

客户流失管理

一 学习目标



二 客户流失管理定时器

1 需求分析

客户流失管理模块记录了对应流失客户信息即对于企业来说没有任何价值的客户，通过客户流失管理模块来进行相应数据维护。

流失客户定义规则 客户创建日期 超过6个月 1.没有产生任何订单 2.产生订单 最后一次下单日期距离现今超过六个月

2 页面原型

3 流程图

4 数据库信息

(t_customer_loss 客户流失表)

	列名	数据类型
id主建	id	int(11)
客户编号	cus_no	varchar(40)
客户名称	cus_name	varchar(20)
客户经理	cus_manager	varchar(20)
最后下单时间	last_order_time	date
确认流失时间	confirm_loss_time	date
流失状态	state	int(11)
流失原因	loss_reason	varchar(1000)
有效状态	is_valid	int(4)
创建时间	create_time	datetime
更新时间	update_time	datetime

(t_customer_retrieve 客户流失暂缓表)

	列名	数据类型
id主建	id	int(11)
流失id	loss_id	int(11)
措施	measure	varchar(500)
有效状态	is_valid	int(4)
创建时间	create_date	datetime
更新时间	update_date	datetime

5 编写CustomerTask

```
@Component
public class CustomerTask {

    @Resource
    public CustomerService customerService;
    //每天的16点59分0秒触发这个任务
    @Scheduled(cron = "0 59 16 * * ?")
    public void lossTask(){
        customerService.updateCustomerLossState();
        System.out.println("你好, 我被触发了--");
    }

}
```

6 编写Service

```
public void updateCustomerLossState(){
    List<CustomerLoss> list = customerDao.queryCustomerLoss();
    if(list!=null&&list.size()>0){
        for (int i = 0; i < list.size(); i++) {
            CustomerLoss customerLoss = list.get(i);
            customerLoss.setState(0);
            customerLoss.setIsValid(1);
            customerLoss.setCreateDate(new Date());
            customerLoss.setUpdateDate(new Date());
        }
    }
    AssertUtil.isTrue( customerLossDao.insertBatch(list)<1,"客户流失数据添加失败");
}
```

7 编写Dao

```
public List<CustomerLoss> queryCustomerLoss();
public Integer insertBatch(List<CustomerLoss> list);
```

8 编写Mapper

```
<insert id="insertBatch" parameterType="list">

    insert into
t_customer_loss(cus_no,cus_name,cus_manager,last_order_time,state,is_valid,
    create_date,update_date
    ) values
    <foreach collection="list" item="item" separator=",">
        (
            #{item.cusNo},#{item.cusName},#{item.cusManager},#
{item.lastOrderTime},
            #{item.state},#{item.isValid},#{item.createDate} ,#{item.updateDate}
        )
    </foreach>
```

```
</insert>

<select id="queryCustomerLoss" resultType="customerLoss">
  <![CDATA[
    SELECT
      khno as cusNo,name as cusName,cus_manager as cusManager ,o.order_date as
lastOrderTime
    FROM
      t_customer c
    left JOIN t_customer_order o on c.id = o.cus_id
    WHERE
      DATE_ADD(c.create_date, INTERVAL 6 DAY) < NOW()
      and c.is_valid = 1 and c.state = 0
      AND c.id not in (
        SELECT DISTINCT
          o.cus_id
        FROM
          t_customer_order o
        LEFT JOIN t_customer c ON c.id = o.cus_id
        WHERE
          DATE_ADD(o.order_date, INTERVAL 6 DAY) > NOW()
      )
  ]]>
</select>
```

9 编写Vo

```
public class CustomerLoss extends BaseVO {
    private Integer id;

    private String cusNo;

    private String cusName;

    private String cusManager;

    @JsonFormat(pattern="yyyy-MM-dd",timezone="GMT+8")
    private Date lastOrderTime;
    @JsonFormat(pattern="yyyy-MM-dd",timezone="GMT+8")
    private Date confirmLosTime;

    private Integer state;

    private String lossReason;

    public Integer getId() {
        return id;
    }

    public void setId(Integer id) {
```

```
        this.id = id;
    }

    public String getCusNo() {
        return cusNo;
    }

    public void setCusNo(String cusNo) {
        this.cusNo = cusNo == null ? null : cusNo.trim();
    }

    public String getCusName() {
        return cusName;
    }

    public void setCusName(String cusName) {
        this.cusName = cusName == null ? null : cusName.trim();
    }

    public String getCusManager() {
        return cusManager;
    }

    public void setCusManager(String cusManager) {
        this.cusManager = cusManager == null ? null : cusManager.trim();
    }

    public Date getLastOrderTime() {
        return lastOrderTime;
    }

    public void setLastOrderTime(Date lastOrderTime) {
        this.lastOrderTime = lastOrderTime;
    }

    public Date getConfirmLosTime() {
        return confirmLosTime;
    }

    public void setConfirmLosTime(Date confirmLosTime) {
        this.confirmLosTime = confirmLosTime;
    }

    public Integer getState() {
        return state;
    }

    public void setState(Integer state) {
        this.state = state;
    }

    public String getLossReason() {
        return lossReason;
    }
}
```

```

    }

    public void setLossReason(String lossReason) {
        this.lossReason = lossReason == null ? null : lossReason.trim();
    }
}

