**시스템 구조 설계서**

**System Architecture Specifications**

**2019. 05. 23**

# **- Locker팀 (smart locker) -**

정대한 신현빈 안현준 최호열 박주빈

**참여 현황**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 구분 | 소속 | 성명 | 날짜(최종일) | 서명 |
| 작성자 | Locker 팀 | 정대한 | 2019.05.23 |  |
| 검토자 | Locker 팀 | 신현빈 | 2019.05.23 |  |
| Locker 팀 | 박주빈 | 2019.05.23 |  |
| Locker 팀 | 안현준 | 2019.05.23 |  |
| Locker 팀 | 최호열 | 2019.05.23 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 승인자 |  | 김진술 |  |  |

**개정 이력**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 개정 일자 | 버전 | 개정 내역 | 작성자 | 승인자 |
| 2019.05.12 | v1.0 | 초안 작성, 블록 다이어그램 추가 | 정대한 |  |
| 2019.05.14 | v1.1 | 배치, 상태 다이어그램 추가 | 신현빈 |  |
| 2019.05.16 | v1.2 | 패키지, 액티비티 다이어그램 추가 | 박주빈 |  |
| 2019.05.18 | v1.3 | 클래스 다이어그램 추가 | 안현준 |  |
| 2019.05.20 | v1.4 | 순차 다이어그램 추가, 데이터베이스 스키마 추가 | 최호열 |  |
| 2019.05.23 | v1.6 | 총정리, 최종검토 | 정대한 | 김진술 |

**목차**

**- Locker(smart locker)………………………………………………………………………………………………………1**

**1 개요.……………………………………………………………………………………………………………………………4**

1.1 목적…………………………………………………………………………………………………………………………………………..4

1.2 범위…………………………………………………………………………………………………………………………………………..4

1.3 참고 자료………………………………………………………………………………………………………………………………….4

1.4 용어 및 약어…………………………………………………………………………………………………………………………….4

**2 시스템 구조 - 다이어그램…………………………………………………………………………………..…………5**

2.1 시스템구조 - 블록 다이어그램………………………………………………………………………………………………..5

2.1.1 구조의 장점………………………………………………………………………………………………………………………6

2.1.2 구조의 단점………………………………………………………………………………………………………………………6

2.1.3 구성도 설명………………………………………………………………………………………………………………………6

2.2 동적 구조 - 순차 다이어그램…………………………………………………………………………………………………..7

2.2.1 스마트 라커 제작 및 서버의 목적 및 역할…………………………………………………………………….7

2.2.2 데이터베이스 스키마………………………………………………………………………………………………………..8

2.3 정적 구조 - 클래스 다이어그램………………………………………………………………………………………………9

2.4 그 외 다이어그램……………………………………………………………………………………………………………………10

2.4.1 배치 다이어그램……………………………………………………………………………………………………………..11

2.4.2 액티비티 다이어그램………………………………………………………………………………………………………12

2.4.3 패키지 다이어그램………………………………………………………………………………………………………….13

**1. 개요**

**1.1 목적**

본 문서는 “스마트 라커 개발” 사업에서 스마트 라커에 대한 시스템 구조 설계서이다.

**1.2 범위**

이 문서에서는 “스마트 라커 개발” 사업의 최종 결과물을 스마트 라커와 이와 연결되는 안드로이드 어플리케이션과 각 통신을 지원하는 서버 프로그램으로 범위를 정한다.

