

# 需求规格说明

变更人	变更时间	变更内容	版本号
孙宇鹏	2022.07.08	添加功能需求	V0.0.1
孙宇鹏	2022.07.09	更新了用户界面	V0.0.2

## 需求规格说明

- 1.引言
  - 1.1 目的
  - 1.2范围
  - 1.3参考文献
- 2.总体描述
  - 2.1 项目前景
    - 2.1.1 背景与机遇
    - 2.1.2 业务需求
  - 2.2 项目功能
  - 2.3 用户特征
  - 2.4 约束
  - 2.5 假设与依赖
- 3.详细需求描述
  - 3.1 对外接口需求
    - 3.1.1 用户界面
      - UI1 账户管理
      - UI2 制定收款单
      - UI3 制定付款单
      - UI4 制定现金费用单
      - UI5 查看经营状况表
      - UI6 查看经营历程表
      - UI7 查看销售明细表
      - UI8 员工打卡
      - UI9 员工信息
      - UI10 员工打卡查询
      - UI11 促销策略
    - 3.1.2 通信接口
    - 3.1.3 软件接口
  - 3.2 功能需求
    - 3.2.1 账户管理
      - 3.2.1.1 特征
      - 3.2.1.2 刺激/响应序列
      - 3.2.1.3 相应功能需求
    - 3.2.2 制定收款单
      - 3.2.2.1 特征
      - 3.2.2.2 刺激/响应序列
      - 3.2.2.3 相应功能需求
    - 3.2.3 制定付款单
      - 3.2.3.1 特征
      - 3.2.3.2 刺激/响应序列
      - 3.2.3.3 相应功能需求
    - 3.2.4 制定工资单
      - 3.2.4.1 特征

- 3.2.4.2 刺激/响应序列
    - 3.2.4.3 相应功能需求
  - 3.2.5 查看销售明细表
    - 3.2.5.1 特征:
    - 3.2.5.2 刺激/响应序列
    - 3.2.5.3 相应功能需求
  - 3.2.6 查看经营状况表
    - 3.2.6.1 特征:
    - 3.2.6.2 刺激/响应序列
    - 3.2.6.3 相应功能需求
  - 3.2.7 查看经营历程表
    - 3.2.7.1 特征
    - 3.2.7.2 刺激/响应序列
    - 3.2.7.3 相应功能需求
  - 3.2.8 员工管理
    - 3.2.8.1 特征:
    - 3.2.8.2 刺激/响应序列
    - 3.2.8.3 相应功能需求
  - 3.2.9 员工打卡
    - 3.2.9.1 特征
    - 3.2.9.2 刺激/响应序列
    - 3.2.9.3 相应功能需求
  - 3.2.10 薪酬规则制定
    - 3.2.10.1 特征
    - 3.2.10.2 刺激/响应序列
    - 3.2.10.3 相应功能需求
  - 3.2.11 制定促销策略
    - 3.2.11.1 特征:
    - 3.2.11.2 刺激/响应序列:
    - 3.2.11.3 相应功能需求
- 3.3 非功能性需求
  - 3.3.1 安全性
  - 3.3.2 可维护性
  - 3.3.3 易用性
  - 3.3.4 可靠性
  - 3.3.5 业务规则
  - 3.3.6 约束
- 3.4 数据需求
  - 3.4.1 数据定义
  - 3.4.2 默认数据
  - 3.4.3 数据格式要求
- 3.5 其他

# 1.引言

---

## 1.1 目的

一民营灯具开关企业因规模不断扩大，业务不断增长，因此决定开发一套ERP系统，方便本公司进行企业管理。这是一个为库存管理人员、销售人员、销售经理、财务人员、人力资源人员、总经理等不同职务的员工提供库存管理、销售管理、财务管理、人事管理、企业经营管理等功能的平台，能够很大程度地方便公司进行企业管理。

## 1.2范围

NJU ERP系统是一个在线企业管理网站，开发的目的是帮助企业的员工进行线上企业管理，主要的业务包括（库存管理人员）库存管理、（销售人员、销售经理）销售管理、（财务人员）财务管理、（人力资源人员）人事管理、（总经理）企业经营管理。

## 1.3参考文献

1. IEEE标准。
2. 《软件工程与计算（卷二）：软件开发的技术基础》刘钦、丁二玉著。
3. NJU ERP系统用例文档。
4. MSCS软件需求规格文档。

## 2.总体描述

### 2.1 项目前景

#### 2.1.1 背景与机遇

- 公司规模不断扩大，企业业务量、办公场所、员工数都发生增长；
- ERP系统日趋流行；
- NJU ERP系统可以帮助企业不同职位的员工协同线上办公，很大程度的提高了企业办公的效率，加强了办公的可靠性。

