**1. 개요**

**1.1 기존 문제 시나리오**

A씨는 2주전 B씨와 야구 경기장에 가기 위해 회사에 월차를 신청한 후 약속을 잡았다. 그러나 약속 전날 우천 예상으로 경기가 취소되었다. 월차까지 사용한 A씨는 난감할 수 밖에 없었다.

**1.2 프로젝트 개요**

사용자는 앱을 통해 달력확인 및 일정 관리를 할 수 있으며, 기존 일정 관리 앱과 다르게 날씨 정보 사전 확인을 통해 좀 더 유연한 일정 관리를 할 수 있다.

**- 서비스 명 : Plus Calendar**

**1.3 서비스 구성**

1) 사용자 애플리케이션

- OS : 안드로이드(linux)

- 개발 언어 : Java,

- 지원 SDK 버전 : 21 ~ 23

- 데이터베이스 : SQL lite

- 요약 : 사용자가 일정을 등록 및 확인 할 수 있음

2) 서버

- 환경 : Amazon AWS, Ubuntu OS

- 개발 언어 : node.js

- 데이터베이스 : mongo DB

- 요약 : 사용자의 요청을 처리, 데이터베이스에 사용자 및 일정 등록. GCM 연동

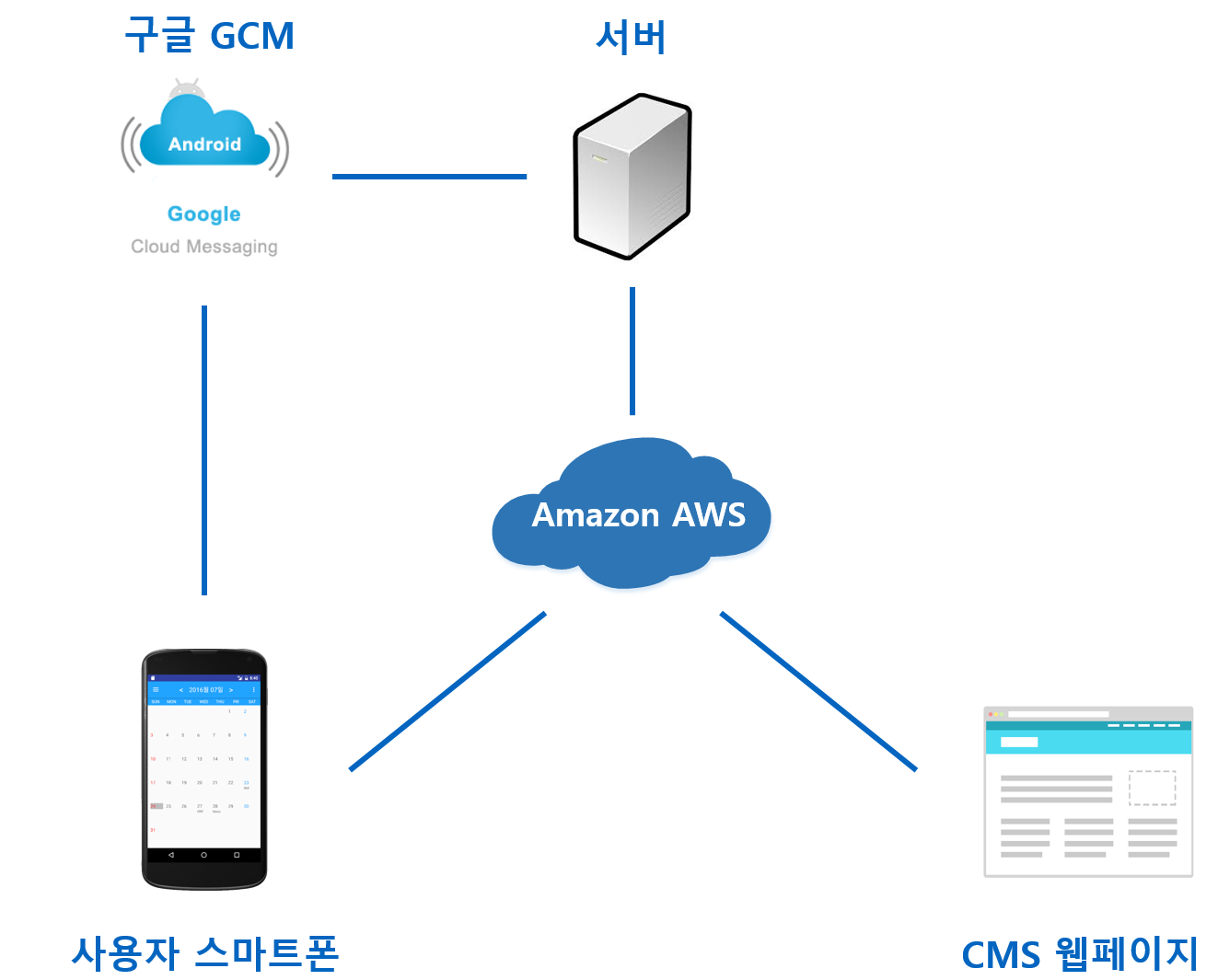
3) 관리자 웹 서버

- 개발 언어 : Java script

- 사용 프레임워크 : bootstrap

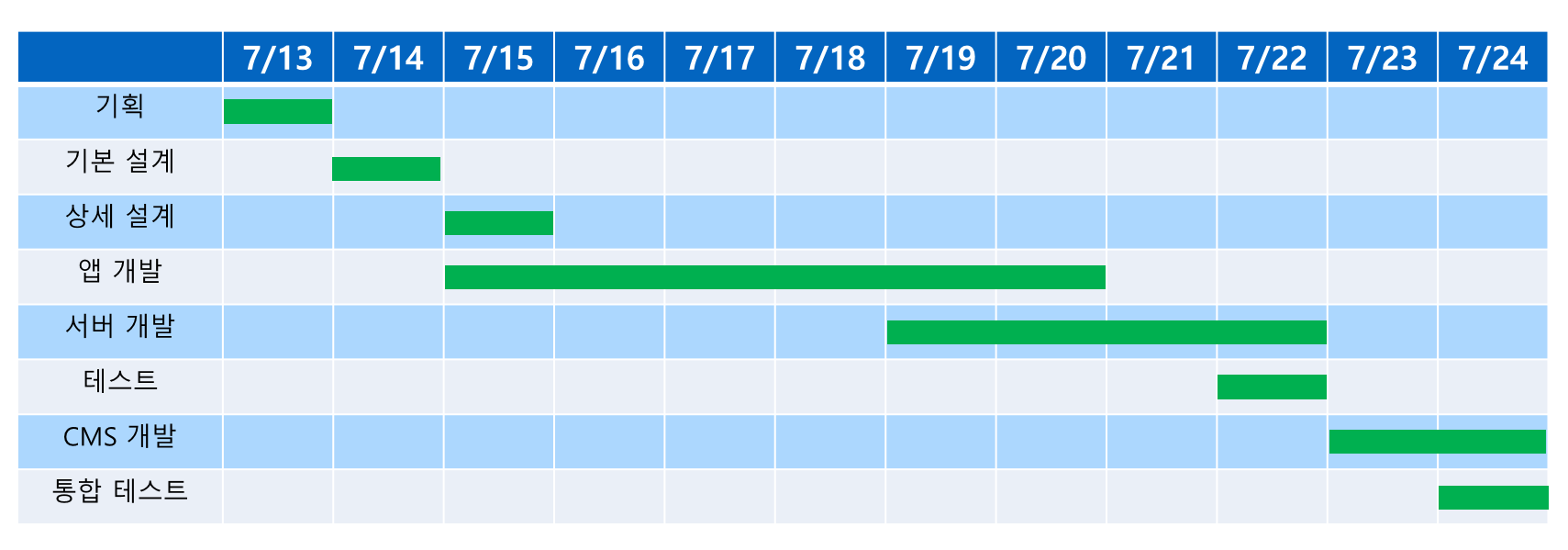
- 요약 : 관리자가 회원들을 관리 할 수 있음

**1.3 서비스 구성**



<그림 1. 서비스 구성도>

**1.4 개발 일정**



<그림 2. 일정 시퀀스>

**2. 기능 명세**

**2.1.1 앱 기능**

1) 로그인 및 로그아웃 : 페이스북 계정으로 로그인 가능

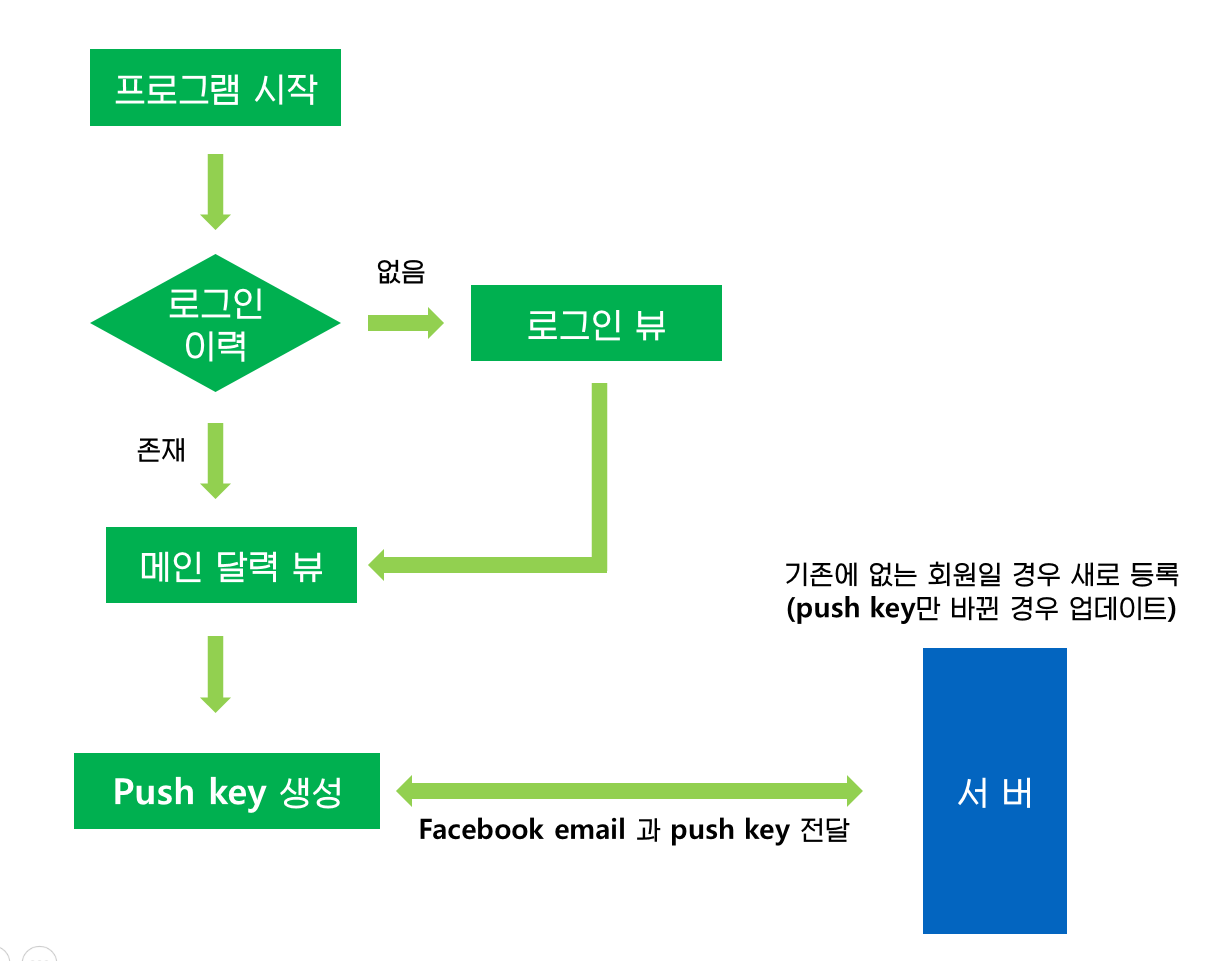
2) 달력 확인 : 사용자는 실시간으로 달력을 확인 할 수 있음

3) 일정 등록 : 제목, 시작날짜, 종료날짜, 지역, 날씨, 색상 지정 가능

4) 날씨 알람 : 일정에 원하는 지역과 날씨를 기입하면 기상청으로부터 데이터를 받아와 실시간으로 알람(만약 원하는 날씨가 아닐 경우)

