【墨玉-机器人系统】

项目开发总结报告

【PDSR-201】【1.1】

2020.06

分工说明

小组名称	墨玉（组号：201）	
学号	姓名	本文档中主要承担的工作内容
17373203	陆广炎	3 实际开发结果、5 缺陷与处理
17373261	陈宇轩	4 开发工作评价
17373200	陈宇航	1 引言、2 引用文件、3 实际开发结果、4 开发工作评价
17373217	段牧知	4 开发工作评价、6 经验与教训
17373468	赵子敬	3 实际开发结果

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0	6.15	陈宇航	陆广炎	PDSR 初稿
1.1	6.19	陈宇航	赵子敬	修改了文档格式和个别语病

1 引言

1.1 标识

表 1.1.1 标识表

编号	系统名称	版本号
1	Linux-Ubuntu	V16.04
2	ROS	Kinetic
3	Node.js	V12.18.0
4	NPM	V6.14.4
5	Vue	V2.6.10
6	Python	V3.7.7
7	Django	V3.0.6

1.2 系统概述

作为一种专用的计算机系统，嵌入式系统是软件和硬件的综合体，还可以涵盖机械等附属装置。它嵌入在装置或设备中实现特定的控制逻辑，如机器人控制单元等。

本项目目标为基于嵌入式系统设计一款可以灵活行进，识别目标物体并可抓取目标物体的机器人，用于酒店服务，主要功能是房间送物和引导带路。该机器人软件部分使用 ROS 机器人算法平台进行开发设计，硬件部分则直接采用拥有硬件里程计、激光测距雷达、立体视觉相机和语音输入输出阵列等一系列硬件设施且可完美搭载 ROS 算法的启智 ROS 机器人。

除了上述的主要功能，本项目对于设计出的机器人还有以下非功能性需求：1、机器人结构稳定，可长期使用而不会轻易损毁。2、机器人硬件软件并行设计，耦合度较低。这样即使现有硬件坏了，也可较为方便地进行软件移植，在短时间内获得可新使用的机器人。3、可扩展性强。即当用户具有新需求时，可通过对现有机器人软件系统进行扩展并重新移植，增加现有机器人的功能。4、在规定时间内(当前设置反应时间为 5 秒)对于正确的用户需求做出响应。5、对于错误

的用户需求进行提示并正确处理；若出现错误可在有限时间内恢复；此外还拥有简单的指令格式，易于使用和学习。

1.3 文档概述

本文档主要是用来总结软工项目-墨玉机器人自2020年3月截止至2020年6月总计3个月的项目开发成果。其中，本文档主要由以下六部分组成：引言，引用文件，实际开发成果，开发工作评价，缺陷和处理以及经验和教训。

每部分的作用如下：

引言部分负责对项目和文档本身的内容进行了总结和概述，此外还对文中的术语和缩略词进行了解释。

引用文件部分列出本文档引用的和产生的所有文档及代码的编号、标题、修订版本、日期。

实际开发成果部分由项目产品，主要功能以及性能，基本流程，进度，人员工作量五部分组成，详细地反应了项目开发小组三个月内的工作流程，人员工作量，进度以及里程碑成果。

开发工作评价部分列出了项目开发小组以三个月来的开发经历为基础，对开发效率，产品质量，技术方法以及出错原因的评价与分析。除此之外，该部分还对风险管理部分进行了较为详细地阐述。

缺陷与处理部分分别列出在需求评审阶段、设计评审阶段、代码测试阶段、系统测试阶段和验收测试阶段发生的缺陷及处理情况

最后，经验和教训部分列出从本项目开发工作中得到的最主要的经验与教训及对今后的项目开发工作的建议。

1.4 术语和缩略词

表 1.4.1 术语和缩略词表

业务及技术术语/缩略词	全称
ROS	The Robot Operating System
URDF	Unified Robot Description Format

NPM	Node Package Manager
SLAM	Simultaneous Localization and Mapping

2 引用文件

表 2 引用文件表

编号	标题	版本	发行日期
1	启智 ROS 开发手册	V1.1.0	2018/11/09
2	启智 ROS 机器人手册	V1.1.0	2020/2/19
3	ROS 系统入门	V1.1.0	2020/3/05
4	SDP-软件开发计划文档	V1.3	2020/6/8
5	SRS-软件需求规格说明文档	V2.0	2020/6/8
6	SDD-软件设计说明文档	V2.0	2020/6/8
7	STD-软件测试说明文档	V2.1	2020/6/8

3 实际开发结果

3.1 产品

a. 版本说明：

由于无法返校，本项目在五月初才最终确定以仿真形式完成开发于测试，因此从五月初至今本的开发过程中项目只有两个版本，版本号 0.0 与 1.0

版本 0.0 是 5 月 22 日上线的，仅仅满足最初基本需求的版本，其中实现了仿真环境下的机器人控制、建图和导航功能，物品识别和语音识别模块暂时没有调试成功，机械臂模块没有上线；实现了前端界面的基本功能，实现了前端数据库，但前后端交互暂时还存在一定问题。

版本 1.0 是 6 月 4 日上线的，完善了基本功能的最终交付版本，其中完善了仿真环境下机器人的控制、建图、导航、物品识别和语音识别功能，完善了前端和数据库的交互以及前后端的交互，除机械臂模块在仿真环境中无法开发和测试以外，其他功能全部实现。

版本 1.0 是在版本 0.0 的基础上完善而成的，修复了 0.0 存在的问题（如语音识别和物品识别相关依赖的缺失等等），更好地串连了各个模块，并完善了前后端的交互。由于版本 0.0 的缺陷较多，没有独立上线交付的能力，因此版本 1.0 开发时直接覆盖了 0.0 的版本，在 github 仓库里仅保留了最新的 1.0 版本，但通过 commit 记录可以看到版本更新覆盖的记录。

b. 最新的版本软件单元、代码量及地址如下表呈现：

表 3.1.1 软件单元信息表

软件单元	共享代码	独立开发代码	存储地址	访问链接
前端界面	1366	1585	Team201/MoyuRobotServer	https://github.com/sebuaa2020/Team201/tree/master/MoyuRobotServer
前后端交互	约 2000	约 1000	Team201/moyu-manage-system	https://github.com/sebuaa2020/Team201/tree/master/moyu-manage-system
ROS 仿真	约 8000	约 1000	Team201/Team201_catkin_ws	https://github.com/sebuaa2020/Team201/tree/master/Team201_catkin_ws
数据库	0	663	Team201/database	https://github.com/sebuaa2020/Team201/tree/master/database

