**项目管理文档**

1. **项目综述**
2. **项目简要介绍**

描述软件需求规格说明中所定义的软件项目的背景、起源。该软件的建设目标，涉及的业务范围，等等

自二十一世纪以来，我国老龄化人口逐年上升，如今，现在越来越多的年轻人喜欢到大城市就业，这已经导致了一个日益严重的社会现象：子女漂泊他乡，老人独守空巢。独居老人不仅要独自面对生活中的所有问题，还要承受无人陪伴的心情所带来的孤独与无助。目前最适合我国基本国情的养老模式是社区养老模式，它不仅能为老年人在生活上提供帮助，还能迎合老人想要待在熟悉环境的心愿。然而，在人口老龄化的趋势下，相对应的老年人口的数据信息量高速膨胀，传统的数据采集方式日益显现出其弊端，无法满足如今的需求。因为开发一款高效，快捷，便于统计管理的颐养社区系统就显得很有必要了。本文通过对养老社区办公流程的分析研究，设计出一款流程简洁，响应迅速的颐养社区系统，旨在提高人员信息采集的效率，从而腾出更多的时间与空间陪伴老人，进一步改善老人的生活质量，推动养老社区的进步。

本文首先对颐养社区系统的研究背景和国内外发展现状进行了调研。其次，介绍了颐养社区的整体结构、功能需求和非功能需求。再次，对颐养社区系统中的用户管理、楼宇管理、问卷管理、病患管理进行了详细的描述。最后，提出的了现阶段颐养社区系统中还需改进的地方以及未来的发展趋势。

本文主要采用C/S（Client-Server）结构和Java语言完成系统软件部分的开发与设计。主要使用目前广泛应用的SpringBoot框架搭建系统的后端，结合MySQL数据库存储用户的信息，主要使用JavaFx框架构建系统的前端。

通过编写测试用例对系统进行功能测试和性能测试。经过测试，系统达到了预期效果，具有较好的实用性。

1. **项目面向的对象**

编写提示：对项目面向的用户或用户群体进行简要说明，可以从下面两个方面考虑：描述本项目面向的用户（客户、最终用户）或用户群体；用户或用户群的特征、需求、业务现状、正待解决的业务问题，在本系统中的职责、权限等。

用户群体：颐养社区中的员工；颐养社区中的老板

特征：都是颐养社区中的人

需求：高效的日常办公

业务现状：管理员工和老人

正待解决的问题：传统方式工作效率太低

职责权限：普通员工->用户权限、老板->管理员权限

1. **项目特点**

编写提示：简要描述项目提供的主要特征。简要可以从下列几个方面考虑：指明所开发的项目怎样满足用户或用户群体的需求，解决哪些用户关心的业务需求。可以说明所开发的项目为什么具有吸引力和它所能带来的好处、业务流程的优化、竞争优势、重要商业利润、提供给客户的价值。可以描述采用技术、架构等方面的特点及优势等

…

1. **项目中的角色**

编写提示：阐述本项目操作过程中涉及的各种角色及其职责。各种角色的具体行为将在功能性需求中描述。

|  |  |
| --- | --- |
| 角色名称 | 职责描述 |
| 项目经理 | 总揽全局，设计系统，规划开发 |
| 前端工程师 | 开发系统的用户页面 |
| 后端工程师 | 开发系统的数据接口 |
| 数据库工程师 | 制定系统的数据表 |
| 测试工程师 | 测试系统的功能 |