#### 2.1.2 业务需求

**BR1:**在使用系统一周后，库存管理人员可以熟练地进行商品管理，库存出入、查看、盘点。

**BR2:**在使用系统一周后，销售人员、销售经理可以熟练地进行销售管理、客户管理、审批单据。

**BR3:**在使用系统一周后，财务人员可以熟练地进行账户管理，制定收款单、付款单、现金费用单，查看经营状况表、经营历程表、销售明细表。

**BR4:**在使用系统一周后，人力资源人员可以熟练地进行员工的增删改查、员工的打卡管理。

**BR5:**在使用系统一周后，总经理可以熟练地进行销售管理，增删改查促销策略，审批单据，查看经营状况表、经营历程表、销售明细表。

**BR6:**在使用系统一周后，除总经理外所有员工可以熟练地使用ERP系统进行线上签到打卡。

**BR7:**系统上线运行六个月后，积压的库存减少40%

- 最好情况：60%
- 最坏情况：20%

**BR8:**系统上线运行六个月后，销售额提高20%

- 最好情况：40%
- 最坏情况：10%

### 2.2 项目功能

本阶段产品主要有一下新增功能，随着版本更新，功能可能会有增删。

**SF1:**库存管理人员可以进行商品管理和商品分类管理

**SF2:**库存管理人员可以查看库存情况，管理商品出入库，进行库存盘点

**SF3:**销售人员、销售经理可以进行进货、退货的管理，制定销售单据，增删改查客户，查看促销策略

**SF4:**财务人员可以进行账户管理，制定收款单、付款单、现金费用单

**SF5:**财务人员和总经理可以查看经营状况表、经营历程表、销售明细表

**SF6:**人力资源人员可以进行员工的增删改查，员工打卡记录查询

**SF7:**总经理可以进行销售类单据的审批，促销策略的增删改查

## 2.3 用户特征

用户	用户特征
库存 管理 人员	是企业内的库存管理人员，负责对商品分类和商品的信息的管理。能够简单使用办公信息化系统，对新系统持积极态度，不希望增加现有的工作量。
进货 销售 人员	负责完成进货单和销售单。对新系统基本持积极态度。不希望增加现有工作量。
财务 人员	办公信息化系统较强。对新系统持积极态度。不希望增加现有工作量。
人力 资源 人员	办公信息化系统较强。对新系统持积极态度。不希望增加现有工作量。
总经 理	能够熟练使用办公信息化系统。对新系统持积极态度。

## 2.4 约束

**CON1:** 企业工作人员使用WEB。

**CON2:** 开发者需要在3个月内完成开发任务和测试。

**CON3:** 在开发中，开发者需要提交软件需求规格说明文档、设计描述文档。

## 2.5 假设与依赖

**AE1:** 每天平台登录用户数不会有太大的波动，系统被访问次数稳定。

# 3.详细需求描述

## 3.1 对外接口需求

### 3.1.1 用户界面

UI总体风格：简约直观扁平化设计，帮助用户快速获取到需要的内容，高效率获取所需信息。注意，在不同角色登录状态下，鼠标悬浮于左下角用户名上时会显示角色名。

【以下展示页面均在Chrome浏览器上进行，在MacOS Safari等浏览器上的展示与下图无异】

UI1 账户管理

系统使用简约直观扁平化设计，帮助用户快速定位查询账户、添加账户、编辑账户、删除账户等。

界面图示为：

账户管理

查询账户

点击查询

添加账户

账户1

账户名称: 1

账户余额: 2

操作: [编辑](#) [删除](#)

账户2

账户名称: 1

账户余额: 2

操作: [编辑](#) [删除](#)

- UI1.1 用户输入账户名称后，点击“点击查询”即可显示对应账户信息
- UI1.2 用户点击“添加账户”后，即可设置账户名称即余额，进行账户添加
- UI1.3 用户点击编辑，即可编辑已有账户的名称（余额不可修改）
- UI1.4 用户点击删除，即可删除已有的账户

UI2 制定收款单

系统使用简约直观扁平化设计，帮助用户快速定位设置收款单信息，制定收款单等。

界面图示为：

制定收款单

单据编号

客户

操作员

3

转账列表:

银行账户

转账金额

备注

总额汇总

提交

- UI2.1 用户输入收款单信息后，点击“提交”，即可新增一个收款单

### UI3 制定付款单

系统使用简约直观扁平化设计，帮助用户快速定位设置付款单信息，制定付款单等。

界面图示为：

制定付款单

单据编号

客户

操作员

3

转账列表:

银行账户

转账金额

备注

总额汇总

提交

- UI3.1 用户输入付款单信息后，点击“提交”，即可新增一个付款单

### UI4 制定现金费用单

系统使用简约直观扁平化设计，帮助用户快速定位设置现金费用单信息，制定现金费用单等。

界面图示为：

# 制定现金费用单

单据编号

操作员

3

银行账户

条目清单:

条目名

金额

备注

总额

提交

- UI4.1 用户输入现金费用单信息后，点击“提交”，即可新增一个现金费用单

## UI5 查看经营状况表

系统使用简约直观扁平化设计，帮助用户快速定位设置起止时间，查看经营状况表等。

界面图示为：

## 经营情况表

请选择一个时间段:



