**软件开发计划文档**

**题目：**基于python实现的模拟C/S模式的考研信息查询系统

**专业：**人工智能

**日期：**2023年4月15日

**小组成员人数：** 6 人

**小组成员名单：**

王月（组长）、艾思琦、常东阳、蒋湘、王怡正、赵亮

**目录**

[1 引言 3](#_Toc17453)

[1.1 编写目的 3](#_Toc11969)

[1.2 背景 3](#_Toc4504)

[1.3 定义 4](#_Toc26825)

[1.4 参考资料 4](#_Toc28099)

[1.5 标准、条约和约定 4](#_Toc25123)

[1.6 项目整体设计的WBS 4](#_Toc21071)

[1.7 开发计划文档编写的WBS 5](#_Toc21590)

[2 项目概述 5](#_Toc26444)

[3 项目团队组织 8](#_Toc28986)

[3.1 组织结构 8](#_Toc8497)

[3.2 人员职责 8](#_Toc9594)

[3.3 协作与沟通 8](#_Toc9156)

[4 实施计划 9](#_Toc28774)

[5 支持条件 12](#_Toc14641)

[6 关键问题 13](#_Toc810)

## 1 引言

1.1 编写目的

为了保证项目团队按时、保质地完成项目目标，同时便于项目团队成员更好地了解项目情况，使项目开发中各个过程有序且便于管理，有必要对项目生命周期内的工作任务范围、工作任务分解、项目团队组织结构、团队成员的工作责任、开发期间团队内沟通协作方式、开发进度等内容以书面形式描述出来，作为项目团队成员之间的共识与约定、项目生命周期内的所有项目活动的行动基础、项目团队开展和检查项目工作的依据，从而明确开发目标，保证开发质量，使项目开发顺利进行。

本项目开发计划用于从总体上指导《基于python实现的模拟C/S模式的考研信息查询系统》项目顺利进行并最终得到通过评审的项目产品。

本项目开发计划面向项目组全体成员。

1.2 背景

由于就业形势、个人追求等因素，近几年来选择考研的人越来越多。从国家统计局公开的数据可以看出，考研报名人数在近几年一直稳定增长，2022年考研报名人数更是从377万急剧升至457万，报考人数仍有只增不减的趋势。

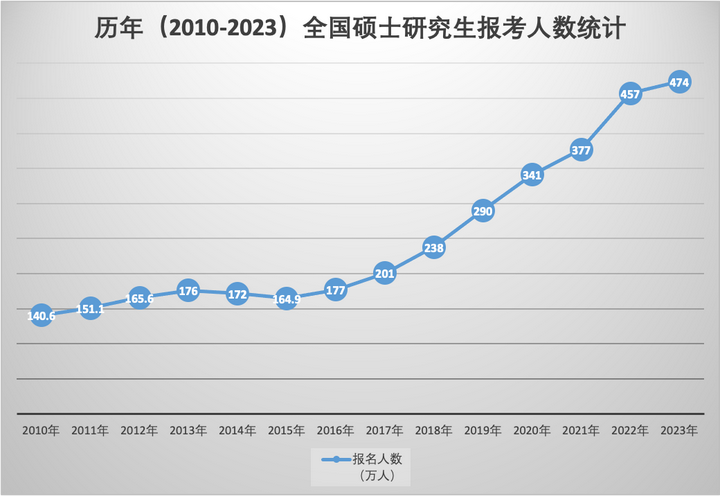


图1 历年（2010-2023）全国硕士研究生报考人数统计（国家统计局）

同时，通过学院内部的调查以及与不同专业、不同学校的同学们交流也可以看出，很多人将考研作为毕业去向的首选。

不同于高考时老师和学校会为学生尽可能提供报考信息，也不同于资格证考试和等级考试等有明确的考试目标和考试内容的考试，很多同学在面对考研时十分茫然，我们搜集了很多计划考研的同学们对考研信息搜集的需求，结合这些需求设计了我们的系统。

《基于python实现的模拟C/S模式的考研信息查询系统》主要功能是，为广大考研学子提供一个整合的、全面的信息搜集平台，通过多种筛选条件精确定位目标院校的详细考研信息，方便用户根据自身报考需求快速获取对应的考研咨询。