1. **运行环境**

描述软件的运行环境，包括硬件平台、操作系统和版本，还有其它的软件组件或与其共存的应用程序。可以分别对服务器端和客户端的运行环境进行描述。

服务端要求：

操作系统：Windows10系统

软件要求：系统安装了Java JDK1.8

硬件要求：CPU：Intel Core i5-7300 2.5GHZ，内存8GB

客户端要求：

操作系统：Windows7及以上

软件要求：系统安装了Java JDK1.8

硬件要求：64位处理器，核心至少两个，频率1GHZ以上，内存至少4GB

1. **技术和实现**

描述使用的技术、工具、编程语言和数据库。开发规范和标准。数据转换格式标准。等等

技术：Java语言，SpringBooot框架，Mybatis框架，JavaFX组件，数据库MySQL

开发标准：MVC架构

数据转换格式标准：JSON

1. **需求**
2. **需求背景**

项目启动原因，竞品分析

自二十一世纪以来，我国老龄化人口逐年上升，如今，现在越来越多的年轻人喜欢到大城市就业，这已经导致了一个日益严重的社会现象：子女漂泊他乡，老人独守空巢。独居老人不仅要独自面对生活中的所有问题，还要承受无人陪伴的心情所带来的孤独与无助。目前最适合我国基本国情的养老模式是社区养老模式，它不仅能为老年人在生活上提供帮助，还能迎合老人想要待在熟悉环境的心愿。然而，在人口老龄化的趋势下，相对应的老年人口的数据信息量高速膨胀，传统的数据采集方式日益显现出其弊端，无法满足如今的需求。因为开发一款高效，快捷，便于统计管理的颐养社区系统就显得很有必要了。本文通过对养老社区办公流程的分析研究，设计出一款流程简洁，响应迅速的颐养社区系统，旨在提高人员信息采集的效率，从而腾出更多的时间与空间陪伴老人，进一步改善老人的生活质量，推动养老社区的进步。

1. **项目目标**

目标尽量量化，作为项目上线后验证目标达到情况的依据。

目标：

1. 基本满足颐养社区日常工作需求
2. 达到流畅的工作效果
3. **修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 修订日期 | 修订人 | 修订说明 | 修订原因 |
|  |  |  |  |
|  |  |  |  |

1. **功能列表**

/\*

功能模块拆分、功能描述和功能优先级。功能拆分：一定要拆楚一个功能的组合，拆分成最小的功能点，功能点相互独立方便参与者理解需求，评估工作量。功能优先级：P0级的功能是灵魂，P1和P2是对用户体验有绝对性的意义，P3对产品存活有决定性作用，P4是锦上添花的作用。

\*/

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 功能模块 | 主要功能点 | 优先级 | 功能描述 |
| 1 | 老人管理 | 新增老人 | P1 | 往系统中添加老人 |
| 2 | 修改老人 | P1 | 修改已经入住的老人信息 |
| 3 | 删除老人 | P2 | 删除已经入住的老人信息 |
| 4 | 搜索老人 | P0 | 万能搜索，便于查找老人 |
| 5 | 浏览老人 | P1 | 浏览所有老人信息，统筹管理 |
| 6 | 员工管理 | 录入员工 | P1 | 往系统中录入员工 |
| 7 | 移除员工 | P2 | 从系统中移除员工 |
| 8 | 搜索员工 | P0 | 万能搜索，便于管理员查找员工 |
| 9 | 分类浏览员工 | P1 | 浏览系统中的所有员工 |
| 10 | 更新员工信息 | P1 | 更新员工的个人信息 |
| 11 | 楼宇管理 | 添加楼栋 | P1 | 添加社区中的楼栋信息 |
| 12 | 删除楼栋 | P2 | 移除社区中的楼栋信息 |
| 13 | 添加房间 | P1 | 为楼栋添加房间信息 |
| 14 | 移除房间 | P2 | 为楼栋移除房间信息 |
| 15 | 添加床位 | P1 | 为房间添加床位信息 |
| 16 | 移除床位 | P2 | 为房间移除床位信息 |
| 17 | 分配床位 | P0 | 为床位分配老人 |
| 18 | 问卷调查 | 添加问题 | P1 | 预定义问题 |
| 19 | 删除问题 | P2 | 移除已定义问题 |
| 20 | 分配问题 | P0 | 给问卷模板分配问题 |
| 21 | 添加模板 | P1 | 为问卷制定模板 |
| 22 | 修改模板 | P1 | 修改模板的信息 |
| 23 | 删除模板 | P2 | 移除问卷的模板 |
| 24 | 制作问卷 | P1 | 根据模板制作可以回答的问卷 |
| 25 | 发布问卷 | P1 | 将制作完成的问卷发布出去 |
| 26 | 回答问卷 | P0 | 回答已经发布的问卷 |
| 27 | 浏览回答 | P1 | 浏览对问卷的回答 |
| 28 | 个人中心 | 修改资料 | P2 | 用户修改自己的个人信息 |

