

QR코드를 이용한 택배 안심배송 서비스

2021.10.22

중부대학교 정보보호학과
지도교수 : 이병천 교수님



1 조

권현주
신세영
유현진

목차

- 조원 편성
- 주제 선정
- 구상도
- 추진 경과
- 개발 환경 및 개발 내용
- 개발 시스템 운영
- 결론 및 기대 효과

조원 편성

이름	역할
권현주(조장)	DB 구축 및 관리, 백엔드 설계 및 구현
신세영	프론트·백엔드 설계 및 구현
유현진	DB 구축 및 관리, 프론트·백엔드 설계 및 구현, 보고서 작성

주제 선정

224만 유튜버, **택배 운송장 무심코 버렸다가...**

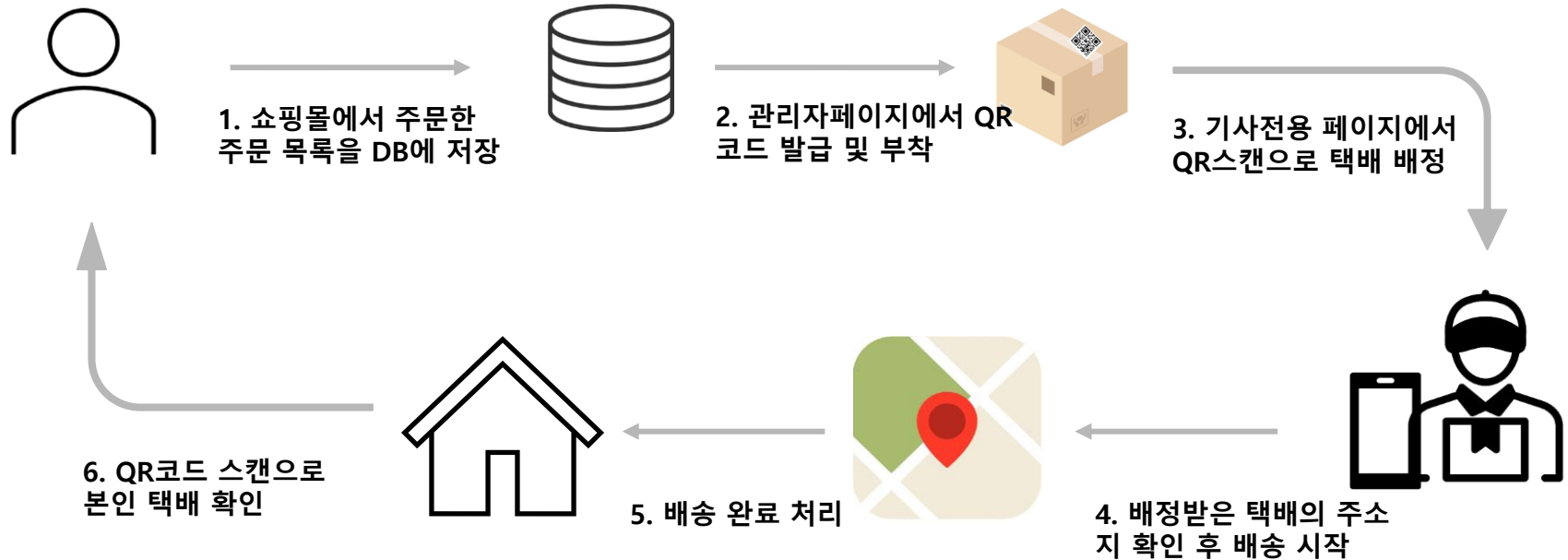
택배 운송장 연락처 보고 음란문자 보낸 70대 징역 6월

“눈 가리고 아웅”... CJ대한통운, 택배 개인정보 유출 ‘나 몰라라’

택배 운송장에 보낸사람받는사람 이름, 주소, 휴대폰번호 그대로 노출
스미싱 피해 무방비 노출... CJ측 “택배기사가 제거해야” 책임 떠넘기기

고객의 개인 정보 유출 방지를 위한 QR코드 안심배송 서비스 개발 추진

구상도



추진 경과

	3월	4월	5월	6월	7월	8월	9월	10월
자료조사 및 연구	■							
프론트엔드 개발		■						
백엔드 개발			■					
QR코드 생성, 인식/지도 시스템			■					
시스템 검증 및 보완						■		

