Hostel world系统

体系结构设计文档

V1.0

陈自强

2017年3月1日

南京大学软件学院

更新历史

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人员 | 日期 | 变更原因 | 版本号 |
| 陈自强 | 2017-02-14 | 最初草稿 | V0.1 |
| 陈自强 | 2017-03-01 | 补充完善 | V1.0 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1. 引言 4](#_Toc476660998)

[1.1. 编制目的 4](#_Toc476660999)

[1.2 词汇表 4](#_Toc476661000)

[1.3参考资料 4](#_Toc476661001)

[2. 产品描述 4](#_Toc476661002)

[3. 体系结构概述 4](#_Toc476661003)

[4. 项目结构及各开发包 5](#_Toc476661004)

[5. 用户界面层 5](#_Toc476661005)

[6. 逻辑层分解 6](#_Toc476661006)

[业务逻辑层类的职责 6](#_Toc476661007)

[7 数据层分解 6](#_Toc476661008)

[8 数据库设计 6](#_Toc476661009)

[account 6](#_Toc476661010)

[hotel 7](#_Toc476661011)

[hotel\_new 7](#_Toc476661012)

[live\_mes 7](#_Toc476661013)

[member 8](#_Toc476661014)

[modify\_application 8](#_Toc476661015)

[open\_application 8](#_Toc476661016)

[pay\_record 9](#_Toc476661017)

[point\_convert 9](#_Toc476661018)

[recharge 9](#_Toc476661019)

[reserved 10](#_Toc476661020)

[rooms 10](#_Toc476661021)

[settlement 10](#_Toc476661022)

# 引言

## 编制目的

本报告详细完成对Hostel World 系统的概要设计，达到指导详细设计和开发的目的，同时实现测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户编写，是了解系统的导航。

## 1.2 词汇表

|  |  |
| --- | --- |
| **缩写或单词** | **解释** |
| Member | 会员 |
| Manager | Hostel World 总经理 |
| Point | 积分 |
|  |  |

## 1.3参考资料

《Hostel World 系统需求规格说明文档》

# 产品描述

参考 《Hostel World 系统需求规格说明文档》中对产品的概括描述。

# 体系结构概述

Hostel World 系统中，选择了MVC结构的风格，将系统分为3个部分 （View、Model、Controller）能够很好的示意整个高层抽象，并与Spring框架高度兼容。

# 项目结构及各开发包

|  |  |
| --- | --- |
| **开发（物理）包** | **预计规划的大概功能** |
| Resources | 前端代码，下含statics：静态文件、templates：模板html文件、image：图像、js、css等，使用Thymeleaf作为模板引擎 |
| DAO | 与数据库交互，使用Hibernate作为ORM工具，使用Spring DATA JPA减少模板代码 |
| Entity | 存储于数据库表所互相映射的实体类 |
| Controller | 负责处理前端的HTTP请求 |
| Service | 负责提供服务，封装业务逻辑 |
| VO | 存储View层Bean |
| Util | **辅助类** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 用户界面层

根据需求，系统存在以下主要界面：**登录界面、注册界面、会员主界面、酒店详情界面、会员个人信息界面、会员记录界面、酒店主界面、酒店记录界面、酒店统计界面、总经理主界面、总经理财务结算界面、HostelWorld统计界面**

# 6. 逻辑层分解

业务逻辑层包括多个针对界面及业务逻辑处理对象。

### 业务逻辑层类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| AccountService | 实现登录与个人账户信息修改的服务 |
| FileService | 实现图片上传及持久化的相关服务 |
| MemberService | 实现会员业务逻辑的服务 |
| HotelService | 实现酒店业务管理的业务逻辑的服务 |
| ManagerService | 实现总经理管理的业务逻辑的服务 |
| MemberQualifiedService | 实现按计划日常进行的服务 |

# 7 数据层分解

数据层采用Spring Data JPA，基本解决了冗余代码的问题。每一个Repository对应数据库中的一个表。

# 8 数据库设计

## account

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| mail | varchar(50) | No |  |  |
| password | varchar(20) | No |  |  |
| type | int(11) | No | 0 |  |

