
포트폴리오

목차

보안 채팅 프로그램

프로젝트 개요

- 신뢰성 있는 사용자 간의 채팅 서비스 제공.
- 안전한 채팅 데이터의 송/수신 기능 제공.

프로젝트 개발 배경

- 인터넷 기술의 발달에 따른 다양한 메신저 해킹 위협을 해결하기 위해 고민.

보안 채팅 프로그램

프로젝트 주요 기능

멀티 채팅 룸 기능

- 다수의 이용자가 "Vector" 기반의 채팅 룸을 생성할 수 있고 채팅 룸 안에서의 "채팅 서비스"를 이용 할 수 있음.
- 채팅 룸의 인원이 0명이면 해당 채팅 룸은 자동으로 삭제.

신뢰성 있는 사용자 간의 채팅 기능

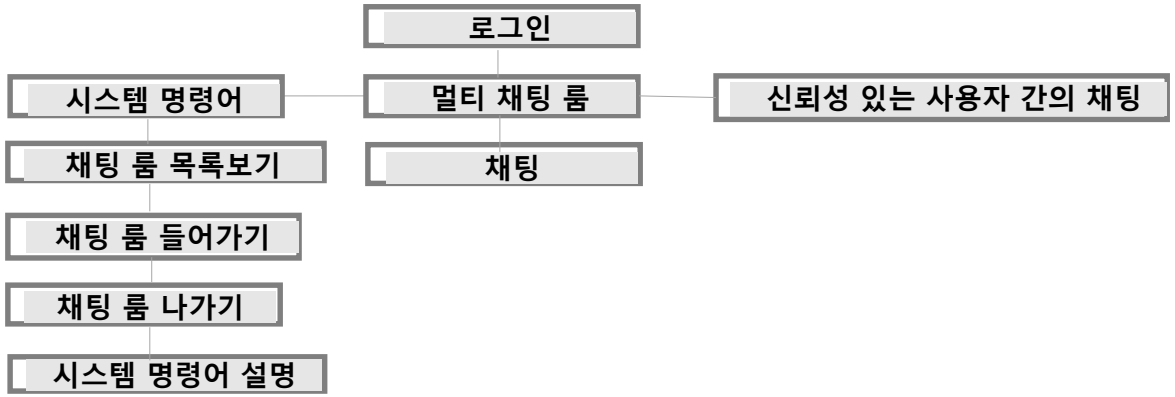
- 채팅 데이터를 "SSLEngine"의 "Wrapping"으로 암호화하고 "Unwrapping"으로 복호화하여 안전한 채팅을 할 수 있도록 함.
- "SSL 인증서"를 통한 통신방법으로 "신뢰성 있는 사용자들 간의 통신"을 지원함.
- "SocketChannel과 Selector"를 활용한 "NIO Channel"을 통하여 "스레드(클라이언트) 수의 증가"로 인한 "성능 이슈" 해결.

시스템 명령어 기능

- 채팅 서비스에 필요한 다양한 명령어 기능을 제공.

시스템 구성 및 개발 환경

시스템 구성



개발 환경

Front-End

- Swing

Back-End

- JAVA
- SSLEngine
- SocketChannel
- Selector

보안 채팅 프로그램

본인의 역할

- “SSLEngine”과 “NIO Channel”기반의 채팅 프로그램 백엔드 개발
- “Vector” 기반의 멀티 채팅 룸 구조와 “HashMap” 기반의 유저 정보 구조 구현

개인 개발 핵심 코드

MessengerServer.java (서버 구동 부분)

```
public static void main(String args[]) {
        try {
                MessengerServer server = new MessengerServer("TLS","localhost",8700);
                server.startServer();
        } catch (Exception e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
        }
}
```

SSL/TLS 방식의 서버 구동 시작

```
public class MessengerServer extends MessengerBasic  {

public boolean isActiveSvr;
public SSLContext context;
public Selector selector;
MessengerRoomUserInfo roomUserInfo;

public MessengerServer(String protocol, String hostAddress, int portNumber) throws Exception {
    roomUserInfo = new MessengerRoomUserInfo();
        context=SSLContext.getInstance(protocol);
        context.init(createKeyManagers(".WWbinWW.keystoreWWSSLSocketServerKeyWW","123456","123456"),
        createTrustManagers(".WWbinWW.keystoreWWSSLSocketServerKeyWW","123456"), new SecureRandom());

        SSLSession session = context.createSSLEngine().getSession();
        AppData=ByteBuffer.allocate(session.getApplicationBufferSize());
        NetData=ByteBuffer.allocate(session.getPacketBufferSize());
        NodeAppData=ByteBuffer.allocate(session.getApplicationBufferSize());
        NodeNetData=ByteBuffer.allocate(session.getPacketBufferSize());
        session.invalidate();

        selector=SelectorProvider.provider().openSelector();
        ServerSocketChannel channel = ServerSocketChannel.open();
        channel.configureBlocking(false);
        channel.socket().bind(new InetSocketAddress(portNumber));
        channel.register(selector, SelectionKey.OP_ACCEPT);

        isActiveSvr = true;
}
```

SSL 세션 생성

서버에서 보내고 받을 Buffer 초기화

NIO Channel 생성

보안 채팅 프로그램

MessengerServerReceiver.java (서버 측 데이터 수신 부분)

```
synchronized protected void recv(SocketChannel channel,SSLEngine engine) throws Exception {

        NodeNetData.clear();
        int waitToRecvMillis=50;
        try {
int byterecv = channel.read(NodeNetData);   → 채널을 통해 암호화된 데이터 수신

        if(byterecv > 0) {
                NodeNetData.flip();
                while(NodeNetData.hasRemaining() && !isClosed) {
                        NodeAppData.clear();
 result = engine.unwrap(NodeNetData, NodeAppData);   → Unwrapping으로 데이터 복호화 처리
                        switch(result.getStatus()) {
                        case OK:
                                NodeAppData.flip();

                                Charset charset = Charset.defaultCharset();
 String message = charset.decode(NodeAppData).toString();   → 복호화된 데이터를 문자열로 변환
```

MessengerServerSender.java (서버 측 데이터 송신 부분)

```
synchronized public void send(SocketChannel channel,SSLEngine engine,String m)throws IOException{

        AppData.clear();
        AppData.put(m.getBytes());
        AppData.flip();

        while(AppData.hasRemaining()) {
                NetData.clear();
 SSLEngineResult result =engine.wrap(AppData,NetData);
                switch(result.getStatus()) {
                case OK:                          ↓ Wrapping으로 데이터 암호화 처리
                NetData.flip();

                while(NetData.hasRemaining()) {
                        channel.write(NetData);
                }                                 ↓
                break;          채널을 통해 암호화된 데이터 송신
```

보안 채팅 프로그램

MessengerRoomUserInfo.java (채팅 룸 정보 구조, 사용자 정보 구조 부분)

```
import javax.net.ssl.SSLEngine;

class UserInfo{
        public String id;
        public SSLEngine engine;
        public SelectionKey key;
        public String roomname;
}
```

