

스마트 카트(Smart Cart)

팀 명 : 쇼핑을도와조

팀구성원

No.	구분	성명	소속명	No.	구분	성명	소속명
1	팀장	위혜진	동덕여자대학교	6	팀원		
2	팀원	양혜임	동덕여자대학교	7	팀원		
3	팀원	이아룬	동덕여자대학교	8	팀원		
4	팀원	이지호	동덕여자대학교	9	팀원		
5	팀원			10	팀원		

2017. 10. 06

1. 목차

2. 개요

2.1 작품명

2.2 작품 개요

2.3 목적

3. 작품 설명

3.1.1 Software 구성

3.1.2 Hardware 구성

3.2 Software 흐름도 및 클래스 다이어그램

3.3 Software 기능

3.4 프로그램 사용법

3.5 개발환경

4. 프로그램 설명

4.1 파일 구성

4.2 함수별 기능

4.3 주요 함수의 흐름도

4.4 기술적 차별성

5. 응용 분야

6. 향후 계획

7. 제작자 정보

8. 개발 단계별 기간 및 투입 인원

2. 개요

2.1. 작품명

스마트 카트(Smart Cart)

2.2. 작품 개요

기존의 카트를 사용하기 위해선 항상 손으로 카트를 잡고 방향을 조절해야 합니다. 그러나 몸이 불편한 장애인이나 노약자는 카트를 미는 것 자체가 쉽지 않고 마트의 주요 고객인 주부의 경우 아이를 돌보며 카트를 밀어야 하는 불편함이 있습니다. 기존 카트의 이러한 단점들을 해결하고자 손을 자유롭게 쓸 수 있으며 거동이 불편한 사람도 쉽게 쓸 수 있는, 사용자를 따라다니는 기능이 있는 스마트 카트 Smart Cart를 개발하였습니다. 고객은 모바일의 앱을 통하여 쇼핑에 필요한 여러 정보를 얻을 수 있으며 블루투스를 통해 카드(아두이노)와 연결되어 자유롭게 쇼핑할 수 있습니다. 카트는 사용자를 따라오고, 카트에 상품이 담기면 자동으로 구매 리스트에 추가되어 쇼핑 목록을 확인 할 수 있습니다. 관리자는 웹 페이지를 통해 마트 내부의 상품 정보 및 카트의 사용자 연결 상황을 알 수 있고, 회원 관리를 효과적으로 할 수 있습니다. 저희의 스마트 카트 Smart Cart 프로젝트는 쇼핑을 더욱 편리하게 하여 마트 이용 고객이 두 손을 자유롭게 보다 편리하고 합리적인 쇼핑을 할 수 있도록 도와주고, 마트 관리자의 관리 업무에도 도움을 줄 것입니다.

2.3. 목적

스마트 카트의 개발 목표는 마트에서의 쇼핑을 더욱 편리하게 만드는 것입니다. 사용자를 따라다니는 카트의 주행뿐만 아니라 카트에 담기는 상품 정보를 자동으로 인식하고 상품 검색을 통한 위시리스트 작성할 수 있는 기능 등을 통하여 기존의 쇼핑을 하며 고객의 불편함을 해소하고 보다 합리적인 구매를 할 수 있도록 도와줍니다. 또한 관리자도 쉽게 회원, 상품, 카트 등의 정보에 접근하고 수정할 수 있도록 함으로써 마트 내의 정보를 편리하게 관리할 수 있도록 합니다. 더 나아가 지속적인 개발을 통하여 추후 스마트 카트가 자율 주행으로 고객에게 상품의 위치를 안내하고 자동으로 결제를 도와주는 기능을 추가하여 보다 편리한 도움을 주기를 기대합니다.

3. 작품 설명 (최대한 자세하게 기술)

3.1.1 Software 구성

o 서버:

마트 운영에 필요한 상품, 카트 등의 데이터 및 회원 정보를 담고 있는 데이터베이스를 구축하였습니다. 또한 카트 연결 상황, 상품 및 회원 관리 등의 마트의 운영을 위한 관리자 웹 페이지를 구현하였습니다. 웹 페이지를 통해 데이터베이스에 직접 접근하지 않아도 보다 더 쉽게 상품, 회원, 카트 연결 상황 등의 정보에 대하여 검색, 추가, 삭제를 가능하게 하였습니다.

모바일 어플리케이션과의 통신으로 회원가입을 통한 회원의 정보를 데이터베이스에 추가할 수 있습니다. 또한 모바일 어플리케이션의 로그인, 상품검색(이름별, 종류별, 브랜드별 검색), 카트 연결 상황, nfc 태그 정보에 따른 상품 정보 인식 등의 요청에 따라 DB 검색 후 해당 결과를 json 형태의 데이터로 만들어 보내줍니다.

o 모바일:

