



# **BOOKINGO**

## **Software Requirement Specification**

2021.4.25.

**Introduction to Software Engineering**

**TEAM 1**

**팀장 임서현**

**조원 강종현**

**조원 이승우**

**조원 임재현**

## Contents

1. Introduction .....	8
1.1. Purpose .....	8
1.2. Scope .....	8
1.3. Definitions, Acronyms, and Abbreviation .....	8
1.4. References .....	11
1.5. Overview .....	12
2. Overall Description .....	13
2.1. Product Perspective .....	13
2.1.1. System Interfaces .....	13
2.1.2. User Interfaces.....	13
2.1.3. Hardware Interfaces .....	14
2.1.4. Software Interfaces.....	14
2.1.5. Communications Interfaces.....	14
2.1.6. Memory Constraints.....	14
2.1.7. Operations .....	14
2.2. Product Functions .....	16
2.2.1. Login .....	16
2.2.2. Booking Shuttle Bus Seat.....	16
2.2.3. Booking Lecture Seat .....	16
2.2.4. Arranging Lecture Room Seating.....	17
2.2.5. Checking Reservation .....	17

2.2.6. Changing Reservation .....	17
2.2.7. Bus/Lecture slot creation/deletion.....	17
2.3. User Characteristics .....	18
2.3.1. System Administrator .....	18
2.3.2. Student.....	18
2.3.3. Faculty .....	18
2.4. Constraints .....	18
2.5. Assumptions and Dependencies .....	19
3. Specific requirements.....	19
3.1. External interface requirements .....	19
3.1.1. User interfaces.....	19
3.1.2. Hardware Interfaces .....	35
3.1.3. Software Interfaces.....	36
3.1.4. Communication Interfaces .....	36
3.2. Functional Requirements .....	37
3.2.1. Use Case.....	37
3.1.2. Use Case Diagram.....	47
3.2.2. Data Dictionary .....	47
3.2.2. Data Flow Diagram .....	50
3.2. Performance Requirements.....	51
3.3.1. Static numerical requirement .....	51
3.3.2. Dynamic numerical requirement.....	51

3.4. Logical Database Requirements .....	52
3.5. Design Constraint .....	52
3.6. Standards Compliance .....	53
3.7. Software System Characteristics .....	53
3.7.1. Product Requirements .....	53
3.7.2. Organizational Requirements .....	55
3.7.3. External Requirement.....	56
3.8. Organizing the Specific Requirements .....	56
3.8.1. Context Model.....	56
3.8.2. Process Model .....	58
3.8.3. Interaction Model .....	58
3.8.4. Behavior Model.....	58
3.9. System Architecture.....	60
3.10. System Evolution.....	61
3.10.1. Limitation and Assumption .....	61
3.10.2. Evolutions of Hardware and Change of User Requirements .....	61
4. Supporting Information .....	62
4.1. Software Requirement Specification .....	62
4.2. Document History.....	62

## List of Tables

[Table 1]	Definition .....	8
[Table 2]	Acronyms & Abbreviations.....	10
[Table 3]	User Interface of Login .....	19
[Table 4]	User Interface of Main Page.....	21
[Table 5]	User Interface of 강의실 좌석 예약(학생).....	22
[Table 6]	User Interface of 강의실 좌석 예약(학생) -2 .....	23
[Table 7]	User Interface of 강의실 좌석(교직원) .....	25
[Table 8]	User Interface of 강의실 좌석 배치(교직원).....	26
[Table 9]	User Interface of 강의실 좌석 형태 선택 팝업(교직원) .....	28
[Table 10]	User Interface of 학생 좌석 배정(교직원) .....	29
[Table 11]	User Interface of 학생 좌석 배정 팝업(교직원).....	31
[Table 12]	User Interface of 셔틀 좌석 예약 .....	32
[Table 13]	User Interface of 셔틀 버스 좌석 현황(학생, 교직원).....	33
[Table 14]	Hardware Interfaces .....	35
[Table 15]	Software Interfaces .....	36
[Table 16]	Communication Interfaces .....	36
[Table 17]	Use case of 로그인 및 로그아웃 .....	37
[Table 18]	Use case of 메인 페이지 접속 .....	39

[Table 19]	Use case of 셔틀버스 좌석 예약 .....	39
[Table 20]	Use case of 강의실 좌석 예약 .....	41
[Table 21]	Use case of 강의실 좌석 배치도 설정 .....	42
[Table 22]	Use case of 셔틀버스 좌석 예약 확인 .....	43
[Table 23]	Use case of 강의실 좌석 예약 확인 .....	44
[Table 24]	Use case of 셔틀버스 좌석 예약 변경/취소 .....	44
[Table 25]	Use case of 강의실 좌석 예약 변경 및 취소 .....	45
[Table 26]	Use case of 버스/강좌 타임슬롯 생성/삭제 .....	46
[Table 27]	User .....	47
[Table 28]	Lecture .....	48
[Table 29]	Bus .....	48
[Table 30]	Seat_Arrangement .....	48
[Table 31]	Bus_Reservation_Status .....	49
[Table 32]	Lecture_Reservation_Status .....	49
[Table 33]	Lecture_Taken .....	49
[Table 34]	User_Reservation .....	50
[Table 35]	Document History .....	62

## List of Figures

[Figure 1]	Use Case Diagram for BOOKINGO .....	47
[Figure 2]	Data Flow Diagram .....	51
[Figure 3]	Entity Relationship Diagram .....	52
[Figure 4]	Context Model.....	57
[Figure 5]	Process Model .....	58
[Figure 6]	Sequence Diagram.....	60
[Figure 7]	System Architecture .....	60

# 1. Introduction.

## 1.1. Purpose

본 문서는 학교 시설 내 좌석 예약 서비스를 제공하기 위한 시스템을 설계하는데 있어 필요한 소프트웨어 요구사항을 명세하기 위한 목적으로 작성되었다. 해당 문서는 성균관대학교 2021학년도 1학기 소프트웨어공학개론 Team1(강종현, 이승우, 임서현, 임재현)에 의하여 작성됐다. 본 문서의 대상 독자는 소프트웨어를 개발하는 Team1, 개발을 지도하는 지도교수, 그리고 본 소프트웨어의 요구사항을 파악하고 싶은 사용자들이다.

## 1.2. Scope

본 문서에서 서술할 어플리케이션 서비스의 이름은 BOOKINGO이다. BOOKINGO는 기존 KINGO앱에 없는 셔틀 버스와 강의실 좌석 예약 시스템을 운영한다. 학교 내 시설을 사용하면서 학생과 교직원이 느꼈던 불편한 점을 해소해주는 것을 목표로 한다. BOOKING은 크게 셔틀 좌석 예약, 강의실 좌석 예약 두 부분으로 구성된다. 학교 구성원임을 인증한 유저는 로그인 후 크게 셔틀 좌석 예약 서비스와 강의실 좌석 예약 서비스를 이용할 수 있다.

## 1.3. Definitions, Acronyms, and Abbreviation

다음 테이블은 이 문서에서 사용된 기술적 용어들의 정의를 정리했다.

[Table 1] Definition

Terms	Definitions
-------	-------------



안드로이드 (Android)	구글에서 개발한 모바일 운영체제
iOS	애플의 모바일 운영체제
서버(Server)	클라이언트에게 네트워크를 통해 정보나 서비스를 제공하는 컴퓨터 또는 프로그램
클라이언트(Client)	네트워크를 통하여 서버라는 다른 컴퓨터 시스템 상의 원격 서비스에 접속할 수 있는 응용 프로그램이나 서비스
사용자(User)	서비스를 이용하는 사람. 본 서비스에서는 학생 및 교직원이 있다.
시스템 관리자 (System Administrator)	서비스를 관리하는 사람. 예약 시스템을 모니터링하고 예약 창을 관리한다.
시스템(System)	BOOKINGO를 system으로 칭하며 문서에 서술되지 않은 BOOKINGO의 모든 기능을 포함한다.
터미널(Terminal)	System administrator가 system에 접근하기 위한 도구
데이터베이스 (Database)	System에서 사용되는 data의 집합으로 일반적으로 컴퓨터 시스템에 저장된 구조화된 정보 또는 데이터의 체계적인 집합을 의미한다.
클라우드	인터넷 기반의 컴퓨팅으로 인터넷 통신망에 접속해 컴퓨팅 자원을 원하

	는 대로 가져다 쓸 수 있는 서비스를 의미한다.
클라우드 데이터베이스	Firebase와 같은 클라우드 컴퓨팅 플랫폼에 상주하는 데이터베이스
Object	이름과 값으로 구성된 집합
Integer	정수
String	문자열
List	배열
토글(Toggle)	하나의 설정 값으로부터 다른 값으로 전환하는 것

다음 테이블은 머리글자(acronym) 및 약어(abbreviation)들의 정의를 정리했다.

[Table 2] Acronyms & Abbreviations

Acronyms & Abbreviations	Explanation
UX	User Experience
UI	User Interface
DB	Database

DBMS	Database Management System
MVC	Model – View – Controller
OS	Operating System
GUI	Graphical User Interface
IDE	Integrate Development Environment
RAM	Random Access Memory
GB	Gigabyte
MB	Megabyte
JSON	JavaScript Object Notation
N/A	Not Available or Not Applicable
GLS	Gold Lawn Square (Student Intranet)
ASIS	Advanced Sungkyunkwan Information Square

## 1.4. References

▷ “Software Requirements Specification for Gephi Version 0.92”, Last Edited: Feb. 2017

[https://gephi.org/users/gephi\\_srs\\_document.pdf](https://gephi.org/users/gephi_srs_document.pdf)

▷ 2020sprint\_41class\_team1, “SRS\_TEAM1.docx”, Last Edited: May. 14, 2020.

[https://github.com/skkuse/2020spring\\_41class\\_team1/tree/master/docs](https://github.com/skkuse/2020spring_41class_team1/tree/master/docs)

▷ Use Case Assumption vs Pre-Condition

<https://seilevel.com/requirements/use-case-assumption-vs-pre-condition>

▷ GenMyModel, used when making Use Case Diagram

<https://www.genmymodel.com/>

▷ Korean translation of IEEE Std 830. Contribute to pers0n4/srs-kor development by creating an account on GitHub.

