



PORTFOLIO  
정민지  
(MINJI JUNG)



소개



FRONT & BACKEND WEB DEVELOPER  
정 민 지  
(MINJI JUNG)

● **기본정보**

- 생년월일 : 1990.07.23
- 나이 : 30세
- 이메일 : minji6654@gmail.com
- 연락처 : 010-3857-8956
- 주소 : 서울시 동대문구 왕산로 122, 포스빌 1612호

● **학력 / 교육**

- 인하대학교 항공우주공학과 학사 졸업 (2014.02)
- KH 정보교육원 자바 Framework 개발자 양성과정 수료 (2019.12 ~ 2019.07)

● **경력**

- 현대엔지니어링 화공플랜트사업본부 프로젝트관리팀 Project Engineer (2014.01 ~ 2018.06)

● **자격증 / 어학**

- SQL Developer (2019.07)
- 정보처리기사 (2019.05)
- TOEIC 975점 (2018.11)
- OPIc IH (2018.12)



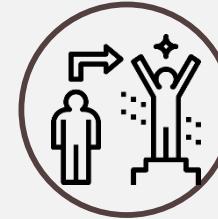
목표



신입 자바 개발자



다양한 프로젝트 경험



Full-Stack 개발자



사용언어/기술



## FRONT-END

HTML



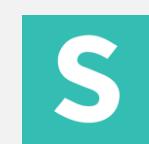
CSS



jQuery

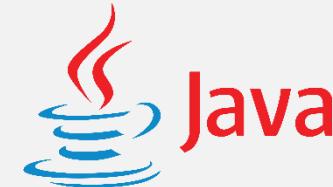


Bootstrap



Google Maps APIs

## BACK-END



JDBC  
Java Database Connectivity



ORACLE  
D A T A B A S E



spring



socket.io



## TOOLS

# 개발 프로젝트 - 1

# 여행일정 작성 및 공유 웹 애플리케이션



Travel Interface screenshot showing a travel itinerary for Sydney/Melbourne. The left panel displays a daily schedule for July 22nd, and the right panel shows a map of Melbourne with numbered pins indicating travel points.

Github : <https://github.com/Julia8956/travellInterface>

## 프로젝트 명 : TRAVEL INTERFACE

- 수행 기간 : 2019.06.20 ~ 2019.07.17 (4주)
- 개발 목표 : 여행관련 일정과 정보를 한 사이트내에서 제공해 누구나 쉽게 여행 계획을 세우고 정보획득 및 수익 창출을 기대 할 수 있는 서비스를 제공하는 웹 사이트 구축

## 사용 기술 및 개발 환경

- Language : JAVA 8, HTML5, CSS3, JavaScript, JSP 2.5, Servlet 3.1
- Framework : Spring, Mybatis
- Database : oracle 11.2.0, MongoDB 3.2.1
- WAS : Apache Tomcat 8.5.40, node.js
- Library : JQuery, ojdbc8, Java Mail, Apache POI, commons io, Jackson, Spring security, aspectJ, Socket.io
- API : Google Maps, Google Places, Quill, KaKao Login, 아임포트, Fixer
- 형상관리 : github



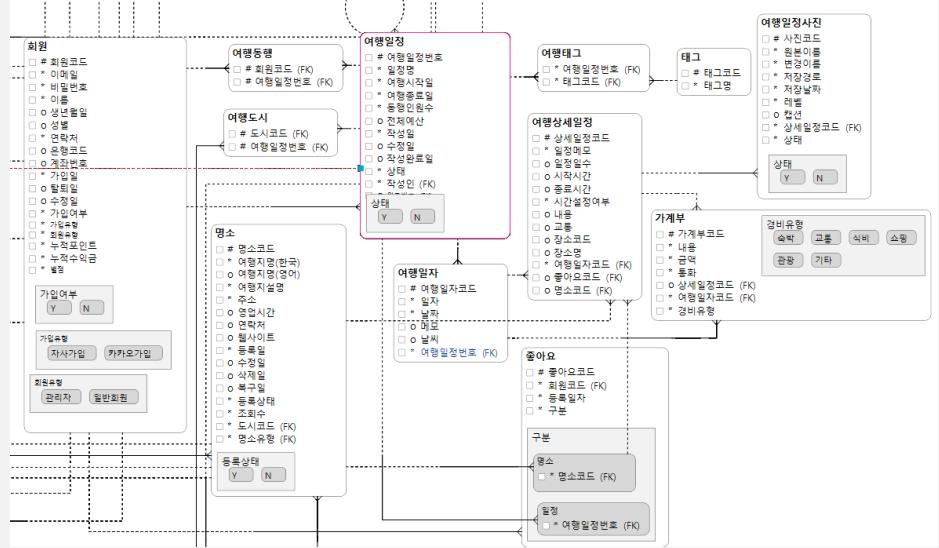
# 개발 프로젝트 - 1



# 여행일정 작성 및 공유 웹 애플리케이션



## DB 모델링



## 공통 역할

- 프로젝트 기획 : 주제 선정 및 기획, 기능 정의서 작성
- UI설계 : 프로토타입 제작(카카오 오픈), Usecase Diagram, 스토리보드 작성
- DB모델링 : 논리 모델링, 물리 모델링
- 클래스 설계 : 클래스 다이어그램, 시퀀스 다이어그램 작성

## 담당 구현 기능

- 여행 일정 만들기/수정/공유/삭제
- Google Maps API를 이용한 지도에서 장소 타입 별 검색으로 위치 및 상세정보 조회
- 지도를 연동한 상세 일정 작성/수정/삭제, Jquery sortable 기능을 이용한 드래그로 일정 편집
- 좋아요 추가한 장소와 추천 장소 조회 및 상세 일정 작성 연결
- 일자 별 여행 루트 지도에서 조회
- 여행 태그 추가/수정/삭제
- Quill Editor API를 이용하여 여행 후기 글 등록(사진 업로드) 및 수정, 여행사진 갤러리 관리, 대표 사진 변경
- 여행 예산 입력 및 Fixer API를 사용하여 실시간 환율 조회, 현지화 환산, 경비 탑입 별 지출 합계 조회
- Apache POI를 이용하여 여행 일정 및 가계부 엑셀 다운로드
- Node.js와 socket.io를 이용, 여행 일정 공유를 위한 서버 구축
- 여행 동행 초대/삭제 및 실시간 일정 공유하여 함께 작성 기능
- 구매한 일정으로 나의 새 일정 작성하기(일정 오버라이딩) 기능



## 구현 기능 - 1



# 여행일정 작성 및 공유 웹 애플리케이션



Melba Restaurant

+ 일정에 추가

3/1 Southgate Ave, Southbank VIC 3006 오스트레일리아

+61 1800 641 107

★★★★★ 4.2

월요일: 오전 6:30 ~ 오후 11:00, 화요일: 오전 6:30 ~ 오후 11:00, 수요일: 오전 6:30 ~ 오후 11:00, 목요일: 오전 6:30 ~ 오후 11:00, 금요일: 오전 6:30 ~ 오후 11:00, 토요일: 오전 6:30 ~ 오후 11:00, 일요일: 오전 6:30 ~ 오후 11:00

