**文档编号：Thryve可视化编程工具 – SRS – *3.0***

***Thryve蓝图可视化编程工具***

**软件需求规格说明书**

**日期：2025年4月21日**

**文档变更历史记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 变更日期 | 变更人员 | 变更内容详情描述 | 变更后的版本号 |
| 1 | 2025\4\17 | 全员 | 撰写了软件需求规格说明书初稿 | V1.0 |
| 2 | 2025\4\21 | 张瑞欣 | 完善了引言部分和软件的一般性描述 | V2.0 |
| 3 | 2025\4\22 | 张瑞欣 | 完善了软件的部分用例图 | V3.0 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目录**

[1. 引言](#_Toc8397)

[1.1 编写目的](#_Toc2032)

[1.2 读者对象](#_Toc3477)

[1.3 软件项目概述](#_Toc8237)

[1.4 文档概述](#_Toc16319)

[1.5 定义](#_Toc21806)

[1.6 参考资料](#_Toc2528)

[2. 软件的一般性描述](#_Toc22652)

[2.1软件产品与其环境之间的关系](#_Toc897)

[2.2假设与前提条件](#_Toc26392)

[3. 软件功能需求描述](#_Toc21198)

[3.1 软件需求的用例模型](#_Toc16102)

[3.2 软件需求的分析模型](#_Toc21730)

[3.2.1“监视老人状况”的用例描述](#_Toc3155)

[3.2.2“远程控制机器人”的用例描述](#_Toc8519)

[3.2.3“语音/视频双向交互”的用例描述](#_Toc31303)

[3.2.4“检测和通知异常情况”的用例描述](#_Toc2722)

[3.2.5“自主跟随老人”的用例描述](#_Toc7852)

[3.2.6“用户登录”的用例描述](#_Toc3500)

[3.2.7“获取老人信息”的用例描述](#_Toc4102)

[3.2.8“提醒服务”的用例描述](#_Toc26840)

[3.2.9“系统设置”的用例描述](#_Toc5363)

[4. 其它软件需求描述](#_Toc10848)

[4.1 性能要求](#_Toc9359)

[4.2 设计约束](#_Toc12177)

[4.3 界面要求](#_Toc26380)

[4.4 进度要求](#_Toc10119)

[4.5 交付要求](#_Toc8195)

[4.6 验收要求](#_Toc31930)

[5. 软件原型](#_Toc30787)

# **1. 引言**

## 1.1 编写目的

1)本文档的目的在于方便用户、分析人员和软件设计人员进行理解和交流。用户通过需求规格说明书在分析阶段即可初步判定目标软件能否满足其原来的期望，但是本文档主要是作为设计人员的软件开发的基本出发点和系统维护人员发现和添加新功能需求的基础，也是维护人员的技术支持文档之一。

2)本文档支持目标系统的确认。软件开发目标是否完成不应由系统测试阶段的人为因素决定，而应根据需求规格说明书中确立的可测试标准决定。

3)本文档控制系统进化过程。在需求分析完成后，如果用户追加需求，那么需求规格说明书将用于确定追加需求是否为新需求。如果是，开发人员必须针对新需求进行需求分析，扩充需求规格说明书，进行软件再设计。

## 1.2 读者对象

用户，分析人员，软件设计人员，项目管理人员。

## 1.3 软件项目概述

* 项目名称: **Thryve**
* 用户单位: 教育企业、自由职业者、创意团队、程序开发人员
* 开发单位: 武汉大学计算机学院23级软件工程五鼠上将小组
* 软件项目的背景和大致功能：

随着工作场所和个人项目管理越来越依赖多种文档格式（如图片、PDF、表格和文本文件），传统的文件处理和编辑方式逐渐显得繁琐和低效。尤其是对于需要批量处理、快速组织和操作多个文件的场景，现有的软件工具常常缺乏灵活性和自动化功能。因此，如何在工作流程中高效管理和处理相关的文件，成为了一个急需解决的问题。