**1.3 참고자료**

- 안드로이드 스튜디오를 활용한 실전 앱만들기/ 조상철/ 영진닷컴

- 데이터베이스배움터/ 홍익경/ 생능출판

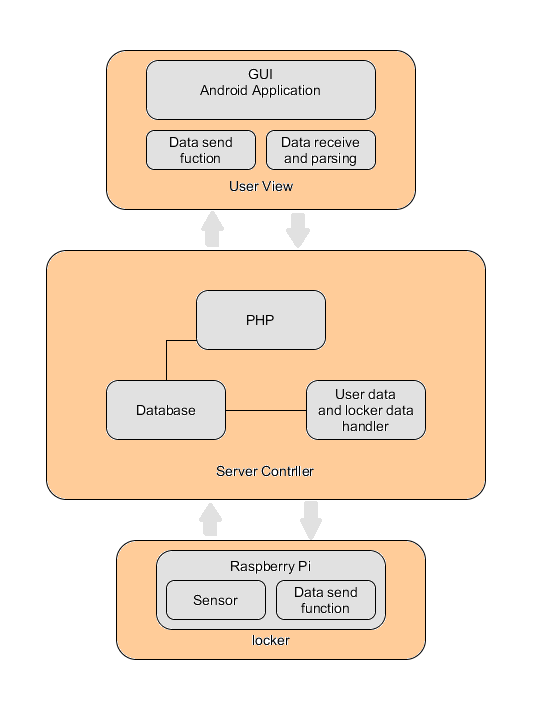
- 라즈베리파이 3 네트워크 프로그래밍/ 이용진/ 복두출판사

**1.4 용어 및 약어**

|  |  |
| --- | --- |
| PHP | Hypertext Preprocessor |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| DB | Database |
| IDE | Integrated development environment |

**2. 시스템 구조 - 다이어그램**

**2.1 시스템 구조 - 블록 다이어그램**



### <그림1 블록다이어그램>

스마트라커의 전체적인 시스템은(그림 1)과 같은 구조로 되어 있다. 사용자 어플리케이션과 서버, 스미트라커에 장착된 라즈베리 파이로 이루어진 구조이며 사용자는 안드로이드 어플리케이션의 GUI를 이용하여 실시간으로 전송되는 사물함 정보를 육안으로 확인할 수 있는 것과 동시에 비어있는 사물함을 예약할 수 있다. 서버는 사용자의 스마트폰의 안드로이드 어플리케이션과 라즈베리 파이 중간 역할로 통신하면서 사용자의 안드로이드 어플리케이션에서 보내는 JSON 형태의 Data send Fuction 메세지를 받아와 사물함에 장착된 라즈베리 파이에 전송한다. 서버는 Database에 저장된 내용을 스마트폰에 JSON 형태로 보내는 역할도 한다. 또한 스마트라커에 장착된 LED는 직접적으로 사용자에게 사물함의 정보를 보여준다. 서버는 라즈베리파이의 위에 있으며 위에서 상술한 것과 같이 서버와 스마트폰은 JSON 형태로 통신한다.

**2.1.1 구조의 장점**

**-** 라즈베리파이 위에 서버가 있으므로 별도의 서버 비용이 발생하지 않는다.

**-** JSON 형태의 통신은 앱의 보안성을 높여준다.

**2.1.2 구조의 단점**

**-** 서버가 라즈베리파이 위에 있기 때문에 IP가 달라질 수 있다. 따라서 공유기를 사용해 IP를 선지정 해줘야 한다.

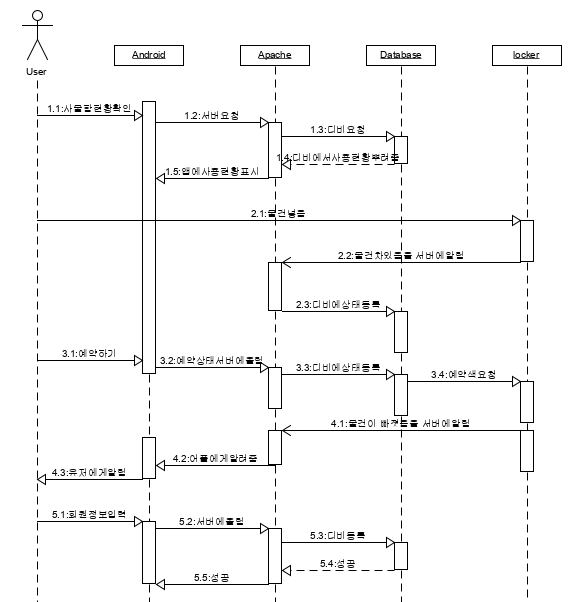
- 공인 IP가 없는 관계로 실제로 서버를 구현할 때에는 집에서 사용하고 있는 공유기의 포트포워딩을 이용하여 포트를 정하여 외부로 나갈 수 있도록 한다.

- 라즈베리파이의 성능 한계로 서버의 성능이 제한된다.

