|  |  |  |  |
| --- | --- | --- | --- |
| **tcllogo** | | | |
| **ChatBot聊天机器人项目结项报告** | | | |
| 文档发布/owner | 研发部 | 文档版本/1.0 | Version 1.1 |
| 保密等级/classification | 按公司标准化部门规定Company Confidential | | |

**版权说明**

本文档中的内容属于TCL公司所有，未经书面许可严禁以任何方式披露给第三方。

**Disclosure**

The information contained in this document is proprietary to TCL and shall not be disclosed by the recipient to third persons without the written permission of TCL.

**修改记录/Revision history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **状态/版本**  **Status Version** | **日期**  **Data** | **作者**  **Drafter** | **修改内容描述**  **Description of change(s)** | **批准**  **Approved** |
| Draft 0.1 |  |  | 创建 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目 录

目 录 4

1 项目背景和工作目标 6

1.1 立项背景 6

1.2 项目工作目标 6

2 项目计划和执行 6

2.1 关键里程碑执行情况 6

2.2 工作量和人力投入 6

2.3 风险识别和管理 6

2.4 项目突发情况说明 6

2.5 技术评审 6

3 技术开发总结 6

3.1 需求工作总结 6

3.2 软件整体架构描述 7

3.3 重点开发内容 7

3.4 技术难点和问题 7

4 测试和质量工作总结 7

4.1 项目测试情况 7

4.2 BUG数据汇总和分析 7

4.3 测试用例总结 7

4.4 软件质量分析 7

5 经验教训和改进建议 7

5.1 项目经验教训 7

5.2 改进建议 8

6 后续安排 8

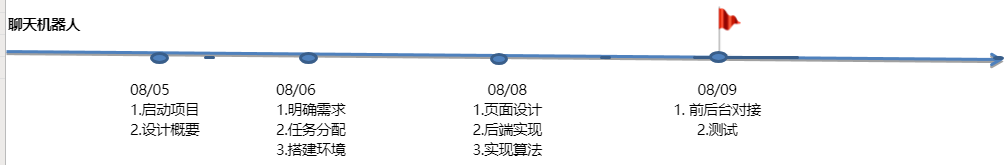
7 项目资料清单 8

1. 项目背景和工作目标
   1. 立项背景

聊天机器人，是一种通过自然语言模拟人类进行对话的程序。近年来，聊天机器人受到了学术界和工业界的广泛关注。微软推出了基于情感计算的聊天机器人小冰，百度推出了用于交互式搜索的聊天机器人小度，进而推动了聊天机器人产品化的发展，且符合国家的科研及产业化发展方向。本项目主要实现一个简易的聊天机器人及其技能开发。

* 1. 项目工作目标

1. 分析理解需求矩阵，完成组内分工，输出概要设计。
2. 按分工进行开发，实现前端、算法、闲聊和歌曲请求四个部分的功能模块并测试，输出测试阶段文档，各会议纪要等。
3. 全员参与项目开发的完整流程后，输出结项报告，理解掌握项目开发的基本方法。
4. 项目计划和执行
   1. 关键里程碑执行情况



* 1. 工作量和人力投入

|  |  |  |
| --- | --- | --- |
| 人员 | 主要任务 | 工作量 |
| 曹玉慧 | 负责项目前端部分，组织讨论和项目跟进 | 40h |
| 朱晨澍 | 负责项目服务端部分，前后端以及算法部分整合，文档编写 | 40h |
| 李充厅 | 负责项目服务端部分，编写文档 | 40h |
| 付丽 | 负责项目算法部分以及各纪要，编写文档 | 40h |
| 李涛 | 负责项目算法部分，编写文档 | 40h |
|  |  | 200h |

* 1. 风险识别和管理

1. 时间风险：未在规定的时间内完成
   1. 原因：对开发流程不熟悉，可能会导致不能按期交付项目。
   2. 应对办法：明确细化需求，制定详细和明确的开发计划，合理分工，责任到人，咨询导师。

1. 多人多版本风险：
   1. 原因：算法和后端均有两人在负责，除此之外还有需要组员共同撰写的一些文档，如果各自为战就可能造成后期代码合不上的结果。
   2. 应对方案： 使用版本控制工具，小组内使用的是Git，分模块开发项目。

1. 技术风险：
   1. 原因：项目的算法部分虽然简单但是难以想到，但这部分是整个项目最核心的部分，没有此部分的支持无法完成用户需求。
   2. 应对方案：善用工具（Google、GitHub、Stack Overflow），寻找资料，和导师多交流。
   3. 项目突发情况说明