Thryve项目正是为了应对这一需求而设计，它结合了图形化编程界面与自动化文件处理技术，提供了一种全新的、直观且高效的解决方案。用户通过简单的蓝图编程方式，可以快速构建各种文件操作流程，如批量裁剪图片、提取表格数据、重组文本文件内容等。得益于低代码特性，Thryve特别适合没有大量编码经验的用户，能够让相关从业人员、创意团队甚至编程入门者迅速上手，完成复杂的文件处理任务，而无需深入编写代码。

此外，Thryve的界面和操作设计直观简便，适合作为编程入门教学的工具，帮助新手快速了解和实践编程思维。对于有更高需求的开发者和专业人士，Thryve提供了自定义模块的功能，用户可以根据自身需求，编写代码扩展和定制化工作流，从而充分发挥该工具的灵活性和强大功能。

本软件旨在帮助用户提高工作效率，特别是在需要频繁处理大量文档的情况下，极大地减少人工操作，提高文件处理的精确度与速度。同时，Thryve支持创新功能如大模型接口集成、文件权限管理等，使得软件具备了高度的可扩展性和智能化特性，能够适应未来技术的快速发展。

## 1.4 文档概述

1）软件的一般性描述部分。它包括软件产品与其环境的关系、软件受到的限制和约束以及软件开发前的假设与前提条件。

2）功能需求描述部分。它主要分为系统的划分，软件各子系统的功能，设计约束和性能、界面、交付、验收四个方面的要求。

3）其它软件需求描述部分。它包括性能要求、设计约束、界面要求、进度要求、交付要求和验收要求。

4）软件原型。主要设计了软件运行不同功能时的界面，比如用户登录界面。

## 1.5 定义

下面给出文档中对蓝图块的定义:

|  |
| --- |
| **class** BlueprintBlock {  **public**:      string blockID;                // 全局唯一ID      string name;                   // 显示名称      string category;               // 分类标签----分类筛选      vector<Pin> inputPins;         // 输入引脚----输入数据类型列表      vector<Pin> outputPins;        // 输出引脚----输出数据类型列表      map<std::string, Param> params; // 参数设置----用户可配置参数      virtual Data execute(**const** Data& input);  // 执行函数  }; |

|  |  |  |
| --- | --- | --- |
| 类别 | 子类型/示例 | 说明与功能 |
| 控制块 | 分支（If）、循环（While, ForEach）、延时、跳转 | 控制蓝图执行流程 |
| 数据块 | 常量、变量、全局变量、数据源绑定 | 提供和保存数据，支持文本/数值/布尔等类型 |
| 运算块 | 加减乘除、逻辑运算（与或非）、字符串拼接 | 对数据进行变换 |
| 文件操作块 | 导入文件、保存文件、获取文件信息 | 统一入口处理文件（支持图像/表格/文本/PDF等） |
| 图像块 | 裁剪、色道分离、图像拼接、OCR识别 | 图像处理任务，输入输出为图像数据块 |
| 表格块 | 表格筛选、列重排、数据合并、公式计算 | 对 CSV/Excel 表格文件的结构和内容操作 |
| 文本块 | 搜索、替换、正则匹配、关键词提取 | 支持正则语法、关键词标记、文件注释 |
| 加密块 | 加密、解密、签名、添加水印 | 文件保护相关操作 |
| 流程块 | 子图执行、脚本入口/出口、模块调用 | 多蓝图协作、函数复用 |
| 插件块 | 由外部用户或开发者定义的块 | 加载外部模块，实现扩展性 |

表1：蓝图相关定义

## 1.6 参考文献

[1] 软件工程，作者：彭蓉，出版社：清华大学出版社，2024年8月第一版。

# **2. 软件的一般性描述**

## 2.1 产品透视

Thryve作为一个独立的系统运行，能够处理包括图片、表格、文本文件、PDF等多种类型的文档。用户无需依赖其他外部系统即可完成文件的批量处理、自动化工作流的创建以及自定义模块的扩展。