개발 환경 및 개발 내용 (1/9)

개발 환경

Web

Angular
Ionic
Node.js



DB &

Server

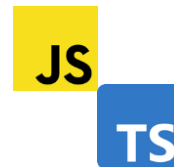
FireBase



Development

Language

Javascript
Typescript



개발 환경 및 개발 내용 (2/9)

개발 내용(1/8)

```
import { QRCodeModule } from 'angularx-qrcode';
```

home > orders > 71d6N18Ritnwmf...		
capstone2-178ba	orders	71d6N18Ritnwmf6SXu2h
+ 컬렉션 시작	+ 문서 추가	+ 컬렉션 시작
Delivery	71d6N18Ritnwmf6SXu2h	+ 필드 추가
carts	Gt1njRcgXrLiG18b8sxcg	address: "경기 고양시 덕양구 대자동 613-1"
orders	HViAYI0D1PjyZorM2Prf	d_number: "41281_227"
products	0dqDxJMr1WdXiJdjPk4	deliverystart: 2021년 10월 9일 오전 12시 41분 15초 UTC+9
profile	P6wB6egC5v712N36Pk9k	deliverysuccess: 2021년 10월 7일 오후 3시 12분 30초 UTC+9
	R3mcYPtIN8B17wTPRurr	detailaddress: "세종관"
	Raj1KosUU5HH7LmNjvAp	dname: "김당근"
	XXdav0oZU7mDZNAG2s01	dstate: "배송시작"
	ZfwHJ2DMvROJA1HeG5Dv	hHXB7BxHsSEm6Qk0vjN: 4

npm install angularx-qrcode
명령어로 module 설치

NPM(Node Package Manager) :
Node.js 기본 패키지 관리자

```
<tr *ngFor="let p of orders | async" class="{{p.dstate}}">  
<qrcode [qrdata]="p.id" [width]="256" [errorCorrectionLevel]="M"></qrcode>
```


개발 환경 및 개발 내용 (3/9)

개발 내용(2/8)

```
async startScan() {
  // Not working on iOS standalone mode!
  const stream = await navigator.mediaDevices.getUserMedia({
    video: { facingMode: 'environment' }
  });
  this.videoElement.srcObject = stream;
  // Required for Safari
  this.videoElement.setAttribute('playsinline', true);
  this.loading = await this.loadingCtrl.create({});
  await this.loading.present();
  this.videoElement.play();
  requestAnimationFrame(this.scan.bind(this));
}
```

카메라 권한을 허용하여 스캔

```
async scan() {
  if (this.videoElement.readyState === this.videoElement.HAVE_ENOUGH_DATA) {
    if (this.loading) {
      await this.loading.dismiss();
      this.loading = null;
      this.scanActive = true;
    }
    this.canvasElement.height = this.videoElement.videoHeight;
    this.canvasElement.width = this.videoElement.videoWidth;
    this.canvasContext.drawImage(
      this.videoElement,
      0,
      0,
      this.canvasElement.width,
      this.canvasElement.height
    );
    const imageData = this.canvasContext.getImageData(
      0,
      0,
      this.canvasElement.width,
      this.canvasElement.height
    );
    const code = jsQR(imageData.data, imageData.width, imageData.height, {
      inversionAttempts: 'dontInvert'
    });
    if (code) {
      this.scanActive = false;
      this.scanResult = code.data;
    } else {
      if (this.scanActive) {
        requestAnimationFrame(this.scan.bind(this));
      }
    }
  } else {
    requestAnimationFrame(this.scan.bind(this));
  }
}
```

개발 환경 및 개발 내용 (4/9)

개발 내용(3/8)

스캔값과 DB `profile/(계정의uid) / cart(주문목록)`의 내용과 일치해야 데이터(이름, 주소, 전화번호)를 표시한다.

```
getOrders(){
  this.orderCollection= this.afs.collection(`profile/${this.getUserUid()}/cart`);
  return this.orderCollection.valueChanges({idField: 'id'});
}
```

```
<div *ngFor="let p of u_orders | async">
  <ion-card *ngIf="p.id == scanResult">
    <ion-card-header>
      <ion-card-title>배송내역 확인</ion-card-title>
    </ion-card-header>
    <ion-card-content>
      받는분 성함 : {{p.name}} <br>
      받는분 주소 : {{p.address}} {{p.detailaddress}}
      배송상태 : {{p.dstate}}
      <ion-button (click)="showQrToast()">환불요청
    </ion-card-content>
  </div>
```

