3.8 Organizing the Specific Requirements

이 목차에서는 Unified Modeling Language 와 tabular form을 이용하여 저희의 시스템 모델을 설명합니다. 각 단계에서 system의 abstract model들을 보실 수 있으며, 각 단계마다 다른 관점에서 접근하여 표현합니다.

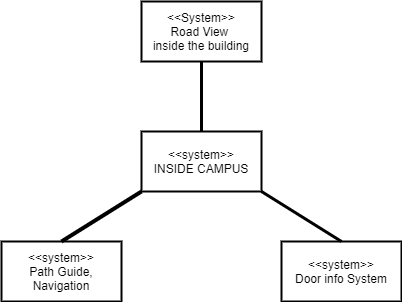
**3.8.1 Context model**

Figure : Context Model

**3.8.2 process model**

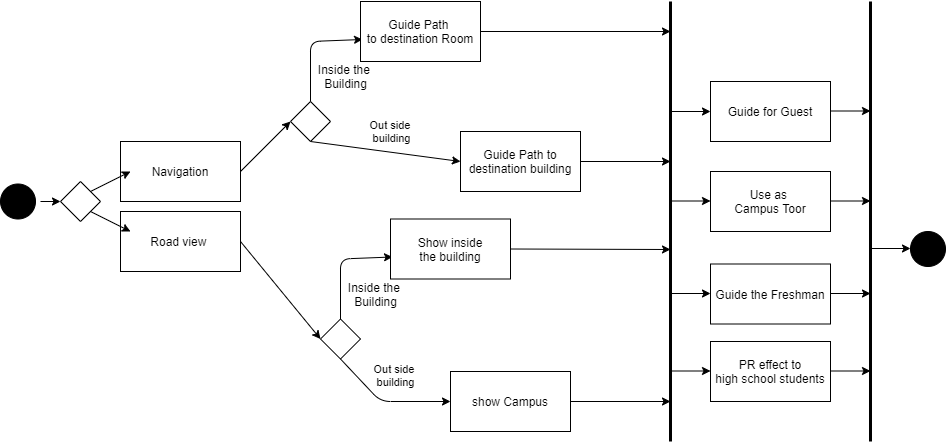
****

그림 : Process Model

**3.8.3 Interaction Model**

See 3.2.2. Use Case Diagram

**3.8.4 Behavior model – 각 기능에 따른 Sequence model들**

1) 경로 검색 기능