```

三 客户流失管理查询显示

1 需求分析

能够查询显出流失客户的基本信息

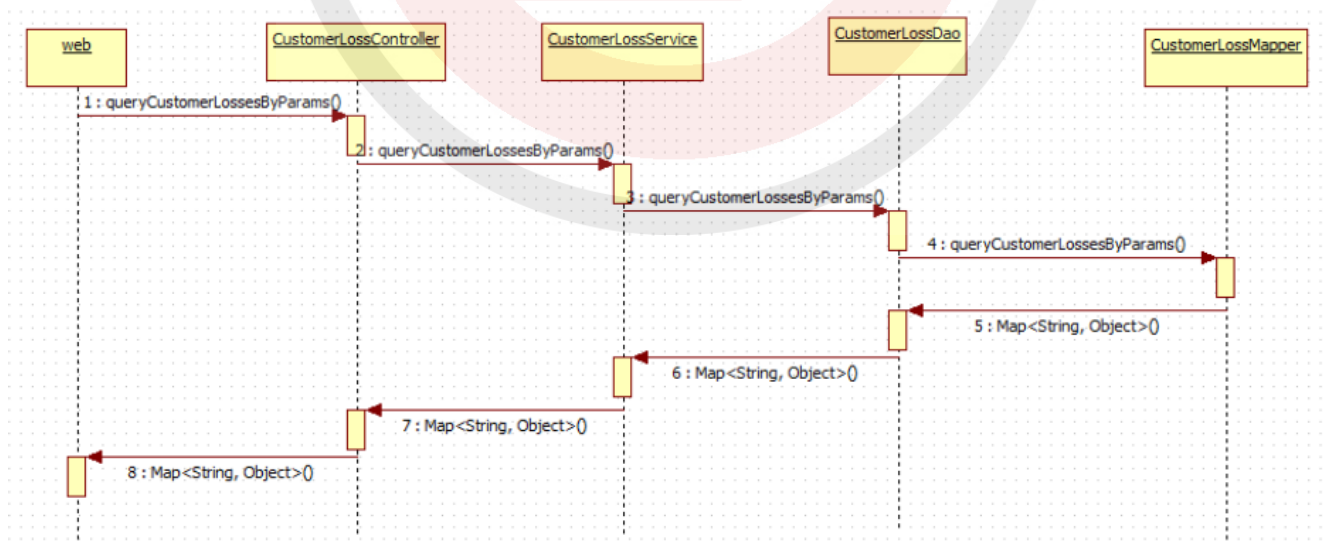
2 页面原型

客户名称:客户编号:客户经理:创建时间:

查询

	id	客户名称	客户经理	客户编号	上次下单时间	确认流失时间	流失状态	流失原因	创建时间	更新时间	操作
1	135	北京大牛科技	小张	KH21321321		2019-05-07	确认流失	脑溢血	2018-01-15	2018-01-15	查看流失详情
2	136	风驰科技	小红	KH20150526073022			暂缓流失		2018-01-15	2018-01-15	添加暂缓措施
3	137	test002	sxt	KH201704131032864			暂缓流失		2018-01-15	2018-01-15	添加暂缓措施
4	138	test003	test02	KH2017041310391067			暂缓流失		2018-01-15	2018-01-15	添加暂缓措施
5	139	阿里巴巴	test01	KH201709181112739			暂缓流失		2018-01-15	2018-01-15	添加暂缓措施

3 流程图



4 编写Controller

```

@Controller
@RequestMapping("customer_loss")

```

```
public class CustomerLossController extends BaseController {

    @Resource
    public CustomerLossService customerLossService;

    @RequestMapping("index")
    public String index() {
        return "customer_loss";
    }

    @RequestMapping("queryCustomerLossesByParams")
    @ResponseBody
    public Map<String, Object> queryCustomerLossesByParams(CustomerLossQuery customerLossQuery) {
        return customerLossService.queryCustomerLossesByParams(customerLossQuery);
    }
}
```

5 编写Service

```
@Service
public class CustomerLossService {
    @Resource
    public CustomerLossDao customerLossDao;

    public Map<String, Object> queryCustomerLossesByParams(CustomerLossQuery customerLossQuery){

        PageHelper.startPage(customerLossQuery.getPage(), customerLossQuery.getRows());
        List<CustomerLoss> list =
customerLossDao.queryCustomerLossesByParams(customerLossQuery);
        PageInfo<CustomerLoss> pageInfo = new PageInfo<>(list);
        Map<String, Object> map = new HashMap<>();
        map.put("total", pageInfo.getTotal());
        map.put("rows", pageInfo.getList());

        return map;
    }
}
```

6 编写Dao

```
public List<CustomerLoss> queryCustomerLossesByParams(CustomerLossQuery customerLossQuery);
```