开始日期

至

结束日期

查询

导出Excel

allowance

cashCoupon

costOfSales

costPriceAdjustmentIncome

destroy

gift

🕒

开始日期

至

结束日期

查询

📄 导出Excel

2022-07-01

00:00:00

>

2022-08-12

00:00:00

« <

2022 年 7 月

> »

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

2022 年 8 月

> »

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

清空

确定

- UI5.2 用户点击“查询”，即可显示对应时间内的经营情况
- UI5.3 用户点击“导出EXCEL”，可以将表格中数据导出为.xlsx文件

#### UI6 查看经营历程表

系统使用简约直观扁平化设计，帮助用户快速定位设置筛选信息，查看经营历程表等。

界面图示为：

# 经营历程表

请选择一个时间段:

🕒

开始日期

至

结束日期

请选择单据类型:

选择单据类型

▼

请选择客户:

选择客户

▼

请输入业务员:

null

确认

📄 导出Excel

- UI6.1 用户可以在时间选择框中选择起止时间，点击“确定”后设置起止时间



🕒

开始日期

至

结束日期

查询

📄 导出Excel

2022-07-01

00:00:00

>

2022-08-12

00:00:00

« <

2022 年 7 月

> »

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

2022 年 8 月

> »

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

清空

确定

- UI6.2 用户可以在下拉框中选择单据类型和客户，如不选择则为任意
- UI6.3 用户在输入框中输入业务员ID，可以筛选制定业务员操作的单据
- UI6.4 用户点击“确认”，即可显示对应时间内的经营历程
- UI6.5 用户点击“导出EXCEL”，可以将表格中数据导出为.xlsx文件

## UI7 查看销售明细表

系统使用简约直观扁平化设计，帮助用户快速定位设置筛选信息，查看销售明细表等。

界面图示为：

请选择一个时间段: 🕒

开始日期

至

结束日期

请选择商品名:

选择商品名

▼

请选择客户:

选择客户

▼

请输入业务员:

null

确认

📄 导出Excel

- UI7.1 用户可以在时间选择框中选择起止时间，点击“确定”后设置起止时间

🕒

开始日期

至

结束日期

查询

导出Excel

2022-07-01

00:00:00

>

2022-08-12

00:00:00

« <

2022 年 7 月

> »

日	一	二	三	四	五	六
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

2022 年 8 月

> »

日	一	二	三	四	五	六
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

清空

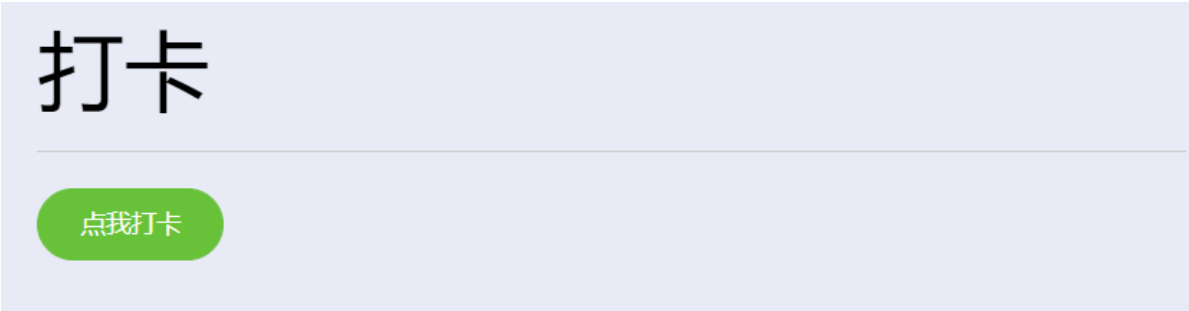
确定

- UI7.2 用户可以在下拉框中选择单据类型和客户，如不选择则为任意
- UI7.3 用户在输入框中输入业务员ID，可以筛选制定业务员操作的单据
- UI7.4 用户点击“确认”，即可显示对应时间内的经营历程
- UI7.5 用户点击“导出EXCEL”，可以将表格中数据导出为.xlsx文件

