



CONTENTS 1. "보안성이 뛰어난 채팅 프로그램"

CONTENTS 2. < 「기자인 쇼핑몰"

CONTENTS 3. "Home Security and Care System"

CONTENTS 4. "Block-Chain을 활용한 근로 계약서 관리 시스템"

☞ 프로젝트 개요

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

<u>프로젝트 개발 배경</u>

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

₩ 노컷뉴스

피싱·해킹에 두려운 이용자들...어떻게 해야 하나?

페이스북 반년만에 또 해킹, 5000만 사용자 범죄 노출...내 계정 ... 인터넷 발달과 전세계적으로 높아진 스마트폰 보급률은 동시에 피싱이나 해킹 위협을 높이고 있다.

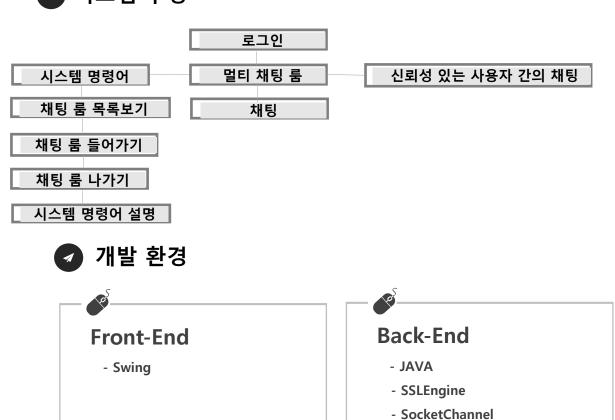


Oct 8, 2018

<u>프로젝트 주요 기능</u>

- ☑ 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.
 - Selector와 SocketChannel을 통하여 신뢰성 있는 사용자간의 데이터를 송수신 할 수 있도록 함.
- ☑ 시스템 명령어 기능
 - 채팅 서비스에 필요한 다양한 명령어 기능을 제공.