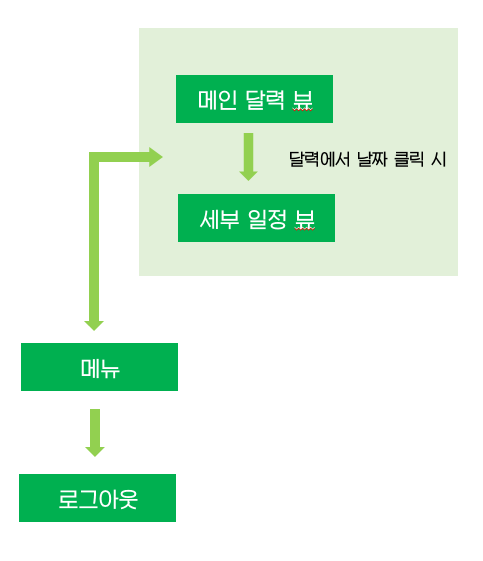
**2.1.2 앱 시퀀스 다이어그램**

1) 로그인



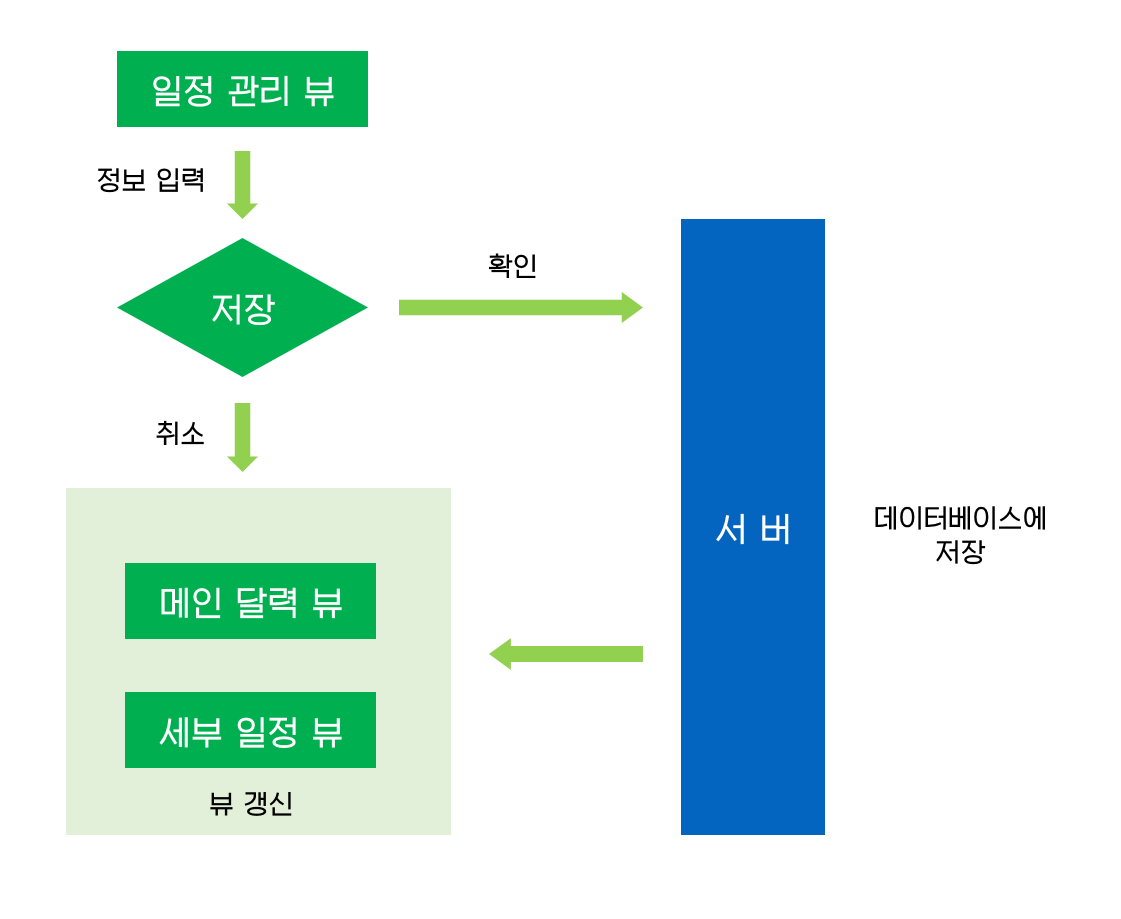
<그림 3. 로그인 시퀀스 다이어그램>

2) 달력 확인



<그림 4. 달력확인 시퀀스 다이어그램>

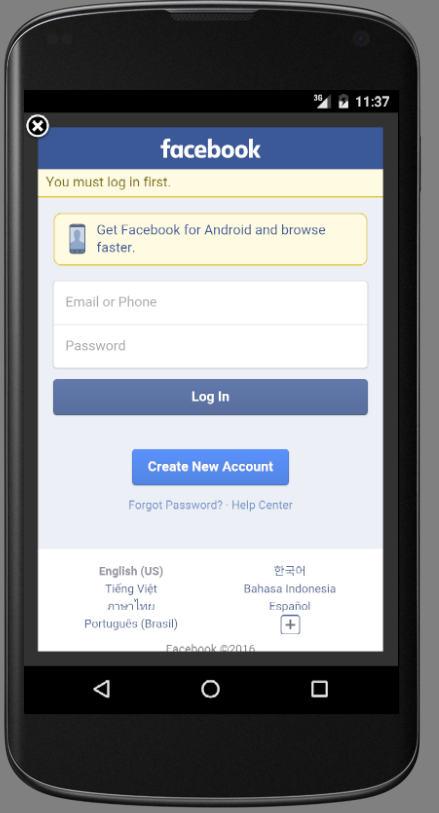
3) 일정 등록



<그림 5. 일정 등록 시퀀스 다이어그램>

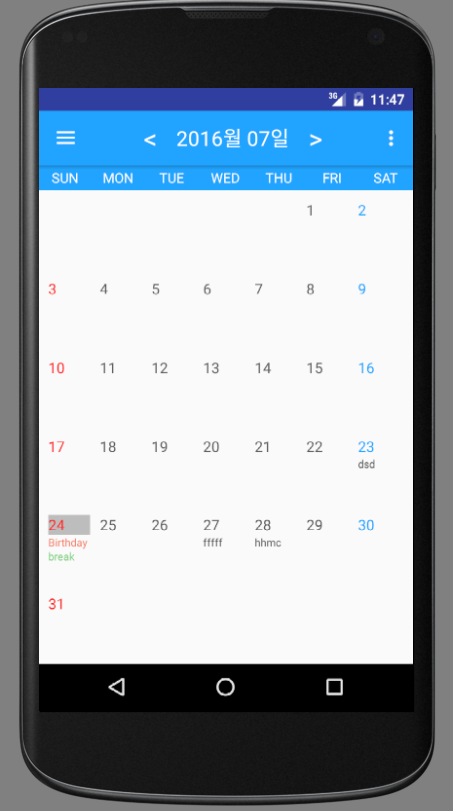
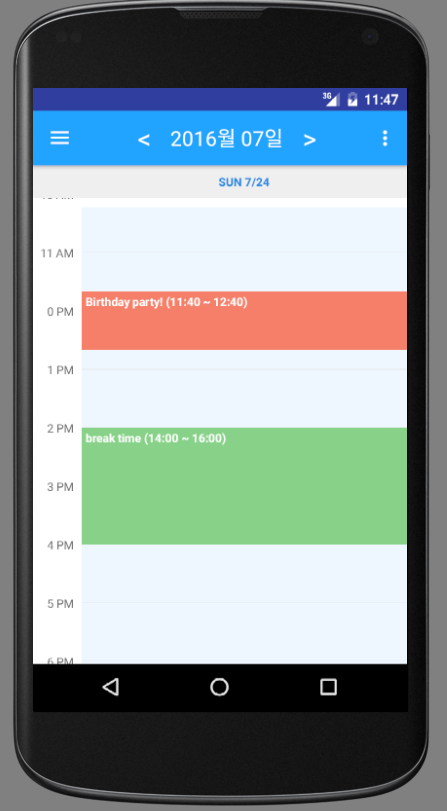