스캔 완료 후 표시되는 내용

profile/ (계정의uid) / cart(주문목록)

The screenshot shows a mobile application interface with a profile page. The profile page has a header with the name "928VHywUSx0Qj5xcYeyqPVic112" and a list of items. The "cart" section is highlighted with a red box, showing a list of items with their IDs and details.

profile	928VHywUSx0Qj5xcYeyqPVic112	cart
+ 문서 추가	+ 컬렉션 시작	+ 문서 추가
7SdG4doq6ZQu7nUaMM5xefC6zs2	cart	71d6N18Ritnmwf6SXu2h
81C4ezGCKmaCV0AHCYos50H1QxX2		Gt1njRcgXrLiG18b8sxsG
928VHywUSx0Qj5xcYeyqPVic112		HV1AYI0D1PjyZorM2Prf
BCf0xMzpG7N08N2Df6cgTSZh2t1		R3mcYPtIN8B17wTPRurr
X0K6maLWPgPBytrfz1fFM9CieQW2	+ 필드 추가	Raj1KosUU5HH7LmNjvAp
	address: "경기 고양시 덕양구 대자동 613-1"	WZSMbgsWmtE0RvGhh0ec
	detailaddress: "세종관"	XXdav0oZU7mDZAG2s01
	email: "aa@naver.com"	ZfwHJ2DMvR0JA1HeG5Dv
	name: "김춘식"	
	phonenumber: "012-3456-7890"	
	si: "41281"	

개발 환경 및 개발 내용 (5/9)

개발 내용(4/8)

전체 주문 목록에서 스캔한 값과 비교하여 같으면 기사에게 배정

```
<div *ngFor="let p of u_orders | async">
<ion-card *ngIf="p.id == scanResult">
  <h4>택배 코드 : {{p.d_number}} <ion-button class="add" (click)="showQrToast()">추가</ion-button></h4>
```

```
async showQrToast() {
const toast = await this.toastCtrl.create({
message: `배송지를 추가하시겠습니까?`,
position: 'top',
color: 'dark',
duration: 5000,
buttons: [
{
text: '저장',
handler: () => {
this.addD_orders();
```

```
addD_orders(){
const uid = this.authService.getUserUid();
this.D_items = this.itemService getCompareItems(this.profil
```

```
getCompareItems(profileName, scanResult, uid){
var db = this.afs;

this.ordersNotAssign.get().toPromise()
.then(function(querySnapshot){
querySnapshot.forEach(function(doc) {

if(doc.id == scanResult){
DdocRef.update({{
dname : profileName, dstate : "배송시작",
deliverystart : Timestamp.now(),lastUpdate:Timestamp.now()}})
```

배정된 시간, 배송상태, 기사
의 이름을 추가

개발 환경 및 개발 내용 (6/9)

개발 내용(5/8)

```
this.uid = this.authService.getUserUid()
this.afs.collection('Delivery').doc(this.uid).get().subscribe({
  next: (doc) => {
    const key = Object.keys(doc.data())
    console.log("key : ", key)
    if( key.includes('privatekey')!= true){
      this.forgeservice.deliveryCert();
    }
  }
})
```

deliveryCert() [1]

```
var forge = require('node-forge');

//키쌍 생성
let keyPair = forge.pki.rsa.generateKeyPair(2048);
var publicKey = keyPair.publicKey;
var privateKey = keyPair.privateKey;

var publicKeyPem = forge.pki.publicKeyToPem(publicKey);
var privateKeyPem = forge.pki.privateKeyToPem(privateKey);

var caCert = forge.pki.createCertificate();
caCert.publicKey = publicKey;
// alternatively set public key from a csr
//cert.publicKey = csr.publicKey;
```

```
caCert.setSubject(this.caAttrs, "utf-8");
caCert.setIssuer(this.caAttrs);
caCert.setExtensions([ // 인증기관의 인증
  {
    name: 'basicConstraints',
    cA: true
  }, {
    name: 'keyUsage',
    keyCertSign: true,
    digitalSignature: true,
    nonRepudiation: true,
    keyEncipherment: true,
    dataEncipherment: true
  }, {
    name: 'extKeyUsage',
    serverAuth: true,
    clientAuth: true,
    codeSigning: true,
    emailProtection: true,
    timeStamping: true
  }, {
    name: 'nsCertType',
    client: true,
    server: true,
    email: true,
    objsign: true,
    sslCA: true,
    emailCA: true,
    objCA: true
  }, {