无

* 1. 技术评审

对于概要设计和敏捷story邀请导师进行评审；

每天开晨会总结目前进度和计划并进行会议纪要；

小组每个成员进行代码走读并做好走读纪要；

多次与导师沟通算法部分以及项目中的其他问题。

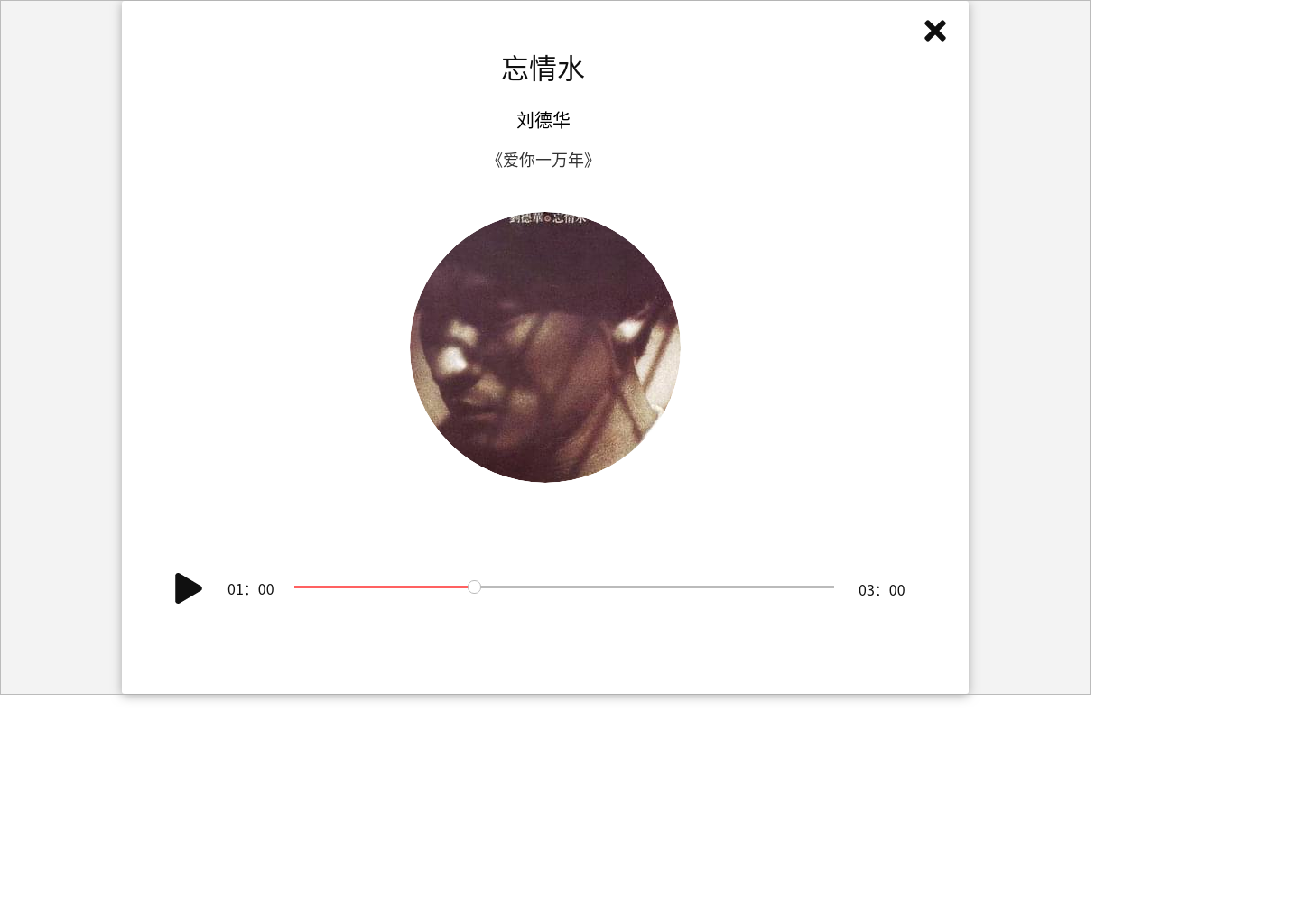
1. 技术开发总结
   1. 需求工作总结

 图3.1 需求功能描述

聊天页面整体效果图：



音乐界面效果图：



* 1. 软件整体架构描述

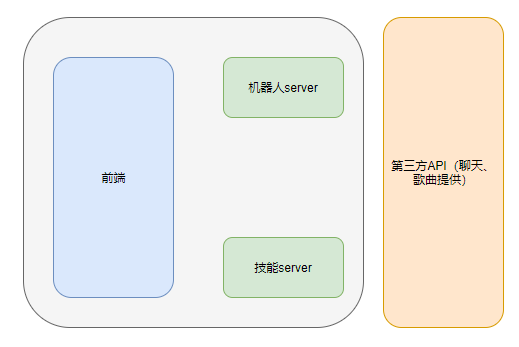


图3.2 聊天机器人整体架构图

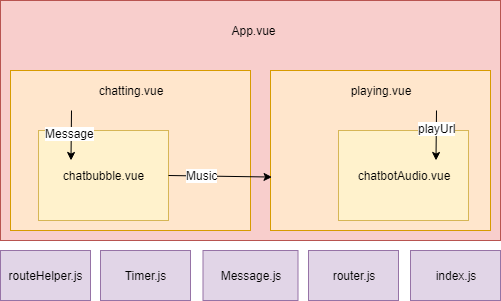


图3.3 聊天机器人前端模块架构图

* 1. 重点开发内容

1. Pattern算法

思路：

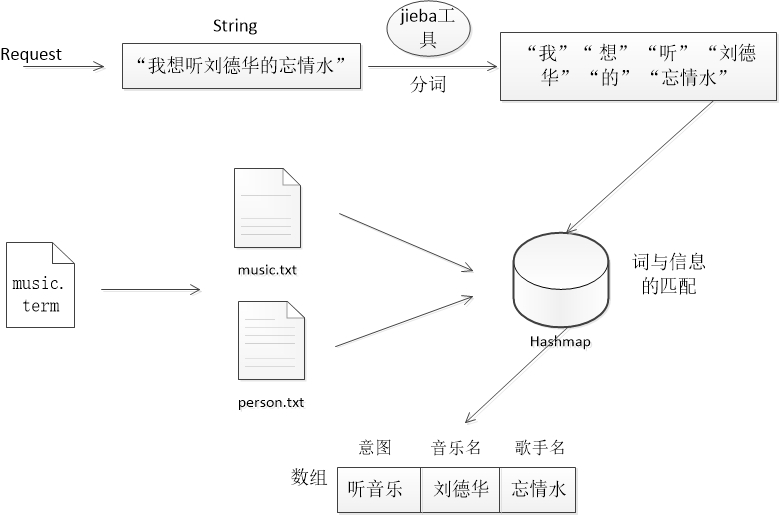


图3.4 Pattern算法思路

实现：

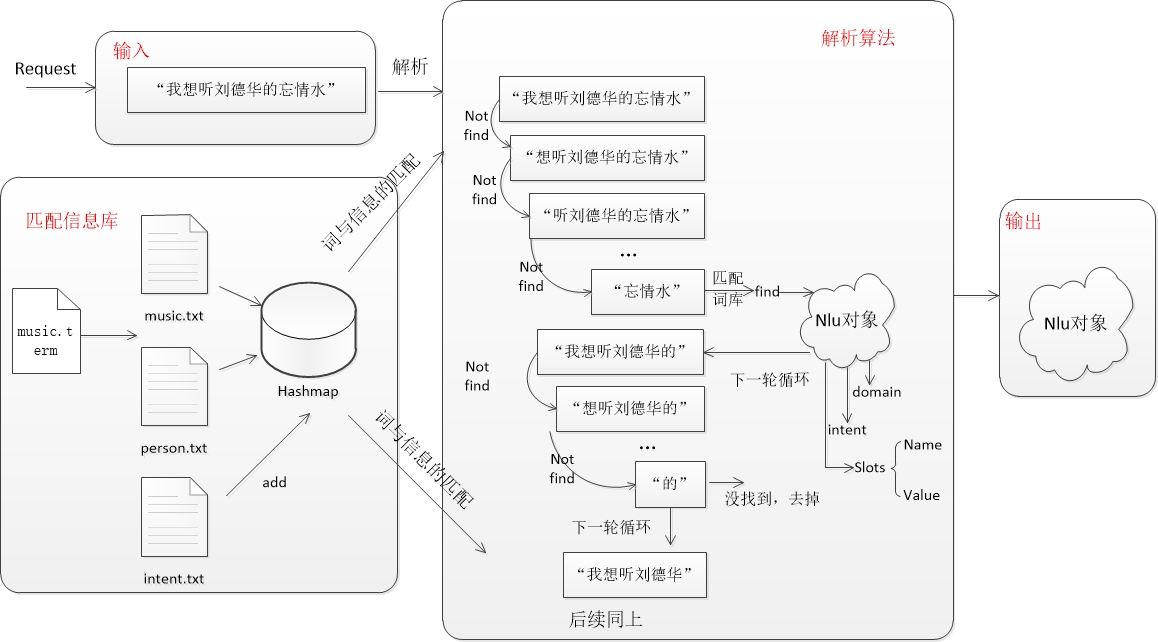


图3.4 Pattern算法思路

1. 后端服务服务接口

利用Spring boot搭建后端服务。利用spring MVC搭建后端接口。通过cors实现前端访问的跨域。

1. 第三方api调用

利用Spring内置的RestTemplate完成第三方api的接口访问。

1. 前端模块

界面设计与实现，音乐播放器的实现，以及接口访问。

* 1. 技术难点和问题

Pattern算法：如何根据输入的信息判断用户是否有听歌意图；当用户有听歌意图时，如何根据得到的信息提取出所需的歌名与歌手名的信息。

前后端数据交互：如何正确的接受前端的访问请求，合理的处理前端请求，转发到第三方的api上，并接受其数据返回，处理后返回前端。

前端模块：如何实现播放器，播放器支持进度条的点击跳转；如何实现聊天气泡和消息的自动滚动。

1. 测试和质量工作总结
   1. 项目测试情况
2. 第一轮测试用例21个，2个用例未通过，0个不可测。
3. 第二轮测试用例21个，3个用例未通过，0个不可测。