채팅 사용자 정보

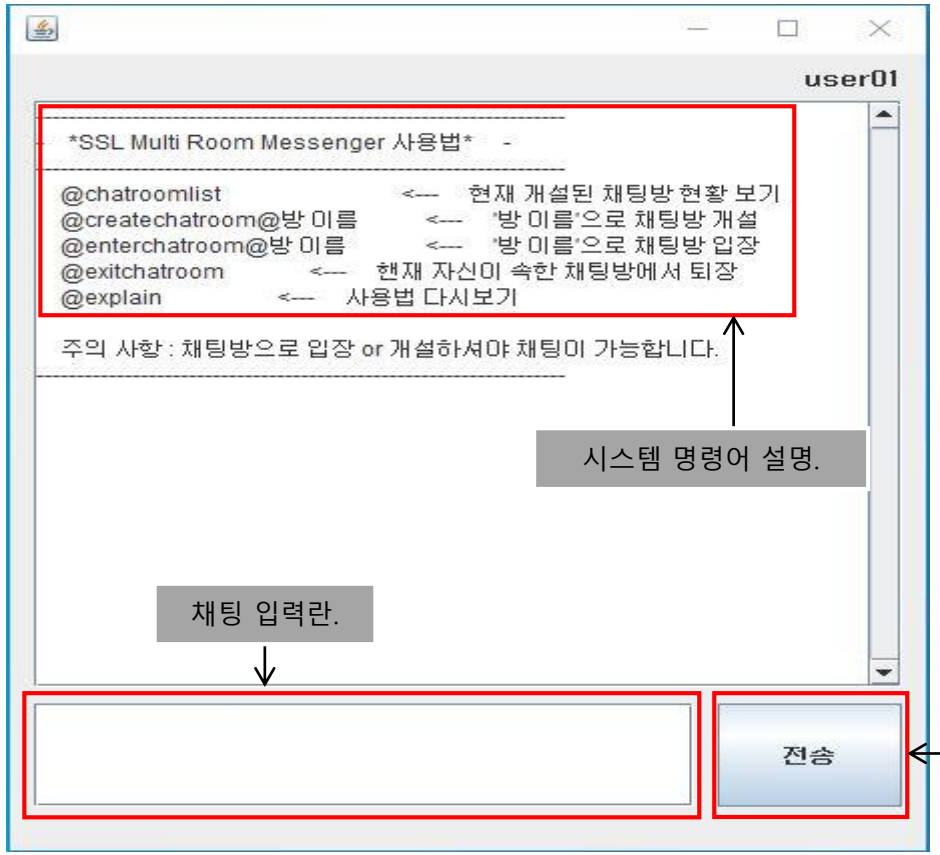
```
public class MessengerRoomUserInfo {
        public Vector<String> room;
        public Vector<UserInfo> info;
        public HashMap<String,Vector<UserInfo>> map = new HashMap<String,Vector<UserInfo>>();

        public MessengerRoomUserInfo(){
                room = new Vector<String>();
                info = new Vector<UserInfo>();
                map  = new HashMap<String,Vector<UserInfo>>();
        }
}
```

채팅 룸 정보

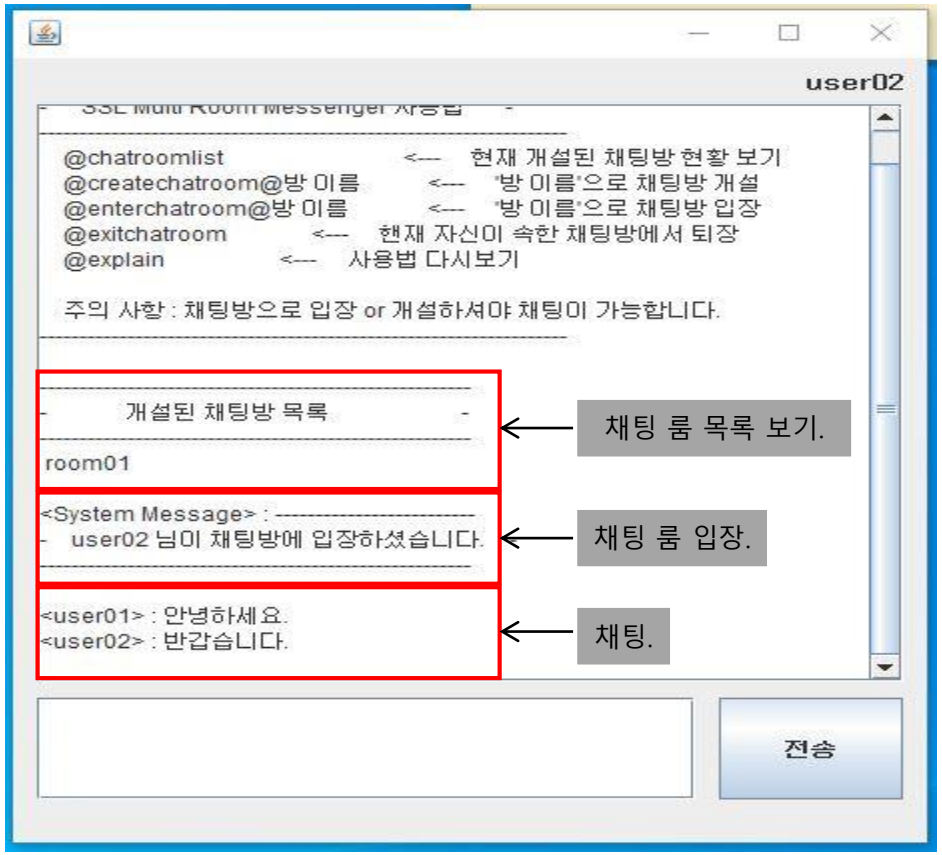
보안 채팅 프로그램

프로젝트 핵심 결과물

채팅 프로그램(메인 화면 부분)



채팅 프로그램(채팅 부분)



프로젝트 개발 기간 및 기여도

기여도 : 55 %

세부추진내용	과제수행기간							진도율(%)
	1주	2주	3주	4주	5주	6주	7주	
기획문서작성	■	■						20 %
시제품 개발			■	■	■			60 %
디버깅					■	■		80 %
프로젝트 개선						■		90 %
발표							■	100 %

보안 채팅 프로그램

프로젝트 기대효과

- **“신뢰성 있는 사용자들간의 채팅”을 통해 해킹 위협으로부터 “안전성” 제공.**
- **채팅 데이터의 “암호화”와 “복호화”를 통해 데이터 유출 위협으로부터 “보안성” 제공.**

프로젝트 후 느낀 점

- **“JAVA” 기반의 네트워크 프로그래밍 능력 향상.**
- **“SSL/TLS” 기반의 보안 서버 구현 능력 배양.**
- **정보 관리를 위한 “자료구조” 활용 능력 향상.**

