

목 차

<u>1. BeBlet의 개요</u>	<u>2</u>
1.1 연구 개발의 필요성	<u>2</u>
1.2 선행 연구 및 개발	<u>2</u>
1.3 전체 시스템의 구성, 기능, 설계 목표	<u>2-4</u>
<u>2. BeBlet의 구조</u>	<u>5</u>
2.1 기능 요구 사항	<u>5</u>
2.2 전체 구조	<u>5</u>
2.2.1 BeBlet의 동작 환경 및 구조	<u>5</u>
<u>3. BeBlet의 기능</u>	<u>6</u>
3.1 서버 관리	<u>6</u>
3.2 DB 관리	<u>6</u>
3.3 비즈니스 로직 처리	<u>6</u>
3.4 서버와 클라이언트 통신	<u>6</u>
3.5 유저 경험 향상	<u>7</u>
3.6 기능 명세서	<u>8-9</u>
3.7 API 리스트	<u>8-9</u>
3.8 프로그램 사용법	<u>9-12</u>
<u>4. 업무 분담 및 추진 일정</u>	<u>13</u>
4.1 업무 분담	<u>14</u>
4.2 추진 일정	<u>15</u>
<u>5. 참고 자료</u>	<u>16</u>

1. BeBlet 개요

1.1 연구 개발의 필요성

-대학생활에서 필기, 강의자료 복습 등으로 많이 쓰이는 태블릿 PC를 대여해주는 공유경제 플랫폼 어플리케이션을 만들려고 한다. 학생에게는 가격 측면에서 다소 부담이 있는 태블릿을 대여하여 사용할 수 있는 서비스이기 때문에 경제적으로 절약할 수 있고, 학과, 학년에 상관없이 교수님의 설명을 듣는 어떤 과목에서든 필요하기 때문에 학생들의 수요가 많을 것이다. 학생들은 강의 자료를 프린트를 하지 않아도 되고, 실시간으로 교수님의 설명을 필기하고, 관련 자료를 인터넷에서 찾아보는 등의 활동으로 학업 성취도를 높일 수 있다.

-태블릿은 강의 중 필기 등 수업 시간 이외에도 ebook 서비스로 독서량을 늘려 교양을 높이거나, coursera 등 유익한 학업 영상으로 관심 분야의 추가 공부를 하는 데에도 쓰일 수 있어서 활용도가 높다.

-현재 코로나 사태로 교육 방식은 크게 변화하였으며, 태블릿의 수요가 폭발적으로 증가하고 있다. 또한, 그 전부터 IT 발전의 흐름에 따라 교육 방식이 변화하고 있었고, 현 시점을 기점으로 교육에 많은 IT 기술 및 기기들이 사용될 것으로 예상 되기 때문에 “BeBlet”에 대한 수요가 충분할 것으로 보인다.

1.2 선행 연구 및 개발

-선행되어 개발된 서비스로는 에어비앤비의 숙박 공유, 클로젯 세어의 패션 아이템 공유, 키크고잉의 전동킥보드 공유 서비스등이 있다.

-“BeBlet”는 기존에 공유되지 않았던 콘텐츠인 “태블릿”을 공유하는 서비스를 제공하고, 유저 편의성에 집중한 앱을 개발할 예정이다.

-기존에 존재하던 서비스들과 달리 학업에 사용되는 물품을 대상으로 서비스를 진행하기 때문에 장시간 대여하는 사용자들이 많을 것으로 예상된다.

1.3 전체 시스템의 구성, 기능, 설계 목표

-설계 목표 :

ㄴ 표준 웹브라우저 환경에서 구동되는 웹 소스코드를 이용해 웹 어플리케이션을 제작 후 모바일 내장 브라우저로 감싸 하이브리드 앱을 빌드 함으로써, 크로스 플랫폼 개발을 진행한다.

ㄴ 개발 및 유지보수에 드는 비용을 줄이고, 개발하려는 서비스에 적합한 모델로 설계하려고 한다.

-기능 :

ㄴ 클라이언트 :

유저로부터 유저의 데이터를 받아 서버로 전송한다.

서버로부터 현재 서비스 진행 측의 물건 보유 현황과 유저 자신의 현재 상태(대여상태, 잔여 금액 등)를 클라이언트로 받아와 렌더링해 유저에게 보여준다.

ㄴ 서버 :

유저로부터 전송받은 유저 데이터를 저장 및 처리한다.