7 编写Mapper

```
<select id="queryCustomerLossesByParams" resultMap="BaseResultMap"
parameterType="customerLossQuery">
```

```
select
<include refid="Base_Column_List" />
from t_customer_loss
<where>
    is_valid =1

    <if test="customerManager != null and customerManager != ''">
        and cus_manager like concat('%',{customerManager},'%')
    </if>
    <if test="customerNum != null and customerNum != ''">
        and cus_no like concat('%',{customerNum},'%')
    </if>
    <if test="customerName != null and customerName != ''">
        and cus_name like concat('%',{customerName},'%')
    </if>
    <if test="time != null and time!='' ">
        <![CDATA[
            and create_date <= #{time}
        ]]>
    </if>
</where>
</select>
```

8 编写Query

```
public class CustomerLossQuery extends BaseQuery {

    private String customerName;
    private String customerNum;
    private String customerManager;
    private String time;

    public String getCustomerName() {
        return customerName;
    }

    public void setCustomerName(String customerName) {
        this.customerName = customerName;
    }

    public String getCustomerManager() {
        return customerManager;
    }

    public void setCustomerManager(String customerManager) {
        this.customerManager = customerManager;
    }

    public String getCustomerNum() {
```




```
        return customerNum;
    }

    public void setCustomerNum(String customerNum) {
        this.customerNum = customerNum;
    }

    public String getTime() {
        return time;
    }

    public void setTime(String time) {
        this.time = time;
    }
}
```

9 前台js

```
$(function () {

    $("#dg").datagrid({
        rowStyler: function (index, rowData) {
            if (rowData.state == 0) {
                return "background-color:yellow";
            } else if (rowData.state == 1) {
                return "background-color:red";
            }
        }
    });
});

function formatterState(val) {
    if (val == 0) {
        return "暂缓流失";
    }
    if (val == 1) {
        return "确认流失";
    }
}

function formatterOp(val,rowData) {
    if(rowData.state==0){
        var href = "javascript:openCustomerRepriDetailTab('客户流失暂缓处理"+rowData.id+"',"+rowData.id+")";
        return "<a href=" + href + ">添加暂缓措施</a>";
    }

    if(rowData.state==1){
        var href = "javascript:openCustomerRepriDetailTab('客户流失情况
```

```
_" + rowData.id + "',' + rowData.id + '");  
    return "<a href=" + href + ">查看流失详情</a>";  
}  
  
}  
  
function searchCustomerLosses() {  
  
    $("#dg").datagrid("load",{  
        customerName: $("#customerName").val(),  
        customerManager : $("#customerManager").combobox("getValue"),  
        customerNum: $("#customerNum").val(),  
        time: $("#time").datebox("getValue")  
    })  
  
}  
  
function openCustomerRepriDetailTab(title,lossId) {  
    window.parent.openTab(title,ctx+"/customer_loss/toRepreivePage/"+lossId);  
}
```

四 客户流失暂缓措施操作

1 需求分析

对即将流失的客户进行挽留，并且将挽留计划保存起来

2 页面原型

客户流失信息

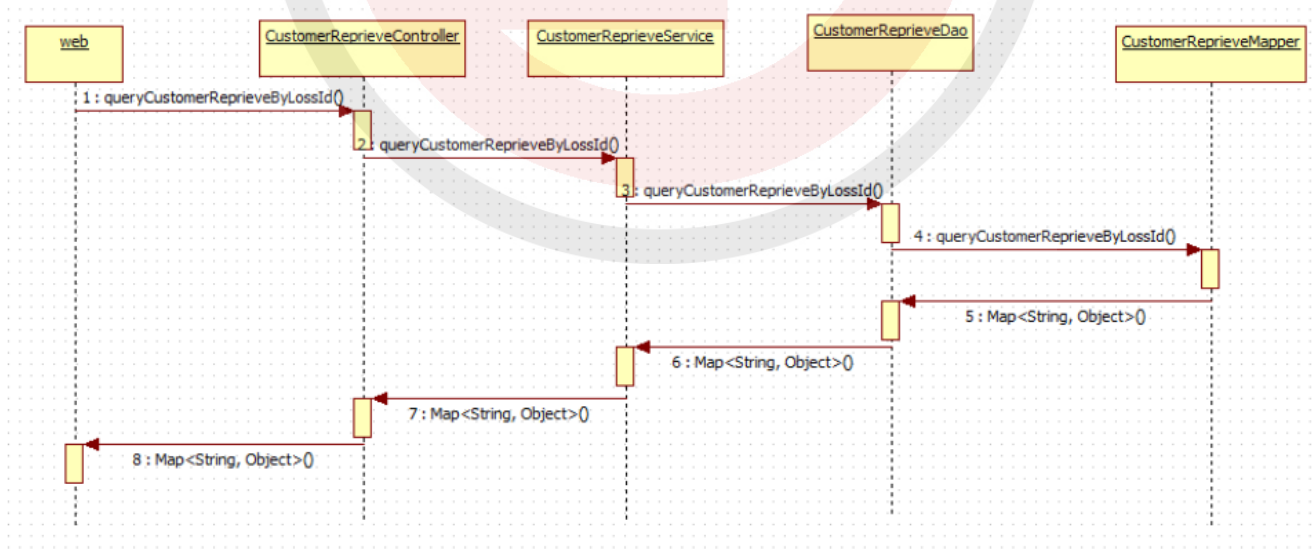
客户名称:	<input type="text" value="风驰科技"/>	客户编号	<input type="text" value="KH20150526073022"/>
客户经理:	<input type="text" value="小红"/>	上次下单时间:	<input type="text"/>
确认流失时间:	<input type="text"/>		
流失原因:	<input type="text"/>		