**지도 탑입 검색  
상세 검색**

Google Maps  
Google Places

원하는 장소 탑입을 선택 시 현재지도에서 장소리스트를 조회

☞ Google Maps, Places API를 이용하여, service 객체의 textSearch 메소드를 호출하여 장소 탑입에 맞는 marker를 그리고, place\_id 정보를 조회해와서 이를 getDetail 메소드로 전달하여 구글에서 제공하는 place detail 정보를 받아와 지도에 위치를 표시하고 상세정보를 div에 구성

```

typeMarkers.forEach(function(marker) {
    marker.setMap(null);
});
typeMarkers = [];
var request = {
    location: map.getCenter(),
    radius: '1000',
    type: [placeType]
};
service.textSearch(request, function(results, status, pagination) {
    var bounds = new google.maps.LatLngBounds();
    var placeList = document.getElementById('placeList');
    if (status === google.maps.places.PlacesServiceStatus.OK) {
        for (var i = 0; i < results.length; i++) {
            var li = document.createElement('li');
            li.innerHTML += '<input type="hidden" name="placeId" value=""' +
                + results[i].place_id + '" />' + '<p><strong>' +
                + results[i].name + '<small style="color:purple"> review:' +
                + results[i].rating + '/5.0 </small></strong></p><p class="help">' +
                + results[i].formatted_address + '</p>';
            placeList.appendChild(li);
            bounds.extend(results[i].geometry.location);
            addMarker({
                location:results[i].geometry.location,
                icon:icons[placeType].icon,
                map_icon_label:icons[placeType].map_icon_label,
                place_id:results[i].place_id,
                info: '<h5 class="title is-5">' + results[i].name + "'</h5>" +
                    + results[i].formatted_address
            });
        }
        map.fitBounds(bounds);
        moreBtn.disabled = !pagination.hasNextPage;
        getNextPage = pagination.hasNextPage && function() {
            pagination.nextPage();
        }
        map.setCenter(results[0].geometry.location);
    }else return;
}

```

## 구현 기능 - 2

# 여행일정 작성 및 공유 웹 애플리케이션



### 시간 정보를 입력하여 일정 insert

☞ 해당 일자의 기존 일정 리스트의 시작시간을 for문으로 각각 조회해와서 시간차가 가장 적게 존재하는 일정을 찾아 해당 일정의 뒤에 새로운 일정이 들어갈 수 있도록 일정의 순서 컬럼을 재설정하여 일정이 시간순서대로 재배치 될 수 있도록 비즈니스로직 작업.

```

else {
    sch.setIsTimeset("Y");
    if(count > 0) {
        ArrayList<TrvSchedule> schList = td.selectSchList(sqlSession, sch.getDayId());
        int number = selectSchNumber(schList, sch.getStartTime());
        sch.setSchNumber(number);
        System.out.println("number : " + number);
    } else {
        sch.setSchNumber(count + 1);
    }
}

@Override
public int selectSchNumber(ArrayList<TrvSchedule> schList, String startTime) {
    int difference = 1440;
    int number = 0;
    int hour = Integer.parseInt(startTime.substring(0, startTime.indexOf(":")));
    int minute = Integer.parseInt(startTime.substring(startTime.indexOf(":") + 1));

    for(int i = 0; i < schList.size(); i++) {
        String start = schList.get(i).getStartTime();
        System.out.println(start);
        if(start != null) {
            int hr = Integer.parseInt(start.substring(0, start.indexOf(":")));
            int min = Integer.parseInt(start.substring(start.indexOf(":") + 1));
            int diff = (hour * 60 + minute) - (hr * 60 + min);
            if(diff > 0 && diff < difference) {
                difference = diff;
                number = schList.get(i).getSchNumber() + 1;
            }
        }
    }

    if(number == 0) number = 1;

    for(int i = 0; i < schList.size(); i++) {
        if(schList.get(i).getSchNumber() >= number) {
            schList.get(i).setSchNumber(schList.get(i).getSchNumber() + 1);
            int result = td.updateSchNumber(sqlSession, schList.get(i));
        }
    }
}

return number;
}

```



## 구현 기능 - 3



# 여행일정 작성 및 공유 웹 애플리케이션



2018-09-12

**DAY 3** MEMO

- 09:00 - 12:00 퍼핑빌리투어 (X) (edit)
- 관광: 40.0 (AUD) / 투어차량  
▶ 블그레이브
- 12:00 - 13:00 카페거리에서 점심 (X) (edit)
- 식비: 30.0 (AUD) /  
▶ Seedling Cafe (Little Collins Corner Equitable place)
- 13:00 - 15:00 필립 아일랜드 투어 (X) (edit)
- 관광: 50.0 (AUD) / 투어차량  
▶ 필립 섬
- 17:00 - 19:00 유레카타워 전망대 야경구경~! (X) (edit)
- 관광: 50.0 (AUD) /  
▶ 유레카 타워

일정 추가

2018-09-13

**DAY 4** MEMO

- 12:00 - 13:00 점심먹기 (X) (edit)
- 식비: 30.0 (AUD) /  
▶ Melba Restaurant
- 13:00 - 13:30 공원산책! (X) (edit)
- 기타: 5.0 (AUD) / 트램타고 가기  
▶ 알렉산드리아 가든스
- 13:30 - 15:00 내셔널갤러리 구경! (X) (edit)
- 관광: 15.0 (AUD) /  
▶ 내셔널 갤러리 오브 빅토리아
- 크라운 카지노구경~!! (X) (edit)
- 관광: 50.0 (AUD) /  
▶ Crown Melbourne

일정 추가

여행 일정 편집을 쉽게 도와주기 위해 드래그로 원하는 일정을 원하는 위치에 끌어서 이동하는 기능

- ☞ Jquery Sortable 기능을 이용하여 같은 날짜 안에서 이동이 발생했는지, 다른 날짜로 일정이 이동 되었는지를 확인하고, 일정 리스트를 시간 순서에 맞게 정렬하기 위해 해당 일자의 상세 일정들의 schNumber 컬럼값(일정순서)을 모두 update.

상세 일정 update

JQuery Sortable

```

$(".dayList").sortable({
  connectWith:".dayList",
  start:function(event, ui) {
    startDayId = $(this).parent().find("input[name=dayId]").val();
    start = $(this).sortable("serialize", { key: "sch" });
    startArr = $(this).sortable("toArray");
  },
  update:function(event, ui) {
    var ul = $(this);
    var dayNumber = ul.attr("id").charAt(3);
    var item = ui.item.attr("id");
    var schId = item.substring(item.indexOf("_") + 1);
    var changeDayId = $(this).parent().find("input[name=dayId]").val();
    var change = $(this).sortable("serialize", { key: "sch" });
    var changeArr = $(this).sortable("toArray");

    if(ui.sender == null) {
      if(startArr.length == changeArr.length) { //1.같은 day에서 수정할때
        $.ajax({
          url:"updateSchNumber.trv?" + change,
          type:"post",
          data:{dayId:changeDayId},
          success:function(data) {
            updateSchList(dayNumber, data.updList);
            socket.emit('reorderSchListInADay', {
              dayNumber:dayNumber,
              schList:data.updList,
              room:"${ trv.trvId }"
            });
          },
          error:function(data) {
            alert('updateSchNumber 서버전송 실패');
          }
        });
      }
    }
  }
});

```



여행!!!

함께 여행하는 동행을 초대해서 일정을 공유하고 같이 작성해보세요!

user17@gmail.com

user17@gmail.com(정호석) 동행 추가

user18@gmail.com (박지민)	삭제
user19@gmail.com (김태형)	삭제
user20@gmail.com (전정국)	삭제

닫기

여행을 함께 가는 동행을 초대해서 일정 작성페이지를 공유하고 실시간으로 일정을 함께 편집할 수 있는 기능

☞ node.js를 이용하여 travel server를 따로 구축, socket.io 라 이브러리를 사용하여 클라이언트와 서버를 소켓으로 연결하여 일정 작성 event를 실시간으로 주고받음으로써 업데이트.