유저 편의를 위한 데이터 가공 처리한다.

-전체 시스템의 구성

ㄴ 클라이언트 : Vue.js를 이용해 SPA 제작-> Apache Cordova를 이용해 Hybrid App으로 빌드한다.

-추후 PWA를 도입해 유저 경험을 향상시켜줄 수 있다.

-이렇게 Virtual DOM을 이용해 동적으로 페이지를 렌더링하는 SPA를 제작하면 검색엔진에 잡히지 않아 추후 마케팅적으로 손해를 볼 수 있으므로 주요 기능 개발 후 SSR(Server Side Rendering)을 통해 서버에서 렌더링한 소스코드를 클라이언트로 전송해 유저 경험과 마케팅요소를 모두 살릴 수 있을 것으로 판단된다.

ㄴ DBMS : MySQL을 이용해 DB 설계 및 구현한다.

ㄴ 서버 : Java를 이용해 서버측 Application을 구현. Client로부터 받는 request에 payload로 response를 줄 때에도 동일하게 JSON 파일로 packing하여 전송한다. 캐시를 통해 요청 및 응답 속도를 향상시켜 사용자의 편의성을 높임. Spring을 이용하여 MVC 패턴으로 서버를 구축한다. 서버 인프라의 경우 AWS의 서버를 대여한다.