<https://github.com/pers0n4/srs-kor>

## 1.5. Overview

본 Requirements Specification 문서는 4개의 section으로 구성된다. 첫 번째 Introduction 섹션은 개요로서 본 문서의 목적과 대상독자, 산출물의 범위와 문서를 이해하는데 필요한 용어, 참조문서, 그리고 문서의 구성에 대해서 다룬다. 두 번째 Overall Description 섹션에서는 종합적 기술로서 소프트웨어의 조망, 제품의 기능, 사용자의 특성, 제약사항, 가정 및 의존성, 요구사항 분배에 대해 기술한다. 세 번째 Specific Requirements 섹션에서는 소프트웨어의 외부 인터페이스 요구사항, 시스템의 특징, 성능에 대한 요구사항 등 소프트웨어에 대한 요구사항을 상세하게 기술한다. 마지막 Supporting Information 섹션에는 참고한 문서 형식과 문서 업데이트 타임라인이 포함된다.

## 2. Overall Description

### 2.1. Product Perspective

BOOKINGO 서비스는 학교 내 시설을 사용하면서 학생과 교직원이 느꼈던 불편한 점을 해소해주는 것을 목표로 한다. BOOKINGO 서비스를 통해 학생들은 자리가 없어서 셔틀을 타지 못했던 불상사를 예방할 수 있다. 또한 원하는 자리에 앉기 위해 강의실 앞에서의 대기 시간도 줄어들면서 복도의 혼잡도 문제도 개선된다. 교직원은 버스의 수요에 맞게 탄력적으로 셔틀을 운영할 수 있게 되고 출결 관리도 한결 편해진다.

#### 2.1.1. System Interfaces

BOOKINGO 시스템은 Firebase와 Android Studio IDE를 기반으로 하는 모바일 어플리케이션 서비스다. 구글에서 2016년에 발표했던 클라우드 데이터베이스인 Firebase는 서버 측 프로그래밍 없이도 어플리케이션을 만들 수 있도록 지원해준다. Firebase는 데이터베이스, 분석, FCM(Firebase Cloud Message), 파일 저장공간 등 많은 부분을 손쉽게 사용 가능하다. 또한 Real-time Database를 통해 어플리케이션 데이터를 자동 동기화할 수 있다. 시스템에서 예약에 관련된 필요한 정보들은 모두 Firebase의 Firestore를 이용하여 저장된다. Android Studio IDE를 이용할 때 파일명, 변수명 및 소스 코드의 구조는 모두 Kotlin Style Guide에 따라 작성된다.

#### 2.1.2. User Interfaces

유저는 모바일 기기로 시스템을 사용한다. 관리자는 Firebase로 기본적인 GUI를 제공 받는다. 예약창의 삭제, 관리 등의 기능을 수행할 수 있다. 학생 및 교직원은 해당 유저에 맞는 기능을 사용할 수 있는 화면이 나타난다. 학생의 경우 로그인을 한 후 메인 페이지에서 홈 화면, 강의실 좌석 예약 화면, 셔틀 버스 좌석 예약 화면을 통해 원하는

기능을 수행할 수 있다. 교직원의 경우 강의실 좌석 예약 화면에서 추가로 강의실 좌석을 배치하고, 학생들을 배정하는 기능을 수행할 수 있다.

### **2.1.3. Hardware Interfaces**

시스템은 안드로이드 모바일 기기를 대상으로 하며 최소 1GB RAM과 512MB의 저장공간이 필요하다.

### **2.1.4. Software Interfaces**

시스템은 최소한 Android 6.0 (API 23)을 요구하고 있다. 한 기기에서 한 명의 유저만 로그인하여 시스템을 이용할 수 있다.

### **2.1.5. Communications Interfaces**

시스템은 실시간 동기화가 필수적이므로 사용되는 모바일 기기가 인터넷에 연결되어 있어야 한다. 시스템은 500명의 유저가 동시에 접속해도 원활하게 동작해야 하며 3만 개의 데이터를 동시에 처리할 수 있어야 한다.

### **2.1.6. Memory Constraints**

시스템은 최소 1GB RAM이 있는 모바일 기기에서 실행되어야 한다. 또한 최소 512MB의 저장공간이 필요하다.

### **2.1.7. Operations**

#### **2.1.7.1 System Administrator**

- Bus/Lecture slot creation/deletion

좌석을 예약할 수 있는 타임 슬롯을 생성한다.

**2.1.7.2. Student and Faculty****- Login & out**

사용자는 로그인 시 성균인 아이디 및 비밀번호를 사용하여 접속한다.

**- Main page access**

메인 페이지에는 공지사항과 더불어, 금일의 예약 내역이 표시된다.

**- Booking shuttle bus seat**

해당 기능을 통해 사용자는 자신이 원하는 날짜 및 시간의 셔틀버스를 예약할 수 있다.

**- Booking lecture seat (Student only)**

학생 본인이 수강하는 수업의 좌석을 예약할 수 있는 기능이다.

**- Checking bus seat reservation**

사용자가 예약한 셔틀버스 좌석의 예약 현황을 확인할 수 있다.

**- Checking lecture seat reservation**

사용자가 기존에 예약한 강의실의 좌석을 확인할 수 있는 기능이다.

**- Checking/ cancelling bus seat reservation**

사용자가 예약한 셔틀버스의 좌석을 변경하거나 취소할 수 있는 기능이다.

**- Changing/ cancelling lecture seat reservation**

사용자가 예약한 강의실의 좌석을 변경하거나 취소할 수 있는 기능이다.

**- Arranging lecture room seating (Faculty only)**

교직원의 수업 방식에 적절한 형태로, 강의실의 좌석 배치를 설정할 수 있다.

## 2.2. Product Functions

### 2.2.1. Login

사용자는 로그인 시 성균인 아이디 및 비밀번호를 사용하여 접속한다. 만약 입력한 정보가 올바르다면, 사용자는 셔틀버스 예약/변경, 그리고 강의실 좌석의 예약/변경의 서비스를 사용할 수 있다. 사용자가 서비스 이용을 마치고 싶다면, 어플리케이션을 종료한다. 이 때 세션이 종료되면서 자동으로 로그아웃이 이루어진다.

### 2.2.2. Booking Shuttle Bus Seat

사용자는 자신이 원하는 날짜 및 시간의 셔틀버스를 예약할 수 있다. 행선지, 날짜, 시간을 선택하고, 좌석 조회 버튼을 누른다. 정상적으로 시간이 조회되었을 경우, 해당 셔틀 버스의 좌석 현황을 확인할 수 있다. 이 때, 음영처리 된 좌석은 기 예약되었거나, 선택이 불가능한 좌석으로, 클릭할 수 없다. 만석인 경우, 화면 하단의 “알림을 받겠습니다” 체크박스를 통해 좌석이 취소될 경우 또는 예약 시간 오픈 알림을 받을 수 있다. 해당 시간의 셔틀버스 배차가 취소되는 경우, “알림을 받겠습니다” 체크 여부와 무관하게, 사용자는 팝업 알림을 받게 된다. 좌석을 선택한 후, 좌측 하단의 “예약” 버튼을 클릭하면, 셔틀 버스의 날짜, 시간, 좌석 위치를 확인하는 팝업창이 나타나고, 최종적으로 예약을 결정하게 된다.

### 2.2.3. Booking Lecture Seat

학생 본인이 수강하는 수업의 좌석을 예약할 수 있다. 학생이 선택한 강좌가 본인의 좌석을 선택할 수 있는 강좌일 경우 강의실의 좌석 배치도 페이지로 이동하게 된다. 학생은 강의실의 좌석 중 음영처리 또는 색상으로 처리된 좌석을 제외하고 예약을 할 수 있다.



#### **2.2.4. Arranging Lecture Room Seating**

교직원은 수업 방식에 적절한 형태로 강의실의 좌석 배치를 설정할 수 있다. 일반적인 이론 수업의 경우 사각형 배열의 좌석 배치도를 설정할 수 있고 수강인원이 적은 경우 강의실 뒷좌석들을 예약 불가로 변경하는 등 적절한 좌석 배치를 설정할 수 있다. 실습, 팀, 실험 수업 등에서 조별로 좌석을 배치할 수도 있다.

#### **2.2.5. Checking Reservation**

사용자는 예약한 강의실 좌석 및 셔틀버스 좌석의 예약 현황을 확인할 수 있다. 메인 페이지에서 예약 내역의 정보를 확인할 수 있다. 당일 예약의 경우 메인 화면의 “Today”란에 예약 내역이 표시된다.

#### **2.2.6. Changing Reservation**

사용자는 예약한 좌석을 변경하거나 취소할 수 있다. 당일 예약의 경우 메인 화면의 예약 내역을 클릭해서 수정할 수 있다. 셔틀 또는 강의실 탭을 클릭해서 취소/변경 버튼으로 좌석을 변경하거나 취소할 수 있다.

#### **2.2.7. Bus/Lecture slot creation/deletion**

시스템 관리자는 좌석 예약 슬롯을 생성하거나 삭제할 수 있다. 강좌 좌석의 경우 ASIS강의 정보를 기반으로 하기 때문에 보강 일정이 아닌 이상 별도의 타임 슬롯 생성이 불필요하다. 다만 휴강과 같은 이유로 수업이 진행되지 않는 경우, 해당 타임 슬롯을 삭제할 수 있다. 버스 역시 축제 등의 이유로 타임 슬롯을 추가해야 하는 경우가 생길 수 있다. 운행을 불가피하게 하지 못하게 되는 경우 타임 슬롯을 삭제해야 할 수 있다.

## 2.3. User Characteristics

### 2.3.1. System Administrator

시스템 관리자는 시스템 관리 권한을 가지고 있는 교직원이다. 시스템 관리자는 예약창을 생성하고 삭제할 수 있다. 시스템 관리자는 데이터 베이스를 관리할 수 있는 능력을 가지고 있어야 하며 셔틀 버스 등 교내 서비스와 관련 있는 인물이어야 한다.

### 2.3.2. Student

학생은 가장 일반적인 유저다. GLS학생인트라넷에 정보가 있는 교내 학생이며 성균관대학교 학생 데이터베이스와 연동되어 로그인을 할 수 있다. 앱에 접속해서 강의실 좌석 및 버스 좌석 예약을 할 수 있고 예약 현황을 볼 수 있다.

### 2.3.3. Faculty

교직원은 강의실 예약에 대해 특수한 권한을 가진 유저다. Faculty는 교수는 물론이고, 조교, 그리고 교직원들 모두 아우르고 있다. 교직원은 버스 좌석 예약에 대해서는 학생과 같은 기능을 사용할 수 있다. 수업에 관련된 권한을 갖고 있는 조교나 교수의 경우 본인이 맡은 강의에 대해서 강의실 좌석 범위를 지정할 수 있다.

## 2.4. Constraints

시스템은 여러 가지 제약사항을 지키면서 서술된 내용을 바탕으로 구현되어야 한다.