**2.1.3 구성도 설명**

|  |  |
| --- | --- |
| 구성 | 설명 |
| GUI  Android Application | 실제로 사용자가 안드로이드 어플리케이션을 통해 GUI 환경으로 사물함의 정보를 확인, 예약 하거나 기타 회원정보관리를 할수 있다. |
| Data send  fuction | JSON의 형태로 지정된 IP의 PHP파일에 데이터를 송신한다. |
| Data receive and parcing | JSON의 형태로 데이터를 받고 파싱한다. |
|  | 데이터의 이동경로를 보여주며 어플리케이션-서버 통신에서는 JSON 형태의 데이터 이동 이고 서버-라즈베리파이에서는 응용프로그램에 의한 질의이다. |
|  | 데이터의 이동경로를 보여주며 어플리케이션-서버 통신에서는 JSON 형태의 데이터 이동 이고 서버-라즈베리파이에서는 응용프로그램에 의한 질의이다. |
| PHP | 서버의 PHP파일이 JSON형태의 통신을 지원한다. |
| Database | 라즈베리파이의 디스크내 데이터베이스로 사물함 정보, 유저 정보, 예약 정보 등을 저장한다. |
| User data and locker data handler | Database를 관리하는 DBMS를 나타낸 것으로 실제로는 My sql을 사용한다. |
| Sensor | 라즈베리파이에 장착된 센서로 사물함의 사용여부를 판단한다. |
| Data send  function | 라즈베리파이에서 센싱한 값을 DB에 저장하는 역할을 하는 프로그램(함수)를 의미한다. |

**2.2 동적구조 - 순차다이어그램**



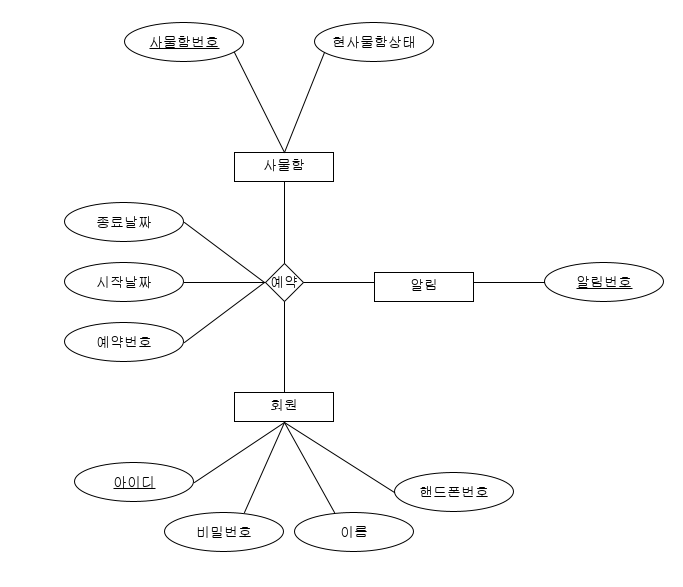
**<그림2 순차다이어그램>**

스마트라커의 전체적인 시스템(그림 2)은 다음과 같은 구조로 작동한다. 사용자 어플리케이션에서 실시간 사물함 정보를 요청하면 서버를 통해 DB 정보에 접근한다. DB는 요청받은 정보를 서버로 보내주고 앱은 서버로부터 정보를 받아 화면에 표시한다. 유저가 스마트라커에 물건을 넣으면 라커에 장착된 라즈베리 파이의 센서가 물건이 들어왔음을 서버에 알리고 서버는 이 정보를 DB에 저장한다. 유저가 앱으로 사물함을 예약하면 정보를 서버에 알리고 예약 정보를 DB에 저장한다. 그리고 사물함에 예약 정보를 전달하여 LED의 색을 바꾼다. 반대로 차있던 물건이 나가면 이를 서버를 통해 앱으로 전달한다. 유저는 이를 통해 원격에서도 사물함이 비워졌다는 것을 알 수 있다. 유저가 앱을 통해 회원 정보를 입력하면 서버를 통해 DB에 등록되어 사용자 등록이 가능하다.

**2.2.1 스마트라커** **제작 및 서버의 목적 및 역할**

스마트라커에서 서버는 데이터베이스를 저장하고 사물함과 스마트폰 어플리케이션간의 통신을 지원해주기 위해 존재한다. 서버는 JSON형식으로 받아들인 데이터를 파싱하여 DB에 질의하거나 저장하고 데이터를 JSON형식으로 스마트폰 어플리케이션에 보내준다. JSON을 이용한 통신에는 PHP파일이 사용된다. 서버의 PHP파일 JSON을 이용한 통신은 어플리케이션의 보안을 높여준다.

## **2.2.2 데이터베이스 스키마**



**<그림3 데이터베이스스키마>**

본 스키마(그림 3)는 서버에 저장된 DB의 스키마를 표현한다. reletionship set을 table로 표현하였고 나머지 테이블을 삼진관계로 표현하였다. 예약 테이블은 사물함, 알림, 회원 테이블의 기본키들을 외래키로 참조하고 사물함번호, 알림번호, 아이디를 합쳐서 기본키로 구성된다.