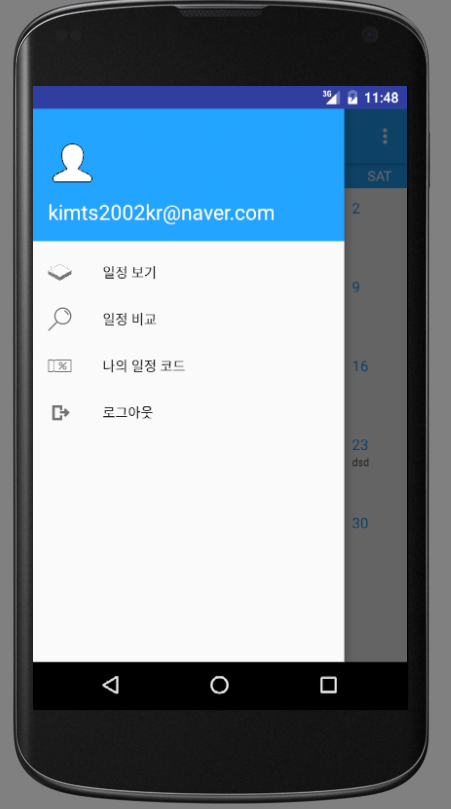
**2.1.3 앱 화면**

1) 로그인 뷰

** **

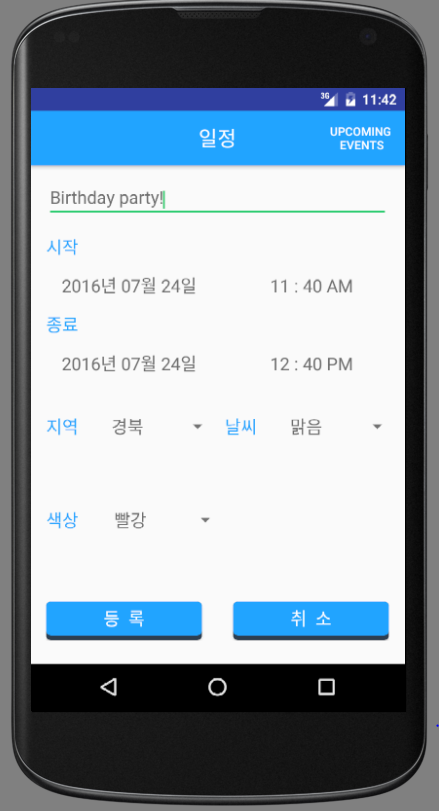
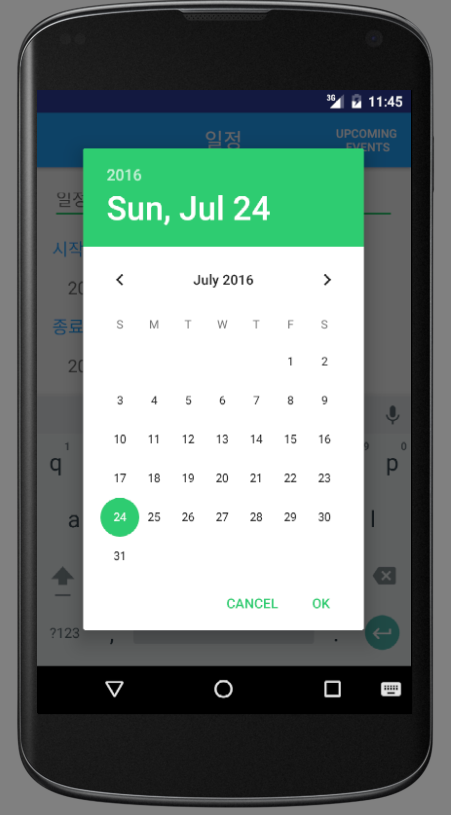
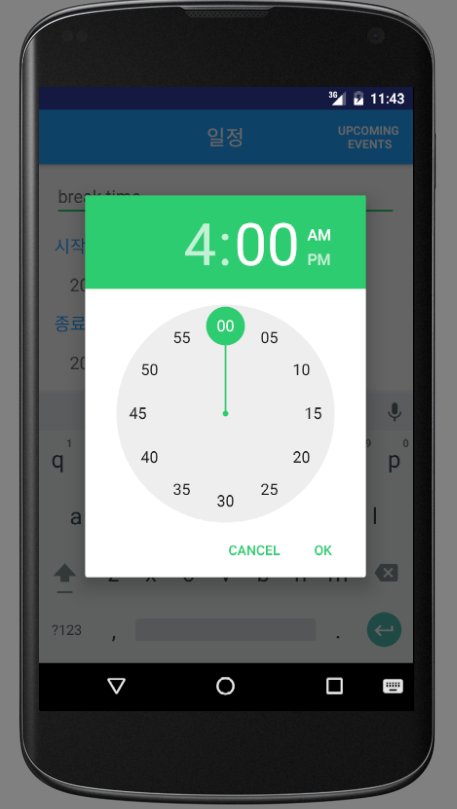
<그림 6. 로그인 화면>

2) 달력 확인 : 월 달력에는 일정이 작은 글씨로 최대 3개 정도 표현 됨. 세부 일정은 색으로 구분되어 있으며 제목과 시간이 나옴(겹쳐지는 일정도 표현).

