

자동 출석체크 시스템 구현

2020.11.03

중부대학교 정보보호학과

지도교수 : 김민수 교수님

3조

진영훈
반정원
이슬기
최명진
한상우

목 차

- 01 조원 편성
- 02 주제 선정
- 03 구 상 도
- 04 추진 경과
- 05 개발 환경 및 개발 내용
- 06 개발 시스템 운영
- 07 결론 및 기대 효과

이름	역할
진영훈(조장)	전자출결 시스템 개발, 백엔드 개발
반정원	프론트엔드 개발, ppt 작성
이슬기	프론트엔드 개발, 백엔드 개발
최명진	전자출결 시스템 개발, 보고서 작성
한상우	백엔드 개발, 서버 관리



주제 선정

누구를 위한 전자출결? 학생들 불편만 '가중'

일방적으로 이용 강요하는 학교 vs. 불편함을 느끼는 학생들

15.06.15 15:12 | 최종 업데이트 15.06.15 16:58 | 하도영(hdy2479) ▼

👍 좋아요 98개 + 크게 | - 작게 | 📄 인쇄 | 🔗 URL줄이기 | ☆ 스크랩

🔍 원고료로 응원하기

'불통' 전자출결 시스템 때문에 많은 대학생이 불편을 겪고 있

전자 출결 시스템(Electronic attendance-absence recording system)은 학생증 또는 애플리케이션을 이용하여 신속하고 정확하게 출결을 기록하는 시스템이다. 기존의 호명식 출결 방식에 비해 효율적이고 편리하다는 장점이 있지만, 학생들은 수업 시간에 출결을 위해 스마트폰을 꺼내거나 앱을 실행하는 불편을 겪고 있다.

정부는 2010년부터 학부교육 선도대학 육성사업(아래 ACE사업)을 지원함으로써 대학이 학부 교육에 관심을 갖도록 하고 교육 과정 및 교육 지원 시스템을 개선하도록 유도하였다. 위해서는 대학 학과 수업 때 '선도 모델'을 도입하는 것이 중요하다.

전자출결 도입 1년, 학생들 '출퇴' 여전

한림미디어랩 The H | 승인 2019.09.27 08:51 | 댓글 0

한 교양 수업에서만 20여 명 적발

한림대(총장 김종수)는 지난해 학생들의 편의를 위해 '전자출결' 제도를 도입했다.

이 제도는 기존에 교수가 학생을 호명하던 것과 다르게 학생이 전자기계에 모바일 어플 혹은 학생증을 태그해 출석을 확인하는 방식으로 강의실 일정 반경 안에만 있으면 출석 인증을 할 수 있다. 그런데 이를 악용해 출석 체크만 한 뒤 수업을 듣지 않고 가는 '출퇴'를 하는 학생이 늘어 문제가 되고 있다.