1. **逻辑展示**

通过流程图（包含功能流程图和页面流程图）、思维导图（即展示产品结构框架）、原型图等，帮助自己梳理思路，以免需求遗漏。

图示

描述已自动生成

1. **详细描述**

功能详细描述，包括边界、异常等。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 主要功能点 | 功能描述 | 边界 | 异常 |
| 1 | 新增老人 | 往系统中添加老人 | 人名字数限制，电话号码规范等等 | 参数不合法 |
| 2 | 修改老人 | 修改已经入住的老人信息 | 人名字数限制，电话号码规范等等 | 参数不合法 |
| 3 | 删除老人 | 删除已经入住的老人信息 | 老人需与床位无关联 | 未先解除关联 |
| 4 | 搜索老人 | 万能搜索，便于查找老人 | 搜索条件可以是老人信息的任意字段 | 非法输入 |
| 5 | 浏览老人 | 浏览所有老人信息，统筹管理 | 老人需在系统中 | 信息展示错误 |
| 6 | 录入员工 | 往系统中录入员工 | 人名字数限制，电话号码规范等等 | 参数不合法 |
| 7 | 移除员工 | 从系统中移除员工 | 员工需在系统中 | 参数不合法 |
| 8 | 搜索员工 | 万能搜索，便于管理员查找员工 | 搜索条件可以是员工信息的任意字段 | 参数不合法 |
| 9 | 分类浏览员工 | 浏览系统中的所有员工 | 员工类别将在医生护士护工中 | 参数不合法 |
| 10 | 更新员工信息 | 更新员工的个人信息 | 人名字数限制，电话号码规范等等 | 参数不合法 |
| 11 | 添加楼栋 | 添加社区中的楼栋信息 | 楼栋名称中文，字数有一定限制 | 参数不合法 |
| 12 | 删除楼栋 | 移除社区中的楼栋信息 | 楼栋下不能有房间，需先移除房间 | 参数不合法 |
| 13 | 添加房间 | 为楼栋添加房间信息 | 同楼栋下房间不能重名，字数有一定限制 | 参数不合法 |
| 14 | 移除房间 | 为楼栋移除房间信息 | 房间下不能已有床位，需先移除床位 | 参数不合法 |
| 15 | 添加床位 | 为房间添加床位信息 | 同房间内床位不能重名，有一定字数限制 | 参数不合法 |
| 16 | 移除床位 | 为房间移除床位信息 | 床位上不能有病人，需要先移除病人 | 参数不合法 |
| 17 | 分配床位 | 为床位分配老人 | 没有病人的床位添加未被分配的病人，已有病人的床位移除病人 | 参数不合法 |
| 18 | 添加问题 | 预定义问题 | 添加问题以及四个选项，长度有一定限制 | 参数不合法 |
| 19 | 删除问题 | 移除已定义问题 | 问题不能已经被模板引用，需先解除 | 参数不合法 |
| 20 | 分配问题 | 给问卷模板分配问题 | 将未引用的问题添加进模板，将已引用的问题移除模板 | 参数不合法 |
| 21 | 添加模板 | 为问卷制定模板 | 添加问卷模板 | 参数不合法 |
| 22 | 修改模板 | 修改模板的信息 | 修改模板的名字 | 参数不合法 |
| 23 | 删除模板 | 移除问卷的模板 | 移除问题模板 | 参数不合法 |
| 24 | 制作问卷 | 根据模板制作可以回答的问卷 | 将模板当前的问题状态定格，制作成为问卷 | 参数不合法 |
| 25 | 发布问卷 | 将制作完成的问卷发布出去 | 设置问卷的发布状态，只有发布了才能做 | 参数不合法 |
| 26 | 回答问卷 | 回答已经发布的问卷 | 用户对已经发布的问卷进行回答 | 参数不合法 |
| 27 | 浏览回答 | 浏览对问卷的回答 | 浏览用户已经对问卷的回答 | 参数不合法 |
| 28 | 修改资料 | 用户修改自己的个人信息 | 用户修改自己的个人资料，各字段有一定的校验规则 | 参数不合法 |