- ✓ 시스템 구성 및 개발 환경
 - ☑ 시스템 구성



- Selector

- 본인의 역할
 - "SSLEngine"과 "SocketChannel"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 class MessengerServer extends MessengerBasic {
public static void main(String args[]) {
                                                                                                      public boolean isActiveSvr;
                                                                                                      public SSLContext context;
         try {
                                                                                                      public Selector selector;
                   MessengerServer server = new MessengerServer("TLS","localhost",8700);
                                                                                                      MessengerRoomUserInfo roomUserInfo;
                   server.startServer():
                                                                                                      public MessengerServer(String protocol, String hostAddress, int portNumber) throws Exception {
                                                                                                         roomUserInfo = new MessengerRoomUserInfo();
          } catch (Exception e) {
                                                                                                                context=SSLContext.getInstance(protocol);
                   // TODO Auto-generated catch block
                                                                                                                context.init(createKeyManagers(".\\bar{W}\bin\bar{W}\keystore\bar{W}\SSLSocketServerKey\bar{W}\bar{W}","123456","123456"),
                                                                                                                createTrustManagers(".\\wbin\\wk.keystore\\wkssLSocketServerKey\\wksp.","123456"), new SecureRandom());
                   e.printStackTrace();
                                                          SSL/TLS 방식의 서버 구동 시작
                                                                                                               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();
                                                          Non-Blocking 서버 채널 성
                                                                                                                channel.configureBlocking(false);
                                                                                                                channel.socket().bind(new InetSocketAddress(portNumber));
                                                                                                                channel.register(selector, SelectionKey.OP ACCEPT);
```

isActiveSvr = true;

MessengerServerSender.java (서버 측 데이터 송신 부분) MessengerServerReceiver.java (서버 측 데이터 수신 부분) synchronized public void send(SocketChannel channel,SSLEngine engine,String m)throws IOException(synchronized protected void recv(SocketChannel channel,SSLEngine engine) throws Exception { AppData.clear(); NodeNetData.clear(); AppData.put(m.getBytes()); int waitToRecvMillis=50; AppData.flip(); try { int byterecv = channel.read(NodeNetData); -> 채널을 통해 암호화된 데이터 while(AppData.hasRemaining()) { NetData.clear(); if(byterecv > 0) { SSLEngineResult result = engine.wrap(AppData,NetData); NodeNetData.flip(); switch(result.getStatus()) { while(NodeNetData.hasRemaining() && !isClosed) { case OK: Wrapping으로 데이터 암호화 처리 NodeAppData.clear(); result = engine.unwrap(NodeNetData, NodeAppData); NetData.flip(); → Unwrapping으로 데이터 복호화 처리 switch(result.getStatus()) { while(NetData.hasRemaining()) { case OK: channel.write(NetData); NodeAppData.flip(); |읔 통해 암호화된 데이터 송신| Charset charset = Charset.defaultCharset(); String message = charset.decode(NodeAppData).toString(); 복호화된 데이터를

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

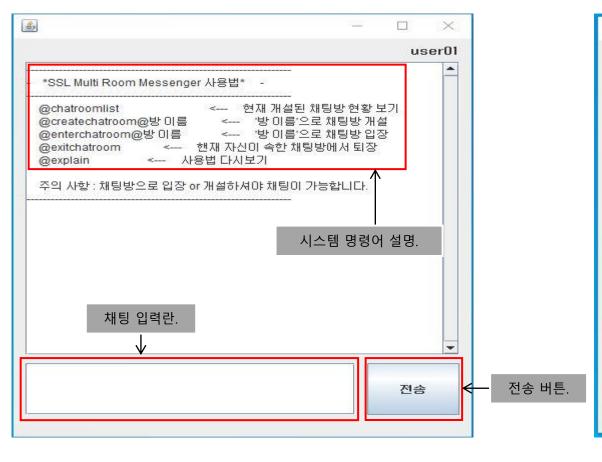
import javax.net.ssl.SSLEngine;

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

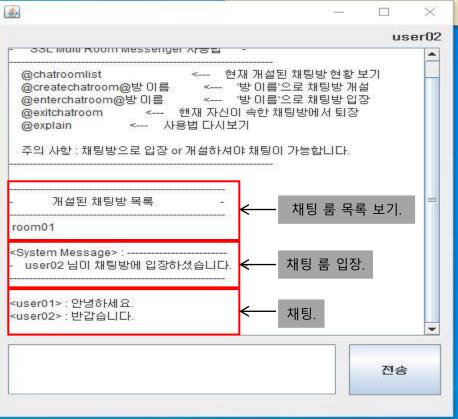
```
public class MessengerRoomUserInfo {
    public Vector<String> room;
    public Vector<UserInfo> info;
    public HashMap<String,Vector<UserInfo>> map = new HashMap<String,Vector<UserInfo>>();

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

- <u>프로젝트 핵심 결과물</u>
- 채팅 프로그램(메인 화면 부분)



☑ 채팅 프로그램(채팅 부분)



☑ 프로젝트 개발 기간 및 기여도

기여도 : 55 %

세부추진내용			TI = 9 (a/)					
	1주	2주	3주	4주	5주	6주	7주	진도율(%)
기획문서작성								20 %
시제품 개발								60 %
디버깅								80 %
프로젝트 개선								90 %
발표								100 %

<u>프로젝트 기대효과</u>

- ☑ "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.

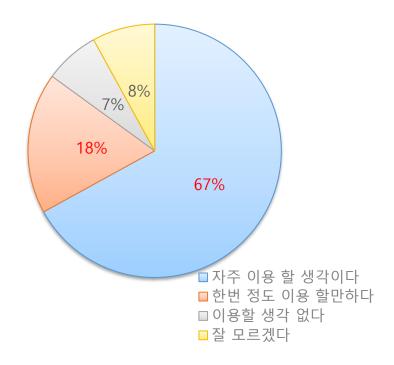
☑ 프로젝트 개요

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

<u>프로젝트 개발 배경</u>

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

디자인 쇼핑몰이 생긴다면 사용할 의향이 있는가?