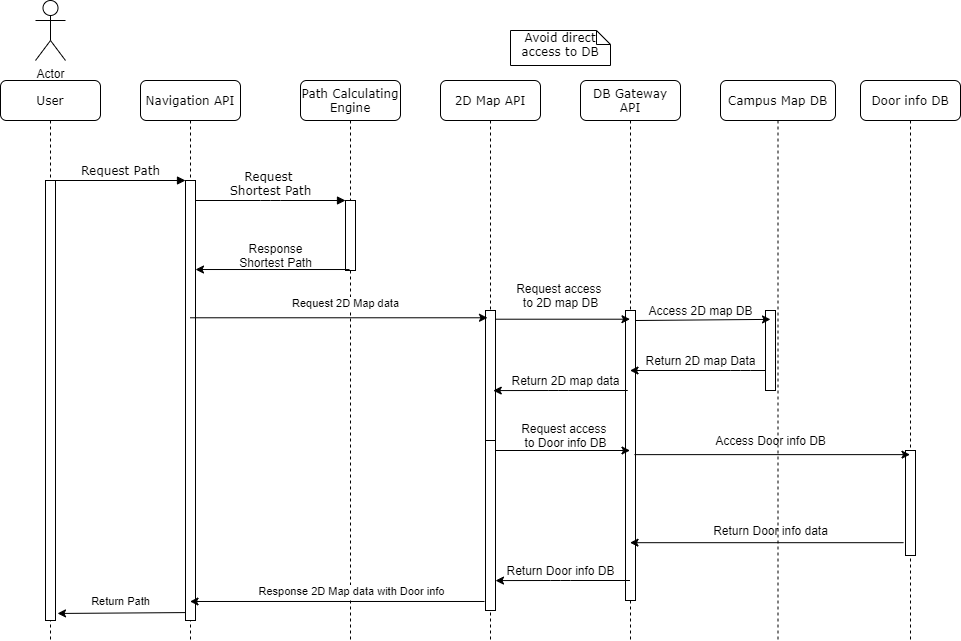


Figure : Sequence Model - Path Guide function

2) 건물 내부 로드뷰 기능

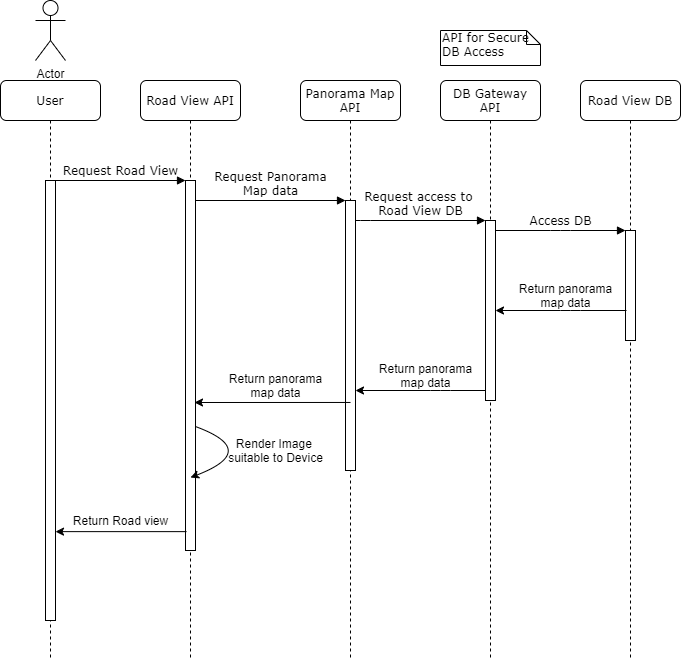


Figure : Sequence Model - Road View function

**3.9 System Architecture**

다음은 전체적인 개발 계획에서의 시스템의 구조를 도식화로 표현한 것입니다. 각 기능 별로 API 서버들이 분리되어 구성이 되며, 궁극적으로는 Cloud-Native 한 환경에서 운영하는 것을 목표로 합니다. 또한 유지보수 측면에서 긍정적인 효과를 볼 수 있는 Micro Service Architecture를 도입하여 각 API 서버들을 Docker로 유지합니다.