其中 ROS 仿真包含了 gazebo 仿真界面，以及控制机器人运动和数据传输的底层代码，此部分使用了较多成熟的共享代码，也在原有的共享代码中做了交多修改，但代码系统中源码、配置文件、cMake 文件等代码量不容易归一化统计，在此仅列出大致的代码量，独立开发的代码包含了独立撰写的代码文件和修改力度交大的共享代码文件（如 launch 文件）。前端界面使用成熟的框架修改和增添进行开发，是和用户直接交互的模块。前后端交互模块将上述两个模块串联起来，主要使用代码调用 ROS 仿真系统中的包。各个模块都与数据库交互，实现数据的存储和调用。

c. 数据库的说明、存储地址和访问链接如下表呈现：

表 3.1.2 数据库信息

数据库名称	说明	存储地址	访问链接
ROSDATABASE	经过需求分析发现，本项目涉及到的数据存储需求不大，可以采用 python 的 sqlite3 库提供的轻型数据库作为系统数据库	Team201/database/src	https://github.com/sebuaa2020/Team201/tree/master/database/src

3.2 主要功能和性能

本项目最初的目标为基于嵌入式系统设计一款可以灵活行进，识别目标物体并可抓取目标物体的机器人，用于酒店服务，主要功能是房间送物和引导带路。该机器人软件部分使用 ROS 机器人算法平台进行开发设计，硬件部分则直接采用拥有硬件里程计、激光测距雷达、立体视觉相机和语音输入输出阵列等一系列硬件设施的启智 ROS 机器人。机器人不仅具备灵活行走，行进避障，前往目标地点等基本功能，还具备识别目标物体并抓取等功能，除了用于酒店服务之外，还具有广阔的应用场景。

但随着疫情的发展，返校取用真实设备开发的希望逐渐渺茫，课程组也于五月初宣布具体代码的开发和测试基于仿真进行。因此原有的开发计划和需求文档中，部分软件模块的最终实现和原有的计划有着较大的不同。下面逐项对比原定的开发目标和最终交付版本的对应关系：

表 3.2.1 开发目标与最终交付产品对应关系表

功能需求 (SRS-201-2.0 文档)		原定对应软件模块 (SDP-201-1.3 文档)		最终版本完成情况
控制功能	4.1 小节	底盘控制模块 语音识别模块	2.1.1 小节 2.1.6 小节	【达到目标】 实现了使用键盘和语音两种方式控制仿真界面中机器人的运动。但由于仅在仿真界面运行，系统删去了原

				有的控制真实机器人底盘硬件的代码，仅发布速度控制信息到相应节点即可。
建图功能	4.2 小节	建图模块 系统用户界面 数据库模块	2.1.2 小节 2.1.7 小节 2.1.8 小节	【达到目标】 实现了手动控制和自动探索两种建图方式，实现了建图后关键点的标定，并实现了将地图和关键点信息存储在数据库。
巡航功能 送物功能	4.3 小节 4.4 小节	导航模块 系统用户界面 数据库模块	2.1.3 小节 2.1.7 小节 2.1.8 小节	【达到目标】 实现了仿真环境中机器人的顶点导航和巡航功能：从数据库中选取对应的地图，实现机器人自动运动到目的地。巡航和送物的功能实质都是机器人的导航，其中同样测试完善了动态的避障功能，保证了机器人在环境中遇到动态的障碍物不会发生问题。
取物功能	4.5 小节	导航模块 机械臂模块 视觉识别模块 系统用户界面 数据库模块	2.1.3 小节 2.1.4 小节 2.1.5 小节 2.1.7 小节 2.1.8 小节	【未完全达到】 取物功能设计是客户在房间里选择需要的物品，机器人运行到储物间后抓取相应物品再送至客户房间。该功能除了机械臂模块外，其他模块均完成了既定的功能：导航模块可以确保机器人在客户房间和储物间之

				间运动，视觉识别模块可以保证机器人在储物间正确地确定待取物品的位置，数据库模块保障了存储所需物品余量和位置。机械臂模块无法实现的原因是仿真环境中缺少真实机械臂的各方面参数信息，无法完成仿真。

3.3 基本流程

a. 系统原计划采用迭代增量模型，下图是文档 SDP-201-1.3 中描述迭代增量开发流程的图：

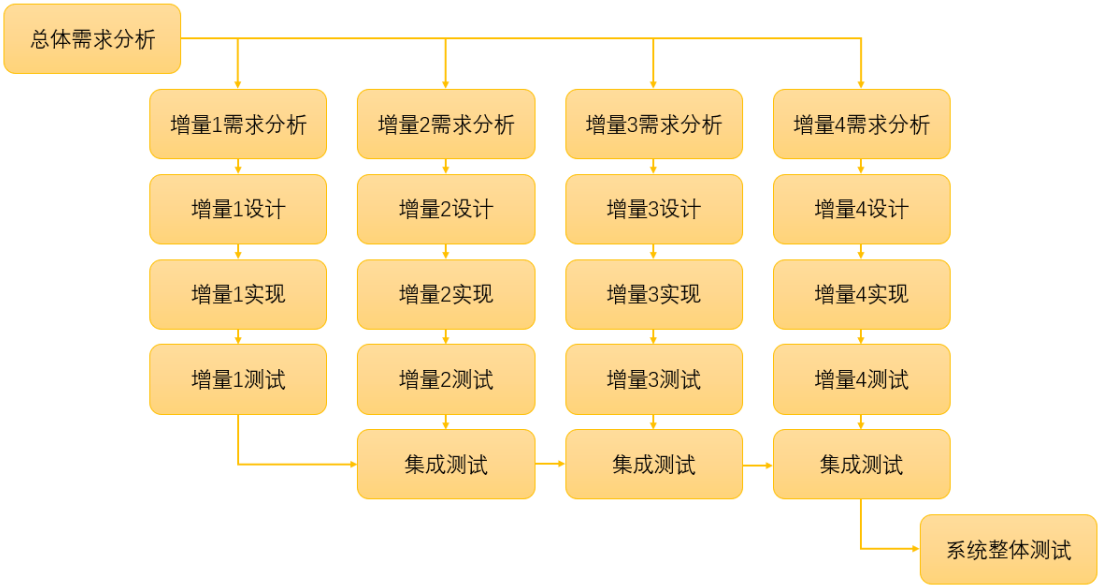

图 3.3.1 迭代增量模型图

在实际开发过程中，单个模块基本遵循了迭代增量原则，即单个模块需求分析-设计-实现-测试。但由于课程安排原因，在五月初才最终确定使用仿真系统来进行开发和测试，因此在五月的开发产生了一系列变化，由于时间紧迫，多个模块同时进行了开发，最终同时进行了集成测试，引入了敏捷开发的特点。