패션 디자인학과 학생 35명을 대상으로 한 앙케트 조사 자료

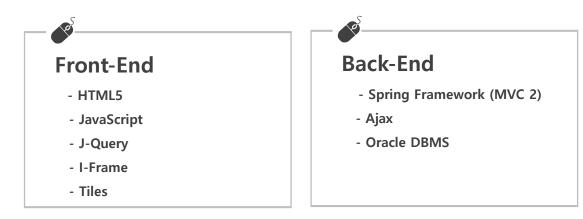
● 프로젝트 주요 기능

- ☑ 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



- **본인의 역할**
 - ☑ "Spring Framework"를 활용한 "디자인 쇼핑몰 기능"의 "구매", "상품", "장바구니" 백엔드 개발
- <u>개인 개발 핵심 코드</u>

ShopController.java (구매 부분)

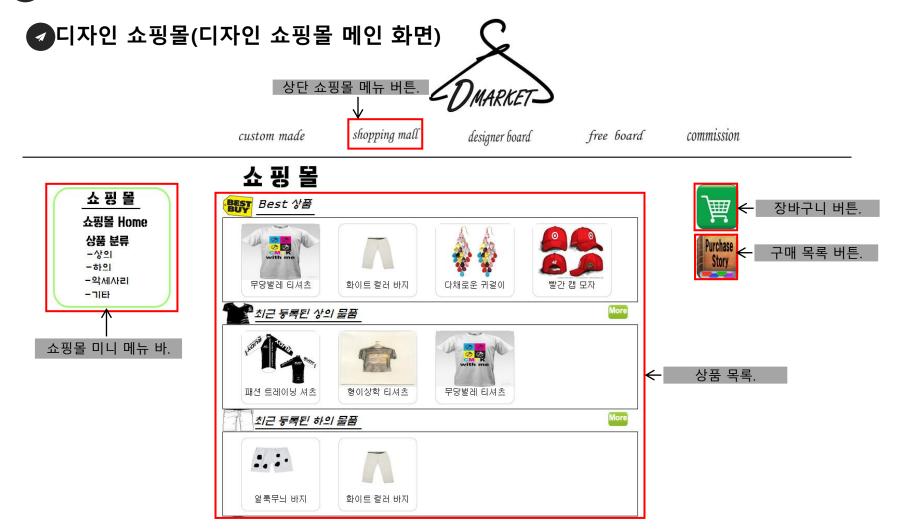
String status="입금대기"; phs.setMem_num(mem_num); phs.setP address(addr); ┛ 구매 정보 등록 phs.setP_date(time); phs.setP_receiver(receiver); phs.setP_post_num(post_num); phs.setDes num(des num); phs.setP_phone_num(phone_num); phs.setP_status(status); phsdao.Insert_Purchase(phs); product p =new product(); p.setMinus_amount(minus_amount); p.setPro_num(pro_num); pdao.Update amount(p); return"redirect:/spring/customer_purchase list?category_num=6&pageNum=1";

ShopController.java (상품 부분)

```
MultipartFile img_stream_1 = p.getImg_stream_1();
@RequestMapping(value = "/product_regist", method = RequestMethod.POST)
public String product_regist(@ModelAttribute ("product") product p,int pageNum,int category_num,HttpSession session,
                                                                                                     if(img_stream_1.getOriginalFilename().equals("")){
Model model ,HttpServletRequest request) throws IOException {
                                                                                                            return"redirect:/spring/product_regist?category_num="+category_num+"&pageNum="+pageNum;
                                                                                                     }else{
 String file_path = "C:/workspace/INNO/WebContent/shop_item/";
                                                                                                            p.setPro_date(time);
 String file_path = "/workspace-sts-3.4.0.RELEASE/INNO/WebContent/shop_item/";
                                                                                                            Integer des_num=ddao.select_des_num(m.getMem_num());
 String tmp_file_path=session.getServletContext().getRealPath("/");
                                                                                                            String ext1=img_stream_1.getOriginalFilename().substring(img_stream_1.getOriginalFilename().lastIndexOf(".")+1);
                                                    의류 디자인 이미지를 저장소에 저장
                                                                                                            String img_name_1=des_num+"_"+time2+"_1."+ext1;
 SimpleDateFormat dayTime = new SimpleDateFormat("yyyy-MM-dd HH:mm");
 String time = dayTime.format(new Date(System.currentTimeMillis()));
                                                                                                            p.setPro_img(img_name_1);
                                                                                                            byte[] fileData1 = img_stream_1.getBytes();
  Member m = new Member();
 m = (Member)session.getAttribute("member");<── 세션에서 쇼핑몰 판매 회원 정보 획득
                                                                                                            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();
                                                                                                             p.setDes_num(des_num);
                                                                                                            pdao.Insert_Product(p);
                                                                                                            model.addAttribute("product",p);
```

```
ShopController.java (<u>장바구니 부분</u>)
@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();
                                                                               장바구니에 디자인 상품 담기
                   cart.setMem_num(mem_num);
                   cart.setPro_num(pro_num);
                  cartdao.Insert_Cart(cart);
          return "redirect:/spring/cart_list?pageNum=1&category_num=5";
```