⦁사물함

- 사물함번호 : 사물함을 구분하기 위한 번호(기본키)

- 현사물함상태 : 사용가능, 사용중, 예약중을 나타냄

⦁회원

- 아이디 : 회원을 구분하기 위한 번호(기본키)

- 비밀번호 : 아이디를 인증하기 위한것

- 이름 : 회원정보

- 휴대폰번호 : 본인인증을 위함

⦁알림

- 알림번호 : 알림을 구분하기 위한 번호(기본키)

⦁예약

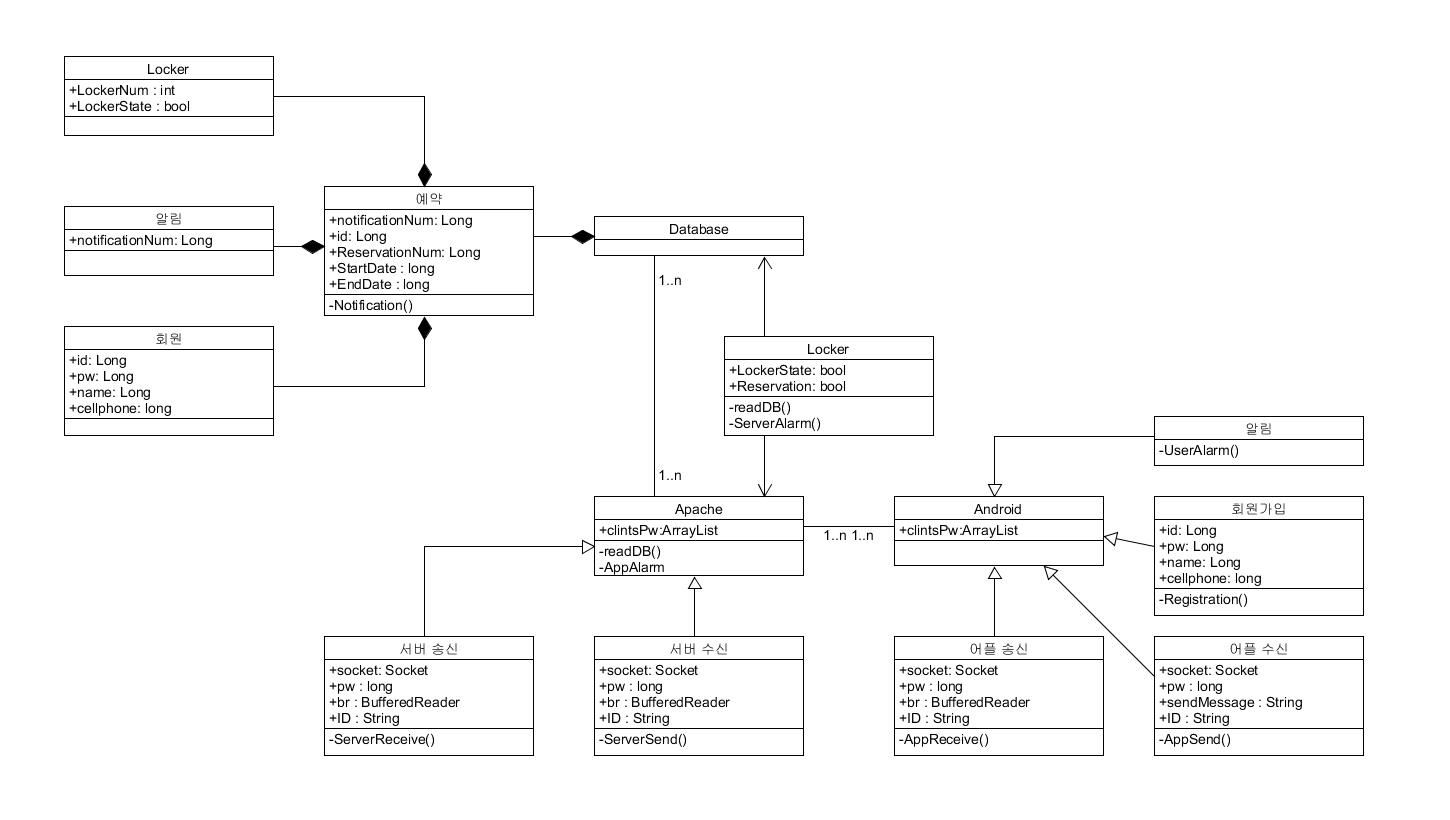
- 아이디, 사물함번호, 알림번호 : (기본키)