除此以外，我们还希望它支持与其他系统或模块进行集成，尤其是在以下几个方面：

* 与大模型接口集成：通过集成如LangChain和HuggingFace等大模型接口，用户可以将AI处理功能集成到工作流中，增强系统智能化处理能力。
* 与文件管理系统集成：能够与其他文件管理系统（如云存储服务）进行集成，支持文件同步、备份和恢复等功能。
* 系统间数据交换：Thryve可以与其他系统通过标准API进行数据交换。例如，用户可以从外部系统导入数据（如CSV文件），或者将处理结果（如分析报告）导出到其他平台进行进一步处理或存储。

## 2.2 产品功能概述

Thryve软件具备一系列强大的功能，旨在通过图形化编程简化文件处理与自动化工作流的创建。以下是软件的主要功能模块：

1. 文件批量处理

* 图片处理：Thryve支持批量处理图片文件，功能包括裁剪、色道分离、格式转换等。用户可以通过简单的图形化编程流程，快速实现这些操作。
* 表格数据重组与提取：对于Excel或CSV文件，Thryve能够帮助用户高效处理大量表格数据。
* 文本文件操作：支持批量文本搜索、替换、检阅编辑等操作，尤其适合用于大规模文件的处理和内容管理。
* PDF文件操作：包括页面分割、合并、提取文本、加密与解密等功能，帮助用户高效处理PDF文件。

2. 图形化编程

* 蓝图编程：Thryve采用图形化编程方式，用户可以通过拖拽组件并配置操作参数，快速构建自动化工作流。无需编写复杂的代码，极大简化了编程过程。
* 节点化设计：用户通过图形化节点设计来构建文件处理流程，节点支持自定义配置与逻辑设置。
* 流程可视化：每个蓝图流程在界面上都有清晰的可视化展现，用户可以实时查看、调整和优化工作流。

3. 用户定制与扩展

* + 插件扩展支持：Thryve支持用户通过编程自定义插件和模块，扩展软件功能。开发者可以为特定任务编写自定义逻辑，满足更加个性化的需求。
  + 大模型接口集成：支持集成大规模语言模型接口（如LangChain和HuggingFace），为复杂任务提供AI增强能力。用户可以在工作流中引入AI操作，如图像处理、自然语言处理等。

4. 跨平台支持

* 多操作系统兼容：Thryve支持在Windows、Mac和Linux等操作系统上运行，确保在不同平台之间的无缝切换。

5. 安全性与审计

* 操作审计与日志：系统记录所有用户的操作行为，提供审计日志，确保文件操作的可追溯性和安全性。

## 2.3 用户特征

根据我们的前期调查与信息搜索，我们大致确定了以下几类的用户以及相关的特征。

|  |  |  |  |
| --- | --- | --- | --- |
| 用户类型 | 描述 | 需求和能力 | 使用场景 |
| 教育企业 | 提供编程教育的机构或学校 | 对编程知识有所了解  需要简单易用的教学工具 | 使用图形化编程界面帮助学生理解编程思维，尤其适用于入门级编程教育。 |
| 自由职业者 | 独立工作者，处理多种类型的文档和数据 | 无需复杂编码能力  需要高效的文件处理与自动化工具 | 通过蓝图编程快速处理日常任务，如图片裁剪、数据提取等。 |
| 创意团队 | 设计、媒体等创意行业的团队 | 具备一定创意思维但缺乏深厚编码能力  需要高效完成文件操作 | 使用可视化编程处理创意设计项目中的文件操作，快速实现功能。 |
| 程序开发人员 | 专业的开发人员，具备一定的编码能力 | 可以通过编码自定义模块  需要灵活扩展功能 | 自定义插件或模块，扩展现有功能，满足专业需求。 |

表2：用户特征

## 2.4软件产品与其环境之间的关系

本软件结合了前端和后端技术，提供一个强大的跨平台文件处理与蓝图编程工具。软件的组成和所依赖的环境如下表所示：

|  |  |
| --- | --- |
| 系统组成部分 | 外部环境 |
| 系统环境 | Windows, Mac, Linux 操作系统 |
| 前端框架与图形化编程界面 | Javascript,React |
| 后端及文件处理部分 | Pytho及相关文件处理库 |