4. 第三轮测试全部通过。
   1. BUG数据汇总和分析
5. Bug分析，算法并非最优，对于特殊词语，如出现与歌名同样的词语（该词语在语句中并不代表歌名）时，会匹配到歌名信息，产生Bug。
6. 算法存在不足。闲聊状态，词槽未置空，造成输出与预测不符，产生Bug。

4.3 测试用例总结

* 1. 第一轮测试：用例共21个，通过19个，失败2个，不可测0个。
  2. 第二轮测试：用例共21个，通过18个，失败3个，不可测0个。
  3. 第三轮测试：用例共21个，通过21个。
  4. 软件质量分析

1. 对于所有的测试用例，软件都有合理的输出，保证了聊天内容的有效性和正确性。
2. 软件能够满足需求矩阵功能需求，具有较好的完整性。
3. 软件分为前端、机器人服务、技能服务和解析算法等多个模块，各个模块分工明确，通过前端与机器人服务的交互，有效的实现了前后段分离的思想。
4. 软件各个模块都有相应的信息输出提示，保证了软件运行的可靠性。
5. 机器人服务和技能服务模块根据需求提供了有效的接口，提高了程序的可读性和可扩展性。
6. 综上，软件符合设计要求，质量高！
7. 经验教训和改进建议
   1. 项目经验教训
8. 一个完整的项目一般可以划分为多个模块，比如本项目可以划分为:前端UI模块，后端服务器模块，用户请求解析模块。必须要根据项目模块，以及项目团队各成员的擅长点，明确分工，才能保证项目的高效率进行。
9. 团队在分工进行模块化编程时，各模块必须提前预留好友好的编程接口，在进行项目对接时才不会出现由于接口类型，参数不一致等问题导致无法对接。例如本项目在后端服务器开发过程中，提前定义好了解析算法所需要返回的数据类型，以及方法的框架，解析算法只需要返回一个对象便可以完美对接。
10. 拿到一个项目，应该详细地分析项目需求矩阵，用心做好项目概要设计，通过绘制用例图、时序图以及UML类图，理清楚项目的框架，进行分工，再投入代码的编写，而不是直接盲目地进行代码编写。
11. 在项目开发过程中，各模块应该编写测试用例进行测试，先排查各模块可能出现的语法错误，逻辑错误，再进行对接，系统总体进行测试，这样可以尽量避免因为某一模块的低级错误导致耗费大量的时间来进行debug。