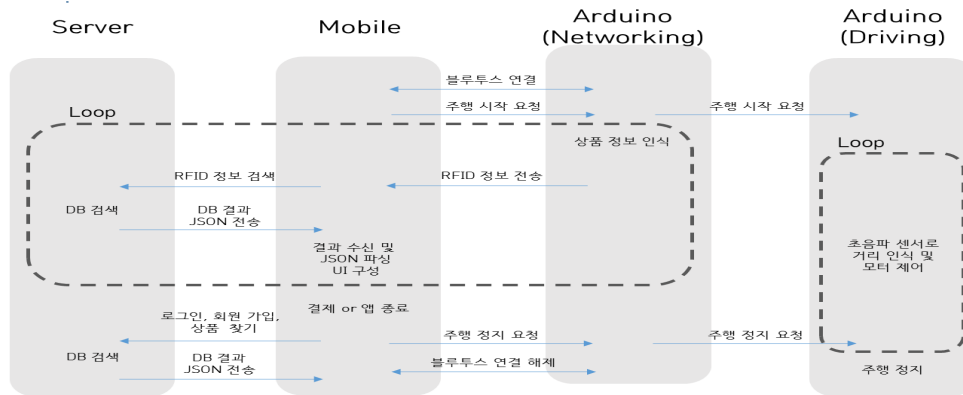
어플리케이션을 이용하여 사용자의 스마트 카트 사용에 대한 접근성을 높였습니다. 회원가입, 로그인, 상품 검색, 위시리스트, 쇼핑 기능을 담고 있습니다. 상품 검색은 서버와의 통신을 통해 DB에 쿼리를 전송하고 json 형태의 데이터를 받아 파싱하여 원하는 상품을 위시리스트에 추가할 수 있습니다. 쇼핑 기능은 현재 사용할 수 있는 스마트 카트의 목록을 보여주며, 카트에 상품이 담기면 상품들의 총액을 자동으로 계산하여 보여줍니다.

3.1.2 Hardware 구성

o 아두이노:

두 개의 Arduino UNO 보드를 사용하여 통신과 주행 기능을 나누어 수행합니다. 모바일과 블루투스로 통신하기 위한 블루투스 모듈을 사용합니다. 상품의 NFC태그를 정보를 인식하기 위한 RFID모듈을 사용합니다. 사용자와의 거리를 지속적으로 측정하기 위한 초음파 센서를 가지며 바퀴의 구동 및 방향 제어를 위한 모터 드라이버와 DC 모터가 있습니다.

3.2. Software 흐름도 및 클래스 다이어그램 (개발언어에 따라 선택)



- Software 흐름도 및 설명

모바일과 아두이노의 블루투스 연결 -> 모바일에서 아두이노에게 주행 시작 요청 신호를 보냄 -> 신호를 받은 아두이노가 DC모터 제어 시작 -> 상품을 카트에 담았을 때 아두이노가 RFID모듈 사용, 상품의 NFC태그정보 인식, 모바일로 태그정보 전송 -> 모바일에서 수신 받은 태그 정보를 사용하여 서버와 통신, 관련 정보가 있는지 DB 검색을 요청 -> 서버에서 데이터베이스 검색을 수행 -> 해당 요청에 관한 질의 결과가 있으면 모바일에게 json형식의 데이터를 보내줌 -> 모바일에서 전달받은 json 형식의 데이터를 파싱하여 사용, 상품의 이름과 가격 표시, 상품들의 총액을 계산하여 표시 -> 사용자가 결제를 원할 경우(계산을 위한 특정 지점에 가거나 계산 버튼 클릭) 모바일과 아두이노와의 데이터 송수신을 멈춤, 모바일에서 결제 페이지로 화면이 전환됨

3.3. Software 기능 (알고리즘 설명 포함)

[서버 <-> 모바일 <-> 아두이노] 의 형태로 통신하고 운용됩니다.

o 서버

모바일의 로그인, 상품 검색, 상품 정보(nfc태그)등에 관한 요청에 따라 그에 알맞은 정보를 DB에서 찾아 json 형태의 데이터를 만들어 모바일로 전송합니다. 모바일에서 회원가입 정보를 수신하여 DB에 저장합니다. 관리자 웹 페이지에서도 서버를 통하여 DB에 데이터를 요청하고 결과를 처리합니다.

o 모바일

사용자의 쇼핑을 돕기 위한 어플리케이션입니다. 서버와의 통신을 통하여 DB에 접근 가능하여 회원가입, 로그인, 상품 검색 등의 기능을 수행합니다. 상품 검색을 통하여 메모장 역할을 하는 위시리스트를 작성할 수 있습니다.

또한 블루투스를 이용하여 아두이노(이하 카트)와 통신합니다. 어플리케이션은 카트의 주행시작 및 멈춤 요청 신호를 카트와 주고받으며 쇼핑 시에는 상품 정보(nfc태그 정보)를 카트로부터 전달 받습니다.

서버와의 통신은 사용자의 쇼핑이나 검색 행위에서 비롯되는 상품 정보를 서버에 요청하고 그에 따른 결과인 json 형식의 데이터를 파싱하여 사용자가 보기 편하도록 리스트로 만들어 보여줍니다.

o 아두이노

아두이노는 크게 주행용 보드와 통신용 보드로 나누어집니다.