表3：软件与外界环境的交互关系

## 2.5假设与前提条件

### 2.5.1 假设

用户具备基本的计算机操作能力和对文件操作的常识，能够理解和使用图形化编程界面。

软件的运行环境是现代化的操作系统（Windows, Mac, Linux），并且用户已安装了必要的运行时环境，如 Python 和 Node.js。

用户的硬件设备能够支持软件所需的最低系统要求，包括一定的内存和处理器性能，尤其是在批量处理大文件时。

### 2.5.2 前提

本软件会不断随着用户需求和技术发展进行更新和优化，前提是具备稳定的技术支持和社区反馈。

基于低代码设计理念，假设用户能够从现有的预设模块中选择并快速配置工作流，满足大部分文件处理需求。对于有更高需求的用户，可以通过自定义编码模块来扩展功能。

软件的部署不依赖于特定硬件平台或操作系统，能够兼容主流操作系统并在云平台上顺利运行，支持跨平台的功能。

# 

# **3. 软件功能需求描述**

## 3.1 软件需求的用例模型



图1：空巢老人看护系统--用例图

## 3.2 软件需求的分析模型

本部分描述系统的9个用例的设计。

### 3.2.1“插件下载”的用例描述

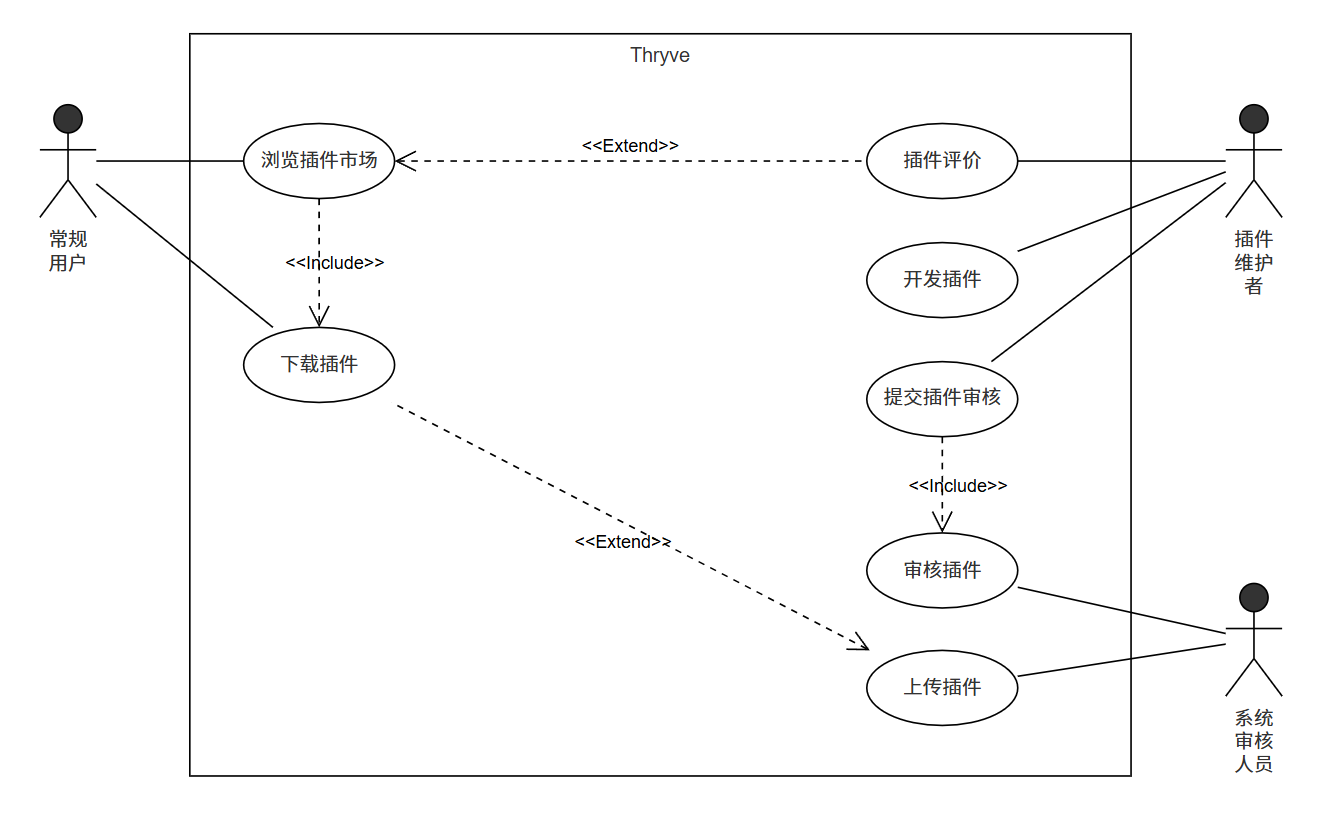


图2：Thryve—“插件下载”用例图

|  |  |
| --- | --- |
| 用例名 | 用户插件下载 |
| 用例描述 | 描述用户下载插件，以及插件开发的流程。 |
| 参与者 | 用户，插件维护人员，系统审核人员 |
| 过程 | 1. 用户浏览插件市场，选择合适的插件下载。 2. 用户根据自己的使用体验提交反馈。 3. 开发者根据相关需求开发软件并且提交审核。 4. 审核人员审核插件防止出现非法行为。如果审核通过，则上传插件。 |