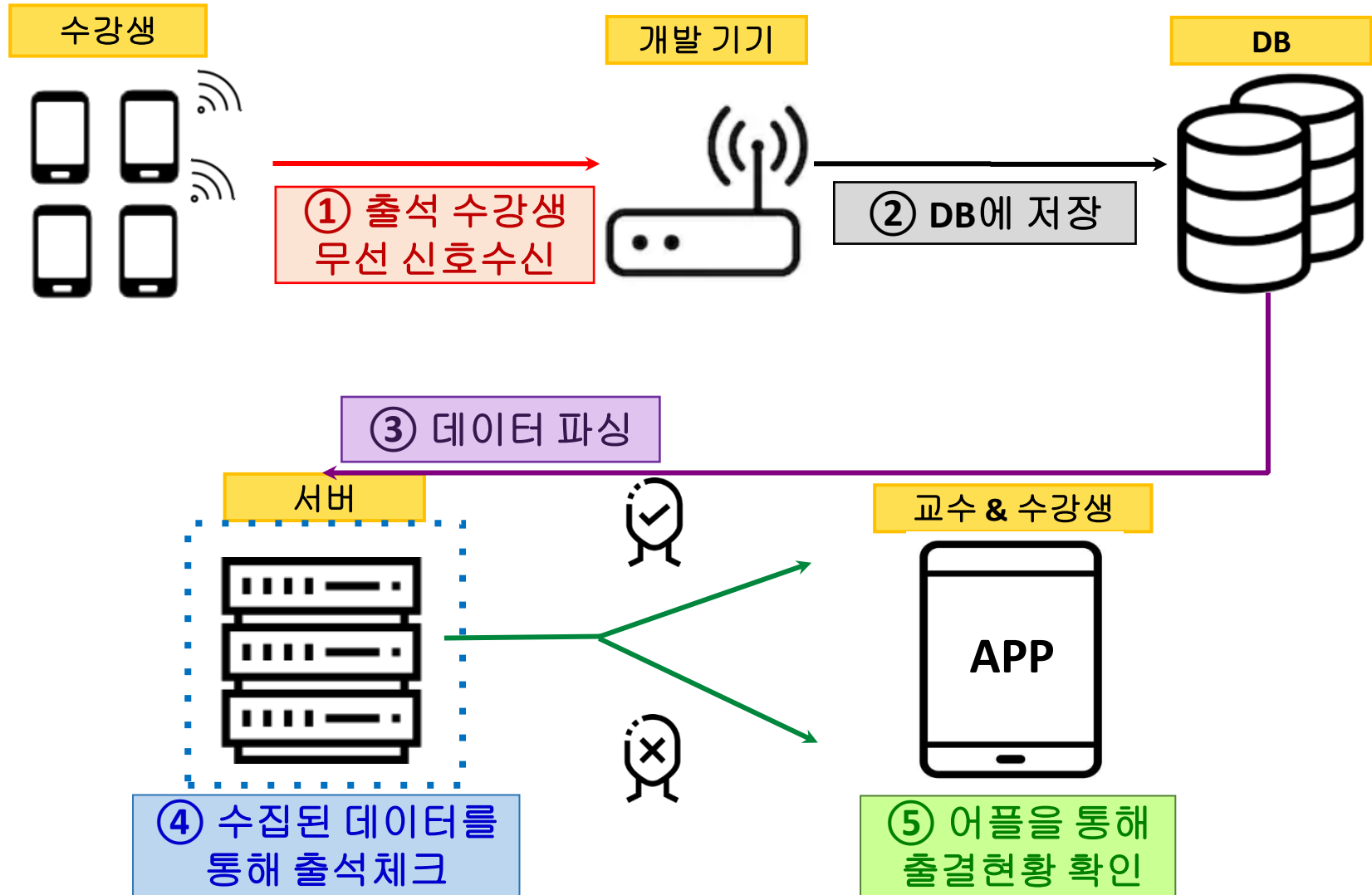
한 교양 수업에서는 20여 명이 출퇴한 것이 적발됐다. A 교수는 "수업 인원이 강의실을 꽉 채워 남는 자리가 없어야 하는데 빈자리가 많다. 학생들이 출퇴 하는 것 같은데 수업이 끝나고 출석 체크를 다시 하겠다"고 말했다. 출석 체크를 다시 한 결과 20여 명이 출퇴를 한 것으로 밝혀졌다. 다른 수업에서도 학생들이 출석 체크만 한 뒤 사라지는 모습은 쉽게 볼 수 있다.

출퇴보다 더욱 큰 문제는 학생들의 출퇴에 대한 그릇된 인식이었다. 출퇴를 하지 않은 학생이 출퇴를 하는 학생과 동일한 출결 점수를 받는 것은 불공평하다는 글을 한림대 커뮤니티인 에브리타임에 올리자 반박글이 이어졌다. 출퇴를 하는 사람들이 "너도 해" "내가 내 돈 내고 수업을 안 듣겠는데 무슨 상관이야"고 말하면서 논쟁이 벌어졌다. 이처럼 많은 학생들이 출퇴가 잘못이라고 생각하지 않았던 것이다.

한림대 교무팀은 "교수가 전자 출결 시스템 안에 비밀번호를 생성하고 학생들에게 재출석을

출석체크 자동화 및 부정행위 방지가 가능한 전자출결 시스템 개발 추진

구상도



추진 경과

추진 기간 (2020년) 수행 업무	3월	4월	5월	6월	7월	8월	9월	10월
자료 조사 및 연구								
전자출결 시스템 개발								
백엔드 개발								
프론트엔드 개발								
시스템 검증 및 보완								



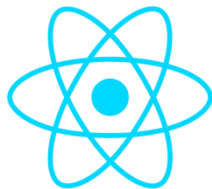
개발 환경 및 개발 내용 (1/10)

