1. 项目技术方案
2. 项目技术架构

景点智能识别语言翻译APP的建设作为一项较为复杂的系统工程，其中的项目技术架构设计对项目实施成功与否起着至关重要的作用。

本项目采用MVC风格。这种架构使用表示层向用户呈现界面，并接受用户请求发送给业务逻辑层，使用控制器负责分发业务逻辑以处理用户请求，并调用数据访问层提供的持久性操作，数据访问层负责执行数据库持久性操作。按照“松耦合，层次化”的基本原则进行实施。

本项目的三层架构示意图如下：

接受并储存用户数据

景点识别模块

GPS模块

翻译模块

用户管理模块

广告模块

前端系统

前端展示

前端展示

前端展示

前端展示

前端展示

控制器

控制器

本项目的架构示意图

在这个结构中，用户在前端发起的如登陆请求、和其他功能请求由控制器处理，控制器可以将用户上传的图片或者其他数据储存在数据库中。然后将请求分发到对应的模块中处理

该App具体构成如下：

1. 前端展示：用户界面的设计，这是App核心。包括数据展示和查询，还有调用手机GPS功能。
2. 各个业务逻辑处理模块：这是App的核心。各个模块如何调用第三方的库来完成业务逻辑的处理
3. 数据的存储与管理。对数据库的设计使得数据便于存储和读取，也是一个重要的部分。
4. 拟采用的建模工具、编程语言、编程工具和框架

建模工具：powerdesign

编程语言：Java

编程框架：spring cloud

1. 项目风险分析和里程碑计划
2. 项目风险分析
3. 技术和架构风险：软件架构缺乏经验，新技术较多需要学习，此项风险较为最大的一项。应组织评审，及早发现缺陷。以及向专家咨询，使得项目组能够快速学习和掌握新的知识。
4. 质量低劣；在开发进度紧张的情况下，开发人员可能会减少质量保证措施，牺牲质量。开发人员能力不足也是造成质量低劣的另一个重要原因。缓解方法为：开展多层次的测试，采用迭代开发过程，让测试提前进行，及早发现问题。
5. 需求误解；软件人员误解了需求，将导致开发出的软件不符合要求而不得不返工。缓解措施为：进行多次讨论会，明确需求；组织客户、专家对需求进行评审，尽早发现需求的缺陷。
6. 时间风险：此项风险与技术风险有关，如果学习新技术的时间过长，导致开发时间不够，就会使得项目无法完工。缓解措施为：做好进度计划，按计划行事。今早发现问题尽早处理。
7. 迭代的开发过程

采用scrum的迭代开发过程，强调发挥个人的创造性和能动性，鼓励团队成员进行自我管理，使用自己认为最好的方法和工具进行开发

|  |  |  |  |
| --- | --- | --- | --- |
| **迭代** | **应对的风险** | **任务描述** | **应提交的成果** |
| 项目启动  6月20日~6月30日 | 需求误解风险；  技术与架构风险； | 需求调研与分析；  学习新技术，准备好项目所需的技术；  完善架构设计；  开发界面原型，与用户沟通，并根据反馈改进； | 项目基本模型；  完成产品代办事项；  完善的项目架构设计； |
| Sprint 1 (两周)  7月1日~7月14日 | 时间风险；  质量风险； | 架构分析与设计；  架构实现与搭建；  R1的需求分析、设计与实现；  系统测试，并获得用户反馈，进行缺陷修复与改进。 | 完成系统版本1（R1）的开发 |
| Sprint 2 (两周)  7月15日~7月29日 | 时间风险；  质量风险； | 在R1的基础上进行R2的需求分析、设计与实现；  系统测试，并获得用户反馈，进行缺陷修复与改进。 | 完成系统版本2（R2）的开发 |
| Sprint 4 (两周)  7月30日~8月14日 | 质量风险； | 安装R2，撰写用户手册，对用户进行培训，系统上线运行，并根据系统运行反馈进行缺陷修复与改进。 | 完成系统版本2的发布 |

1. 项目预期成果

项目验收时应交付以下成果：

《项目计划》

《迭代计划》（每个迭代开始前编写迭代计划）

《迭代评估报告》（每个迭代结束后编写迭代评估报告）

《SRS文档》和用例模型（.oom）

《软件架构文档》和分析设计模型（.oom）

《测试用例》和《测试报告》

《项目总结报告》

源代码和可执行代码

项目安装包

演示视频文件（包括安装、运行、功能等）

演示PPT