- 예약번호 : 예약을 구분하기 위한 번호

- 시작날짜 : 사용시작 예정 날짜

- 종료날짜 : 사용완료 예정 날짜

**2.3 정적구조 - 클래스 다이어그램**

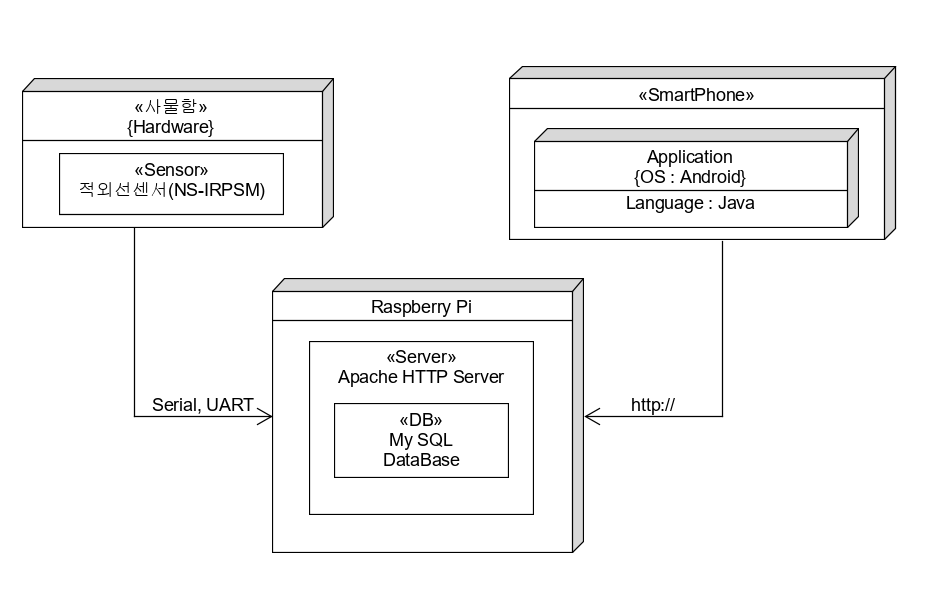


**<그림4 클래스 다이어그램>**

스마트라커의 전체적인 클래스 구조(그림 4)는 위와 같다. 어플리케이션, 서버(Apache), 스마트라커, 데이터베이스 클래스는 시스템의 골격이 되는 클래스로써, 각자 관련 클래스들에게 정보를 받아와 처리하며 통신한다. 앱 클래스는 어플 송수신 클래스와 AppReceive() 함수와 AppSend() 함수를 통해 ID와 패스워드 데이터를 주고 받는다. 서버 클래스 역시 ServerReceive() 함수와 ServerSend() 함수를 통해 서버 송수신 클래스와 ID와 패스워드 데이터를 주고 받는다. 스마트라커 클래스는 현재 라커의 사용 상태를 Bool 값으로써 서버 클래스와 데이터베이스 클래스에게 전달해준다. 데이터베이스 클래스는 예약 클래스와 라커 클래스, 서버 클래스로부터 데이터를 받아 처리하여 다시 각 클래스에게 전달한다. 서버 클래스와 라커 클래스는 readDB() 함수를 통해 데이터베이스 클래스의 정보를 읽는다.