**  **

<그림 7. 일정 확인 화면>

3) 일정 등록 : 월 달력에는 일정이 작은 글씨로 최대 3개 정도 표현 됨. 세부 일정은 색으로 구분되어 있으며 제목과 시간이 나옴(겹쳐지는 일정도 표현).

**  **

<그림 8. 일정 등록 화면>

**2.2.1 서버 역할**

1) 유저 관리 (추가, 삭제)

2) 스케줄 기록 및 삭제, 조회

3) 기상청으로부터 실시간 날씨 조회 (하루에 한번 체크)

4) Push 서비스 제공

5) Admin 관리 페이지(CMS) 제공

**2.2.2 서버 RESTful API**

1) 유저 추가 : 유저 이메일과 push토큰을 전달

**- URL** : /api/email/token GET

**- header** : application/json

- **Return** : 성공 시 1, 실패 시 0 (json)

- **Example** :

URL : [52.78.25.63/api/kimts2002kr@naver.com](mailto:52.78.25.63/api/kimts2002kr@naver.com)/cU32mwsFahQ:APA91bF0y3RhyJ5aZFNov6hvI0TzPJGwHdVgUnb

2) 유저 삭제 : 유저 이메일을 받아 회원 정보 및 일정 등록 정보 모두 삭제

**- URL** : /api/del/email GET

**- Body** : json 데이터 (데이터 구조는 아래 명시)

- **Return** : 성공 시 204 리턴, 실패 시 500 리턴 (http)

- **Example** : Url : [52.78.25.63/api/del/kimts2002kr@naver.com](mailto:52.78.25.63/api/del/kimts2002kr@naver.com)

3) 스케줄 확인: 유저 이메일과 관련된 모든 스케줄 가지고 옴

**- URL** : /api/email GET

**- header** : application/json

- **Return** : json 타입의 스케줄 표 리스트(아래 명세)를 가지고 옴

- **Example** : Url : [52.78.25.63/api/kimts2002kr@naver.com](mailto:52.78.25.63/api/kimts2002kr@naver.com)

4) 스케줄 확인: 유저의 스케줄 등록

**- URL** : /api/email POST

**- header** : application/json

**- body** : json 타입의 스케줄 정보 (아래 명세)

- **Return** : 성공 시 1, 실패 시 0 (json)

- **Example** : Url : [52.78.25.63/api/kimts2002kr@naver.com](mailto:52.78.25.63/api/kimts2002kr@naver.com)

Body : { "id" : 9, "color" : "GREEN", "weather" : "NONE", "ed\_time" : "Jul 24, 2016 4:00:00 PM", "st\_time" : "Jul 24, 2016 2:00:00 PM", "loc" : "109", "email" : "kimts2002kr@naver.com", "title" : "break time"}

5) 특정 스케줄 삭제: 유저의 스케줄 삭제

**- URL** : /api/del/email/id GET

**- header** : application/json

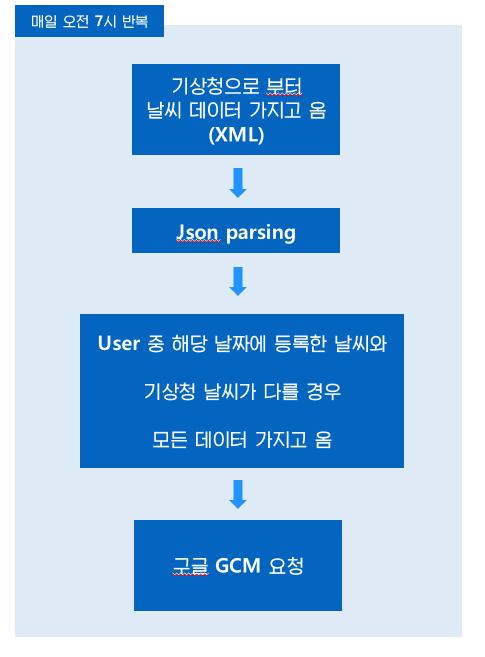
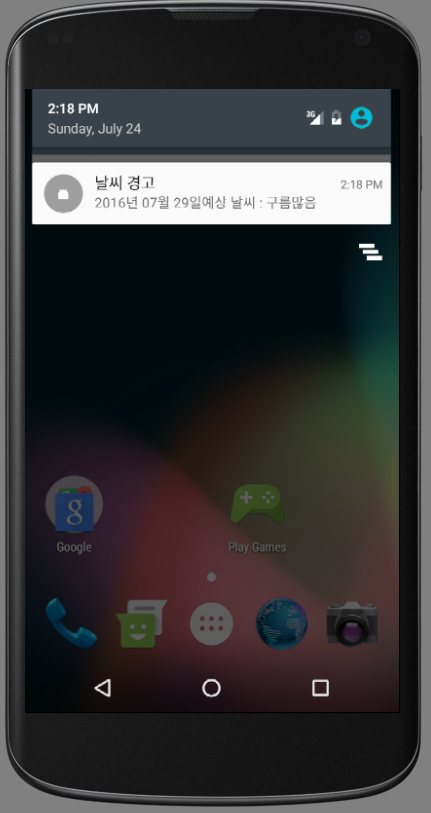
**- body** : json 타입의 스케줄 정보 (아래 명세)

- **Return** : 성공 시 204 리턴, 실패 시 500 리턴 (http)

