都市商圈灵活用工

软件需求规约

版本 <1.3>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 04/06/2019 | 1.0 | 立项 | 全员 |
| 12/06/2019 | 1.1 | 修复数个错别字；添加“用户登录”规约。 | 陈奕君 |
| 16/06/2019 | 1.2 | 修改性能需求相关章节。 | 陈奕君 |
| 03/09/2019 | 1.3 | 修改相关章节 | 周佳懿 |

目录

目录

[1. 简介 4](#_Toc18437523)

[1.1 目的 4](#_Toc18437524)

[1.2 定义、首字母缩写词和缩略语 4](#_Toc18437525)

[1.3 参考资料 4](#_Toc18437526)

[2. 整体说明 4](#_Toc18437527)

[3. 具体需求 4](#_Toc18437528)

[3.1 功能 4](#_Toc18437529)

[3.1.1 Use case 图 4](#_Toc18437530)

[3.1.2 商家注册 规约 5](#_Toc18437531)

[3.1.3 职员注册 规约 6](#_Toc18437532)

[3.1.4 商家登录 7](#_Toc18437533)

[3.1.5 用户登录 7](#_Toc18437534)

[3.1.6 用户信息管理 规约 8](#_Toc18437535)

[3.1.7 用户筛选职位并应聘 规约 9](#_Toc18437536)

[3.1.8 商家管理岗位 规约 10](#_Toc18437537)

[3.1.9 商户信息管理 规约 11](#_Toc18437538)

[3.1.10 商家管理店铺 规约 12](#_Toc18437539)

[3.1.11 商家确认员工打卡信息并结算工资 规约 13](#_Toc18437540)

[3.1.12 与商家结算月末账单 规约 14](#_Toc18437541)

[3.1.13 员工打卡 规约 15](#_Toc18437542)

[3.1.14 商户显示统计信息 规约 16](#_Toc18437543)

[3.1.15 商家筛选简历规约 17](#_Toc18437544)

[3.1.16 管理员管理用户规约 18](#_Toc18437545)

[3.2 易用性 18](#_Toc18437546)

[3.3 可靠性 18](#_Toc18437547)

[3.4 性能 19](#_Toc18437548)

[3.5 可支持性 19](#_Toc18437549)

[3.6 设计约束 19](#_Toc18437550)

[3.7 联机用户文档与使用标准 19](#_Toc18437551)

[3.8 适用的标准 20](#_Toc18437552)

软件需求规约 (简化版)

# 简介

## 目的

本文档描述了都市商圈灵活用工应用软件的软件需求。编写该文档的目的在于明确系统范围，并规范记录该系统的各项需求指标与约束。

## 定义、首字母缩写词和缩略语

本规约中没有需要特殊说明的定义、首字母缩写词与缩略语。

## 参考资料

本规约中部分背景说明描述参考<http://www.199it.com/archives/751736.html>。

# 整体说明

本项目的开发是为构建一个面向都市商圈的灵活用工平台，实现商圈用工兼职信息发布、员工招聘、灵活排班、考勤管理与工资结算基本流程，通过人力资源分时共用、合理分配，提高都市商圈人力资源利用效率。与目前流行的兼职平台（如兼职猫、斗米）相比，本项目面向都市商圈这样一个细分方向，创新点在于灵活排班带来的人力资源分时共用能力、相关岗位推荐能力与即时结算工资的能力，方便劳资双方，减少都市商圈用工成本并提高员工收入。

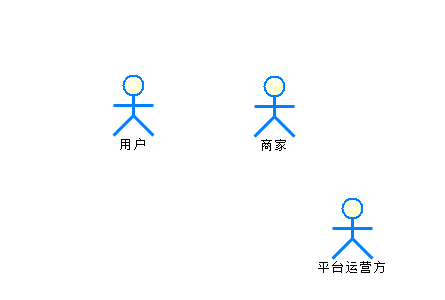


图1 系统角色图

软件的用户定位为企业主或企业人力资源管理人员，作为平台中的资方，与求职应聘者等兼职工作相关人员，作为平台中的劳方。资方有一定的人力资源管理相关的专业能力，能提供企业相关资质与法律证明，负责管理发布的招聘信息、员工信息与财务信息等。劳方为一般的兼职人员，多为18-30岁普通年轻人，有一定文化基础与基本的使用手机与移动互联网应用的能力，对于手机应用的操作没有障碍。