UI8 员工打卡

系统使用简约直观扁平化设计，帮助用户快速定位打卡功能。

界面图示为：



- UI8.1 用户点击“点我打卡”按钮，即可进行当日的打卡
  - UI8.1.1 用户若当日首次打卡，会提示打卡成功

localhost:8000 显示

打卡成功



- UI8.1.2 用户若当日重复打卡，则会提示打卡失败

localhost:8000 显示

打卡失败

确定

UI9 员工信息

系统使用简约直观扁平化设计，帮助用户快速定位员工信息的增删改查。

界面图示为：



- UI9.1 点击“新增员工”，即可添加员工信息
  - UI9.1.1 用户可以在新增员工页面设置员工信息，点击“确定”后添加

员工信息

id

姓名

性别

选择员工性别

▼

生日

电话

岗位

选择用户级别

▼

- UI9.1.2 添加成功后，新增用户的信息会在页面显示



UI10 员工打卡查询

系统使用简约直观扁平化设计，帮助用户快速定位员工打卡信息查询。

界面图示为：

打卡

请输入员工用户名

查询

- UI10.1 用户在输入框内输入用户名，点击查询，即可显示用户的打卡信息

打卡

请输入员工用户名

rlzyry

查询

打卡记录	员工	打卡时间
123	rlzyry	2002-07-03T16:00:00.000+00:00
125	rlzyry	2022-07-03T16:00:00.000+00:00
127	rlzyry	2022-07-06T16:00:00.000+00:00
129	rlzyry	2022-07-08T16:00:00.000+00:00

UI11 促销策略

系统使用简约直观扁平化设计，帮助用户快速定位促销策略的增、删、查。

界面图示为：

促销策略

增加策略

请选择促销策略种类：

选择促销种类

提交

查看策略

选择策略类型：

选择促销种类

查看

删除策略

选择策略类型：

选择促销种类

输入策略id：

0

删除

- UI11.1.1 用户在增加策略模块中选择不同的策略种类，会显示对应的不同UI

### 增加策略

请选择促销策略种类：

用户级别

提交

请输入策略id：

0

请输入开始时间：

请输入结束时间：

请选择客户级别：

选择用户级别

请输入折扣 (0.01-0.99)：

0

请输入代金券金额：

0

### 增加策略

请选择促销策略种类：

特价包

提交

请输入策略id：

0

请输入开始时间：

请输入结束时间：

请输入优惠金额：

0

请输入关联商品id：

### 增加策略

请选择促销策略种类：

总价满减

提交

请输入策略id：

0

请输入开始时间：

请输入结束时间：

满减价格：

0

请输入代金券金额：

0

- UI11.1.2 用户输入对应的策略信息，点击“提交”后，即可新建对应的促销策略
- UI11.1.3 用户在查看策略模块选择策略类型，点击“查看”，即可显示对应类型的所有已制定的策略

#### 查看策略

选择策略类型：

用户级别

查看

策略id	开始时间	结束时间	用户级别	折扣	代金券金额
12	2022-01-03T16:00:00.000+00:00	2022-01-04T16:00:00.000+00:00	3	0.7	0

- UI11.1.4 用户在删除策略模块，选择策略类型，并输入策略ID，点击删除，即可删除指定促销策略

### 3.1.2 通信接口

后端使用RESTful的接口，由前端使用get和post请求调用并返回

### 3.1.3 软件接口

使用MySQL作为底层数据库存储各类数据

## 3.2 功能需求

### 3.2.1 账户管理

#### 3.2.1.1 特征

财务人员可以在账户管理界面设置账户名、余额信息来进行增加账户。财务人员输入账户名，系统根据账户名，显示对应账户信息，若账户名不存在则不显示。财务人员修改想要修改的账户信息，仅可修改账户名，余额不可修改。成功则在系统修改对应账户信息，失败则提示用户失败。财务人员删除想要删除的账户。成功则在系统中删除这条账户信息，并提示用户删除成功。失败则提示用户删除失败。

优先级=高

#### 3.2.1.2 刺激/响应序列

- 增加账户

刺激：用户请求增加账户

响应：系统显示增加用户界面

刺激：用户设置账户信息并提交

响应：系统根据用户设置的信息，显示结果

- 查看账户

刺激：用户请求查看账户信息

响应：系统根据账户名，显示账户信息。若账户不存在，则不显示

- 修改账户信息

刺激：用户请求修改账户信息

响应：系统显示修改界面

刺激：用户修改信息后，提交修改

响应：系统根据用户提交信息，返回修改结果。

- 删除账户

刺激：用户请求删除账户信息

响应：系统根据用户请求，返回删除结果

#### 3.2.1.3 相应功能需求

<b>accountService.addBankAccount</b>	<b>根据用户设置的信息增加账户</b>
accountService.showBankAccount	根据用户设置的信息显示账户信息
accountService.updateBankAccount	根据账户名修改账户信息
accountService.deleteBankAccount	根据账户名删除账户信息
Response.buildSuccess	返回增加账户的结果