먼저 통신용 보드는 모바일과의 통신, RFID 정보 인식 및 I2C 통신에서의 마스터 역할을 합니다. 모바일과의 블루투스 연결을 통하여 주행 시작 신호가 들어오면 I2C 통신을 통하여 주행용 보드와 주행 준비 및 주행 시작 신호를 주고받습니다. 카트 주행 중에는 RFID 인식을 통하여 상품 정보를 읽어오고 모바일로 전송합니다.

주행용 보드는 초음파 센서를 이용하여 카트 바로 앞의 고객 위치를 지속적으로 인식하고 거리를 측정합니다. 이 정보를 통하여 주행 준비 시 어플리케이션을 구동하는 고객이 근처에 오기까지 대기합니다. 카트 주행 시에는 모터 드라이버를 제어하여 카트의 방향을 조절하여 고객을 따라다닙니다. 또 I2C 통신에서 슬레이브 역할을 하여 마스터에서 필요한 정보를 요청할 경우 전송해 주는 역할을 맡습니다.

3.4. 프로그램 사용법 (Interface)

- 고객

- 1) 스마트 카트의 사용자는 모바일 앱의 블루투스 통신을 이용하여 현재 사용 가능한 카트를 확인 한 후 사용 할 카트를 고르고 연결합니다.
- 2) 연결이 완료되면 카트가 사용자에게 움직이기 시작했다는 신호를 주고 사용자를 따라다니기 시작합니다.
- 3) 앱은 서버와 통신하여 DB에 접근 가능함으로 사용자는 앱을 통해 회원가입 및 로그인 등의 사용자 인증을 할 수 있습니다. 또한 원하는 상품을 검색하여 현재 마트에

있는 상품에 대한 정보를 받아 볼 수 있습니다. 이와 같이 상품 검색을 통하여 메모장 역할을 하는 위시리스트를 기록할 수 있습니다.

4) 아두이노를 활용하여 만든 카트는 사용자와의 거리를 지속적으로 측정합니다. 따라서 사용자와의 일정 간격을 두고 따라다니며 사용자와 가까운 특정 거리에서는 움직임을 멈추고 사용자의 움직임을 기다립니다. 따라서 사용자는 카트를 손으로 밀 필요 없이 쇼핑을 합니다.

5) 사용자가 상품을 골라 카트에 담게 되면 카트가 상품의 nfc 태그를 인식하고 블루투스 통신으로 앱에게 정보를 전송해줍니다. 앱은 전송 받은 상품의 nfc 태그를 서버에게 보내어 nfc 태그에 해당하는 상품의 정보를 가져옵니다. 앱은 서버에서 받아온 상품의 정보를 토대로 현재 카트에 담겨 있는 상품 목록을 만들어 총액을 자동으로 계산하여 알려주기 때문에 사용자의 계획적인 소비를 도와줍니다.

6) 사용자와 카트가 쇼핑을 마치고 계산대 근처로 가면 카트가 위치 정보를 담은 nfc를 읽어 카트의 위치를 파악하고, 앱에게 신호를 주어 사용자에게 계산을 진행 할지에 대한 여부를 묻습니다. 카트는 계산대에 가서 물건 하나하나의 가격을 확인하지 않아도 앱에 기록되어 있는 정보들만으로도 계산이 가능하게 되기 때문에 계산 과정이 보다 간편해지게 됩니다.

- 관리자

1)메인 페이지에서 원하는 업무(카트 연결 상황, 상품보기, 상품추가, 상품삭제, 상품검색, 회원관리)를 선택합니다.

2)카트 연결 상황 : 현재 모든 카트들의 연결 상태를 리스트를 통해 확인할 수 있습니다.

3)상품 보기 : DB에 저장되어있는 모든 상품들의 대한 정보(id, 상품명, 가격, 할인율)를 확인할 수 있습니다.

4)상품 추가 : 추가하고자하는 상품의 정보를 입력하면 DB에 저장할 수 있습니다.

5)상품 삭제 : DB에 저장되어있는 상품을 삭제 버튼을 통해 삭제할 수 있습니다.

6)상품 검색 : 상품명, 종류, 브랜드 별로 검색하여 결과를 확인할 수 있습니다.

7)회원 관리 : 가입한 사용자들에 대한 정보를 확인할 수 있고 탈퇴버튼을 통해 특정 사용자의 계정을 삭제할 수도 있습니다.

3.5. 개발환경 (언어, Tool, 사용시스템 등)

구분			상세내용
S/W 개발환경	서버	OS	Ubuntu
		개발환경(IDE)	AWS EC2(Ubuntu), Eclipse, Apache, PHP, MySQL
		개발도구	PuTTY, FileZilla Client, json_simple.jar
		개발언어	Java, Jsp, Jdbc, SQL
		기타사항	
	모바일	OS	
		개발환경(IDE)	Android Studio
		개발도구	
		개발언어	Java
		기타사항	
H/W 구성장비		디바이스	Arduino UNO 2개 (networkingBoard, drivingBoard), 블루투스 모듈, 모터 드라이버 2개, RFID 모듈
		센서	초음파 센서 3개, DC 모터 4개
		통신	블루투스 통신, RFID 통신, I2C 통신
		개발언어	sketch
		기타사항	
기타		의사소통관리	kakaotalk, slack
		기타사항	