表2：“用户插件下载”用例描述表

### 3.2.2“版本控制与协作”的用例描述

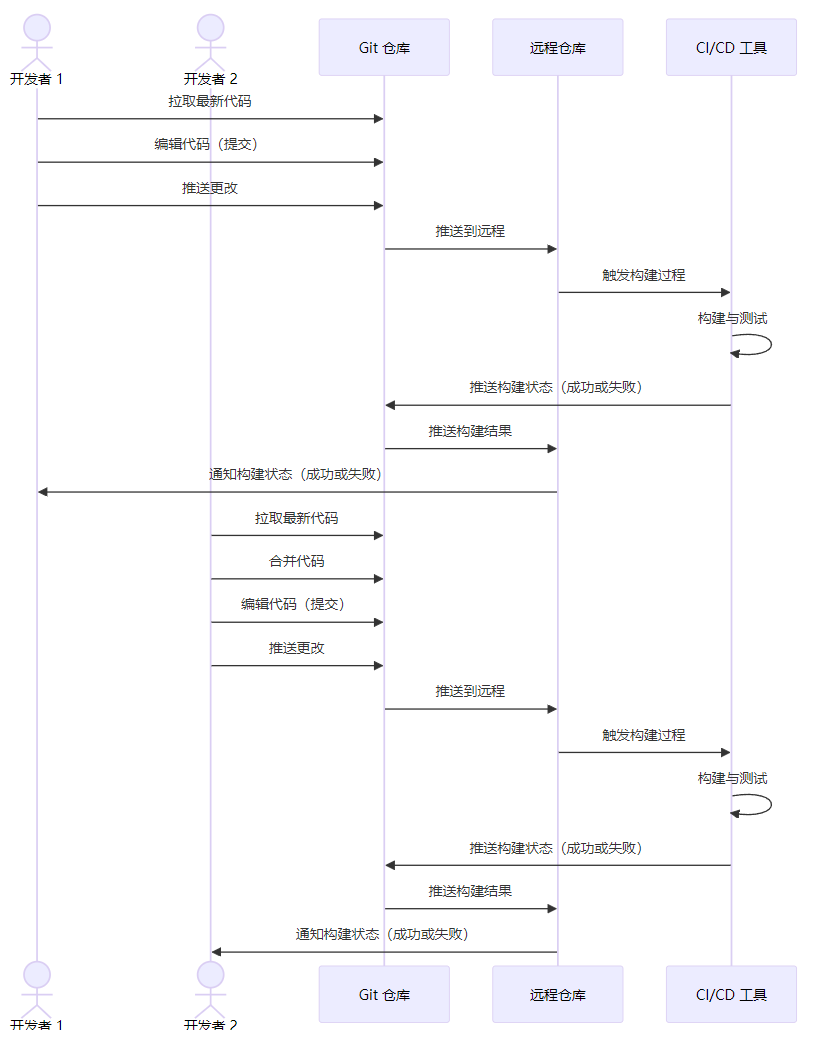


图3：Thryve—“版本控制与协作”顺序图

|  |  |
| --- | --- |
| 用例名 | 版本控制与协作 |
| 用例描述 | 展示了开发者1和开发者2在协作开发过程中如何分别进行不同操作。开发者1仅进行推送操作，而开发者2则需要进行拉取、合并等团队协作操作。此外，CI/CD工具在推送代码后进行构建与测试，并将结果回传给开发者。 |
| 参与者 | 开发者1、开发者2、Git仓库、远程仓库、CI/CD工具 |
| 过程 | 1. **开发者1**拉取最新代码并进行修改，提交后将更改推送到Git仓库。 2. **开发者2**拉取最新代码，并与本地修改合并。 3. 开发者2修改代码后，将更改提交并推送到Git仓库。 4. 远程仓库接收到推送后，触发CI/CD工具执行构建与测试。 5. CI/CD工具执行构建与测试，并将结果推送回Git仓库。 6. Git仓库更新构建结果，并通知开发者1和开发者2构建状态（成功或失败）。 7. 开发者根据构建状态调整后续开发工作。 |

表5:“远程控制机器人”用例描述

### 3.2.3“语音/视频双向交互”的用例描述



图4 ：空巢老人看护系统—“语音/视频双向交互”顺序图

|  |  |
| --- | --- |
| 用例名 | 语音/视频双向交互 |
| 用例描述 | 家属、医生、老人之间通过语音和视频进行交互 |
| 参与者 | 老人、家属和医生 |
| 过程 | 1. 用户跟“语音通话”界面交互 2. 手机客户端将通话请求发送给云端服务器 3. 建立手机客户端与云端服务器同asterisk服务器的voip链接，实时通话开始 |

表4:“语音/视频双向交互”用例描述

### 3.2.4“检测和通知异常情况”的用例描述



图5 ：空巢老人看护系统—“检测和通知异常情况”顺序图

