4. System Architecture - Frontend

4.1. Objectives

4.2. Subcomponents

이 형식 그대로 복사해서 써주세요. 글씨 크기 및 형식은 위에 스타일을 눌러서 적용시킬 수 있습니다.

스타일 적용 방법 : 스크롤해서 하이라이트된 채로 위에 1.큰제목 / 1.1.중간제목/1.1.1.세번째제목/1.1.1.1.가장작은제목 중 맞는 것을 누르면 바로 적용됩니다. 또는 쓰기 전에 스타일을 눌러주세요.

번호매기기 없이 앞에 숫자는 그대로 써주세요. 이는 이후 목차를 만들 때 필요합니다.

4.2.2. Booking

본문은 표준.

4.2.2.1. Attributes

Booking 객체가 갖는 특성은 다음과 같다

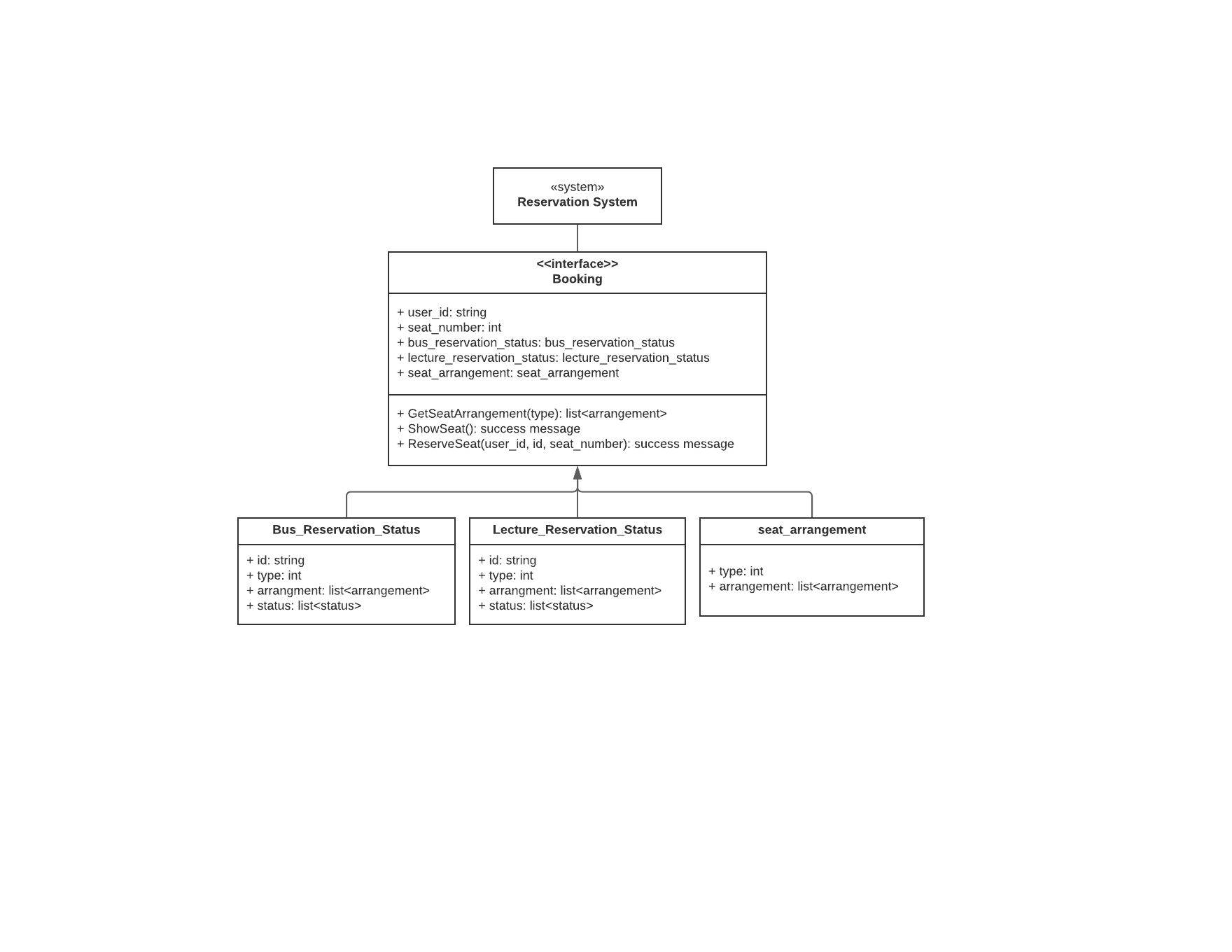
* **User id**: 유저의 kingo id
* **Reservation type**: 예약 유형 (강의실/버스)
* **Id:** 강의실/버스의 id
* **Time**: 강의 시간/버스 출발 시간
* **Location**: 강의 장소/버스 출발 장소
* **Seat type**: 좌석 배치 유형
* **Seat arrangement**: 좌석 배치