4. 프로그램 설명 (최대한 자세하게 기술)

4.1. 파일 구성

o 서버

- 1) 관리자 웹페이지의 UI를 위한 CSS 외부 스타일 시트(.css)
- 2) Json 형식의 데이터를 위한 json_simple.jar 라이브러리 파일
- 3) 소스코드(.jsp, .html)

(1)DB에 접근하여 수정하는 소스코드들

CartConnectionChange.jsp, DeleteUser.jsp, DeleteProductResult.jsp

insert.jsp, join.jsp

(2) DB 쿼리 결과를 모바일에게 Json형식의 데이터로 전송하는 소스코드들
CartConnectionJson.jsp, Login.jsp, ProductQuery.jsp, NFCQuery.jsp

(3) DB 쿼리 결과를 관리자 페이지에 출력하는 소스코드들
CartConnectionStateList.jsp, list.jsp, ProductDelete.jsp
ProductQueryResponse.jsp, ProductQueryResult.jsp, UserManagement.jsp

(4) 관리자 웹페이지 HTML 소스코드들
index.html(메인 화면), insert.html(상품 추가)

o 모바일

1) 사용자의 편리성을 위한 UI/UX를 다룬 layout, menu 등의 xml 파일

2) 앱의 기능들을 정의한 Activity인 java파일

LoadingActivity : 앱의 시작 화면

LoginActivity : 사용자 인증 과정의 한 부분인 로그인 화면.

로그인 - 사용자는 서버에 저장된 자신의 정보(아이디와 비밀번호)를 입력, 로그인.

로그인 버튼 클릭 - 사용자가 입력한 정보들을 사용하여 웹서버로 회원이 맞는지 확인을 요청한다.

사용자가 회원이 맞으면 회원 정보를 담은 웹페이지 결과를 전송받고 회원 정보를 알려주는 다이얼로그 이후 메인화면(MainActivity)으로 전환된다.

회원가입 버튼 클릭 - 회원가입을 위한 화면(MemberActivity)로 화면 전환.

MemberActivity : 회원가입을 위한 정보 입력 화면.

가입 확인 버튼 클릭 - 입력한 정보들이 웹서버로 전송되고 데이터베이스에 저장됨.

MainActivity : 앱 내의 기능들을(상품 검색, 위시리스트, 가계부, 사용자 정보) 한눈에 볼 수 있는 화면. 기능이름의 버튼을 클릭하면 해당 기능의 화면으로 전환된다.

SearchAdapter : 검색 결과 목록을 만들 커스텀 어댑터를 정의한 클래스이다.

ProductData : 검색 결과목록을 만들 커스텀 어댑터에 사용될 데이터를 정의한 클래스.

SearchActivity : 검색 기능 구현.

사용자가 상품명, 상품종류, 상품 브랜드 중 하나를 골라 검색할 수 있다.

검색 버튼 클릭 - 사용자의 입력값을 사용하여 웹서버로 관련 정보를 요청.

만약 사용자가 검색한 내용이 마트에 있는 상품, 종류, 브랜드 중에 하나라면 서버로부터 관련 정보를 받게 된다. 결과 json데이터를 파싱하고 목록으로 만들어 사용자에게 보여준다.

목록 중 항목 클릭 - 항목에 관한 세부정보를 보여주는 화면(DetailActivity)로 전환.

DetailActivity : 상품에 관한 세부 정보를 보여주는 화면. 해당 상품명으로 서버에 관련 정보를 요청 후 요청 결과인 json데이터를 파싱.

WishAdapter : 위시리스트를 만들 커스텀 어댑터를 정의한 클래스이다.

WishData : 위시리스트를 만들 커스텀 어댑터에 사용될 데이터를 정의한 클래스이다.

WishActivity : 쇼핑할 목록을 미리 적어두는 메모장 기능을 수행하는 화면.

항목 추가 - SQLite를 사용. 데이터베이스에 항목 정보 추가.

항목 삭제 - SQLite를 사용. 데이터베이스에 해당 항목을 찾아 삭제.

BasketAdapter : 쇼핑 목록을 만들 커스텀 어댑터를 정의한 클래스이다.

BasketData : 쇼핑 목록을 만들 커스텀 어댑터에 사용될 데이터를 정의한 클래스이다.

BluetoothActivity : 쇼핑 기능 구현

카트에 담은 상품에 대한 정보를 얻기 위해 아두이노와의 블루투스 통신을 구현.

블루투스를 이용한 데이터 수신 - 상품의 nfc태그 정보를 받으며 이 태그 정보를 사용하여 웹서버에게 관련 정보를 요청하고 결과를 전달받는다.

결과 정보는 json파싱하여 필요한 정보를 가져오며 이를 이용해 사용자의 현재 쇼핑 목록을 만들어 보여주고 상품들의 총액을 계산하여 실시간으로 알려준다.

계산하기 - 사용자가 쇼핑을 마치고 계산을 원할 경우 혹은 사용자가 계산을 위한 장소에 도착할 경우 계산을 도와주는 화면(PaymentActivity)으로 전환된다.

PaymentActivity,PaymentActivity2 : 결제가 이루어지는 모습을 보여주는 가상 결제화면.