### 동행 초대 & 실시간 일정 공유

Node.js & Socket.io

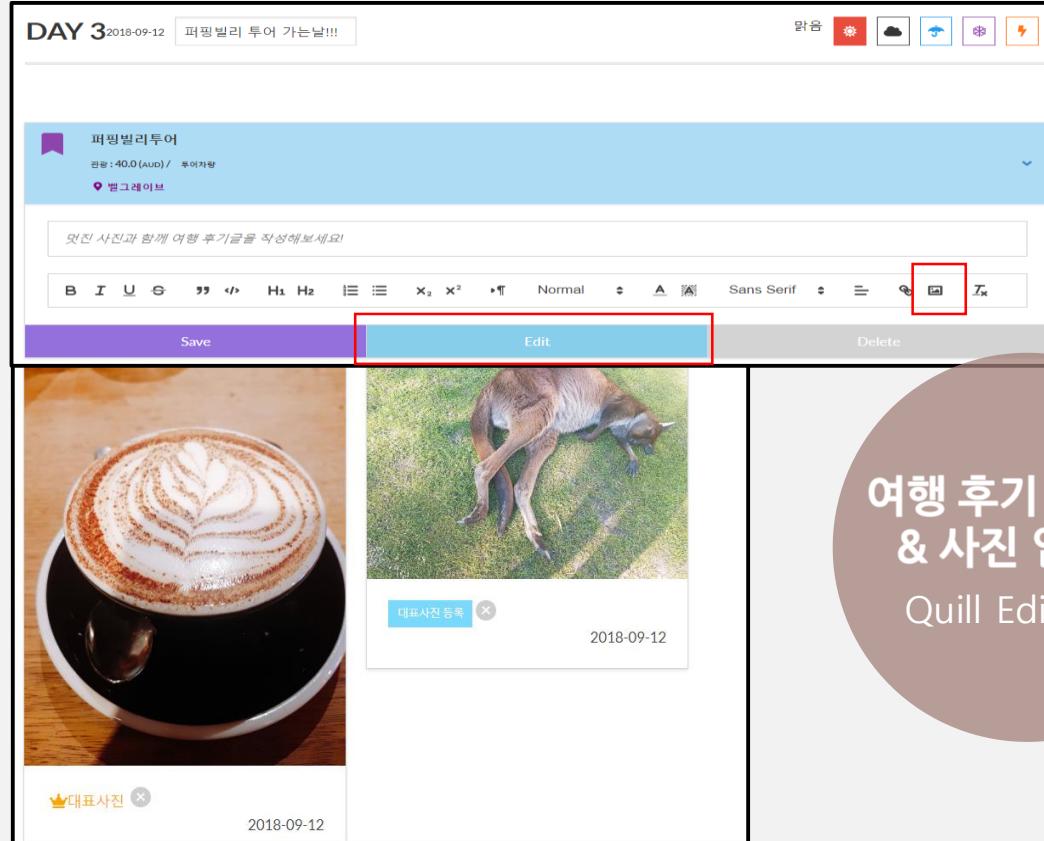
```
const express = require('express');
const app = express();
const http = require('http').createServer(app);
const io = require('socket.io')(http);
const { addMember, removeMember, getMember, getMembersInRoom } = require('./utils/memb
...
io.on('connection', (socket) => {
  //member in
  socket.on('memberIn', ({ name, email, room }) => {
    console.log(name + " user connected");
    const { error, member } = addMember({ id:socket.id, name, email, room });
    socket.join(member.room);
    io.to(member.room).emit('updateMemberList', getMembersInRoom(room));
  });
  //member disconnected
  socket.on('disconnect', () => {
    const member = removeMember(socket.id);
    console.log(member);
    if(member) {
      console.log(member.name + ' user disconnected');
      //delete(members[socket.id]);
      io.to(member.room).emit('updateMemberList', getMembersInRoom(member.room));
    }
  });
  const members = [];
  const addMember = ({ id, name, email, room }) => [
    const member = { id, name, email, room }
    members.push(member)
    return { member }
  ];
});
```



## 구현 기능 - 5



# 여행일정 작성 및 공유 웹 애플리케이션



### 여행 후기 글 작성 & 사진 업로드

Quill Editor API

여행 상세 일정에 editor를 이용하여 후기 글을 작성하고 사진을 업로드를 하여 여행 갤러리를 만들 수 있는 기능

☞ Quill Editor API를 각 상세일정에 추가하고 사진 업로드 버튼에 handler를 추가함으로써 사진 업로드시 serve에 파일을 저장하고 editor에 해당 파일을 imbed

```
quill.getModule('toolbar').addHandler('image', function() {
  console.log("handler작동");
  var fileInput = document.createElement('input');
  fileInput.setAttribute('type', 'file');
  fileInput.setAttribute('name', 'image');
  fileInput.click();

  fileInput.onchange = function() {
    var fd = new FormData();
    var file = $(this)[0].files[0];
    fd.append('image', file);
    fd.append('schId', schId);
    $.ajax({
      url:"insertSchFile.trv",
      type:"POST",
      data:fd,
      cache:false,
      processData:false,
      contentType:false,
      success:function(data) {
        console.log('ajax success called');
        var range = quill.getSelection();
        quill.insertEmbed(range.index, 'image',
          'http://localhost:8001/ti/resources/uploadFiles/' + data.changeName);
        addGallaryImage(dayNumber, data.changeName, data.fileLevel, data fileId);
        socket.emit('addGallaryImage', {
          @RequestMapping("insertSchFile.trv")
          public ModelAndView insertSchFile(HttpServletRequest request, ModelAndView mv,
          @RequestParam(name = "image", required = false) MultipartFile image, int schId) {
            String root = request.getSession().getServletContext().getRealPath("resources");
            String filePath = root + "\\uploadFiles";
            String originName = image.getOriginalFilename();
            String ext = originName.substring(originName.lastIndexOf("."));
            String changeName = CommonUtils.getRandomString();
            try {
              image.transferTo(new File(filePath + "\\" + changeName + ext));
            }
          }
        });
      }
    });
  }
});
```

# 구현 기능 - 6

# 여행일정 작성 및 공유 웹 애플리케이션



The screenshot shows a travel budget application interface. At the top, there's a summary table with currency conversion rates:

전체 예산		TOTAL	
		830.009 KRW = 1 AUD	TOTAL
800,000 원		=	963.84 AUD

Below this is a breakdown of expenses by category:

숙박	교통	식비	쇼핑	관광	기타	TOTAL	BALANCE
124,501 원	41,500 원	74,700 원	83,000 원	211,652 원	4,150 원	539,503 원	260,497 원
150.00 AUD	50.00 AUD	90.00 AUD	100.00 AUD	255.00 AUD	5.00 AUD	650.00 AUD	313.84 AUD

At the bottom, there's a timeline view for four days (2018-09-10 to 2018-09-13) with a modal overlay showing a transaction entry form for "가계부 항목" (Family Budget Item) on DAY 2.