4.2.2.2. Methods

Booking class가 갖는 method는 다음과 같다

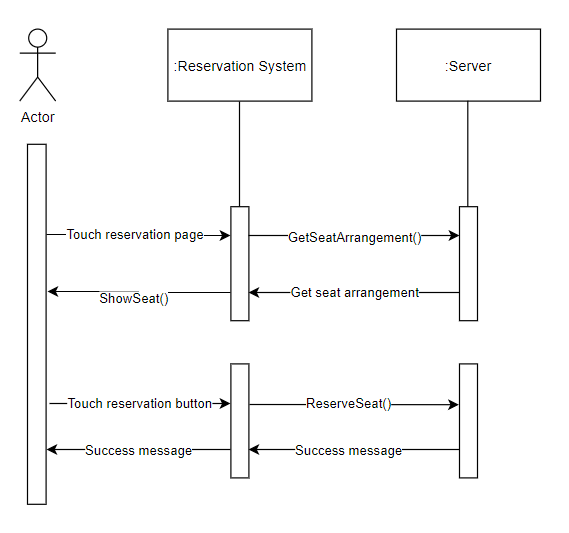
* GetSeatArrangement()
* ShowSeat()
* ReserveSeat()

4.2.4.3. Class Diagram



[그림 1] Class diagram – Booking

4.2.4.4. Sequence Diagram



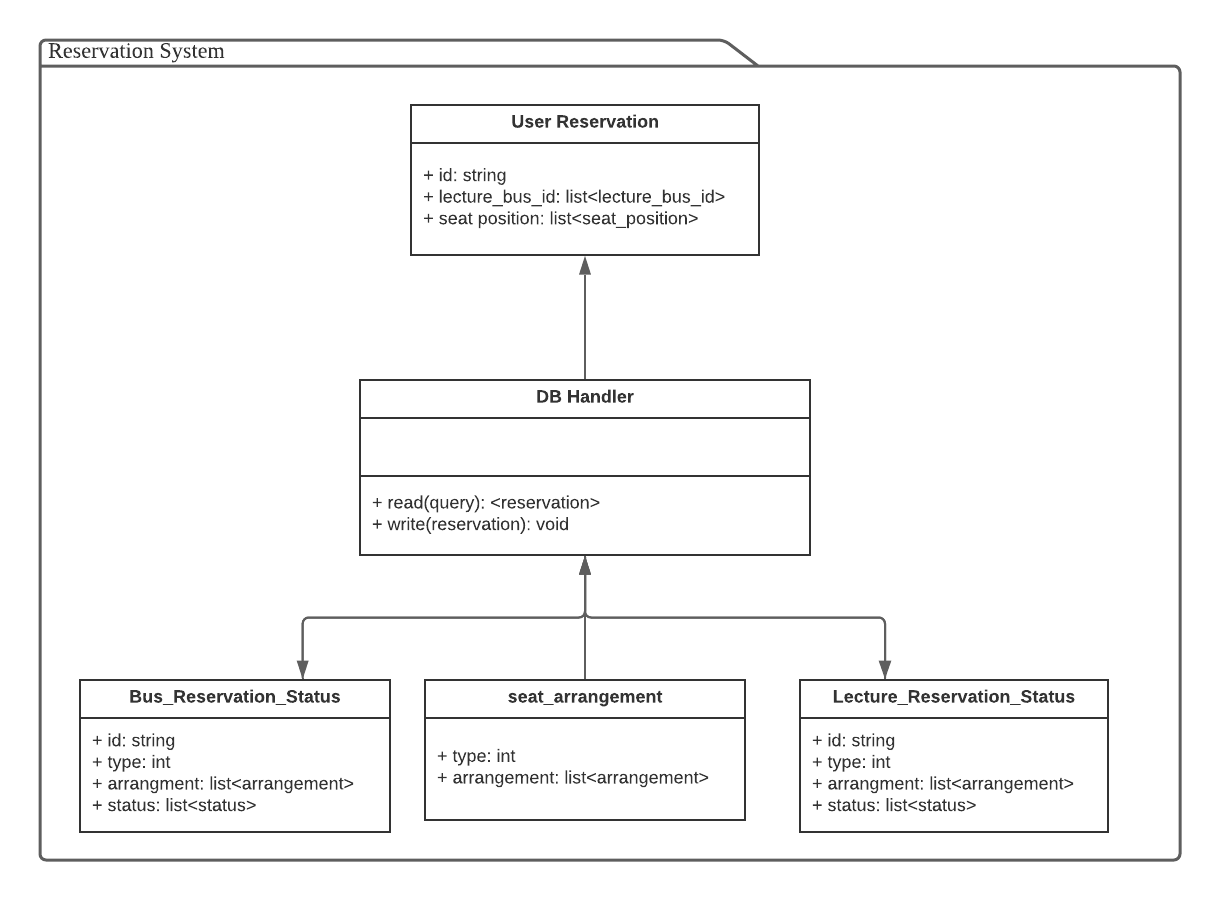
[그림 2] Sequence diagram – Booking

5. System Architecture – Backend

5.3. Subcomponents

5.3.1. Reservation System

5.3.1.2. Class Diagram

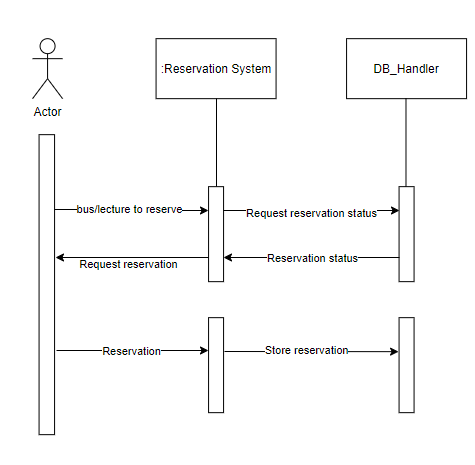


[그림 3] Class diagram – Reservation System

* Class Description

**Reservation System Class:** 예약을 담당하는 역할을 한다. 유저가 시스템에 예약을 요청하면 시스템은 유저 예약 목록에 해당 예약을 더해준다. 동시에 해당 예약 사항에 대해 버스나 강의의 좌석 예약 상태를 업데이트 한다.

5.3.1.2. Sequence Diagram



[그림 4] Sequence diagram – Reservation System

6. Protocol Design

6.1. Objectives

이 장은 각 시스템 간 상호작용에 쓰일 프로토콜의 구조와 각 인터페이스가 어떻게 정의되었는지에 대해서 다룬다.

6.2. JSON

JSON(JavaScript Object Notation)은 키-값의 한 쌍으로 이루어진 데이터를 전달하기 위해 텍스트를 사용해서 표현하는 개방형 표준 포맷이다.