暂缓流失列表

编号	暂缓措施

10 ▾ < << 第 1 共1页 >> > 显示0到0,共0记录

3 流程图



4 编写Controller

```

@RequestMapping("toRepreivePage/{lossId}")
public String toRepreivePage(@PathVariable("lossId") Integer lossId, Model model) {
    model.addAttribute("customerLoss",
        customerLossService.queryCusstomerLossById(lossId));
}
  
```



```
        return "customer_repri";
    }

    @Controller
    @RequestMapping("customer_repri")
    public class CustomerReprieveController extends BaseController {

        @Resource
        public CustomerReprieveService customerReprieveService;

        @RequestMapping("customerReprieveByLossId")
        @ResponseBody
        public Map<String,Object> queryCustomerReprieveById(CustomerRepriQuery
customerRepriQuery){

            return
customerReprieveService.queryCustomerReprieveByLossId(customerRepriQuery);
        }
    }
```

5 编写Service

```
public Map<String,Object> queryCusstomerLossById(Integer id){
    Map<String, Object> stringObjectMap = customerLossDao.queryById(id);
    return stringObjectMap;
}

@Service
public class CustomerReprieveService {
    @Resource
    public CustomerReprieveDao customerReprieveDao;

    @Resource
    public CustomerLossService customerLossService;

    public Map<String,Object> queryCustomerReprieveByLossId(CustomerRepriQuery
customerRepriQuery){

        PageHelper.startPage(customerRepriQuery.getPage(),customerRepriQuery.getRows());
        List<CustomerReprieve> orderList =
customerReprieveDao.queryRepriByLossId(customerRepriQuery.lossId);
        PageInfo<CustomerReprieve> pageInfo = new PageInfo<>(orderList);
        Map<String,Object> map = new HashMap<>();
        map.put("total",pageInfo.getTotal());
        map.put("rows",pageInfo.getList());
        return map;
    }
}
```

6 编写Dao

```
public Map<String,Object> queryById(Integer id);

    public interface CustomerReprieveDao {
        public List<CustomerReprieve> queryRepriByLossId(Integer lossId);
    }
```

7 编写Mapper

```
<select id="queryById" parameterType="int" resultType="map">
    select
    id,cus_name as cusName,
    cus_no as cusNo,
    cus_manager as cusManager,
    state,loss_reason as lossReason,
    DATE_FORMAT(last_order_time,'%Y-%m-%d') as lastOrderTime,
    DATE_FORMAT(confirm_loss_time,'%Y-%m-%d') as confirmLosTime
    from t_customer_loss
    where
    is_valid =1 and id = #{id}
</select>

<select id="queryRepriByLossId" parameterType="java.lang.Integer"
resultMap="BaseResultMap">
    select
    <include refid="Base_Column_List" />
    from t_customer_reprieve
    where loss_id = #{id,jdbcType=INTEGER} and is_valid = 1
</select>
```

8 编写Query

```
public class CustomerRepriQuery extends BaseQuery {

    public Integer lossId;

    public Integer getLossId() {
        return lossId;
    }

    public void setLossId(Integer lossId) {
        this.lossId = lossId;
    }
}
```

9 前台js



```
$(function () {  
  
    var lossId = $("#lossId").val();  
    $("#dg").datagrid({  
        url: ctx + "/customer_repri/customerRepriveByLossId?lossId=" + lossId,  
        saveUrl: ctx + "/customer_repri/insertReprive?lossId=" + lossId,  
        updateUrl: ctx + "/customer_repri/updateReprive?lossId=" + lossId  
    })  
    var state = $("#state").val();  
    if (state == 1) {  
        $("#toolbar").remove();  
        $("#dg").datagrid("disableEditing");  
    }  
});
```