- 시스템은 하나의 터미널만 지원한다.
- 시스템이 원활하게 동작하기 위한 모바일 최소 사양은 1GB RAM과 512MB의 저장공간이 있어야 한다.
- 예약에 변동이 있다면 10초 이내에 데이터베이스가 업데이트되어야 한다.

- 좌석 배치 UI는 3초 이내에 로드되어야 한다.
- 모든 데이터는 JSON 객체의 형태로 저장되며 Data type은 Integer, String, List가 있다.
- Firebase가 따르는 Apache License를 따라야 한다.
- Google Developers에서 배포하는 Kotlin Style Guide를 따라 코드를 작성한다.
- 규제정책으로 저작권법을 준수해야 한다.
- 성균관대학교 구성원만 사용할 수 있으므로 로그인 시 교내 구성원임을 엄격하게 확인한다.
- 시스템은 상시 네트워크에 연결되어 예약 현황을 업데이트해야 한다.
- 사용자의 개인정보가 외부로 유출되어서는 안되며 개인정보를 수집하기 전 동의를 꼭 구해야 한다.

## 2.5. Assumptions and Dependencies

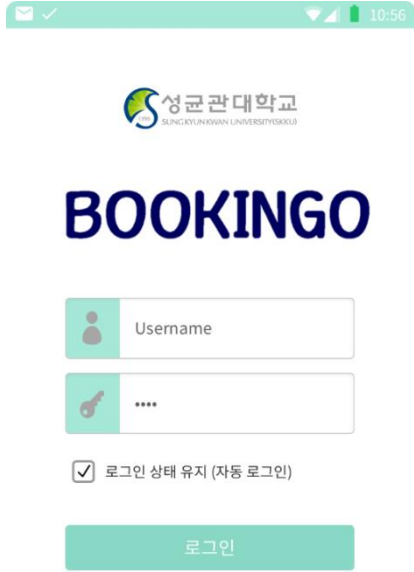
이 문서의 서비스들은 모두 안드로이드 플랫폼임을 가정한다. 따라서 안드로이드 운영 체제를 지원하지 않고 있는 모바일 기기에서는 작동이 어려울 수 있다. DBMS로는 firebase를 사용하고 client측 인터페이스를 작성할 때 Android Studio IDE를 사용한다. 교내 서비스이므로 교내 데이터베이스에 크게 의존하고 있다. 강의실 정보, 교직원 정보, 학생 정보 등을 참고할 수 있어야 한다.

## 3. Specific requirements

### 3.1. External interface requirements

#### 3.1.1. User interfaces

[Table 3] User Interface of Login

Name	UI-1 Login
Purpose / Description	BOOKINGO 에 접속하기 위한 로그인을 진행하는 페이지
Input source / Output destination	사용자(학생, 교직원) / 서버
Range / Accuracy / Margin of error	입력란 & 체크 박스 & 버튼의 범위 지정/ 사용자의 터치 정확도/ 터치 민감도의 오차 범위
Unit	입력란, 버튼, 체크 박스
Time/Velocity	<로그인> 버튼을 누르고 난 후 / 입력에 따른 즉각적 반응
Format and configuration of screen	

Data type	Image , Text, Query, Button
-----------	-----------------------------

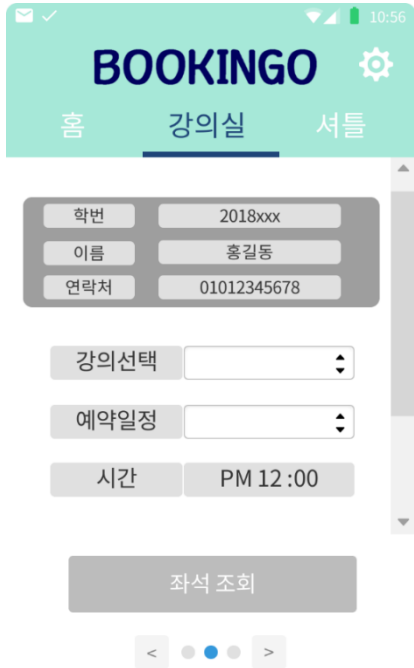
[Table 4] User Interface of Main Page

Name	UI-2 Main Page
Purpose / Description	로그인한 사용자는 Main Page 에서 오늘 예약된 좌석 현황과 관리자가 등록한 공지사항을 볼 수 있다.
Input source / Output destination	사용자 / 서버
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	<로그인> 버튼을 누르고 난 후 / 입력에 따른 즉각적 반응

Format and configuration of screen	
Data type	Image , Text

[Table 5] User Interface of 강의실 좌석 예약(학생)

Name	UI-3 강의실 좌석 예약(학생) - 1
Purpose / Description	<p>학생의 좌석 예약을 지원하는 페이지 / 학생은 좌석 예약을 원하는 강의를 선택할 수 있다. 선택을 완료하면 해당 강의에 대한 일정을 선택할 수 있다. 강의와 일정을 선택하면 학생은 해당 날짜에 맞는 강의 시간이 나타난다. 선택을 완료하고 좌석 조회 버튼을 누르면 좌석 조회 및 예약이 가능한 페이지로 이동한다.</p>
Input source / Output destination	사용자(학생) / 서버


	서버 / 사용자(학생)
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	<좌석조회> 버튼을 누르고 난 후 / 입력에 따른 즉각적 반응
Format and configuration of screen	
Data type	Image , Text, Query, Drop Box, Button

[Table 6] User Interface of 강의실 좌석 예약(학생) -2

<b>Name</b>	<b>UI-4 강의실 좌석 예약(학생) -2</b>
-------------	------------------------------


Purpose / Description	학생의 좌석 예약을 지원하는 페이지 / 학생이 좌석 예약을 할 수 있는 단계다. 이미 예약된 좌석은 회색, 이용이 불가능한 좌석은 빨간색, 교직원의 재량에 의해 자유석으로 지정된 좌석은 파란색으로 표현된다. 자유석은 청강생 및 예약을 하지 못한 학생들이 예약 없이 앉을 수 있는 좌석이다. 추가로 예약 시간 이전에 알림을 원하는 경우 선택에 따라 알림을 제공한다.
Input source / Output destination	사용자(학생) / 서버 서버 / 사용자(학생)
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	좌석을 클릭한 이후, <예약>, <취소/변경> 버튼을 클릭한 이후 / 입력에 따른 즉각적 반응



Format and configuration of screen	
Data type	Image , Text, Query, Button,


[Table 7] User Interface of 강의실 좌석(교직원)

Name	UI-5 강의실 좌석(교직원)
Purpose / Description	<p>교직원의 강의실 관련 업무를 지원하는 페이지 / 교직원을 위한 강의실 배정 화면에서 교직원은 강의실의 좌석 배치를 지정하거나 학생을 랜덤 또는 수동으로 배정할 수 있다. 교직원이 해당 강의와 일정을 선택하면 해당 강의의 시간이 자동으로 나타난다. 이후 교직원이 수행하고자 하는 기능을 클릭하면 해당 기능을 지원하는 페이지로 이동한다.</p>

Input source / Output destination	사용자(교직원) / 서버 서버 / 사용자(교직원)
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	<좌석 배치>, <수동 배정>, <랜덤 배정> 버튼을 클릭한 이후 / 입력에 대한 즉각 반응
Format and configuration of screen	
Data type	Image , Text, Drop Box

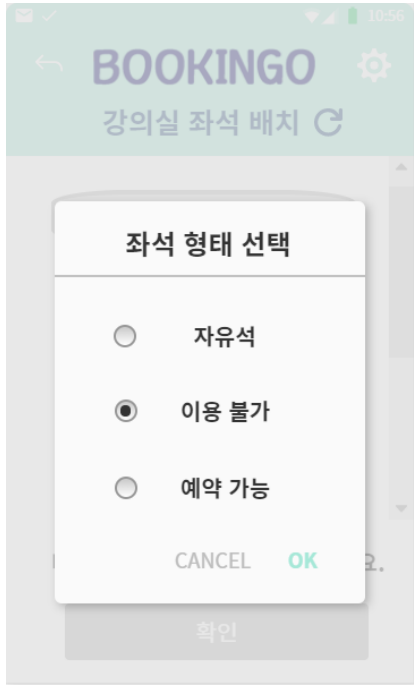
[Table 8] User Interface of 강의실 좌석 배치(교직원)

Name	UI-6 강의실 좌석 배치(교직원)
Purpose / Description	교직원이 강의실의 배치 수정을 할 수 있는 페이지 / 교직원이 강의실의 좌석을 배치하는 기능을 지원하는 페이지이다. 좌석 배열에 수정이 필요한 경우 드래그를 통해 좌석선택이 가능하다. 좌석의 상태를 자유석 / 이용 불가로 설정하고 싶은 경우 혹은 설정을 취소하고 싶은 경우에는 좌석을 클릭 후 나오는 팝업에서 선택 가능하다.
Input source / Output destination	사용자(교직원) / 서버 서버 / 사용자(교직원)
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	좌석을 클릭한 이후, <확인> 버튼을 클릭한 이후 / 입력에 따른 즉각적 반응

Format and configuration of screen	
Data type	Image , Text, Button,


[Table 9] User Interface of 강의실 좌석 형태 선택 팝업(교직원)

Name	UI-7 강의실 좌석 형태 선택 팝업(교직원)
Purpose / Description	교직원이 좌석의 유형을 선택할 수 있는 팝업 / 좌석은 자유석 또는 이용 불가를 선택할 수 있다. 유형을 바꾼 후에 다시 예약 가능한 좌석으로 바꾸고 싶다면 예약 가능을 선택할 수 있다.
Input source / Output destination	사용자(교직원) / 서버 서버 / 사용자(교직원)

Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	<OK>버튼을 클릭한 이후 / 입력에 따른 즉각적 반응
Format and configuration of screen	
Data type	Text, Query, Click Button,


[Table 10] User Interface of 학생 좌석 배정(교직원)

<b>Name</b>	<b>UI-8 학생 좌석 배정 (교직원)</b>
-------------	----------------------------

Purpose / Description	교직원이 수동으로 학생을 배정하는 기능을 지원하는 페이지 / 특정 학생을 배정하고 싶은 경우 좌석을 클릭하면 학생을 선택할 수 있는 팝업이 나타난다.
Input source / Output destination	사용자(교직원) / 서버
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	배정하고 싶은 좌석을 클릭한 후 / 입력에 따른 즉각적 반응
Format and configuration of screen	 <p>The screenshot shows a mobile app interface titled 'BOOKINGO' with the subtitle '학생 좌석 배정' (Student Seat Assignment). It features a grid of seats represented by colored squares (blue, red, grey) and empty squares. A confirmation button labeled '확인' (Confirm) is at the bottom. The text '배정하고 싶은 좌석을 선택하세요.' (Select the seat you want to assign) is displayed above the button.</p>
Data type	Text, Query, Button,