DBHelper : SQLite를 사용한 데이터베이스 테이블을 정의한다.

UserInfoActivity : 사용자의 로그인 정보(아이디, 비밀번호)를 이용하여 서버에 해당 사용자 정보를 요청하고 검색 결과인 사용자의 정보(이름, 아이디, 핸드폰 번호, 보유 포인트)를 보여준다.

o 아두이노:

1) networkingBoard.ino

- 모바일과의 블루투스 통신
- RFID 정보 인식
- 주행 보드와의 I2C 통신을 통하여 카트 주행 및 정지 신호 전송, 사용자 인식 정보 요청

2) drivingBoard.ino

- 초음파 센서를 통한 거리 인식
- 인식된 거리에 따른 모터 제어

- 통신 보드와의 I2C 통신을 통하여 카트 주행 및 정지 신호 수신, 사용자 인식 정보 송신

4.2. 함수별 기능

o 서버:

* 소스코드 기준 별 기능에 대하여 적었습니다.

(1)DB에 접근하여 수정하는 소스코드들

CartConnectionChange.jsp : DB에 저장되어 있는 카트의 연결 상태를 변경해 준다.

DeleteUser.jsp : DB에 저장되어있는 회원 정보를 삭제한다.

DeleteProductResult.jsp : DB에 저장되어있는 상품을 삭제한다.

insert.jsp : 상품 추가 기능. 입력한 상품 아이디, 이름, 가격, 브랜드, 위치, 할인을 DB에 저장한다.

join.jsp : 모바일에서 회원가입 한 회원의 정보를 DB에 추가한다.

(2) DB 쿼리 결과를 모바일에게 Json형식의 데이터로 전송하는 소스코드들

CartConnectionJson.jsp : DB에 저장되어 있는 카트의 연결 상태를 모바일에게 Json형식의 데이터로 보내준다.

Login.jsp : 모바일에서 로그인할 때 DB에 저장된 id와 password 인지 확인 후 모바일로 해당 회원의 정보를 Json형식의 데이터로 보내준다.

ProductQuery.jsp : 요청한 해당 상품이 있는지 DB에서 찾아 결과를 Json형식의 데이터로 보내준다.

NFCQuery.jsp : 상품마다 저장된 NFC 태그 값을 받아와 DB에서 해당하는 상품을 찾아 이름과 가격을 Json형식의 데이터로 보내준다.

(3) DB 쿼리 결과를 관리자 페이지에 출력하는 소스코드들

CartConnectionStateList.jsp : 현재 모든 카트의 연결 상태 리스트를 보여준다.

list.jsp : DB에 저장되어있는 모든 상품들 목록을 리스트로 보여준다.

ProductDelete.jsp : DB에 저장되어있는 모든 상품들 목록을 리스트로 보여주고 삭제 버튼을 누르면 DeleteProductResult 파일로 넘어가 상품을 삭제한다.

ProductQueryResponse.jsp : 상품명, 종류, 브랜드 별로 검색한다.

ProductQueryResult.jsp : 검색한 결과를 리스트로 보여준다.

UserManagement.jsp : DB에 저장되어있는 회원들 정보를 리스트로 보여주고 삭제 버튼을 누르면 DeleteUser 파일로 넘어가 회원 정보를 삭제한다.

o 모바일:

* 소스코드 파일에 기재한 BluetoothActivity를 기준으로 내용을 적었습니다.

- (Button) stop : 블루투스와의 통신 종료 버튼.

블루투스 데이터 수신 쓰레드 종료. InputStream, OutputStream, Socket 닫음.

- onDestroy() : 액티비티 종료 시 동작.

블루투스 데이터 수신 쓰레드 종료. InputStream, OutputStream, Socket 닫음.

- checkBluetooth : 사용자의 기기가 블루투스 지원 기기일 경우 블루투스 사용에 동의하는지를 묻는 다이얼로그를 띄운다. 사용자가 블루투스 활성화에 동의하는 경우 페어링 된 기기 목록을 보여주는 창을 띄운다.

- getBondedDevices() : 페어링된 장치 목록 얻어오는 함수

- select : 페어링 된 기기 목록을 보여주는 함수.

페어링된 장치 목록 얻어오는 함수(getBondedDevices())를 호출하여 페어링 된 기기가 있는 경우 해당 기기 목록을 화면에 보여준다. 내부클래스 DownloadJson을 호출하여 페어링 된 목록을 서버의 카트 목록과 비교하여 같은 이름을 가진 기기들만 목록으로 구성하여 보여준다. 연결할 장치를 선택한 경우 선택한 장치와 연결을 시도하는 함수(connectToSelectedDevice)를 호출한다.

- connectToSelectedDevice : 선택한 기기와 블루투스로 연결하는 과정. 원격 블루투스 장치와 통신할 수 있는 소켓을 생성하는 함수인

createRfcommSocketToServiceRecord(uuid)를 호출하여 socket 생성.

소켓이 생성 되면 connect() 함수를 호출함으로써 두기기의 연결을 완료한다.