```

```
//self-siign certificate
caCert.sign(privateKey);
```

```
localStorage.setItem('privatekey', privateKeyPem)
localStorage.setItem('cert', caCertPem)
```

기사의 인증서 발급 여부 확인 후 인증서 발급

개발 환경 및 개발 내용 (7/9)

개발 내용(6/8)

Kakao map에서 제공하는
API를 기반으로 제작

```
for (let i = 0; i < length; i++) {
  setTimeout(() => {
    console.log();
    var name = arr1[i].name;
    var phone = arr1[i].phonenumber;
    var address = arr1[i].address;
    var d_number = arr1[i].d_number;
    console.log(name);
    geocoder.addressSearch(address, function(result, status) {
      if (status === kakao.maps.services.Status.OK) {
        var addr = result[0].road_name;
        var coords = new kakao.maps.LatLng(result[0].y, result[0].x);
        var contents =
          '<div class="customover' +
          '<div class="title" sty' +
          d_number +
          '</div>' +
          '</div>';
        search_id.onclick = () => {
          var findID = this.findID;
          var search_result = arr1.find(value => value.d_number === findID);
          geocoder.addressSearch(search_result.address, function(result, status) {
            if (status === kakao.maps.services.Status.OK) {
              var moveLatLon = new kakao.maps.LatLng(result[0].y, result[0].x);
              map.setCenter(moveLatLon);
            }
          });
          this.renderMap(arr1, map, geocoder);
        };
      }
    });
  });
}
```

개발 환경 및 개발 내용 (8/9)

개발 내용(7/8)

배송완료시 배송상태 수정

dstate: "배송완료"

```
complete(item){
  console.log("complete")
  if (item){
    this.itemService.D_ordersCollection.doc(item.Document_ID).update({dstate : "배송완료", deliverysuccess : Times
    this.itemService.ordersCollection.doc(item.Document_ID).update({dstate : "배송완료", deliverysuccess : Times

    this.afs.collectionGroup('cart', ref => ref.where('dstate', 'eq', '배송완료')).doc(item.Document_ID).update({dstate : "배송완료", deliverysuccess : Times
    e => { e.docs.map(
      doc => { if (doc.id == item.Document_ID) {
        var path = doc.ref.path
        //
        this.forgeservice.doc Cert(item.Document ID )
        this.afs.doc(path).update({ dstate : "배송완료", deliverysuccess : Times
      }
    })
  }

  this.showToast();
  if (item.dstate == "배송완료"){
    this.CheckToast();
  }
}

var md = forge.md.sha1.create();
md.update(this.text, 'utf8');

const privateKeyPem = localStorage.getItem('privatekey')
const privateKey = forge.pki.privateKeyFromPem(privateKeyPem)

var signature = privateKey.sign(md)
console.log('Signature: ' + signature)
complete- "5ea49751d05eadb27f197554050aa5450a009cbfec92585ec7
signature:

this.afs.collection('Delivery').doc(this.id)
.collection('D_orders').doc(Document_ID)
.update({'complete-signature' : forge.util.bytesToHex(signature)})
```

기사정보, 운송장 번호,
배송현황, 완료한 시간
에 대한 서명값을 생성
하여 저장

개발 환경 및 개발 내용 (9/9)

개발 내용(8/8)

배정취소시 배송상태 수정

```
delete(item){
  console.log("delete")
  this.itemService.D_ordersCollection.doc(item.Document_ID).update({dname : "미지정" , dstate : "배송준비", lastUpdate : Timestamp.now()})
  this.itemService.ordersCollection.doc(item.Document_ID).update({dname : "미지정" , dstate : "배송준비", lastUpdate : Timestamp.now()})
  this.afs.collectionGroup('cart', ref => ref.where('dname', '!=', "미지정" )).get().forEach(
    e => { e.docs.map(
      doc => { if (doc.id == item.Document_ID) {
        var path = doc.ref.path
        this.afs.doc(path).update({ dname : "미지정", dstate : "배송준비", lastUpdate : Timestamp.now() }) }
      })
    })
  return this.itemService.D_ordersCollection.doc(item.Document_ID).delete()
}
```

개발 시스템 운영 (1/12)

고객 페이지 - 로그인, 회원가입

초기화면

고민은
배송만 늦출


for Bakery



고객 여러분께 편리하고 간편한
주문 시스템과 개인 정보보호를 위한
QR운송장을 제공해드립니다.