☑ 프로젝트 핵심 결과물



☑ 프로젝트 개발 기간 및 기여도

기여도 : 30 %

세부추진내용		진도율(%)								
	1주	2주	3주	4주	5주	6주	7주	8주	9주	신도뀰(%)
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.

<u>프로젝트 개요</u>

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

<u>프로젝트 개발 배경</u>

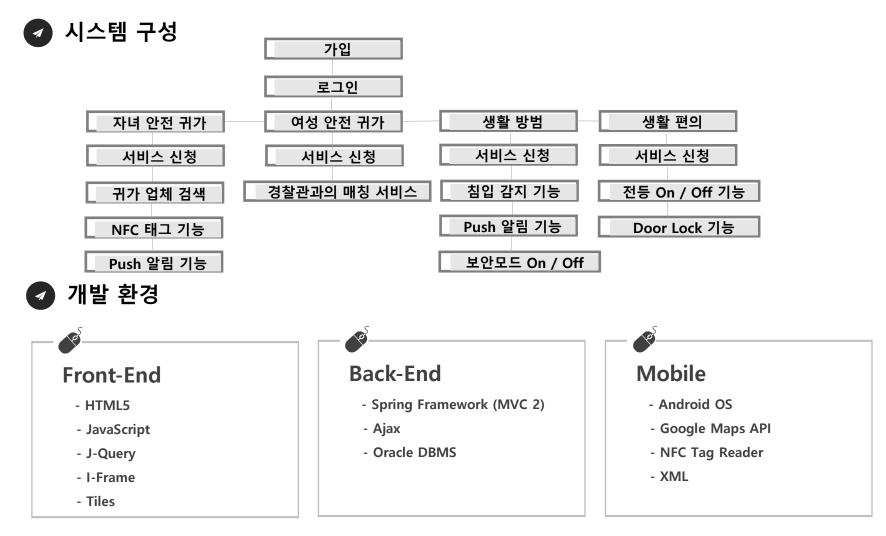
- 주거 생활에서 느낀 불편한 점을 개선하기 위해 고민.
- 어린 자녀, 여성을 대상으로 하는 범죄를 예방하기 위해 고민. 〈 강력범죄(흉악) 피해자 〉





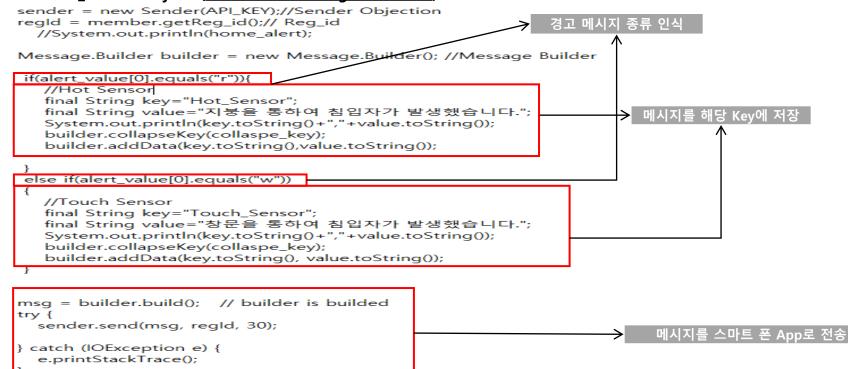
- <u>프로젝트 주요 기능</u>
 - ☑ 자녀 안전 귀가 기능
 - Google Maps API를 활용하여 "자녀의 위치"를 파악하고 "위치 정보"를 보호자의 스마트 폰 App에 표기함.
 - 자녀 귀가 시, 탑승차량의 NFC Card 리더기에 자녀의 NFC Card로 Check-in하게 되면 "귀가 차량의 번호"와 "운전자 정보"를 보호자에게 GCM Push Message로 알림.
 - ☑ 여성 안전 귀가 기능
 - 스마트 폰 App과 Web을 통하여 안전한 귀가를 도울 수 있는 "경찰관과의 매칭 서비스" 제공함.
 - 자신의 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