데이터 송수신을 위한 InputStream과 OutputStream을 얻는다.

sendData()함수를 호출해 아두이노에게 사용자의 쇼핑 준비가 완료됐음을 알린다.

서버에게 선택한 기기의 정보를 전송하여 해당 아두이노(카트)를 사용함을 알린다.

블루투스 통신을 통한 데이터 수신을 위해 beginListenForData 함수를 호출한다.

- sendData : 블루투스 데이터 전송을 담당하는 함수.

- beginListenForData : 블루투스 데이터 수신을 담당하는 함수.

쓰레드를 사용하여 수신된 메시지를 계속 검사한다. 수신되는 데이터가 있을 경우 문자열로 만들고 쇼핑 목록 구성을 위해 shop() 함수를 호출한다.

- shop : 네트워크 통신이 가능할 경우 조합한 url과 함께 내부클래스 (DownloadJsonShop)를 호출한다.
- 내부클래스 DownloadJson : jsoup을 이용하여 서버에 요청한 웹페이지 결과를 받아온다. 받은 결과인 json데이터는 파싱을 통해 필요한 정보를 골라 사용한다.
- 내부클래스 DownloadJsonShop : jsoup을 이용하여 서버에 요청한 웹페이지 결과를 받아온다.

쇼핑한 품목들 리스트화하기 위하여 json파싱으로 상품의 이름과 가격 정보를 가져오며 상품의 가격을 총액에서 더하여 화면에 보여준다. 구성되어 있는 쇼핑 리스트에서 같은 이름의 상품이 있을 경우 (상품을 카트에서 다시 빼낼 경우) 상품의 가격만큼을 총액에서 제한한다.

상품명이 아닌 정보(카트가 결제를 위한 특정 구간에 도착한 경우)를 받으면 쇼핑을 마무리하고 결제를 유도하는 화면으로 전환한다.

o 아두이노:

1) networkingBoard.ino

- setup() : 각종 센서 초기화 및 통신 정보 설정
- loop() : 모바일과의 블루투스 통신, 타 보드와의 I2C 통신, RFID 정보 인식을 지속적으로 반복
- printHex() : 인식한 RFID 정보를 시리얼 모니터에 16진수로 출력
- printDec() : 인식한 RFID 정보를 시리얼 모니터에 10진수로 출력
- printPhoneHex() : 인식한 RFID 정보를 모바일에 16진수로 전송
- userAuthentication() : I2C 통신을 이용하여 주행 보드에 사용자 위치 정보 요청

2) drivingBoard.ino

- setup() : _init() 함수 호출
- loop() : 초음파 센서를 통한 거리 정보를 인식 후 그 정보를 통한 거리 별 카트 방향 제어를 지속적으로 반복
- ultrasonic() : 초음파 센서를 통하여 거리를 cm 단위로 계산
- isInRange() : 파라미터로 받아온 거리 정보를 통하여 사용자의 위치를 구간별로 나누어 리턴
- receiveConnData() : I2C 통신으로 마스터에게 카트 제어 신호가 들어오면 카트 주행 시작 및 중지
- requestConnData() : I2C 통신으로 마스터가 정보를 요청한 경우 초음파 센서로

- 읽어온 거리를 통하여 사용자 인식 정보를 전송
- `_init()` : 각종 센서 초기화 및 통신 정보 설정

4.3. 주요 함수의 흐름도

o 서버:

*주요함수 공통

데이터베이스 연결관련 변수 선언
데이터베이스 연결관련정보를 문자열로 선언
JDBC 드라이버 로드
데이터베이스 연결정보를 이용해 Connection 인스턴스 확보

*로그인 함수 흐름도 (Login.jsp)

sql 쿼리 수행 (select문) => 쿼리 결과가 ResultSet 클래스의 인스턴스로 리턴
=> 마지막 데이터까지 루프를 돌며 파라미터 값으로 넘어온 id와 passwd 비교
=> 파라미터 값과 일치하면 해당 레코드의 정보를 Json형식으로 넘겨줌
=> 루프가 끝날 때 까지 일치하지 않으면 로그인 실패 알림

*상품 검색 함수 흐름도 (ProductQueryResult.jsp)

sql 쿼리 수행 (select문) => 쿼리 결과가 ResultSet 클래스의 인스턴스로 리턴
=> 마지막 데이터까지 루프 수행 => 조건문으로 넘어온 파라미터 값이 상품의 이름인지 브랜드인지 종류인지를 구별(어떤 카테고리로 검색 했는지 구분)
=> 파라미터 값과 일치하면 해당 레코드의 정보를 Json형식으로 넘겨줌
=> 루프가 끝날 때 까지 일치하지 않으면 해당 상품이 존재하지 않습니다 알림

*상품 추가 및 삭제 함수 흐름도 (insert.jsp, DeleteProductResult.jsp, DeleteUser.jsp)

sql 쿼리 수행 (insert문 OR delete문)
=> SQL문에 의하여 DB에 레코드가 변경 되었는지 알기위하여
`pstmt.executeUpdate()` 이용
=> 0보다 큰 수 반환 시 성공적으로 추가 삭제 되었음을 알림

o 모바일:

* 소스코드 파일에 기재한 BluetoothActivity를 기준으로 내용을 적었습니다.