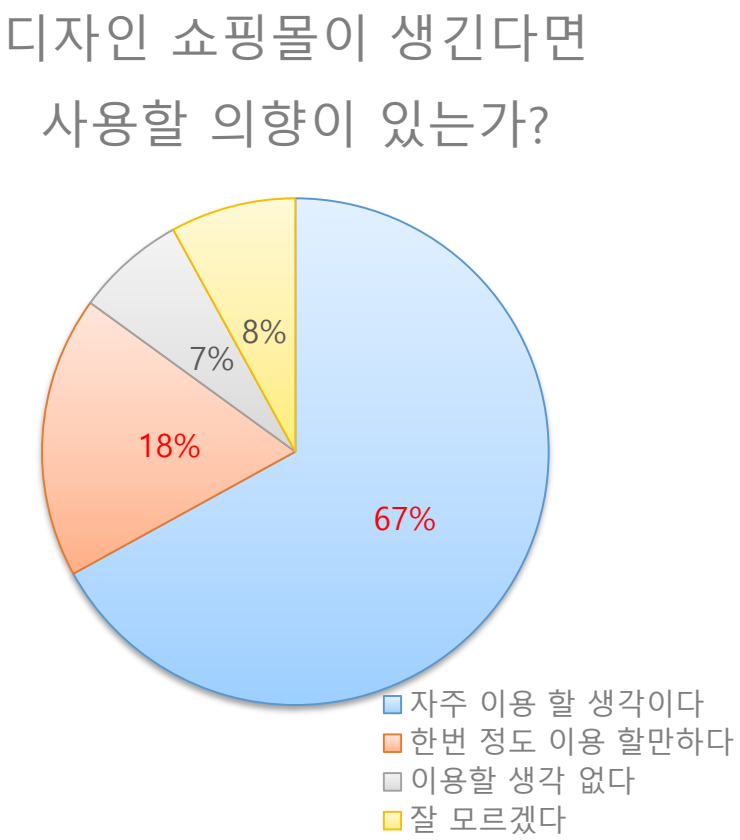
디자인 쇼핑몰

프로젝트 개요

- 웹 기반의 의류 디자인 제작 기능 제공.
- 의류 디자인 구매/판매가 가능한 쇼핑몰 서비스 제공.

프로젝트 개발 배경

- 의류 디자인 전자상거래의 진흥을 위해 고민.
- 초보 디자이너들의 실습 환경 부족문제를 해결하기 위해 고민.



패션 디자인학과 학생 35명을 대상으로 한 앙케트 조사 자료

프로젝트 주요 기능

의류 디자인 제작 기능

- "마우스 펜"과 "텍스트 입력"을 활용한 "의류 디자인" 제작 가능.
- 완성된 "의류 디자인"을 "이미지"로 저장 할 수 있음.

디자인 쇼핑몰 기능

- 디자이너들이 완성된 "의류 디자인"을 "상품 등록"하여 판매할 수 있도록 함.
- 고객은 "의류 디자인 "을 구매하여 다운로드 할 수 있음.

자유 게시판 기능

- 디자인에 대한 다양한 의견을 교류 할 수 있는 게시판.

시스템 구성 및 개발 환경

시스템 구성

개발 환경

Front-End

- HTML5
- JavaScript
- J-Query
- I-Frame
- Tiles

Back-End

- Spring Framework (MVC 2)
- Ajax
- Oracle DBMS

디자인 쇼핑몰

본인의 역할

- “Spring Framework”를 활용한 “디자인 쇼핑몰 기능”의 “구매”, ”상품”, “장바구니” 백엔드 개발

개인 개발 핵심 코드

ShopController.java (구매 부분)

```
@RequestMapping(value = "/product_purchase", method = RequestMethod.POST)
public String product_purchase(@ModelAttribute ("Purchase") Purchase phs,HttpSession session,int pro_num, HttpServletRequest request) {

    Member m =  (Member)session.getAttribute("member");   ← 세션에서 구매 회원 정보 획득
    int mem_num  = m.getMem_num();
    String receiver=request.getParameter("receiver");
    String post_num=request.getParameter("post_num1")+"-"+request.getParameter("post_num2");
    String addr=request.getParameter("addr");
    String phone_num=request.getParameter("phone_num1")+"-"+request.getParameter("phone_num2")+"-"+request.getParameter("phone_num3");
    int des_num=pdao.des_select(request.getParameter("pro_num"));
    int minus_amount=Integer.parseInt(request.getParameter("p_amount"));
    SimpleDateFormat dayTime = new SimpleDateFormat("yyyy-MM-dd HH:mm");
    String time = dayTime.format(new Date(System.currentTimeMillis()));
    String status="입금대기";

    phs.setMem_num(mem_num);
    phs.setP_address(addr);                 ← 구매 정보 등록
    phs.setP_date(time);
    phs.setP_receiver(receiver);
    phs.setP_post_num(post_num);
    phs.setDes_num(des_num);
    phs.setP_phone_num(phone_num);
    phs.setP_status(status);

    phsdao.Insert_Purchase(phs);

    product p =new product();
    p.setMinus_amount(minus_amount);        ← 구매 후 상품 수량 변경
    p.setPro_num(pro_num);
    pdao.Update_amount(p);
    return"redirect:/spring/customer_purchase_list?category_num=6&pageNum=1";
```

디자인 쇼핑몰

ShopController.java (상품 부분)

```
@RequestMapping(value = "/product_regist", method = RequestMethod.POST)
public String product_regist(@ModelAttribute ("product") product p,int pageNum,int category_num,HttpSession session,
Model model ,HttpServletRequest request) throws IOException {

    String file_path = "C:/workspace/INNO/WebContent/shop_item/";
    String file_path = "/workspace-sts-3.4.0.RELEASE/INNO/WebContent/shop_item/";
    String tmp_file_path=session.getServletContext().getRealPath("/");

    SimpleDateFormat dayTime = new SimpleDateFormat("yyyy-MM-dd HH:mm");
    String time = dayTime.format(new Date(System.currentTimeMillis()));

    Member m = new Member();
    m = (Member)session.getAttribute("member");
```

세션에서 쇼핑몰 판매 회원 정보 획득

```
    MultipartFile img_stream_1 =p.getImg_stream_1();

    if(img_stream_1.getOriginalFilename().equals("")){
        return"redirect:/spring/product_regist?category_num="+category_num+"&pageNum="+pageNum;
    }else{
```

의류 디자인 이미지를 저장소에 저장

```
        p.setPro_date(time);
        Integer des_num=ddao.select_des_num(m.getMem_num());

        String ext1=img_stream_1.getOriginalFilename().substring(img_stream_1.getOriginalFilename().lastIndexOf(".")+1);
        String img_name_1=des_num+"_"+time2+"_1."+ext1;

        p.setPro_img(img_name_1);

        byte[] fileData1 = img_stream_1.getBytes();

        FileOutputStream output1 = new FileOutputStream(file_path+img_name_1);
        FileOutputStream output1_tmp = new FileOutputStream(tmp_file_path+"/shop_item/"+img_name_1);

        output1.write(fileData1);
        output1.close();

        output1_tmp.write(fileData1);
        output1_tmp.close();
```

디자인 상품 정보 등록

```
        p.setDes_num(des_num);
        pdao.Insert_Product(p);
        model.addAttribute("product",p);
```

ShopController.java (장바구니 부분)

```
@RequestMapping(value = "/cart_regist", method = RequestMethod.GET)
public String cart_regist(@ModelAttribute ("Purchase") Purchase phs,HttpSession session,
int pro_num,int category_num, HttpServletRequest request) {

        Member m =  (Member)session.getAttribute("member");
        int mem_num = m.getMem_num();
        System.out.println(mem_num);
        Cart cart =new Cart();
        cart.setMem_num(mem_num);
        cart.setPro_num(pro_num);

        cartdao.Insert_Cart(cart);

    return "redirect:/spring/cart_list?pageNum=1&category_num=5";
}
```

세션에서 구매 회원의 정보 획득

장바구니에 디자인 상품 담기

프로젝트 핵심 결과물

디자인 쇼핑몰(디자인 쇼핑몰 메인 화면)

프로젝트 개발 기간 및 기여도

기여도 : 30 %

세부추진내용	과제수행기간									진도율(%)
	1주	2주	3주	4주	5주	6주	7주	8주	9주	
기획문서작성	■	■								10 %
DB설계		■	■	■						30 %
시제품 개발			■	■	■	■				60 %
디버깅					■	■	■			80 %
프로젝트 개선						■	■	■	■	90 %
발표									■	100 %

