



RFID-CART SOLUTION

PORTFOLIO



Smart RFID Solution

CONTENTS

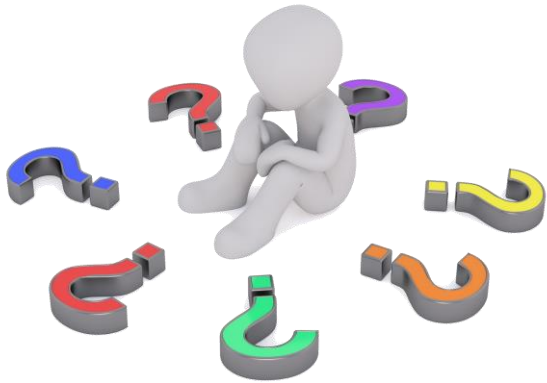


- 프로젝트 소개
- 설계 내역
- 개발 내역
- 구현 사진
- 피드백

1. 프로젝트 소개

고객유치
회전율 ↑

편리한
쇼핑 & 결제



현재 국내 오프라인 대형 매장에서 소비자가 결제 관련하여 큰 고충을 겪는다.

그 이유는 결제 대기시간 때문인데, 구매자가 일일이 물건들을 카트에 빼고, 다시 담고 하는데 걸리는 시간이 물건의 양에 비례해서 늘어나기 때문이다.

1. 프로젝트 소개



NO LINES. NO CHECKOUT.
(NO, SERIOUSLY.)

**JUST WALK OUT
SHOPPING**

RFID태그와 RFID리더기, 안테나 이 세가지 장치 및 센서들을 통하여 계산대를 지나가면 한번에 일괄 구매가 가능한 시스템을 생각하게 되었다.

구매자가 일일이 물건을 내리고 바코드를 찍을 필요 없이, 카트에 담겨있는 상태로 특정 지역을 지나가는 순간 계산대에 구매 목록들이 뜨고, 한번에 결제가 가능하게 하여 구매 소요시간을 줄여주고, 어플을 활용하여 간편결제 시스템까지 만들어 준다.

2. 설계 내역

구분	항목
OS	LINUX(Ubuntu)
Server	Apache-tomcat 8,Node-JS
데이터베이스	Maria DB
개발 언어	Java, Jsp, HTML5, CSS, JavaScript,
개발 환경	Eclipse Luna, gedit
라이브러리	java-json.jar Json-simple-1.1.1 MariaDB-java-client-1.2.0 Nodejs npm(mysql,restful,async,express)
통신 방법	Restful-API

2. 설계 내역

1. S/W 설계(DB)

DataBase : RFID

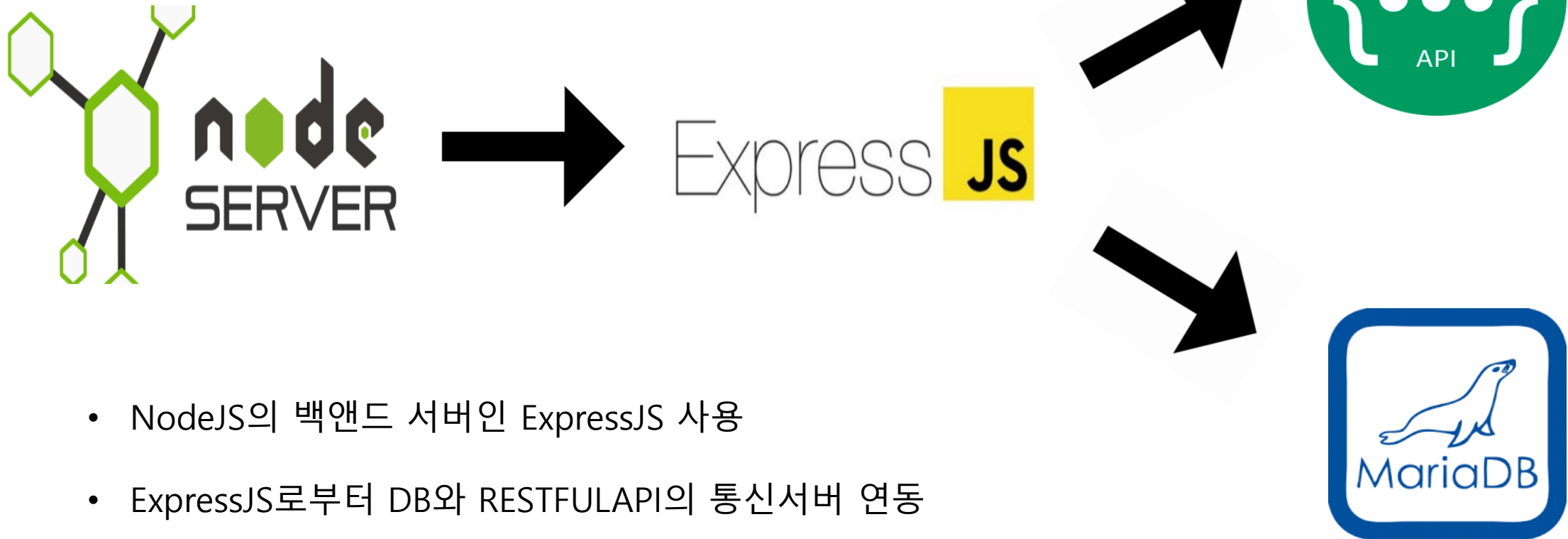
CA01(카트1번)	
Item	Varchar
Price	Int
Tag	Varchar
Amount	Int