**2.4 그 외 다이어그램**

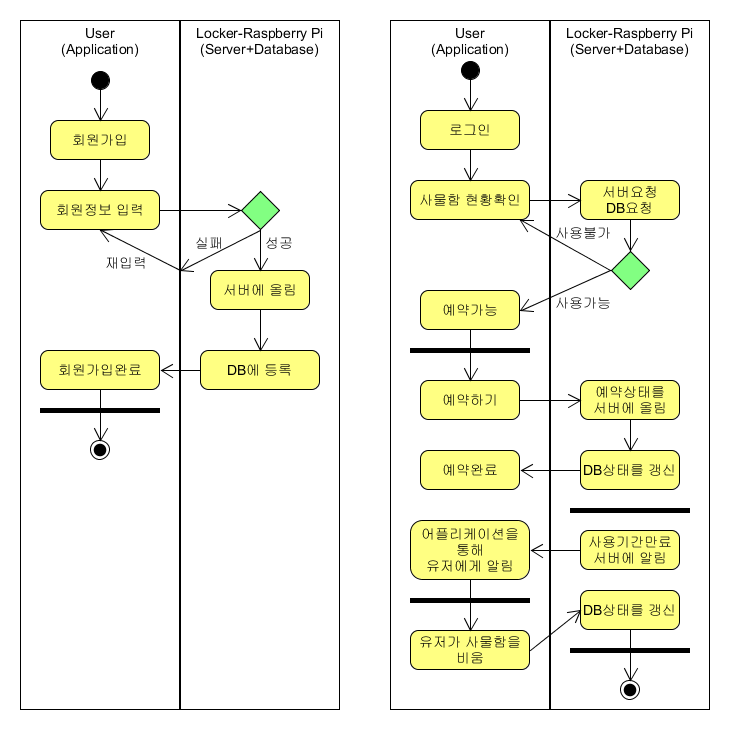
**2.4.1 배치다이어그램**



**<그림6 배치 다이어그램>**

스마트라커 시스템의 전체적 배치(그림 6)는 위와 같다. 라즈베리파이에서 Apache HTTP 서버가 작동하며 서버 위에 My SQL 데이터베이스가 올라가 있다. 사물함에서 NS-IRPSM 적외선 센서를 통해 얻은 데이터와 안드로이드 플랫폼을 기반으로 한 어플리케이션의 예약 정보는 서버를 통해 DB에 저장되고, 다시 실시간으로 LED와 앱 화면에 반영된다.

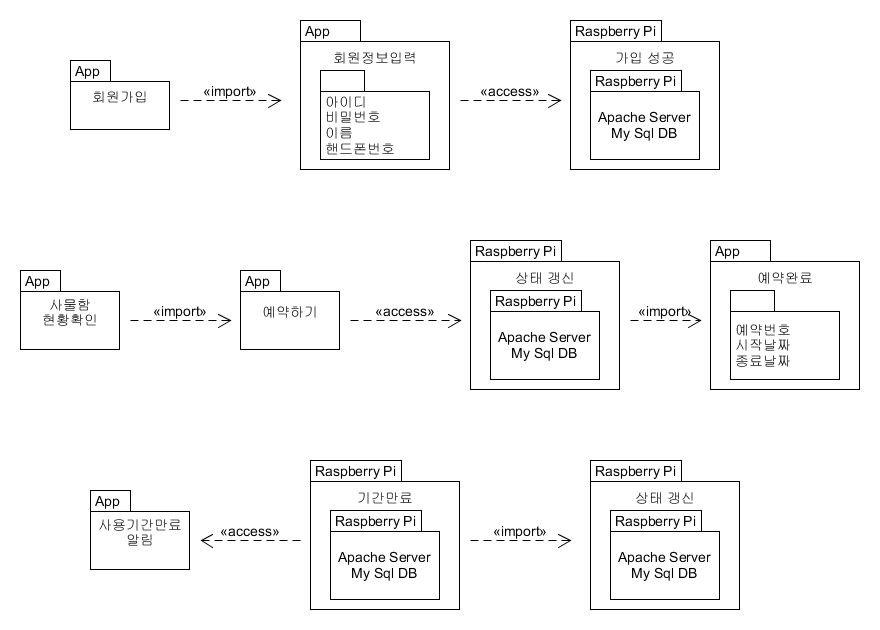
**2.4.2 액티비티 다이어그램**



**<그림7 액티비티 다이어그램>**

스마트라커 상호작용 액티비티(그림 7)은 다음과 같다. 유저가 앱에 사물함 현황을 요청하면 서버에서는 DB를 읽어 라커가 사용가능한지 아닌지 판별해준다. 만약 예약이 가능하다면 유저는 예약을 요청할 수 있다. 서버가 예약을 요청받으면 DB 상태를 갱신한 뒤, 유저에게 예약이 완료되었다고 알려준다.

**2.4.3 패키지다이어그램**



**<그림8 패키지 다이어그램>**

스마트라커 패키징 과정(그림 8)은 다음과 같다. 유저가 앱에 예약 정보를 <<import>>하면 라즈베리 파이의 서버에 <<access>>하여 DB의 예약 상태를 갱신한다. 예약이 완료되면 DB에 예약 번호, 시작 날짜, 종료 날짜가 <<import>>된다. 이후 사용 기간이 만료되면 서버는 DB에 <<import>>하여 상태를 갱신한다.