여행 예산을 실시간 환율을 적용하여 현지화로 환산, 가계부를 입력/수정 시 가계부 항목을 추가하고 경비 타입 별 지출 summary 테이블을 update하여 지출을 관리할 수 있는 기능

☞ 예산 변경 시 ajax로 예산 정보를 db에 update 후, fixer API 이용하여 환율 정보 조회 후 현지화로 환산, balance 수정

## 가계부 환율 계산, 가계부 수정

Fixer API

```

if($("#this").attr("id") === "budgetWon") {
    var value = $(this).val();
    var budgetStr = value.substring(0, value.length - 2).replace(/\,/g, "");
    budgetWon = parseFloat(budgetStr);
    //값이 입력안됐을때
    if(budgetStr === "") {
        $(this).val(0);
        formatCurrency($(this), "budgetWon");
        return;
    }
    //값이 입력됐을 때
    $.ajax({
        url:"updateBudget.trv",
        type:"POST",
        data:{budget:budgetWon, trvId:"${ trv.trvId }"},
        success:function(data) {
            getLocalBudget(budgetWon);
            socket.emit('updateBudget', {
                budget:budgetWon,
                room:"${ trv.trvId }"
            });
        },
        function getLocalBudget(amount) {
            $.ajax({
                url:'http://data.fixer.io/api/latest?access_key=bcc03ec033fd62',
                dataType:'jsonp',
                success:function(json) {
                    var krw = json.rates.KRW;
                    var local = json.rates.${ trvCityList[0].currencyUnit };
                    rate = Math.round((krw / local) * 1000) / 1000;
                    $("#exchangeRate").text(rate);
                    budgetLocal = Math.round((amount / rate) * 100) / 100;
                    $("#budgetLocal").val(budgetLocal);
                    formatCurrency($("#budgetLocal"), "budgetLocal", "blur");
                    updateSummary();
                },
            });
        }
    });
}

```



## 구현 기능 - 7



# 여행일정 작성 및 공유 웹 애플리케이션



			일정 엑셀로 다운받기
2018-09-12	2018-09-13	2018-09-14	
DAY 3	MEMO	MEMO	
퍼핀빌리투어 09:00 관광: 40.0 - (AUD) / 투어 12:00 차량 - 벨그레이브	공원산책! 13:00 기타: 5.0 (AUD) / 트램 - 타고 가기 13:30 알렉산드라 가든스	시드니로 이동~! (장소 정보 없음)	
카페거리에서 점심 식비: 30.0 12:00 (AUD) / - Seedling Cafe (Little)	내셔널갤러리 구경! 13:30 관광: 15.0 (AUD) / 15:00 내셔널 갤러리 오브 빅토리		

일정, 가계부  
엑셀 다운

Apache POI Library

여행일정과 가계부를 엑셀로 다운 받아 따로 관리할 수 있는 기능  
 ↵ apache POI 라이브러리를 사용하여 workbook 객체를 생성하고 db에서 원하는 정보를 HashMap으로 조회해와서 for문으로 row를 생성하여 값을 대입, 스타일 적용

```

@Override
public void selectDownloadSch(int trvId, HttpServletRequest request, HttpServletResponse response) {
    SXSSFWorkbook wb = new SXSSFWorkbook();
    SXSSFSheet sheet = wb.createSheet("schedule");
    for(int i = 0; i <= 7; i++) {
        sheet.trackColumnForAutoSizing(i);
    }

    OutputStream fileOut = response.getOutputStream();
    ArrayList<HashMap> schList = td.selectDownloadSch(sqlSession, trvId);
    Map<String, CellStyle> styles = CommonUtils.createStyles(wb);
    |
    Row title = sheet.createRow(0);
    title.createCell(0).setCellValue("TITLE : ");
    Cell titleCell = title.createCell(1);
    titleCell.setCellValue(schList.get(0).get("trvTitle").toString());
    titleCell.setCellStyle(styles.get("cell_b"));
    title.createCell(5).setCellValue("AUTHOR : ");
    title.createCell(6).setCellValue(schList.get(0).get("userName").toString());
    Row header = sheet.createRow(2);
    header.setHeightInPoints(20f);

    for (int i = 1; i <= 7; i++) {
        Cell cell = header.createCell(i);
        switch(i) {
            case 1: cell.setCellValue("DAY");break;
            case 2: cell.setCellValue("날짜");break;
            case 3: cell.setCellValue("시작시간");break;
            case 4: cell.setCellValue("종료시간");break;
            case 5: cell.setCellValue("일정제목");break;
            case 6: cell.setCellValue("장소명");break;
            case 7: cell.setCellValue("교통정보");break;
        }
        cell.setCellStyle(styles.get("header"));
    }

    for(int i = 0; i <= 7; i++) {
        sheet.autoSizeColumn(i);
    }
    response.setContentType("Application/Msexcel");
    response.setHeader("Content-Disposition",
        String.format("attachment; filename=\"schedule.xlsx\""));
    wb.write(fileOut);
}

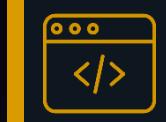
```



## 구현 기능 - 8



# 여행일정 작성 및 공유 웹 애플리케이션



구매한 일정을 오버라이딩 하여 나의 새 일정으로 작성할 수 있도록 하여 사용자의 여행 일정 작성의 편의성을 높여주는 기능

- ▣ 구매한 일정의 id값으로 여행 도시, 일자 별 여행일정, 일자별 가계부 리스트를 조회해와서 새로운 일정 id로 하고 기존 일정 id를 ref로 참조하도록 db에 insert



```

public int insertOverrideTravel(int trvId, int memberId) {
    Travel trv = td.selectTravel(sqlSession, trvId);

    Travel overrideTrv = new Travel();
    overrideTrv.setTrvTitle("제목 없음");
    overrideTrv.setStartDate(trv.getStartDate());
    overrideTrv.setEndDate(trv.getEndDate());
    overrideTrv.setCompNumber(1);
    overrideTrv.setBudget(trv.getBudget());
    overrideTrv.setMemberId(memberId);
    overrideTrv.setTrvRef(trvId);

    int result = td.insertOverrideTrv(sqlSession, overrideTrv);
    int ovTrvId = td.selectTrvId(sqlSession);

    ArrayList<TrvCity> trvCityList = td.selectTrvCityList(sqlSession, trvId);
    for(int i = 0; i < trvCityList.size(); i++) {
        TrvCity trvCity = new TrvCity();
        trvCity.setCityId(trvCityList.get(i).getCityId());
        trvCity.setTrvId(ovTrvId);
        result += td.insertTrvCity(sqlSession, trvCity);
    }

    ArrayList<TrvDay> trvDayList = td.selectTrvDayList(sqlSession, trvId);
    for(int i = 0; i < trvDayList.size(); i++) {
        TrvDay trvDay = new TrvDay();
        trvDay.setDayNumber(trvDayList.get(i).getDayNumber());
        trvDay.setDayDate(trvDayList.get(i).getDayDate());
        trvDay.setTrvId(ovTrvId);
        result += td.insertTrvDay(sqlSession, trvDay);
        int dayId = td.selectTrvDayId(sqlSession);

        ArrayList<TrvCost> costList = td.selectCostList(sqlSession,
                                                       trvDayList.get(i).getDayId());
        for(int j = 0; j < costList.size(); j++) {
            TrvCost cost = costList.get(j);
            cost.setDayId(dayId);
            result += td.insertTrvCost(sqlSession, cost);
        }
    }
}

```



## 개발 프로젝트 - 2



# 개인간 중고물품 대여 웹 애플리케이션



**Share Things**

찾는 물건이 있으신가요?

물품등록하기

CATEGORY

전자기기 취미레저 패션뷰티 유아동 반려동물

공지사항

No.	Title	Date
26	[필독]전체 공지1	2019-05-26
25	대여시 물품 수거/반환 택배 공지5	2019-05-26
24	대여시 물품 수거/반환 택배 공지4	2019-05-26
23	대여시 물품 수거/반환 택배 공지3	2019-05-26
22	대여시 물품 수거/반환 택배 공지2	2019-05-26

