 신원을 확인한 뒤 나머지 작업을 수행합니다.

다음 두 다이어그램은 권한 수정과 문 폐쇄/개방에 대한 다이어그램입니다. 두 유스케이스 모두 공통으로 먼저 작업을 요청한 사용자의 권한을 파악한 뒤 각 요청을 처리합니다.

마지막으로 클래스 다이어그램 입니다. 엔티티는 도어와 유저 두가지를 이용하였으며, 컨트롤의 경우 메인 처리를 담당하는 액세스 컨트롤과 문과 관련된 요청을 처리하는 도어 컨트롤을 두었고, 데이터 베이스와 소통하기 위한 액세스 스토어 클래스를 하나 두었습니다.