[Table 11] User Interface of 학생 좌석 배정 팝업(교직원)

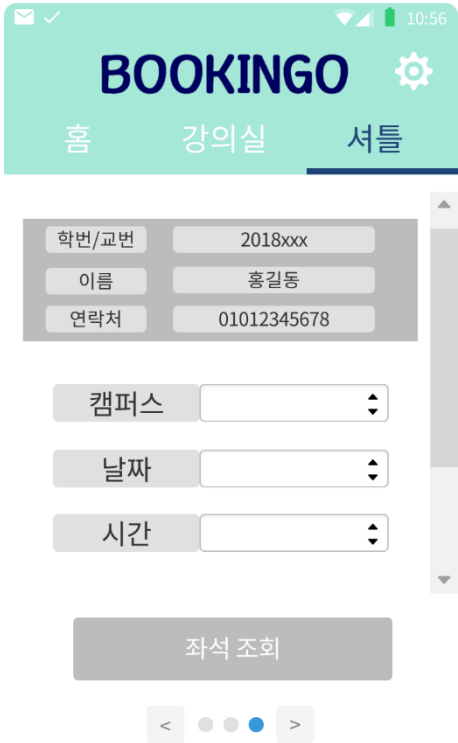
Name	UI-9 학생 좌석 배정 팝업(교직원)
Purpose / Description	학생 좌석을 직접 배정하는 팝업 / 교직원이 좌석을 선택하면 앞서 선택한 강의와 날짜에 대해 해당 강의를 수강하는 학생 리스트가 나오고 해당 리스트에서 학생 선택을 완료하면 학생 배정이 가능하다.
Input source / Output destination	사용자(교직원) / DB
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	<OK> 버튼을 클릭한 이후 / 서버와 유저 간의 통신 시간

Format and configuration of screen	
Data type	Text, Query, Button,

[Table 12] User Interface of 셔틀 좌석 예약

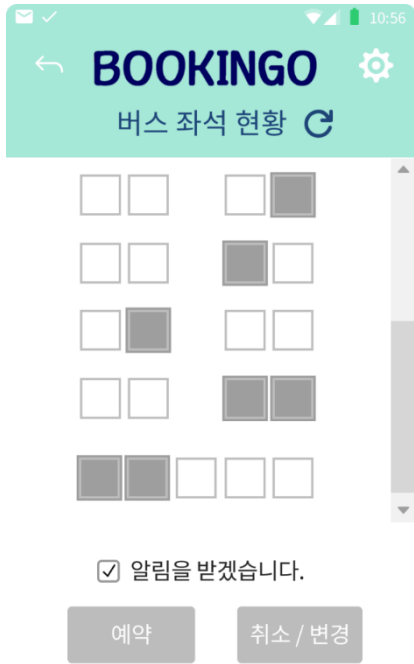
Name	UI-10 셔틀 버스 예약
Purpose / Description	<p>사용자에게 셔틀버스 예약 기능을 제공하는 페이지 / 사용자가 출발하려는 캠퍼스와 날짜, 시간을 선택할 수 있게 한다.</p> <p>&lt;좌석 조회&gt; 버튼을 누르면 해당 셔틀버스의 좌석 현황을 보여주고 예약을 할 수 있는 페이지로 이동한다.</p>
Input source / Output destination	사용자(학생, 교직원) / DB



Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	캠퍼스, 날짜, 시간을 선택하고 <좌석 조회> 버튼을 누를 때 까지 / 서버와 유저 사이 통신 시간
Format and configuration of screen	
Data type	Text, Query, Button

[Table 13] User Interface of 셔틀 버스 좌석 현황(학생, 교직원)

Name	UI-11 셔틀 버스 좌석 현황 (학생, 교직원)
Purpose / Description	셔틀 버스 좌석의 예약 현황을 표시하는 페이지 / 이미 예약된 좌석은 회색, 예약이 가능한 좌석은 빈칸으로 표현된다. 좌석을 선택하고 예약버튼을 누르면 예약이 완료된다. 추가로 예약시간 이전에 알림을 원하는 경우 선택에 따라 알림을 제공한다.
Input source / Output destination	사용자(학생, 교직원) / DB
Range / Accuracy / Margin of error	N/A
Unit	Screen
Time/Velocity	좌석 선택 후 예약 버튼 누를 때 까지 / 서버와 유저 사이 통신 시간

Format and configuration of screen	
Data type	Text, Button

### 3.1.2. Hardware Interfaces

[Table 14] Hardware Interfaces

Name	Hardware Interfaces
Purpose / Description	BOOKINGO 는 안드로이드 운영체제를 사용하는 스마트폰 기기가 필요하다.

### 3.1.3. Software Interfaces

[Table 15] Software Interfaces

Name	Software Interfaces
Purpose / Description	BOOKINGO를 운영하면서 필요한 데이터 관리 / 안드로이드 플랫폼 개발에 보편적으로 이용되는 Android Studio IDE를 사용하여 Client측 개발을 진행하고, 어플리케이션의 디자인을 쉽게 구현할 수 있게 한다.
Input source / Output destination	사용자(학생, 교직원) / 서버 서버 / 사용자(학생, 교직원)
Range / Accuracy / Margin of error	Firebase에 성능 따라 변동이 나타남
Unit	Query
Time/Velocity	서버와 유저 사이 통신 시간
Format and configuration of screen	N/A
Data type	Query

### 3.1.4. Communication Interfaces

[Table 16] Communication Interfaces

Name	Communication Interfaces
Purpose / Description	사용자가 호스트에 대한 연결을 요청 / 쉽게 서버를 이용할 수 있는 Firebase를 이용하여 Server측 개발을 진행하고 실시간으로 동기화되는 데이터베이스를 사용할 수 있다. 실시간으로 예약 현황을 반영해야 하므로 모바일 기기는 네트워크와 상시 연결되어 있어야 한다.
Input source / Output destination	사용자(학생, 교직원) / 서버
Unit	Packet
Time/Velocity	서버와 유저 사이 통신 시간
Format and configuration of screen	N/A
Data type	Query

## 3.2. Functional Requirements

### 3.2.1. Use Case

[Table 17] Use case of 로그인 및 로그아웃

Use Case Name	로그인 및 로그아웃(UC-01, Login & out)
Actor	학생, 교직원

Description	<p>로그인은 학내 구성원 DB에 포함된 인원이 서비스 이용을 위해 접속을 하는 절차를 의미한다.</p> <p>로그아웃은 로그인 상태인 사용자가 시스템에서 나가는 절차를 의미한다.</p>
Normal Course	<p>&lt;로그인&gt;</p> <ol style="list-style-type: none"> <li>1. 사용자는 로그인 시 성균인 아이디 및 비밀번호를 사용하여 접속한다. 별도의 아이디 및 비밀번호를 사용할 필요는 없다. 정보를 입력 후, 로그인 버튼을 누른다. 이 때, 추후 자동 접속여부를 체크박스를 통해 설정할 수 있다.</li> <li>2. 만약 입력한 정보가 올바르다면, 사용자는 셔틀버스 예약/변경, 그리고 강의실 좌석의 예약/변경의 서비스를 사용할 수 있다.</li> </ol> <p>&lt;로그아웃&gt;</p> <ol style="list-style-type: none"> <li>1. 만약 로그인 시 “자동 로그인” 기능을 설정하지 않은 사용자가 서비스 이용을 마치고 싶다면, 어플리케이션을 종료한다. 이 때 세션이 종료되면서 자동으로 로그아웃이 이루어진다.</li> <li>2. 만약 “자동 로그인” 기능을 설정한 사용자가 별도의 로그아웃 절차 없이 어플리케이션을 종료할 경우, 로그아웃이 이루어지지 않는다. 이 경우, 메인 화면 우측 상단의 설정 탭에서 로그아웃 절차를 수행할 수 있다.</li> </ol>
Precondition	<p>&lt;로그인&gt;</p> <p>사용자의 정보가 학내 구성원 DB에 포함되어 있어야만 한다.</p> <p>&lt;로그아웃&gt;</p> <p>사용자가 현재 로그인된 상태이어야한다.</p>
Post condition	홈 화면으로 이동한다.
Assumption	N/A

[Table 18] Use case of 메인 페이지 접속

Use Case Name	메인 페이지 접속(UC-02, Main page access)
Actor	학생, 교직원
Description	로그인 절차 이후, 사용자는 메인(홈) 화면으로 이동하게 된다. 해당 페이지에는 공지사항과 더불어, 금일의 예약 내역이 표시된다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자 로그인이 진행된 이후, 자동으로 홈 화면으로 리디렉션 된다.</li> <li>2. 공지사항이 존재하는 경우, 홈 화면 하단에 표시된다. 만약 공지사항의 세부사항을 확인하고 싶다면, 해당 글을 클릭하여 확인할 수 있다.</li> <li>3. 해당 일자에 사용자가 예약한 강의실 및 셔틀 내역이 있는 경우 이를, 화면 상단의 “Today”란에 표시된다. 만약 예약의 상세한 내용을 확인하고 싶다면, 예약 내역을 클릭한다. 예약 확인 페이지로 이동하게 된다.</li> </ol>
Precondition	사용자가 로그인을 수행해야한다.
Post condition	<p>개별 공지사항을 클릭하였을 경우, 팝업으로 공지사항 상세내용을 전달한다.</p> <p>예약 내역을 클릭하였을 경우, 예약 확인 페이지로 이동한다.</p>
Assumption	N/A

[Table 19] Use case of 셔틀버스 좌석 예약

Use Case Name	셔틀버스 좌석 예약(UC-03, Booking shuttle bus seat)
---------------	---

Actor	학생, 교직원
Description	해당 기능을 통해 사용자는 자신이 원하는 날짜 및 시간의 셔틀버스를 예약할 수 있다. 또한, 좌석을 선택하는 것도 가능하다. “알림을 받겠습니다” 체크박스를 통해 빈 좌석, 배차 취소, 예약 시간 오픈의 알림을 받을 수 있다.
Normal Course	<ol style="list-style-type: none"> <li>1. 메인 화면의 상단의 “셔틀” 탭을 선택한다.</li> <li>2. 본인의 정보, 연락처를 확인한 후, 행선지, 날짜, 시간을 선택하고, 좌석 조회 버튼을 누른다.</li> <li>3. 만약 시간 조회에 실패하는 경우, 관리자에게 해당 타임 슬롯 개설을 요청할 수 있다.</li> <li>4. 정상적으로 시간이 조회되었을 경우, 해당 셔틀 버스의 좌석 현황을 확인할 수 있다.</li> <li>5. 이 때, 음영처리 된 좌석은 기 예약되었거나, 선택이 불가능한 좌석으로, 클릭할 수 없다.</li> <li>6. 만약 좌석 현황을 갱신하고 싶다면, 화면 상단 중앙 부분의 새로고침 버튼을 눌러 수행할 수 있다.</li> <li>7. 만석인 경우, 화면 하단의 “알림을 받겠습니다” 체크박스를 통해 좌석이 취소될 경우 또는 예약 시간 오픈 알림을 받을 수 있다.</li> <li>8. 해당 시간의 셔틀버스 배차가 취소되는 경우, “알림을 받겠습니다” 체크 여부와 무관하게, 사용자는 팝업 알림을 받게 된다.</li> <li>9. 좌석을 선택한 후, 좌측 하단의 “예약” 버튼을 클릭하면, 셔틀 버스의 날짜, 시간, 좌석 위치를 확인하는 팝업창이 나타나고, 최종적으로 예약을 결정하게 된다.</li> </ol>
Precondition	사용자가 예약하고자 하는 시간에 셔틀버스 타임슬롯이 열려 있어야 한다.