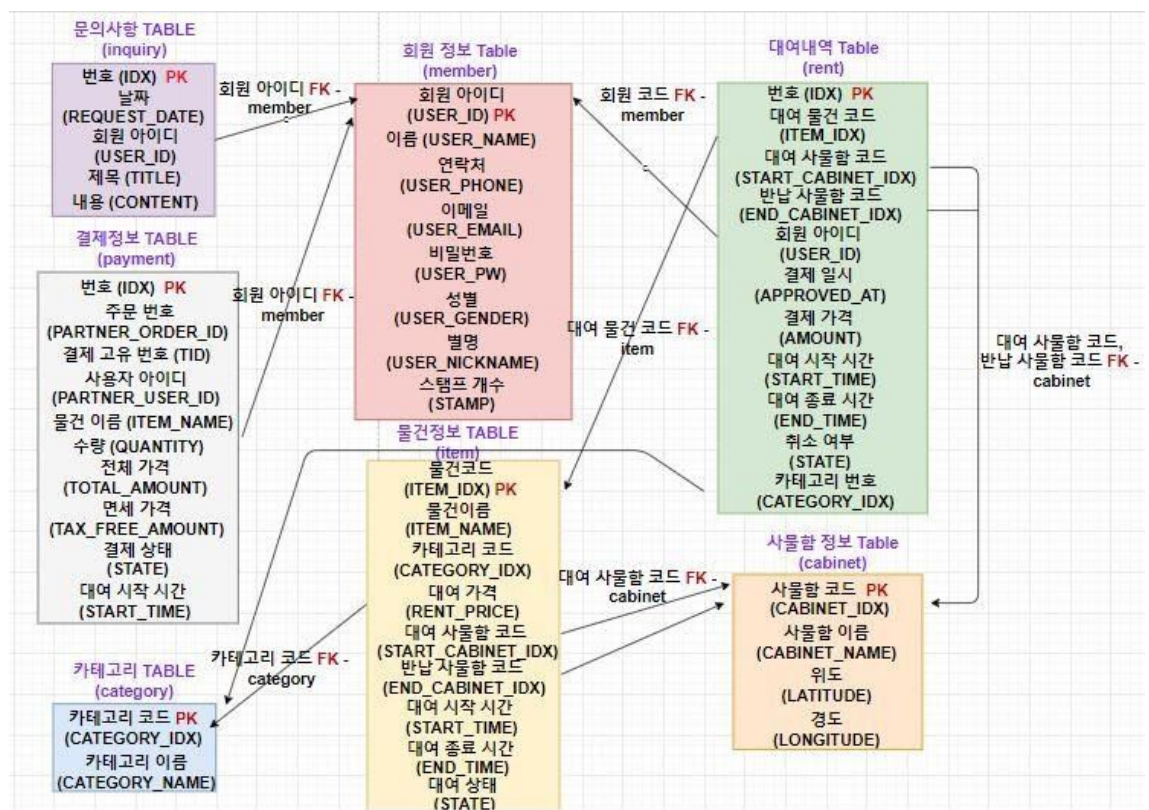


그림 1. DB 테이블 구조

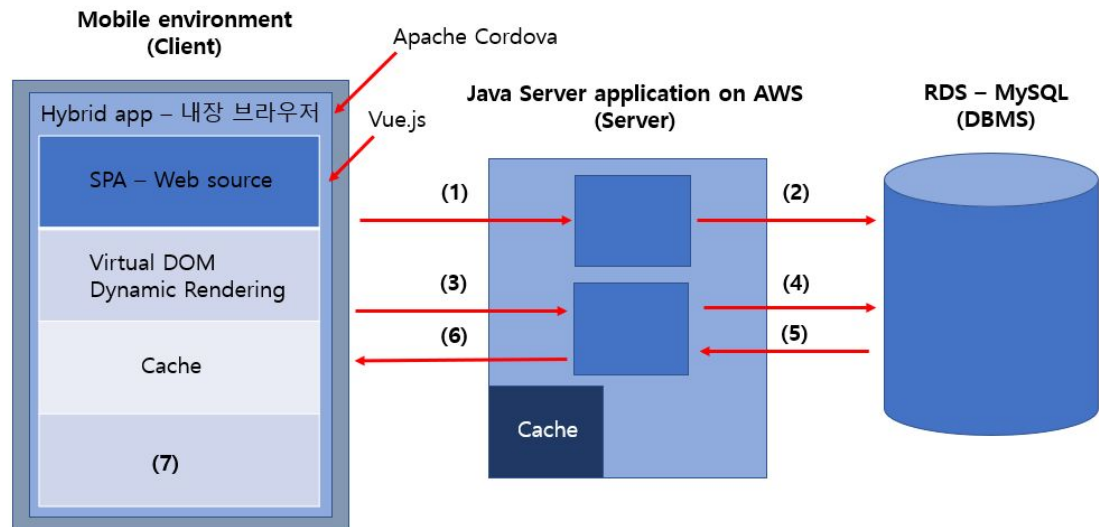


그림 2. 시스템 구성도

시스템 구성도 설명:

1. Client로부터 user data를 서버로 전송
2. Data 처리 및 저장 로직을 거쳐서 DBMS에 저장
3. User request (대여 가능 품목, 대여 중인 품목, 보유 금액 등 '조회' 요청)
4. User request에 따라 DB에 연결하여 데이터 가져옴
5. User request에 해당하는 data를 server측 Application으로 가져옴
6. User가 요청한 data response
7. 동적 Rendering을 통해 user가 요청한 정보를 native app과 거의 유사한 performance로 view

2. BeBlet 구조

2.1 기능 요구 사항

BeBlet은 다음 표와 같은 총괄적인 기능들을 요구하고 있다. 이 총괄 기능 정의는 다음 절의 BeBlet의 개념적 총괄 구조를 설명하기 위한 것이며 각 총괄 기능에 대한 세부적인 기능은 3절 BeBlet의 기능에서 제시된다. 2절의 나머지 부분에서는 각 기능을 구현하기 위한 개념적 세부 구조를 기술한다.

기 능	기능의 간략한 설명
서버 관리	여러 사용자가 비슷한 시간대에 몰려 서버에 부하가 걸릴 것을 고려해 유저 request처리에 load balancing을 진행한다.
DB 연결 및 관리	유저 데이터 저장 및 처리. 유저 편의를 위한 데이터 저장 및 처리
비즈니스 로직 처리	태블릿 대여, 결제, 잔여 캐쉬 환전 등의 비즈니스 로직을 처리하고 데이터를 업데이트
서버와 클라이언트 통신	각 유저로부터 직접 정보를 받는 클라이언트와 데이터들을 정리해놓은 서버 사이에 필요한 정보 전달을 하기 위한 기능
유저 경험 향상	웹 앱에서 새로고침시 발생하는 깜빡임을 없애고 네이티브 앱과 유사한 유저 경험을 줄 수 있도록 구현