1.3 定义

专门术语：

SQL SERVER：系统服务器所使用给的数据库关系系统（DBMS）。

SQL：一种用于访问查询数据库的语言。

缩写：

系统：若未特别指出，统指本基于python实现的模拟C/S模式的考研信息查询系统。

SQL：Structured Query Language（结构化查询语句）。

UML：统一建模语言，一套用来设计软件蓝图的标准建模语言，是一种从软件分析、设计到编写程序规范的标准化建模语言。

UDP：User Datagram Protocol（无连接传输层协议）。

1.4 参考资料

文档格式按照我国GB/T8567-1988国家标准和IEEE/ANSI830-1993标准规范要求进行，包括：

《软件工程项目开发文档范例》

《软件工程国家标准文档》

《需求说明书》

《软件需求说明书编写规范》

参考书籍包括：

《软件工程及应用》 张斌，郭军主编 东北大学出版社。

1.5 标准、条约和约定

本项目遵从以下标准：

GB/T 13702-1992 计算机软件分类与代码

GB/T 19003-2008 软件工程

GB/T 5538-1995 软件工程标准分类法

GB/T 9385-2008 计算机软件需求规格说明

GB/T 5532-2008 计算机软件测试规范

GB/T 18221-2000 信息技术程序设计语言

GB/T 8567-2006 计算机软件文档编制规范

1.6 项目整体设计的WBS

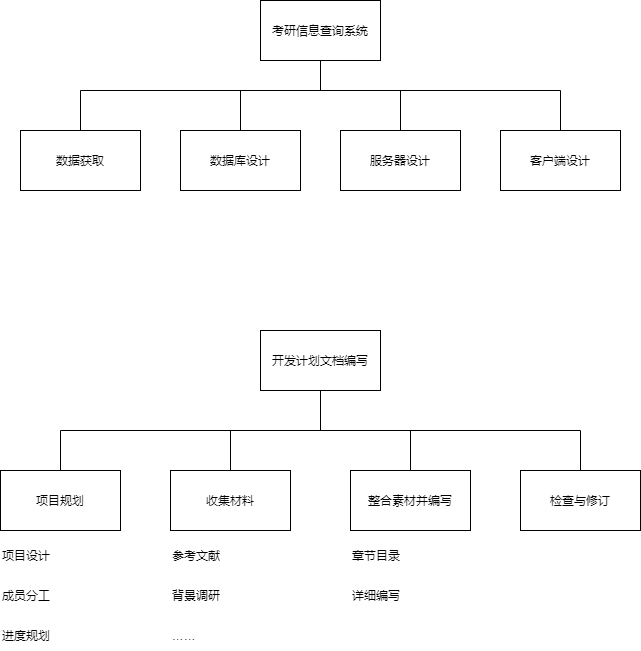


图2 项目整体设计WBS图

1.7 开发计划文档编写的WBS

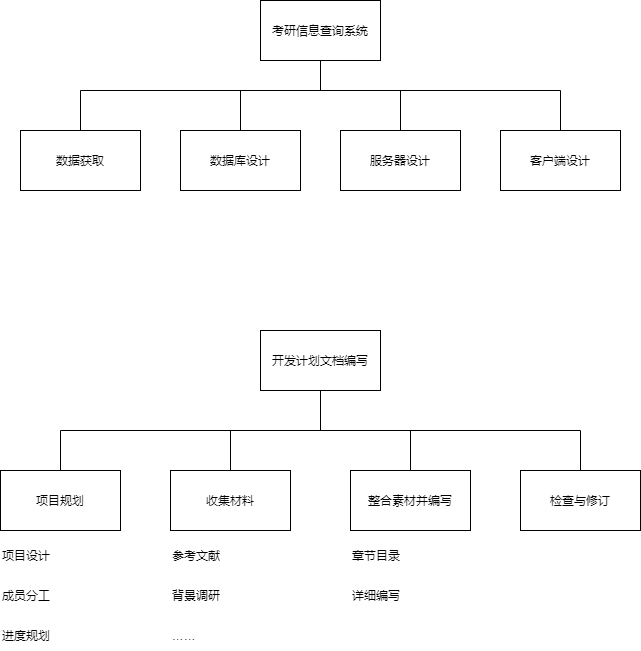


图3 开发计划文档编写的WBS图

## 2 项目概述

随着考研报名人数逐年增长，考研已经成为很多人的选择，同时由于考研的报考特性，考研备考前期信息获取以及院校选择通常极大影响最后考研的结果以及备考难度。很多同学在准备考研时容易发现，尽管官方公开的信息比较全面，但是难以精准搜索或从多条件匹配同学们的信息获取需求。

当考研报名人数逐年攀升，越来越多的人将考研作为自己未来发展的一个重要选择。不过，考研的报考特性决定了备考前期的信息获取和院校选择对于最终考研结果和备考难度有着极大的影响。在备考的过程中，很多同学都会遇到这样的问题：尽管官方公布的信息相对较为全面，但是在进行精确搜索时却十分困难，而从多种条件匹配中获取所需信息也相当具有挑战性。

