OTP를 이용한 스마트 무인 택배보관함

(Delivery Box with One Time Password)

도진우(14051012), 최지훈(17019110)
jinubb@naver.com, souline95@naver.com

지도교수 : 홍기천

Ⅰ. 아이디어 배경

가. 대면 배송의 문제점

1. 소규모(1~2인) 가구의 증가
   - 배송 시간에 집을 비우는 가구가 증가로 대면 배송이 어려워지고 있다.
2. 전염성 바이러스 전파
   - 신종 바이러스의 유행으로 바이러스 전파의 위험성이 커지고 있다.

나. 기존 무인 택배 보관함의 문제점

1. 비밀번호 유출
   - 택배기사가 비밀번호를 지정할 경우 유추하기 쉬운 비밀번호(1234) 혹은 고정 비밀번호로 지정하는 경우가 발생하고 있어 택배 사고의 위험이 있다.
2. 낮은 편의성
   - 택배기사와 주민 모두에게 친숙하지 않은 사용방법이다.
   - 실시간 택배 추적이 불가능하다.

Ⅱ. 작품 설명

가. 기능

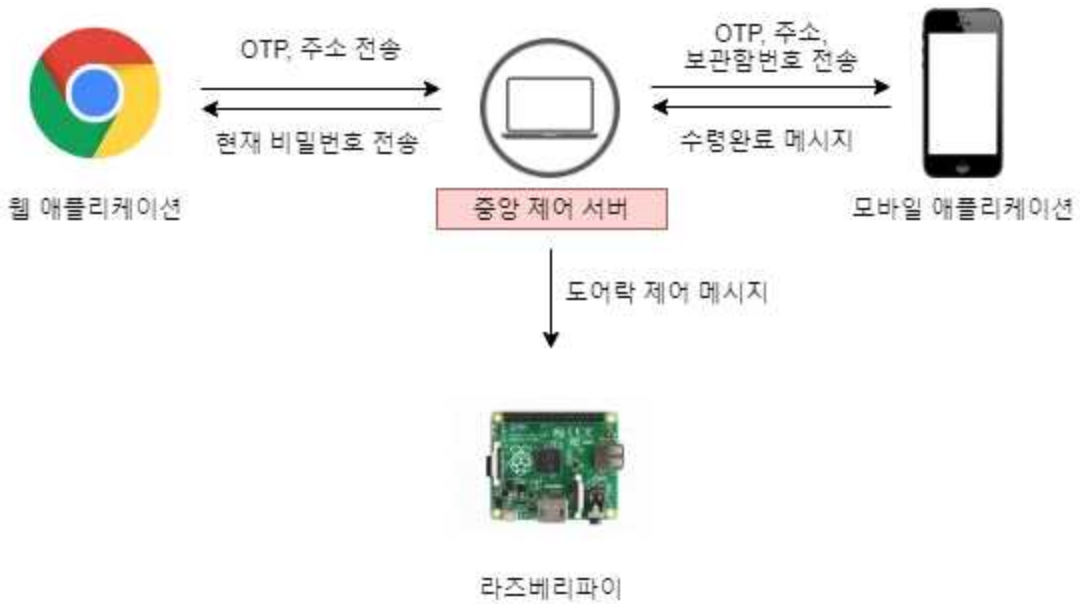
1. 택배 보관 시 비어있는 택배 보관함 번호를 유동적으로 할당한다.
2. OTP(One Time Password)를 생성하여 자동으로 택배 보관함의 비밀번호를 변경한다.
3. 택배 보관 시 수령인의 모바일 애플리케이션에 알림이 울린다.
4. 택배 보관 시 생성된 비밀번호는 수령인의 모바일 애플리케이션을 통해 확인할 수 있다.
5. 터치스크린의 가상 키패드로 비밀번호를 입력받아서 비밀번호가 일치할 경우 보관함의 잠금을 해제한다.

나. 용도

1. 택배기사와 수령인의 비대면 택배배송이 가능하다.
2. 택배 도착 상태를 언제 어디에서나 확인할 수 있다.