五 添加客户流失暂缓措施

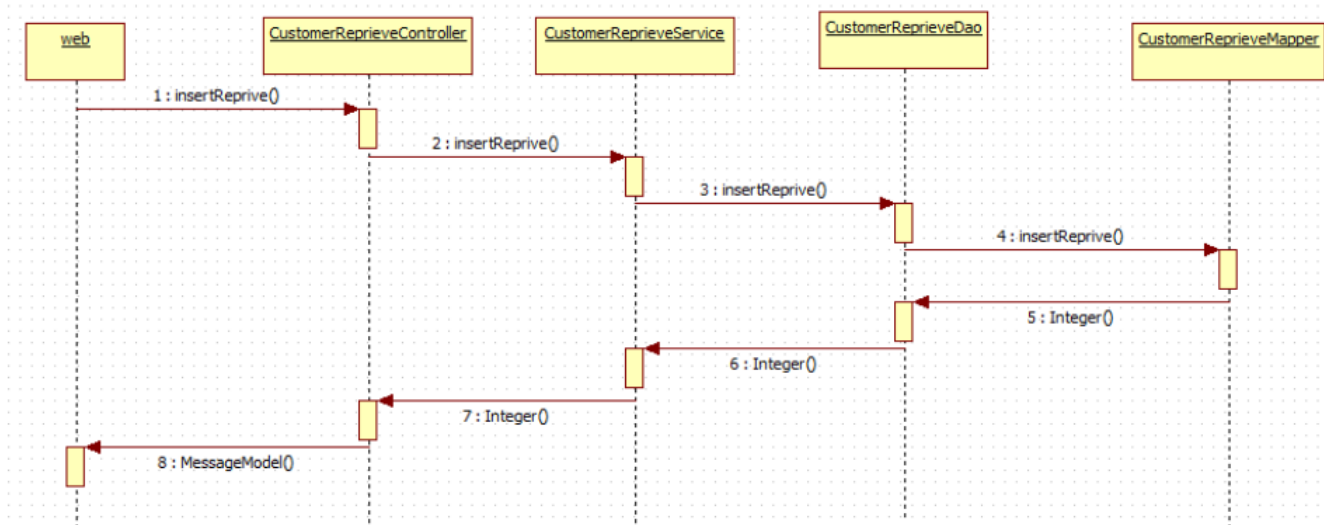
1 需求分析

添加客户流失暂缓措施方案

2 页面原型

暂缓流失列表	
 添加暂缓措施  删除暂缓措施  保存暂缓措施  撤销行  确认流失	
编号	暂缓措施
1	<div></div>

3 流程图



4 编写Controller

```

@RequestMapping("insertReprive")
@ResponseBody
public MessageModel addCustomerReprive(CustomerReprive customerReprive){
    customerRepriveService.insert(customerReprive);
    return success("添加暂缓措施成功");
}

```

5 编写Service

```

/**
 * 1.流失ID 暂缓措施 必须存在
 * 2.添加 额外参数
 * @param customerReprive
 */

public void insert(CustomerReprive customerReprive){
    checkParams(customerReprive.getLossId(),customerReprive.getMeasure());
    customerReprive.setIsValid(1);
    customerReprive.setCreateDate(new Date());
    customerReprive.setUpdateDate(new Date());
    AssertUtil.isTrue(customerRepriveDao.insert(customerReprive)<1,"暂缓措施添加失败");
}

public void checkParams(Integer lossId,String measure){
    AssertUtil.isTrue(StringUtils.isBlank(measure),"暂缓措施不能为空");
    Map<String,Object> map = customerLossService.queryCusstomerLossById(lossId);
    AssertUtil.isTrue(lossId==null||map==null||map.isEmpty(),"流失记录不存在");
}

public Map<String,Object> queryCusstomerLossById(Integer id){
}

```

```
Map<String, Object> stringObjectMap = customerLossDao.queryById(id);  
return stringObjectMap;  
}
```

6 编写Dao

```
public Integer insert(CustomerReprieve customerReprieve);  
  
public Map<String,Object> queryById(Integer id);
```

7 编写Mapper

```
<insert id="insert" parameterType="com.xxx.crm.vo.CustomerReprieve">  
    insert into t_customer_reprieve (id, loss_id, measure,  
    is_valid, create_date, update_date  
    )  
    values (#{id,jdbcType=INTEGER}, #{lossId,jdbcType=INTEGER}, #  
{measure,jdbcType=VARCHAR},  
    #{isValid,jdbcType=INTEGER}, #{createDate,jdbcType=TIMESTAMP}, #  
{updateDate,jdbcType=TIMESTAMP}  
    )  
</insert>  
  
<select id="queryById" parameterType="int" resultType="map">  
    select  
    id,cus_name as cusName,  
    cus_no as cusNo,  
    cus_manager as cusManager,  
    state,loss_reason as lossReason,  
    DATE_FORMAT(last_order_time,'%Y-%m-%d') as lastOrderTime,  
    DATE_FORMAT(confirm_loss_time,'%Y-%m-%d') as confirmLosTime  
    from t_customer_loss  
    where  
    is_valid =1 and id = #{id}  
</select>
```