项目预期实现岗位信息管理功能（高优先级）、排班与考勤管理功能（高优先级）、工资结算管理功能（高优先级）、统计信息查看功能（中优先级）、商家店铺管理功能（中优先级）、系统推荐信息管理功能（中优先级）、用户与商家信息管理功能（中优先级）与扩展的考勤管理功能（低优先级）等，作为项目功能性需求。项目采用前后端分离的B-S架构开发，后端采用微服务架构实现。项目的其他非功能性需求可查阅本文档第三节。

# 具体需求

## 功能

### Use case 图

系统用例如下所示。注意，某些包含CRUD的用例中包含CRUD的子流，为简洁，在下图中没有体现。

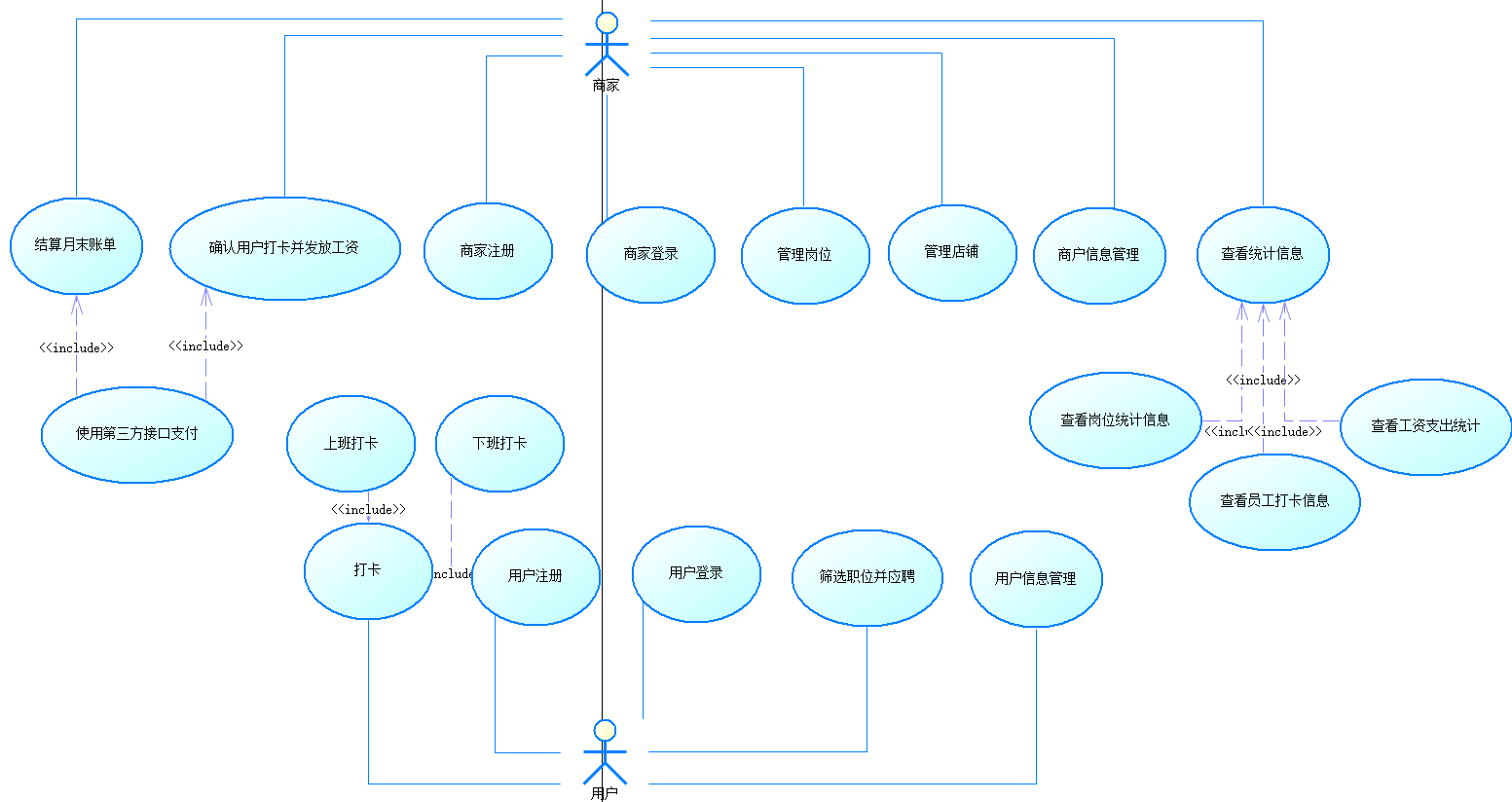


图2 系统用例图

### 商家注册 规约

#### Description

商家根据系统提供的表单，填上相应的信息来进行注册。

#### Actor

主要执行者是任何未注册用户。

#### Flow of Events

事件流从商家打开商家端网页首页开始。

##### Basic Flow

1. 商家选择“商家注册”选项
2. 系统生成供商家注册的表单
3. 商家在表单中填上相应信息并提交
4. 系统验证表单信息的正确性
5. 系统提示注册成功

##### Alternative Flows

1a. 商家进行其他操作

用例结束。

1b. 当前时间不开放注册

系统提示用户“当前时间段不开放注册”，用例结束。

4a. 提交的表单数据不合法