12. 项目在开发过程中，团队成员应该多交流，提供思路、建议，及时反馈各模块的进度。同时应该多角度思考问题，不要一个方法硬磕到底，同时善于运用工具，借鉴现有的代码思路。
    1. 改进建议
13. 给关键的类、方法、属性，写上较为详解的注释，有助于代码结构的理解以及二次开发。
14. 用户请求解析算法方面，后续可以更为优化，比如歧义句的处理等等，可以探究模式匹配以外的解析算法。
15. 后续安排
    1. 前端模块

1.实现在另一个窗口打开音乐播放界面的功能，并且刷新后保留音乐信息。

2.增加音乐播放的动画效果。

3.增加本地聊天记录的存储。

* 1. 算法模块

1.改进算法，优化歧义词的处理，例如，如何将主语或谓语，与使用谓语或主语为歌名的歌曲进行更合理的区分。

* 1. 后端模块

1.增加聊天记录的云端存储。

2.配合算法与前端增加对应接口。

1. 项目资料清单

|  |  |
| --- | --- |
| 名称 | 路径 |
| 代码 | https://svn08.tclking.com/svn/rapidgrowth2019/projects/chatbot/B4\_chatbot/src |
| 概要设计 | https://svn08.tclking.com/svn/rapidgrowth2019/projects/chatbot/B4\_chatbot/doc/设计/ |
| 晨会纪要 | https://svn08.tclking.com/svn/rapidgrowth2019/projects/chatbot/B4\_chatbot/src/doc/每日纪要/ |
| 测试用例 | https://svn08.tclking.com/svn/rapidgrowth2019/projects/chatbot/B4\_chatbot/src/doc/测试/ |
| 开发计划表 | https://svn08.tclking.com/svn/rapidgrowth2019/projects/chatbot/B4\_chatbot/src/doc/ |