### 3.2.2 制定收款单

#### 3.2.2.1 特征

财务人员设置收款单信息，请求新增收款单。

成功则在系统中新增收款单信息，并提示用户增加成功。失败则提示用户添加失败。

优先级=高

#### 3.2.2.2 刺激/响应序列

刺激：财务人员向系统申请录入收款单

响应：系统显示一份空白的收款单

刺激：财务人员选择单据类型并录入相关信息

响应：系统 根据所选的单据类型以及录入的信息生成一张新的单据

刺激：财务人员选择向单据里新增一条转账列表

响应：系统显示一条新的转账列表记录

刺激：财务人员选择银行账户、输入转账金额和备注

响应：系统显示已输入的转账信息，计算并显示转账总金额

刺激：财务人员确认相关信息并提交

响应：系统提交单据给总经理进行审批

#### 3.2.2.3 相应功能需求

<b>receiptService.makeReceipt</b>	<b>制定收款单</b>
Response.buildSuccess	返回制定收款单的结果

### 3.2.3 制定付款单

#### 3.2.3.1 特征

财务人员设置付款单信息，请求新增付款单。

成功则在系统中新增付款单信息，并提示用户增加成功。失败则提示用户添加失败。

优先级=高

### 3.2.3.2 刺激/响应序列

刺激：财务人员向系统申请录入付款单

响应：系统显示一份空白的付款单

刺激：财务人员选择单据类型并录入相关信息

响应：系统根据所选的单据类型以及录入的信息生成一张新的单据

刺激：财务人员选择增加一条条目清单记录

响应：系统生成一条空白的条目清单记录

刺激：财务人员输入条目名、金额和备注

响应：系统显示已输入的条目列表，计算并显示总金额

响应：系统提交单据给总经理进行审批

### 3.2.3.3 相应功能需求

<code>paymentOrderService.makePaymentOrder</code>	制定付款单
<code>Response.buildSuccess</code>	返回制定付款单的结果

## 3.2.4 制定工资单

### 3.2.4.1 特征

财务人员设置工资单信息，请求新增工资单。

成功则在系统中新增工资单信息，并提示用户增加成功。失败则提示用户添加失败。

优先级=高

### 3.2.4.2 刺激/响应序列

刺激：财务人员向系统申请录入工资单

响应：系统显示一份空白的工资单

刺激：财务人员录入相关信息

响应：系统根据录入的信息生成一张新的单据

刺激：财务人员确认相关信息并提交

响应：系统提交单据给总经理进行审批

### 3.2.4.3 相应功能需求

<code>payrollService.makePayroll</code>	制定工资单
<code>Response.buildSuccess</code>	返回制定工资单的结果

## 3.2.5 查看销售明细表



#### 3.2.5.1 特征:

财务人员、总经理为快速了解公司的销售情况，可以设置指定范围，来查看相应范围内的商品销售情况  
优先级=低

#### 3.2.5.2 刺激/相应序列

刺激：用户选择查看销售明细表

响应：系统显示空白页面，供用户选择查询范围

刺激：用户指定范围，包括时间区间、商户名、客户、业务员等，并提交请求

响应：系统在销售出货单据记录中筛选数据并以列表形式显示

#### 3.2.5.3 相应功能需求

<b>saleDetailsService.showSaleDetailsBySomeConditions</b>	<b>查看销售明细表</b>
Response.buildSuccess	返回查看销售明细表的结果

### 3.2.6 查看经营状况表

#### 3.2.6.1 特征:

财务人员、总经理为快速了解公司的销售情况，可以设置指定范围，来查看相应范围内的公司经营情况  
优先级=低

#### 3.2.6.2 刺激/响应序列

刺激：用户选择查看经营状况表

响应：系统显示空白页面，供用户选择查询范围

刺激：用户指定范围，包括时间区间、商户名、客户、业务员等，并提交请求

响应：系统显示统计信息，包括收入类（销售收入、商品类收入）、支出类（销售成本、商品类支出）、利润。

#### 3.2.6.3 相应功能需求

<b>operationStatementService.showStatementOfOperation</b>	<b>查看经营状况表</b>
Response.buildSuccess	返回查看经营状况表的结果

### 3.2.7 查看经营历程表

#### 3.2.7.1 特征

财务人员、总经理为快速了解公司的销售情况，可以设置指定范围，来查看相应范围内的公司的所有单据

优先级=中

### 3.2.7.2 刺激/响应序列

刺激：用户选择查看经营历程表

响应：系统显示空白页面，供用户选择查询范围

刺激：用户指定范围，包括时间区间、商户名、客户、业务员等，并提交请求

响应：系统显示制定范围内公司的所有单据

### 3.2.7.3 相应功能需求

<b>businessHistoryService.showBusinessHistoryBySomeConditions</b>	<b>查看经营历程表</b>
Response.buildSuccess	返回查看经营历程表的结果

## 3.2.8 员工管理

### 3.2.8.1 特征：

人力资源人员需要管理公司的员工，对员工进行增删改查

优先级 = 高

### 3.2.8.2 刺激/响应序列

- 查询帐户信息

刺激：人力资源人员输入相关员工关键字

响应：系统显示该员工信息

- 添加员工信息

刺激：人力资源人员请求添加新员工

响应：系统进入员工信息编辑界面

刺激：人力资源人员编辑员工信息并保存

响应：系统提示添加成功

- 修改员工信息

刺激：人力资源人员请求更改员工信息

响应：系统进入员工信息编辑界面

刺激：人力资源人员更改员工信息并保存

响应：系统提示更新成功

- 删除员工信息

刺激：人力资源人员选中待删除客户员工选择删除

响应：系统询问是否确认删除

刺激：人力资源人员确认删除

响应：系统完成删除操作并更新员工列表

### 3.2.8.3 相应功能需求

employeeService.addEmployee	增加员工
employeeService.deleteEmployee	删除员工
employeeService.updateEmployee	修改员工信息
employeeService.getAllEmployees	获取所有员工信息
Response.buildSuccess	返回对员工增删改查的结果