|  |  |
| --- | --- |
| 用例名 | 检测和通知异常情况 |
| 用例描述 | 机器人检测到老人摔倒后发送报警信息，用户通过手机客户端接收报警信息 |
| 参与者 | 机器人 |
| 过程 | 1. 系统对老人信息进行分析，检测老人是否出现异常（如摔倒） 2. 如果出现异常，将该异常信息通知给远端手机APP |

表3：“检测和通知异常情况”用例描述表

### 3.2.5“自主跟随老人”的用例描述



图6：空巢老人看护系统—“自主跟随老人”顺序图

|  |  |
| --- | --- |
| 用例名 | 自主跟随老人 |
| 用例描述 | 机器人随老人的移动而移动，以在安全距离观察老人 |
| 参与者 | 机器人 |
| 过程 | 1.用户通过手机“移动控制”界面发出自动跟随请求  2.云端服务器将请求转发给机器人  3.机器人开启自主移动模式，开始自动跟随老人 |

表6:“自主跟随老人”用例描述

### 3.2.6“用户登录”的用例描述



图7：空巢老人看护系统—“用户登录”顺序图

|  |  |
| --- | --- |
| 用例名 | 用户登录 |
| 用例描述 | 用户通过账号和密码登录系统 |
| 参与者 | 家属、医生 |
| 过程 | 1. 用户输入账号和密码  2. 系统验证用户账号和密码的正确性和合法性  3. 如正确则登录成功，否则提示用户重新输入账号和密码 |

表7:“用户登录”用例描述

### 3.2.7“获取老人信息”的用例描述



图8：空巢老人看护系统—“获取老人信息”顺序图

|  |  |
| --- | --- |
| 用例名 | 获取老人信息 |
| 用例描述 | 获得老人的视频、图像和语音等信息 |
| 参与者 | 机器人 |
| 过程 | 1. 机器人运动到可观察老人的安全距离 2. 通过传感器获得老人的视频、图像和语音等信息 3. 将获取的老人信息通过互联网传送到远端的移动APP 上 |

