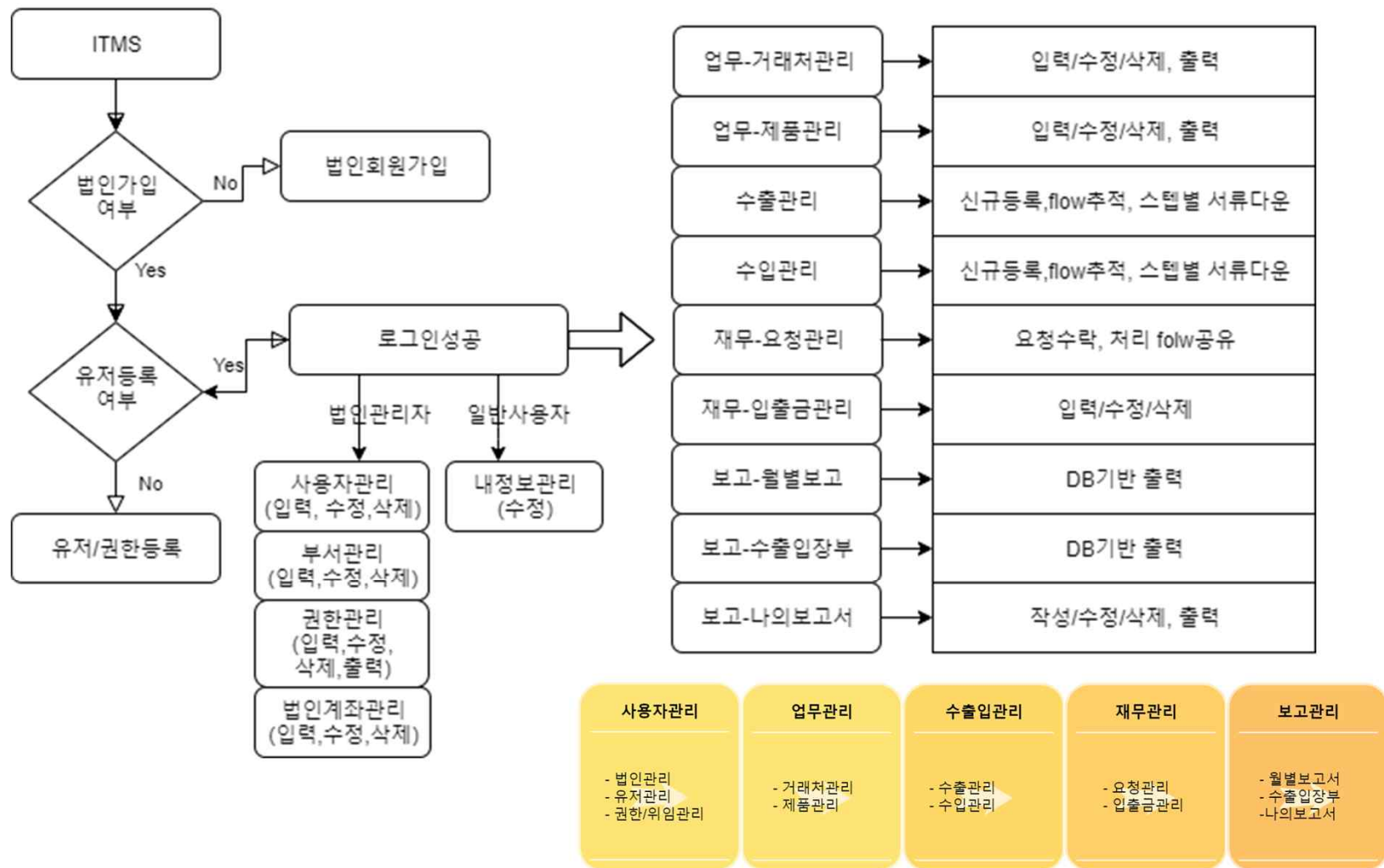


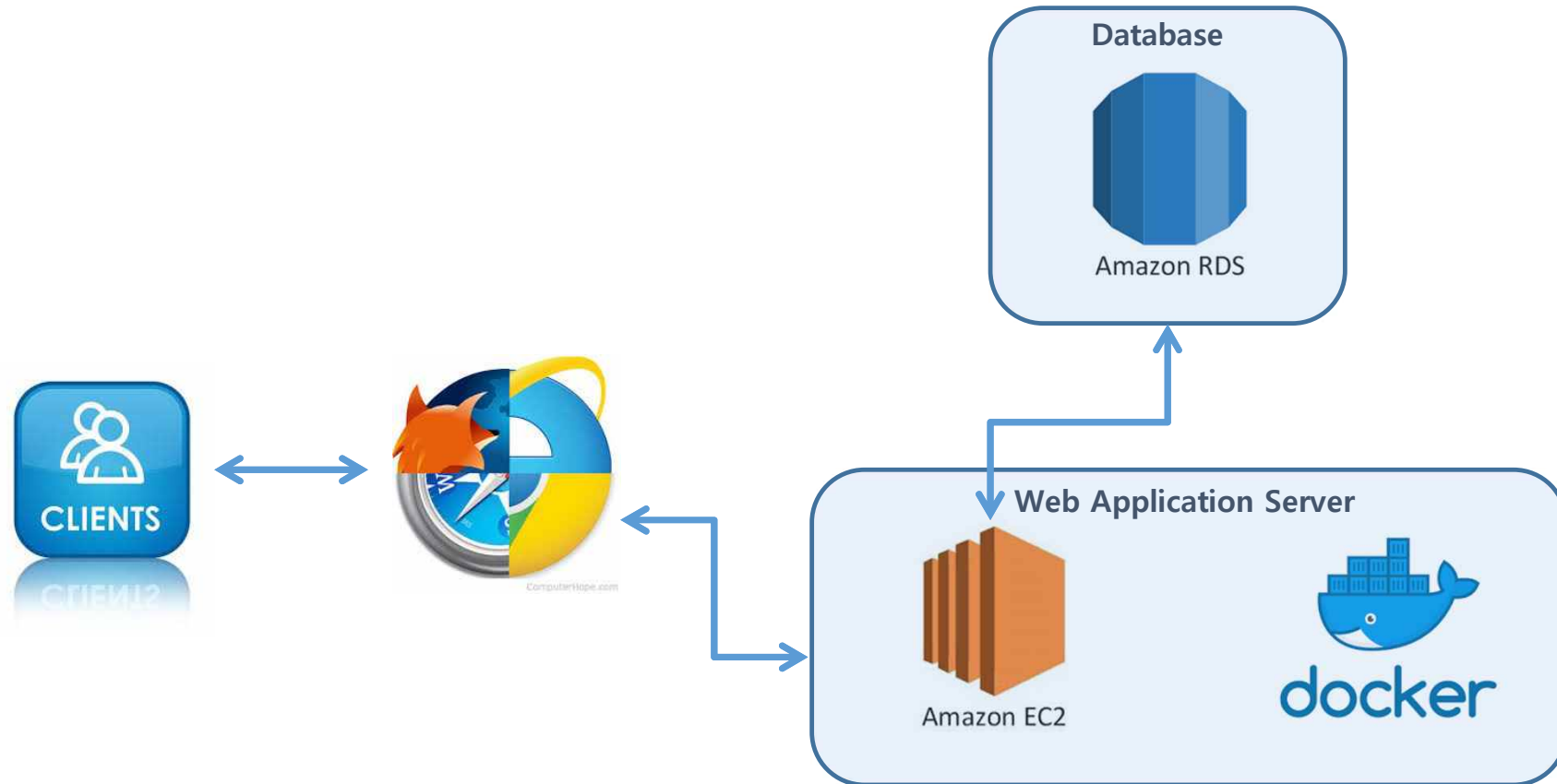
## 개요

프로젝트명	Java Spring MVC2-mybatis 활용 무역거래관리프로그램(ITMS)
개발기간	2019.12.23-2020.01.23
참여인원	5명
구현	사용자관리/업무관리/수출입관리/재무관리/보고관리
개발환경	Windows10, Linux(Ubuntu)
사용 도구	eGov, IntelliJ, DataGrip, Workbench, Git, Github, GitKraken
사용 기술	Java, JSP, MySQL(RDS), HTML, CSS, JS, jQuery, Ajax
플랫폼서비스	AWS

# 요약흐름도



## 시스템구성도



# 1)월별보고서

항목		2019년 종합보고												비고
		1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	
수출총액														
수입총액		173	80	100	160	160								
제반비용	수입물품대	0	0	0	0	0								
	관세	0	80	0	160	160								
	원류비	173	0	100	0	0								
(A)매출총이익 (매출액 - 매출원가)														
경상지출	관리비용	불류비	5	1	0	0	0							
		기타	0	0	0	0	0							
	합계(B)		5	1	0	0	0							
관세환급액(C)														
외환차손익(D)														
경상손익	일출손익 (A+B+C+D)													

```
<tr style="...*">
  <td colspan="3">수입총액</td>
  < :forEach var="row" items="${list}">
    <td>< :formatNumber value="${row.import_total}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td style="..." rowspan="3">제반비용</td>
  <td colspan="2">수입물품대</td>
  < :forEach var="row" items="${list}">
    <td>< :formatNumber value="${row.payment}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td colspan="2">관세</td>
  < :forEach var="row" items="${list}">
    <td>< :formatNumber value="${row.customs}" pattern="#,###"/></td>
  </:forEach>
</tr>
```

```

<tr style="...">
  <td colspan="3">수입총액</td>
  <:forEach var="row" items="${list}">
    <td><:formatNumber value="${row.import_total}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td style="..." rowspan="3">제반비용</td>
  <td colspan="2">수입물품대</td>
  <:forEach var="row" items="${list}">
    <td><:formatNumber value="${row.payment}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td colspan="2">관세</td>
  <:forEach var="row" items="${list}">
    <td><:formatNumber value="${row.customs}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td colspan="2">물류비</td>
  <:forEach var="row" items="${list}">
    <td><:formatNumber value="${row.logistical_cost}" pattern="#,###"/></td>
  </:forEach>
</tr>
<tr>
  <td colspan="3">(A) 매출총이익<br>(매출액 - 매출원가)</td>

```

재무관리(입출금) 테이블의 DB를 Read.  
List 결과값을 foreach로 반복처리.