개발 환경

Development Language



APP



React Native

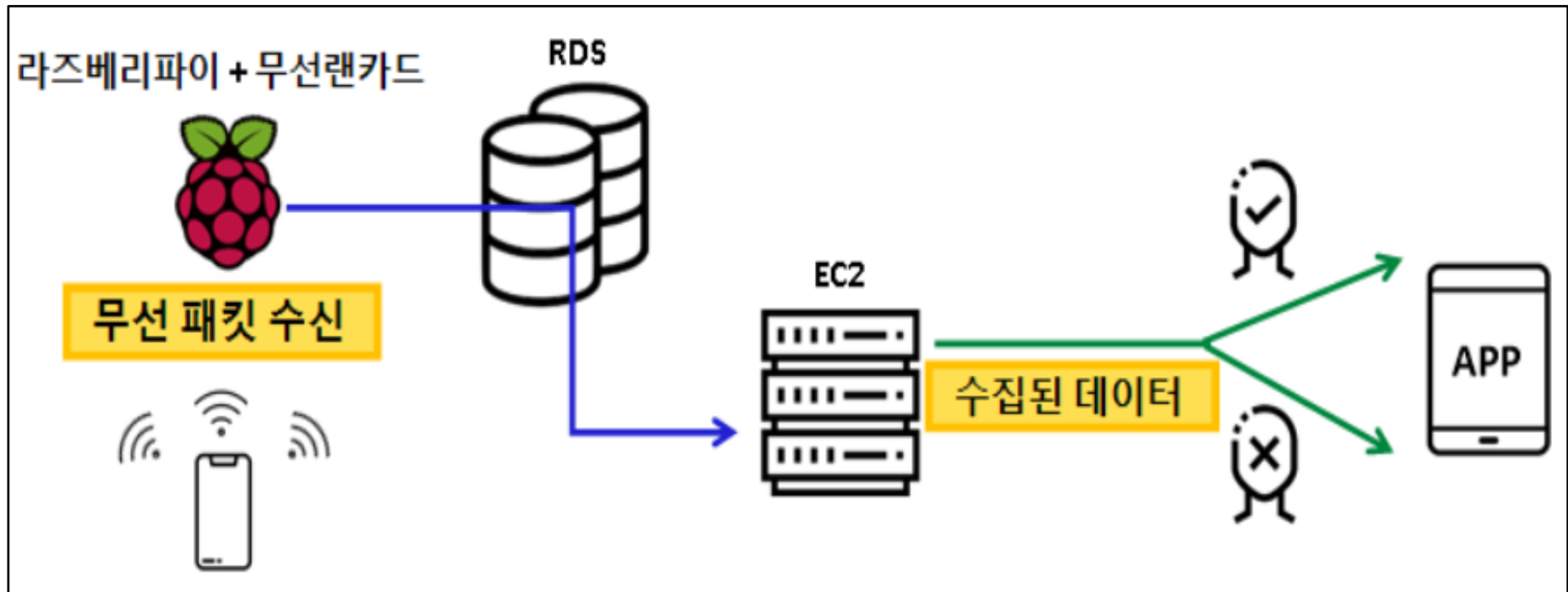
DB & Server



개발 환경 및 개발 내용 (2/10)

개발 내용(1/9)

○ 시스템 구성체계



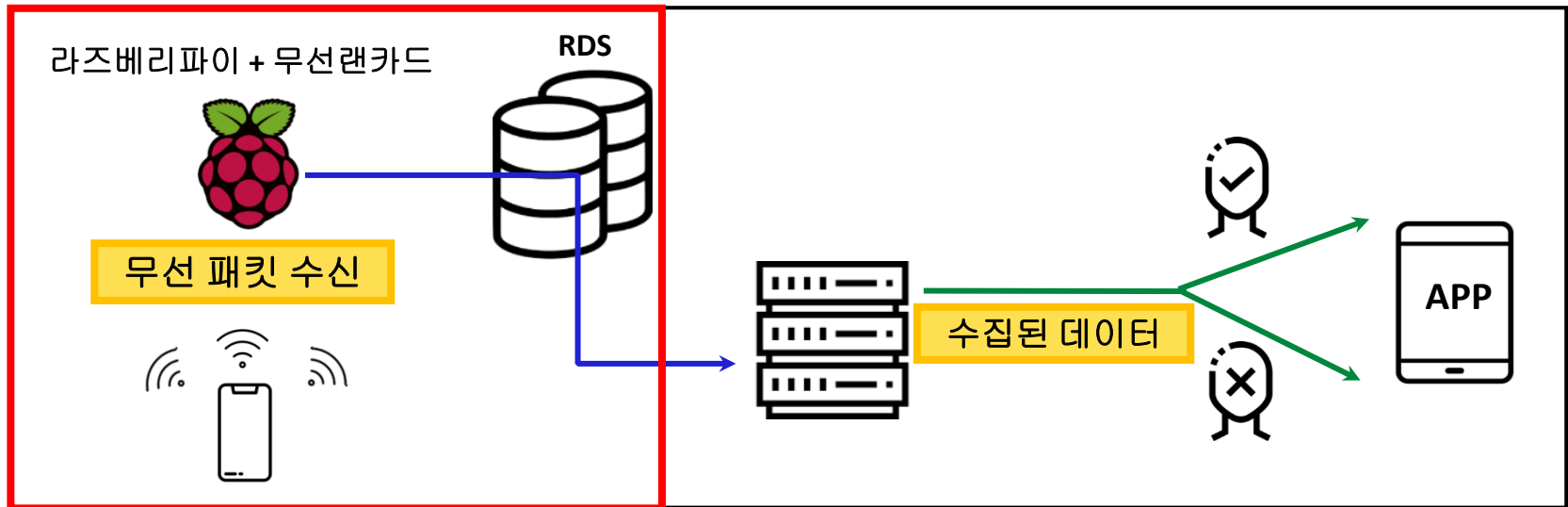
- 라즈베리파이 : 간편하게 들고 다니기 용이한 소형 컴퓨터
- RDS : AWS 데이터베이스. 클라우드로 언제 어디서든 접근 가능
- EC2 : AWS 서버. 클라우드로 언제 어디서든 접근 가능