## 3.2.9 员工打卡

### 3.2.9.1 特征

员工每天需要登录系统打卡。若打卡成功，系统则获取员工用户名和当天日期，在系统内记录打卡记录；若当天该员工已打卡过，则提示打卡失败。

优先级= 高

### 3.2.9.2 刺激/响应序列

刺激：员工登录系统，点击打卡选项

响应：若当日员工未打卡，则系统记录该员工打卡次数，并提示打卡成功；若员工当日已打卡，则提示打卡失败

### 3.2.9.3 相应功能需求

employeeService.addPunch	增加一条员工打卡记录
Response.buildSuccess	返回员工打卡的结果

## 3.2.10 薪酬规则制定

### 3.2.10.1 特征

人力资源人员制定薪酬，设置每个岗位对应的薪酬规则

### 3.2.10.2 刺激/响应序列

刺激：人力资源人员为员工设置薪酬规则并提交

响应：系统设置对应人员薪酬规则

### 3.2.10.3 相应功能需求

employeeService.setEmployeeSalaryCalculatingModeById	修改员工薪酬规则
Response.buildSuccess	返回修改的结果

## 3.2.11 制定促销策略

### 3.2.11.1 特征:

总经理为不同级别的用户或不同总价制定合适的促销策略

### 3.2.11.2 刺激/响应序列:

- 查看促销策略

刺激: 总经理选择查看已有促销策略

响应: 系统显示已有促销策略的表格

- 增加促销策略

刺激: 总经理选择新建一条促销策略, 填写促销策略的内容

响应: 系统保存新建的促销策略单, 并生成相应的库存赠送单

- 更改促销策略

刺激: 总经理选择一条已有的促销策略单

响应: 系统显示选中的促销策略单的详细内容

刺激: 总经理更改促销策略的内容

响应: 系统更新促销策略的状态以及库存赠送单的状态

- 删除促销策略

刺激: 总经理选择一条已有的促销策略单

响应: 系统删除选中的促销策略单, 并删除对应的库存赠送单

### 3.2.11.3 相应功能需求

<code>promotionStrategy.addPromotion</code>	添加促销策略, 采用策略模式, 根据收到的参数, 增加不同类型的促销策略
<code>promotionStrategy.deletePromotionById</code>	根据id, 删除指定的促销策略
<code>promotionStrategy.getPromotions</code>	根据策略类型, 显示指定类型的所有促销策略