系统提示用户“表单数据不合法，请检查您的输入”，并回到基本流的第二步。

5a. 系统发生内部错误导致注册失败

系统提示用户“很抱歉，注册失败”，并回到基本流的第二步。

#### Special Requirements

无。

#### Pre-Conditions

无。

#### Post-Conditions

#### 商家的注册信息被提交给数据库保存。

#### Extension Points

无。

### 职员注册 规约

#### Description

#### 这个用例解释了用户发起注册的过程。

#### Actor

#### 主要的执行者是未注册的普通用户。

#### Flow of Events

用例从潜在用户在微信中打开微信小程序开始。

##### Basic Flow

1. 系统向微信服务器申请用户授权，显示同意授权界面。

2. 用户点击“同意授权”。

3. 系统通过OAuth2.0协议与微信服务器交互进行验证，获取用户的元数据。

4. 系统向用户显示注册表单，其中需要用户填写的信息包括：用户名、真实姓名、身份证号、年龄、性别、地区、文化程度。

5. 系统核验用户填写的所有信息，并将结果存储到后台数据库。

6. 提示用户注册成功，跳转到"我的"界面。

##### Alternative Flows

1a. 微信服务器返回不正常的结果

系统提示用户“授权异常”，退出用例。

2a. 用户点击“不同意”

