**3. Specific Requirements**

* 1. **Functional Requirements**
     1. **Data Dictionary**

[표 1] User

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK | Not Null | 유저의 성균인 아이디 |
| name |  | Not Null | 유저의 이름 |
| student\_faculty\_id |  | Not Null | 유저의 학번 / 교직원 번호 |
| user\_type |  | Not null | 유저의 유형 (학생 / 교직원) |

[표 2] Lecture

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK | Not Null | 강의 번호 |
| type | FK | Not Null | Ref. Seat arrangement type |
| name |  | Not Null | 강의 이름 |
| faculty |  | Not Null | 담당 교직원 이름 |
| time |  | Not Null | 강의 시간 |
| location |  | Not Null | 강의 장소 |

[표 3] Bus

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK | Not Null | 버스 번호 |
| type | FK | Not Null | Ref. Seat arrangement type |
| departing\_from |  | Not Null | 버스 출발점 |
| departure\_time |  | Not Null | 버스 출발 시간 |

[표 4] Seat\_Arrangement

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| type | PK | Not Null | 좌석 배치 유형 |
| arrangement |  | Not Null | 배치된 좌석의 위치 |

[표 5] Bus\_Reservation\_Status

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK/FK | Not Null | Ref. Bus |
| type | FK | Not Null | Ref. Seat arrangement |
| arrangement | FK | Not Null | Ref. Seat arrangement |
| status |  | Not Null | 좌석의 예약 상태 |

[표 6] Lecture\_Reservation\_Status

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK/FK | Not Null | Ref. Lecture |
| type | FK | Not Null | Ref. Seat arrangement |
| arrangement | FK | Not Null | Ref. Seat arrangement |
| status |  | Not Null | 좌석의 예약 상태 |

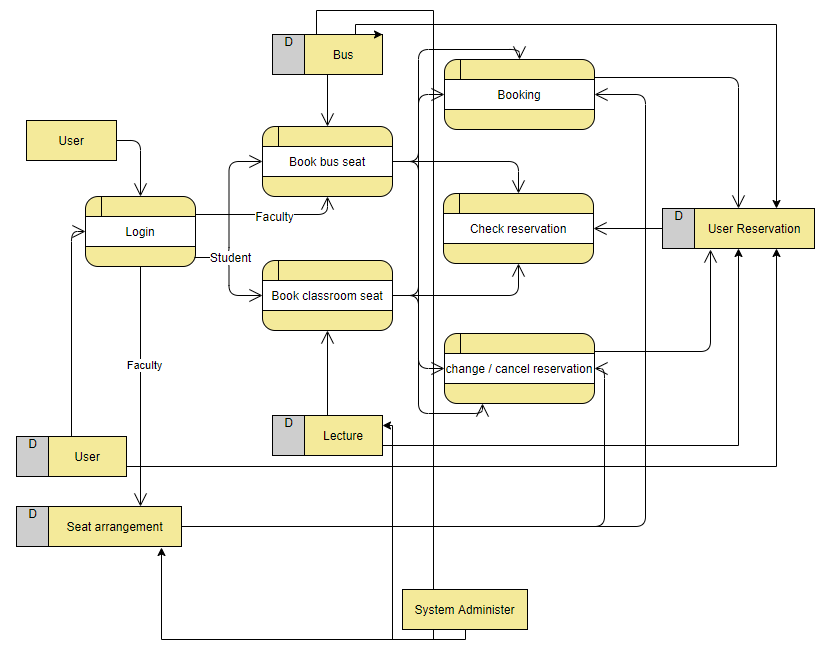
[표 7] Lecture\_Taken

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK/FK | Not Null | Ref. User |
| lecture\_id | FK | Not Null | Ref. Lecture |
| time | FK | Not Null | Ref. Lecture |
| location | FK | Not Null | Ref. Lecture |

[표 8] User\_Reservation

| **Field** | **Key** | **Constraint** | **Description** |
| --- | --- | --- | --- |
| id | PK/FK | Not Null | Ref. User |
| lecture\_bus\_id | FK | Not Null | Ref. Lecture and Bus |
| seat\_position |  | Not Null | 예약된 좌석의 위치 |

* + 1. **Data Flow Diagram**

 [그림 3] Data flow diagram

* 1. **Performance Requirements**

다음의 요구사항은 예상을 기반으로 작성되었으며 향후에 변동될 수 있다.