# 1)월별보고서

Controller – Dao 를 통해 DB에 접근하여 mapper 내의 SQL 쿼리를 처리,

항목		2019년 종합보고												비고
		1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	
수출총액														
수입총액		173	80	100	160	160								
재반비용	수입물품대	0	0	0	0	0								
	관세	0	80	0	160	160								
	분류비	173	0	100	0	0								
(A)매출총이익 (매출액 - 매출원가)														
경상지출	관리비용	분류비	5	1	0	0	0							
		기타	0	0	0	0	0							
	합계(B)		5	1	0	0	0							
관세환급액(C)														
외환차손익(D)														
경상손익	합계손익 (A+B+C+D)													

```
@Controller
@RequestMapping("report/monthly/")
public class MonthlyController {
    @Autowired
    MonthlyDao monthlyDao;

    @RequestMapping(value = "/monthly.do" , method = RequestMethod.GET)
    public ModelAndView list(ModelAndView mav) throws Exception {
        List<MonthlyDto> list=monthlyDao.list();
        mav.setViewName("report/monthly/monthly");
        mav.addObject("list",list);
        return mav;
    }
}
```

```
import ...
@Repository
public class MonthlyDaoImpl implements MonthlyDao {
```

```
@Controller
@RequestMapping("report/monthly/")
public class MonthlyController {
    @Autowired
    MonthlyDao monthlyDao;

    @RequestMapping(value = "/monthly.do", method = RequestMethod.GET)
    public ModelAndView list(ModelAndView mav) throws Exception {
        List<MonthlyDto> list=monthlyDao.list();
        mav.setViewName("report/monthly/monthly");
        mav.addObject("list", list);
        return mav;
    }
}
```

```
import ...
@Repository
public class MonthlyDaoImpl implements MonthlyDao {
    @Autowired
    SqlSessionTemplate mybatis;
    @Override
    public List<MonthlyDto> list() { return mybatis.selectList("monthly.list_finance_transfer"); }
}
```

```
package com.mega.ims.report.monthly;
import java.util.Date;

public class MonthlyDto {
    private int action_type;
    private String item;
    private int amount_usd;
    private int amount_krw;
    private int amount_hdc;
    private Date closed_date;

    public int getAction_type() {
        return action_type;
    }

    public void setAction_type(int action_type) {
        this.action_type = action_type;
    }
}
```

처리 된 DB를 Controller 에서 전달처리+다음페이지 지정을 처리.

# 1)월별보고서

항목		2019년 종합보고													비고
		1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	합계	
수출총액															
수입총액		173	80	100	160	160									
제반비용	수입물품대	0	0	0	0	0									
	관세	0	80	0	160	160									
	물류비	173	0	100	0	0									
(A)미충종이익 (대출액 - 대출원가)															
경상지출	관리비용	물류비	5	1	0	0	0								
		기타	0	0	0	0	0								
	합계(B)		5	1	0	0	0								
간접원급액(C)															
외환차손액(D)															
경상손익	합계손익 (A+B+C+D)														

```
<mapper namespace="monthly">
  <resultMap type="com.mega.ITMS.report.monthly.SumMonthlyDto" id="sumMo
    <result property="payment" column="payment"/>
    <result property="customs" column="customs"/>
    <result property="logistical_cost" column="logistical_cost"/>
    <result property="domestic_logistical_cost" column="domestic logis
    <result property="others" column="others"/>
    <result property="import_total" column="inport_total"/>
    <result property="manage_total" column="manage_total"/>
  </resultMap>
  <select id="list_finance_transfer" resultType="java.lang.Integer" resu
    SELECT
      round(sum(case when item=1 then amount krw end)/1000) as payme
      round(sum(case when item=2 then amount krw end)/1000) as custo
      round(sum(case when item=3 then amount krw end)/1000) as logis
      round(sum(case when item=4 then amount krw end)/1000) as domes
      round(sum(case when item=5 then amount_hdc end)/1000) as other
      round(sum(case when item=6 then amount_hdc end)/1000) as other
```

Query 내에서 합계 및 단위처리(KRW,천원)를 하여 소수점을 자르고, 매월 그룹처리.

```

<mapper namespace="monthly">
  <resultMap type="com.mega.ITMS.report.monthly.SumMonthlyDto" id="sumMonthlyDto">
    <result property="payment" column="payment"/>
    <result property="customs" column="customs"/>
    <result property="logistical_cost" column="logistical_cost"/>
    <result property="domestic_logistical_cost" column="domestic_logistical_cost"/>
    <result property="others" column="others"/>
    <result property="import_total" column="import_total"/>
    <result property="manage_total" column="manage_total"/>
  </resultMap>
  <select id="list_finance_transfer" resultType="java.lang.Integer" resultMap="sumMonthlyDto">
    SELECT
      round(sum(case when item=1 then amount_krw end)/1000) as payment,
      round(sum(case when item=2 then amount_krw end)/1000) as customs,
      round(sum(case when item=3 then amount_krw end)/1000) as logistical_cost,
      round(sum(case when item=4 then amount_krw end)/1000) as domestic_logistical_cost,
      round(sum(case when item=5 then amount_hdc end)/1000) as others,
      round(sum(case when 4 > item then amount_krw end)/1000) as import_total,
      round(sum(case when item > 3 then amount_krw end)/1000) as manage_total
    FROM finance_transfer
    GROUP BY MONTH(closed_date)
  </select>
</mapper>