## 3.3 非功能性需求

### 3.3.1 安全性

**Safety1:** 系统应该只允许经过验证和授权的用户访问

**Safety2:** 系统应该按照用户的身份验证用户相应的操作权限

**Safety3:** 系统中有默认的管理员账户, 该账户只允许管理员用户修改口令

**Safety4:** 用户只能查看和维护自己的信息 **Safety5:** 为防信息泄露, 用户的账号、密码、姓名(名称)、联系方式必须密文存储

### 3.3.2 可维护性

**Modifiability1:** 当系统要增加促销优惠券的发放, 需要能够在1人一天内完成

**Modifiability2:** 如果系统要制定新的单据, 要能够在1人一天内完成

**Modifiability3:** 如果系统要增加新的员工, 要能够在1人一天内完成

### 3.3.3 易用性

**Usability:** 用户不需要额外的培训，对于任意操作可以在10分钟内完成

### 3.3.4 可靠性

**Reliability:** 当客户端与服务端通信时，如果网络出现故障，系统不能出现故障

### 3.3.5 业务规则

**BR1:** 适用的顾客促销策略（所需客户等级，参照日期）

- 客户等级  $\geq$  所需客户等级
- 开始日期早于等于参照日期
- 结束日期晚于等于参照日期

**BR2:** 适用的总价促销策略（所需商品总价，参照日期）

- 商品总价  $\geq$  所需商品总价
- 开始日期早于等于参照日期
- 结束日期晚于等于参照日期

### 3.3.6 约束

**IC1:** 系统要在网络上分布为一个服务器和多个客户端

**IC2:** 系统要为不同的人员提供各自的部署。

**IC3:** 系统后端使用Java，MySQL，springboot进行开发，前端使用Vue进行开发。

## 3.4 数据需求

### 3.4.1 数据定义

```
bank_account(
    `bank_account_name` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci PRIMARY KEY COMMENT '银行账户名称',
    `balance` decimal(10, 2) NOT NULL COMMENT '账户余额'
)
`receipt_sheet` (
    `id` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
COMMENT 'SKD + 日期 + index = 收款单编号',
    `supplier` int(11) NULL DEFAULT NULL COMMENT '供应商名字',
    `seller` int(11) NULL DEFAULT NULL COMMENT '销售商名字',
    `operator` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT
NULL COMMENT '操作员名字',
    `total_amount` decimal(10, 2) NOT NULL COMMENT '总额汇总',
    `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
DEFAULT NULL COMMENT '单据状态',
    `create_time` datetime(0) NOT NULL COMMENT '创建时间',
    PRIMARY KEY (`id`) USING BTREE
)
`transfer_record` (
    `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增id',
    `receipt_sheet_id` varchar(32) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL COMMENT '对应收款单编号',
    `bank_account_name` varchar(255) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NOT NULL COMMENT '银行账户名称',
    `transfer_amount` decimal(10, 2) NOT NULL COMMENT '转账金额',
```

```

        `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '备注',
        PRIMARY KEY (`id`) USING BTREE
    )
    `payment_order_sheet` (
        `id` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
        COMMENT 'FKD + 日期 + index = 收款单编号',
        `supplier` int(11) NULL DEFAULT NULL COMMENT '供应商名字',
        `seller` int(11) NULL DEFAULT NULL COMMENT '销售商名字',
        `operator` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT
        NULL COMMENT '操作员名字',
        `total_amount` decimal(10, 2) NOT NULL COMMENT '总额汇总',
        `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '单据状态',
        `create_time` datetime(0) NOT NULL COMMENT '创建时间',
        PRIMARY KEY (`id`) USING BTREE
    )
    `cash_expense_sheet`;
    CREATE TABLE `cash_expense_sheet` (
        `id` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
        COMMENT 'XJFKD + 日期 + index = 付款单编号',
        `operator` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT
        NULL COMMENT '操作员名字',
        `bank_account_name` varchar(255) CHARACTER SET utf8mb4 COLLATE
        utf8mb4_general_ci NOT NULL COMMENT '银行账户名称',
        `total_amount` decimal(10, 2) NOT NULL COMMENT '总额',
        `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '单据状态',
        `create_time` datetime(0) NOT NULL COMMENT '创建时间',
        PRIMARY KEY (`id`) USING BTREE
    )
    `term_sheet_record` (
        `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增id',
        `payment_order_id` varchar(32) CHARACTER SET utf8mb4 COLLATE
        utf8mb4_general_ci NOT NULL COMMENT '对应付款单编号',
        `name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
        COMMENT '条目名',
        `money` decimal(10, 2) NOT NULL COMMENT '金额',
        `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '备注',
        PRIMARY KEY (`id`) USING BTREE
    )
    `payroll` (
        `order_id` varchar(32) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT
        NULL COMMENT 'GZD + 日期 + index = 工资单编号',
        `stuff_id` int(11) NOT NULL COMMENT '员工编号',
        `name` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL
        COMMENT '姓名',
        `bank_account_name` varchar(255) CHARACTER SET utf8mb4 COLLATE
        utf8mb4_general_ci NOT NULL COMMENT '银行账户名字',
        `row_wages` decimal(10, 2) NOT NULL COMMENT '应发工资',
        `individual_income_tax` decimal(10, 2) NOT NULL COMMENT '个人所得税',
        `unemployment_insurance` decimal(10, 2) NOT NULL COMMENT '失业保险',
        `housing_provident_fund` decimal(10, 2) NOT NULL COMMENT '住房公积金',
        `tax` decimal(10, 2) NOT NULL COMMENT '扣除税款',
        `real_wages` decimal(10, 2) NOT NULL COMMENT '金额',
        `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '单据状态',