2021 택배왔조

시작하기

로그인

Login

Email

Password

로그인

회원가입

회원가입

회원가입

이름

김겨울

이메일

winter@season.com

비밀번호

.....

회원가입

개발 시스템 운영 (2/12)

고객 페이지 - 프로필, 기본 배송지

기본 배송지 설정x

김겨울 고객님

winter@season.com

131-1111-1111

기본 배송지 수정하기

기본 배송지 수정

주소
경기 고양시 덕양구 대자동 613-1

41281

상세주소
세종관 8층

전화번호
131-1111-

중부대학교

도로명 전체 지역명 전체

38144 영문보기 | 지도
도로명 경북 경주시 봉황로 65 (계명대학교 경주동산분원)

지번 경북 경주시 서부동 147-2 (행정동: 중부동)

10279 영문보기 | 지도
도로명 경기 고양시 덕양구 동천로 305 (중부대학교 고양캠퍼스)

지번 경기 고양시 덕양구 대자동 613-1

완료

다음 주소 API를 이용하여 주소 검색

기본 배송지 설정 완료

김겨울 고객님

winter@season.com

131-1111-1111

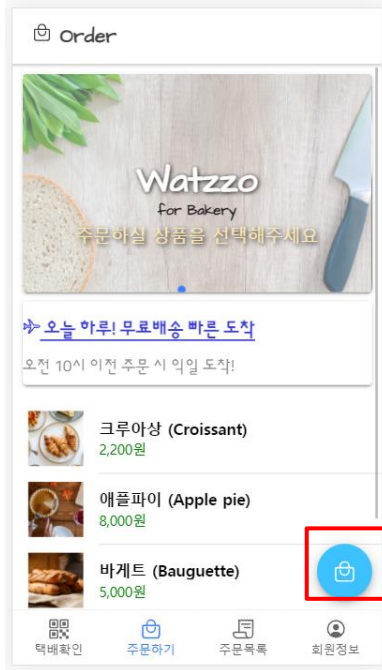
경기 고양시 덕양구 대자동 613-1
세종관 8층 831호

기본 배송지 수정하기

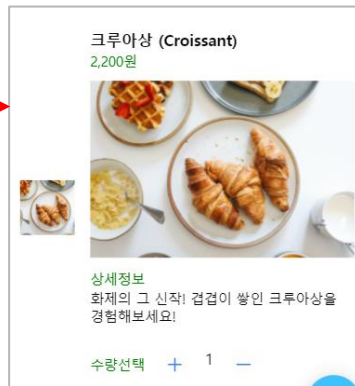
DB에 기본배송지 저장

개발 시스템 운영 (3/12)

고객 페이지 - 물품 주문 (1/2)



클릭 시 상세정보
열람 및 수량 선택



기본 배송지가 등록된 경
우 기본 배송지로 설정



개발 시스템 운영 (4/12)

고객 페이지 - 물품 주문 (2/2) . 주문 목록

클릭 시 상세내용 확인 가능

Order Confirmation

주문목록 확인

- 크루아상 (Croissant) x 1
- 애플파이 (Apple pie) x 2
- 바게트 (Baguette) x 1
- 식빵 (Loaf bread) x 1


배송지 확인

이름: 김겨울

주 : 경기 고양시 덕양구 대자동 613-소 1

상세주소: 세종관 8층 831호

전화번호: 131-1111-1111



Order

오늘 하루 무료배송 빠른 도착
오전 10시 이전 주문 시 익일 도착!

크루아상 (Croissant)
2,200원

Success
주문이 완료되었습니다.

확인

상세정보
회제의 그 신작! 겹겹이 쌓인 크루아상을 경험해보세요!