表7:“获取老人信息”用例描述

### 3.2.8“提醒服务”的用例描述



图9：空巢老人看护系统—“提醒服务”顺序图

|  |  |
| --- | --- |
| 用例名 | 提醒服务 |
| 用例描述 | 提醒老人按时服药和保健 |
| 参与者 | 定时器 |
| 过程 | 1. 机器人移动到接近老人的安全距离 2. 播放语音提醒老人按时完成相关事务（如服药和保健） |

表7:“提醒服务”用例描述

### 3.2.9“系统设置”的用例描述



图10：空巢老人看护系统—“系统设置”顺序图

|  |  |
| --- | --- |
| 用例名 | 系统设置 |
| 用例描述 | 配置系统，设置系统参数 |
| 参与者 | 系统管理员 |
| 过程 | 1. 设置机器人控制软件的IP 地址和端口号  2. 设置机器人与老人的安全距离  （1）设置机器人与老人的安全距离  （2）检查安全距离设置是否超出预定的范围  （3）提示操作成功或者重新输入  3. 设置机器人的移动速度  （1）设置机器人的移动速度  （2）检查速度设置是否超出预定的范围  （3）提示操作成功或者重新输入  4. 设置用户账号和密码  （1）设置用户的账号和密码  （2）检查账号和密码的输入是否合法  （3）提示操作成功或者重新输入  重复执行步骤1至4，直至用户选择退出设置。 |

表7:“系统设置”用例描述

# **4. 其它软件需求描述**

## 4.1 性能要求

1).功能响应时间要求:网络连接建立后，云端服务器一定可以响应用户发送的请求;非常重要的功能不超过100ms,其余功能不超过460ms

2).视频清晰度要求:支持720p清晰度,信号传输流畅(对应的延迟要求为,在网络信号满格或少一格时不超过120ms),无卡顿无花屏

3).语音通讯质量要求:语音清晰可辨,信号传输流畅（延迟在网络信号满格或少一格时不超过100ms）

4).可靠性要求:用户使用该系统时,云端服务器和手机客户端能够访问,机器人正常运行。系统支持7\*24长期稳定运行,并且系统维护不能影响到机器人的性能发挥

## 4.2 设计约束

1).硬件约束:对TurtleBot2机器人进行应用设计时需要考虑很多复杂和困难的问题,如机器人动作控制,各种场景处置

2).时间约束:必须要在2018年6月初完成这项开发工作的主要任务

3).技术约束:精细地控制机器人的运动是一项复杂的工作,现在我们必须高效利用现有的高质量开源代码去合理重组改造,圆满完成我们的工作

## 4.3 界面要求

1）机器人端和电脑端软件：没有界面需求，故未设计界面。

2）手机客户端软件：用户的主要可视化界面，要求具有较高的观赏性。布局紧凑，有联系的组件尽量布局在一个页面中。针对机器人操控及功能选择菜单界面进行界面美化、优化，着重关注用户使用手机客户端软件关注老人实况的整个流程中的各个界面。