카테고리별 TOP5

전자기기 취미레저 패션뷰티 **유아동** 반려동물

Github : <https://github.com/Julia8956/shareThings>

## 프로젝트 명 : SHARE THINGS

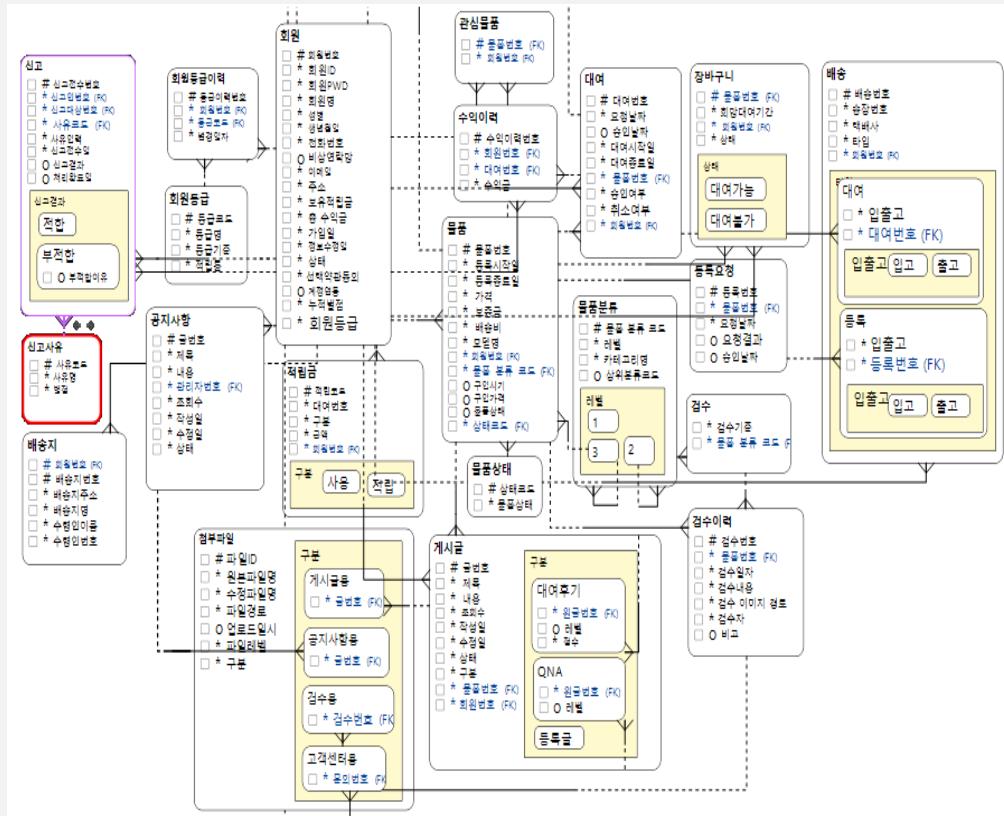
- 수행 기간 : 2019.05.10 ~ 2019.05.27 (3주)
- 개발 목표 : 사용자가 중고 물품을 등록하고 기업이 인수하여 대여 관리를 함으로써 사용자에게 중간 수수료를 뺀 대여 수익금을 되돌려 주는 웹 사이트 구축

## 사용 기술 및 개발 환경

- Language : JAVA 8, HTML5, CSS3, JavaScript, JSP 2.5, Servlet 3.1
- IDE : Eclipse Oxygen
- Database : Oracle 11.2.0
- WAS : Apache Tomcat 8.5
- Library : JQuery, ojdbc6, Java Mail, apache POI, cos, gson, Bootstrap 3, Sematic UI
- API : CoolSMS, 아임포트, 다음 주소 검색
- 형상관리 : github



## DB 모델링



## 담당 구현 기능

- 회원가입 (비밀번호 암호화, 이메일 인증)
- ID/PWD찾기 (이메일 / 휴대폰 인증), 로그인
- 메인페이지 TOP5 물품 조회, 키워드 검색
- 대여일, 대여 가격, 카테고리에 따른 상세 필터 검색, 정렬 기능
- JQuery datepicker를 사용한 대여 가능 기간 조회 기능
- 대여, 장바구니추가, 관심 물품 추가 / 회원간 신고 기능
- Q&A, 후기(사진 업로드) 및 댓글 작성, SMS 알림 전송
- 공지사항 조회, 검색, 첨부파일 등록/ 다운로드 기능
- DB 관리

## 공통 역할

- 프로젝트 기획 : 주제선정 및 기획, 기능 정의서 작성
- UI설계 : 프로토타입 제작(카카오오픈), Use Case Diagram, 스토리보드 작성
- DB모델링 : 논리모델링, 물리모델링
- 클래스 설계 : 클래스 다이어그램, 시퀀스 다이어그램 작성



## 구현 기능 - 1



# 개인간 중고물품 대여 웹 애플리케이션



The screenshot shows a user registration form with several validation messages displayed in red boxes:

- 아이디 \*: 아이디는 한글 및 특수문자 포함할 수 없습니다!
- 비밀번호 \*: 비밀번호를 다시 확인해주세요! (영문, 숫자 각 1자이상 포함 중 6자 이상)
- 비밀번호 재확인 \*: 비밀번호가 일치하지 않습니다!
- 이름(실명) \*: 유효하지 않은 이름입니다! (한글로만 입력 가능)

### 회원가입

#### 정규표현식

### 회원가입 기능

☞ 회원가입 시 정규표현식을 이용하여 아이디에 한글이 입력되거나 비밀번호에 조건에 맞지 않는 값이 들어왔는지 확인후 메시지를 보여주어 사용자가 재입력 할 수 있도록 하는 기능

```

var koreanCheck = /^[^a-z|0-9]/gi;
var koreanCheck2 = /[ㄱ-ㅎ|ㅏ-ㅣ|[ㅏ-ㅣ]/gi;
var blankCheck = /\w+/";
var engAtLeastOneCheck = /[a-z]+/i;
var numAtLeastOneCheck = /[0-9]+/;
var numberCheck = /^[^0-9]/;
$("#userId").change(function() {
    if(koreanCheck.test($("#this").val())){
        $("#idWrongMsg").show();
    }else{
        $("#idWrongMsg").hide();
    }
});
$("#userPwd").change(function() {
    var pwdLengthCheck = /\w{6}/;
    if(!pwdLengthCheck.test($("#this").val())){
        $("#pwdWrongMsg").show();
    }else if(koreanCheck2.test($("#this").val())){
        $("#pwdWrongMsg").show();
    }else if(!engAtLeastOneCheck.test($("#this").val())){
        $("#pwdWrongMsg").show();
    }else if(!numAtLeastOneCheck.test($("#this").val())){
        $("#pwdWrongMsg").show();
    }else if($("#this").val() == $("#userId").val()){
        $("#pwdWrongMsg").show();
    }else{
        $("#pwdWrongMsg").hide();
    }
});
$("#email").change(function() {
    var atCheck = /@+/;
    if(koreanCheck2.test($("#this").val())){
        $("#emailWrongMsg").show();
    }else if(!atCheck.test($("#this").val())){
        $("#emailWrongMsg").show();
    }else{
        $("#emailWrongMsg").hide();
    }
});

```



## 구현 기능 - 2



# 개인간 중고물품 대여 웹 애플리케이션



회원 가입

아이디 \*

비밀번호 \*

비밀번호 재확인 \*

이름(실명) \*

성별 \*

생년월일 \*

휴대폰 번호 \*

이메일 주소 \*

주소 \*

비상 연락망

(\*: 필수 입력)

☞ 회원 가입시 서블릿 filter와 wrapper클래스를 거쳐 비밀번호를 SHA512방식을 이용해 암호화하여 DB에 저장

비밀번호 암호화  
SHA 512

```
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
    throws IOException, ServletException {

    HttpServletRequest hRequest = (HttpServletRequest)request;
    LoginWrapper lw = new LoginWrapper(hRequest);
    chain.doFilter(lw, response);
}

public LoginWrapper(HttpServletRequest request) {
    super(request);
}

@Override
public String getParameter(String key) {
    String value = "";
    if(key != null && key.equals("userPwd")) {
        value = getSha512(super.getParameter(key));
    }else {
        value = super.getParameter(key);
    }
    return value;
}

public static String getSha512(String pwd) {
    String encPwd = null;
    try {
        MessageDigest md = MessageDigest.getInstance("SHA-512");
        byte[] bytes = pwd.getBytes(Charset.forName("UTF-8"));
        md.update(bytes);
        encPwd = Base64.getEncoder().encodeToString(md.digest());
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return encPwd;
}
```



## 구현 기능 - 3



# 개인간 중고물품 대여 웹 애플리케이션



The screenshot shows a web browser displaying the ShareThings website. A modal dialog box is open, prompting the user to click a link to verify their email. The dialog contains the text: "아래 링크에 접속하여 이메일 인증을 진행해주세요." and a button labeled "이메일 인증하기". Below the dialog, a smaller window shows the result of the verification: "127.0.0.1:8888 내용: 인증에 성공했습니다!" (Email verification successful!). The ShareThings logo is visible at the bottom left of the page.