프로젝트 기대효과

- **“의류 디자인 제작 기능”을 통하여 디자이너들의 “실습 환경” 제공.**
- **“디자인 쇼핑몰”기능을 통하여 “의류 디자인 전자상거래”의 진흥 기대.**

프로젝트 후 느낀 점

- **“디자인 쇼핑몰” 구현 경험을 통해 전자상거래의 “비즈니스 로직”과 “구현 기술” 배양.**
- **“Spring Framework”를 활용한 “MVC 2 모델” 구현 방식을 습득.**

Home Security and Care System

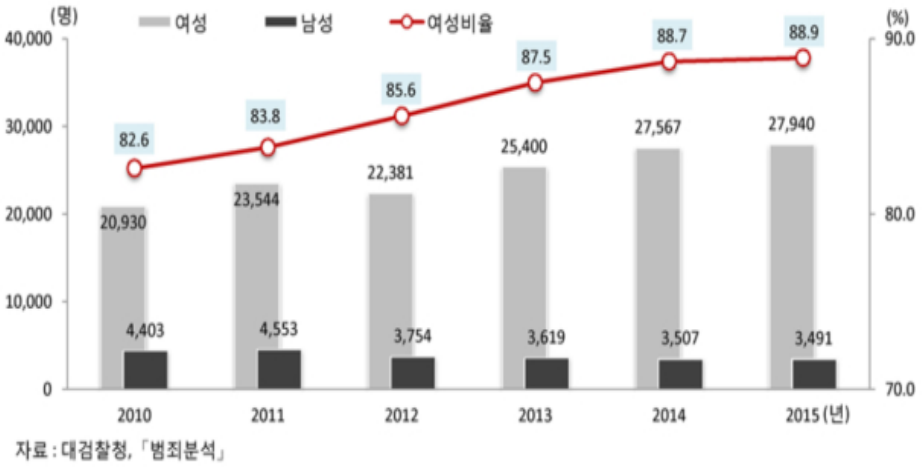
프로젝트 개요

- 주거 침입 예방 및 스마트 홈 서비스 제공.
- 자녀, 여성 고객의 안전 귀가를 도울 수 있는 서비스 제공.

프로젝트 개발 배경

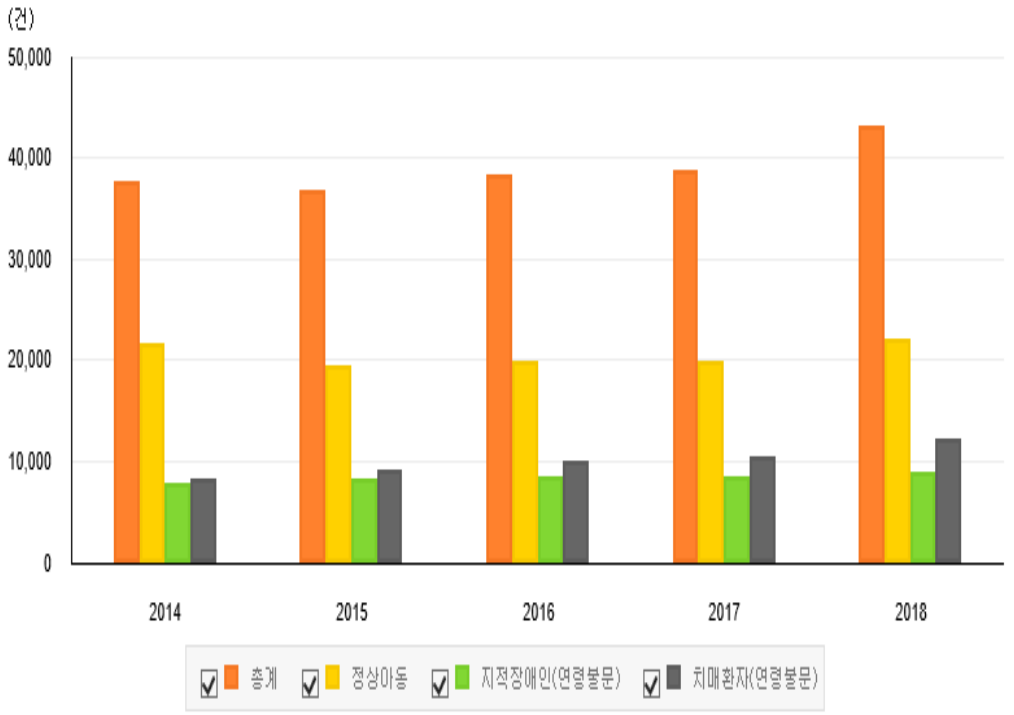
- 주거 생활에서 느낀 불편한 점을 개선하기 위해 고민.
- 어린 자녀, 여성을 대상으로 하는 범죄를 예방하기 위해 고민.

〈 강력범죄(흉악) 피해자 〉



자료 : 대검찰청, 「범죄분석」

실종아동등 신고접수 및 처리현황



프로젝트 주요 기능

자녀 안전 귀가 기능

- Google Maps API를 활용하여 "자녀의 위치"를 파악하고 "위치 정보"를 보호자의 스마트 폰 App에 표기함.
- 자녀 귀가 시, 탑승차량의 NFC Card 리더기에 자녀의 NFC Card로 Check-in하게 되면
  "귀가 차량의 번호"와 "운전자 정보"를 보호자에게 GCM Push Message로 알림.

여성 안전 귀가 기능

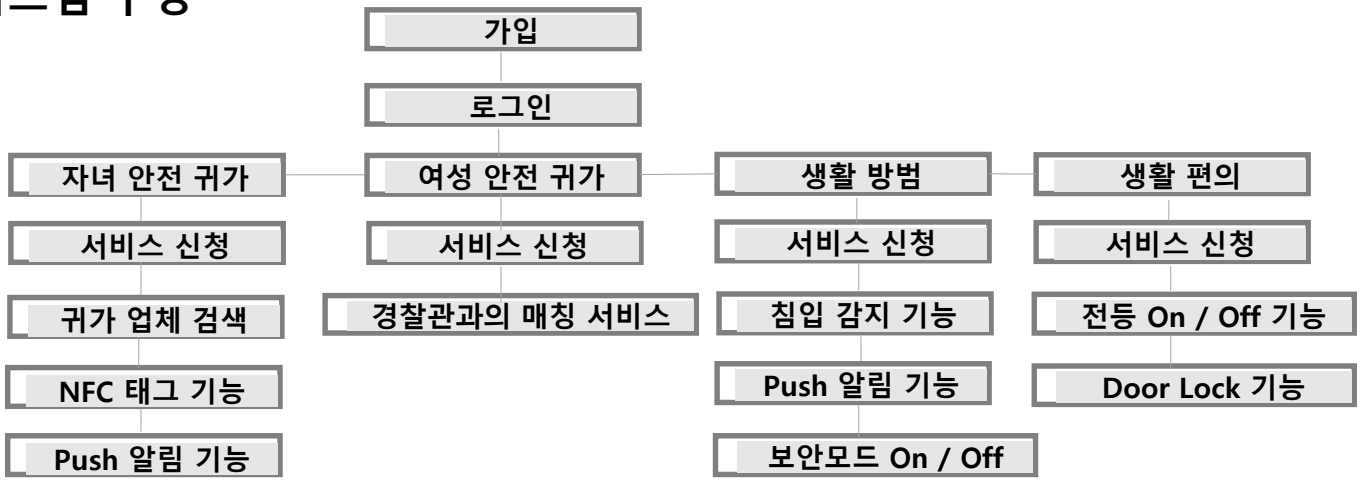
- 스마트 폰 App과 Web을 통하여 안전한 귀가를 도울 수 있는 "경찰관과의 매칭 서비스" 제공함.
- 자신의 위치를 기준으로 1km 내의 "경찰서의 위치가 Marker"로 표기됨.