<표 2-1> BeBlet의 기능

2.2 전체 구조

2.2.1 BeBlet 의 동작 환경

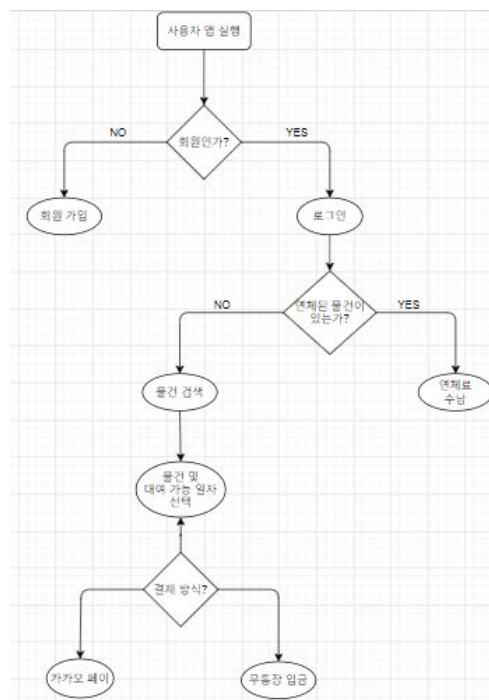


그림 3. BeBlet의 소프트웨어 동작 환경

3. BeBlet 기능

3.1 서버 관리

서버 인프라의 경우 AWS 서버를 대여해 이용할 예정이라 서버 OS설치, 서버 하드웨어 스펙은 신경 쓰지 않아도 괜찮다. 다만 서비스시 하나의 Layer에서만 User들의 request를 처리하게 된다면 성능이 떨어지게 돼 로드 밸런싱이 필수적이다. 추후 변경될 수 있는 사항이지만 현재까지 정해진 내용으로는 Web Server인 Apache에서 image나 단순 text data등 static data를 처리하고, WAS인 Tomcat에서 Server측 Application에 의해 변동되는 dynamic data를 처리할 예정이다. 이러한 방식으로 서비스를 구축하면 유저수가 늘어남에 따라 WAS만 늘려 Apache가 제공하는 로드 밸런싱 솔루션을 적용할 수 있다는 장점도 있다.

3.2 DB 관리

AWS가 서비스하는 분산 관계형 데이터베이스인 RDS를 통해 DBMS를 서버와 연동하여 사용한다. DBMS로는 MySQL을 사용하며, 데이터는 회원 정보, 물품 정보, 신청 물품에 대해 테이블을 나누어 관리한다. 클라이언트로부터 서버가 request를 받을 때, DB에서 쿼리를 진행해 데이터를 추출한 후, MyBatis의 mapper 인터페이스를 이용하여 데이터를 서버에 전송할 예정이다.

3.3 비즈니스 로직 처리

- 유저 인증 → SNS에서 제공하는 로그인 관련 API를 이용해 유저정보를 받아오고, 그 정보를 이용해 유저 인증을 진행. 유저 인증 진행 시 해당 인증 정보를 클라이언트에 캐싱해 자동로그인 기능 구현
- 대여 정보 인증 → DB에 대여 관련 정보를 저장하고 유저 인증 정보와 동기화해 대여 정보 인증을 진행. 대여 기간에 대한 정보를 DB에 저장할 때에는 DB 연결을 최소화해 성능 최적화할 수 있도록 설계해 저장.
- 결제 로직 → 이용가능한 PG사 API중 초기 시장진입이 목표이므로 이용 유저수가 많은 PG사의 API를 적용해 결제 기능을 수행. 결제 시 의도하지 않은 경로로 해당 요청을 서버로 보내도 서버에서 이를 차단해 결제 관련 버그를 잡아야 한다.

3.1.4 서버와 클라이언트 통신

클라이언트에서 유저로부터 입력 받은 정보를 객체로 묶는 과정을 거치고, 이를 JSON 파일로 묶어 서버로 보내는 request의 body 부분에 포함시켜 전송한다. 이 때 Vue.js에서 제공하는 data binding 기능을 이용해 Front-end 설계 시 나누어 설계했던 컴포넌트들 간의 data를 한 데 묶어(vuex library) 보낼 수 있다. 서버에서는 클라이언트로 전송받은 JSON 파일을 open해 JSON 파싱을 진행하고, 이를 저장 및 처리하고 response에 포함시킬 데이터를 JSON 파일로 다시 묶어서 클라이언트로 전송한다. 이 때 클라이언트는 JS로 구동되고, 서버 측에서는 Java로 구동되기 때문에 Data Type에 유의하여 통신을 진행해야 한다.