```

```

        `create_time` datetime(0) NOT NULL COMMENT '创建时间',
        PRIMARY KEY (`order_id`) USING BTREE
    )
    `sale_returns_sheet` (
        `id` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '销售退货单id',
        `sale_sheet_id` varchar(31) CHARACTER SET utf8mb4 COLLATE
        utf8mb4_general_ci NULL DEFAULT NULL COMMENT '关联的销售单id',
        `operator` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
        NULL DEFAULT NULL COMMENT '操作员',
        `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '单据状态',
        `create_time` datetime(0) NULL DEFAULT NULL COMMENT '创建时间',
        `total_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '退货的总金额',
        `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
        NULL DEFAULT NULL COMMENT '备注'
    )
    `sale_returns_sheet_content` (
        `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增id',
        `purchase_returns_sheet_id` varchar(31) CHARACTER SET utf8mb4 COLLATE
        utf8mb4_general_ci NULL DEFAULT NULL COMMENT '销售退货单id',
        `pid` char(16) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
        DEFAULT NULL COMMENT '商品id',
        `quantity` int(11) NULL DEFAULT NULL COMMENT '数量',
        `total_price` decimal(10, 2) NULL DEFAULT NULL COMMENT '该商品的总金额',
        `unit_price` decimal(10, 2) NULL DEFAULT NULL COMMENT '该商品的单价',
        `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
        NULL DEFAULT NULL COMMENT '备注',
        PRIMARY KEY (`id`) USING BTREE
    )
    `total_promotion_strategy` (
        `id` int(11) NOT NULL COMMENT '方法编号',
        `condition` decimal(10, 2) NULL
        DEFAULT NULL COMMENT '状况',
        `amount` decimal(10, 2) NULL DEFAULT
        NULL COMMENT '大小',
        `begin_time` DATE NULL DEFAULT NULL
        COMMENT '开始时间',
        `end_time` DATE NULL DEFAULT NULL
        COMMENT '结束时间'
    )
    (
        `id` int(11) NOT NULL COMMENT '方法编号',
        `level` int(11) NOT NULL COMMENT
        '级别',
        `discount` decimal(10, 2) NULL
        DEFAULT NULL COMMENT '折扣',
        `amount` decimal(10, 2) NULL
        DEFAULT NULL COMMENT '大小',
        `begin_time` DATE NULL DEFAULT
        NULL COMMENT '开始时间',
        `end_time` DATE NULL DEFAULT NULL
        COMMENT '结束时间'
    )
    (

```

```

        `id` int(11) NOT NULL COMMENT '方法编号',
        `pid_combination` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '商品',
        `amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '大小',
        `begin_time` DATE NULL DEFAULT NULL COMMENT '开始时间',
        `end_time` DATE NULL DEFAULT NULL COMMENT '结束时间'
    )
    `employee_punch` (
        `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '打卡记录',
        `name` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '打卡员工用户名',
        `punch_time` date NULL DEFAULT NULL COMMENT '打卡时间',
        PRIMARY KEY (`id`) USING BTREE
    )
    `employee` (
        `id` int(11) NOT NULL COMMENT '员工编号',
        `gender` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '性别',
        `birthday` DATE NULL DEFAULT NULL COMMENT '生日',
        `name` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '姓名',
        `phone` varchar(15) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '电话号码',
        `role` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NOT NULL COMMENT '员工岗位',
        `basic_salary` decimal(10, 2) NULL DEFAULT NULL COMMENT '员工基本工资',
        `post_salary` decimal(10, 2) NULL DEFAULT NULL COMMENT '员工岗位工资',
        `salary_granting_mode` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '薪资发放方式',
        `salary_calculating_mode` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '薪资计算方式',
        `account` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '员工账户',
        PRIMARY KEY (`id`) USING BTREE
    )
    `sale_returns_sheet`;
CREATE TABLE `sale_returns_sheet` (
    `id` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '销售退货单id',
    `sale_sheet_id` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '关联的销售单id',
    `operator` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '操作员',
    `state` varchar(31) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '单据状态',
    `create_time` datetime(0) NULL DEFAULT NULL COMMENT '创建时间',
    `total_amount` decimal(10, 2) NULL DEFAULT NULL COMMENT '退货的总金额',
    `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL DEFAULT NULL COMMENT '备注'
)

```



```

`sale_returns_sheet_content` (
  `id` int(11) NOT NULL AUTO_INCREMENT COMMENT '自增id',
  `purchase_returns_sheet_id` varchar(31) CHARACTER SET utf8mb4 COLLATE
utf8mb4_general_ci NULL DEFAULT NULL COMMENT '销售退货单id',
  `pid` char(16) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci NULL
DEFAULT NULL COMMENT '商品id',
  `quantity` int(11) NULL DEFAULT NULL COMMENT '数量',
  `total_price` decimal(10, 2) NULL DEFAULT NULL COMMENT '该商品的总金额',
  `unit_price` decimal(10, 2) NULL DEFAULT NULL COMMENT '该商品的单价',
  `remark` varchar(255) CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci
NULL DEFAULT NULL COMMENT '备注',
  PRIMARY KEY (`id`) USING BTREE
)

```

**DR1:** 系统需存储的数据实体以及关系如上

**DR2:** 系统需要存储系统创建以来的所有单据信息。

**DR3:** 系统需要保证数据不溢出表示范围。

**DR4:** 系统需要存储的数据实体及其关系参见用例文档中的概念类图。

**DR5:** 系统需要存储所有员工信息，打卡记录。

### 3.4.2 默认数据

默认数据用于以下情况：

- 用户在查看经营历程表和销售明细表时，未选择商品名、客户、业务员

**Default1:**商品名默认为所有商品

**Default2:**客户默认为所有客户

**Default3:**业务员默认为所有业务员

### 3.4.3 数据格式要求

**Format1:** 日期的格式必须是yyyy-MM-dd

**Format2:** 所有金额必须是正整数

**Format3:** 所有折扣必须是0.01到0.99之间的浮点数

**Format4:** 一切字符串形式的输入必须满足字符数小于等于511

## 3.5 其他

暂无