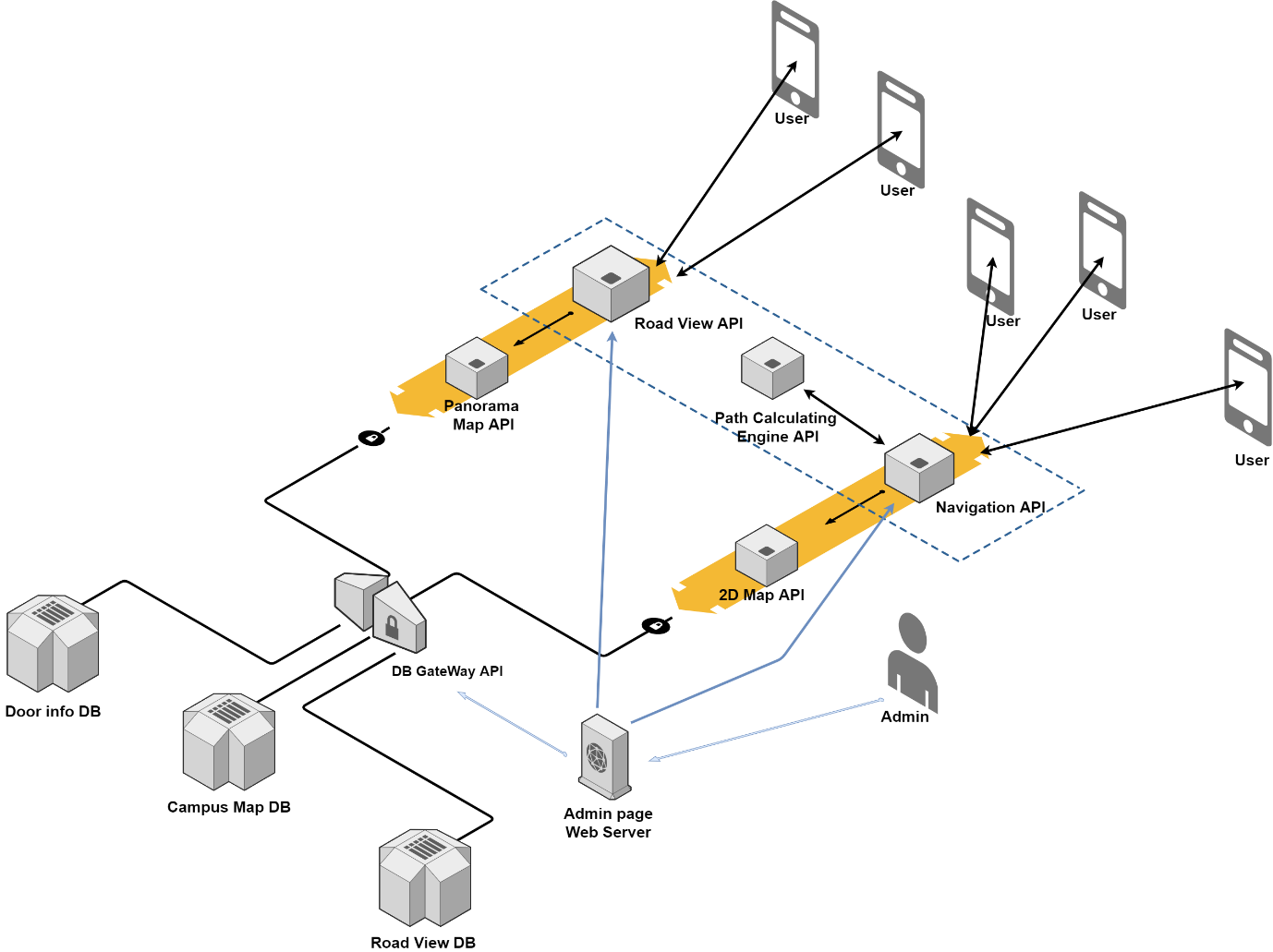


Figure : System Architecture

**3.10 System Evolution**

이 목차에서는 시스템이 기반이 되는 환경에 근본적인 추정들을 다룬다. 또한 하드웨어 유지보수나 요구사항의 변화로 인한 예상되는 수정 사항들도 다룬다.이 목차는 미래의 유지보수를 책임지는 개발자와 디자이너에게 유용할 것이다..

3.10.1 Limitation and Assumption

현재 우리의 앱은 성균관대학교 하나의 학교에만 서비스를 하게 된다. 따라서 다른 학교로도 추가적으로 제공할 가능성도 존재한다.

3.10.2 Evolutions of Hardware and Change of User Requirements

하드웨어가 바뀌거나, 교체가 필요한 경우에도 해당 교체 디바이스와 연결되는 API 또는 서버는 문제없이 작동할 것이다. Restful API구조와 Micro service Architecture로 구성을 하므로, Component 간의 Communication에 있어서 형식만 지켜서 요청 & 응답을 하게 된다면 문제는 없을 것이다. 추가적인 User requirements로 예상되는 것은 현재 기능은 학교 캠퍼스로 제한이 되어있지만, 학교 근처 맛집 거리, 카페, 등등 주변의 장소들로 가는 길안내와 로드뷰 기능도 요구될 수 있다. 따라서 우리는 미리 DB 구조를 설계할 때, 학교 건물만이 아닌, 다른 건물도 포함될 가능성을 염두에 두고 설계해야 하며, 학교 주변의 파노라마 사진들도 촬영해서 로드뷰 기능으로 사용할 준비를 해야 할 것이다.