```

# 1)수출입장부

```

<tbody>
<c:forEach var="row" items="${exportList}">
  <tr>
    <th><script>document.write(nol);</script>
    </th>
    <script>x1()</script>
    <th>${row.customer}</th>
    <th><fmt:formatDate value="${row.issued_date}" pattern="yyyy-MM-dd"/></th>
    <th>${row.po_num}</th>
    <th>${row.p_name}</th>
    <th>${row.p_spec}</th>
    <th>${row.qty}</th>
    <th><fmt:formatNumber value="${row.unit_price}" pattern="#,###"/></th>
    <th><fmt:formatNumber value="${row.qty*row.unit_price}" pattern="#,###"/></th>
    <th><fmt:formatDate value="${row.close_date}" pattern="yyyy-MM-dd"/></th>
    <th>${row.notes}</th>
  </tr>
</c:forEach>
</tbody>
<tfoot class="full-width">
<tr>
  <th colspan="8">합계</th>
  <th colspan="3">금액</th>
</tr>
</tfoot>

```

NO	거래처	발주일	PO번호	상품명	상세구적	수량	단가(USD)	소계(USD)	입고일	비고
				이온전자배터리		500	4,000,000	2,000,000,000		NOTE
				USB무선		600	30,000	18,000,000		nothing
				와인		1000	550,000	550,000,000		JISOO
				USB무선		500	30,000	15,000,000		JISOO
				와인		1000	200,000	200,000,000		note
				64G USB 4.0		4000	70,000	280,000,000		note
				USB무선		30000	30,000	900,000,000		note
				볼록		500	80,000	40,000,000		note
				화이트		100	90,000	9,000,000		note
				컴퓨터 전원시스템		2000	90,000	180,000,000		note
				볼록		1000	95,000	95,000,000		note
				USB무선		1000	50,000	50,000,000		note
				볼록		100	950,000	95,000,000		note
				와인		200	899,000	179,800,000		note
				화이트		300	65,000	19,500,000		note
				와인		200	200,000	40,000,000		note
				볼록		100	95,000	9,500,000		note
				와인		100	550,000	55,000,000		note
				볼록		100	95,000	9,500,000		note
				와인		100	550,000	55,000,000		note
				볼록		100	95,000	9,500,000		note
				USB무선		100	30,000	3,000,000		note
				볼록		100	95,000	9,500,000		note
				와인		100	550,000	55,000,000		note
				USB무선		100	30,000	3,000,000		note
				볼록		100	95,000	9,500,000		note
				와인		100	550,000	55,000,000		note
				USB무선		100	30,000	3,000,000		note
				볼록		100	95,000	9,500,000		note
				와인		100	550,000	55,000,000		note
				USB무선		100	30,000	3,000,000		note

수출입거래기록 조회를 위한 jsp 처리 페이지.



## 2)수출입장부

수입관리						상세규격	수량	단가(USD)	소계(USD)	입고일	비고
NO	거래처	발주일	PO번호	제품명							
<pre> &lt;tbody&gt; &lt;c:forEach var="row" items="\${exportList}"&gt;   &lt;tr&gt;     &lt;th&gt;&lt;script&gt;document.write(nol);&lt;/script&gt;     &lt;/th&gt;     &lt;script&gt;x1()&lt;/script&gt;     &lt;th&gt;\${row.customer}&lt;/th&gt;     &lt;th&gt;&lt;fmt:formatDate value="\${row.issued_date}" pattern="yyyy-MM-dd"/&gt;&lt;/th&gt;     &lt;th&gt;\${row.po_num}&lt;/th&gt;     &lt;th&gt;\${row.p_name}&lt;/th&gt;     &lt;th&gt;\${row.p_spec}&lt;/th&gt;     &lt;th&gt;\${row.qty}&lt;/th&gt;     &lt;th&gt;&lt;fmt:formatNumber value="\${row.unit_price}" pattern="#,###"/&gt;&lt;/th&gt;     &lt;th&gt;&lt;fmt:formatNumber value="\${row.qty*row.unit_price}" pattern="#,###"/&gt;&lt;/th&gt;     &lt;th&gt;&lt;fmt:formatDate value="\${row.close_date}" pattern="yyyy-MM-dd"/&gt;&lt;/th&gt;     &lt;th&gt;\${row.notes}&lt;/th&gt;   &lt;/tr&gt; &lt;/c:forEach&gt; &lt;/tbody&gt; &lt;tfoot class="full-width"&gt; &lt;tr&gt;   &lt;th colspan="8"&gt;합계&lt;/th&gt;   &lt;th colspan="4"&gt;금액&lt;/th&gt; &lt;/tr&gt; &lt;/tfoot&gt; </pre>											
						이온전자배터리	500	4,000,000	2,000,000,000		NOTE
						USB무선	600	30,000	18,000,000		nothing
						와인	1000	550,000	550,000,000		JISOO
						USB무선	500	30,000	15,000,000		JISOO
						와인	1000	200,000	200,000,000		note
						64G USB 4.0	4000	70,000	280,000,000		note
						USB무선	30000	30,000	900,000,000		note
						블랙	500	80,000	40,000,000		note
						화이트	100	90,000	9,000,000		note
						컴퓨터 전원시스템	2000	90,000	180,000,000		note
						블랙	1000	95,000	95,000,000		note
						USB무선	1000	50,000	50,000,000		note
						블랙	100	950,000	95,000,000		note
						와인	200	899,000	179,800,000		note
						화이트	300	65,000	19,500,000		note
						와인	200	200,000	40,000,000		note
						블랙	100	95,000	9,500,000		note
						와인	100	550,000	55,000,000		note
						블랙	100	95,000	9,500,000		note
						와인	100	550,000	55,000,000		note
						블랙	100	95,000	9,500,000		note
						USB무선	100	30,000	3,000,000		note
						블랙	100	95,000	9,500,000		note
						와인	100	550,000	55,000,000		note
						USB무선	100	30,000	3,000,000		note
						블랙	100	95,000	9,500,000		note
						와인	100	550,000	55,000,000		note
						USB무선	100	30,000	3,000,000		note
						블랙	100	95,000	9,500,000		note
						와인	100	550,000	55,000,000		note
						USB무선	100	30,000	3,000,000		note