수량선택 + 0 -

애플파이 (Apple pie)
2,200원

택배확인 주문하기 주문목록 회원정보

Order List

List

배송 시작

11140_242 2021/10/18 1:54 AM

배송 완료

41287_241 2021/10/18 1:54 AM

41281_240 2021/10/18 1:45 AM

택배확인 주문하기 주문목록 회원정보

41281_240 2021/10/18 1:45 AM

주소 : 경기 고양시 덕양구 대자동 613-1 세종관 8층 831호

받는분 성함 : 김겨울

받는분 전화번호 : 131-1111-1111

기사님 성함 : 미지정

배송 상태 : 배송준비
2021/10/18 5:55 PM

주문 목록

- 크루아상 (Croissant) : 1
- 애플파이 (Apple pie) : 2
- 바게트 (Baguette) : 1
- 식빵 (Loaf Bread) : 1

환불요청

개발 시스템 운영 (5/12)

관리자 페이지 - 로그인, 메인 화면, 주문 목록

로그인

메인 화면

주문 목록

Wazzo Admin page
주문정보 확인

이름	전화번호	주소	주문목록	운송장	배정 기사	배송 상태	배송 현황
김준식	012-3456-7890	경기 고양시 덕양구 대자동 613-1 세종관	크루아상 (Croissant) : 2 애플파이 (Apple pie) : 바게트 (Bauguette) : 식빵 (Loaf Bread) :	QR운송장	홍길동	배송완료	주문완료 (배송준비) : 2021/10/17 11:23 PM 배송시작 : 2021/10/18 9:22 PM 배송완료 : 2021/10/18 9:42 PM
김산타	010-111-111	경기 고양시 덕양구 대자동 613-1 세종관 8층	크루아상 (Croissant) : 1 애플파이 (Apple pie) :	QR운송장	미지정	주문완료	주문완료 (배송준비) : 2021/10/19 12:36 AM

개발 시스템 운영 (6/12)

관리자 페이지 - QR 운송장

Watzzo Admin page
주문정보 확인

이름	전화번호	주소	주문목록	운송장	배정 기사	배송 상태	배송 현황
김준식	012-3456-7890	경기 고양시 덕양구 대자동 613-1 세종관	크루아상 (Croissant) : 2 애플파이 (Apple pie) : 바게트 (Baguette) : 식빵 (Loaf Bread) :	QR운송장		배송완료	주문완료 (배송준비) : 2021/10/17 11:23 PM 배송시작 : 2021/10/18 6:03 PM 배송완료 : 2021/10/18 6:03 PM

운송장인쇄

window.print() 함수 실행
으로 인쇄 가능



개발 시스템 운영 (7/12)

관리자 페이지 - 기사 계정 생성

기사 정보

기사 계정 생성

계정 이메일

jb@univ.com

비밀번호

.....

6자리 이상의 비밀번호를 입력해주세요.

회원가입

기사 정보 입력

이름

이정원

전화번호

091-111-1111

완료

기사 계정 생성 시 table에 표시

계정 만들기

이름

전화번호

홍길동

010-1234-5678

홍반장

0102345-6789

김당근

010-2323-0000

test1@naver.com

테스트

010-2222-3333

testtest@naver.com

이름

전화번호

기사계정

이정원

091-111-1111

jb@univ.com

홍길동

010-1234-5678

honggildong@naver.com

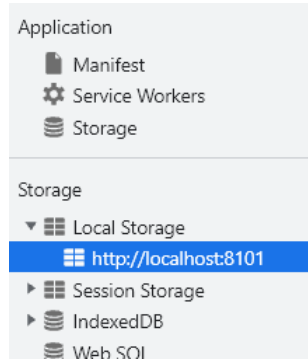
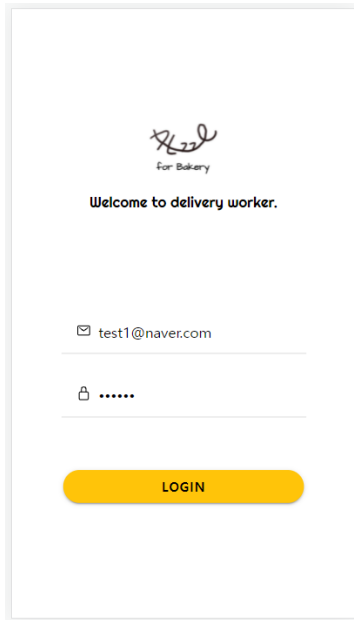
홍반장