개발 환경 및 개발 내용 (3/10)

개발 내용(2/9)

○ 전자출결 시스템 개발

- 3개의 스레드를 사용하여 프로그램 개발
- 캡처한 패킷 데이터를 파싱하여 **AWS RDS**와 연동 및 관리
- 라즈베리파이와 무선랜카드 연결



개발 환경 및 개발 내용 (4/10)

개발 내용(3/9)

○ 전송시스템 구현

```
int savedata(char* argv){
    char* dev = argv;
    int runQuery(){
        sprintf(insert_query + query_len-2, "; ");

        if(mysql_query(conn, insert_query)) {
            errorMsg("MySQL insert_Query empty");
        }
        else{
            printf("\n\n-----query_len : %d, insert qu
            printf("%s",insert_query);
            printf("\n\n-----");
        }

        d.clear();
        memset(insert_query, 0, sizeof(char));
        sprintf(insert_query, "insert INTO Log VALUES");
        query_len=22;

        return 0;
    }
}
```

패킷 저장

패킷 전송

채널 탐색

```
void ch_hopping(char* dev)
{
    while (true) {
        char cmd[32] = "iwconfig ";
        strcat(cmd, dev);
        strcat(cmd, " channel ");
        sprintf(cmd + 23, "%d", ch);
        system(cmd);
        sleep(1);
        ch += 6;
        ch %= 13;
        if(!ch) ch = 13;
    }
}
```

패킷 저장, 전송 및 채널 탐색을 위한 3개의 thread 개발

개발 환경 및 개발 내용 (5/10)

개발 내용(4/9)

○ APP 로그인 프로그램

```
if (response.status === 200){
  db.transaction( tx => {
    tx.executeSql(
      "insert into info values ( 1, ?, ? )",
      [studentCode, macAddress],
      () => {
        console.log("insert success");
      },
      () => {
        console.log("insert fail");
      }
    );
  });
  this.setState({
    studentName: responseJson.name,
    isLogin: true
  })
}
else if (response.status === 400) {
```

사용자 인증

```
def login(request):
  if request.method == 'POST':
    data = JSONParser().parse(request)
    search_grade_number = data['grade_number']
    obj = Member.objects.get(grade_number=search_grade_number)
    dictName = {"name": obj.name}
    if data['mac_address'] == obj.mac_address:
      return JsonResponse( dictName,status=200)
    else:
      return HttpResponse(status=400)
  else:
    return HttpResponse(status=400)
```

확인 후 조치

인가된 사용자를 인증, 로그인 가능토록 조치



개발 환경 및 개발 내용 (6/10)

개발 내용(5/9)

○ 기본 APP 프로그램

```
render() {
  //this.props.screenProps.code == 학번 정보
  render() {
    return (
      <View style={styles.container}>
        <View style={styles.logocontainer}>
          <Image style={styles.picturelogo} source={picturelogo} />
          <Image style={styles.textlogo} source={textlogo} />
        </View>
        <View style={styles.maincontainer}>
          <View style={styles.mainleftcontainer}>
            <View style={styles.personal}>
              <TouchableOpacity>
                <Image style={styles.imagesetting} source={personal} />
              </TouchableOpacity>
              <Text style={styles.textsettings}>
                출석체크
              </Text>
            </View>
            <View style={styles.settings}>
              <TouchableOpacity>
```

사용자 화면

관리자 화면

사용자와 관리자를 구분하여 메인스크린 적용

개발 환경 및 개발 내용 (7/10)

개발 내용(6/9)

○ 프론트엔드 APP 프로그램

```
render() {
  //this.props.navigation.state.params == 강의명&시간
  export default class SubDetail extends React.Component{
    static navigationOptions = {
      render() {
        //console.log(this.props.navigation.state.params.students);
        let studentlist = this.props.navigation.state.params.students;
        return (
          <View style={styles.container}>
            <View style={styles.sub_container}>
              <Text style={styles.sub_text}>
                {this.props.navigation.state.params.week}주차 출석 현황
              </Text>
            </View>
            <View style={styles.main_container}>
              <View style={styles.topBlock}>
                <Text style={styles.topText}>이름</Text>
                <Text style={styles.topText}>학번</Text>
              </View>
            </View>
          </View>
        );
      }
    }
  }
}
```

강의명 및 시간 정보

사용자 출결 현황

수강생 출결 현황 확인

사용자와 관리자 모두 사용자의 출결 현황 확인 및 수정 가능

개발 환경 및 개발 내용 (8/10)

개발 내용(7/9)