b. 设计说明文档中表明了整体设计体系结构、各个代码模块体系结构，以及设计与需求的对应关系来说明需求的可追踪性。下表是 SDD-201-2.0 文档中需求用例与功能需求，以及设计模块与功能需求的对应关系：

表 3.3.1.a 需求用例与功能需求的对应关系表

SRS 需求用例	用例描述	SDD 对应模块
2.1 宾客引领	宾客底达酒店完成入住手续后由机器人引领宾客到达房间	巡航功能
2.2 巡查	酒店管理者无需亲自走动即可查看酒店公共空间各角落情况	巡航功能
2.3 房间送物（酒店管理者送物）	客户点餐或第三方外卖送达前台，酒店管理者接受物品并使用机器人无接触送至客户房间	巡航功能
2.4 房间送物（客户自主订物）	客户需要酒店提供毛巾、饮用水等物品，机器人自动运行至储物间，无接触送至房间	巡航功能 取物/送物功能
2.5 机器人适应新环境	酒店公共空间翻新或装修，格局变化，机器人应适应变化	控制功能 建图功能

表 3.3.1.b 设计模块与功能需求的对应关系表

SRS 功能需求	功能描述	SDD 对应模块

控制功能	用户可以通过键盘和语音两种方式对机器人下达命令。使用键盘操控时，系统会自动启动键盘控制节点,之后用户可以通过键盘控制机器人的移动。使用语音操控	6.1 底盘控制模块 6.8 语音识别模块
建图功能	机器人既可由用户操控手动建图,也可由本身实现自动建图。启动建图后机器人会启动可视化节点，依据 SLAM 算法建图并储存行程信息。	6.2 建图模块 6.5 激光雷达数据模块 6.6 Kinect 相机数据模块
巡航功能	机器人可按照固定路线进行巡航,同时在路上可启动可视化节点以实现避障的功能。	6.3 导航与避障模块 6.6 Kinect 相机数据模块
送物/取物功能	机器人可通过移动机械臂的方式取走或传递其开启可视化节点后观察到的物体。	6.3 导航与避障模块 6.4 机械臂模块 6.6 Kinect 相机数据模块 6.7 视觉识别模块

后续的开发过程比本遵循了设计文档的模块设计方式，除使用仿真无需添加的硬件控制模块外，其他模块均已实现，并对应地实现了功能需求。

c. 单元测试用例设计的依据

单元测试用例设计依据大致如下：

- 首先根据本组 SDD-软件设计说明文档中接口设计. 详细设计以及需求可追踪性说明部分，提取出需要进行测试的功能单元。
- 结合 SRS-软件需求规格说明文档，分析出每个功能单元的作用以及在各种情况下（正常/异常情况下）的处理流程。
- 对每一个功能单元，分析其应用场景，看是否需要区分与设计黑白盒测

试。

- 在每个功能样例设计阶段，首先基于该功能模块的流程图，状态图等，从而有的放矢，在测试过程中覆盖尽可能多的分支。而黑盒测试设计阶段，则根据 SDD-软件设计说明文档中接口设计，基于其提前约定好的输入数据，设计其对应的黑盒测试样例。

d. （测试发现的错误和修改过程）

在测试过程中

- 对于前端界面设计模块，多为点击测试，其具体测试情况详见 STD-软件测试说明文档。
- 其余模块所采取的测试方法为命令行测试，即通过命令行作为对应的功能模块的输入途径，详细的测试用例已在 STD-软件测试说明文档中进行描述。可根据其用例标识找到对应的测试数据。

3.4 进度

下表是开发计划中（SDP-201-1.3 文档第 6 小节）的计划进度表。

图 3.4.1 计划进度图

下表是略去各成员分工的最终实际进度：

图 3.4.2 实际进度图

总体来说，项目整体的各部分进度和原计划进度都有较大的差距，主要原因在于前期没有确定在疫情期间是否能够返校完成原定的真机开发计划，直到第 12 周和 13 周左右才最终确定使用仿真完成整个系统的开发和测试，因此原定的迭代增量开发不得不采用敏捷开发的特点，用户界面、视觉检测和语音识别模块一同在 13-15 周完成了开发。

3.5 人员工程量

a.列出开发过程中的原计划和实际分工情况对比，若有调整，指出其原因；

表 3.5.1 分工情况对比表

姓名	职位	计划分工	实际分工
陆广炎	项目经理	统筹项目计划；对接客户；监督项目完成进度和质量；调动资源满足开发需求；制定并熟悉系统架构；协助项目具体模块开发；评审和修改文档格式。	统筹项目计划；对接客户；监督项目完成进度和质量；调动资源满足开发需求；协助项目具体模块开发；评审和修改文档格式；数据库设计和开发
赵子敬	ROS 工程师	熟悉 ROS 架构和各个接口；熟悉机器人与传感器硬件特性；指定并熟悉系统架构；为软件工程师提供系统接口；移植工程代码到硬件保证顺利运行；负责系统运行展示。	熟悉 ROS 架构和各个接口；熟悉机器人与传感器硬件特性；指定并熟悉系统架构；为软件工程师提供系统接口；移植工程代码到硬件保证顺利运行；负责系统运行展示。
段牧知	软件工程师	熟悉系统架构；编写项目软件	熟悉系统架构；编写项目软件工