8 前台js

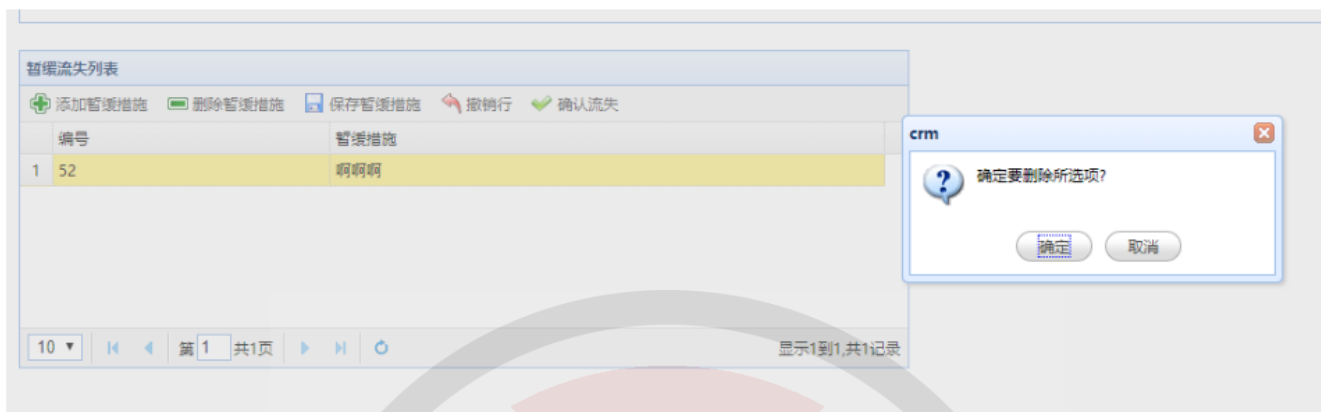
```
function saveCustomerRepri() {  
    $("#dg").datagrid("addRow");  
    $("#dg").datagrid("load");  
}
```

六 删除客户流失暂缓措施

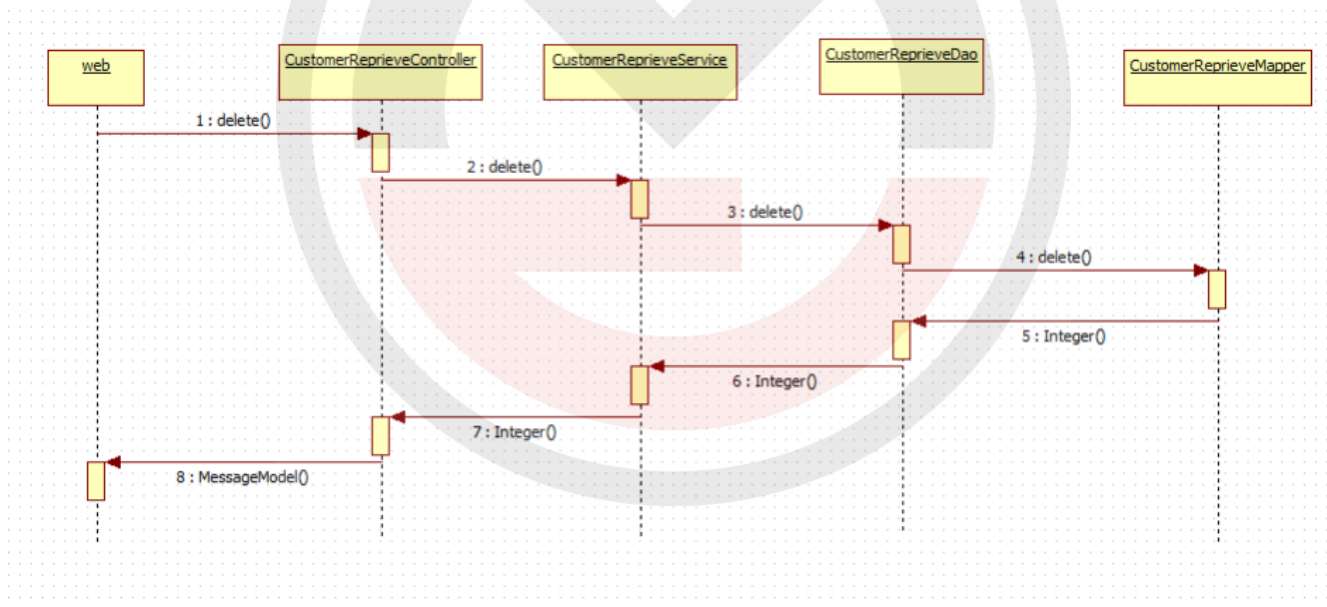
1 需求分析

删除客户流失暂缓措施方案

2 页面原型



3 流程图



4 编写Controller

```
@RequestMapping("delete")
@ResponseBody
public MessageModel delete(Integer id){
    customerReprieveService.delete(id);
    return success("删除暂缓措施成功");
}
```

5 编写Service

```
public void delete(Integer id){  
    AssertUtil.isTrue(customerReprieveDao.delete(id)<1,"删除暂缓措施失败");  
}
```

6 编写Dao

```
public Integer delete(Integer id);
```

7 编写Mapper

```
<delete id="delete" parameterType="java.lang.Integer">  
    update t_customer_reprieve  
    set is_valid = 0  
    where id = #{id,jdbcType=INTEGER}  
</delete>
```