退出用例。

3a. 微信服务器返回用户信息时返回异常结果

系统提示用户“用户信息获取失败”，退出用例。

5a. 用户提交的信息格式不正确

系统提示用户“信息不正确，请重新填写”，将不正确的信息标红，回到用例第四步。

#### Special Requirements

无。

#### Pre-Condition

用户从未在小程序中注册过。

#### Post-Condition

新用户信息被写入数据库中。

#### Extension Points

无。

### 商家登录

#### Description

这个用例描述了商家从Web端登录的流程。

#### Actor

该用例的主要执行者是商家。

#### Flow of Events

这个用例从商家点击Web端主页的“登录”按钮开始。

##### Basic Flow

1. 系统显示登录对话框，包含“用户名”、“密码”文本框与“登录”按钮。

2. 商家输入用户名与密码并按下登录按钮

3. 系统验证用户名与密码的正确性

4. 登录成功，系统提示用户。

##### Alternative Flows

2a. 用户在对话框外单击退出对话框

用例终止。

3a. 用户名或密码不正确

系统提示商家“用户名或密码不正确”，回到基本流第一步。

#### Special Requirements

无。

#### Pre-Conditions

商家已打开系统Web端主页。

#### Post-Conditions

商家成功登录系统。

#### Extension Points

无。

### 用户登录

#### Description

这个用例描述了普通用户从微信小程序端登录的流程。

#### Actor

该用例的主要执行者是普通用户。

#### Flow of Events

这个用例从用户打开微信小程序开始。

##### Basic Flow

1. 系统向微信申请授权，并显示微信授权页面。

2. 用户点击同意授权按钮。

3. 系统通过OAuth2.0协议与微信服务器通信，获取用户元数据进行验证。

4. 登录成功，系统提示用户。

##### Alternative Flows

1a. 用户已经授权过

系统直接通过OAuth2.0协议与微信服务器通信，获取用户元数据进行验证。若授权过期则提示用户“授权过期”，转到基本流第二不。若授权成功，则提示用户“登录成功”并退出用例。

3a. 微信服务器返回错误请求

系统提示用户 “授权失败”，退出用例。

#### Special Requirements

无。

#### Pre-Conditions

用户已经在系统中通过微信注册。

#### Post-Conditions

用户成功登录系统，或授权失败。

#### Extension Points

无。

### 用户信息管理 规约

#### Brief Description

这个用例解释了用户如何修改、查询自己的用户信息。

#### Actor

主要执行者为普通用户。

#### Flow of Events

##### Basic Flow

1. 用户选择“用户信息管理”选项
2. 系统显示用户的所有信息，包括用户名、真实姓名、身份证号、年龄、性别等。在每条信息右侧，存在一个“编辑”按钮。
3. 用户点击任意一条信息左侧的“编辑按钮”。
4. 系统弹出修改该条信息的对话框。
5. 用户修改信息并点击“确定”。
6. 系统将更新的信息写入数据库，提示用户“修改成功”。

##### Alternative Flows

1a. 数据库连接失败，系统无法获取消息

提示用户“目前无法获取消息”，用例终止。

1-4a. 用户退出页面

用例终止。

5a. 用户点击“取消”

对话框直接关闭，回到基本流的第二步。

#### Special Requirements

用例需保证，在用户修改信息并点击“确定”前没有任何对持久化数据库的修改。

#### Pre-Conditions

用户已登录系统。

#### Post-Conditions

用户修改后的用户信息提交到数据库保存。

#### Extension Points

无。

### 用户筛选职位并应聘 规约

#### Brief Description

这个用例解释了用户如何筛选职位并应聘岗位。

#### Actor

主要执行者是普通用户。

#### Flow of Events

##### Basic Flow

这个用例从用户选择查看岗位界面开始。

1. 页面显示所有岗位的列表。
2. 如果用户选择“搜索词查询”，那么子流**搜索词查询**开始执行。

如果用户选择“类别查询”，那么子流**类别查询**开始执行。

如果用户选择“查看岗位详情”，那么子流**查看岗位详情**开始执行。

如果用户选择“应聘岗位”，那么子流**应聘岗位**开始执行。

###### 搜索词查询

1. 用户输入搜索词。
2. 系统搜索与搜索词匹配的岗位，并返回岗位列表。

###### 类别查询

1. 用户选择岗位时间，或岗位地点，或岗位薪资，或岗位内容，点击“筛选”。
2. 系统搜索与所选类别匹配的岗位，并返回岗位列表。

###### 应聘岗位

1. 用户选中岗位
2. 用户点击“投递简历”
3. 系统显示“投递成功”