생활 방범 및 편의 기능

- 집 내부에 Arduino "문 열림" 감지 센서, Arduino "온도" 감지 센서를 부착하고 "외부의 침입이 감지"되면
  "경고음"과 함께 GCM Push Message로 "위험"을 거주자에게 알림.
- "전등 On/Off", "Door Lock" 기능을 스마트 폰 App에서도 사용할 수 있도록 서비스 제공.

시스템 구성 및 개발 환경

시스템 구성



개발 환경

Front-End

- HTML5
- JavaScript
- J-Query
- I-Frame
- Tiles

Back-End

- Spring Framework (MVC 2)
- Ajax
- Oracle DBMS

Mobile

- Android OS
- Google Maps API
- NFC Tag Reader
- XML

본인의 역할

- "GCM"과 "Spring Framework" 기반의 "알람 메시지"의 백엔드 개발
- "Spring Framework" 기반의 "자녀 안전 귀가" 백엔드 개발

개인 개발 핵심 코드

-GCM_Controller.java (GCM Push Message 기능 부분)

```
sender = new Sender(API_KEY);//Sender Objection
regId = member.getReg_id();// Reg_id
   //System.out.println(home_alert);

Message.Builder builder = new Message.Builder(); //Message Builder

 if(alert_value[0].equals("r")){
     //Hot Sensor
     final String key="Hot_Sensor";
     final String value="지붕을 통하여 침입자가 발생했습니다.";
     System.out.println(key.toString()+","+value.toString());
     builder.collapseKey(collaspe_key);
     builder.addData(key.toString(),value.toString());

 }
 else if(alert_value[0].equals("w"))
 {
     //Touch Sensor
     final String key="Touch_Sensor";
     final String value="창문을 통하여 침입자가 발생했습니다.";
     System.out.println(key.toString()+","+value.toString());
     builder.collapseKey(collaspe_key);
     builder.addData(key.toString(), value.toString());
 }

msg = builder.build();    // builder is builded
try {
   sender.send(msg, regId, 30);

} catch (IOException e) {
   e.printStackTrace();
}

return "redirect:/safe/scan_success";
```

경고 메시지 종류 인식

메시지를 해당 Key에 저장

메시지를 스마트 폰 App로 전송

-Children_Controller.java (자녀 안전 귀가 기능 부분)

```
@RequestMapping(value="/safe_request")
public String safe_Request(Model model,@RequestParam String id, @RequestParam int category){

        if(category == 2){
        List<ChildrenBean> request_list = dao.requestSelect(id);

        model.addAttribute("request_list",request_list);

        }else {
                ChildrenBean children = dao.p_requestSelect(id);
                model.addAttribute("children",children);
        }

        return "c_safe_request";
}

@RequestMapping(value="/requestView")
public String requestView(Model model,@RequestParam int num){

        ChildrenBean c = dao.selectChildren(num);

        model.addAttribute("c",c);

        return "c_requestView";

}
@RequestMapping(value="/requestDelete")
public String requestDelete(@RequestParam String id){

        System.out.println(id);
        dao.requestDelete(id);

        return "c_safe_request";
}
```

자녀 안전귀가 신청

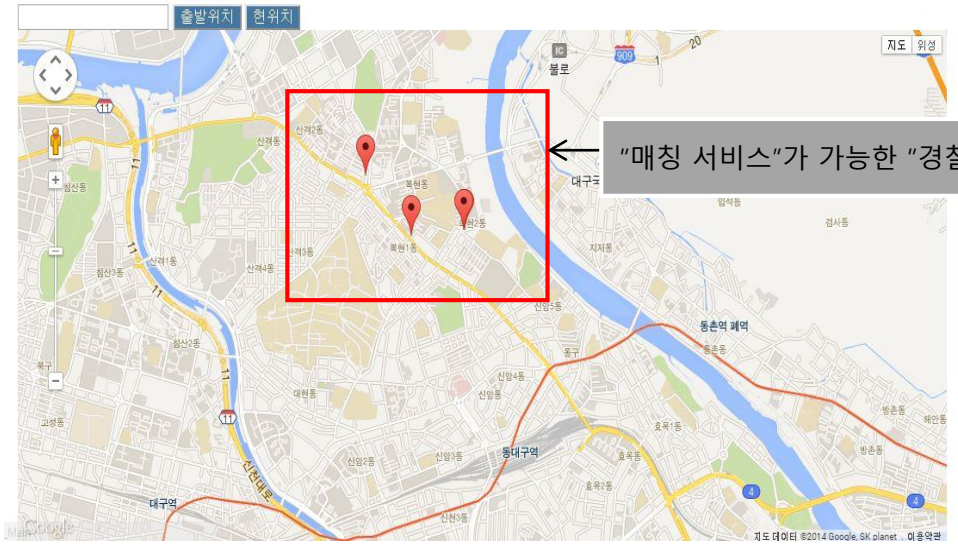
자녀 안전귀가 신청정보 출력

자녀 안전귀가 신청 삭제

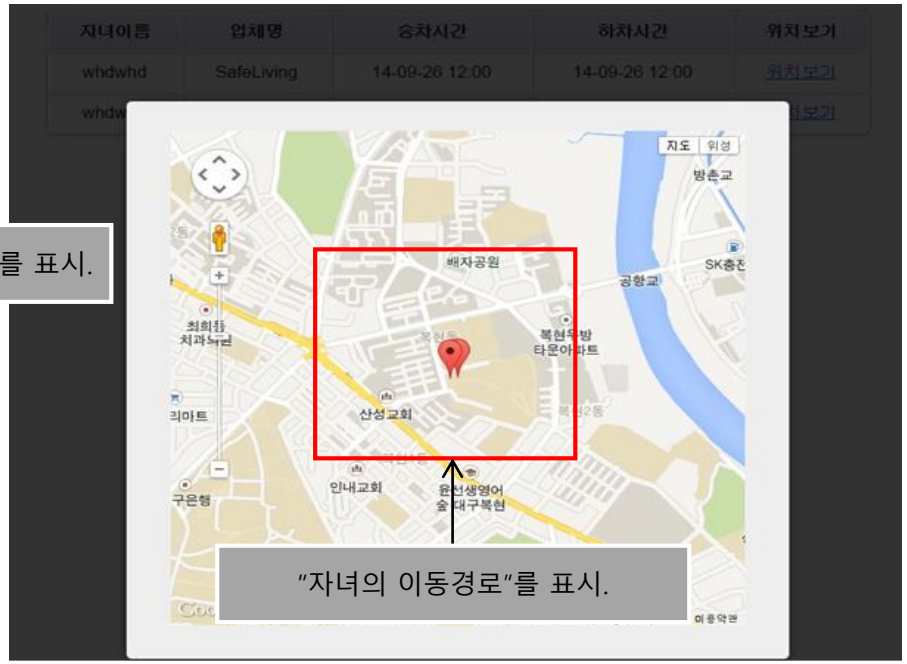
프로젝트 핵심 결과물

여성 안전 귀가 서비스(경찰관과의 매칭 서비스 부분)

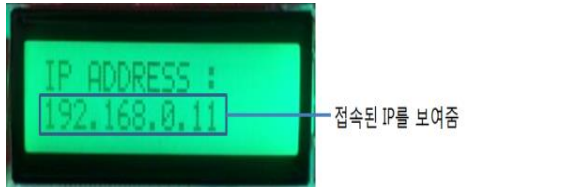
선택하실 곳을 선택하시면 예약페이지로 넘어갑니다.