8 前台js

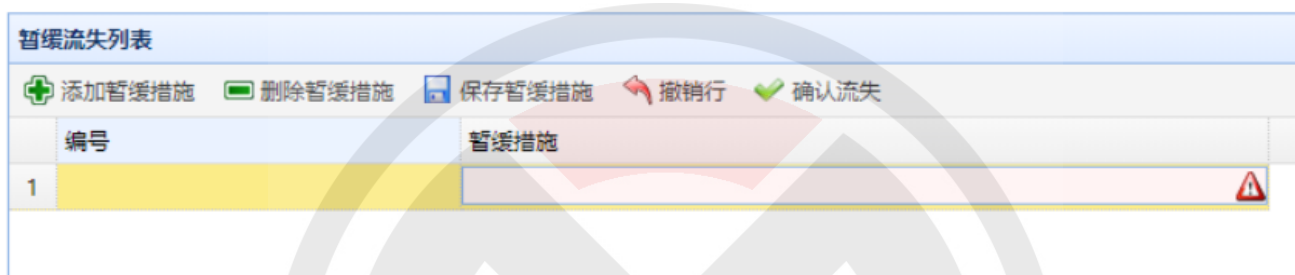
```
function delCustomerRepri() {  
    var rows = $("#dg").datagrid("getSelections");  
  
    if (rows.length == 0) {  
        $.messager.alert("来自crm", "请先选择要删除的记录", "info");  
        return;  
    }  
  
    $.messager.confirm("crm", "确定要删除所选项?", function (r) {  
        if (r) {  
            $.ajax({  
                type: "post",  
                url: ctx + "/customer_repri/delete",  
                data: "id=" + rows[0].id,  
                dataType: "json",  
                success: function (data) {  
                    $.messager.alert("来自crm", data.msg, "info");  
                    if (data.code == 200) {  
                        $("#dg").datagrid("load");  
                    }  
                }  
            })  
        }  
    })  
}
```

七 更新客户流失暂缓措施

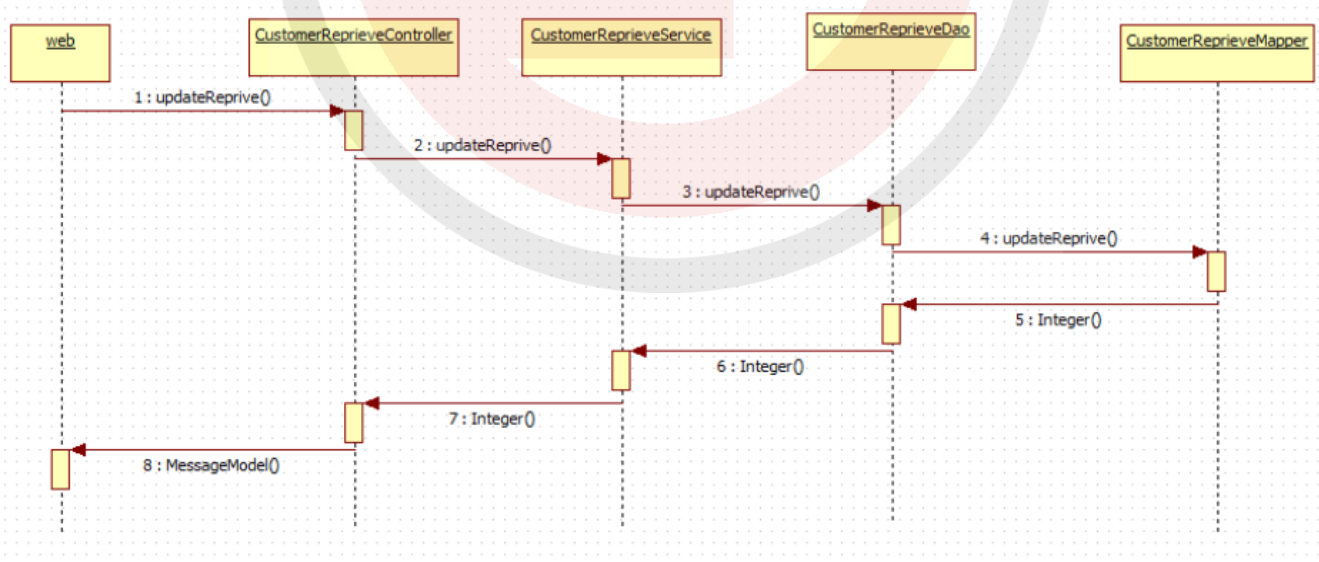
1 需求分析

更新客户流失暂缓措施方案

2 页面原型



3 流程图



4 编写Controller



```
@RequestMapping("updateReprive")
@ResponseBody
public MessageModel updateCustomerReprive(CustomerReprive customerReprive){
    customerRepriveService.update(customerReprive);
    return success("添加暂缓更新成功");
}
```

5 编写Service

```
public void update(CustomerReprive customerReprive){

checkParams(customerReprive.getId(),customerReprive.getLossId(),customerReprive.getMeasure());

    customerReprive.setUpdateDate(new Date());
    AssertUtil.isTrue(customerRepriveDao.update(customerReprive)<1,"暂缓措施修改失败");

}
```