1. **风控说明**

可能出现的风险点和策略

|  |  |  |  |
| --- | --- | --- | --- |
| 风险点 | 影响 | 等级 | 改善方案 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. **设计**
2. **概要设计**

1.1设计原则

描述本系统的结构设计原则和总体设计指导思想，主要从系统设计实现的目标来考虑，比如：数据实时性描述、可扩充性描述、可维护性、先进性、数据完整性、安全性等描述。

1.2设计规范

说明不限于以下使用的设计规范或软件开发的国家标准或规范，主要包括但不限定以下几种：

* 命名约定

规定系统和子系统名, 程序名, 数据库表(文件)名, 数据名, 变量名等的编制规范。

* 界面约定

规定屏幕界面的总体布局，如菜单行、显示主体、图标按钮、提示信息、出错信息等规范化，统一风格。

* 程序编写规范

根据采用的编程工具特点，制定规范化要求，使程序易读易懂，可维护，可移植。

具体选用的规范，只要对设计有所帮助就可以罗列，编号及相关规范标题可以自行决定。

1.3总体结构

描述系统地整体架构和框架。要求用图表进行阐述，并用文字进行解释。说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的标识符和功能，分层次地给出各元素之间的控制与被控制关系。

1.4流程设计

说明本系统的基本设计概念和处理流程，尽量使用图表的形式。

1.5性能设计

说明对本系统的主要性能要求，包括安全性、稳定性、扩展性等方面的要求。

1.6尚未解决的问题

说明在概要设计过程中尚未解决而设计者认为在系统完成之前必须解决的各个问题。

1. **数据库设计**

2.1物理结构设计

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

2.1.1库存储

对各个表的存储量及增长趋势，估算数据量，规划存储结构及其分布。

2.1.2表存储

根据各个表的增删改查操作频度、数据量、惯用操作等特点，确定表的存储方式和参数。

2.1.3索引

2.1.4其他

2.2数据字典

各类代码表（包括取值）。

2.3注意事项

关于数据库管理维护需要注意的事项。

2.4数据库表设计

给出本系统内所使用的每个数据结构中的每个数据项的存储要求，访问方法、存取单位、存取的物理关系（索引、设备、存储区域）、设计考虑和保密条件。

| 逻辑表名 | 商品表 | | 物理表名 | | product | |
| --- | --- | --- | --- | --- | --- | --- |
| 主键 | product\_id | | 索引 | |  | |
| 逻辑字段名 | 物理字段名 | 数据类型 | 空值 | 默认值 | | 备注 |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |
|  |  |  |  |  | |  |

1. **接口设计（前后端分离项目）**

3.1外部接口

说明本系统同外界的所有接口的安排包括软件与硬件之间的接口、本系统与各支持软件之间的接口、人机交互接口关系。

|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 接口详细描述 | 接口类型 | 对应构件名称 |
|  |  |  |  |

3.2内部接口

说明本系统之内的各个系统元素之间的接口的安排。

|  |  |  |  |
| --- | --- | --- | --- |
| 接口ID | 接口详细描述 | 接口类型 | 对应构件名称 |
|  |  |  |  |

1. **详细设计**

4.1开发环境

建议写明开发所需的软硬件环境，例如使用的数据库、开发工具、中间件、硬件环境、网络环境等。

4.1.1数据库管理系统

建议写明使用的数据库产品的名称以及发行厂商、版本号、语言或代码集、安装位置、配置参数等。

4.1.2开发工具、中间件

建议写明使用的开发工具、中间件的名称以及发行厂商、版本号、补丁包等信息。

4.1.3硬件环境

建议写明开发的硬件环境信息：主机、内存容量、磁盘空间、操作系统等。

4.2程序系统的结构

用一系列图表列出本程序系统内的每个程序（包括每个模块和子模块）的名称、标识符和它们之间 的层次结构关系。

4.3模块命名的规则

提示：模块设计人员确定本软件的模块命名规则（例如类、函数、变量等），确保模块设计文档的风格与代码的风格保持一致。

4.4系统架构设计

4.4.1模块汇总表

提示：对系统的总体软件结构进行必要的描述，采用图表的方式表示出模块间的层次关系和调用关系，其详细程度必须达到在本设计说明书中出现的每个模块在图表中都有反映，以便开发人员能对系统有个整体的了解。这里模块是指相对独立的软件设计单元，例如对象类、函数包等等。

|  |  |
| --- | --- |
| **子系统A** | |
| 模块名称 | 功能简述 |
|  |  |
| *…* |  |
| **子系统B** | |
| 模块名称 | 功能简述 |
|  |  |
| *…* |  |