○ DB 구축 및 장고 연동

```
CREATE DATABASE `Project` CHARACTER SET utf8 COLLATE
```

DB 구축

```
CREATE TABLE `Pr
```

```
`student_id` CHA
```

```
`name` VARCHAR(5
```

```
`grade` TINYINT(
```

```
`department` VAR
```

```
PRIMARY KEY (`nu
```

```
ENGINE = InnoDB
```

```
CREATE TABLE `Pr
```

```
`mac_addr` CHAR(
```

```
`device_kind` VA
```

```
`Member_student_
```

```
PRIMARY KEY (`nu
```

```
ENGINE = InnoDB
```

```
CREATE TABLE `Pr
```

```
`lecture_name` V
```

```
`lecture_room` C
```

```
from django.db import models
```

Django 연동

```
class Log(models.Model):
```

```
    num = models.AutoField(primary_key=True)
```

```
    mac_addr = models.CharField(max_length=12)
```

```
    lecture_room = models.CharField(max_length=7)
```

```
    pwr = models.IntegerField()
```

```
    count = models.IntegerField()
```

```
    time = models.DateTimeField()
```

```
class Meta:
```

```
    managed = False
```

```
    db_table = 'Log'
```

```
class Student(models.Model):
```

```
    num = models.AutoField(primary_key=True)
```

DB 구축

장고 연동

```
mysql> use Project
```

```
Database changed
```

```
mysql> use Project
```

```
Database changed
```

```
mysql> show tables;
```

```
Table
```

```
Log
```

```
Student
```

```
Subject
```

```
Subject_time
```

```
auth_group
```

```
auth_group_permissions
```

```
auth_permission
```

```
auth_user
```

```
auth_user_groups
```

```
auth_user_user_permissions
```

```
django_admin_log
```

```
django_content_type
```

```
django_migrations
```

```
django_session
```

```
student
```

Django와 RDS 연동 및 DB 테이블 생성

개발 환경 및 개발 내용 (9/10)

개발 내용(8/9)

○ 데이터를 통한 사용자의 출결 여부 판단

- 수업참여 시간 20분 이내는 결석, 20분 이후 45분 이내는 지각으로 기록.

분류	횟수
출석	9번 이상
지각	4번 이상
결석	3번 이하

```
timeSentence1 = timeObj.year + "-" + arrayDay[i]
timeSentence2 = timeObj.year + "-" + arrayDay[i] + " " + timeObj.finish_time + ":00.000000+00:00"

def logtest(request):
    data = JSONParser().parse(request)
    grade_number = data['grade_number']
    lecture_name = data['lecture_name']
    lecObj = Subject.objects.get(lecture_name=lecture_name)
    memObj = Member.objects.get(grade_number=grade_number)
    lecture_id = lecObj.lecture_id
    lecture_room = lecObj.lecture_room
    mac_address = memObj.mac_address
    subject_num_id = lecObj.id
    timeObj = Subject_time.objects.get(Subject_num_id = subject_num_id)

    #array 선택
    if timeObj.day=='09-07':
        arrayDay = day1
    elif timeObj.day=='09-08':
        arrayDay = day2
    elif timeObj.day=='09-09':
        arrayDay = day3
    elif timeObj.day=='09-10':
        arrayDay = day4
```

