* 1. **Software System Characteristics**

시스템의 중요한 특징은 non-functional requirements에서 나타난다. Non-functional requirement는 서비스, 시스템 및 프로세스의 제약이나 품질을 뜻한다. Non-functional requirement는 Product Requirement, Organization Requirements, External Requirement으로 나눌 수 있다.

* 1. **Product Requirements**

Product Requirements는 완성된 제품이 동작할 때 반드시 따라야 하는 특정한 방식들을 규정한 요구사항을 의미한다.

**3.7.1.1. Usability Requirements**

System은 버스 및 강의실의 좌석을 예약하는 시스템이다. 예약하는 과정이 복잡하지 않아야 하며, 버스 시간대, 강의 이름, 그리고 날짜 등 예약과 관련된 사항이 화면에 명확하게 드러나 직관적이어야 한다.

**3.7.1.2. Performance Requirements**

System은 여러 사용자가 동시에 접속하게 되는 좌석 예약 시스템이다. 따라서 예약 현황을 실시간으로 업데이트해주는 것이 매우 중요하다. 사용자의 Reservation에 관한 모든 활동은 서버의 데이터베이스에 10초 이내로 업데이트되어야 한다. 학생이 선점하려는 좌석이 비었는지 확인하는 활동은 5초 이내로 진행되어야 한다. 좌석 배치 화면은 3초 이내로 보여져야 한다. 유저의 예약 활동이 완료되면 유저의 메인 페이지에 완료된 예약 현황이 3초 이내로 반영되어야 한다.

**3.7.1.3. Dependability Requirements**

본 시스템은 교내 중요한 서비스를 사용하는 시스템이므로 원활하게 접속 가능해야 한다. 본 시스템의 Service Level Agreement(SLA) level은 98% 정도여야 한다. 또한 예약 과정에서 오류가 발생할 경우 사용자의 신뢰를 잃을 수 있기 때문에 1000개의 예약 중 최대 1개만 실패가 일어날 수 있다.

**3.7.1.4. Security Requirements**

본 시스템은 성균관대 구성원만 사용할 수 있다. 따라서 로그인을 할 때 교내 구성원 여부를 엄격하게 확인해야 한다. 시스템은 버스 및 강의실 예약 시스템 관련교직원에게만 시스템 관리자 권한을 주어야 한다. 시스템은 시스템 관리자가 처음 주어진 비밀번호를 관리자가 원하는 비밀번호로 바꿀 수 있도록 하고 바뀐 비밀번호로 접속할 수 있도록 해야 한다. 시스템은 사용자가 세 번 비밀번호를 실패했을 때 접근을 제한해야 한다. 본 시스템은 사용자의 개인정보가 유출되지 않도록 주의해야 하며 사용자가 시스템을 안전하게 이용할 수 있도록 데이터베이스 관리에도 유의해야 한다.

* 1. **Organizational Requirements**

Organizational Requirements는 개발하는 팀이나 조직의 영향을 받는 요구사항으로 조직 정책의 결과나 절차 등이 있다.

**3.7.2.1. Environmental Requirements**

본 시스템은 모바일 기기를 대상으로 한 어플리케이션이다. 따라서 사용자가 시스템에 접속하려면 상시 네트워크에 연결되어 있어야 한다. 서버 역시 사용자와 연결되고 데이터베이스를 관리해야 하기 때문에 상시 구동되어야 한다. 또한 시스템은 학생 정보, 강의실 정보, 교직원 정보 등을 얻기 위해 교내 데이터베이스에 의존하고 있다.

**3.7.2.2. Operational Requirements**

유저가 로그인을 할 때 교내 데이터베이스의 정보를 확인하게 된다. 시스템은 유저에게 정확한 예약 현황이 반영된 화면을 3초 내로 보여준다. 예약을 변경하면 데이터베이스는 5초 내로 변화를 반영하여야 한다.

**3.7.2.3. Developmental Requirements**

본 시스템은 안드로이드 기기를 대상으로 개발하는 어플리케이션이기 때문에 개발환경은 Android Studio IDE로 통일하고 언어는 Kotlin으로 제한한다. Maintainability를 위해서 코드를 작성할 때 한 함수 및 메소드 당 100줄 이상 작성하지 않는다. 변수와 함수명은 카멜표기법을 따른다.

* 1. **External Requirement**

External Requirements는 외부 요인으로부터 발생하는 요구사항으로 시스템 개발 프로세스에 대한 규제 및 제약사항 등이 있다.

**3.7.3.1. Regulatory Requirements**

처음 시스템을 사용하는 사용자에게 먼저 시스템은 개인정보를 수집하기 위해 동의를 구해야 한다. 그리고 정보를 외부에 제공하게 될 경우에도 사용자에게 동의를 먼저 구해야 한다.

**3.7.3.2 Safety/Security Requirements**

사용자의 개인정보가 외부로 유출되지 않도록 해야 한다. 사용자의 예약 현황에 오류가 생기지 않아야 한다.