4.4.2模块关系图

建议绘制模块结构图或层次图，或者画上包图。

4.5子系统A的模块设计

4.5.1模块A-n

|  |  |
| --- | --- |
| 模块名称 |  |
| 功能描述 |  |
| 接口与属性 | 提示：用专业的设计（开发）工具来设计本模块的接口与属性，说明函数功能、输入参数、输出参数、返回值等。此处粘贴即可。 |
| 数据结构  与算法 | 提示：不论是采用经典的还是专用的数据结构与算法，都应该作必要的描述。不仅用于指导程序的实现，还可以让人们清楚地了解该对象类是如何设计的。 |
| 补充说明 |  |

4.6子系统B的模块设计

4.6.1模块B-n

|  |  |
| --- | --- |
| 模块名称 |  |
| 功能描述 |  |
| 接口与属性 | 提示：用专业的设计（开发）工具来设计本模块的接口与属性，说明函数功能、输入参数、输出参数、返回值等。此处粘贴即可。 |
| 数据结构  与算法 | 提示：不论是采用经典的还是专用的数据结构与算法，都应该作必要的描述。不仅用于指导程序的实现，还可以让人们清楚地了解该对象类是如何设计的。 |
| 补充说明 |  |

4.6.2模块B-n界面

4.7裁剪说明

以上各章内容，可根据项目需要增加内容，或者根据实际情况修改章节结构。

1. 开发
2. 测试
3. 测试方式

例如：本次测试为系统测试，采用黑盒测试方式。测试用例情况，测试的流程等等。

1. 测试资料

例如：测试方案、测试用例、测试流程等。

1. 测试内容

|  |  |  |  |
| --- | --- | --- | --- |
| 测试系统 | 版本号 | 模块数 | 模块标号 |
| XX子系统 | XXXXX\_vX.X.1 | 2 | …… |
| …… |
| XX子系统 | XXXX\_vX.X.0 | 2 | …… |
|  |  |  | …… |
| XX子系统 | XXXX\_v2.2.0 | 1 | …… |

1. 测试结果与缺陷分析

4.1测试覆盖分析

例如：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **测试系统** | **需求编号** | **测试类型** | **用例个数** | **执行个数** | **执行用例BUG数** | **BUG未解决个数** | **是否通过** |
| XX系统 | XXXX-950 | 系统测试 | 3 | 3 | 0 | 0 | 是 |
| XXXX-952 | 系统测试 | 8 | 8 | 1 | 0 | 是 |
| XX系统 | XXXXX-81 | 系统测试 | 10 | 10 | 0 | 0 | 是 |
| XXXXX-92 | 系统测试 | 12 | 12 | 0 | 0 | 是 |
| XX系统 | XXX-71 | 系统测试 | 12 | 12 | 3 | 0 | 是 |

4.2缺陷的统计与分析

|  |  |  |
| --- | --- | --- |
| **测试系统** | **BUG率 （BUG数/测试用例个数）** | **测试未通过率 （未通过BUG个数/BUG个数）** |
| XX系统 | X.X% | 0% |
| XX系统 | XX.XX% | 0% |

1. 测试结论

例如：本次测试XX系统XXXX\_vX.X.X版本需求xxx个、xxx个用例，均测试通过。

XXXX年XX月，测试过程中记录的的缺陷均已全部解决，并且回归测试通过。

目前，系统的运行状况为：

a.各系统流程正常；

b.需求各个功能点实现正确；

c.暂不修复问题，不影响系统正常使用，已经记录XXXX；

综上可知，各系统功能正确实现，流程稳定，均达到上线要求。

附件：

项目预算核算模板

项目变更申请表