수집된 시간대 정보 확인

출결 여부 결정

50분 기준, 최대 10번의 신호를 통해 출석, 지각, 결석을 판단

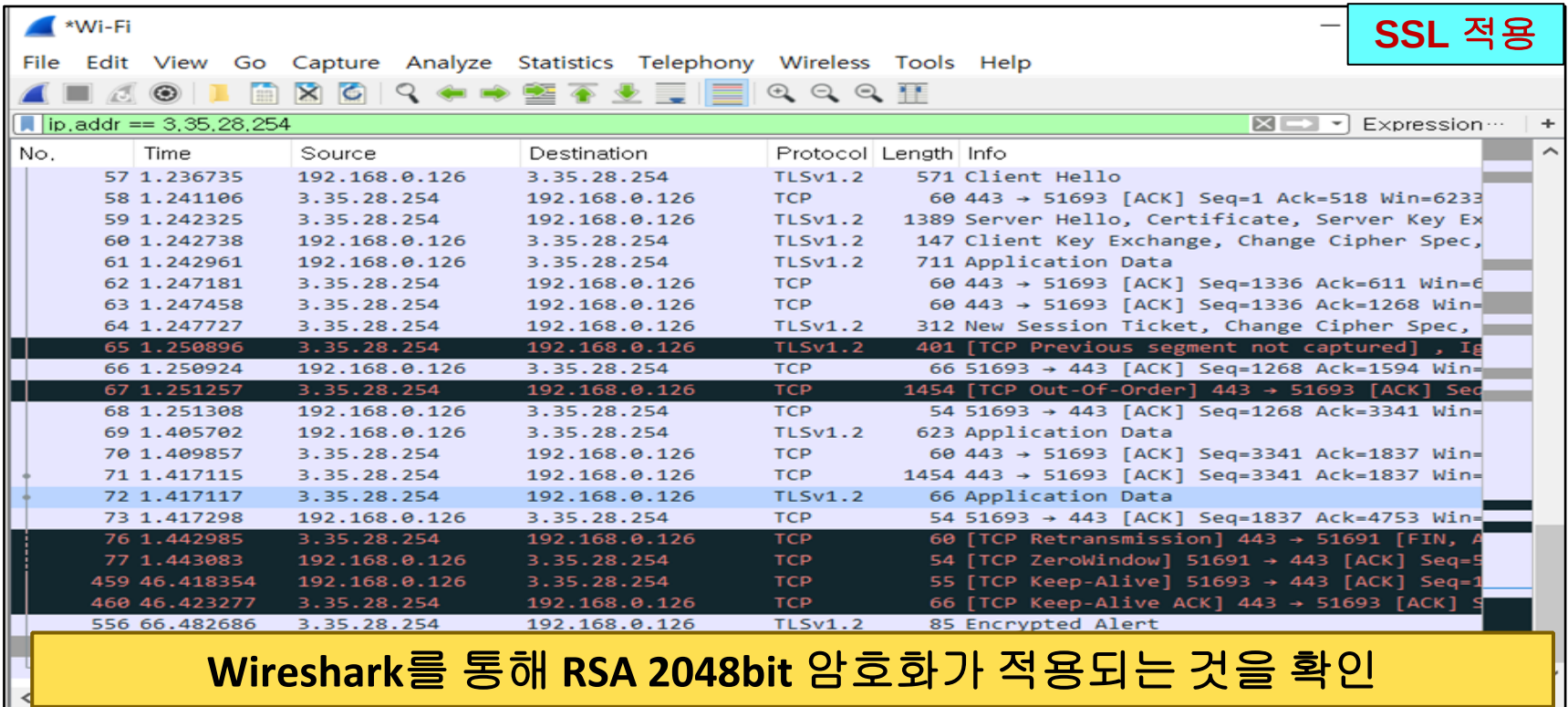
개발 환경 및 개발 내용 (10/10)

개발 내용(9/9)

○ 안전한 통신을 위한 서버 SSL 적용

- Nginx 웹 서버에 OpenSSL을 활용한 통신 암호화 기법 적용.

SSL 적용



No.	Time	Source	Destination	Protocol	Length	Info
57	1.236735	192.168.0.126	3.35.28.254	TLSv1.2	571	Client Hello
58	1.241106	3.35.28.254	192.168.0.126	TCP	60	443 → 51693 [ACK] Seq=1 Ack=518 Win=6233
59	1.242325	3.35.28.254	192.168.0.126	TLSv1.2	1389	Server Hello, Certificate, Server Key Ex
60	1.242738	192.168.0.126	3.35.28.254	TLSv1.2	147	Client Key Exchange, Change Cipher Spec,
61	1.242961	192.168.0.126	3.35.28.254	TLSv1.2	711	Application Data
62	1.247181	3.35.28.254	192.168.0.126	TCP	60	443 → 51693 [ACK] Seq=1336 Ack=611 Win=6
63	1.247458	3.35.28.254	192.168.0.126	TCP	60	443 → 51693 [ACK] Seq=1336 Ack=1268 Win=
64	1.247727	3.35.28.254	192.168.0.126	TLSv1.2	312	New Session Ticket, Change Cipher Spec,
65	1.250896	3.35.28.254	192.168.0.126	TLSv1.2	401	[TCP Previous segment not captured] , Ig
66	1.250924	192.168.0.126	3.35.28.254	TCP	66	51693 → 443 [ACK] Seq=1268 Ack=1594 Win=
67	1.251257	3.35.28.254	192.168.0.126	TCP	1454	[TCP Out-Of-Order] 443 → 51693 [ACK] Seq=
68	1.251308	192.168.0.126	3.35.28.254	TCP	54	51693 → 443 [ACK] Seq=1268 Ack=3341 Win=
69	1.405702	192.168.0.126	3.35.28.254	TLSv1.2	623	Application Data
70	1.409857	3.35.28.254	192.168.0.126	TCP	60	443 → 51693 [ACK] Seq=3341 Ack=1837 Win=
71	1.417115	3.35.28.254	192.168.0.126	TCP	1454	443 → 51693 [ACK] Seq=3341 Ack=1837 Win=
72	1.417117	3.35.28.254	192.168.0.126	TLSv1.2	66	Application Data
73	1.417298	192.168.0.126	3.35.28.254	TCP	54	51693 → 443 [ACK] Seq=1837 Ack=4753 Win=
76	1.442985	3.35.28.254	192.168.0.126	TCP	60	[TCP Retransmission] 443 → 51691 [FIN, A
77	1.443083	192.168.0.126	3.35.28.254	TCP	54	[TCP ZeroWindow] 51691 → 443 [ACK] Seq=5
459	46.418354	192.168.0.126	3.35.28.254	TCP	55	[TCP Keep-Alive] 51693 → 443 [ACK] Seq=1
460	46.423277	3.35.28.254	192.168.0.126	TCP	66	[TCP Keep-Alive ACK] 443 → 51693 [ACK] S
556	66.482686	3.35.28.254	192.168.0.126	TLSv1.2	85	Encrypted Alert