Ⅲ. 작품 특징 및 독창성

가. Node.js 통합 제어 서버



1. 소켓통신
   - 웹 애플리케이션과 서버 간의 양방향 소켓 통신
   - 생성된 OTP를 중앙 제어 서버로 전송
2. 유동적인 보관함 할당
   - 보관함의 상태변수를 서버에 저장하여 유동적인 보관함 할당
3. 메시지 전송
   - 이벤트가 발생하였을 경우 다른 클라이언트에게 메시지 전송

나. MQTT 통신

1. MQTT 서버
   - Eclipse mosquitto server

- <Topic, Message>를 <Key, Value>로 구독(Subscribe) 혹은 발행(Publish)
- TCP/IP를 기반으로 low byte의 message를 전달하는 통신 프로토콜

2. 클라이언트

ⅰ. 통합 제어 서버

- 소켓통신을 통해 웹으로부터 새로운 택배 보관 정보를 받는다.
- 생성된 OTP를 비어있는 사물함에 할당시킨다.
- OTP와 할당된 사물함 정보, 택배 수령자의 정보를 JSON으로 파싱하여 MQTT서버로 발행한다.

ⅱ. 모바일 앱

- 통합 제어 서버에서 받은 데이터를 구독한다.
- Callback를 통해 해당하는 택배 수령자의 device에 출력한다.
- 택배를 수령한 후 수령한 택배의 번호를 MQTT서버로 발행한다.

ⅲ. 라즈베리파이

- 통합 제어 서버에서 받은 데이터와 모바일 앱에서 받은 데이터를 구독한다.
- Callback을 통해 택배 보관함의 비밀번호와 상태를 변경한다.

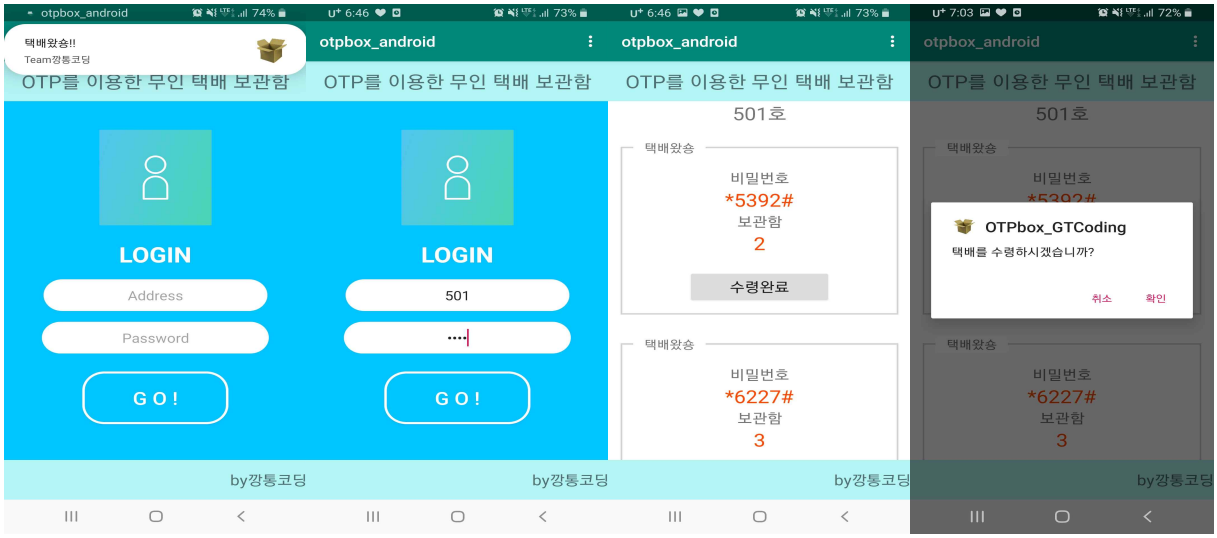
Ⅳ. 작품 설계

가. 작품구성

1. 웹 애플리케이션

- 배송주소를 입력받아서 OTP 생성
- 비밀번호를 입력받아서 일치할 경우 잠금 해제
- 모든 박스에 할당된 경우 택배 보관 disable

2. 모바일 애플리케이션



- 해당 사용자에게 택배가 보관된 경우 push notification
- 로그인 시 해당 사용자의 택배 도착 정보를 확인
- OTP를 입력하여 택배 수령
- 택배 수령완료 시 택배보관함 상태 변경