		工程代码；代码单元测试；代码维护与版本更新；数据资源管理。	程代码；代码单元测试；代码维护与版本更新；前端设计和开发
陈宇航	软件工程师	熟悉系统架构；编写项目软件工程代码；代码单元测试；代码维护与版本更新；数据资源管理。	熟悉系统架构；编写项目软件工程代码；代码单元测试；代码维护与版本更新；前端设计和开发
陈宇轩	视觉工程师	熟悉系统架构；专门负责编写视觉部分代码；代码单元测试；代码维护与版本更新；数据资源管理。	熟悉系统架构；代码单元测试；代码维护与版本更新；前后端交互开发；部分底层代码实现

b.团队主要采用 git 完成项目合作的管理：

表 3.5.2 项目管理工具

项目管理工具	存储地址和访问链接
GIT	https://github.com/sebuaa2020/Team201/

c.以列表形式明确每个阶段提交物的每人工程量（百分比）。

表 3.5.3 人员工程量分配

提交物编号版本	名称	陆广炎	陈宇轩	陈宇航	段牧知	赵子敬
SDP V1.3	项目开发计划	20%	20%	20%	20%	20%
SRS V2.0	软件需求规格说明	20%	20%	20%	20%	20%
SDD V2.0	软件设计说明	20%	20%	20%	20%	20%
STD V2.1	软件测试说明	20%	20%	20%	20%	20%
PDSR V1.0	项目开发总结报告	20%	20%	20%	20%	20%
总体工程量分配	——	20%	20%	20%	20%	20%

4 开发工作评价

4.1 对开发效率的评价

我们 3 月份到 4 月中旬间主要在熟悉 ROS 开发平台，以学习别人代码为主，真正自己写代码开发是从 4 月中旬开始。而我们的项目到 6 月中旬基本全部完成，总共开发了 2 个月左右。

这两个月中，我们主要进行的工作就是程序开发和文档撰写。下面对开发效率进行分析。

程序部分我们的工作量如下图所示：

表 4.1.1 工作量表

模块名称	共享代码	自己开发代码
前端网页	1366	1585
ROS 仿真	约 8000	约 1000
数据库	0	700
前后端交互	2000	1000

其中段牧知和陈宇航同学负责前端网页部分，总共独自开发代码为 1600 行左右。大致工作量为段牧知 700 行，陈宇航 900 行。

赵子敬同学负责 ROS 仿真部分，工作量为约 1000 行。（注：ROS 仿真部分包含了较多 ROS 底层的代码，其中源码、配置文件、cMake 文件等文件代码不容易归一化进行统计，例如仿真界面 world 的配置代码多达两千余行，但大部分是图形化界面中自动生成。因此此处仅粗略统计源码的代码量，自己开发的代码包含了自己开发的代码和有较大改动的共享代码文件）

陆广炎同学负责数据库部分的设计，工作量为 700 行代码。

陈宇轩同学负责前后端交互代码，独自开发代码为 1000 行左右。

文档部分，我们总共编写了 5 个文档，对应字数分别如下：

表 4.1.2 文档字数说明表

文档名称	文档字数
SDD（软件设计说明书）	11497

SDP（系统开发计划）	5402
SRS（软件规格说明书）	6019
STD（测试说明）	6399
PDSR（项目总结报告）	7000

我们组共 5 个人，我们每次分别撰写不同的部分，文档编写工作量基本一样。

文档总共 36317 个字，因此我们五个人每人都是每月编写 3.6 千字左右。

4.2 对产品质量的评价

a.系统功能性评价及依据；

功能需求从业务需求的五大用例场景中提炼出来，其简要的功能分块有：控制、建图、引导、巡航、送物、取物六大功能。

其中控制、建图、引导、巡航、送物功能均已实现。其中引导和巡航部分机器人都能自动进行路径规划按照最优路线来。

取物部分则因为当前代码不支持机械臂功能而没有实现。

具体依据见附件里附的演示视频。

b.系统非功能性评价及依据。

性能指标（按照普通程度的室内复杂程度）：

- SLAM 算法构建地图的速度不低于 10 分钟/100 平方米。
- 机器人导航过程中的运行速度可设定等级，最高速度不超过 4 米/秒。
- 命令发布到响应的时间不超过 1 秒
- 目的地设定到规划路径完成时间不超过 1 秒

这些要求均已满足，依据可以见我们附件里的演示视频。从演示视频可以看出我们的系统运行极其顺畅，不存在卡顿不响应的情况。

质量属性：

- 系统可移植性较强，通过安装文档可以移植到纯净的 Ubuntu16.04 系统

我们的项目完全基于 Ubuntu16.04 平台开发，一个组员在安装完全部配置后可以顺利运行出结果，因此我们的系统可移植性较强、

- 系统鲁棒性较强，用户界面应对各种异常输入不会向外部抛出异常

我们的前端页面对于各种输入都做了足够的逻辑判断，如果出现不符合规定的输入的话会直接显示出来并拒绝执行，而不会出现系统卡机报异常的情况。

示例图片（各种异常判断）

图 4.2.1 用户登录界面异常

输入坐标不符合规范：

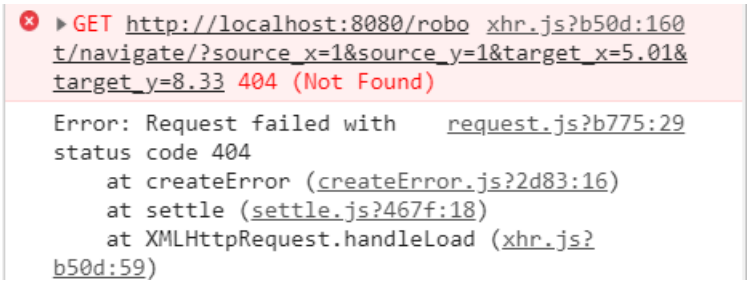

图 4.2.2 输入坐标异常

送货未送入房间：

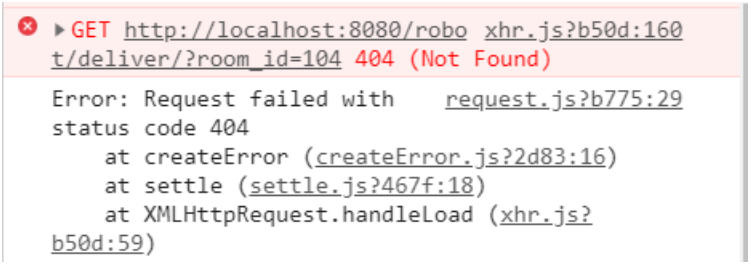

图 4.2.3 送货未送入房间异常

前端对于特殊情况会做出足够的判断。

4.3 对技术方法的评价

a.指出所采用的模型、技术、方法、工具、手段等：

本项目采用 Web 技术进行可联网的前后端交互。

交互模型采用 MVC 模型构建前后端交互系统，与之相关的工具是：利用 SQLite3 数据库构建 Model，与数据库交互的操作由 Django 框架解决；View 利用前端框架 Vue 和后端框架 Django 实现；Controller 利用后端框架 Django 实现，负责 HTTP 请求的分析和业务逻辑的处理，Controller 中涉及 ROS 操作的部分使用 ROS 官方提供的 rospy 工具解决。