**3.3.1. Static numerical requirement**

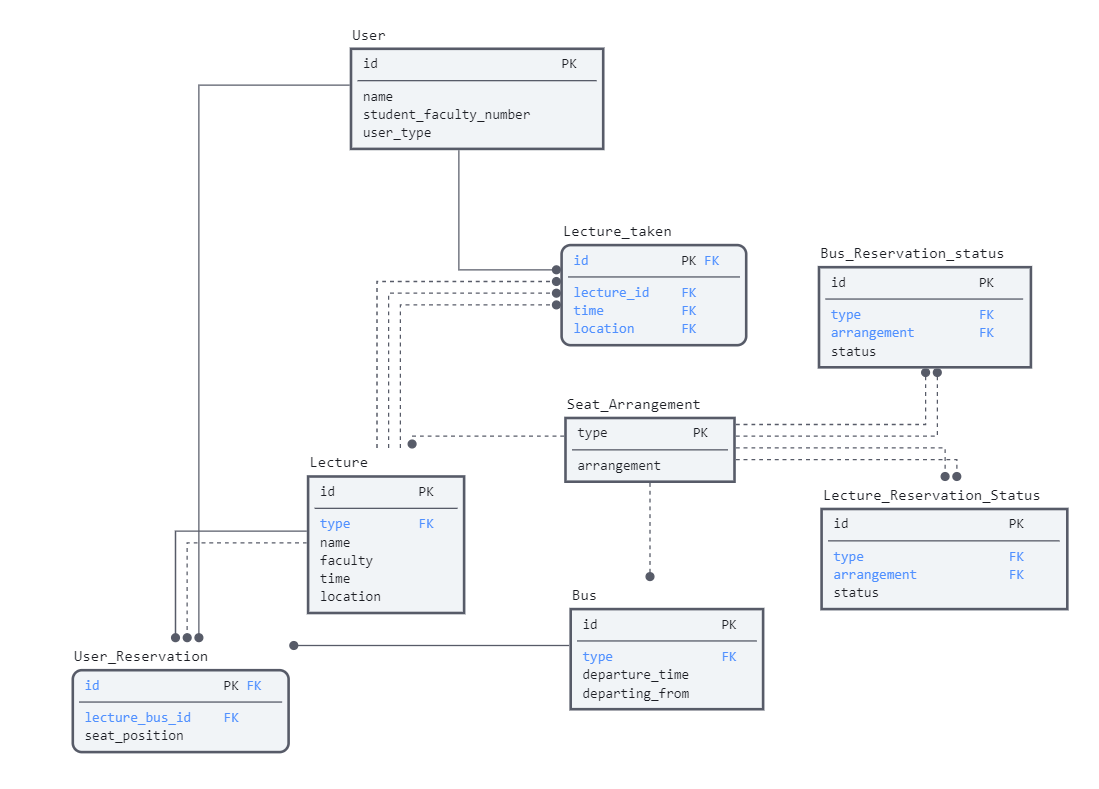
* 시스템은 하나의 터미널만 지원한다.
* 한 기기에서 한 명의 유저만 시스템을 이용할 수 있다. 여러 기기에서 동시 연결은 불가능하다.
* 시스템은 500명의 유저가 동시에 접속해도 원활하게 동작해야 한다. 시스템은 동시에 30000개의 데이터를 처리할 수 있어야 한다.
* 시스템이 원활하게 동작하기 위한 모바일 최소 사양은 1GB RAM과 512MB의 저장공간, 안드로이드 6.0 이상의 모바일 기기이다.

**3.3.2. Dynamic numerical requirement**

* 예약에 변동이 있다면 10초 이내에 데이터베이스가 업데이트되어야 한다.
* 유저의 예약 가능 여부는 데이터베이스와 비교하여 5초 이내에 확인되어야 한다
* 로그인 과정은 5초 이내에 완료되어야 한다
* 좌석 배치 UI는 3초 이내에 로드되어야 한다.
* 예약 현황은 3초 이내에 로드되어야 한다.
  1. **Logical Database Requirements**

모든 데이터는 Firebase의 Firestore를 이용하여 저장된다. 데이터는 JSON 객체의 형태로 저장되며 객체에 들어가는 Data type은 Integer, String, List가 있다.

저장된 데이터는 예약, 예약 확인, 예약 변경 및 취소 등에 이용된다.

 [그림 2] Entity Relationship Diagram

* 1. **Design Constraint**

Firebase가 Apache lisence를 따르고 있으므로 시스템 역시 Apache lisence를 따라야 한다. Apache liscence의 경우 소스 코드의 공개 의무나 상업적 이용에 제한을 두고 있지 않으므로 시스템이 apache liscence를 갖고 있음을 밝혀야 한다.

* 1. **Standards compliance**

시스템 내부의 모든 프로그램에서 사용되는 파일명, 변수명, 메소드명, 클래스명 등과 소스 코드의 구조는 Google developers에서 배포하는 Kotlin Style Guide를 따른다. Firestore를 이용할 때 데이터의 구조는 평면적으로 구성한다.