* 1. **Organizing the Specific Requirements**

1. **Context Model**



1. **Process Model**



[Instructor]

[Student]

1. **Interaction Model**

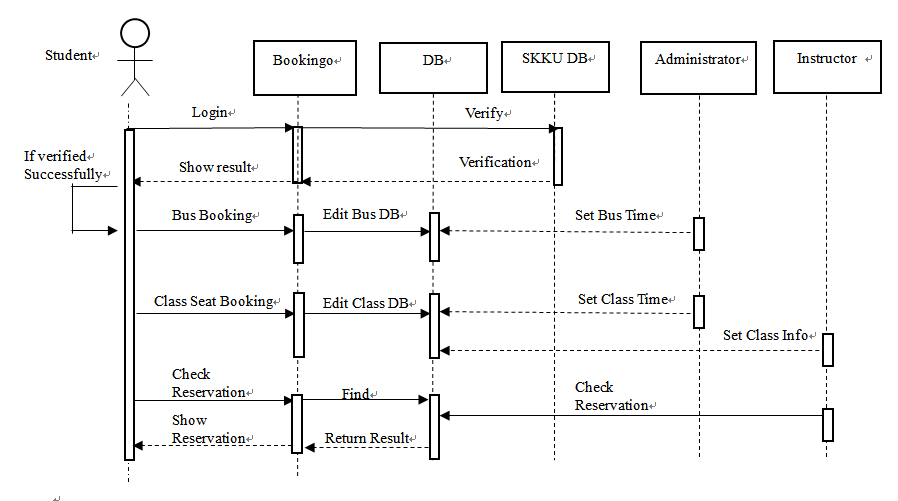
3.2.2. Use Case Diagram을 참고한다.

1. **Behavior Model**

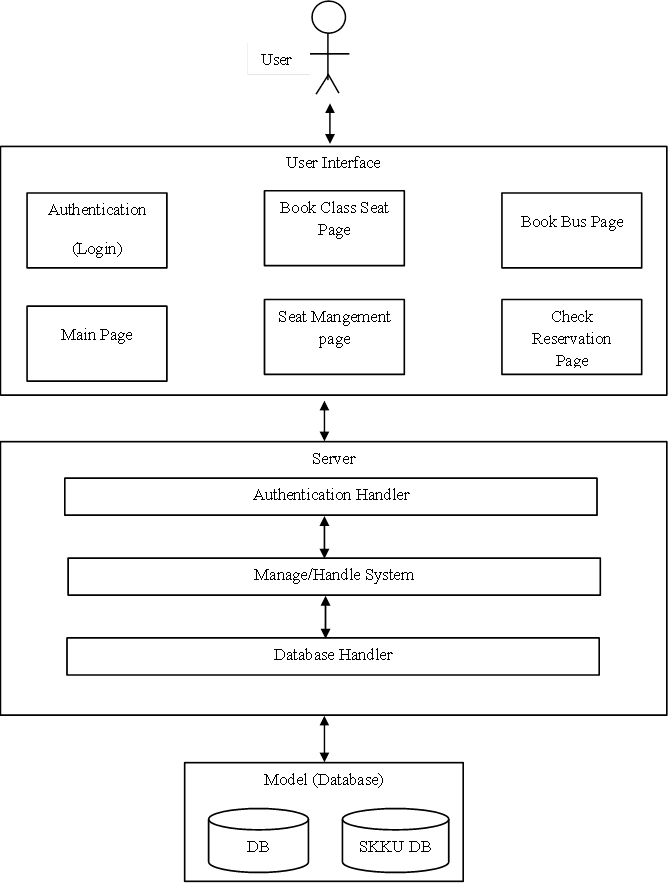
**3.8.4.1. Data Flow Diagram**

3.2.4. Data Flow Diagram을 참고한다.

**3.8.4.2. Sequence Diagram**



* 1. **System Architecture**



* 1. **System Evolution**

현 시스템은 완성된 시스템이 아니다. 이 섹션에서는 시스템의 기본적인 가정과 한계점을 살펴본다. 그리고 환경과 유저에 따라 생길 수 있는 요구사항 변화에 대해 알아본다.

1. **Limitation and Assumption**

본 시스템은 교내 킹고앱을 보완하는 서비스로 작용할 것을 가정하고 있다. 따라서 교내 데이터베이스의 강의실 정보, 교직원 정보, 학생 정보 등을 참고할 수 있고 사용할 수 있는 상황을 전제로 두고 있다. 버스의 운영시간, 버스 내의 좌석 개수, 강의실 내 좌석 개수 등도 모두 미리 파악된 것으로 가정하고 있다.

현재 예약과 관련된 서비스를 모두 한 곳에서 볼 수 없다는 한계점이 발생할 수 있다. 분산된 예약 서비스로 불편함을 느낄 수 있다.

1. **Evolutions of Hardware and Change of User Requirements**

본 시스템은 현재 모바일 기기를 대상으로 개발할 예정이다. 추후 접근성을 높이기 위해 web 플랫폼으로 확장한다면 접근성을 크게 높일 수 있다. 특히 학생들의 리스트를 스크롤해서 배정해야 하는 교수들의 경우 web으로 접근하는 방식을 더 선호할 가능성이 높다.

본 시스템은 현재 학생들과 시스템 사이 소통 창구가 부족한 상태이다. 이후 버스 추가 인설이나 강의실 환경이 실시간으로 변했을 때 빠른 피드백을 위해 소통 창구를 개설할 필요가 있다.

본 시스템은 현재 버스와 강의실 좌석이라는 예약시스템만 제공하고 있다. 교내 서비스 관련 다양한 예약서비스를 한 곳에 모아 통합 예약 서비스를 제공할 수 있는 방법을 고려해볼 필요성이 있다.