b.给出所采用技术方法的应用效果评价：

采用前后端交互模式使得项目可以利用互联网进行远程交互，提高了项目的实用性，更加贴合现实需求。

采用 MVC 模型较好地降低了前端、后端以及数据库之间的耦合度，提高代码的可扩展性。

使用 Vue、Django 框架则提高了代码的开发效率，省去了繁琐的 HTTP 协议请求分析，避免重复造轮子。同时 Vue 框架也能帮助本项目构建更友好、更现代化的前端可视化界面。

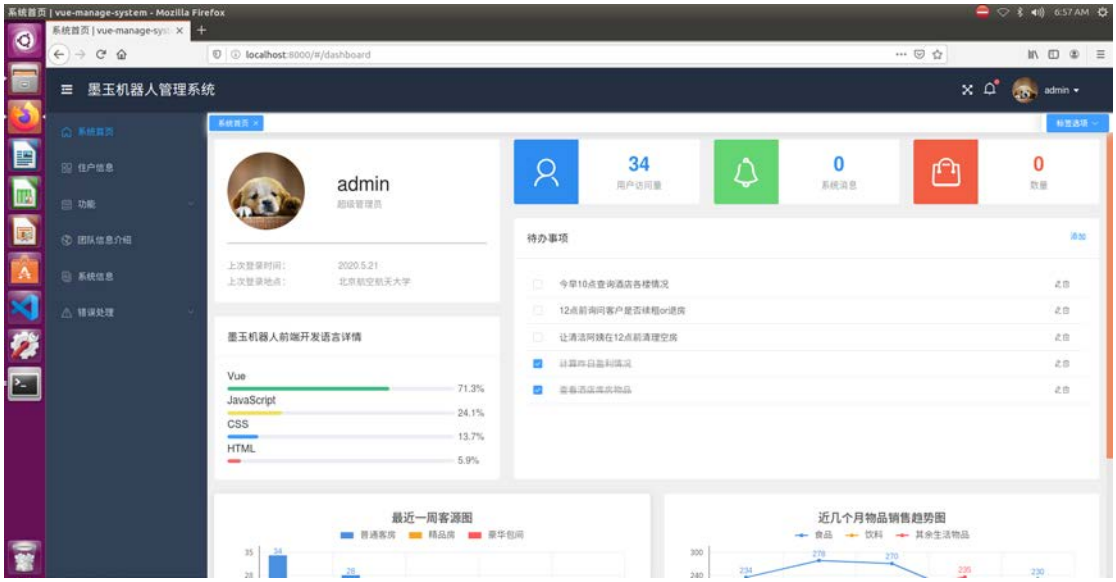

图 4.3.1 前端界面展示

4.4 问题原因分析

4.4.1 主要问题

表 4.4.1 主要问题

问题序号	问题描述
1	导航与避障模块无法实现动态避障。
2	服务器无法正常返回静态文件（如 html、js 脚本文件等）。
3	Gazebo 无法正常加载机器人模型。
4	语音识别模块无法正常使用。
5	Ubuntu 虚拟机中 Gazebo 打开后闪退
6	Python3 无法使用 roslaunch 等 ROS 官方包

4.4.2 出现原因

表 4.4.2 出现原因

问题序号	出现原因
1	local costmaps 的配置文件中没有添加 obstacle layer，导致局部路径规划没有考虑检测到的障碍物。
2	django 的 setting 文件中没有配置静态文件目录。
3	小组自己构建的 gazebo 地图中包含机器人模型，导致加载 urdf 模型时提示重复错误。
4	没有安装语音控制模块所需的科大讯飞依赖包。
5	可能与显卡驱动有关，禁用显卡硬件加速后可解决（但需增加提供给虚拟机的内存和 CPU 资源）。
6	ROS Kinetic 版本对 Python3 的支持不完全。

4.5 风险管理

4.5.1 初期预计的风险

在项目初期 SDP-项目开发计划文档的编写中，项目开发小组预计了以下风

险：

1. 由于实际项目开发过程中不可控因素而导致的人员安排，工作分配等方面所导致的估算风险

2. 开放平台，硬/软件，数据库性能，以及代码风格等方面所带来的技术风险

3. 软件开发人员突发情况和沟通机制方面所产生的人员变动风险

4. 由于客户需求存在可变性，因此可能带来需求变动方面的风险

4.5.2 实际发生的风险

在历时 3 个月左右的项目开发过程中，实际发生了如下风险：

1. 由于对实际开发任务理解不清晰，导致工作量分配存在问题。在某些开发阶段中，项目开发人员分配得到的任务工作量占比不均衡。

2. 由于嵌入式编程和其他类型的软件编程存在一定差异，在实际开发过程中，项目开发人员在开发平台方面遇到了一些困难。具体体现 ROS 开发不熟练。

3. 受限于疫情影响，需求发生了变更。其中，机械臂抓取以及反馈需求变成可选需求。

4.5.3 风险减缓或消除情况

当风险实际发生时，项目开发小组采取了以下措施来减缓甚至消除风险。

1. 前期由于软件开发经验不足，项目开发小组的任务分配存在一定的问题。痛定思痛，在后续的开发过程，我们为了保证软件开发工作的顺利进行，会定期通过腾讯会议进行商讨，从而协调好软件开发不同时期不同开发角色之间的工作安排以及工作量。

2. 因为本小组绝大多数开发人员以前尚未接触嵌入式开发，所以一部分项目开发人员在开发平台方面遇到了一些困难。在后续开发过程中，我们通过请教有经验的开发人员以及合理安排任务，成功地减缓了这方面地风险。