수출입거래기록 조회를 위한 jsp 처리 페이지.



## 2)수출입장부

수입관리

NO	거래처	발주일	PO번호	제품명	상세규격	수량	단가(USD)	소계(USD)	입고일	비고
1	1	2020-01-21	001	배터리	이온전지배터리	500	4,000,000	2,000,000,000		NOT
2	3	2020-01-21	002	키보드	USB무선	600	30,000	18,000,000		noth
3	4	2020-01-21	004	결핵시노트	와인	1000	550,000	550,000,000		JISO
4	4	2020-01-21	004	키보드	USB무선	500	30,000	15,000,000		JISO
5	8	2020-01-21	003	결핵시노트	와인	1000				
6	11	2020-01-22	004	USB	64G USB 4.0	4000				
7	11	2020-01-22	004	키보드	USB무선	30000				
8	8	2020-01-22	005	아이폰4s	블랙	500				
9	14	2020-01-22	006	아이폰4s	화이트	100				
10	14	2020-01-22	006	전원장치	컴퓨터 전원시스템	2000				
11	5	2019-01-22	20190101001	아이폰4s	블랙	1000				
12	5	2019-01-22	20190101001	마우스	USB무선	1000				
13	1	2019-02-22	20190225001	아이폰4s	블랙	100				
14	1	2019-02-22	20190225001	결핵시노트	와인	200				
15	1	2019-02-22	20190225001	아이폰4s	화이트	300				
16	1	2019-03-05	20190305001	결핵시노트	와인	200				
17	4	2019-03-06	20190306001	아이폰4s	블랙	100				
18	4	2019-03-06	20190306001	결핵시노트	와인	100				
19	4	2019-04-06	20190406002	아이폰4s	블랙	100				
20	4	2019-04-06	20190406002	결핵시노트	와인	100				
21	4	2019-05-06	20190506003	아이폰4s	블랙	100				
22	4	2019-05-06	20190506003	키보드	USB무선	100				
23	4	2019-06-06	20190606004	아이폰4s	블랙	100				
24	4	2019-06-06	20190606004	결핵시노트	와인	100				
25	4	2019-06-06	20190606004	키보드	USB무선	100				
26	4	2019-07-06	20190706005	아이폰4s	블랙	100				
27	4	2019-07-06	20190706005	결핵시노트	와인	100				
28	4	2019-07-06	20190706005	키보드	USB무선	100				
29	4	2019-08-15	20190815001	아이폰4s	블랙	100				
30	4	2019-08-15	20190815001	결핵시노트	와인	100				
31	4	2019-08-15	20190815001	키보드	USB무선	100				
32	4	2019-09-15	20190915002	아이폰4s	블랙	100				
33	4	2019-09-15	20190915002	결핵시노트	와인	100				
34	4	2019-09-15	20190915002	키보드	USB무선	100				

```

@Controller
@RequestMapping("report/tradeRecord/")
public class TradeRecordController {
    @Autowired
    TradeRecordDao tradeRecordDao;
    @RequestMapping(value = "/tradeRecord.do", method = RequestMethod.GET)
    public ModelAndView record(ModelAndView mav) {
        List<TradeRecordDto> importList=tradeRecordDao.importList();
        List<TradeRecordDto> exportList=tradeRecordDao.exportList();
        mav.setViewName("report/tradeRecord/tradeRecord");
        mav.addObject("importList", importList);
        mav.addObject("exportList", exportList);
        return mav;
    }
}

```

```

import ...
@Repository
public class TradeRecordDaoImpl implements TradeRecordDao{
    @Autowired
    SqlSessionTemplate mybatis;
    @Override
    public List<TradeRecordDto> importList() { return mybatis.selectList("tradeRecord.import"); }
    public List<TradeRecordDto> exportList() { return mybatis.selectList("tradeRecord.export"); }
}