Wireshark를 통해 RSA 2048bit 암호화가 적용되는 것을 확인



개발 시스템 운영 (1/6)

사용자 등록

○ 웹 페이지를 통한 사용자 정보 입력

Pease sign in

사용자 정보 등록

이름

홍길동

학년

1학년 ▼

학과

정보보호학과 ▼

학번

91512345

맥주소

A82BB9E2EFC8

☐ [개인정보 처리방침](#)을 읽어보았으며, 이에 동의합니다.

웹페이지를 통하여 사용자 정보를 등록

개발 시스템 운영 (2/6)

사용자 관리

○ Django Admin 페이지를 통한 사용자 관리

홈 > Memberauth > Members

수강생 출결 현황 확인

변경할 member 선택

액션:

▼

실행

4 중 아무것도 선택되지 않았습니다.

☐ MEMBER

☐ 정보보호학과 - 4학년 / 홍길동

☐ 정보보호학과 - 4학년 / 진영훈

☐ 정보보호학과 - 1학년 / 테스트

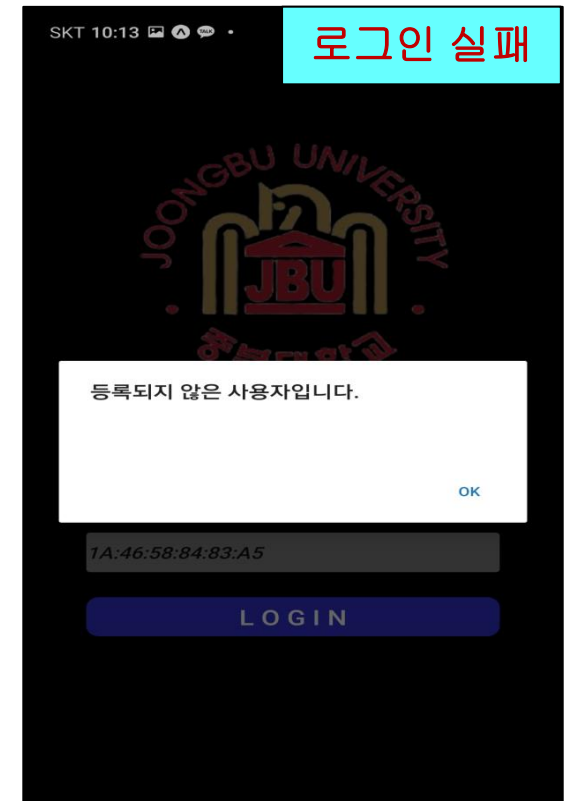
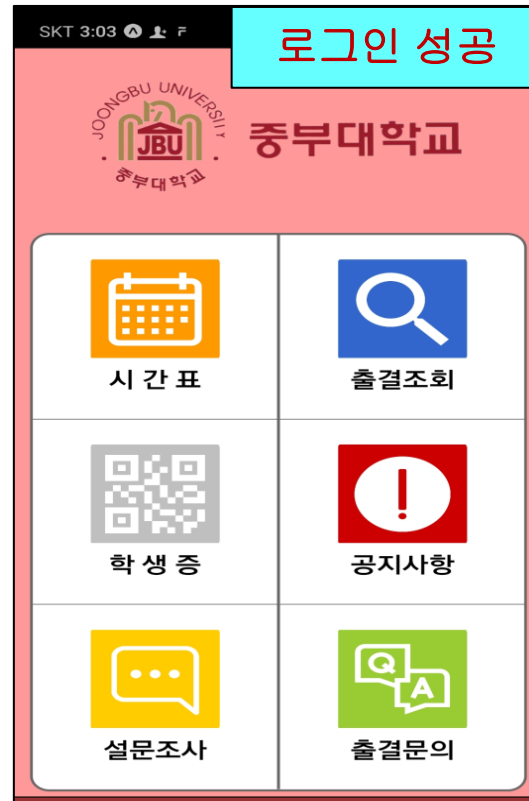
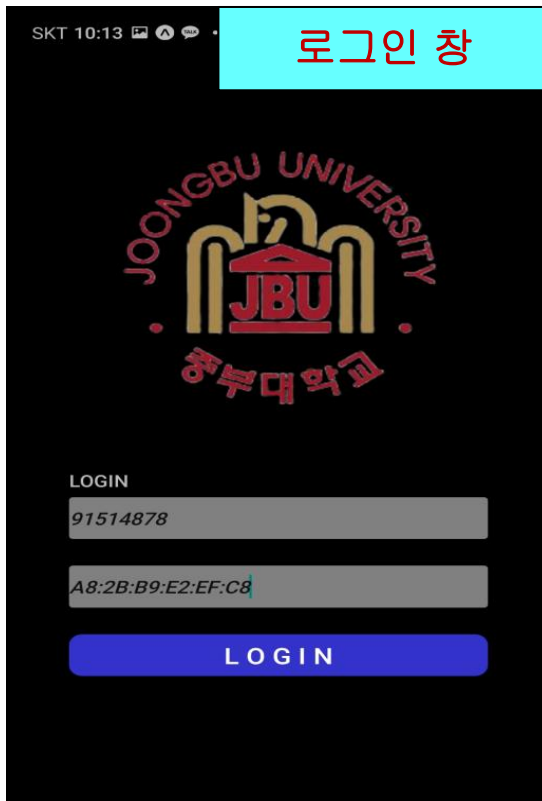
☐ 정보보호학과 - 4학년 / 이슬기