Post condition	사용자의 예약 정보가 예약 DB로 이관된다.
Assumption	N/A

[Table 20] Use case of 강의실 좌석 예약

Use Case Name	강의실 좌석 예약(UC-04, Booking lecture seat)
Actor	학생
Description	학생 본인이 수강하는 수업의 좌석을 예약할 수 있는 기능이다. GLS의 수업과 연동되어 해당 과목의 시간은 자동으로 표현된다. 예약하고자 하는 날짜는 달력형태로 제공된다.
Normal Course	<ol style="list-style-type: none"> <li>1. 메인 화면의 상단 탭 중, “강의실”을 선택한다.</li> <li>2. 사용자의 정보, 연락처를 확인한 후, 강좌, 날짜를 선택한다. 이 때 토글로 내려오는 강좌는 본인의 GLS 상 수강/청강 신청한 강좌로 제한된다. 선택 후, 하단의 “좌석 조회” 버튼을 누른다.</li> <li>3. 만약 교직원의 결정으로, 해당 강좌의 좌석 배정이 임의로 정해졌다면, “좌석을 선택할 수 없는 강좌입니다”라는 팝업창이 나타난다.</li> <li>4. 사용자가 본인의 좌석을 선택할 수 있는 강좌의 경우, 강의실의 좌석 배치도 페이지로 이동하게 된다.</li> <li>5. 강의실의 좌석 중 음영처리, 또는 색상으로 처리된 좌석은 선택할 수 없다. 이 때, 청색은 자유석, 적색은 기 예약된 좌석, 그리고 음영처리 된 좌석은 예약 불가 좌석임을 의미한다.</li> <li>6. 사용자의 좌석을 선택하였으면, 좌측 하단의 “예약” 버튼을 누른다.</li> <li>7. 최종적으로 사용자의 강좌, 날짜, 좌석 위치를 확인하는 팝업</li> </ol>

	창이 나타난다.
Precondition	예약하고자 하는 강좌의 예약 가능 여부, 타임슬롯 존재 여부, 좌석 배치도 형태가 확정되어 있어야 한다.
Post condition	사용자의 예약 정보가 예약 DB로 이관된다.
Assumption	N/A

[Table 21] Use case of 강의실 좌석 배치도 설정

Use Case Name	강의실 좌석 배치도 설정 (UC-05, Arranging lecture room seating)
Actor	교직원
Description	교직원의 수업 방식에 적절한 형태로, 강의실의 좌석 배치를 설정할 수 있다. 수업 수강 인원이 적은 경우, 강의실 뒷 부분을 예약 불가로 변경할 수 있다. 또한 일반적인 이론 수업의 경우, 사각형 배열의 좌석 배치도를 설정할 수 있다. 반대로 실습, 팀, 실험 수업의 경우, 좌석 배치도를 변경해 조별로 좌석을 배치하는 것 또한 가능하다.
Normal Course	<ol style="list-style-type: none"> <li>1. 메인 화면 상단의 “강의실” 탭을 선택한다.</li> <li>2. 사용자의 정보, 연락처를 확인한 후, 강의와 일정을 선택한다.</li> <li>3. 이후, 강의실의 기본 좌석 배치도를 유지한 채, 좌석을 임의로 배정하고 싶다면, 화면 좌측 하단의 “랜덤 배정”을 클릭한다.</li> <li>4. 강의실 좌석 배정을 직접 설정하고자 한다면, 화면 하단의 “수동 배정”을 클릭한다.</li> <li>5. 수동 좌석 배치를 선택했을 경우, 강의실 좌석 배치도 페이지로 이동한다. 기본 좌석 배치도가 우선 표시된다.</li> <li>6. 서비스에서 제공하는 배치도 템플릿을 활용해 설정하는 것이 가능하며, 반대로 화면의 좌석을 드래그 하는 방식으로 배치</li> </ol>

	<p>도를 만드는 것 또한 가능하다.</p> <p>7. 사용자의 선호에 맞추어 강의실 배치도가 만들어지면, 확인 버튼을 클릭해 이를 최종 확정한다.</p>
Precondition	해당 강의가 사전에 ASIS에 탑재되어 있어야 한다. 강의실의 인원 정보가 확보되어 기본 좌석 배치도를 제공할 수 있어야 한다.
Post condition	좌석 배치도가 예약 DB로 이관되며, 학생이 예약을 진행할 때의 배치도로 제공된다.
Assumption	N/A

[Table 22] Use case of 셔틀버스 좌석 예약 확인

Use Case Name	셔틀버스 좌석 예약 확인 (UC-06, Checking bus seat reservation)
Actor	학생, 교직원
Description	사용자가 예약한 셔틀버스 좌석의 예약 현황을 확인할 수 있다.
Normal Course	<ol style="list-style-type: none"> <li>1. 당일 예약의 경우, 메인 화면 “Today” 란의 예약 내역을 클릭한다.</li> <li>2. 이외 일자의 예약의 경우, 상단 “셔틀” 탭을 클릭한다.</li> <li>3. 화면 우측 하단의 취소/변경 버튼을 클릭한다.</li> <li>4. 티켓 형태로 각 예약 내역의 정보가 표시된다. (날짜, 시간, 좌석위치)</li> </ol>
Precondition	예약 DB에 사용자의 예약 내역이 존재해야 한다.
Post condition	N/A
Assumption	기존의 예약 내역이 존재해야 한다.

[Table 23] Use case of 강의실 좌석 예약 확인

Use Case Name	강의실 좌석 예약 확인 (UC-07, Checking lecture seat reservation)
Actor	학생, 교직원
Description	사용자가 기존에 예약한 강의실의 좌석을 확인할 수 있는 기능이다.
Normal Course	<ol style="list-style-type: none"> <li>1. 메인(홈) 화면 상단의 “강의실” 탭을 클릭하거나, 당일 강의의 경우, 메인 화면 “Today” 란의 예약 내역을 클릭한다.</li> <li>2. 우측 하단의 “취소/변경” 버튼을 클릭한다.</li> <li>3. 사용자가 예약한 강의실 좌석의 내역이 리스트 형태로 표시된다.</li> </ol>
Precondition	예약 DB에 사용자의 예약 내역이 존재한다.
Post condition	N/A
Assumption	기존 예약 내역이 존재한다.

[Table 24] Use case of 셔틀버스 좌석 예약 변경/취소

Use Case Name	셔틀버스 좌석 예약 변경/취소 (UC-08, Changing/cancelling bus seat reservation)
Actor	학생, 교직원
Description	사용자가 예약한 셔틀버스의 좌석을 변경하거나 취소할 수 있는 기능이다.
Normal Course	<ol style="list-style-type: none"> <li>1. 당일 예약의 경우, 메인 화면의 예약 내역을 클릭한다.</li> </ol>

	<ol style="list-style-type: none"> <li>이외 일자의 예약의 경우, 상단 “서틀” 탭을 클릭한다.</li> <li>화면 우측 하단의 취소/변경 버튼을 클릭한다.</li> <li>티켓 형태로 각 예약 내역의 정보가 표시된다. (날짜, 시간, 좌석위치)</li> <li>각 예약 내역을 클릭한 후, 날짜, 시간 및 좌석을 변경하거나 취소할 수 있다.</li> </ol>
Precondition	예약 DB에 사용자의 예약 내역이 존재해야 한다.
Post condition	예약의 변경된 사항이 새로이 예약 DB에 이관된다.
Assumption	변경하고자 할 예약이 존재해야 한다.

[Table 25] Use case of 강의실 좌석 예약 변경 및 취소

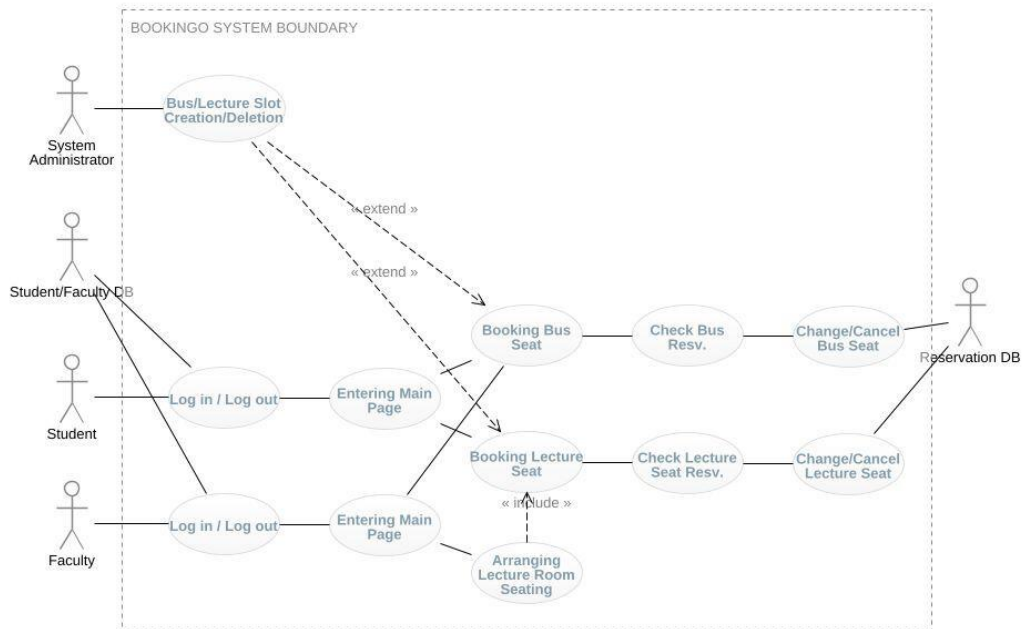
Use Case Name	강의실 좌석 예약 변경 및 취소 (UC-09, Changing/cancelling lecture seat reservation)
Actor	학생
Description	사용자가 예약한 강의실의 좌석을 변경하거나 취소할 수 있는 기능이다.
Normal Course	<ol style="list-style-type: none"> <li>메인(홈) 화면 상단의 “강의실” 탭을 클릭하거나, 당일 강의의 경우, 메인 화면 “Today” 란의 예약 내역을 클릭한다.</li> <li>우측 하단의 “취소/변경” 버튼을 클릭한다.</li> <li>사용자가 예약한 강의실 좌석의 내역이 리스트 형태로 표시된다.</li> <li>각 예약 내역을 클릭한후, 좌석 위치를 변경하거나, 취소하는 것이 가능하다.</li> </ol>

Precondition	예약 DB에 사용자의 예약 내역이 존재해야 한다.
Post condition	예약의 변경된 사항이 새로이 예약 DB에 이관된다.
Assumption	기존 예약 내역이 존재한다.