```

DB Read처리, 처리된 DB의 전달을 위한  
Controller-Dao

## 2)수출입장부

수입관리										
NO	거래처	발주일	PO번호	제품명	상세규격	수량	단가(USD)	소계(USD)	입고일	비고
1	1	2020-01-21	001	배터리	이온전지배터리	500	4,000,000	2,000,000,000		NOTE
2	3	2020-01-21	002	키보드	USB무선	600				
3	4	2020-01-21	004	결핵시노트	와인	100				
4	4	2020-01-21	004	키보드	USB무선	500				
5	8	2020-01-21	003	결핵시노트	와인	100				
6	11	2020-01-22	004	USB	64G USB 4.0	400				
7	11	2020-01-22	004	키보드	USB무선	300				
8	8	2020-01-22	005	아이폰4s	블랙	500				
9	14	2020-01-22	006	아이폰4s	화이트	100				
10	14	2020-01-22	006	전원장치	컴퓨터 전원시스템	200				
11	5	2019-01-22	20190101001	아이폰4s	블랙	100				
12	5	2019-01-22	20190101001	마우스	USB무선	100				
13	1	2019-02-22	20190225001	아이폰4s	블랙	100				
14	1	2019-02-22	20190225001	결핵시노트	와인	200				
15	1	2019-02-22	20190225001	아이폰4s	화이트	300				
16	1	2019-03-05	20190305001	결핵시노트	와인	200				
17	4	2019-03-06	20190306001	아이폰4s	블랙	100				
18	4	2019-03-06	20190306001	결핵시노트	와인	100				
19	4	2019-04-06	20190406002	아이폰4s	블랙	100				
20	4	2019-04-06	20190406002	결핵시노트	와인	100				
21	4	2019-05-06	20190506003	아이폰4s	블랙	100				
22	4	2019-05-06	20190506003	키보드	USB무선	100				
23	4	2019-06-06	20190606004	아이폰4s	블랙	100				
24	4	2019-06-06	20190606004	결핵시노트	와인	100				
25	4	2019-06-06	20190606004	키보드	USB무선	100				
26	4	2019-07-06	20190706005	아이폰4s	블랙	100				
27	4	2019-07-06	20190706005	결핵시노트	와인	100				
28	4	2019-07-06	20190706005	키보드	USB무선	100				
29	4	2019-08-15	20190815001	아이폰4s	블랙	100				
30	4	2019-08-15	20190815001	결핵시노트	와인	100				
31	4	2019-08-15	20190815001	키보드	USB무선	100				
32	4	2019-09-15	20190915002	아이폰4s	블랙	100				
33	4	2019-09-15	20190915002	결핵시노트	와인	100				
34	4	2019-09-15	20190915002	키보드	USB무선	100				

```

<mapper namespace="tradeRecord">
  <resultMap type="com.mega.ITMS.report.tradeRecord.TradeRecordDto" id="tradeRecordDto">
    <result property="customer" column="b_id"/>
    <result property="issued_date" column="issued_date"/>
    <result property="po_num" column="doc_no"/>
    <result property="p_name" column="name"/>
    <result property="p_spec" column="spec"/>
    <result property="qty" column="qty"/>
    <result property="unit_price" column="price"/>
    <result property="total" column="total"/>
    <result property="close_date" column="closed_date"/>
    <result property="notes" column="notes"/>
  </resultMap>
  <select id="import" resultMap="tradeRecordDto">
    select basic_trade.*, order_detail.*, product.*
    from basic_trade
    inner join order_detail on basic_trade.id=order_detail.basic_id
    inner join product on order_detail.p_id=product.id where type_tr=2
  </select>
  <select id="export" resultMap="tradeRecordDto">
    select basic_trade.*, order_detail.*, product.*
    from basic_trade
    inner join order_detail on basic_trade.id=order_detail.basic_id
    inner join product on order_detail.p_id=product.id where type_tr=1
  </select>
</mapper>

```

연관된 DB 일괄 row처리를 위한 inner join

### 3)나의보고서

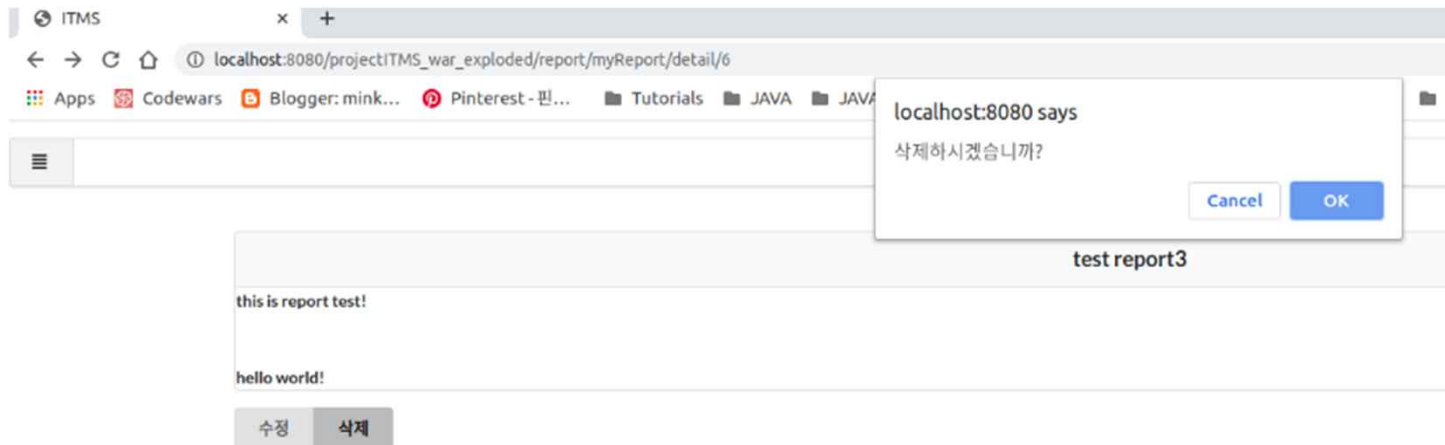
The screenshot displays a web application titled "신규보고작성" (New Report Writing). It features a form with a "제목" (Title) field containing "test report3". Below the title is a CKEditor with the text "this is report test!" and "hello world!". A code editor overlay shows the HTML source code for the CKEditor, including a `<div>` with a `<div class="field">` containing a `<textarea>` and a `<script>` block that uses `CKEDITOR.replace` to initialize the editor. The code is as follows:

```
<tbody>
<tr>
  <th colspan="16">
    <div class="ui form">
      <div class="field">
        <textarea class="form-control" id="p_content" name="content"></textarea>
        <script type="text/javascript">
          CKEDITOR.replace('p_content', {
            height: 450
          });
        </script>
      </div>
    </div>
  </th>
</tr>
</tbody>
```