###### 查看岗位详情

1.显示岗位详情页面。

##### Alternative Flows

1. 用户点击取消
   1. 如果在**任意一个**子流中，用户选择取消，那么操作中止，重新从基本流的开头开始。

#### Special Requirements

无。

#### Pre-Conditions

用户需要登陆之后才能执行这个用例。

#### Post-Conditions

无。

#### Extension Points

无。

### 商家管理岗位 规约

#### Brief Description

这个用例解释了商家对其岗位进行增删查改的过程。

#### Actor

主要的Actor是商家。

#### Flow of Events

用例从商家在界面上点击"岗位管理"开始。

##### Basic Flow

* + 1. 系统向商家显示其目前已有的所有岗位信息
    2. 如果商家点击添加，那么子流**添加岗位**开始执行。

如果商家在某一条目上点击编辑，那么子流**编辑岗位**开始执行。

如果商家在某一条目上点击停止招聘，那么子流**停止招聘**开始执行。

###### 添加岗位

1. 系统向商家显示岗位需求表格，需要填写的内容包括：岗位名称、时间、工资、需求。
2. 商家填写表格后点击确定，系统核验内容有效性，并存储到数据库中，跳至2.1.1。

###### 编辑岗位

1. 跳至2.1.2.1，其中表格内的内容为原先的信息。

###### 停止招聘

1. 系统向商家发出确认请求。
2. 商家点击确认，岗位停止接收用户的申请，跳至2.1.1。

#### Alternative Flows

1a. 查询发现目前没有岗位信息

系统提示商家没有已存在的岗位，基本流继续。

2a. 数据库连接错误

系统提示商家发生了内部错误，退出用例。

1.2a 添加岗位时，岗位信息填写不正确或不完整

系统提示商家表单填写错误，并将错误的内容标红，回到“添加岗位”子流的第一步。

#### Special Requirements

无。

#### Pre-Conditions

商家需要登录之后才能执行这个用例

#### Post-Conditions

商家所做相关修改生效。

#### Extension Points

无。

### 商户信息管理 规约

#### Brief Description

这个用例解释了商户如何增加、修改、查询自己的用户信息。

#### Actor

主要执行者为商户。

#### Flow of Events

这个用例开始于商户在Web界面选择“商户信息管理”选项。

##### Basic Flow

1. 系统显示商户的所有信息。每条信息条目左侧，都有一个“修改”与“删除”按钮。
2. **根据商户所点击的按钮，选择如下子流中的一个进入：**

2.1 商户增加信息

3.1.1 商户选择“增加信息”选项

3.1.2 商户选择需要增加的信息类型，输入相应的信息内容并提交

2.2 商户修改信息

3.2.1 商户选择“修改信息”选项

3.2.2 商户选择需要修改的信息类型，输入相应的修改结果并提交

4. 系统处理商户对自己信息的操作（增加/修改）

5. 系统提示商户操作成功

##### Alternative Flows

1a. 系统提示无法获取用户信息

系统提示用户“获取数据失败”，用例继续。

1-5a. 商户退出

用例退出。

5a. 数据库发生错误

系统提示商户操作失败，并提示错误信息，用例继续。

#### Special Requirements

无。

#### Pre-Conditions

商户已登录系统。

#### Post-Conditions

商户修改后的商户信息提交到数据库保存。

#### Extension Points

无。

### 商家管理店铺 规约

#### Brief Description

这个用例解释了商家如何管理店铺信息，包括查看、新增和删除店铺。

#### Actor

这个用例的主要执行者是商家。

#### Flow of Events

##### Basic Flow

这个用例从商家选择管理店铺界面开始

1. 页面显示商家拥有的所有店铺的列表。
2. 如果商家选择“新增店铺”，那么子流**新增店铺**开始执行。

如果商家选择“修改店铺信息”，那么子流**修改店铺信息**开始执行。

如果商家选择“删除店铺”，那么子流**删除店铺**开始执行。

如果商家选择“查看店铺详情”，那么子流**查看店铺详情**开始执行。

###### 新增店铺

1. 商家填新店铺信息，包括店铺名字、店铺地址、店铺类型、店铺开门时间。
2. 系统增加新的店铺。

###### 修改店铺信息

1. 商家修改店铺信息，可以是店铺名字、店铺地址、店铺类型、店铺开门时间中的一个或多个。

2. 系统更新店铺信息。

###### 删除店铺

1. 商家选择店铺并选择删除
2. 系统删除店铺。