회원 가입 후 가입시 입력한 이메일로 인증 코드가 포함된 url을 전송해 사용자가 클릭을 통해 이메일 인증을 할 수 있는 기능  
 ↵ java Mail 라이브러리를 이용해 사용자 이메일주소를 SHA256방식을 이용해 암호화한 후 링크를 전송하여 사용자가 클릭 시 코드의 일치 여부를 확인하여 member 테이블의 emailVerif 컬럼의 값을 'Y'로 update.

```

String host = "http://127.0.0.1:8888/st";
String from = "sharethings1224@gmail.com";
String to = newMember.getEmail();
String subject = "쉐어띵스 인증 메일입니다.";
String content = "아래 링크에 접속하여 이메일 인증을 진행해주세요." + "<br><a href='"
+ host + "/views/member/emailVerif.jsp?code="
+ new EmailVerification().getSHA256(to) + "'>이메일 인증하기</a>";

Properties p = new Properties();
p.put("mail.smtp.user", from);
p.put("mail.smtp.host", "smtp.gmail.com");
p.put("mail.smtp.port", "465");
p.put("mail.smtp.starttls.enable", "true");
p.put("mail.smtp.auth", "true");
p.put("mail.smtp.debug", "true");
p.put("mail.smtp.socketFactory.port", "465");
p.put("mail.smtp.socketFactory.class", "javax.net.ssl.SSLSocketFactory");
p.put("mail.smtp.socketFactory.fallback", "false");

Authenticator auth = new Gmail();
Session ses = Session.getInstance(p, auth);
ses.setDebug(true);
MimeMessage msg = new MimeMessage(ses);
try {
    msg.setSubject(subject);
    Address fromAddr = new InternetAddress(from);
    msg.setFrom(fromAddr);
    Address toAddr = new InternetAddress(to);
    msg.addRecipient(Message.RecipientType.TO, toAddr);
    msg.setContent(content, "text/html; charset=UTF-8");
    Transport.send(msg);
} catch (MessagingException e) {
}
  
```



## 구현 기능 - 4



# 개인간 중고물품 대여 웹 애플리케이션



The screenshot shows the homepage of the 'Share Things' website. At the top, there's a search bar and a button to add items. Below the header, there are category tabs: 전자기기, 취미레저, 패션뷰티, 유아동, and 반려동물. A featured item is displayed with a red border, showing a laptop and the text '거의 새것같은 노트북!!' (Almost new laptop!!). To the right is a list of notices with titles like '[밀독]전체 공지1' and dates from 2019-05-26 to 2019-05-28. Below that is a 'TOP5 물품 조회' section with five items: a stroller, a child car seat, a living room set, a mattress, and a white Audi car model.

메인페이지에서 전체 TOP5 물품, 카테고리별 TOP5 물품 조회 기능

☞ 물품의 대여 횟수 desc로 ORDER BY를 하여 정렬하고, carousel 놓을 등록 글 정보와 사진, 리뷰 정보를 Join하여 rownum이 1부터 5사이인 물품만 조회.

```

String verif = request.getParameter("verif");
HashMap<String, Object> ctgTop5map = new BoardService().getTop5List();
ArrayList<Notice> notice5List = new NoticeService().getNotice5List();
String page = "";
if(!ctgTop5map.isEmpty() && !notice5List.isEmpty()) {
    page = "views/main/main.jsp";
    request.setAttribute("ctgTop5map", ctgTop5map);
    request.setAttribute("notice5List", notice5List);
    if(verif != null) {
        request.setAttribute("verif", "success");
    }
}
SELECT ROWNUM, RNUM, SUB.* , RV.*
FROM (SELECT B.BNO, P.PNO, M.USER_NAME, B.BTITLE, B.BCOUNT, B.BDATE,
A.ANO, A.CHANGE_NAME, A.FILE_PATH, P.PRICE, P.DEPOSIT, COUNT(RT.RT_NO) AS RTCOUNT
FROM BOARD B
JOIN MEMBER M ON(M.UNO = B.BWRITER)
JOIN ATTACHMENT A ON (B.BNO = A.BNO)
JOIN PRODUCT P ON (B.PNO = P.PNO)
JOIN PSTATUS PS ON (P.SID = PS.SID)
LEFT JOIN RENTAL RT ON (P.PNO = RT.PNO)
WHERE B.BTYPE = ? AND B.STATUS = 'Y' AND A.ATYPE = 'B' AND A.FILE_LEVEL = 0
AND PS.SID IN ('PS10', 'PS11', 'PS12', 'PS13', 'PS14', 'PS15')
GROUP BY B.BNO, P.PNO, M.USER_NAME, B.BTITLE, B.BCOUNT, B.BDATE,
A.ANO, A.CHANGE_NAME, A.FILE_PATH, P.PRICE, P.DEPOSIT
ORDER BY RTCOUNT DESC, BCOUNT DESC) SUB
LEFT JOIN (SELECT PARENT_BNO, NVL(COUNT(BNO), 0) AS RVCOUNT,
NVL(ROUND(AVG(REVIEW_STAR), 1), 0) AS RVSTAR
FROM BOARD RV
WHERE BLEVEL = 0 AND BTYPE = ?
GROUP BY PARENT_BNO) RV ON (SUB.BNO = RV.PARENT_BNO)
WHERE ROWNUM BETWEEN 1 AND 5;

```



## 구현 기능 - 5



# 개인간 중고물품 대여 웹 애플리케이션



공지사항

공지(제목+내용) 검색

번호	제목	작성자	작성일
26	[필독]전체 공지1	관리자	2019-05-26
25	대여시 물품 수거/반환 택배 공지5	관리자	2019-05-26
24	대여시 물품 수거/반환 택배 공지4	관리자	2019-05-26
23	대여시 물품 수거/반환 택배 공지3	관리자	2019-05-26
22	대여시 물품 수거/반환 택배 공지2	관리자	2019-05-26
21	대여시 물품 수거/반환 택배 공지1	관리자	2019-05-26
20	사이트 이용 규칙 위반 공지5	관리자	2019-05-26
19	사이트 이용 규칙 위반 공지4	관리자	2019-05-26
18	사이트 이용 규칙 위반 공지3	관리자	2019-05-26
17	사이트 이용 규칙 위반 공지2	관리자	2019-05-26