[Table 26] Use case of 버스/강좌 타임슬롯 생성/삭제

Use Case Name	버스/강좌 타임슬롯 생성/삭제 (UC-10, Bus/Lecture slot creation/deletion)
Actor	시스템 관리자
Description	좌석을 예약할 수 있는 타임 슬롯을 생성한다. 이렇게 만들어진 타임 슬롯이 있어야 사용자가 예약을 진행할 수 있다. 강좌 좌석의 경우, ASIS 강의 정보를 기반으로 하기 때문에 보강 일정이 아닌 이상, 별도의 타임슬롯 생성이 불필요하다. 다만 휴강과 같은 이유로 수업이 진행되지 않는 경우, 해당 타임 슬롯을 삭제하는 것은 가능하다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자로부터 타임슬롯 생성 요청이 들어오거나, 셔틀 배차가 취소되거나, 강좌가 취소되는 등 타임슬롯의 변경 사유가 발생한다.</li> <li>2. 관리자는 셔틀 버스의 경우 타임 슬롯을 생성 또는 삭제를 수행한다.</li> <li>3. ASIS와 연동된 강좌에서 특정 일자의 강의를 생성 또는 삭제하는 것이 가능하다.</li> </ol>
Precondition	타임슬롯의 변경 사유의 발생
Post condition	예약 DB에 타임슬롯의 변동이 반영된다. 이로 인해 영향을 받는 사용자에게 알림을 전송한다.
Assumption	N/A

### 3.1.2. Use Case Diagram



[Figure 1] Use Case Diagram for BOOKINGO

### 3.2.2. Data Dictionary

[Table 27] 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)

[Table 28] 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

[Table 29] 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

[Table 30] Seat\_Arrangement

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



[Table 31] 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

[Table 32] 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

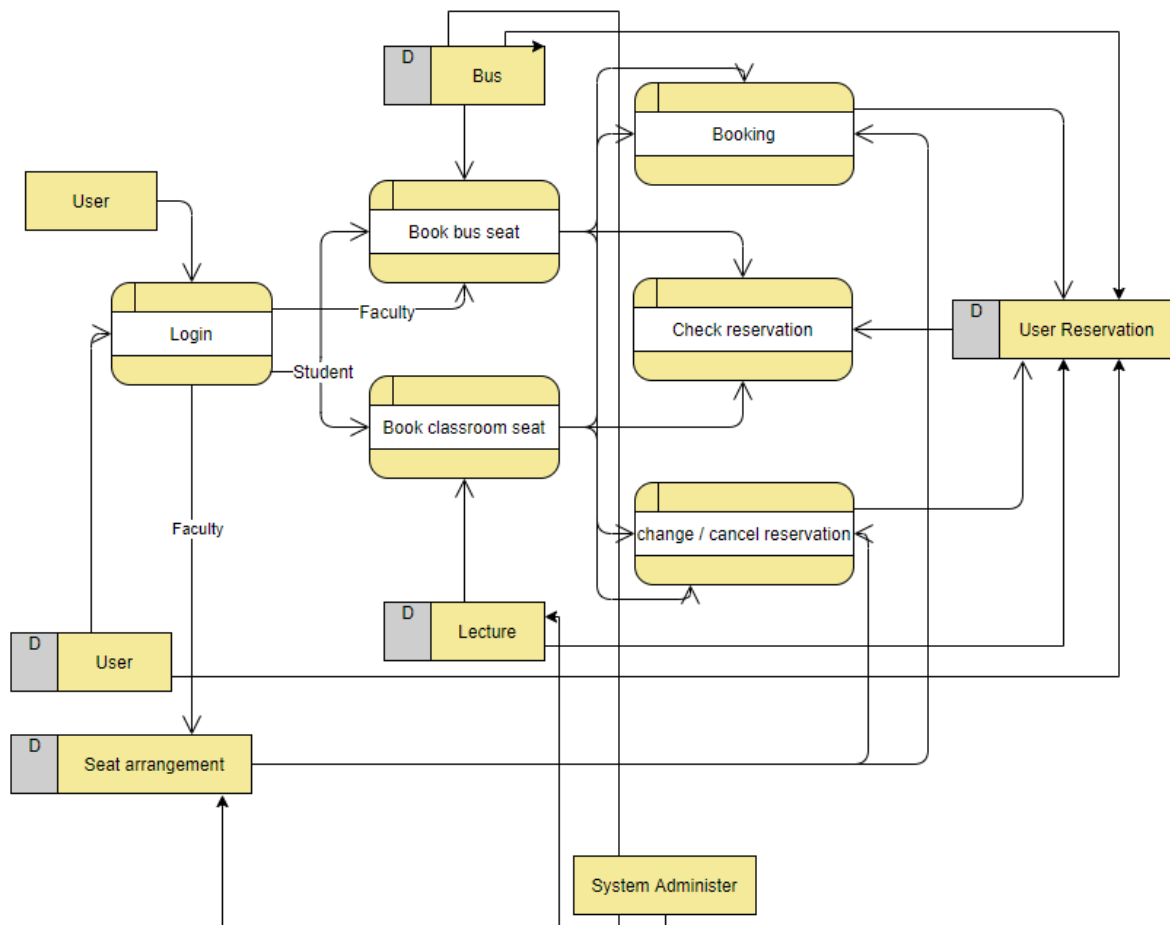
[Table 33] Lecture\_Taken

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

[Table 34] 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

### 3.2.2. Data Flow Diagram



[Figure 2] Data Flow Diagram

## 3.2. 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

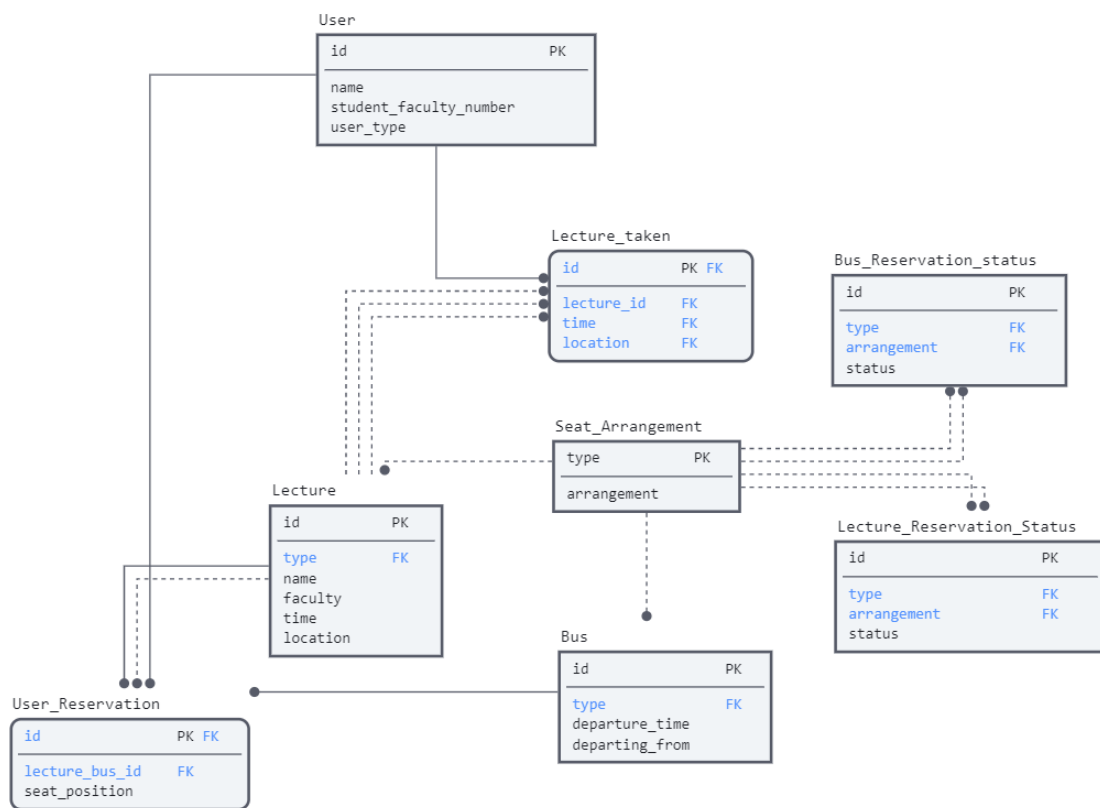
- 예약에 변동이 있다면 10초 이내에 데이터베이스가 업데이트되어야 한다.
- 유저의 예약 가능 여부는 데이터베이스와 비교하여 5초 이내에 확인되어야 한다
- 로그인 과정은 5초 이내에 완료되어야 한다
- 좌석 배치 UI는 3초 이내에 로드되어야 한다.

- 예약 현황은 3초 이내에 로드되어야 한다.

### 3.4. 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.



[Figure 3] Entity Relationship Diagram

### 3.5. Design Constraint

Firebase가 Apache license를 따르고 있으므로 시스템 역시 Apache license를 따라

야 한다. Apache license의 경우 소스 코드의 공개 의무나 상업적 이용에 제한을 두고 있지 않으므로 시스템이 Apache license를 갖고 있음을 밝혀야 한다.