* 4.2 함수별 기능을 바탕으로 적은 내용입니다.

블루투스 활성화(checkBluetooth)

=> 페어링 된 기기 목록 보여줌

(select --- getBondedDevices()호출, 내부클래스 DownloadJson호출)

=> 원하는 기기 선택 후 연결

(connectToSelectedDevice --- createRfcommSocketToServiceRecord(uuid)호출,
sendData()호출)

=> 블루투스 데이터 수신 (beginListenForData)

=> 수신 데이터로 쇼핑 목록 구성 (shop, 내부클래스 DownloadJsonShop호출)

=> 쇼핑 종료 및 결제 유도 (onDestroy())

o 아두이노:

각 보드별 setup 및 초기화

=> 모바일과 통신용 보드(networkingBoard)의 블루투스 연결,

주행 시작 요청 신호 수신

=> 통신용 보드에서 I2C 통신으로 주행용 보드에게 사용자 인식 정보 요청

(networkingBoard.userAuthentication())

=> 주행용 보드(drivingBoard)는 초음파 센서를 통해 사용자와의 거리 측정 후

통신용 보드로 인식 정보 송신

(drivingBoard.ultrasonic() --- drivingBoard.isInRange())

--- drivingBoard.requestConnData())

=> 인식 성공 시 I2C 통신으로 주행용 보드에게 주행 시작 신호 전송

(drivingBoard.receiveConnData())

=> 주행용 보드에서 초음파 센서로부터 읽어온 거리 정보를 이용하여 모터 제어

(drivingBoard.ultrasonic() --- drivingBoard.isInRange())

=> 쇼핑 중 RFID 정보 인식 및 모바일로 Hex 코드 전송

(networkingBoard.printPhoneHex())

=> 모바일과 통신용 보드 블루투스 연결 해제

=> 통신용 보드에서 I2C 통신으로 주행용 보드에게 주행 정지 신호 전송

(drivingBoard.receiveConnData())

4.4. 기술적 차별성

스마트 카트는 하드웨어 구현에만 그친 것이 아니라 모바일 애플리케이션과 관리자 화면(웹 페이지)과 같은 소프트웨어적인 측면까지 갖췄습니다. 모바일은 앱을 통하여 사용자에게 마트 내의 상품의 정보를 제공하여 자신이 원하는 상품이 현재 마트에 있는지 확인할 수 있고, 종류로 상품을 검색 하였을 시에는 브랜드별, 상품별 가격 비교를 통해 합리적인 구매를 도와줍니다. 또한 메모장을 들고 다닐 필요 없이 앱의 위시리스트 기능을 이용하여 자신이 구매할 목록을 만들 수 있으며, 카트에 담기는 상품의 총액을 그때그때 마다 계산하여 보여주기 때문에 과소비를 막을 수 있어 쇼핑을 보다 현명하게 할 수 있도록 도와줍니다. 관리자 웹 페이지는 데이터베이스에 직접적으로 접근하지 않아도 상품, 사용자 정보 등을 손쉽게 추가, 삭제할 수 있게 하였습니다. 따라서 데이터베이스를 다루지 못하는 직원이라도 편리하게 웹 페이지의 버튼을 클릭 하는 것만으로도 마트 운영에 필요한 정보들을 좀 더 편리하게 관리할 수 있게 해줍니다.

하드웨어 측면에서도 아두이노를 주행과 통신 기능을 두 개의 보드에 분리함으로써 더욱 효율적으로 기기를 제어할 수 있도록 하였습니다.

구성적 측면에서 아두이노, 모바일, 서버가 각자의 역할을 수행하며 서로 통신하기 때문에 데이터 관리 및 송수신에 용이하며 하드웨어만의 기능도 충실히 수행할 수 있습니다.

5. 응용 분야

스마트 카트는 기존 사용 목적에 따라 마트에서의 사용에 대단히 유용할 것이라고 생각합니다. 무거운 짐들과 함께 이동하며 쇼핑해야 하는 사용자들에게 무게감 없이 가벼운 두 손으로 쇼핑을 할 수 있도록 도와준다는 강점이 있기 때문입니다. 이런 스마트 카트의 특징은 아이와 함께 쇼핑하는 주부들이나 거동이 불편한 장애인분들 또한 쇼핑을 보다 더 쉽게 만들어줍니다. 또한 함께 사용하게 되는 애플리케이션은 간단한 검색으로 정보를 제공 받을 수 있는 점, 자동으로 구성되는 쇼핑리스트를 통해 쇼핑상황을 실시간으로 확인이 가능한 점 등 사용자가 스마트 카트와 함께하는 쇼핑을 더욱 편리하게 만들어줍니다. 또한 서버의 관리자 페이지는 관리자에게 웹페이지를 통한 몇 번의 클릭만으로도 간편하게 데이터들을 관리할 수 있게 하고 마트의 상황을 빠르게 파악할 수 있게 하여 마트 운영에 도움을 줍니다.

스마트 카트는 이렇게 하드웨어인 카트와 소프트웨어인 애플리케이션, 서버를 한 번에 사용할 수 있게 되면서 사용자는 물론 관리자까지 도와줍니다. 때문에 스마트 카트가 이용되고 응용될 수 있는 곳은 다양합니다.