### 공지사항 Paging

#### 공지사항 목록 조회시 페이징 처리

☞ 목록 하단의 페이지 버튼 클릭시 currentPage정보를 넘겨서 selectList.no 서블릿을 호출, selectList.no 서블릿에서는 전체 목록의 count를 조회해와 maxPage, EndPage, startPage의 값을 계산하고 이를 PageInfo 객체로 만들어 공지사항 목록을 조회 할 때 db에서 원하는 행만 조회해와서 페이징 화면 구성

```

<% for(int p = startPage; p <= endPage; p++) {
    if(p == currentPage) { %>
<a class="ui gray circular Label"><%= p %></a>
<% } else { %>
<a class="ui gray circular Label"
href=<%= request.getContextPath()%>/selectList.no?currentPage=<%= p %><%= p %></a>
<% }
} %>

int currentPage;
int limit;
int maxPage;
int startPage;
int endPage;
|
currentPage = 1;

if(request.getParameter("currentPage") != null) {
    currentPage = Integer.parseInt(request.getParameter("currentPage"));
}
limit = 10;
int listCount = new NoticeService().getnListCount();

maxPage = (int)(Math.ceil((double)listCount / limit));
startPage = ((int)(Math.floor((double)currentPage / limit))) * 10 + 1;
endPage = startPage + 10 - 1;
if(maxPage < endPage) {
    endPage = maxPage;
}
PageInfo pi = new PageInfo(currentPage, limit, maxPage, startPage, endPage);

ArrayList<Notice> nList = new NoticeService().selectNoticeList(pi);
String page = "";
if(!nList.isEmpty()) {
    page = "views/main/noticeList.jsp";
    request.setAttribute("nList", nList);
    request.setAttribute("pi", pi);
}

```



공지사항

제목	[필독]전체 공지1			
작성자	관리자	작성일	2019-06-16	조회수
첨부파일	<a href="#">내래고조PPT_중간발표용_0522.pptx</a> <a href="#">내래고조PPT_역할추가_0521.pptx</a>			

[필독]전체 공지1

목록으로

파일 다운로드

```
int ano = Integer.parseInt(request.getParameter("ano"));
Attachment file = new NoticeService().selectOneAttachment(ano);

BufferedInputStream buf = null;
ServletOutputStream downOut = response.getOutputStream();
File downFile = new File(file.getFilePath() + file.getChangeName());
response.setContentType("text/plain; charset=utf-8");
response.setHeader("Content-Disposition", "attachment; filename=\"" +
new String(file.getOriginName().getBytes("UTF-8"), "ISO-8859-1") + "\"");
response.setContentLength((int)downFile.length());

FileInputStream fin = new FileInputStream(downFile);
buf = new BufferedInputStream(fin);
readBytes = 0;
while((readBytes = buf.read()) != -1) {
    downOut.write(readBytes);
}
downOut.close();
buf.close();
```

### 공지사항 게시글의 첨부파일 다운로드

- ☞ 공지사항에 첨부된 파일을 클릭시 inputStream과 outputStream을 연결하여 서버에 저장된 파일을 불러와서 클라우드에 보내는 응답에 setHeader를 이용해 파일 다운로드.



## 구현 기능 - 7



# 개인간 중고물품 대여 웹 애플리케이션



원하는 물품을 상세 필터를 적용하여 검색, 정렬할 수 있는 기능  
 ↵ 날짜와 대여가격 등을 설정하고 세분류 카테고리를 선택시 db에서 카테고리별로 결과를 조회해와서 union을 통해 합치고, 정렬방법에서 인기순, 리뷰순을 선택 시 물품에 연결된 대여 count와 리뷰 count 혹은 리뷰 평점을 계산해와서 해당 순으로 정렬.

```

for(int i = 0; i < ctgLv3Arr.length; i++) {
    if(i != ctgLv3Arr.length - 1) {
        if(filterMap.get("startDate") != null && filterMap.get("endDate") != null) {
            query += prop.getProperty("selectFilter") + " UNION ";
        }else {
            query += prop.getProperty("selectFilterNoDate") + " UNION ";
        }
    }
}

String orderBy = "";
if(filterMap.get("orderBy").equals("recent")) {
    orderBy = "BDATE DESC";
}else if(filterMap.get("orderBy").equals("popular")) {
    orderBy = "RTCOUNT DESC, BDATE DESC";
}else if(filterMap.get("orderBy").equals("review")) {
    orderBy = "RVSTAR DESC, BDATE DESC";
}else if(filterMap.get("orderBy").equals("priceAsc")) {
    orderBy = "PRICE ASC, BDATE DESC";
}else if(filterMap.get("orderBy").equals("priceDesc")) {
    orderBy = "PRICE DESC, BDATE DESC";
}

stmt = con.prepareStatement("SELECT * FROM (" + query + ") ORDER BY " + orderBy);
if(filterMap.get("startDate") != null && filterMap.get("endDate") != null) {
    for(int i = 0; i < ctgLv3Arr.length; i++) {
        stmt.setString((6*i + 1), (String)filterMap.get("ctgLv2"));
        stmt.setString((6*i + 2), ctgLv3Arr[i]);
        stmt.setDate((6*i + 3), (Date)filterMap.get("endDate"));
        stmt.setInt((6*i + 4), (int)filterMap.get("minPrice"));
        stmt.setInt((6*i + 5), (int)filterMap.get("maxPrice"));
        stmt.setString((6*i + 6), "등록");
    }
}

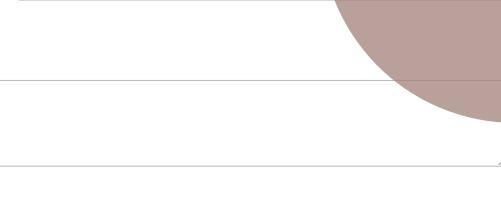
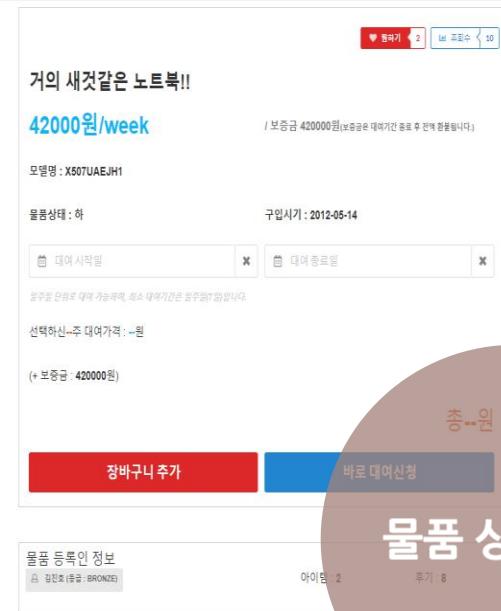
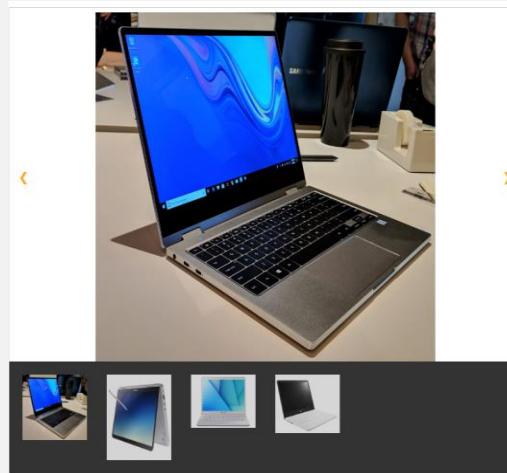
```



## 구현 기능 - 8



# 개인간 중고물품 대여 웹 애플리케이션



Q & A (8)