01023456097

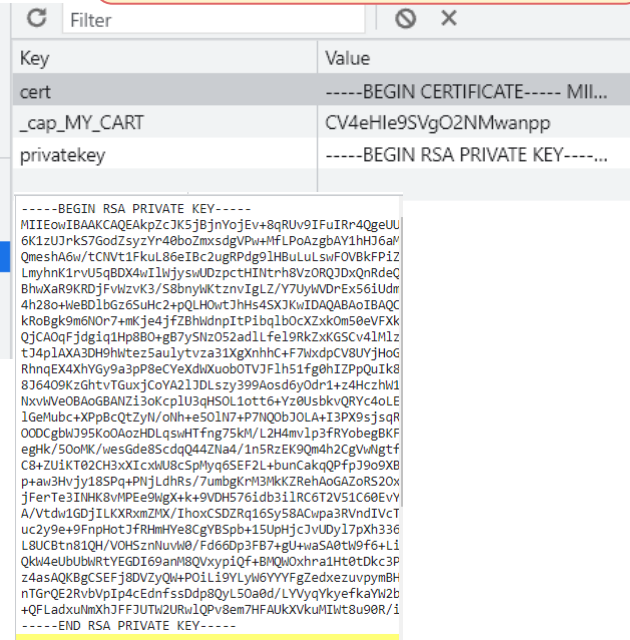
sy@naver.com

개발 시스템 운영 (8/12)

기사 페이지 - 로그인



기사의 개인키와 공개키 저장



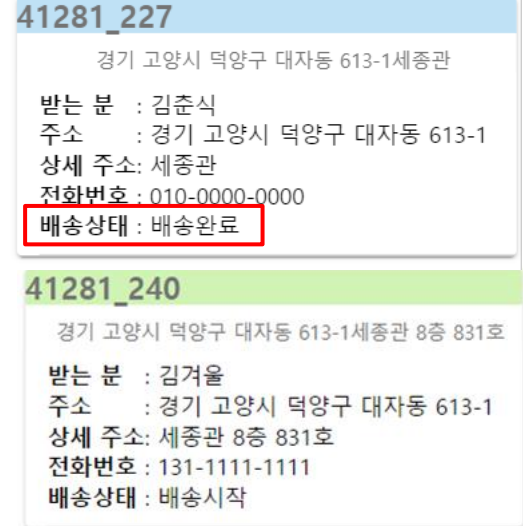
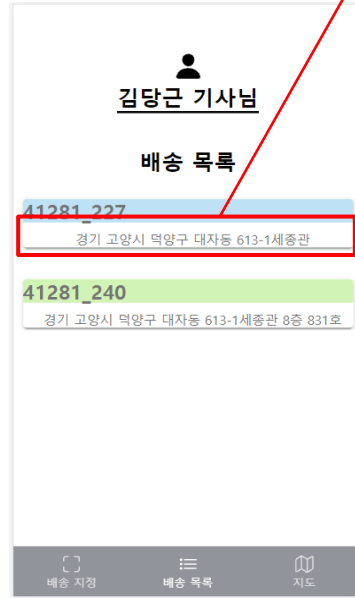
개발 시스템 운영 (9/12)

클릭 시 상세 내용 확인 가능
배송상태에 따라 색을 다르게 표시

기사 - 배송 배정, 배송 목록

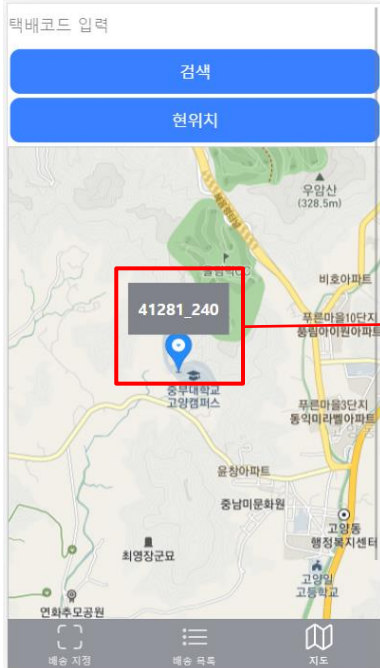


추가 버튼 클릭 시
배송 목록에 표시

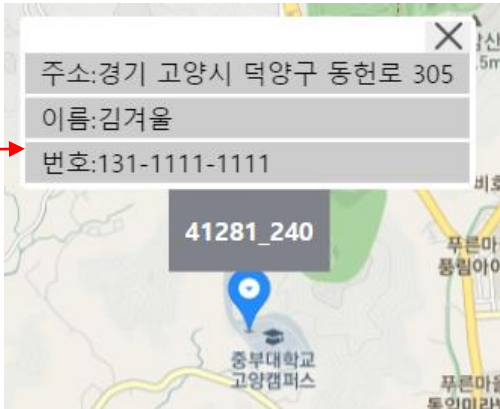


개발 시스템 운영 (10/12)