## 4.4 进度要求

需要开发者在2018年1月中旬给出软件原型,并在同年5月中旬完成全部软件开发工作,完成验收与交付

## 4.5 交付要求

交付内容：

1）.机器人端服务器软件、云端服务器软件、手机客户端软件。

2）.软件设计规格说明书的电子文档

3）.使用说明书的电子文档和纸质文档

## 4.6 验收要求

1).要求整个系统正常运行过程中无Bug，能在用户非正常操作的情况下报告错误但不至于崩溃。

2).要求在不同的实际场景检测中，整个系统确实能够完成看护空巢老人的设计功能。

3).要求整个系统各部分优化完毕，不存在编程过程中遗留的调试代码等影响用户体验的部分。

# **5. 软件原型**

1）本界面为本软件的欢迎界面。



图11：空巢老人看护系统用户端软件功能菜单界面

2）本界面为用户进行登录的界面，用户需要输入注册时所选用的用户名并输入设定的密码即可。



图12：空巢老人看护系统手机端软件登录界面

3）下图为本软件“监视老人状况”的功能界面。



图13：空巢老人看护系统手机端软件“监视老人状况”界面

4）下图为本软件“远程控制机器人”的功能界面。

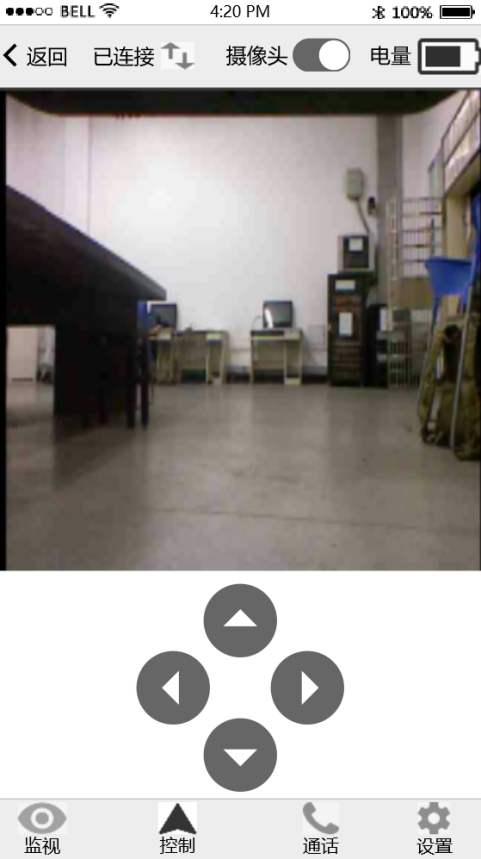


图14：空巢老人看护系统手机端软件“远程控制机器人”界面

5）下图为本软件“视频/语音双向交互”功能的界面。

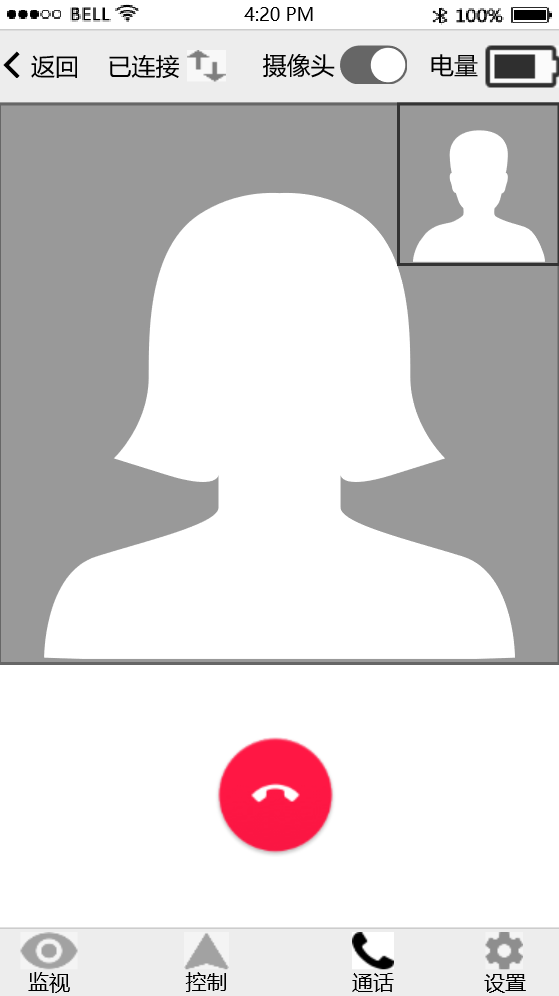


图15：空巢老人看护系统手机端软件“视频/语音双向交互”界面

6）下图为“系统设置”功能界面，可以进行系统参数的设置。

******

图16：空巢老人看护系统手机端软件“系统设置”界面