- **Example** : Url : [52.78.25.63/api/kimts2002kr@naver.com](mailto:52.78.25.63/api/kimts2002kr@naver.com)/12

**2.2.3 서버 기능 시퀀스**

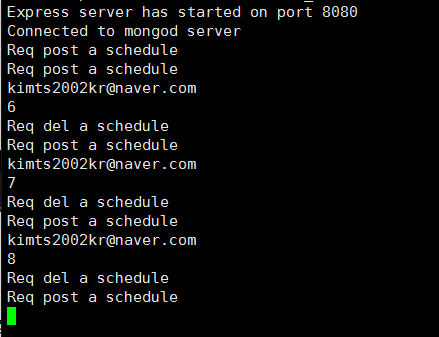
1) 날씨 알림 기능 : 기상청으로부터 10일 정도 날씨 목록을 가지고 와 사용자가 원하는 날씨와 다를 경우 알림 메시지 전송

 ****

<그림 9. 날씨 조회 시퀀스 다이어그램> <그림 10. 실제 알람 메시지가 오는 화면>

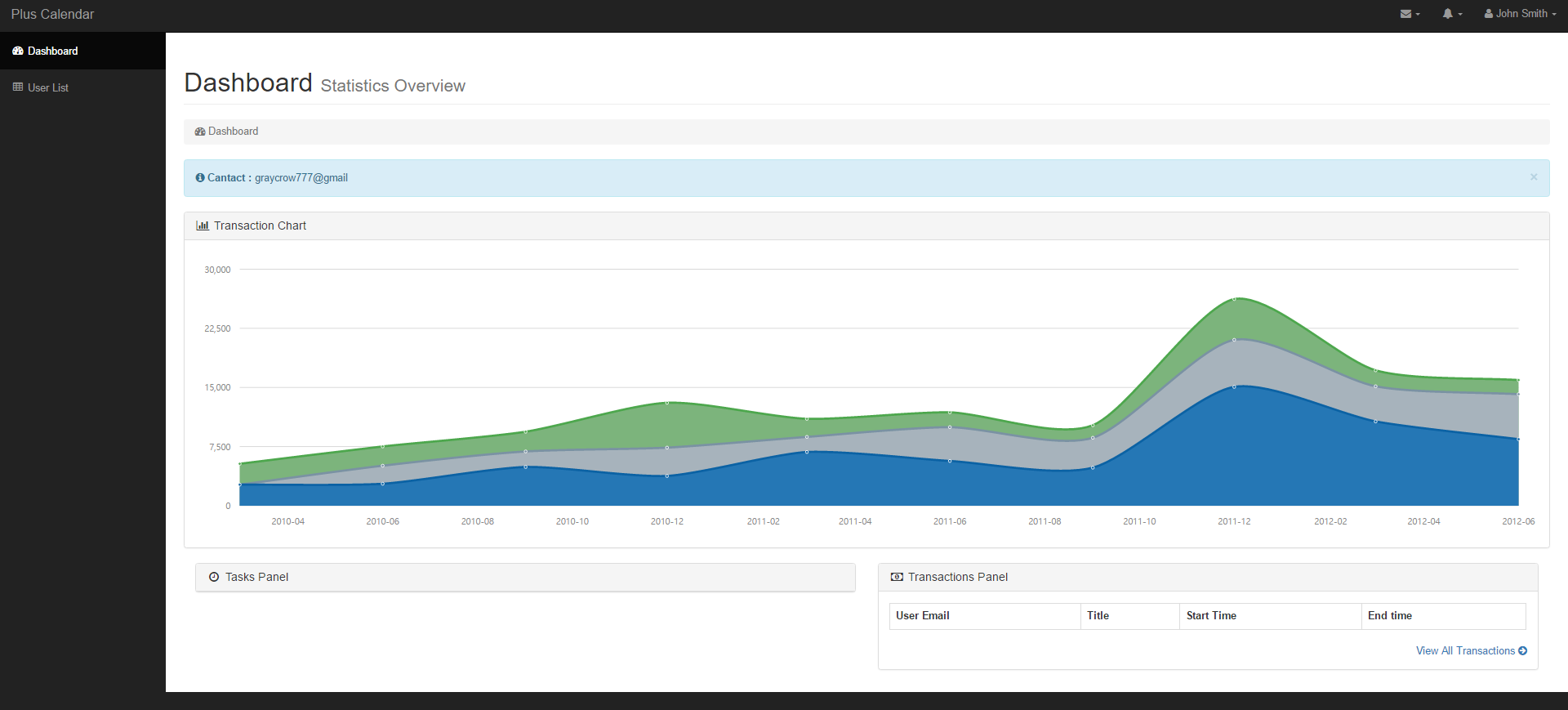
**2.2.4 서버 화면**

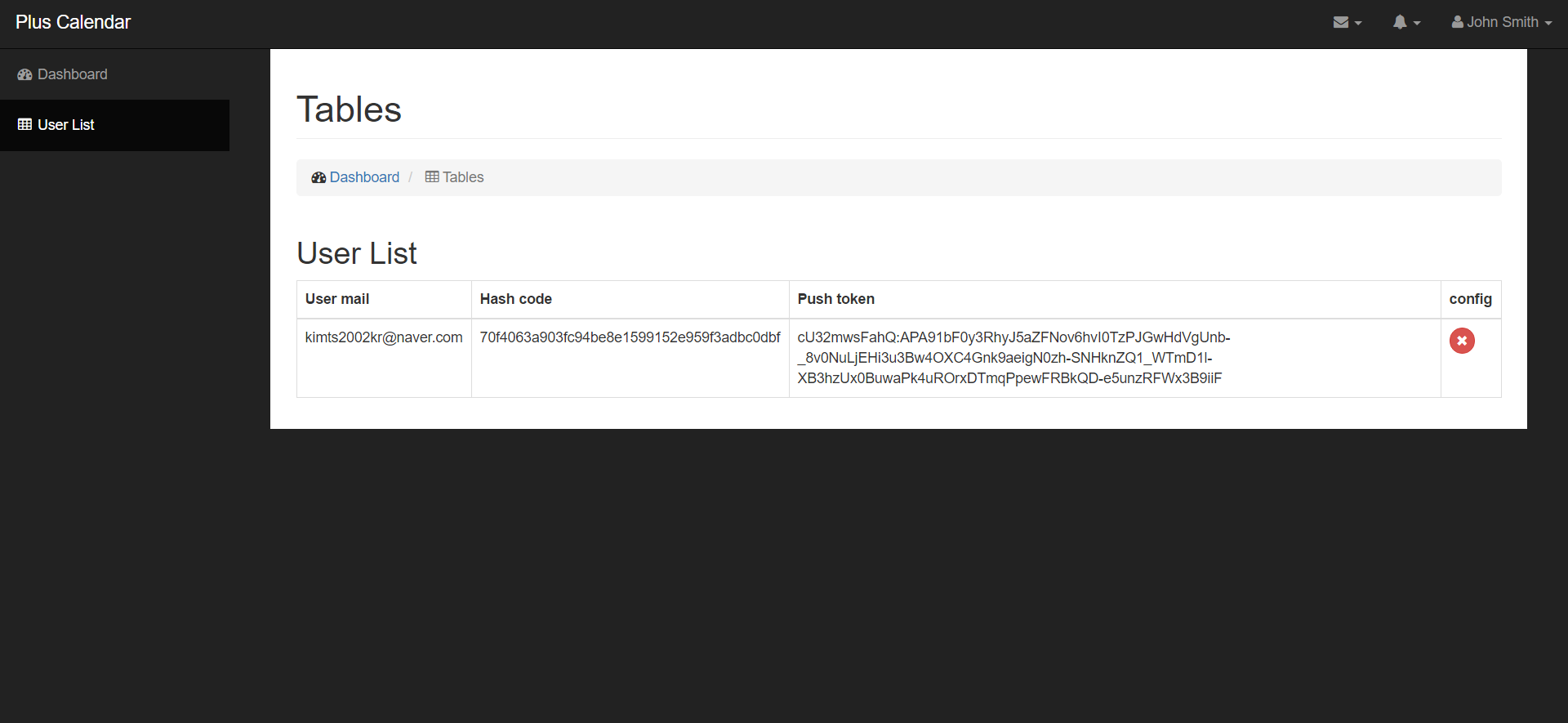
1) 서버 실행 화면

****

<그림 11. 서버 실행 후 로그 화면>

2) CMS 화면 (현재 오류로 유저 목록 조회만 정상 서비스 중)