###### 查看店铺详情

1. 显示店铺详情页面。

#### Alternative Flows

#### Action Cancelled

如果在**任意一个**子流中，商家选择取消，那么操作中止，重新从基本流的开头开始。

#### Special Requirements

无。

#### Pre-Conditions

商家需要登陆之后才能执行这个用例

#### Post-Conditions

商家所做相关修改生效。

#### Extension Points

无。

### 商家确认员工打卡信息并结算工资 规约

#### Brief Description

这个用例解释了商家确认员工打卡信息，并结算工资的过程。

#### Actor

这个用例的主要执行者为商家。

#### Flow of Events

##### Basic Flow

这个用例从商家进入商家端打卡信息页面开始。

1. 页面显示最近十条员工未确认的打卡信息，以可复选的列表形式呈现。
2. 商家选择一条员工打卡信息，按下“确认”按钮。
3. 执行**支付工资**子流。
4. 系统将确认信息写入数据库，提示商家确认成功。

###### 支付工资

1. 系统根据已确认的打卡信息查询工资条目，创建工资账目条目。

2. 系统利用第三方API创建第三方支付服务订单，将工资转到雇员账户中。

3. 系统等待并确认第三方API返回结果，将相关订单信息写入数据库。

4. 系统利用消息推送将转账成功信息推送给目标雇员。

#### Alternative Flows

#### 1a. 无最近的员工打卡信息

1. 屏幕内容显示“无最近打卡信息”，用例终止。

##### 2a. 商家按下“批量确认”按钮

1. 显示一个消息框，内容为“是否确认N条打卡信息”，其中N为商家勾选的打卡信息数量。

2．商家单击“确定”按钮。

3. 系统将确认信息写入数据库，提示商家确认成功。

#### 1-2a. 商家选择退出

1. 用例终止。

#### 1.3a. 系统在写入数据库时发生错误

1. 提示商家系统发生内部错误，回到基本流开始。

#### 1.3b. 第三方支付接口出现异常

1. 回滚数据库条目。

2. 系统利用消息推送将支付失败的信息推送给商家，用例终止。

#### Special Requirements

无。

#### Pre-Conditions

商家已经登录，才能执行这个用例。

#### Post-Conditions

若用例正常终止，相关资金应该到位。

#### Extension Points

无。

### 与商家结算月末账单 规约

#### Brief Description

这个用例解释了系统向商家结算月末账单的过程。

#### Actors

这个用例的主要执行者是商家。

#### Flow of Events

##### Basic Flow

这个用例从系统向商家收取预先支付的工资费用开始.

1. 系统向商家显示提醒消息：“支付账单”。商家点击确认。
2. 系统向商家显示该月份的账单金额以及条目。

如果商家点击确认。那么子流**支付账单**开始执行。

如果商家点击账单有误，那么子流**报告账单问题**开始执行。

###### 支付账单

1. 商家选择支付方式。
2. 页面跳转至第三方支付页面，商家选择支付。
3. 如果第三方支付成功，页面跳转回系统页面，显示支付成功。

如果第三方支付失败，显示支付失败，并跳转回第二步。

###### 报告账单问题

1. 商家提交账单有误之处。
2. 系统后台审核，并在账单有误时更新账单，重新向用户发送支付账单的提醒消息。
3. 跳转到基本流的开始。

#### Alternative Flows

##### Action Cancelled

2a如果在**任意一个**子流中，商家选择取消，那么操作中止，重新从基本流的开头开始。

#### Special Requirements

无。

#### Pre-Conditions

商家需要登陆之后才能执行这个用例。

#### Post-Conditions

无。

#### Extension Points

无。

### 员工打卡 规约

#### Brief Description

这个用例解释了员工打卡的过程。

#### Actor

执行者为员工。

#### Flow of Events

##### Basic Flow

这个用例从员工进入客户端打卡页面开始。

1. 页面显示员工最近一条工作信息。
2. 员工单击屏幕上的按钮。

若员工单击“上班打卡”按钮，则“上班打卡”子流开始执行。

若员工单机“下班打卡”按钮，则“下班打卡”子流开始执行。

###### 上班打卡

1. 系统验证通过后提示用户打卡成功，并将打卡信息记录到数据库中。

###### 下班打卡

1. 系统验证通过后提示用户打卡成功，并将打卡信息记录到数据库中。
2. 系统将员工成功下班打卡的信息放入商家消息队列。
3. 系统将工作信息状态设为“待支付工资”。