3. 택배 보관함

- 서보모터와 걸쇠를 부착하여 도어락 기능 구현
- 홀 센서와 자석을 부착하여 자동 잠금 기능 구현

Ⅴ. 작품 구현

가. 터치스크린

- 터치를 통해 웹 애플리케이션 가상키패드로 배송 주소를 입력
- 생성된 OTP를 일정 시간 확인
- 비밀번호를 입력받아 보관함의 잠금을 해제

나. 스피커

- 사용자 인터페이스를 위해서 이벤트 발생 시 이벤트에 따라서 음성 안내메시지가 출력

다. 전원

- 5V 2.1A 배터리를 사용

Ⅵ. 결론 및 기대 효과

가. 결론

1. Server-side 방식의 기능구현
   - H/W를 줄이고 핵심 로직을 서버에서 수행
   - 펌웨어를 업그레이드하기 용이한 높은 확장성
   - 전체 택배 보관함의 서버 관제
   - 사용자의 VOC를 통한 인터페이스 개선이 용이

2. OTP
   - 택배 보관 시 자동으로 OTP를 생성하여 보안성 증가

3. 비대면 방식의 택배 수령
   - 택배 기사와 택배 수령자가 비대면으로 분실 우려 없이 택배 수령 가능

나. 기대효과

1. 직접 비밀번호를 지정하는 번거로움이 없음
2. 택배 보관마다 무작위로 변경되는 비밀번호로 높은 보안성
3. 자신의 택배 상황을 모바일 앱을 통해 언제 어디에서나 확인
4. 비대면 방식으로 전염성 전파 예방

Ⅶ. 후기

처음 1개월 정도는 아이디어를 상세하게 구상하고 시작한 프로젝트였습니다. 아이디어 구상을 짧게 하고 프로젝트를 시작하면 개발 중에 다양한 수정사항이 생기게 되고 또 그 수정사항이 프로젝트 전반에 타격을 주었습니다. 아이디어를 이해한 뒤 머릿속으로 그림을 그리고 아이디어를 구체화하는 것의 중요성을 알 게 되었습니다.

프로젝트의 목표는 '최대한 시제품에 가깝게 제작하기'입니다. 단순히 기능 구현이 아닌 실제로 제품을 사용하는 소비자가 직접 사용한다고 가정하고 개발했습니다. 음성 안내메시지를 출력하는 사용자 인터페이스나 혹시 오류가 날 경우 해결하는 페이지(제어 웹페이지)도 제작하였습니다.

어려웠던 점은 숙련되지 않은 언어들을 사용해야 한다는 점이었습니다. 웹을 개발하는 HTML, javascript는 구조만 알고 있었고 실제로 프로그램 코드를 작성해본 적이 없었습니다. UI를 그리면서 기능을 구현하고 보니 라즈베리파이 7인치 터치스크린의 해상도와 맞지 않는 사이즈였고 해상도를 맞추려다 보니 예상한 기간보다 더 오래 걸렸습니다.

프로젝트를 진행하면서 '저게 될까?' 싶은 문제도 해결해보고 아이디어 구상하면서 '우리가 할 수 있을까?' 하는 내용도 구현에 성공했습니다. 어려운 문제를 간단하게

해결한 적도 있으며 쉬운 문제를 며칠 동안 고민한 기억도 있습니다. 이러한 경험을 통해서 개발자로써 한층 성장한 느낌이며 개발자를 진로로 더욱 노력하게 되는 계기가 되었습니다.

Ⅷ. 참고 문헌

[1] Python client Eclipse 'paho' Library Document
https://www.eclipse.org/paho/index.php?page=clients/python/docs/index.php
[2] Node.js npm 'paho-mqtt' package Document
https://www.npmjs.com/package/paho-mqtt
[3] Android client Eclipse 'MQTT' Library Document
https://www.eclipse.org/paho/index.php?page=clients/android/index.php
[4] Python 'pigpio' Library Github
https://github.com/fivdi/pigpio
[5] Node.js Document
https://nodejs.org/ko/docs/
[6] Node.js npm 'soket.io' package Document
https://socket.io/docs/