### 3.6. 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. 프로젝트 관리를 위해서 Github와 Slack을 적극적으로 사용한다.

### 3.7. 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.

#### 3.7.1. Product Requirements

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

##### 3.7.1.1. Usability Requirements

시스템은 예약하는 과정에서 불편함이 없어야 한다. 버스 시간대, 강의 이름, 그리고 날짜 등 예약과 관련된 사항이 화면에 명확하게 드러나야 한다. 좌석을 선택할 때 터치에 민감하게 반응해야 하며 좌석 유형에 따라 색깔로도 구분되어야 한다. 화면에서 나타나는 버튼 및 아이콘도 어떤 기능을 수행할지 직관적으로 알 수 있어야 한다.

### 3.7.1.2. Performance Requirements

시스템은 여러 사용자가 동시에 접속하게 되는 좌석 예약 시스템이다. 따라서 예약 현황을 실시간으로 업데이트해주는 것이 매우 중요하다. 사용자의 예약에 관한 모든 활동은 서버의 데이터베이스에 10초 이내로 업데이트되어야 한다. 학생이 선택하려는 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

시스템은 교내 중요한 서비스를 사용하는 시스템이므로 원활하게 접속 가능해야 한다. 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. 서버가 불안정하여 로그인 실패가 나거나 예약 과정에서 접속 오류가 나는 경우가 일어나는 빈도는 적어야 한다. 예약 내역이 오류 없이 서버의 데이터베이스에 저장되어야 한다.

### 3.7.1.4. Security Requirements

시스템은 성균관대학교 구성원만 사용할 수 있다. 따라서 로그인을 할 때 교내 구성원 여부를 엄격하게 확인해야 한다. 시스템은 버스 및 강의실 예약 시스템 관련 교직원에게만 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. 시스템은 사용자의 개인정보가 유출되지 않도록 주의해야 하며 사용자가 시스템

을 안전하게 이용할 수 있도록 데이터베이스 관리에도 유의해야 한다.

### **3.7.2. Organizational Requirements**

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

#### **3.7.2.1. Environmental Requirements**

시스템은 안드로이드 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 로그인할 때 교내 인트라넷의 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**

시스템은 안드로이드 기기를 대상으로 개발하는 어플리케이션이기 때문에 개발환경은 Android Studio IDE로 통일하고 언어는 Kotlin으로 제한한다. Maintainability를 위해서 시스템 내부 모든 프로그램에서 사용되는 파일명, 변수명, 메소드명, 클래스명 등과 소스 코드의 구조는 Kotlin Style Guide를 따른다. 변수명과 함수명은 카멜표기법을 따른다. 코드를 작성할 때 한 함수 및 메소드는 100줄 이상 작성하지 않는다.

### **3.7.3. 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.

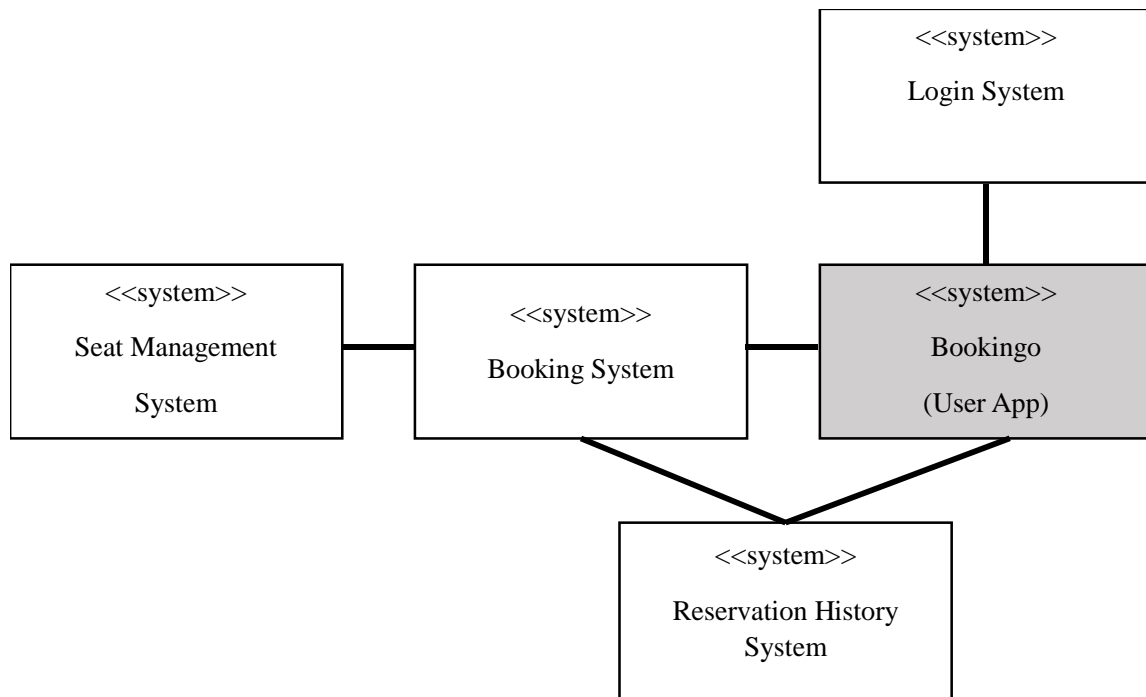
#### **3.7.3.2. Safety/Security Requirements**