#### Alternative Flows

##### 1a. 用户打卡时，无最近一次工作，或最近一次工作已完成

1. 显示一个消息框，内容为“无最近工作”

2. 用户按下“确定”按钮，用例终止。

#### 2a. 用户尝试打卡时，手机定位显示不在店铺附近

1. 提示用户位置不正确，打卡失败，回到基本流的开始。

1.1a 用户打卡时间早于工作开始时间

1. 提醒用户未到工作时间，打卡失败。

1.1b 用户打卡时间晚于工作开始时间5分钟以上

1. 提醒用户上班迟到，打卡成功，将打卡信息记录到数据库中。

#### 2.1a 用户下班打卡时，还未到下班时间

1. 提示用户还未到下班时间，打卡失败，回到基本流的开始。

#### 1-3a. 用户选择退出

1. 用例终止。

#### 1.1c,2.1b. 系统在写入数据库时发生错误

1. 提示用户系统发生内部错误，回到基本流开始。

#### Special Requirements

无。

#### Pre-Conditions

用户已经登录并已经授予应用定位权限，才能执行这个用例。

#### Post-Conditions

系统将相关信息存入数据库中。

#### Extension Points

无。

### 商户显示统计信息 规约

#### Brief Description

这个用例解释了显示岗位统计信息的过程。

#### Actors

主要的执行者为商户

#### Flow of Events

##### Basic Flow

这个用例从商家进入商家端统计页面开始。

1. 商家单击“统计信息”按钮。
2. 系统显示“查看岗位统计信息”、“查看员工打卡信息”、“查看工资支出统计”三个按钮。
3. 用户点选三个按钮中的一个。

若用户点击“查看岗位统计信息”，则子流“查看岗位统计”开始。

若用户点击“查看员工打卡信息”，则子流“查看打卡统计”开始。

若用户点击“查看工资支出统计”，则子流“查看工资支出”开始。

###### 查看岗位统计

1. 系统显示岗位统计列表。

###### 查看打卡统计

1. 系统显示打卡信息列表、平均打卡次数统计图、准时率统计图。

###### 查看工资支出

1. 系统显示工资支出统计列表。

#### Alternative Flows

#### 3a.商家点击导出图表

1. 系统以png格式导出图表，弹出下载对话框。

2. 回到基本流第2步。

#### Special Requirements

无。

#### Pre-Conditions

商家已经登录，才能执行这个用例。

#### Post-Conditions

无。

#### Extension Points

无。

### 商家筛选简历规约

#### Brief Description

这个用例解释了商家如何聘用员工。

#### Actor

这个用例的主要执行者是商家。

#### Flow of Events

##### Basic Flow

这个用例从商家选择”筛选简历”开始

1. 页面显示商家已经发布的所有岗位的列表。
2. 商家选择其中一个岗位。
3. 页面显示应聘这个岗位的所有用户的简历。
4. 商家选择某一个用户。

如果商家选择“聘用员工”，相应用户接收到聘用通知，系统生成打卡条目。

如果商家选择“不予聘用”，相应用户接收到不聘用通知。

如果商家既不选择“聘用员工”又不选择“不予聘用”，那么到了岗位结束招聘时间，相应用户接收到不聘用通知。

#### Alternative Flows

#### Action Cancelled

如果在**任意一个**子流中，商家选择取消，那么操作中止，重新从基本流的开头开始。

#### Special Requirements

无。

#### Pre-Conditions

商家需要登陆之后才能执行这个用例

#### Post-Conditions

商家所做相关操作生效。

#### Extension Points

无。

### 管理员管理用户规约

#### Brief Description

这个用例解释了管理员如何解禁与封禁用户。

#### Actor

这个用例的主要执行者是管理员。

#### Flow of Events

##### Basic Flow

这个用例从管理员选择”管理用户”开始