3.1.5 유저 경험 향상

서버와 클라이언트간 데이터 통신을 진행하면 기존 웹 앱에서는 전체 페이지를 다시 렌더링해야 했기 때문에 화면이 깜빡이고 비교적 느린 반응성을 보였다. 하지만 Vue.js의 경우 가상 DOM을 이용해 미리 Compile된 Source code를 비교적 적은 Cost로 제거하고 다시 렌더링하는 기법을 사용한다. 여기에 Ajax 통신을 이용해 화면 전체를 다시 렌더링 하는 것이 아닌 기존에 클라이언트에 렌더링 되어있던 부분 중 변동된 부분만 다시

렌더링해주는 방식으로 깜빡이지 않는 애플리케이션을 구현할 수 있다. 그 외에도 추후 도입할 예정인 PWA를 이용해 Native Application과 비슷하게 Device level의 기능을 이용할 수 있다.

3.6 기능 명세서

기능	비고
카카오 로그인	카카오톡 연동을 통한 자동 로그인 / 로그아웃 구현
이용 가능 물품 보관함 지도 API 위에 Display	이용 가능한 물품 보관함 정보를 서버로부터 받아온 후, 카카오맵 API를 이용해 Display
물품 보관함 추천 및 지도 위에 display	추천 물품 보관함 좌표에 위치한 Pin은 다른 이미지로 변경해 Display
카카오페이 결제	카카오페이 API를 이용해 클라이언트에서는 카카오톡 실행 intent를 삽입해 결제창을 Display하고 결제 인증 정보는 서버로 Redirect 될 수 있도록 설정
대여 완료 후, 관련 정보 반환	대여 후 하단 ‘마이페이지’ 탭에서 대여중인 물품 혹은 대여 예정 물품 Display
스탬프 적립	물품 대여시 물품 대여 및 반납위치를 모두 추천 물품보관함을 선택해 대여하면 이후 서비스 이용시 추가 혜택을 받을 수 있는 스탬프 제공
캐시를 이용한 물품 정보 디스플레이	서버와의 통신에서는 물품의 Index 정보만을 이용해 통신하고, 물품 세부 정보들은 Client Local Storage에 캐싱해 서버 - 클라이언트간 통신에서의 부하를 줄임
대여	카카오맵, 카카오페이 결제등의 기능들을 활용한 대여 기능
반납	현재 시각보다 대여 시작시간 이후이면 반납 버튼이 자동으로 활성화되며, 서버로 반납 요청을 보냄
결제 취소	현재 시각보다 대여 시작시간 이전이면 결제 취소 버튼이 자동으로 활성화 되며, 서버로 결제 취소

