项

目

报

告

小组成员

组长：余旺

组员：杨涛、黄浩琪、王宏翔

**目录**

[一、项目管理相关 ................................................................................................. 2](#_bookmark0)

[1.1 代码管理平台 ................................................................................................. 2](#_bookmark2)

[1.2](#_bookmark3) 项目模型.......[......................................................................................... 3](#_bookmark3)

[二、软件相关.........................................................................................................](#_bookmark5) 4

[2.1 软件介绍 ......................................................................................................... 5](#_bookmark5)

[2.2 功能点描述 .........................................................................................................6](#_bookmark6)

[2.3 技术方案选择 .....................................................................................................7](#_bookmark7)

[三、小组任务分工 ................................................................................................. 9](#_bookmark9)



**一、项目管理相关**

* 1. **代码管理平台**

采用 Gitee 作为项目版本管理平台，Git 作为项目版本管理工具。

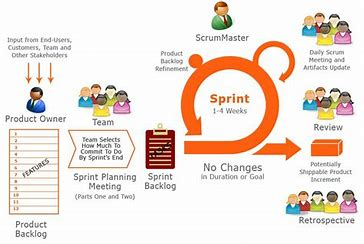






**1.2 项目模型**

**敏捷开发模型（Agile Development）**

敏捷开发（Agile Development）是一种软件开发方法，它强调通过小规模、增量式的迭代开发来快速适应变化、提高团队协作和交付高质量的软件。敏捷开发方法的核心理念是响应变化、灵活适应、以人为本，通过迭代和增量的方式快速交付软件，同时保持对需求的敏感性。

迭代开发： 敏捷开发采用迭代式的开发方法，将整个项目拆分为小的可交付的部分，每个迭代周期都会交付一个可用的软件版本。这有助于及时响应用户的反馈和需求变化。由于时间紧迫，迭代开发在我们小组的此次开发中发挥巨大作用。

增量交付： 敏捷团队通过逐步增加新功能的方式来交付软件，每个增量都是完整的、可运行的部分。这样，用户可以在项目的早期阶段就开始使用部分功能。在本次期末实训中我们小组大量采用。

需求变化： 敏捷开发鼓励灵活对待需求变化，能够在项目进行的任何时候接受并适应新的需求。这与传统的瀑布模型相比，更加注重客户需求的动态性。本次开发实现的功能动态变化。

团队协作： 敏捷团队强调协作和沟通。通过在跨职能团队中的开发人员、测试人员和业务代表之间建立紧密的合作关系，能够更好地理解和解决问题。

敏捷价值观： 敏捷开发遵循一组价值观，包括个体和互动高于流程和 工具、可工作的软件高于详尽的文档、与客户合作高于合同谈判、响应变化高 于遵循计划。这体现在我们的项目管理上。

**二、软件相关**

**2.1 软件名称**

数据库管理系统（Database Management System，简称DBMS）是一种操纵和管理数据库的大型软件，它负责建立、使用和维护数据库。DBMS对数据库进行统一的管理和控制，以确保数据库的安全性和完整性。它充当了数据库和用户之间的中间层，使得用户能够以更高级别的语言（通常是SQL）来处理数据，而不必深入了解数据库内部的工作原理。

**2.2 功能点描述**

**基本功能**

DBMS的主要功能包括但不限于：

1, 数据定义：DBMS提供了数据定义语言（DDL），供用户定义数据库的模式结构与权限约束。

2, 数据操作：通过数据操作语言（DML），用户可以对数据库中的数据进行追加、删除、查询和更新等操作。

3,并发控制：DBMS能够处理多个应用程序和用户同时或不同时刻对数据库的访问和修改，确保数据的一致性和完整性。

4,安全性控制：DBMS提供了一系列安全机制，包括用户认证、授权和访问控制，以保护数据库免受未经授权的访问和修改。

5,数据完整性维护：DBMS通过实施数据完整性规则，确保数据的准确性和一致性。

此外，DBMS还支持事务管理，确保一系列数据库操作要么全部成功，要么在出错时全部回滚，以保持数据的完整性。

**2.3 技术方案选择**

**2.3.1 C++**

本项目选择C++作为主要开发语言，这一决策是出于对C++多方面优势的充分考虑。C++继承了C语言的传统，既支持经典的过程化程序设计，又引入了基于对象的程序设计理念。这使得我们在处理问题时既能使用传统的过程化方法，又能充分利用面向对象的特性，如封装、继承、多态等。

以下是C++的一些显著优势：

* 语言简洁灵活： C++的语法简洁而灵活，提供了丰富的运算符，使得编码变得更加方便快捷。
* 高效的目标代码生成： C++生成的目标代码质量高，执行效率优越，而且具备良好的可移植性，使得我们的程序能够轻松在不同平台上运行。
* 适用于大规模项目： C++在可重用性、可扩充性、可维护性和可靠性等方面表现出众，非常适合开发大中型系统软件和应用程序。
* 支持面向对象编程： C++提供了强大的面向对象编程机制，包括信息隐藏、封装函数、抽象数据类型、继承、多态、函数重载、运算符重载、泛型编程（模板）等特性。这使得团队开发更为简单，同时提高了代码的可维护性和可拓展性。

**2.3.2 QT**

本项目选择Qt作为图形用户界面（GUI）开发框架。Qt是一种跨平台的C++框架，专注于为应用程序提供创建艺术级图形界面所需的所有功能。它是一个完全面向对象的框架，易于扩展，并允许真正的组件编程。Qt在商业领域取得了巨大成功，成为全球数千种应用程序的基础，并且是Linux桌面环境KDE的基础。

以下是Qt框架的一些优势：

1. XML支持： Qt具有对XML的内置支持，使得处理和交换XML数据变得更加便捷。
2. 大量的开发文档： Qt提供了丰富的开发文档，使开发者能够更容易理解和使用框架的各个方面。
3. 2D/3D图形渲染支持： Qt支持2D和3D图形渲染，并且提供对OpenGL的支持，使得开发出视觉效果良好的图形界面变得更加容易。
4. 优秀的跨平台特性： Qt的跨平台特性非常突出，支持多种操作系统，使得开发的应用程序能够在不同系统上轻松运行。
5. 面向对象： Qt具有良好的面向对象特性，通过强大的封装机制，模块化程度高，可重用性强，开发者可以更方便地进行模块化设计。
6. 信号/槽机制： Qt引入了安全类型的信号/槽机制来替代传统的回调函数，简化了各个组件之间的协同工作。
7. 丰富的API： Qt拥有丰富的API，包括250多个C++类，提供了多种功能，如集合、序列化、文件、I/O设备、目录管理、日期/时间等，甚至包括正则表达式处理功能。

Qt框架以其强大的功能和便捷的开发特性成为本项目的理想选择，有助于实现高效、现代化的图形用户界面。能有效的协助我们小组的软件开发过程。

**三、小组任务分工**

余旺：项目管理，技术开发（25%）

杨涛：ppt制作，技术开发（25%）

黄浩琪：美术表现，技术开发（25%）

王宏翔：ppt制作，技术开发（25%）