만약 우체국, 택배회사와 같은 물품을 관리하는 곳들에 스마트 카트가 사용된다면 카트가 담고 있는 물건들을 애플리케이션으로 바로 확인 할 수 있기 때문에 일일이

상품의 배송 지를 확인할 필요가 없으며 관리자들은 배송지의 정보, 배송 품 정보, 배송차량의 현재 정보까지 동시에 확인이 가능해 사용자들의 업무를 편리하게 해줍니다.

병원에서도 스마트 카트는 유용하게 사용될 수 있습니다. 큰 병원일수록 관리하는 의약품들과 수많은 환자 정보들이 있고 그것들을 관리하는 간호사분들은 항상 간이 카트를 끌고 다닙니다. 필요한 물품을 챙기다보면 빠트린 것은 없는지 일일이 확인해야 하는 경우가 발생하기도 합니다. 이 때 스마트 카트를 사용하게 되면 카트와 통신하는 애플리케이션으로 간단하게 정보를 확인할 수 있습니다. 또한 설 틈 없이 움직여야만 하는 간호사분들이 카트를 챙기고 다니지 않아도 알아서 따라 다니기 때문에 신경을 덜 쓰고 체력을 낭비하지 않게 해줍니다. 물품 관리에 있어서도 관리자 웹페이지가 있다면 어디에 어떤 물품이 어떤 용도로 사용되는지와 같은 정보나 혹시나 사라진 물품은 없는지와 같은 정보들을 한눈에 확인이 가능합니다.

이렇듯 스마트카트는 물품 관리와 사용자의 편의성이 동시에 필요한 곳에 유용하게 사용될 수 있으며 앞으로 더 많은 곳에 응용되어 사용될 수 있을 것으로 생각합니다. 또한 이러한 점 때문에 스마트 카트를 사용함으로써 사용처의 업무들은 더욱 체계적인 관리 시스템으로 만들어질 것이라고 생각합니다.

6. 향후 계획

스마트 카트는 현재 사용자를 따라다니는 기능만을 가지고 있습니다. 하지만 저희 팀은 더 나아가 카트가 사용자에게 상품의 위치를 안내하게 하도록 하는 '자율 주행 시스템'이라는 기능을 추가하려고 합니다. 사용자가 구매하고자 하는 상품을 선택하면 카트는 사용자를 따라다니는 주행 모드에서 자율 주행 모드로 전환되어 DB에 저장되어 있는 상품의 위치 정보를 받아 해당 위치로 움직일 수 있도록 하는 것입니다.

또한 지금의 스마트 카트는 쇼핑 후에 특정 위치를 지나면 결제를 할 것 인지 묻는 알림이 뜹니다. 즉, 가상으로 결제가 이루어지는 모습만을 연출하고 있습니다. 이에 그치지 않고 저희 팀은 향후에 결제 승인 시스템을 연동하여 결제를 원하는 버튼을 클릭하는 것만으로도 결제가 이루어지게 할 것입니다.

앞으로 스마트 카트가 정말로 쇼핑에 이용되고 이러한 기능들이 추가 된다면 쇼핑이 좀 더 편하고 수월해질 것이라고 생각합니다. 때문에 지속적인 개발로 스마트 카트를 더 발전 시켜 사람들에게 편리함을 제공해줄 수 있는 기회가 생겼으면 좋겠습니다.

7. 제작자 정보

No.	구분	성명	소속(학교)	부서(학과)	입학년도	담당업무
1	팀장	위혜진	동덕여자대학교	컴퓨터학과	2015년	앱 제작, 업무 분담 및 일정 조율
2	팀원	양혜임	동덕여자대학교	컴퓨터학과	2015년	아두이노 프로그래밍, 카트 제작 및 제어
3	팀원	이아룬	동덕여자대학교	컴퓨터학과	2015년	서버 구축 및 웹 페이지 제작
4	팀원	이지호	동덕여자대학교	컴퓨터학과	2015년	데이터베이스 관리 및 서버 구축

8. 개발 단계별 기간 및 투입 인원

- 투입 인원: 총 4명
- 전 팀원 기획 및 설계 참여 1개월 소요
- 매주 수요일 오후 2시 개발 상황 공유 및 업무 분담 회의
- 개발/구현 및 성능점검/테스트 3개월 소요
- 총 4개월 소요

구분	개발내용	수행일정							
		6월		7월		8월		9월	
		상	하	상	하	상	하	상	하
기획/설계	아이디어 회의								
	아두이노 세부 기획 및 설계								
	usecase작성								
개발 /구현	아두이노-모바일 블루투스 통신								
	가상 서버 구축								
	아두이노 부품 조사 및 테스트								
	모바일-서버 통신. 데이터 파싱								
	아두이노를 탑재한 카트제작								
	아두이노 모터 제어(속도.방향)								
	서버 관리자 화면 구축								
	모바일 기능 보완 및 UI/UX								
테스트 /성능점검	최적화 및 최종버전 테스트								
	최종 보고서 작성								