나의 질문

질문을 입력해주세요

등록하기

### 물품 상세정보 조회 기능

☞ 물품ID를 이용해서 조회 화면에 필요한 사진, 등록인 정보, 등록 정보, QnA 리스트, review 리스트, 대여 리스트, 썸 리스트를 모두 DB에서 조회해온 후 HashMap에 각각 저장하여 view 페이지 구성.

```
public HashMap<String, Object> selectOneBoard(int bno) {
    Connection con = getConnection();
    HashMap<String, Object> bDetailMap = new HashMap<String, Object>();
    int result = new BoardDao().updatebCount(con, bno);
    HashMap<String, Object> bmap = null;
    if (result > 0) {
        commit(con);
        bmap = new BoardDao().selectOneBoard(con, bno);
        if(bmap != null) {
            ArrayList<Attachment> attList = new BoardDao().selectBoardImages(con, bno);
            int uno = (int)bmap.get("uno");
            HashMap<String, Object> bWritermap = new BoardDao().selectbWritermap(con, uno);
            ArrayList<HashMap<String, Object>> QnAlist = new BoardDao().selectQnAlist(con, bno);
            ArrayList<HashMap<String, Object>> reviewList = new BoardDao().selectReviewList(con, bno);
            HashMap<String, Object> rvAttmap = new HashMap<String, Object>();
            for(int i = 0; i < reviewList.size(); i++) {
                int rvno = (int)reviewList.get(i).get("rvNo");
                ArrayList<Attachment> rvAttList = new BoardDao().selectBoardImages(con, rvno);
                rvAttmap.put(String.valueOf(rvno), rvAttList);
            }
            int pno = (int)bmap.get("pno");
            ArrayList<Rental> rentList = new BoardDao().selectRentList(con, pno);
            ArrayList<Reason> rReasonList = new BoardDao().selectRreasonList(con);
            bDetailMap.put("bmap", bmap);
            bDetailMap.put("attlist", attList);
            bDetailMap.put("bWritermap", bWritermap);
            bDetailMap.put("QnAlist", QnAlist);
            bDetailMap.put("reviewList", reviewList);
            bDetailMap.put("rvAttmap", rvAttmap);
            bDetailMap.put("rentList", rentList);
            bDetailMap.put("rReasonList", rReasonList);
        }
    }
}
```

## 구현 기능 - 9

# 개인간 중고물품 대여 웹 애플리케이션



127.0.0.1:8888 내용:  
장바구니에 담겼습니다. 이동하시겠습니까?

유아동 반려동물

/ 보증금 420000원  
(보증금은 대여기간 종료 후 전액 환불됩니다.)

**42000원/week**

모델명 : X507UAEJH1

물품상태 : 하 구입시기 : 2012-05-14

선택하신 2주 대여가격 : 84000원

(+ 보증금 : 42000원) (+ 왕복배송비 : 5000원)

**총 509000원**

장바구니 추가 바로 대여신청

대여 기간을 선택 시 자동으로 대여 가격과 배송비를 계산하여 총 대여 가격을 표시하고 장바구니에 추가 혹은 바로 대여신청 가능  
☞ 물품의 대여 리스트를 조회해서 상세페이지를 구성할 때 JQuery datepicker를 이용하여 대여 가능 기간만 calendar에서 활성화하여 보여주고, 선택한 시작일에 따라 선택 가능한 종료일을 보여주어 사용자의 선택에 따라 대여 가격을 계산.

```
beforeShowDay:function(date) {
    <%
        for(int i = 0; i < rentList.size(); i++) {
            Rental rt = rentList.get(i);
            Date rtStartDate = rt.getRtStartDate();
            Date rtEndDate = rt.getRtEndDate();

        %>
        st = new Date(<%= rtStartDate.getTime() %>);
        end = new Date(<%= rtEndDate.getTime() %>);
        end.setDate(end.getDate() + 2);
        if(date >= st && date <= end) {
            return [false, "disabled", "대여 불가"];
        }

        $("#endRICK").datepicker({
            dateFormat:"yy-mm-dd",
            defaultDate:"+1W",
            changeMonth:true,
            onSelect:function(selectedDate) {
                endDate = $(this).datepicker('getDate');
                startDate = new Date(endDate);
                startDate.setDate(endDate.getDate() - 6);
                startDay = startDate.getDay();
                $("#startPICK").datepicker("option", "maxDate", startDate);

                if($(this).datepicker("getDate") != null && $("#startPICK").datepicker("getDate") != null) {
                    var rtStart = $("#startPICK").datepicker("getDate");
                    var rtEnd = $(this).datepicker("getDate");
                    if(rtStart != null & rtEnd != null) {
                        var period = Math.ceil((rtEnd.getTime() - rtStart.getTime()) / 1000 * 60 * 60 * 24);
                        var totalprice = (period / 7) * <%= bmap.get("price") %>;
                        $("#priceCalc").text(totalprice);
                        $("#periodCalc").text(period / 7);
                        var rentprice = totalprice + <%= bmap.get("deposit") %>;
                        if(totalprice >= <%= bmap.get("diFreeAmount") %>) {
                            $("#dlResult").hide();
                        }else {
                            $("#dlResult").show();
                            rentprice += 5000;
                        }
                        $("#rentPrice").text(rentprice);
                    }
                }
            }
        });
    <%>
```



## 구현 기능 - 10



# 개인간 중고물품 대여 웹 애플리케이션



The screenshot shows a user interface for managing reviews. At the top, there's a navigation bar with tabs like '전자기기' and '취미레저'. Below it, a modal window titled '답글입력' (Reply Input) is open, showing a message from a user named 이선우 (2019-05-26) and a text input field labeled '답변내용' (Reply Content). Below the modal, a list of reviews is displayed. One review by 이선우 (2019-05-26) has a red circle around its reply button, which is highlighted in green. The review content is '만족스러웠어요~잘썼습니다!'.

### 물품 후기 작성

Cool SMS API

사진을 첨부하여 대여 후기를 작성시 물품 등록인에게 SMS 알림 전송, 등록인이 등록된 후기에 답글을 달 수 있는 기능  
 ↵ 사진 첨부 시 formData 객체를 ajax로 전송하여 사진을 서버에 저장, 후기 insert 완료시 cool SMS API를 이용하여 Message 객체를 생성하고 send 메소드를 호출하여 알림 메시지 전송. 로그인 유저가 등록인일 경우 답글 입력 버튼을 활성화.

```

if(ServletFileUpload.isMultipartContent(request)) {
    int maxSize = 1024 * 1024 * 10;
    String root = request.getSession().getServletContext().getRealPath("/");
    System.out.println("root: " + root);
    String filePath = root + "attach_upload/";

    MultipartRequest multiRequest = new MultipartRequest(request, filePath,
        maxSize, "UTF-8", new MyFileRenamePolicy());

    ArrayList<String> saveFiles = new ArrayList<String>();
    ArrayList<String> originFiles = new ArrayList<String>();

    Enumeration<String> files = multiRequest.getFileNames();
    while(files.hasMoreElements()) {
        String name = files.nextElement();
        System.out.println("name : " + name);

        if(multiRequest.getFileSystemName(name) != null && multiRequest.getOriginalFileName(name) != null)
            saveFiles.add(multiRequest.getFileSystemName(name));
        originFiles.add(multiRequest.getOriginalFileName(name));
        System.out.println("fileSystem name : " + multiRequest.getFileSystemName(name));
        System.out.println("originFile : " + multiRequest.getOriginalFileName(name));
    }

    ArrayList<Attachment> rvImgList = new ArrayList<Attachment>();
    for(int i = originFiles.size() - 1; i >= 0; i--) {
        Attachment at = new Attachment();
        at.setFilePath(filePath);
        at.setOriginName(originFiles.get(i));
        at.setChangeName(saveFiles.get(i));
        rvImgList.add(at);
    }

    int result = new BoardService().insertReview(newReview, rvImgList);
    response.setContentType("application/json");
    response.setCharacterEncoding("UTF-8");
    if(result > 0) {
        ArrayList<HashMap<String, Object>> reviewList = new BoardService().selectReviewList(In
        if(!reviewList.isEmpty()) {
            String phone = new MemberService().getBwriterPhone(Integer.parseInt(parentBno));
            int index = phone.indexOf("(");
            new SendSMS().send(phone.substring(index + 1), parentBtitle, "대여 후기");
            new Gson().toJson(reviewList, response.getWriter());
        }
    }
}
  
```



Thank you  
감사합니다