CA02(카트2번)	
Item	Varchar
Price	Int
Tag	Varchar
Amount	Int

Goods(상품 목록)	
Item	Varchar
Price	Int
Tag	Varchar
Amount	Int

2. 설계 내역

1. S/W 설계(NodeJS RESTFUL-API)

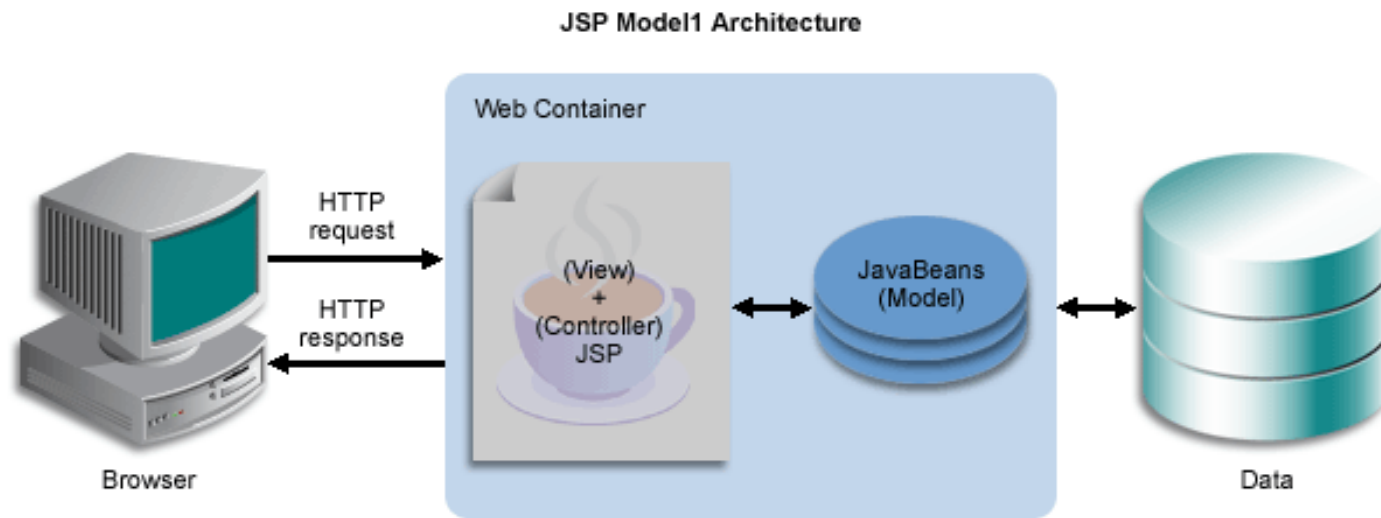


- NodeJS의 백엔드 서버인 ExpressJS 사용
- ExpressJS로부터 DB와 RESTFULAPI의 통신서버 연동
- RESTFULAPI : PUT,POST,GET 사용