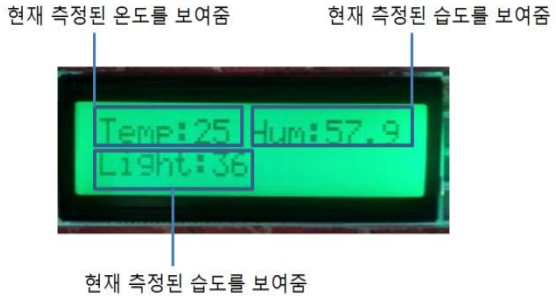


자녀 안전 귀가 서비스(이동 동선 기록 부분)

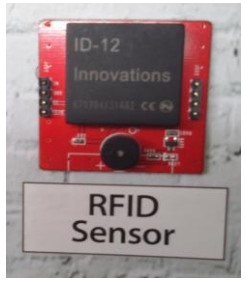


생활 방범 및 편의 기능(방범에 필요한 Arduino 센서)

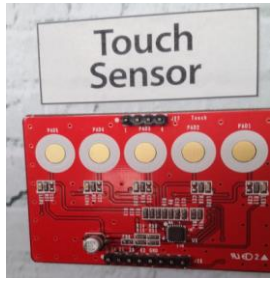




RFID Arduino 센서

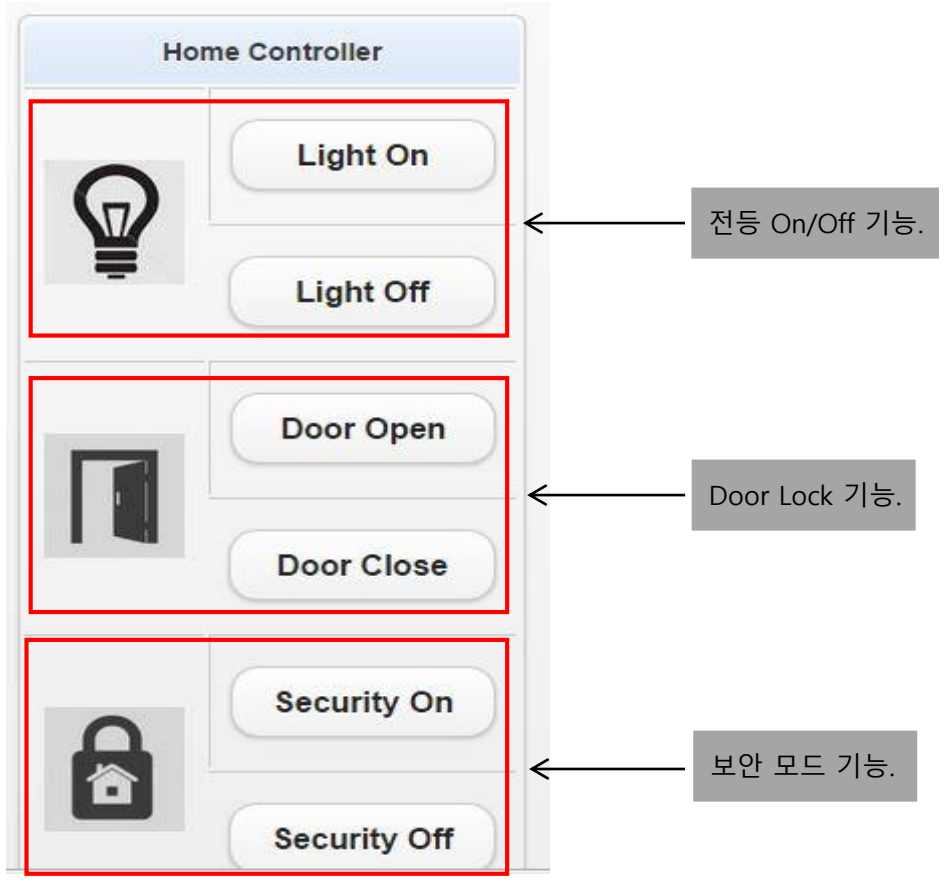


Touch Arduino 센서



Magnetic Arduino 센서

Smart Home Controller (스마트 폰 App 부분)

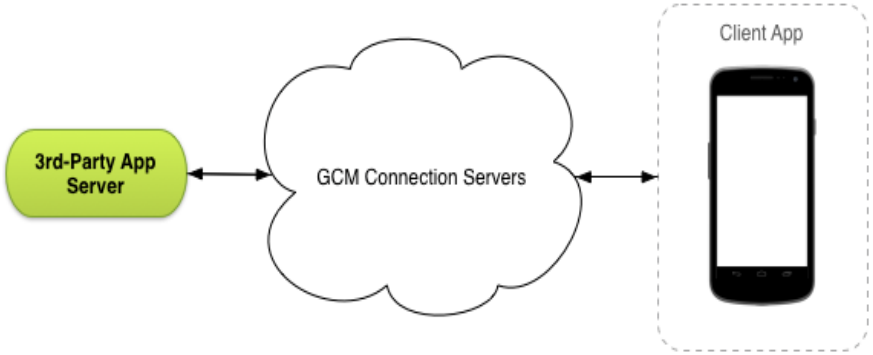


GCM Push Message

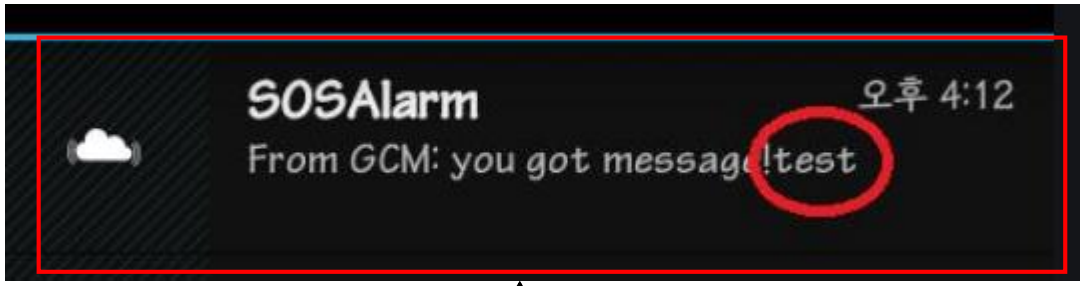
- GCM Push Message를 제공하는 GCM Server의 서비스 만료

(구동 방식 이미지, 예시 이미지로 대체)

구동 방식



GCM push Message 예시



"경고음"과 "침입 경로 정보"를 표시.