At the bottom right of the code editor, there is a "등록" (Register) button with a red "2" badge.

위지위그 ckeditor 라이브러리 사용.

### 3)나의보고서



제목 클릭 시, 해당 보고서의 상세 내용 확인 가능, 수정 및 삭제 기능 포함.

```
<mapper namespace="report">
  <select id="list_report" resultType="com.mega.ITMS.report.myReport.MyReportDto">
    select * from report where com_id=#{com_id} and user_id=#{user_id} order by issued_date
  </select>
  <insert id="insert">
    insert into report
    (com_id, user_id, title, content, issued_date)
    values
    (#{com_id}, #{user_id},#{title},#{content},now())
  </insert>
  <select id="myReportDetail" resultType="com.mega.ITMS.report.myReport.MyReportDto">
    select * from report where id=#{id}
  </select>
  <delete id="myReportDel">
    DELETE FROM report
    WHERE id = #{id}
  </delete>
  <update id="update" parameterType="com.mega.ITMS.report.myReport.MyReportDto">
    update report SET content=#{content}, title=#{title} where id=#{id}
  </update>
</mapper>
```

```
@RequestMapping(value = "edit/{id}")
public ModelAndView myReportEdit(@PathVariable("id") int id, ModelAndView mav) {
    mav.setViewName("report/myReport/myReportEdit");
    mav.addObject("dto", myReportDao.myReportDetail(id));
    return mav;
}

@RequestMapping(value = "del/{id}")
public String myReportDel(@PathVariable int id) {
    myReportDao.myReportDel(id);
    return "redirect:../myReportList.do";
}

@RequestMapping(value = "update/{id}", method = RequestMethod.POST)
public String updateMyReport(@ModelAttribute MyReportDto dto) {
    myReportDao.updateMyReport(dto);
    return "redirect:../myReportList.do";
}
```

## 4) AWS-Docker배포



## 4) Docker-AWS\_EC2-배포

도커파일 생성 후, Root에서 자동+항상 돌아갈수 있도록  
 톰캣의 root 를 삭제한 후,  
 현 프로젝트의 패키지를 Root로 복사, 기본으로 잡아,  
 이미지 빌드를 위한 기본 설정을 해준다.

사전준비사항	
NO	내용
1	프로젝트 내 도커 설정파일
2	메이븐/도커설치
3	Docker-Hub 가입
4	AWS 가입
5	도메인 구입

```

1  #도커 컨테이너 기본으로 돌아가는 프로그램(이미지)
2
3
4  RUN rm -rf /usr/local/tomcat/webapps/ROOT
5
6  #톰캣의 기본 서버 어플을 삭제하고 (톰캣초기화면) 비워주는 것.
7
8  COPY ./target/ITMS-1.0.0-BUILD-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war
9
10 #카피 복사 할 파일경로/이름.. 을 /usr/local/tomcat/webapps/ROOT.war 로 복사시킴
11
12 COPY ./target/ITMS-1.0.0-BUILD-SNAPSHOT /usr/local/tomcat/webapps/ROOT
13
14 #만들어놓은 메이븐 패키지 파일을 루트로 복사하여 기본으로 잡아준다.
15
16 EXPOSE 8080
17
18 #포트는 8080으로 열어주삼~
  
```



## 4) Docker-AWS\_EC2-배포

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range
SSH	TCP	22
HTTP	TCP	80
HTTPS	TCP	443

Add Rule

**Warning**  
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access...

1. EC2인스턴스: linux+docker가 포함되어 있는 타입
2. 인스턴스의 보안그룹은 SSH, HTTP, HTTPS.
3. 고정 IP를 위해 엘라스틱 IP를 신청하여 생성해놓은 인스턴스와 결합 associate 처리.
4. 생성된 고정 IP를 도메인을 구입한 곳으로 접속하여 해당 아이피를 설정.

EC2 > Elastic IP addresses > Allocate Elastic IP address

### Allocate Elastic IP address

Allocate an Elastic IP address by selecting the public IPv4 address pool from which the public IP address is to be allocated. Elastic IP addresses incur charges if they are not associated with a running instance or a network interface that is attached to a running instance.

**Elastic IP address settings**

**Public IPv4 address pool**  
Indicates whether the public IPv4 address is to be allocated from Amazon's pool of public IPv4 addresses, or from your pool of public IPv4 addresses that you brought to your AWS account. [Learn more](#).