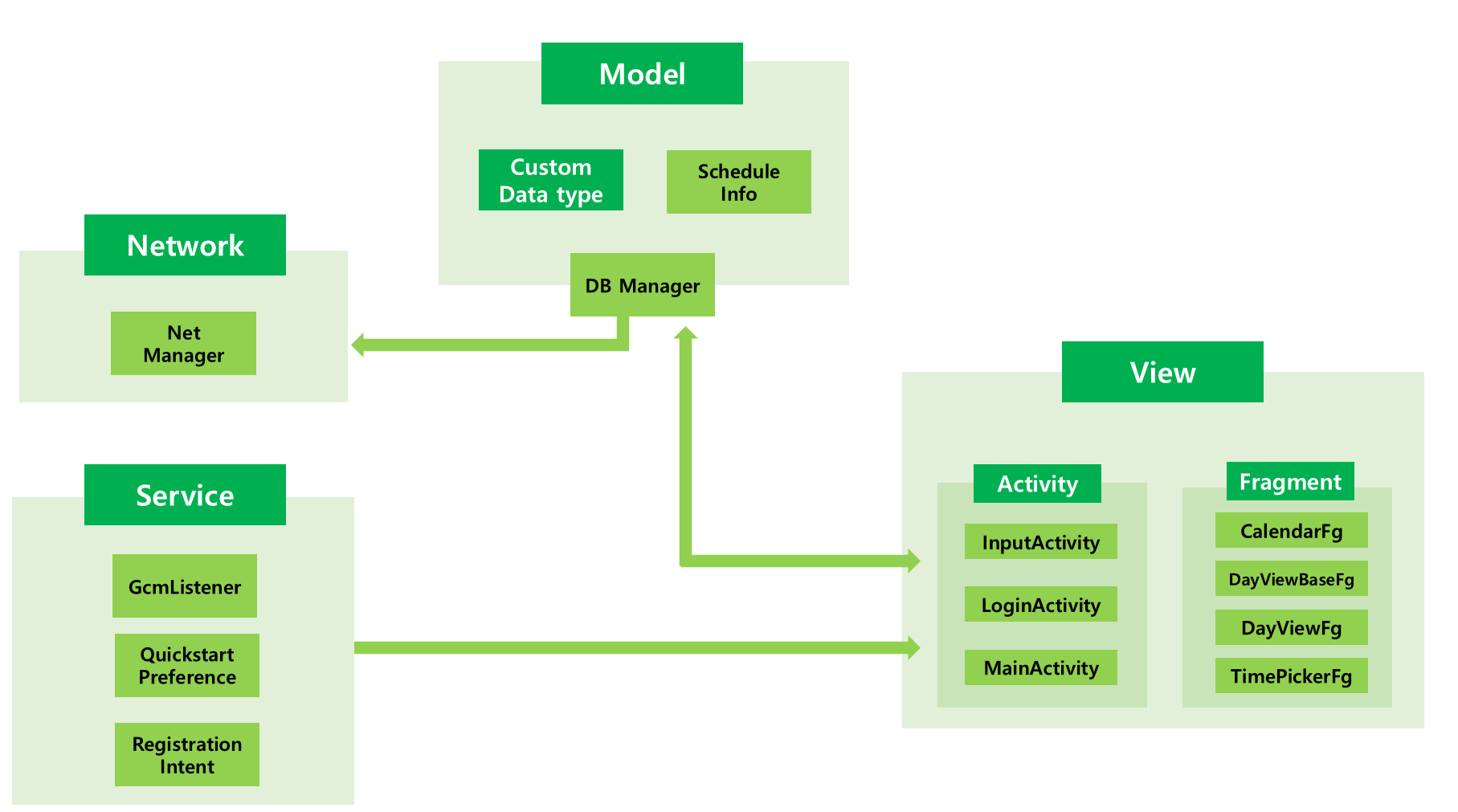


<그림 12. CMS 화면>

**3. 프로젝트 구조**

**3.1 앱 구조**

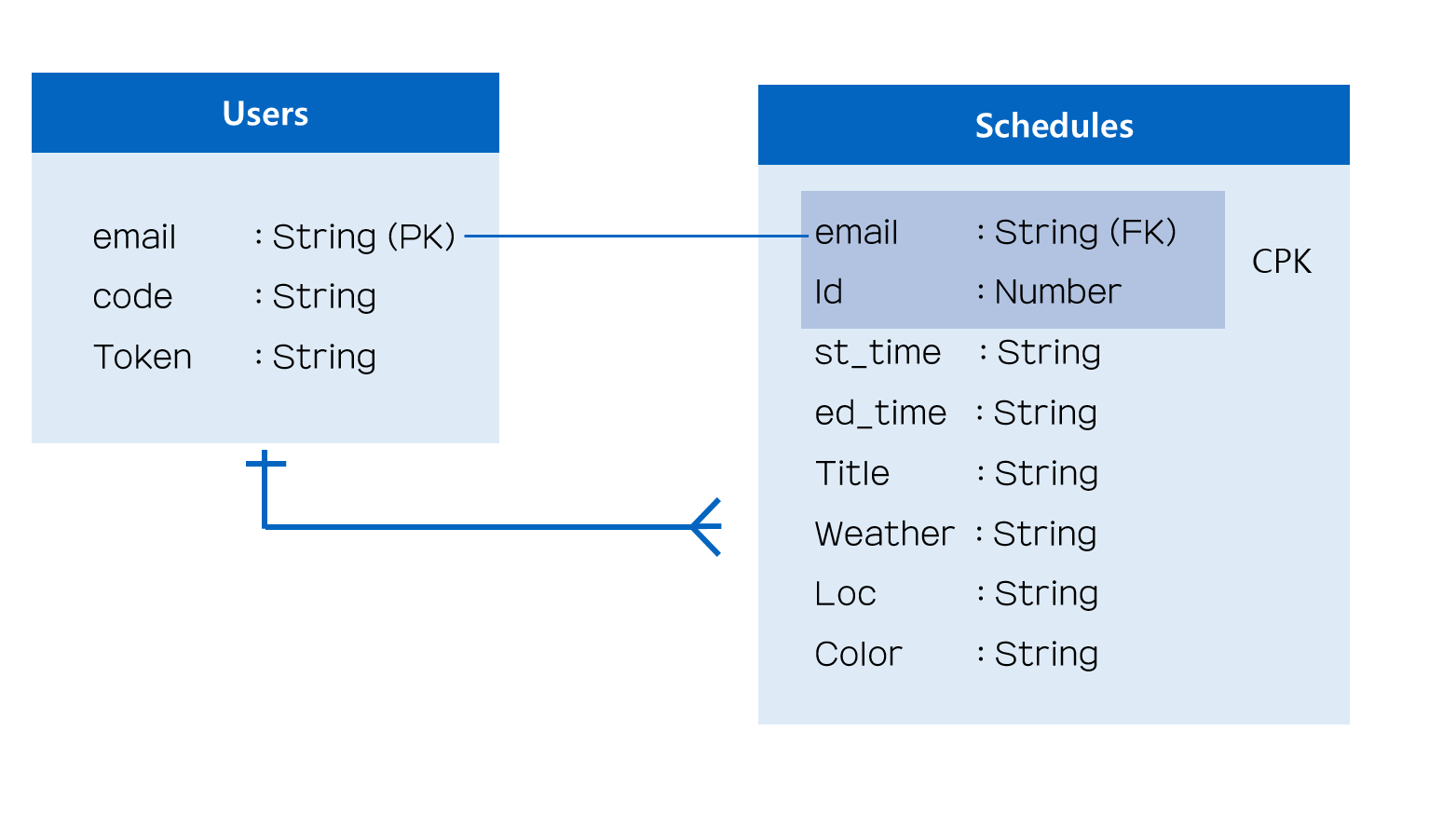
1) 앱 구성도



<그림 13. 앱 구성도(클래스)>

**3.2 서버 구조**

1) 데이터베이스 구성도



<그림 14. DB 구성도>

※ Code 는 mail의 hash 값이며 추후 개발될 달력 공유 기능에 사용됨

**4. 기타**

**4.1 요구 조건 만족도**

|  |  |  |
| --- | --- | --- |
| **기본 사항** |  |  |
| 앱(게임)과 서버 통시하는 로직이 포함되어야함 | 서버 – 클라이언트 구조 | 요구 충족 |
| 클라이언트 게임은 게임첫 설치 시간. 접속시간. 접속후 게임종료시간. 마지막 플레이시간등이 DB에 저장되어야 함 | 게임이 아님 |  |
| 페이스북, 트위터, 이메일등 타 서비스 연계가 1개이상 되어야 함 | 페이스북 연동 로그인 | 요구 충족 |
| 앱스토어에 업로드 되어야함 | 플레이스 스토어 등록 | 요구 충족 |
| 서버 프로그램이 클라우드나 호스팅이 되는 서버에 배포가 되어야 함 | 아마존 클라우드에 서버 위치 | 요구 충족 |