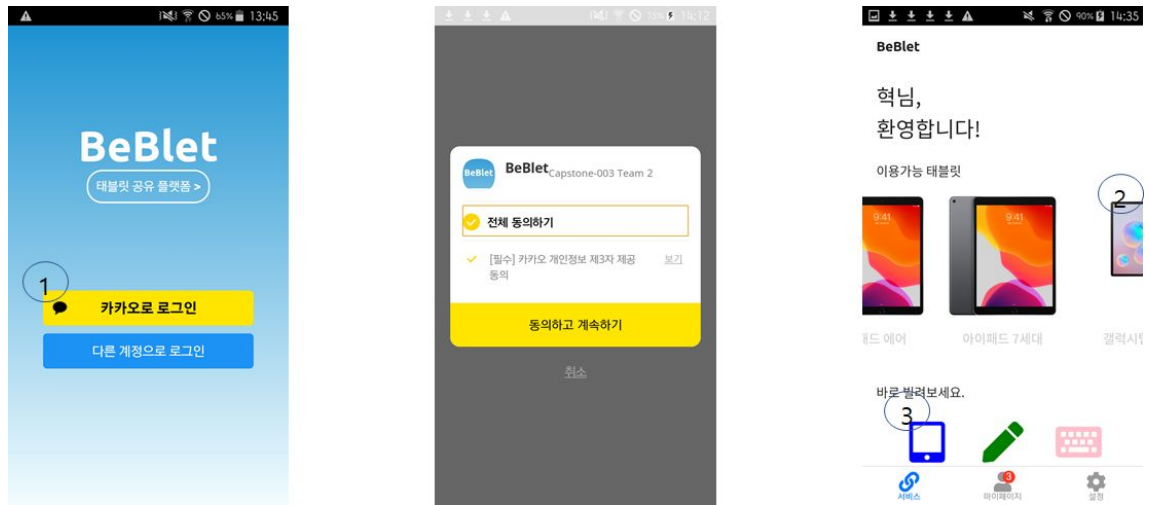
요청을 보냄

3.7 API 리스트

기능	URL	설명
카카오 로그인	/joinAndLoginRequest	- DB에 저장된 유저는 바로 로그인 - DB에 없는 유저는 DB에 새로 생성 및 초기화 후 로그인
물품 목록 요청	/itemList	- DB에 저장된 물품 정보 리스트를 반환 - 물품 리스트 정보 캐시
사용자 정보 요청	/userInfo	- 사용자가 마이페이지에서 스탬프를 비롯한 사용자 정보를 볼 수 있다. - 사용자 정보 반환
사용자 정보 수정	/changeUserInfo	- 사용자는 자신의 정보(닉네임, 메일 등)를 수정할 수 있다.
문의하기	/inquiry	- 문의 사항을 전달받아 DB에 저장
사용 가능 물품 보관함 및 추천 보관함 검색	/recommendCabinet	- 해당 물품을 해당 시간에 사용 가능한 물품 보관함 리스트와 추천 물품 보관함을 탐색 및 반환
대여 신청	/apply	- 결제 후 최종 대여 신청 api - 사용자에게 스탬프 부여 (추천 사물함 이용시 스탬프 2개 부여) - 사용자 정보 업데이트 - 대여 내역을 DB에 저장
카카오 페이	/kakaoPay	- 대여한 것에 대한 카카오페이 결제 진행 - DB에 결제 내역 저장
대여 취소	/kakaoCancel	- 카카오 페이 결제 취소 진행 - 해당 대여 내역 삭제 - 결제 내역 상태 업데이트
대여 목록	/returnRentList	- 사용자가 마이페이지에서 자신의 대여 기록을 연람 가능 - 사용자의 대여 기록 반환
결제 내역	/returnPaymentList	- 사용자가 마이페이지에서 자신의 결제 기록을 연람 가능 - 사용자의 결제 기록 반환

물품 반납	/returnItem	- 사용자가 자신이 대여한 물품을 반납
대여 취소 또는 반납 선택	/CancleOrReturn	- 마이페이지에서 해당 대여 내역을 누르면 사용 기간과 현재 시간을 비교해 대여 취소인지, 반납인지 판단하여 안내문이 뜨도록 한다.