1. 页面显示所有用户的列表。
2. 商家选择其中一个用户。

如果商家选择“封禁用户”，相应用户不能登录。

如果商家选择“解禁用户”，相应用户重新能够登录。

#### Alternative Flows

#### Special Requirements

无。

#### Pre-Conditions

管理员需要登陆之后才能执行这个用例

#### Post-Conditions

管理员所做相关操作生效。

#### Extension Points

无。

## 

## 易用性

* 普通用户无需培训，就能掌握软件的基本操作流程。
* 管理员用户经过一周的训练，就能掌握基本操作流程和后台管理操作流程。
* 普通用户第一次申请岗位，可以在3分钟内完成。
* 普通用户第一次打卡，可以在1分钟内完成。
* 可评测任务次数：5次
* 操作界面需要符合iView的设计规范[[1]](#footnote-1)。
* 为普通用户提供简单操作的操作指南。

## 可靠性

* 普通用户可用时间百分比在99%以上。
* 普通用户没有维护访问权，管理员用户才有维护访问权。
* 平均故障间隔时间MTBF=300小时
* 平均修复时间MTTR=1小时
* 精确度：
  + 系统交互界面的屏幕分辨率至少为960\*640，即iPhone4的屏幕分辨率
  + 计算工资的时候，工资精确到小数点后两位
  + 计算工作时长时，时间精确到分
* 最高错误或缺陷率——0.2（bugs/function-point）
* 错误程度
  + 小错误：软件运行正常，用户很难发现运行结果存在的异常
  + 大错误：软件运行不稳定，且运行的结果与预期结果存在较大差异
  + 严重错误：软件完全不能运行，或软件的某个功能完全不能运行

## 性能

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 事务名 | 平均响应/ms | 最长响应/ms | 吞吐量/s^-1 | 并发用户容量/个 | 降级模式 |
| 用户注册 | 500 | 2500 | 100 | 250 | 暂不接受注册 |
| 用户登录 | 200 | 2500 | 1000 | 1500 | 暂时无法登录 |
| 职员打卡 | 1000 | 5000 | 20000 | 7500 | /(同登录) |
| 岗位发布 | 500 | 2500 | 300 | 1000 | 暂时无法发布岗位 |
| 新增店铺 | 500 | 2500 | 100 | 500 | 暂时无法新增店铺 |
| 员工应聘 | 200 | 2500 | 1000 | 2500 | /(同登录) |
| 结算 | 500 | 2500 | 100 | 500 | 暂时无法结算 |

资源利用情况：打卡高峰时间，平均系统资源占用在80%-90%；一般时间，平均系统资源占用在20%-30%。

## 可支持性

* 编码标准：Google Java Style Guide
* 命名约定：Google Java Style Guide
* 类库：使用Maven和Webpack框架来，实现前后端代码的管理，包括依赖的库和包。
* 维护访问权限：管理员权限（ssh key）
* 维护实用程序：Spring Actuator, Prometheus系统监控, Workbench

## 设计约束

* 软件语言：JAVA
* 软件流程需求：面向对象的并行流程方法
* 开发工具：
  + IDE：JetBrains Intellij IDEA
  + 版本控制：git （采用git-flow）
  + CI：Travis
  + 其他：Prometheus
  + 数据库：MySQL、MongoDB
* 架构及设计约束：
  + 前端：React Native
  + 后端：Spring Boot（包括但不限于Spring Security、Spring Data JPA等）
* 需购买的构件、类库：使用开源项目

## 联机用户文档与使用标准

系统应为商家用户提供后台操作的说明文档。系统开发人员应为运维人员提供详尽的系统说明文档。

## 适用的标准

网站需进行ICP备案，并符合中国大陆的相关法律法规标准。

1. <https://www.iviewui.com/docs/guide/design> [↑](#footnote-ref-1)