Firebase의 실시간 데이터베이스에서 모든 데이터는 JSON 객체로 저장되기 때문에 데이터베이스를 JSON 트리로 볼 수 있다. 이 JSON 트리는 SQL 데이터베이스와는 다르게 테이블이나 레코드가 존재하지 않는다. JSON 트리에 데이터를 추가된 데이터는 이미 존재하는 JSON 구조에 연관된 키를 갖는 노드가 된다.

6.3. OAuth

OAuth란 인터넷 사용자들이 비밀번호를 제공하지 않고 다른 웹사이트 상의 자신들의 정보에 대해 웹사이트나 애플리케이션의 접근 권한을 부여할 수 있는 공통적인 수단으로 사용되는 개방형 표준이다. 이 방법은 여러 기업들이 사용하고 있으며 사용자들이 타사의 애플리케이션이나 웹사이트의 계정에 관한 정보를 공유할 수 있게 해준다.

Firebase 역시 OAuth 2.0을 이용하여 백엔드와 쉽게 연동하여 사용할 수 있다.

6.4. Log In

* Request

[표 1] Log In request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Protocol | OAuth | |
| Request Body | Id | User Kingo Id |
| Request Token | Token for OAuth |
| User | Basic information of the user |

* Response

[표 2] Log In response

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 400 | |
| Success  Response Body | Access Token | Token for access |
| Message | Success message |
| Failure  Response Body | Message | Fail message |

6.5. Reservation

6.5.1. Set Reservation

* Request

[표 3] Set reservation request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | POST | |
| URI | /user/:id/reservation | |
| Parameter | User | User id |
| Lecture Bus Id | 예약 대상 (강의, 버스) id |
| Seat position | 좌석 위치 |
| Header | Authorization | User authentication |

* Response

[표 4] Set reservation response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 409 (Conflict) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Fail message |

6.5.2. Check Reservation

* Request

[표 5] Check reservation request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | GET | |
| URI | /user/:id/reservation | |
| Request Body | id | id |
| Header | Authorization | User authentication |

* Response

[표 6] Check reservation response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 404 (Not Found) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Not Reserved |

6.7. Change Reservation

* Request

[표 7] Change reservation request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | POST | |
| URI | /user/:id/reservation | |
| Parameter | User | User id |
| Lecture Bus Id | 예약 대상 (강의, 버스) id |
| Seat position | 좌석 위치 |
| Header | Authorization | User authentication |

* Response

[표 8] Change reservation response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 409 (Conflict) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Fail message |

6.8. Cancel Reservation

* Request

[표 9] Cancel reservation request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | DELETE | |
| URI | /user/:id/reservation | |
| Request Body | User | User id |
| Lecture Bus Id | 예약 대상 (강의, 버스) id |
| Header | Authorization | User authentication |

* Response

[표 10] Reservation response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 404 (Not Found) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Fail message |

6.9. Seat Arrangement

6.9.1. Set Seat Arrangement

* Request

[표 11] Set seat arrangement request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | POST | |
| URI | /seat/:id/arrangement | |
| Request Body | arrangement | list<arrangement> |
| Header | Authorization | User authentication |

* Response

[표 12] Set seat arrangement response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 404 (Not Found) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Fail message |

6.9.2. Get Seat Arrangement

* Request

[표 13] Get seat arrangement request

|  |  |  |
| --- | --- | --- |
| **Attribute** | **Detail** | |
| Method | GET | |
| URI | /seat/:id/arrangement | |
| Request Body | id | lecture / bus id |
| Header | Authorization | User authentication |

* Response

[표 12] Get seat arrangement response

| **Attribute** | **Detail** | |
| --- | --- | --- |
| Success Code | 200 OK | |
| Failure Code | HTTP error code = 404 (Not Found) | |
| Success  Response Body | Message | Success message |
| Failure  Response Body | Message | Fail message |