3. 由于无法使用实体 ROS 机器人设备，项目需求发生了变更。不过，

经过项目开发小组的不懈努力，截至到 2020 年 6 月 8 号，开发人员成功完成绝大多数除机械臂抓取方面之外的任务，抓取是否成功的节点也在不断开发以及测试。

5 缺陷与处理

5.1 需求评审阶段

5.1.1 缺陷

①完成需求规格说明书后，未对开发计划文档进行及时更新
②对于需求用例，没有加以图示说明
③描述系统的 E-R 图作图不规范
④文档中图表没有对应的说明
⑤根据需求而设计的功能太笼统，没有具体细化

5.1.2 处理

①及时更新开发计划文档

分工说明

小组名称	墨玉（组号：201）	
学号	姓名	本文档中主要承担的工作内容
17373203	陆广焱	5.资源计划
17373261	陈宇轩	4.过程模型
17373200	陈宇航	3.风险管理
17373217	段牧知	1.范围
17373468	赵子敬	2.项目任务概要 6.进度计划

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0	2020/3/9	赵子敬	陆广焱	初稿
1.1	2020/4/14	陈宇轩	陈宇航	添加迭代-增量模型的流程示意图

图 5.1.1 开发计划文档及时更新

②增加需求用例图

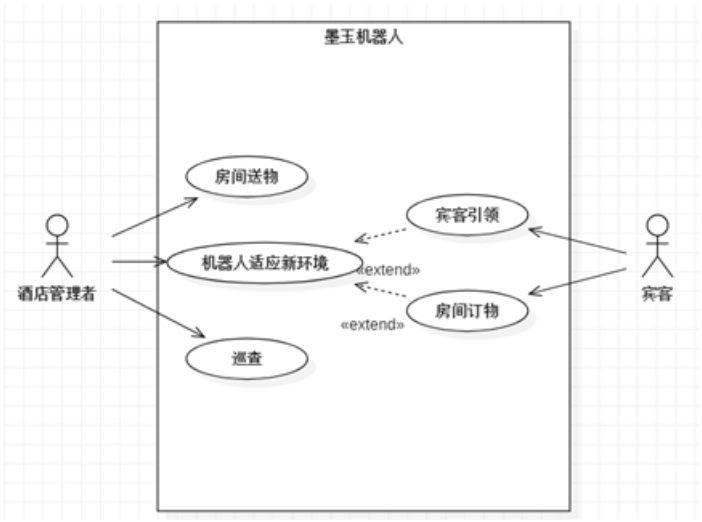

图 5.1.2 增加需求用例图

③规范系统 E-R 图

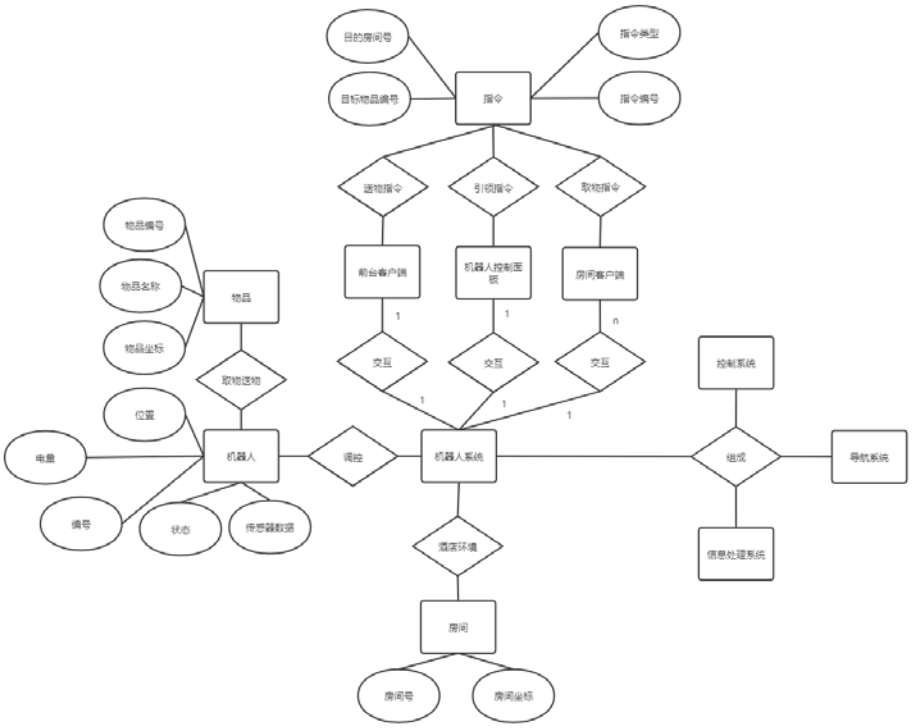

图 5.1.3 规范系统 E-R 图

④为文档中的每个图、表增加编号和对应说明

⑤具体描述每个功能，并添加流程图加以说明

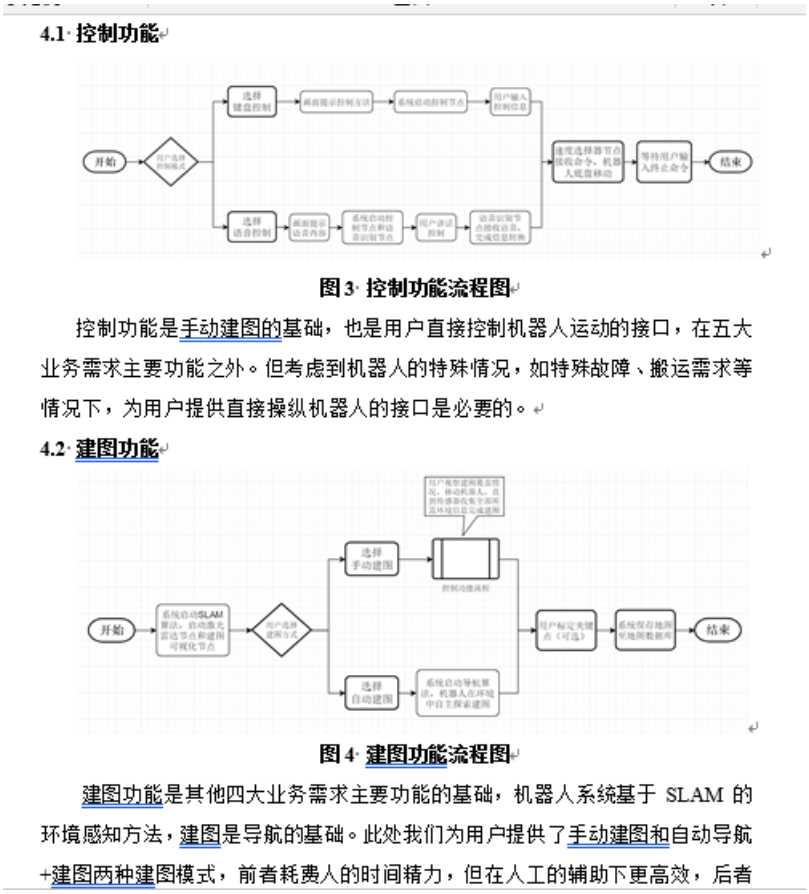

图 5.1.4 细化功能并添加图示描述

5.2 设计评审阶段

5.2.1 缺陷

①未补充时序图来展示模块包含的对象之间是如何协作的

②未说明模块如何实现用例

③用户界面设计部分未阐明详细设计部分

④表格的说明应该在表格上方

5.2.2 处理

①补充时序图来展示模块包含的对象之间的协作

6.3 导航与避障模块

导航模块是实现多项需求用例的核心，它与控制模块相通之处在于都会通过 ROS 消息发布机制向地盘硬件对应的 Topic 发布速度控制信息 Message，但导航模块发布消息不直接来源于用户控制，因此其运行的稳定性和避障能力更为重要。导航模块需要同时提供目的地路径规划、发布速度控制消息、调用 Kinect 相机数据（以供巡查）和基于激光雷达数据避障四方面的功能，四方面协作的活动图如下：

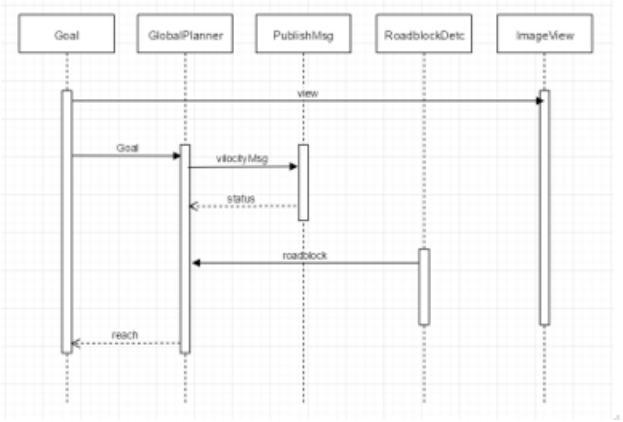

图 5.2.1 添加时序图

②添加用例-功能映射表格补充说明模块与用例的关系

SRS 功能需求与 SDD 模块的对应关系如下表所示：

表 8.1-1 SRS 需求用例与功能需求对应关系

SRS 需求用例	用例描述	SDD 对应功能
2.1 宾客引领	宾客底达酒店完成入住手续后由机器人引领宾客到达房间	巡航功能