6 编写Dao

```
public Integer update(CustomerReprive customerReprive);
```

7 编写Mapper

```
<update id="update" parameterType="com.xxx.crm.vo.CustomerReprive">
    update t_customer_reprive
    <set>
        <if test="lossId != null">
            loss_id = #{lossId,jdbcType=INTEGER},
        </if>
        <if test="measure != null">
            measure = #{measure,jdbcType=VARCHAR},
        </if>
        <if test="isValid != null">
            is_valid = #{isValid,jdbcType=INTEGER},
        </if>
        <if test="createDate != null">
            create_date = #{createDate,jdbcType=TIMESTAMP},
        </if>
        <if test="updateDate != null">
            update_date = #{updateDate,jdbcType=TIMESTAMP},
        </if>
    </set>
    where id = #{id,jdbcType=INTEGER}
</update>
```

八 确认客户流失

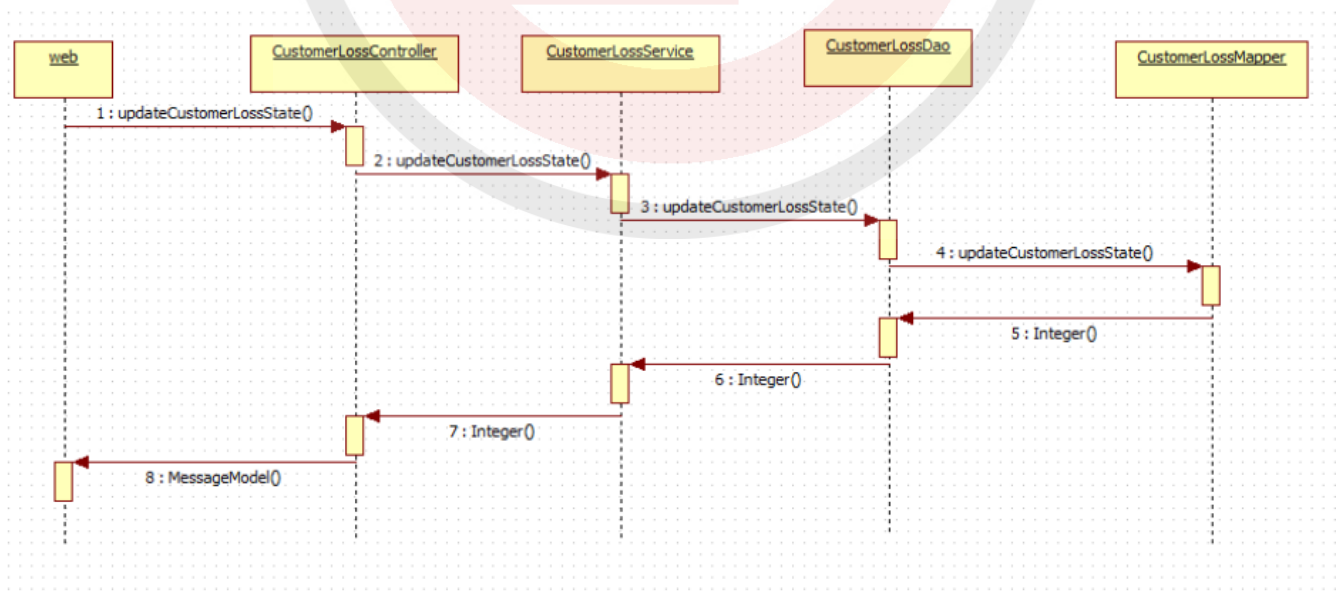
1 需求分析

确认客户已经流失

2 页面原型



3 流程图



4 编写Controller



```
@RequestMapping("updateCustomerLossState")
@ResponseBody
public MessageModel updateCustomerLossState(Integer lossId, String lossReason){
    customerLossService.updateCustomerLossState(lossId, lossReason);
    return success("流失操作执行成功");
}
```

5 编写Service

```
public void updateCustomerLossState(Integer id,String reason){
    AssertUtil.isTrue(StringUtils.isBlank(reason),"流失原因不能为空");
    Map<String,Object> map = queryCusstomerLossById(id);
    AssertUtil.isTrue(map==null||map.isEmpty(),"流失记录不存在");

    AssertUtil.isTrue(customerLossDao.updateCustomerLossState(id,reason) <1,"操作失
败");
}
```

6 编写Dao

```
@Update("update t_customer_loss set state=1 , confirm_loss_time = now(), loss_reason
= #{reason} where id = #{id} ")
public Integer updateCustomerLossState(@Param("id") Integer id, @Param("reason")
String reason);
```

7 前台js

```
function updateCustomerLossState() {

$.messager.confirm("crm", "确定该客户已流失?", function (r) {
    if (r) {
        $.messager.prompt("crm", "请输入流失原因", function (msg) {
            if (msg) {
                $.ajax({
                    type: "post",
                    url: ctx + "/customer_loss/updateCustomerLossState",
                    data: "lossId=" + $("#lossId").val() + "&lossReason=" + msg,
                    dataType: "json",
                    success: function (data) {
                        $.messager.alert("来自crm", data.msg, "info");
                        if (data.code == 200) {
                            $("#toolbar").remove();
                        }
                    }
                })
            }
        })
    }else {
        $.messager.alert("来自crm", "流失原因不能为空!!", "info");
    }
}
```

```
        }  
    }  
  
    }  
}
```