2. 설계 내역

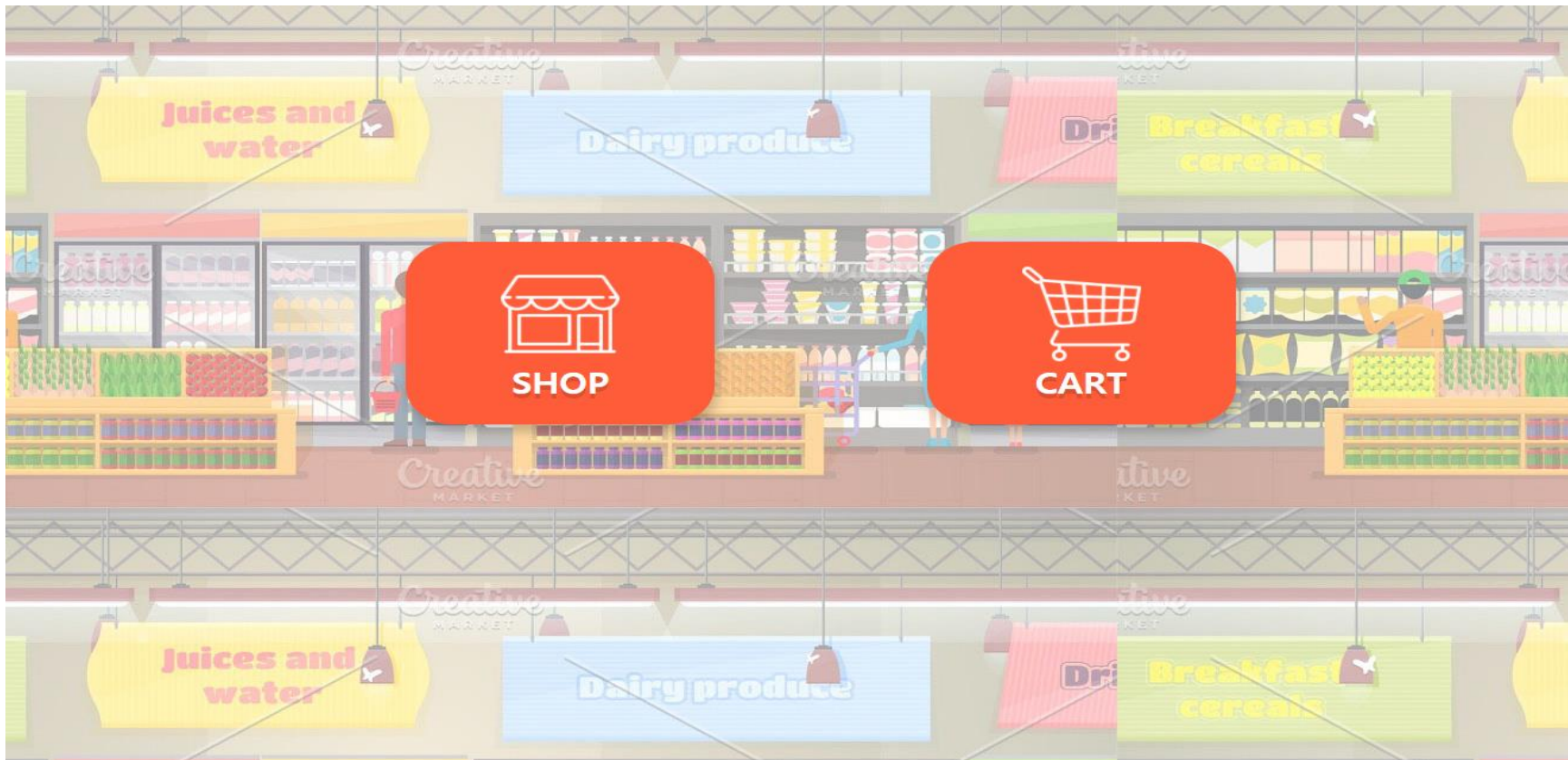
1. S/W 설계(Web)

MVC패턴 MODEL 1 사용



3. 개발 내역

메인 홈페이지



매장 정보



메인

No	제목	작성일	작성자	조회수
1	삼겹살 특가행사 12월 15일 까지	2019.12.01	운영자	12
2	신년맞이 50%세일	2019.12.01	운영자	15
3	아이를 찾습니다.	2019.12.03	고객관리과	50
4	11거 1111 불법주차 차량 배주세요	2019.12.04	주차 관리과	10
5	저녁 8시부터 10시까지 한우 세일	2019.12.05	운영자	30

3. 개발 내역

카트 중간 계산 목록

메인

물품 목록

No	상품명	상품번호	수량	금액
1	COFFE	EA02	1	1500
2	PIZZA	EA01	1	20000
3	COFFE	EA02	1	1500
4	햄버거	EA07	1	5000
5	MONSTER	EA08	1	2000
6	닌텐도 스위치	EA05	1	580000
합 계			6	610000

3. 개발 내역

RFID RESTFULAPI 통신내역

Universal Reader Assistant

Reader URI : COM5 Reading Stop Reading Disconnect 43°C

Tag Results Tag Inspector Write EPC User Memory Lock Tag Untraceable Authenticate