| 데이터는 DB에 저장되어 SQL을 통해서 조회 및 추가가 가능해야 함 | Mongo DB 및 SQL lite 사용 | 요구 충족 |
| 백엔드는 BaaS나 클라우드 제한 없으나 웹서비스 형태이어야 함 | 웹 API 및 CMS 제공 | 요구 충족 |
| **가점 사항** |  |  |
| 클라우드 이용시 가점 | 아마존 클라우드 서비스 사용 | 요구 충족 |
| 친구추가, 랭킹, 선물하기등의 기능 추가시 가점 | 추후 공유 기능을 위한 DB 설계 및 API 는 구축 | 요구 미비 |
| Restful API 형태 설계 적용시 가점 | RESTful API 설계 | 요구 충족 |
| push 구현 되었으면 가점 | 구글 Push 서비스 사용 | 요구 충족 |
| 안드로이드와 아이폰 둘다 대응시 가점 |  | 요구 불만족 |
| DB와 연동되는 별도의 CMS(관리시스템)을 구축시 가점 | CMS 웹페이지 개발 | 요구 충족 |

**4.2 소스 코드 주소**

- 안드로이드 앱 : <https://github.com/KimTaeSeong/SharedCalendar>

- 서버 : <https://github.com/KimTaeSeong/SharedCalendarServerTest>

**4.3 CMS 주소**

- <http://52.78.25.63:8080/>