프로젝트 개발 기간 및 기여도

기여도 : 30 %

세부추진내용	과제수행기간									진도율(%)
	1주	2주	3주	4주	5주	6주	7주	8주	9주	
기획문서작성	■	■								10 %
DB설계		■	■	■						30 %
시제품 개발			■	■	■	■				60 %
디버깅					■	■	■			80 %
프로젝트 개선						■	■	■	■	90 %
발표									■	100 %

프로젝트 기대효과

- **“자녀, 여성 안전 귀가 기능”으로 “자녀, 여성을 대상으로 하는 범죄율” 감소.**
- **“생활 방범 기능”으로 “주거 침입 및 절도 범죄율” 감소.**
- **“생활 편의 기능”으로 “주거 생활의 편의성” 향상.**

프로젝트 후 느낀 점

- **“자녀 안전 귀가 기능” 구현 경험을 통해 “고객 친화적 기능 구현 능력” 향상.**
- **“Spring Framework”를 활용한 “MVC 2 모델” 구현 방식 숙달.**
- **“GCM API”, “NFC Reader”에 대한 이해도와 응용력 향상.**

Block-Chain을 활용한 근로 계약서 관리 시스템

프로젝트 개요

- 스마트 폰 App을 통하여 보다 안전하고 쉽게 "근로계약서"를 작성 및 관리하는 서비스 제공.
- Block-Chain의 특성 중 "정보 저장의 투명성"을 이용하여 "근로계약서의 정보"를 관리.

프로젝트 개발 배경

- "이중 근로계약서" 사례로 인한 피해 속출
- "근로계약서 위 변조" 사례 급증

- 이중 근로계약서 피해 사례

■ 졸업도 안한 학생들이 정직원보다 더한 과로에 시달리는 경우도 있다면서요. 표준협약서가 전혀 제 구실을 못하는 것 아닌가요?

현장실습을 보낼 때에는 3자가 표준 협약서를 쓰게 되어 있습니다. 학교, 학생, 기업이 표준 협약서를 쓰게 돼 있습니다. 표준 협약서는 기본적으로 이게 일반 직원들이 하는 노동의 개념이 아니라 실습이기 때문에 일반 노동자보다는 업무 강도라든가 이런 게 현저히 낮게 되어 있어요. 그래서 하루에 아주 길게 하더라도 7시간이고 여기에 1시간을 더하면 8시간까지 일할 수가 있습니다. 야간이나 휴일 근무는 할 수가 없는 상황이고요.

그런데 막상 학생들이 현장에 나가면 근로 계약서를 따로 작성을 합니다. 따로 작성하는 근로계약서에는 '이 학생과 협의 하에 연장근무 할 수 있다' 이런 식으로 돼 있어요. 그래서 구조적으로 이중계약 문제가 발생할 수밖에 없는 형태입니다. 그렇다 보니까 학교에 보고되거나 교육청에 보고되거나 하는 표준 협약서에는 '하루에 7시간만 일을 시킨다'고 돼 있는 거죠. 근데 실제로 학생과 기업 간에 맺는 근로 계약서는 대부분 다 연장 근무가 가능하도록 된 것이거든요. 이런 이중계약 문제는 현장실습 업체에 만연해 있고 거의 대부분 다 하고 있다고 보면 됩니다. 저희 취재진이 현장에서 학생들을 만나 물어보니 '한 반에 27명이 현장실습을 나갔는데 한 기업 빼고는 다 연장근무를 한다' 이 정도로 말할 정도니까요.

출처 : https://news.sbs.co.kr/news/endPage.do?news_id=N1004510940&plink=ORI&cooper=NAVER

프로젝트 주요 기능

회원 관리 기능

- 고용주와 고용자의 회원으로 구분되어 있으며, "회원 가입", "회원 정보", "회원 탈퇴" 기능 제공.

근로계약서 작성 기능

- 고용주와 고용자가 스마트 폰 App으로 "근로계약서"를 작성 및 합의를 단계적으로 이루어내는 기능 제공.

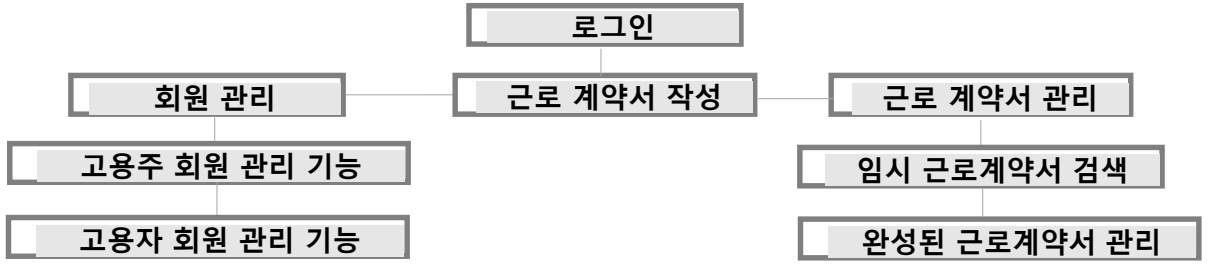
근로계약서 관리 기능

- 스마트 폰 App으로 작성된 "근로계약서의 내용"을 조회하고 관리 할 수 있는 기능.

Block-Chain을 활용한 근로 계약서 관리 시스템

시스템 구성 및 개발 환경

시스템 구성



개발 환경

Back-End

- Node.js
- MySQL
- Ethereum Block-Chain API

Mobile

- Android OS
- FCM API
- XML

Block-Chain을 활용한 근로 계약서 관리 시스템

본인의 역할

- **“Node.js” 기반의 시스템 백엔드 개발**
- **“Ethereum Solidity”를 활용하여 “근로계약서 내용 저장” 부분 구현**

개인 개발 핵심 코드

- App.js (서버 연결 및 모듈 포함 부분)

Ethereum Network와 통신하기 위해 Web3 모듈 사용

Node.js 서버 연결 및 MySQL과 연결

```
var Web3 = require("web3");
web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:8545"));
var express = require("express");
var EtherNode = express();
var server = require("http").createServer(EtherNode);
var io = require("socket.io")(server);
EtherNode.use(express.static(__dirname+'/views/'));

server.listen(8500);

var mysql=require("mysql");
var XMLHttpRequest = require("xmlhttprequest").XMLHttpRequest;
var xmlhttp = new XMLHttpRequest();

dbconn = mysql.createConnection({
    host     :"localhost",
    port : 3306,
    user : "root",
    password : "admit",
    database:"admit_db"
});

var bodyParser = require("body-parser");
EtherNode.use(bodyParser.urlencoded({
            extended: false
}));
EtherNode.use(bodyParser.json({ type: 'application/json' }));

EtherNode.get("/",function(request,response){

            response.sendFile("index.html"); //웹페이지를 리턴.

});
```

Block-Chain을 활용한 근로 계약서 관리 시스템

- App.js (블록화 된 근로계약서 정보를 열람하는 부분)

function createContract(...){...} → 빈 계약 블록을 생성.

function getContract(fn){...} → 생성된 계약 블록을 가져와서 근로계약서 내용을 저장시킴.

function getContractContent(addr){...} → 근로계약서의 정보가 기록된 계약 블록으로부터 내용을 가져옴.

EtherNode.post("/searchcontract/",function(request,response){...} → Ethereum 블록체인에 있는 내용을 불러와 근로계약서의 내용을 열람.

- Contract.sol (근로계약서의 정보가 담길 Solidity 부분)

```
pragma solidity ^0.4.11;

contract EmploymentContract{
    string public employee_name;
    bytes32 public employee_id_sha3_256;
    string public employer_name;


function set_employee_name(string _employee_name){
    if(setted_num < 24){
        employee_name = _employee_name;
        setted_num++;
    }
}
```

→ 계약 블록에 근로계약 내용을 삽입.