- 본인의 역할
 - "GCM"과 "Spring Framework" 기반의 "알람 메시지"의 백엔드 개발
 - ☑ "Spring Framework" 기반의 "자녀 안전 귀가" 백엔드 개발
- <u>개인 개발 핵심 코드</u>

-GCM Controller.java (GCM Push Message 기능 부분)



return "redirect:/safe/scan_success";

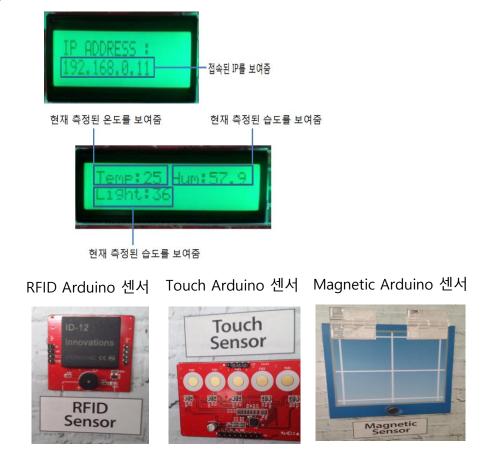
-Children_Controller.java (자녀 안전 귀가 기능 부분)

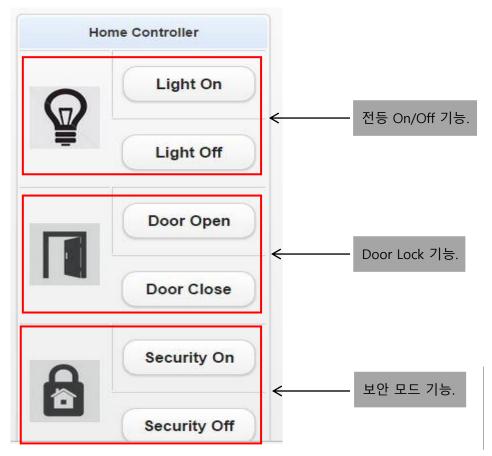
```
@RequestMapping(value="/safe_request")
public String safe_Request(Model model,@RequestParam String id, @RequestParam int category){
         if(category == 2){}
         List < ChildrenBean > request_list = dao.requestSelect(id);
                                                                            자녀 안전귀가 신청
         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);
                                                                          자녀 안전귀가 신청정보 출력
         model.addAttribute("c",c);
         return "c_requestView";
@RequestMapping(value="/requestDelete")
public String requestDelete(@RequestParam String id){
         System.out.println(id);
                                              자녀 안전귀가 신청 삭제
         dao.requestDelete(id);
         return "c_safe_request";
```

- 프로젝트 핵심 결과물
 - ❷여성 안전 귀가 서비스(경찰관과의 매칭 서비스 부분) ❷ 자녀 안전 귀가 서비스(이동 동선 기록 부분)



❷ 생활 방범 및 편의 기능(방범에 필요한 Arduino 센서) ❷ Smart Home Controller (스마트 폰 App 부분)





- **GCM** Push Message
 - GCM Push Message를 제공하는 GCM Server의 서비스 만료 (구동 방식 이미지, 예시 이미지로 대체)



☑ 프로젝트 개발 기간 및 기여도

기여도 : 30 %

세부추진내용		진도율(%)								
	1주	2주	3주	4주	5주	6주	7주	8주	9주	신도절(%)
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.