2.2 巡查	酒店管理者无需亲自走动即可查看酒店公共空间各角落情况	巡航功能
2.3 房间送物（酒店管理者送物）	客户点餐或第三方外卖送达前台，酒店管理者接受物品并使用机器人无接触送至客户房间	巡航功能
2.4 房间送物（客	客户需要酒店提供毛巾、饮用水	巡航功能、

图 5.2.2 添加用例-功能映射表说明模块与用例的关系

③添加用户界面设计详细设计部分

6.10 前端页面实现

本项目采用的是基于 node.js 的渐进式 JavaScript 框架-Vue 实现了软工项目-墨玉机器人的前端页面设计。在具体设计时，我们使用了 vue-cli3 脚手架并借助了了 Element UI 组件库，从而方便我们开发出快速简洁好看的组件。此外，我们进行了分离颜色样式设计，除了支持手动切换主题色之外，还可以很方便使用自定义主题色。

6.10.1 配置环境

```
cd Vue-manage-system    //进入模板目录
npm install     //安装项目依赖，等待安装完成之后，安装失败可用 cnpm 或 yarn

//开启服务器，浏览器访问 http://localhost:8080
npm run serve

//执行构建命令，生成的 dist 文件夹放在服务器下即可访问
npm run build
```

图 5.2.3 用户界面详细设计说明

④更正表格的说明格式

5.3 代码测试阶段

5.3.1 缺陷

①测试时间分配错误

②测试用例未具体化

③未完成机械臂模块，导致不能进行测试

5.3.2 处理

①更正测试时间分配

	关软件包以及讯飞语音识别软件包。
测试人员	陈宇航。
时间分配	1 天。
测试数据	随机语音输入进行测试。

3.2.2 数据库以及前端界面设计模块

表 3.2.2 数据库以及前端测试准备

硬件环境	PC 一台。
软件环境	SQL Server，Visual Studio Code，浏览器。
测试人员	陆广发、段牧知、陈宇航。
时间分配	3 天。
测试数据	在相应的前端页面，测试功能按钮以及输入的表项；对数据库的增删改查进行覆盖测试。

3.2.3 建图模块

表 3.2.3 建图模块测试准备

硬件环境	PC 一台。
软件环境	Ubuntu 16.04LTS 操作系统，搭载 ros-kinectic 环境及相关软件包、gazebo7 仿真环境及相应的模型配置文件。
测试人员	赵子敬、陈宇轩。
时间分配	2 天。
测试数据	在 gazebo 仿真环境中建立的场景地图。

图 5.3.1 更正测试时间分配

②将测试用例具体化

4.5 语音识别模块测试用例

4.5.1 黑盒测试用例

表 4.5.1 视觉识别模块测试用例表

用例标识	初始状态	输入数据	有效等价类	预期结果
4.5.1-1	机器人正常运行。	输入语音数据“前进”。	输入正确的指令关键词。	模拟器中机器人正常前进。
4.5.1-2	机器人正常运行。	输入语音数据“前进并且向上”。	输入中文句子，其中包含正确的关键词。	模拟器中机器人正常前进，忽略“向上”。
4.5.1-3	机器人正常运行。	输入语音数据“向上”。	语音输入不包含关键字。	模拟器中机器人无响应。