## hotel

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| name | varchar(30) | No | HOTEL |  |
| address | varchar(500) | Yes | Nanjing |  |
| status | int(11) | No | 0 |  |
| description | varchar(5000) | Yes | No Description |  |
| pic\_url | varchar(6000) | Yes | *NULL* |  |

## hotel\_new

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| name | varchar(30) | No | HOTEL |  |
| address | varchar(500) | Yes | Nanjing |  |
| status | int(11) | No | 0 |  |
| description | varchar(5000) | Yes | No Description |  |
| old\_hotel | int(11) | Yes | *NULL* |  |
| pic\_url | varchar(6000) | Yes | *NULL* |  |

## live\_mes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| in\_time | timestamp | No | CURRENT\_TIMESTAMP |  |
| out\_time | timestamp | No | 0000-00-00 00:00:00 |  |
| person\_num | int(11) | No | 1 |  |
| person\_mes | varchar(6000) | No | No Description |  |
| pay\_method | int(11) | No | 0 |  |
| member\_id | int(11) | No | 0 |  |
| room\_id | int(11) | No | 0 |  |
| status | int(11) | No | 0 |  |
| hotel\_id | int(11) | No | 0 |  |

## member

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| name | varchar(20) | No | Customer |  |
| status | int(11) | No | 0 |  |
| remain\_days | int(11) | No | 0 |  |
| deposit | int(11) | No | 0 |  |
| points | int(11) | No | 0 |  |
| regist\_date | timestamp | No | CURRENT\_TIMESTAMP |  |
| credit\_card | varchar(6000) | No | None |  |
| accumulate | int(11) | Yes | 0 |  |
| discount | int(11) | Yes | 0 |  |
| member\_level | int(11) | Yes | 0 |  |

## modify\_application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| hotel\_id | int(20) | No | 0 |  |
| new\_hotel\_id | int(20) | No | 0 |  |
| status | int(11) | No | 0 |  |

## open\_application

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| hotel\_id | int(11) | No | 0 |  |
| status | int(11) | No | 0 |  |
| reason | varchar(5000) | No | No description |  |

## pay\_record

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| bill | int(11) | No | 0 |  |
| after | int(11) | No | 0 |  |
| live\_id | int(11) | No | 0 |  |

## point\_convert

Table comments: Point Convert to deposit

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| member\_id | int(11) | No |  |  |
| point | int(11) | No | 0 | Points to be convert |
| amount | int(11) | No | 0 | the amount of convert |
| after | int(11) | Yes | 0 |  |

## recharge

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| member\_id | int(11) | No | 0 |  |
| former | int(11) | No | 0 |  |
| after | int(11) | No | 0 |  |

## reserved

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| room\_id | int(11) | No | 0 |  |
| member\_id | int(11) | No | 0 |  |
| time | timestamp | No | CURRENT\_TIMESTAMP |  |
| status | int(11) | Yes | 0 |  |

## rooms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| name | varchar(60) | Yes | Room |  |
| hotel\_id | int(11) | No | 0 |  |
| available\_time | timestamp | No | CURRENT\_TIMESTAMP |  |
| wifi | int(11) | No | 1 |  |
| breakfast | int(11) | Yes | 0 |  |
| area | int(11) | Yes | 50 |  |
| pic\_url | varchar(6000) | Yes | *NULL* |  |
| type | int(11) | No | 0 |  |
| price | int(11) | No | 100 |  |
| status | int(11) | No | 0 |  |

## settlement

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Type** | **Null** | **Default** | **Comments** |
| id (Primary) | int(11) | No |  |  |
| hotel\_id | int(11) | No |  |  |
| amount | int(11) | No |  |  |
| status | int(11) | No |  |  |
| settledTime | timestamp | No | CURRENT\_TIMESTAMP |  |
| settled\_time | datetime | No |  |  |