<u>프로젝트 후 느낀 점</u>

- "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.

☑ 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.

<u>프로젝트 개발 배경</u>

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

- 이중 근로계약서 피해 사례

■ 졸업도 안한 학생들이 정직원보다 더한 과로에 시달리는 경우도 있다면서요. 표준협약 서가 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 개 념이 아니라 실습이기 때문에 일반 노동자보다는 업무 강도라든가 이런 게 현저히 낮게 되 어 있어요. 그래서 하루에 아주 길게 하더라도 7시간이고 여기에 1시간을 더하면 8시간까 지 일할 수가 있습니다. 야간이나 휴일 근무는 할 수가 없는 상황이고요.

그런데 막상 학생들이 현장에 나가면 근로 계약서를 따로 작성을 합니다. 따로 작성하는 근 로계약서에는 '이 학생과 협의 하에 연장근무 할 수 있다' 이런 식으로 돼 있어요. 그래서 구 조적으로 이중계약 문제가 발생할 수밖에 없는 형태입니다. 그렇다 보니까 학교에 보고되

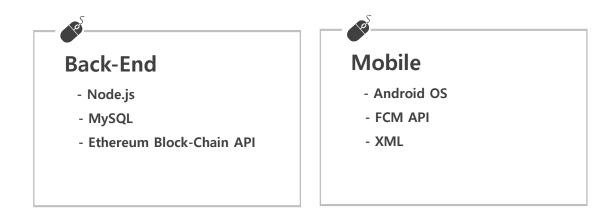
거다 교육성에 모고되거다 하는 표는 답락시에는 아무에 7시간된 물을 시킨다고 돼 있는 거죠. 근데 실제로 학생과 기업 간에 맺는 근로 계약서는 대부분 다 연장 근무가 가능하도록 된 것이거든요. 이런 이중계약 문제는 현장실습 업체에 만연해 있고 거의 대부분 다 하고 있다고 보면 됩니다. 저희 취재진이 현장에서 학생들을 만나 물어보니 '한 반에 27명이현장실습을 나갔는데 한 기업 빼고는 다 연장근무를 한다' 이 정도로 말할 정도니까요.

- ☑ 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.

- ✓ 시스템 구성 및 개발 환경
 - ☑ 시스템 구성



☑ 개발 환경



- 본인의 역할
 - ☑ "Node.js" 기반의 시스템 백엔드 개발
 - **☞** "Ethereum Solidity"를 활용하여

 "근로계약서 내용 저장" 부분 구현

Ethereum Network와 통신하기 위해 Web3 모듈 사용

Node.is 서버 연결 및 MySQL과 연결



- App.js (<u>서버 연결 및 모듈 포함 부분</u>)

```
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.

- App.js (블록화 된 근로계약서 정보를 열람하는 부분)

```
function createContract(...){...}

변계약 블록을 생성.