图 5.3.2 具体化测试用例

③对机械模块功能部分实现错误处理和异常提示功能

5.4 系统测试阶段

5.4.1 缺陷

①开发计划、需求规格说明书、软件设计文档未及时更新

5.4.2 处理

①及时更新开发计划、需求规格说明书、软件设计文档，做到文档之间的协同

墨玉-机器人系统-开发计划-【SDP-201】【1.3】

分工说明

小组名称	墨玉（组号：201）	
学号	姓名	本文档中主要承担的工作内容
17373203	陆广焱	5.资源计划
17373261	陈宇轩	4.过程模型
17373200	陈宇航	3.风险管理
17373217	段牧知	1.范围
17373468	赵子敬	2.项目任务概要 6.进度计划

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0	2020/3/9	赵子敬	陆广焱	初稿
1.1	2020/4/14	陈宇轩	陈宇航	添加迭代-增量模型的流程示意图
1.2	2020/6/8	陈宇航		同步修改文档，更新文档信息
1.3	2020/6/8	陆广焱		更新工作内容

图 5.4.1 更新开发计划

分工说明

小组名称	墨玉（组号：201）	
学号	姓名	本文档中主要承担的工作内容
17373203	陆广焱	2.业务需求 3.数据需求
17373261	陈宇轩	7.运行环境
17373217	段牧知	1.范围
17373200	陈宇航	6.用户界面
17373468	赵子敬	4.功能需求 5.非功能需求

版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
1.0	2020.3.30	陆广焱	赵子敬	初稿
1.2	2020.4.03	陆广焱	赵子敬	初稿的完善和加强
1.3	2020.4.15	陆广焱	段牧知	增加模型图和规范用图
1.4	2020.4.18	赵子敬		规范了流程图
1.5	2020.4.23	陆广焱		更新了数据需求
2.0	2020.6.8	陈宇航		SRS 文档最新版本
2.1	2020.6.8	陆广焱		审核并修改了 SRS 文档

图 5.4.2 更新需求规格说明书

1.3	2020.4.24	段牧知	陈宇轩	对图表信息进行整合修改，完善接口设计

1.3	2020.4.24	陈宇航	陈宇轩	修改表标，撤销界面和不必要的注释
1.4	2020.5.21	段牧知	陈宇轩	修改用户界面部分
1.5	2020.5.21	陆广炎、赵子敬		修改详细设计中后端模块实现部分以及数据库部分
2.0	2020.6.8	陈宇航		文档整体格式修改以及前端页面部分修改
2.1	2020.6.15	赵子敬	陆广炎	增加用例-功能映射表

图 5.4.3 更新软件设计文档

5.5 验收测试阶段

5.5.1 缺陷

①最终产品基本达到了开发计划制定的目标，满足绝客户所提出的绝大部分需求。但是由于技术上的原因，机械臂模块未能调试成功，导致房间取物功能未能实现

5.5.2 处理

①将现阶段产品完成最终测试后先予以交付，后续的需求实现作为产品更新和升级反馈给客户

6 经验与教训

6.1 经验与教训

1、一定要分工明确，且不同人负责的工作之间的交集点尽可能小，这样才能让整个团队最高效地完成开发任务。

举例来说，我们团队一开始想的是开始时所有人一起开发调整 ROS 机器人代码，每个人负责实现机器人一个功能模块。之后一个人设计前后端的接口，剩下每个人负责设计前端的一个网页，如初始导航页、任务布置页等。最后每个人都对其中一个部分进行测试。即我们将整个项目流程大致分为了 3 个阶段，机器人硬件代码调通、前后端开发设计、覆盖性测试 3 个部分。这三个流程中我们每个人都分别处理流程中的一个子任务。看起来很好，但最后发现这样做实际上存在着开发效率方面的问题。

首先，这样做虽然每个流程中我们都在并行处理，但这三个流程本身却是串行执行的。实际上这三个流程中前两个流程本身也能并行处理。

其次，前端和硬件部分对应的知识和要求的软件并不一样，我们每个人都重新学习新知识和软件实际上会造成资源浪费。

最后我们在发现这样做存在开发效率方面的问题后开会对于分工进行了调整。一个同学负责依据提供的 demo 调整完善 ROS 机器人的功能代码，剩下两个同学负责前后端接口和数据库的设计，两名同学负责前端网页的开发。这样做我们每个人分工都足够明确，不同成员之间的工作也基本不存在交叉问题。前后端开发和 ROS 机器人开发两个流程并行进行，时间效率大大增加了。

2、一定要做好时间规划，给自己留下足够的时间预备量来应对可能出现的各种情况。

举例来说，因为项目开发周期很长，我们一直在不紧不慢的推进。我们本来

给初步调通代码留下了 1 周半的开发时间（验收前的一周半），但真到了验收前一周多我们却发现出现了很多意外情况，例如计算机科学方法论 ddl，计算机网络实验大实验等。同学们分心乏术，完全无法按照正常计划开发。最后那周为了完成预定的任务，我们 5 个人不得不经常熬夜到 3-4 点开发来赶进度。

3、工作有交叉点的同学一定要经常彼此交流，以免出现同样的干两次的浪费时间的情况。

举例来说，前端页面的功能性有时候很难划分，每个同学都有自己的设计想法。这时候只有两个人开会讨论对前端包含哪些页面，每个页面包含哪些功能进行划分，才能避免重复开发的情况出现。

4、一定要熟练使用 github 这个优秀的开发工具，在提交前 pull，准备提交时正常 push，避免出现文件丢失的情况。

刚开始使用 github 的时候我们组一位同学忘记了 pull，选择 push-f 的方式强制提交。导致我们组同学的 commit 记录全部丢失，那一轮其他同学的修改也不见了。最后我们还花了一些时间来找回 commit 记录和修改的文件。我们幸运的是有文件备份，故能找回。如果其他同学在 push-f 后全部 pull 了的话，那么无疑这些文件和 commit 记录很难找回，整个团队势必损失惨重，也不能版本回退了。

6.2 建议

1、做好时间规划，留好时间预备量。

2、磨刀不误砍柴工，正式开干前多讨论，合理分工，这样之后能事半功倍。

3、开发前要先熟悉开发工具，如 github 等。用好开发工具会为之后的团队开发带来很大便利。