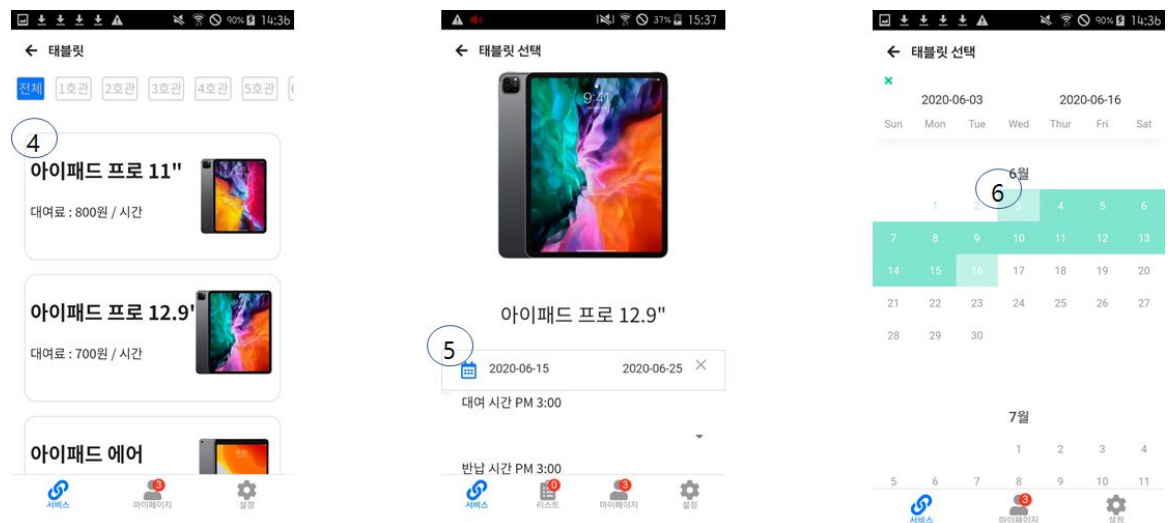
3.8 프로그램 사용법



① 카카오톡 로그인 버튼 터치
-> 카카오톡 연동 동의 후 로그인

② 자동 carousel display

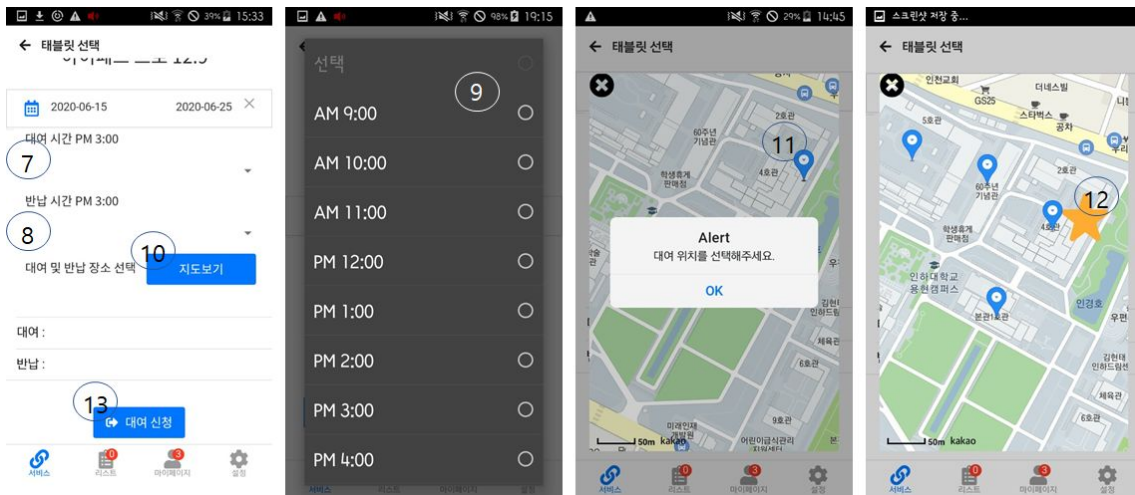
③ 대여를 원하는 아이템 카테고리 선택



④ 대여를 원하는 아이템 선택

⑤ 대여를 원하는 날짜 선택 위해 달력 오픈

⑥ 대여를 원하는 날짜 선택



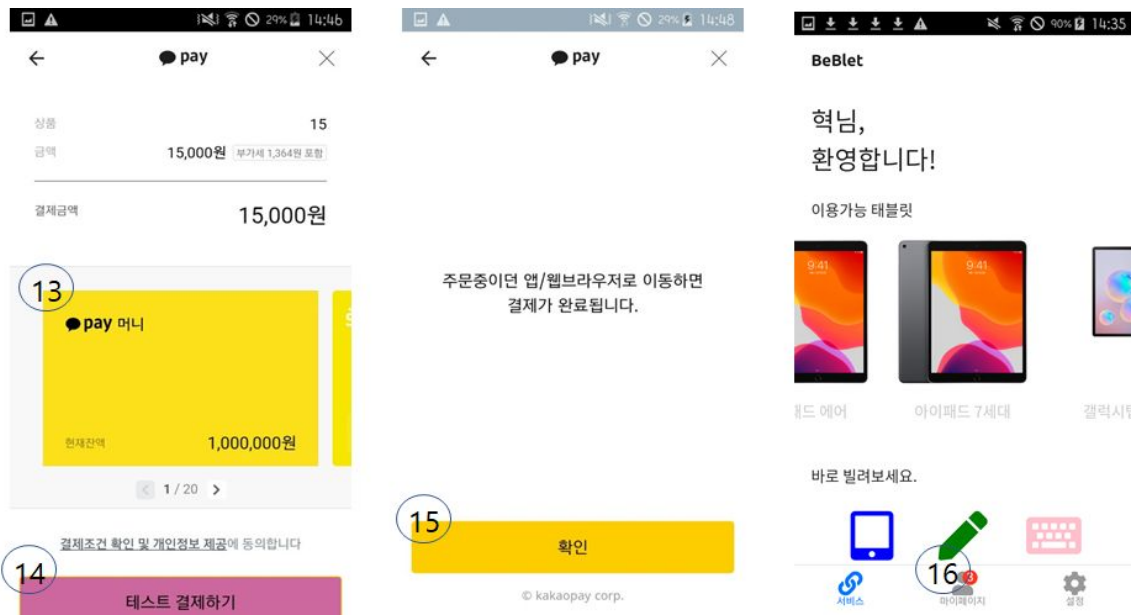
⑦ ⑧ ⑨ 대여 및 반납 시간 선택

⑩ 지도 팝업 오픈

⑪ 대여 및 반납 위치 선택

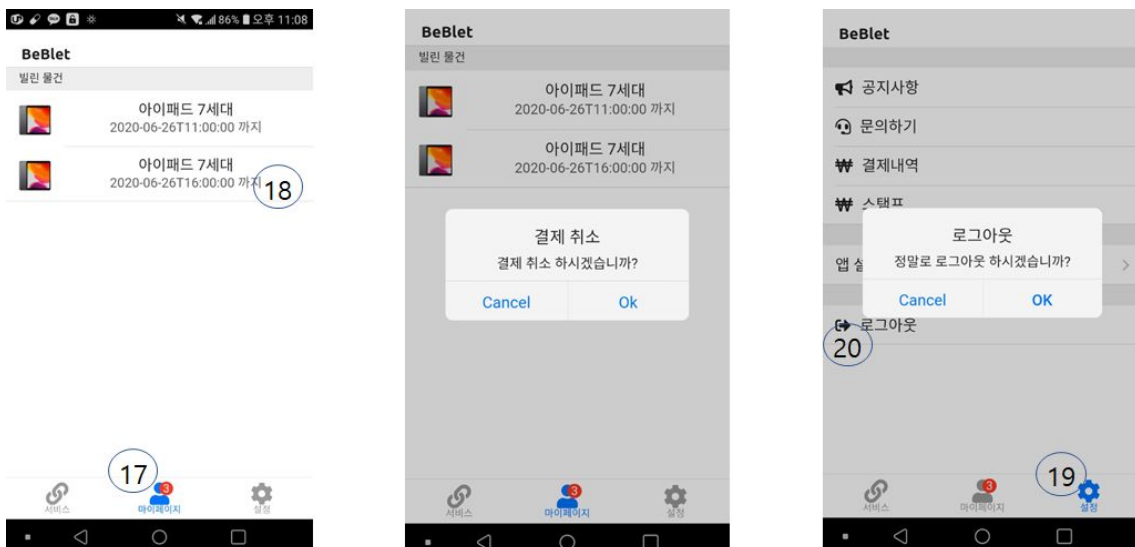
⑫ 추천 대여 및 반납위치 선택

⑬ 대여 신청 및 결제창 오픈



⑬ ⑭ ⑮ 카카오페이 결제

⑯ 마이페이지 탭 이동



⑰ 마이페이지 탭

⑱ 현재 대여중인 물건 display 및 반납, 결제취소 버튼

⑲ 설정 탭 오픈

㉔ 로그아웃

4. 업무 분담 및 추진 일정

4.1 업무 분담

이름	역할
강혁	클라이언트 제작, 서버-클라이언트 통신
손지민	서버측 애플리케이션 제작, DB 설계
유서정	서버측 애플리케이션 제작(REST API 구현), 서버 인프라 구축 및 서비스 배포

4.2 추진 일정

항목	시기	2020년-1학기											
		4월 초	4월 중	4월 말	5월 초	5월 중	5월 말	6월 초	6월 중	6월 말	7월 초	7월 중	7월 말
클라이언트 개발 [강 혁]													
기능별 vue 컴포넌트 개발													
백엔드 - 프론트엔드 API 연동													
서버 - 클라이언트 간 http 통신 디버깅													
애플리케이션 빌드													
백엔드 로직 개발 [손지민]													
DB 설계													
대여 및 결제 관련 비즈니스 로직 개발													
매칭 및 에스프로 비즈니스 로직 개발													
AWS에 배포 후 디버깅													
서버 개발 [유서정]													
AWS 서버 구축													
DB 및 웹 서버 개발													
클라이언트 배포 테스트													
백엔드 애플리케이션 배포 테스트													
서비스 시연													

미정

5. 참고 자료

- Kakao Developers - kakao login :
<https://developers.kakao.com/docs/latest/ko/kakaologin/common>
- Kakao Developers - kakao pay :
<https://developers.kakao.com/docs/latest/ko/kakaopay/common>
- Kakao Developers - kakao map :
<https://developers.kakao.com/docs/latest/ko/kakaomap/common>
- Apache Cordova Document - Hybrid app reference
<https://cordova.apache.org/>
- Vue.js Document
<https://kr.vuejs.org/v2/guide/index.html>
- 유튜브 발표 동영상 링크 :