### 생성된 계약 블록을 가져와서 근로계약서 내용을 저장시킴.

### 전쟁시 기록된 계약 블록으로부터 내용을 가져옴.

### 전쟁시의 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{

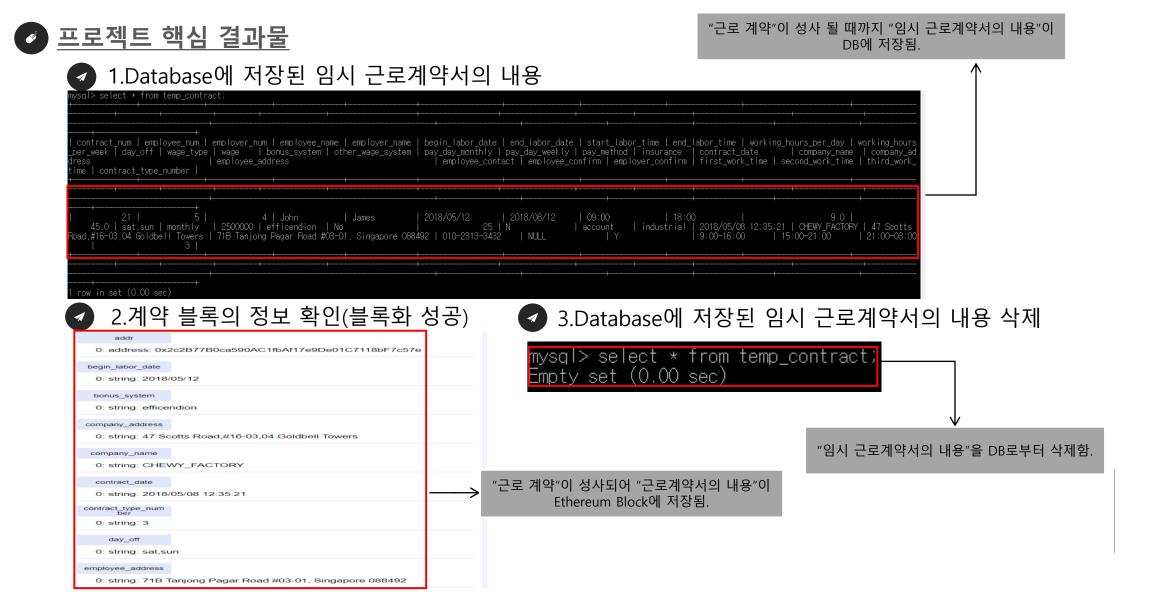
   string public employee_name;

   bytes32 public employee_id_sha3_256;

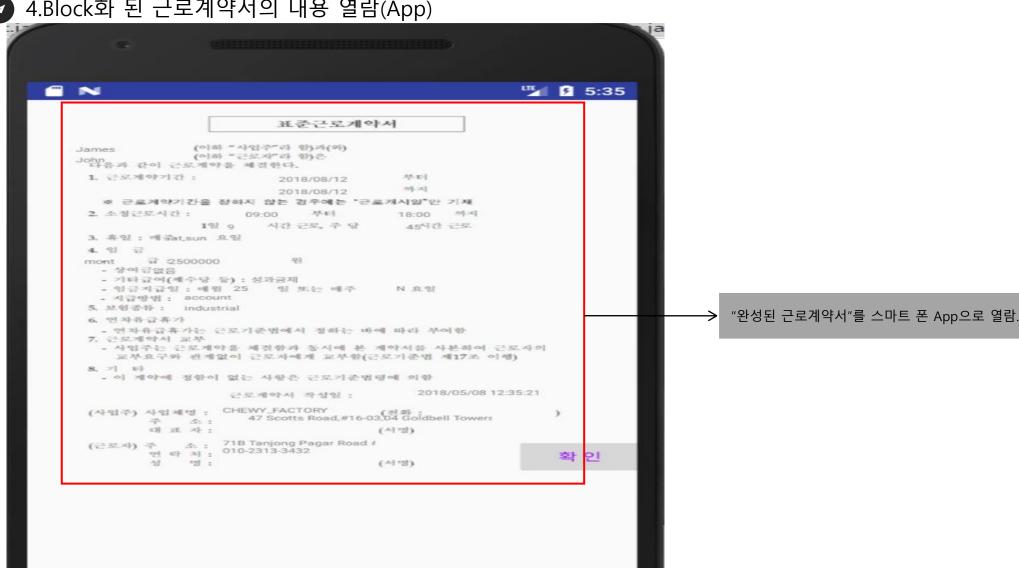
   string public employer_name;
```

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

계약 블록에 근로계약 내용을 삽입.



🕢 4.Block화 된 근로계약서의 내용 열람(App)



☑ 프로젝트 개발 기간 및 기여도

기여도: 40 %

1 1 1 1 7 7 7											i i
세부추진내용		TI = 0 (0/)									
	1주	2주	3주	4주	5주	6주	7주	8주	9주	10주	진도율(%)
기획문서작성											10 %
DB설계											30 %
시제품 개발											60 %
디버깅											80 %
프로젝트 개선											90 %
발표											100 %

<u>프로젝트 기대효과</u>

- ☑ "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.
- Tethereum Solidity 언어" 습득.