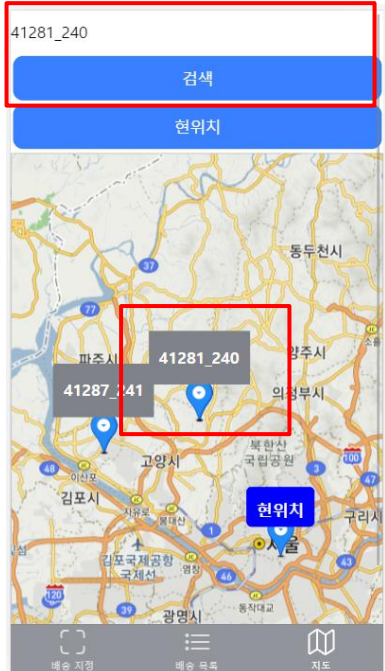
기사 페이지 - 배송지 확인



마커 클릭 시 상세 주소 확인 가능

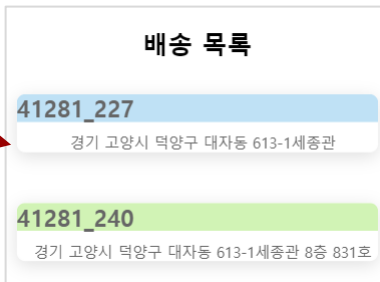
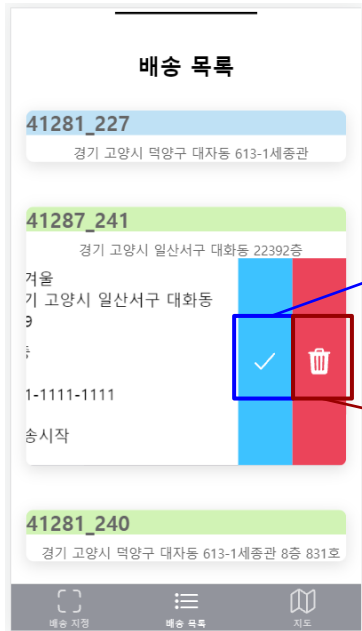


검색 클릭 시 해당 주소지가 중심부로 표시



개발 시스템 운영 (11/12)

기사 페이지 - 배송 상태 변경

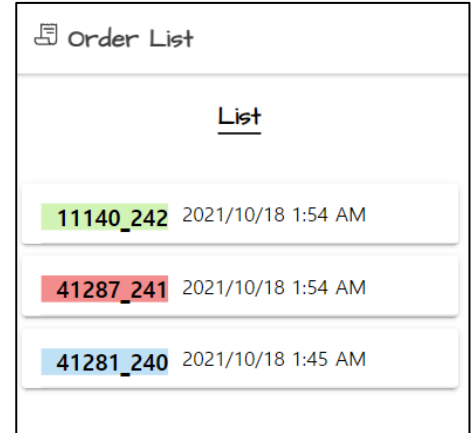


배송완료 시 체크 표시를 클릭하여 DB 배송상태를 배송완료로 변경, 서명값을 생성하여 DB에 저장

배송배정 스캔을 잘못 진행한 경우 배정 취소

개발 시스템 운영 (12/12)

고객 페이지 - 배송 확인, 환불



결론 및 기대 효과

□ 결론

- 기존의 운송장인 아닌 QR코드를 사용하여 일차적으로 쉽게 노출되는 부분을 보완하여 개인 정보보호를 강화하는데 성공
- 전자서명을 이용하여 고객-기사의 부인방지를 보장한다.

□ 기대효과

- 개발 과정 중 어려운 부분은 기술 공유 및 협동으로 팀워크를 향상시키고 모든 조원들이 개발 과정에 참여하며 시스템 기획 및 프로그램 개발 역량 향상
- 2021.08.22 개인정보위에서 택배 운송장 개인 정보보호 강화 대책을 논의했다. 이에 개인 정보보호에 대한 중요성을 다시 한번 인식. 끝.

감사합니다.