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

## **3.8. Organizing the Specific Requirements**

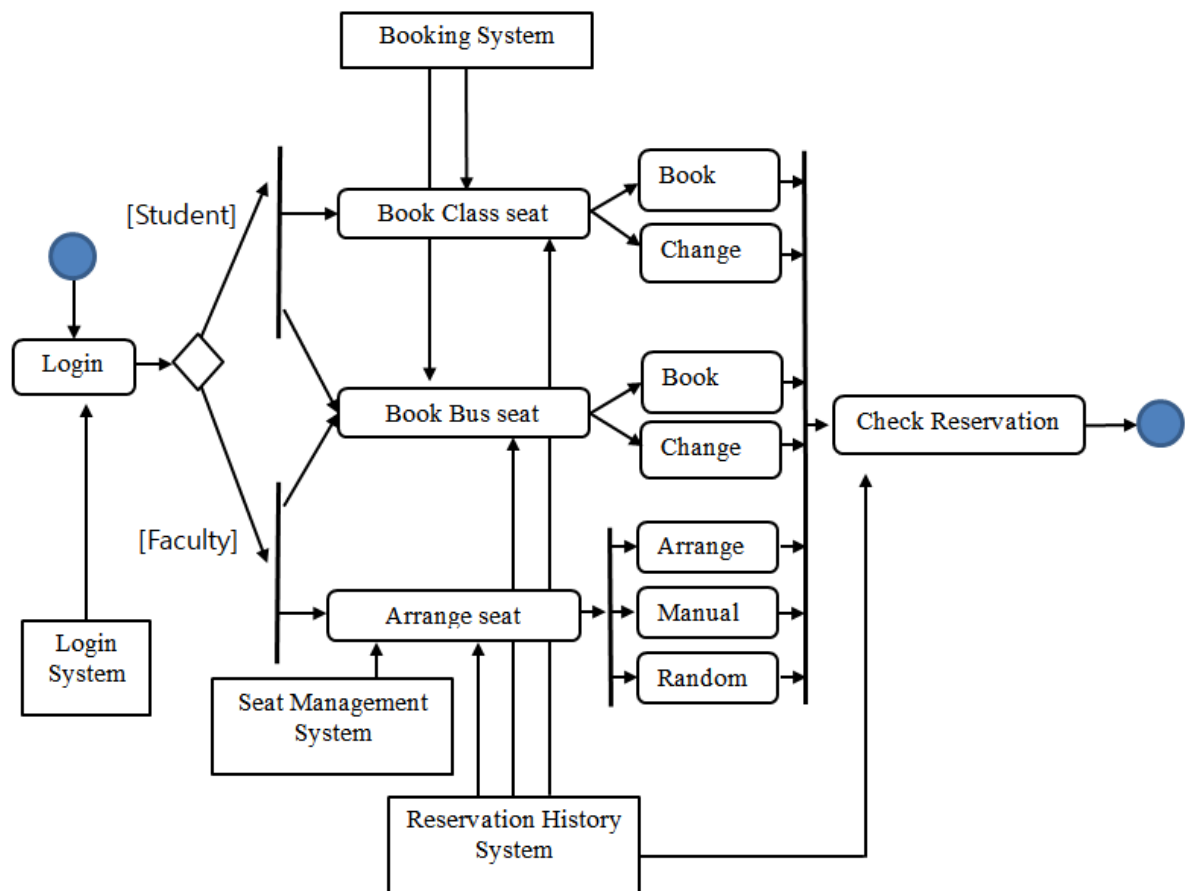
### **3.8.1. Context Model**





[Figure 4] Context Model

### 3.8.2. Process Model



[Figure 5] Process Model

### 3.8.3. Interaction Model

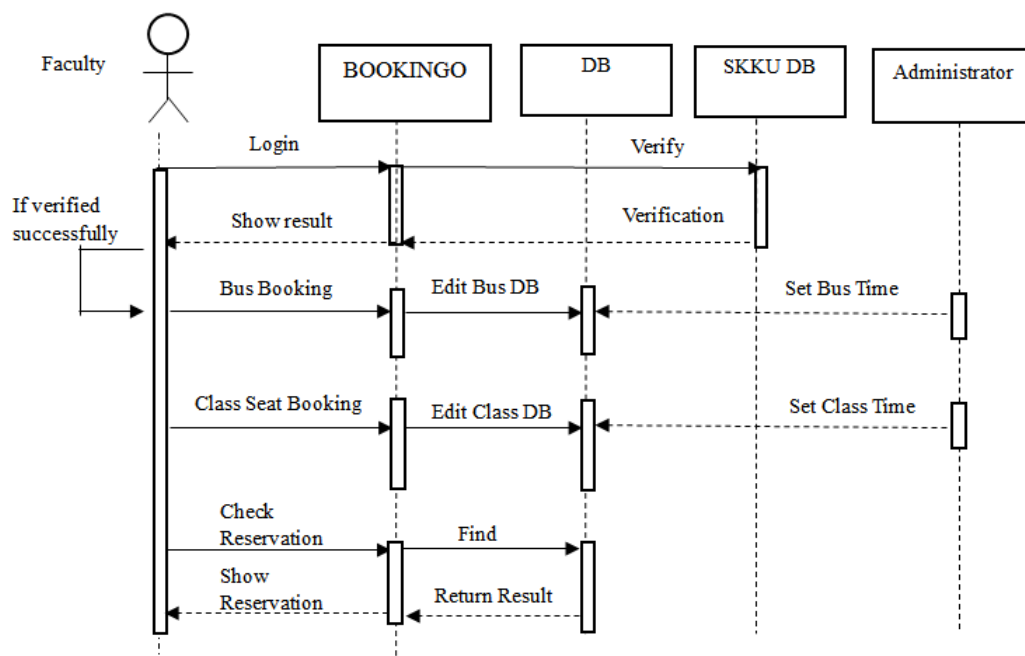
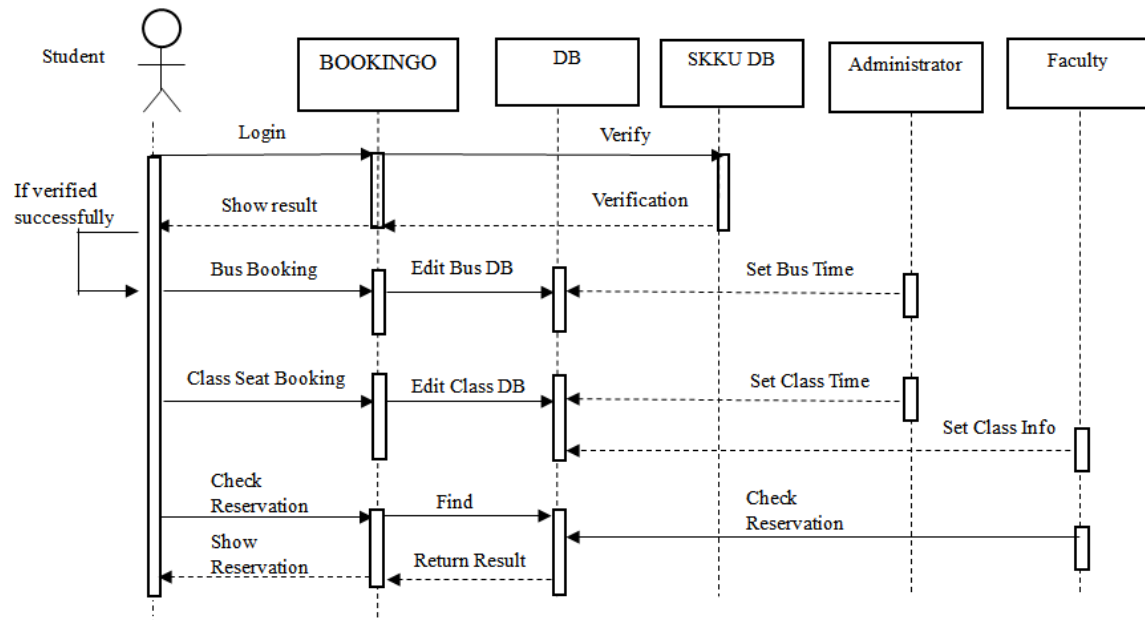
3.2.2. Use Case Diagram을 참고한다.

### 3.8.4. Behavior Model

#### 3.8.4.1. Data Flow Diagram

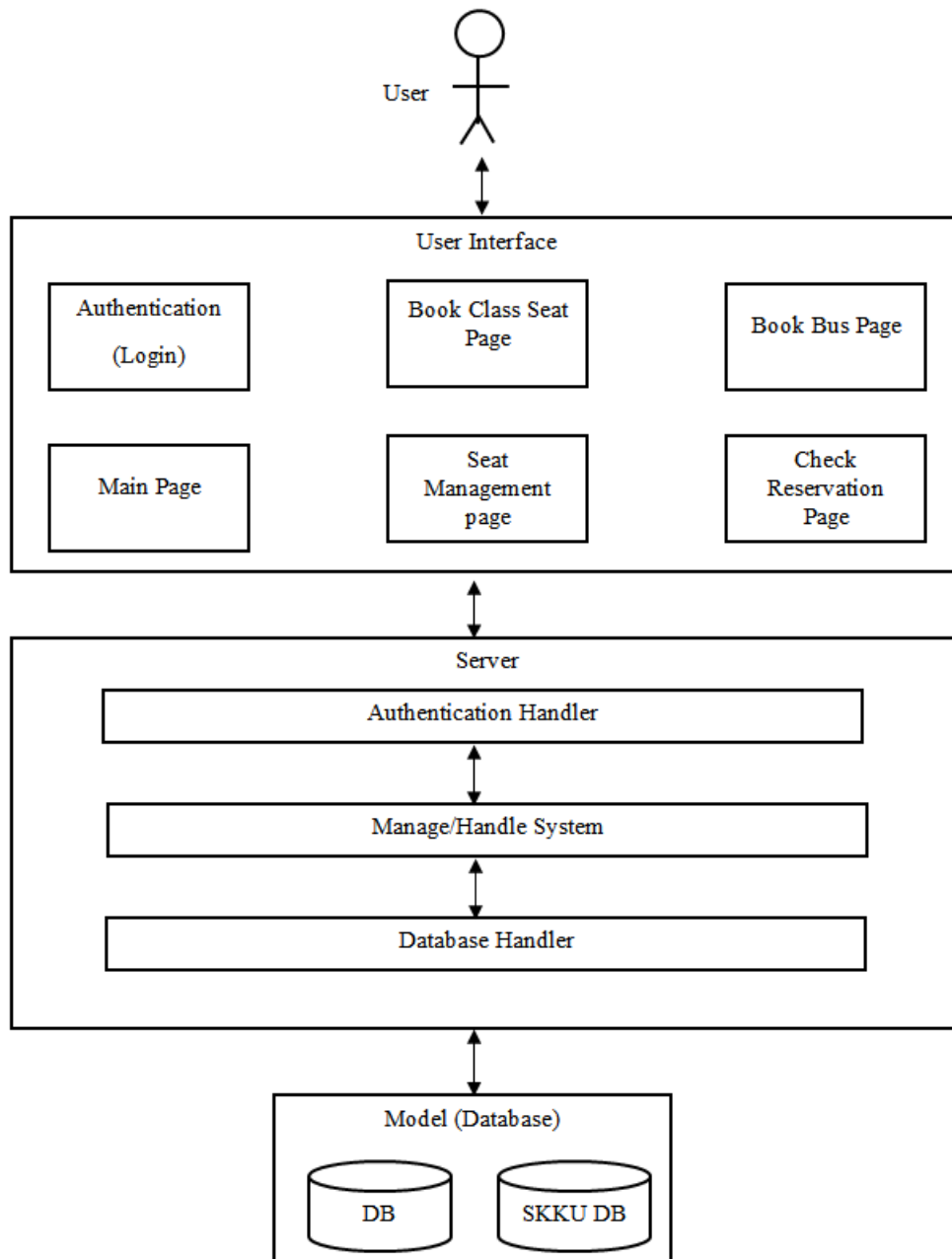
3.2.4. Data Flow Diagram을 참고한다.

## 3.8.4.2. Sequence Diagram



[Figure 6] Sequence Diagram

### 3.9. System Architecture



[Figure 7] System Architecture

### 3.10. System Evolution

현 시스템은 완성된 시스템이 아니다. 시스템이 배포된 이후에도 환경과 유저에 따라 요구사항이 변화할 수 있다. 시스템은 이런 변화에 대해 민감하게 반응하기 위해 어떤 방면에서 변화가 발생할지 예측하는 것이 중요하다. 이 섹션에서는 시스템의 기본적인 가정과 한계점을 살펴본다. 그리고 환경과 유저에 따라 생길 수 있는 요구사항 변화에 대해 알아본다.

#### 3.10.1. Limitation and Assumption

시스템은 교내 킹고앱과 교내 서비스를 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. 두 캠퍼스 간 셔틀 버스가 운영되고 있는 상황 역시 기본적인 가정으로 작용한다.

#### 3.10.2. Evolutions of Hardware and Change of User Requirements

##### 3.10.2.1. 플랫폼 확장

시스템은 현재 안드로이드 운영체제를 사용하는 모바일 기기를 대상으로 개발할 예정이다. 추후 접근성을 높이기 위해 iOS 운영체제를 이용하는 모바일 기기 및 웹 플랫폼으로 확장한다면 접근성을 크게 높일 수 있다. 특히 학생들의 리스트를 개별적으로 배정해야 하는 교직원들의 경우 웹으로 접근하는 방식을 더 선호할 가능성이 높다.

##### 3.10.2.2. 소통 창구 기능 추가

시스템은 현재 학생들과 시스템 사이 소통 창구가 부족한 상태이다. 이후 버스추가 개 설이나 강의실 환경이 실시간으로 변했을 때 빠른 피드백을 위해 소통 창구를 개설할 필요가 있다.

### 3.10.2.3. 통합 예약 서비스 기능 제공

예약과 관련된 서비스를 모두 한 곳에서 볼 수 없다는 한계점이 있다. 분산된 예약 서 비스로 불편함을 느낄 수 있다. 현재 시스템은 현재 버스와 강의실 좌석이라는 예약시 스템만 제공하고 있지만 이후 교내 서비스 관련 다양한 예약서비스를 한 곳에 모아 통 합 예약 서비스를 제공할 수 있는 방법을 고려해볼 필요성이 있다.

## 4. Supporting Information

### 4.1. Software Requirement Specification

본 문서는 IEEE Recommended Practice for Software Requirements Specifications의 형식 을 따르고 있다.

### 4.2. Document History

[Table 35] Document History

#	Date	Description	Writer
0	2021/4/15	개요 작성	임서현
1.1	2021/4/17	Introduction 작성 및 회의	강종현, 이승우, 임재현, 임서현

1.2	2021/4/23	data_1	이승우
1.3	2021/4/23	interface_1	강종현
1.4	2021/4/23	Usecase_1	임재현
1.5	2021/4/23	models_1	임서현
1.6	2021/4/24	data_2, data_3 Usecase_2	이승우, 임재현
1.7	2021/4/24	interface_2, interface_3 models_2, models_3	강종현, 임서현
1.8	2021/4/24	data_4 models_4	이승우, 임서현
1.9	2021/4/24	interface_4 Usecase_3	강종현, 임재현
1.10	2021/4/24	RS_3,4	강종현, 이승우, 임서현, 임재현
1.11	2021/4/24	RS_2,3,4,+table	임서현, 임재현
1.12	2021/4/24	RS_1,2,3,4,ind	임서현
1.13	2021/4/25	Requirement Specification	강종현, 이승우, 임서현, 임재현