☒ Amazon's pool of IPv4 addresses

☐ My pool of public IPv4 addresses (option disabled because no pools found)

Cancel Allocate



## 4) Docker-AWS\_EC2-배포

```
minkyung@min:~/eclipse-workspace$ cd projectITMS/
minkyung@min:~/eclipse-workspace/projectITMS$ mvn package
[INFO] Scanning for projects...
[WARNING]
[WARNING] Some problems were encountered while building the effective model for com.mega:ITMS:war:1.0.0-BUILD-SNAPSHOT
[WARNING] 'dependencies.dependency.(groupId:artifactId:type:classifier)' must be unique: com.fasterxml.jackson.core:jackson-databind:jar ->
[WARNING]
[WARNING] It is highly recommended to fix these problems because they threaten the stability of your build.
[WARNING]
[WARNING] For this reason, future Maven versions might no longer support building such malformed projects.
[WARNING]
[INFO]
[INFO] -----< com.mega:ITMS >-----
[INFO] Building projectITMS 1.0.0-BUILD-SNAPSHOT
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ ITMS ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 15 resources
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:compile (default-compile) @ ITMS ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ ITMS ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 1 resource
[INFO]
[INFO] --- maven-compiler-plugin:2.5.1:testCompile (default-testCompile) @ ITMS ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ ITMS ---
[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ ITMS ---
[INFO] Packaging webapp
[INFO] Assembling webapp [ITMS] in [/home/minkyung/eclipse-workspace/projectITMS/target/ITMS-1.0.0-BUILD-SNAPSHOT]
[INFO] Processing war project
```

- 1) EC2 서버 내 웹어플리케이션 구동을 위한 톰캣 서버환경과 의존성관리를 위해 자바 프로젝트의 폴더에서 기 설치된 메이븐을 통한 package 작업을 진행.

## 4) Docker-AWS\_EC2-배포

```
minkyung@min:~/eclipse-workspace/projectITMS$ docker build . -t yo2020/projectitms:1.1.0
Sending build context to Docker daemon 400.7MB
Step 1/5 : FROM tomcat:8.5.45
--> 365b0a528e2e
Step 2/5 : RUN rm -rf /usr/local/tomcat/webapps/ROOT
--> Using cache
--> bafbf7d28d97
Step 3/5 : COPY ./target/ITMS-1.0.0-BUILD-SNAPSHOT.war /usr/local/tomcat/webapps/ROOT.war
--> bc24561fe594
Step 4/5 : COPY ./target/ITMS-1.0.0-BUILD-SNAPSHOT /usr/local/tomcat/webapps/ROOT
--> 2178bf86acde
Step 5/5 : EXPOSE 8080
--> Running in 11718680c07e
Removing intermediate container 11718680c07e
--> 6ebd01e7f07f
Successfully built 6ebd01e7f07f
Successfully tagged yo2020/projectitms:1.1.0
minkyung@min:~/eclipse-workspace/projectITMS$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/minkyung/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
minkyung@min:~/eclipse-workspace/projectITMS$ docker push yo2020/projectitms
The push refers to repository [docker.io/yo2020/projectitms]
251b008371b2: Layer already exists
1d9060203702: Layer already exists
2ff78e07ee08: Layer already exists
4a3ccf95eef9: Layer already exists
4437177a7bda: Layer already exists
5161954b2663: Layer already exists
```

2) 메이븐 패키징으로 만들어진 war파일을  
도커 이미지로 만드는 작업.

*\$docker build . -t*  
*도커이미지빌드명령어*

*yo2020*  
*docker-hub의 아이디*

*Projectitms*  
*프로젝트명*

*1.1.0*  
*구분을 위한 버전 tag*

3) 도커로그인(docker-hub)

4) 이미지 push /repository

## 4) Docker-AWS\_EC2-배포

```
minkyung@min:~$ ssh -i ~/Documents/minkschungcom.pem ubuntu@minks.shop
Welcome to Ubuntu 18.04.3 LTS (GNU/Linux 4.15.0-1057-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Jan 22 07:35:12 UTC 2020

System load:  0.44           Processes:           96
Usage of /:   37.0% of 7.69GB Users logged in:          0
Memory usage: 22%           IP address for eth0: 172.31.43.185
Swap usage:   0%            IP address for docker0: 172.17.0.1

 * Overheard at KubeCon: "microk8s.status just blew my mind".

      https://microk8s.io/docs/commands#microk8s.status

43 packages can be updated.
0 updates are security updates.

Last login: Mon Jan 20 07:51:45 2020 from 1.210.217.14
ubuntu@ip-172-31-43-185:~$ docker run -d -p 8080:8080 yo2020/projectitms:1.0.0
dc7a29512a20d0fb1f8cb03d98937bb55dd908a344931610278899289cbe1e
ubuntu@ip-172-31-43-185:~$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
dc7a29512a20        yo2020/projectitms:1.0.0  "catalina.sh run"   6 seconds ago      Up 5 seconds       0.0.0.0:8080->8080/tcp  strange_burnell
```

- 5) 나의 컴퓨터-> EC2 로 접속. (도메인을 고정ip로 연결해 놓은 후라, 도메인으로 진입가능)
- 6) docker- hub 로그인 / pull 처리
- 7) Running -d:백그라운드 running 설정(로그 별도확인, 터미널 닫아도 계속 run) -p:포트설정
- 8) Running 중인 컨테이너 확인