Block-Chain을 활용한 근로 계약서 관리 시스템

프로젝트 핵심 결과물

1.Database에 저장된 임시 근로계약서의 내용

```
mysql> select * from temp_contract;
| contract_num | employee_num | employer_num | employee_name | employer_name | begin_labor_date | end_labor_date | start_labor_time | end_labor_time | working_hours_per_day | working_hours
_per_week | day_off | wage_type | wage     | bonus_system | other_wage_system | pay_day_monthly | pay_day_weekly | pay_method | insurance  | contract_date        | company_name  | company_ad
dress                          | employee_address                                    | employee_contact | employee_confirm | employer_confirm | first_work_time | second_work_time | third_work_
time | contract_type_number |
|           21 |            5 |            4 | John          | James         | 2018/05/12       | 2018/08/12     | 09:00            | 18:00          |                   9.0 |
   45.0 | sat,sun | monthly   | 2500000 | efficendion  | No                |              25 | N              | account    | industrial | 2018/05/08 12:35:21 | CHEWY_FACTORY | 47 Scotts
Road,#16-03,04 Goldbell Towers | 71B Tanjong Pagar Road #03-01, Singapore 088492 | 010-2313-3432    | NULL             | Y                | 9:00-16:00      | 15:00-21:00      | 21:00-03:00
     |                    3 |
1 row in set (0.00 sec)
```

"근로 계약"이 성사 될 때까지 "임시 근로계약서의 내용"이 DB에 저장됨.

2.계약 블록의 정보 확인(블록화 성공)

addr	0: address: 0x2c2B77B0ca590AC1fbAf17e9De01C7118bF7c57e
begin_labor_date	0: string: 2018/05/12
bonus_system	0: string: efficendion
company_address	0: string: 47 Scotts Road,#16-03,04 Goldbell Towers
company_name	0: string: CHEWY_FACTORY
contract_date	0: string: 2018/05/08 12:35:21
contract_type_number	0: string: 3
day_off	0: string: sat,sun
employee_address	0: string: 71B Tanjong Pagar Road #03-01, Singapore 088492

"근로 계약"이 성사되어 "근로계약서의 내용"이 Ethereum Block에 저장됨.

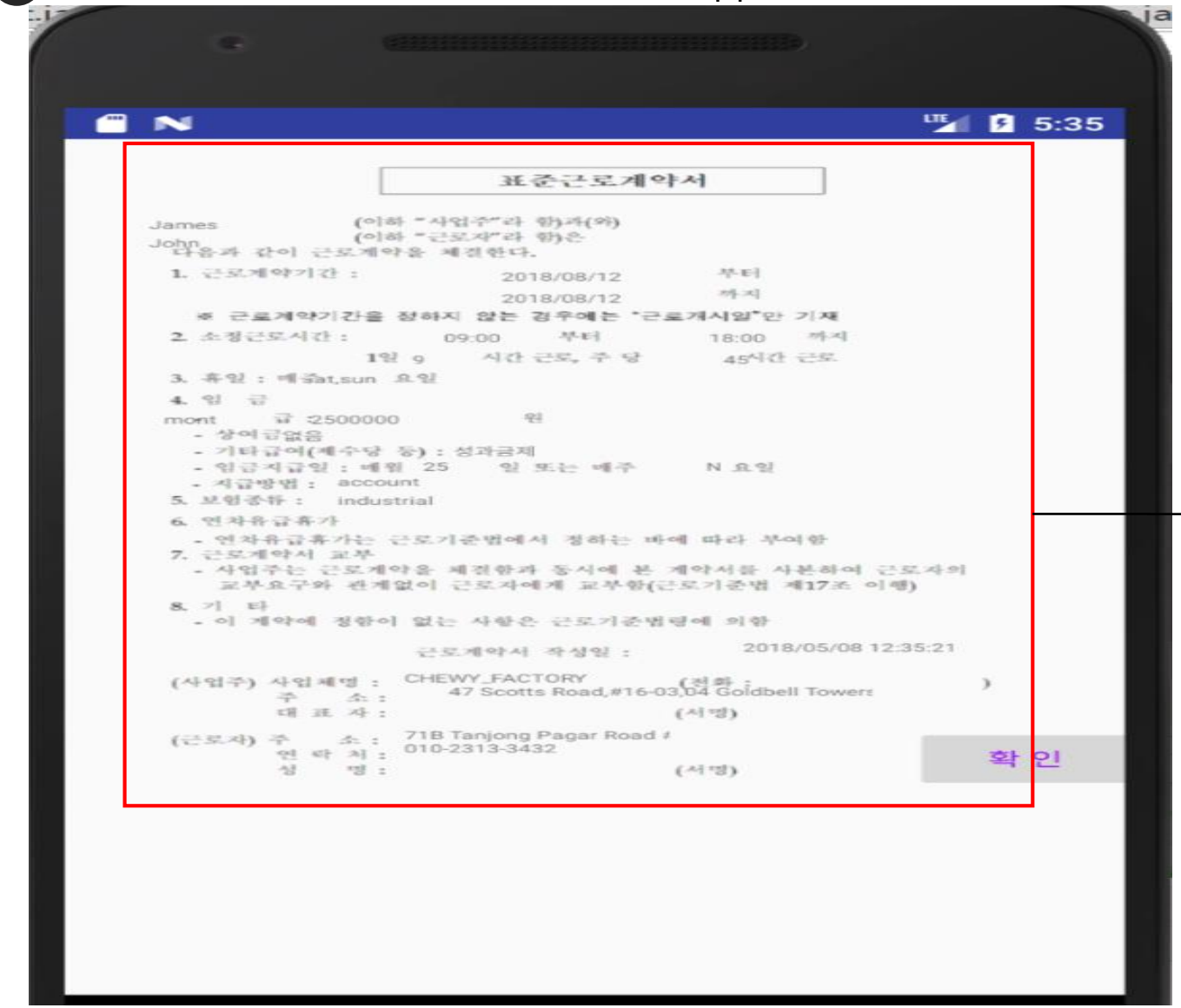
3.Database에 저장된 임시 근로계약서의 내용 삭제

```
mysql> select * from temp_contract;
Empty set (0.00 sec)
```

"임시 근로계약서의 내용"을 DB로부터 삭제함.

Block-Chain을 활용한 근로 계약서 관리 시스템

4.Block화 된 근로계약서의 내용 열람(App)



"완성된 근로계약서"를 스마트 폰 App으로 열람.

프로젝트 개발 기간 및 기여도

기여도 : 40 %

세부추진내용	과제수행기간										진도율(%)
	1주	2주	3주	4주	5주	6주	7주	8주	9주	10주	
기획문서작성	■	■									10 %
DB설계		■	■	■							30 %
시제품 개발			■	■	■	■	■				60 %
디버깅					■	■	■	■			80 %
프로젝트 개선						■	■	■	■		90 %
발표										■	100 %

Block-Chain을 활용한
근로 계약서 관리 시스템

프로젝트 기대효과

- **“이중 근로계약서”나 “근로계약서 위 변조”의 위험 감소.**
- **원거리에서도 고용주와 고용자가 안전하게 “근로계약서” 작성 가능.**
- **스마트 폰 App으로 근로계약서 관리가 가능하기에 “편의성” 향상.**

프로젝트 후 느낀 점

- **기존의 Database와 다른 “탈 중앙화 정보 저장” 기법인 “Block-Chain 기술”에 대한 응용 능력 향상.**
- **“Node.js Server 구현 기술” 습득.**
- **“Ethereum Solidity 언어” 습득.**