Django 관리 페이지를 통해서 학생들의 MAC주소 등의 정보를 관리

개발 시스템 운영 (3/6)

로그인 화면

○ 프론트엔드 로그인 체제

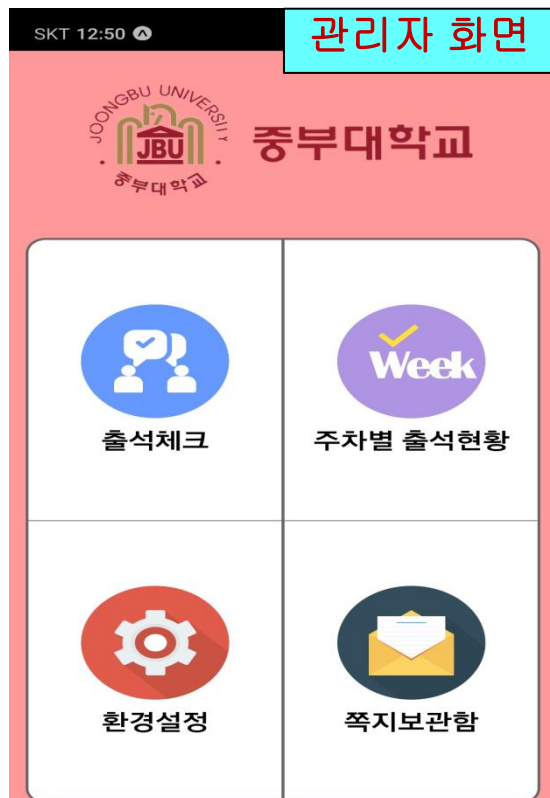


로그인 시 학번 및 MAC주소 입력, 성공과 실패 화면 분리

개발 시스템 운영 (4/6)

홈 화면

○ 사용자와 관리자 분리



사용자와 관리자 화면을 분리하여 각각의 역할을 수행

개발 시스템 운영 (5/6)

출결 현황 확인

○ 시간표 및 출결여부 상세

강의명/시간정보



출결조회

월요일

정보보호프로젝트설계실습1(캡스톤디자인)
13:10~14:00

정보보호프로젝트설계실습1(캡스톤디자인)
14:10~15:00

화요일

수요일

취창업 성공전략
13:10~14:00

목요일

금요일

블록체인기술및응용
13:10~14:00

정보보호법과윤리
16:10~17:00

수강생 출결여부 상세



출결현황 상세

정보보호프로젝트설계실습1(캡스톤디자인)

테스트 출석상세

출석: 1 결석: 15

주차	날짜	시간	출결
1	9월7일	13:10 ~ 14:00	출석
2	9월14일	13:10 ~ 14:00	결석
3	9월21일	13:10 ~ 14:00	결석
4	9월28일		

수강생 출석여부 목록



주차별 출석현황

01주차 출석 현황

이름	학번	출석 여부	상세	변경
이슬기	91514878	출석 여부	상세	변경
윤성준	91514866	출석 여부	상세	변경
테스트	1234	출석 여부	상세	변경

시간표 및 출결여부 상세 화면을 통한 수강생의 출결 여부 확인

개발 시스템 운영 (6/6)

출결 변경 요청

○ 관리자의 수강생 출석정보 변경

출결 정보 변경



출결 정보 변경 전



출결 정보 변경 후



오탐, 미탐의 경우를 위해 요청 시 수동으로 출석 정보를 변경 가능



결론 및 기대 효과

○ 결 론

- 자체 기술력으로 전자출결 시스템을 구현하고, 이를 기반으로 자동 출석 체크 시스템 완성하는데 성공
- 교수 - 학생간 수동형 출석체크에서 능동형 출석체크가 가능한 시스템을 구현

○ 기대 효과

- 기존 중부대 출석체크 앱 등의 한계를 보완하고 기능을 개선하여 유용성 극대화가 가능
- 조원들에게 임무를 적절히 분담하여 필요 기술을 직접 구현하고 팀워크로 연구하는 조직체제를 가동, 기술 역량을 배가하는 계기 마련



Q & A