#	EPC	TimeStamp(msec)	RSSI(dBm)	ReadCount
1	E200001B631502661580ED15	06:40:40.914 PM	-41	152
2	E200001B631502661710ECDD	06:40:40.743 PM	-35	76
3	E200001B631502661890EC99	06:40:41.876 PM	-45	37
4	E200001B631502661690ECE9	06:40:40.600 PM	-33	71
5	E200001B631502661590ED0D	06:40:41.795 PM	-41	65

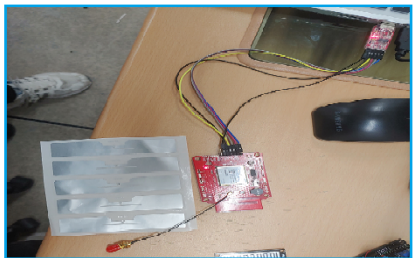
Settings/Status

Connect

Reader Type : ☒ Serial ☐ N

Reader Name : USB Serial Po

☐ Transport I

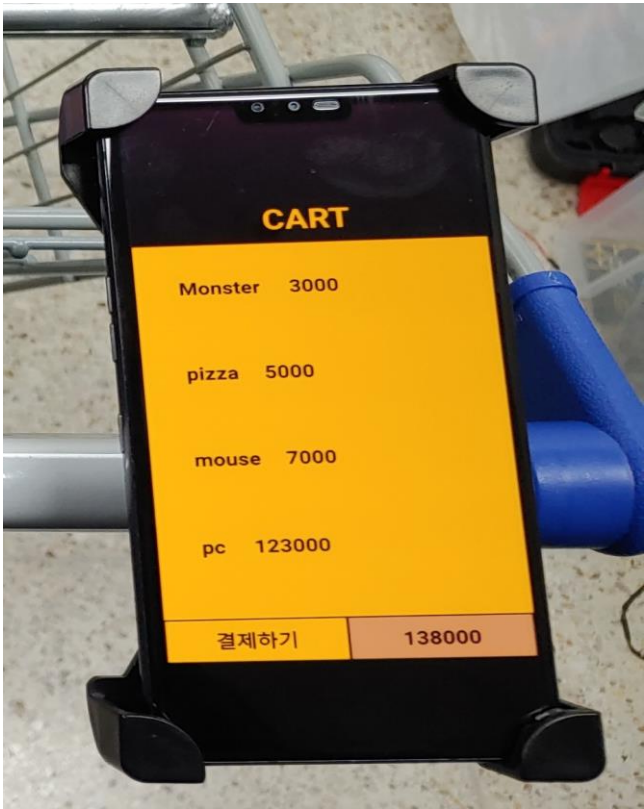
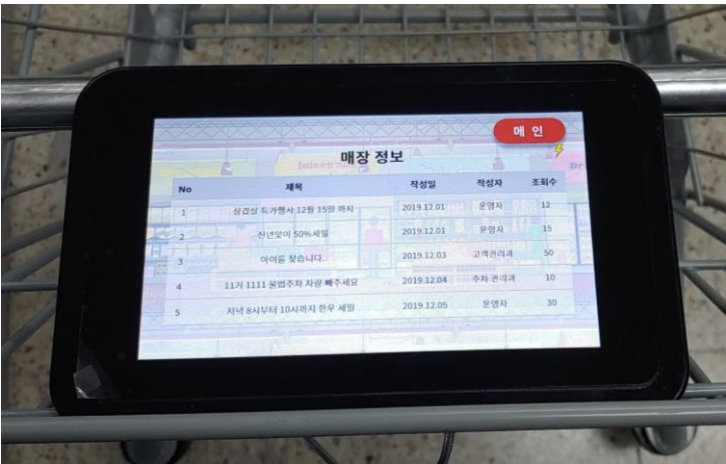
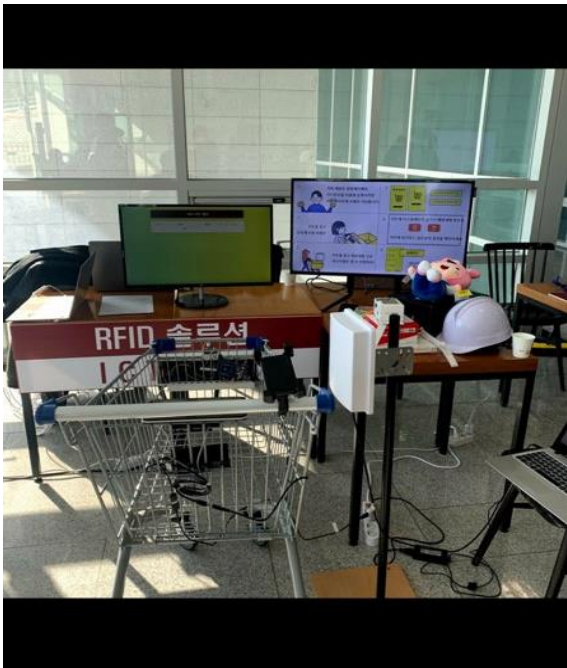


item	price	tag	amount
COFFE	1,500	EA02	1
햄버거	5,000	EA07	1
MONSTER	2,000	EA08	1
닌텐도 스위치	580,000	EA05	1
PIZZA	20,000	EA01	1
COFFE	1,500	EA02	1

1. RFID태그와 리더기를 이용해 얻은 값을 RESTFUL-API POST 통신을 이용하여 서버 데이터베이스에 값을 저장
2. 저장된 값을 데이터 전처리 과정을 이용하여 데이터베이스 goods테이블에 있는 칼럼들로 cart01테이블에 저장

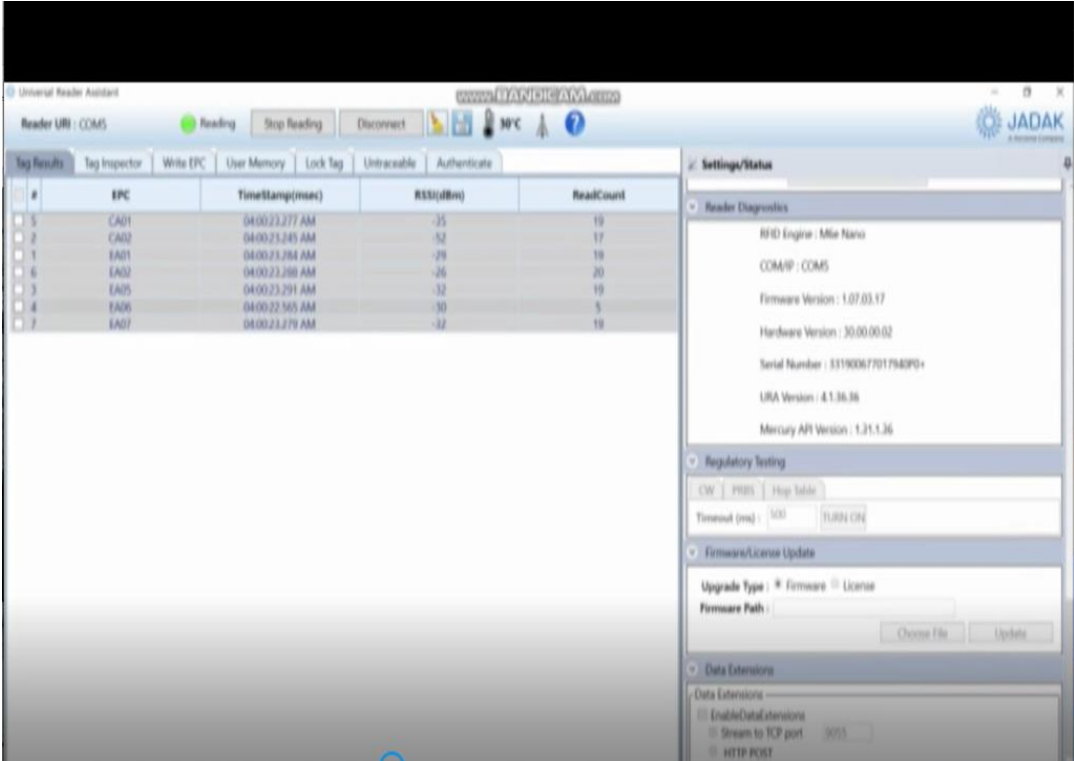
4. 구현 사진

카트 외관 및 UI



4. 구현 사진

통신 구현 사진



5. 피드백



- 창업동아리 아이템으로 기존 틀을 벗어난 새로운 시도를 하여 팀원들과 흥미를 가지며 재미있게 개발을 진행할 수 있었다.
- 협업을 하여 진행한 팀 프로젝트로 버전관리 및 의사소통의 중요성을 크게 알 수 있는 작업이었다.



- 한정된 시간과, 한정된 예산으로 프로젝트를 진행하다보니, 하드웨어적으로 깔끔하게 마감처리 못한 부분이 많았다.
- RESTFUL-API 데이터 전처리 과정에 대한 코드부분들이 클린코드가 되지 못하여 수정이 필요함.