在备考考研之前，考生首先需要做的是了解自己所处的条件和情况，包括自己的学术背景、职业规划、所选专业的就业前景等。这些因素都会影响到备考的方向和重点。因此，在选择报考院校和专业时需要考虑清楚自己的实际情况和需求。但是，在进行院校和专业选择时，仅仅依赖官方公开的信息往往无法满足考生的需求。为了获取更全面、更准确的信息，考生需要采取多种途径，如通过社交媒体、考研论坛、资深考生的分享等方式获取第一手资讯。

此外，在备考考研前期，考生还需要对各个院校的招生政策、考试科目、考试难度等方面进行深入了解。这些信息对于考生制定备考计划和备考策略至关重要。然而，这些信息往往零散分布在各种渠道中，需要考生进行筛选和整理。在这个过程中，考生可以通过查找各种考研资料和指导书籍、请教资深考生和辅导老师等方式获取相关信息。

综上所述，在备考考研前期，考生需要投入大量的时间和精力，多方面地获取、筛选和整理信息。

基于这些问题，我们开发了一款针对考研信息收集的系统。我们统筹了多个官方网站（渠道）给出的相关信息，进行了信息整合，并设计了一个查询系统，根据用户需要快速且精确地检索所需的考研信息，旨在整合分布在互联网各处的离散信息，为搜集信息的同学节省时间与精力，并提供对比功能，便于同学们进行决策。

本系统采用C/S模式，利用python进行开发，后台数据管理系统使用MySQL数据库，系统运行平台支持Windows10、Linux等。主要实现的功能如下：

（1）获取准确且及时的考研信息

（2）支持用户进行多条件查询对比，提供全面的院校招生政策信息

（3）支持管理员后台监测数据库情况

（4）获取用户反馈进行系统升级和维护

2.1项目目标及需求分析

本系统的主要目标包括以下几个方面：

1、提供全面、准确、及时的考研信息：系统旨在为广大考研人员提供最新、最全面的考试资讯和考研动态信息，包括考试时间、科目、形式等详细信息，以及最新的考研政策和报考指南等，满足考研人员对考试信息的查询需求。

2、保证系统获取的信息的正确性和及时性：后台数据需要保证定时更新，并预留更新接口，在成绩查询以及调剂阶段也许需要提高系统更新频率以保证用户能及时获取所需信息。

3、提高系统知名度和用户接受度：通过广告宣传、社交媒体推广等方式提高系统知名度和用户接受度，吸引更多的用户使用系统。在此基础上重视用户反馈，改进系统功能，以提供更优质的服务，提升用户体验和满意度。

4、提高系统安全性和稳定性：系统将采取一系列安全措施，包括数据加密、安全性检测、防御恶意代码和漏洞扫描等，保障系统的安全和稳定运行，确保用户的数据和隐私安全。

2.2产品目标与范围

我们的产品主要需实现以下几个目标：

1、提供全面、准确、快捷地查询对比服务

2、保证信息的准确性和及时性

3、保证系统的稳定性和安全性

产品主要面向广大考研学生群体，任何需要获取考研信息的用户都可以使用本系统进行精确快捷高效的查询服务，减少信息获取的成本，同时也可以通过反馈入口对我们提供的服务进行反馈，帮助我们优化系统并提供更好地服务。

2.3假设与约束

在设计本系统时，需要根据现实情况进行一些假设和约束的制定。下面列举了一些可能的假设和约束：

假设：

1、用户对考研信息查询的需求持续存在，且人数较多。

2、考研目标院校信息的更新频率较高，且分布于互联网各处，较难进行收集整合，系统需要及时更新数据库。

3、用户希望系统的检索更加智能化以符合需求。

4、用户需要系统保证数据的时效性和准确性。

约束：

1、考研信息查询系统需要遵守相关法律法规和规范。

2、考研信息查询系统需要确保系统的安全性和稳定性。

3、考研信息查询系统需要保护用户的隐私和数据安全。

4、 考研信息查询系统需要提供友好的用户界面和交互方式，以便用户使用。

5、考研信息查询系统需要确保数据的准确性和及时性。

通过以上的假设和约束的制定，可以更好地指导系统的设计和开发，以便满足用户的需求，并保障系统的稳定运行和用户数据的安全。

2.4 项目工作范围

考研信息查询系统的项目工作范围包括但不限于以下几个方面：

需求分析：分析用户的需求和要求，明确系统的功能模块和设计方案。

系统设计：制定系统的整体设计方案，包括系统结构、模块划分、数据模型和接口设计等。

开发实现：根据系统设计方案，进行软件开发和编码实现，包括前端、后端服务器通信和数据库等。

测试与调试：对开发完成的系统进行功能和性能测试，进行错误和缺陷的调试和修复。

部署和运维：将系统部署到服务器上，并进行系统的维护和运营，包括系统升级、数据备份和恢复等。

用户培训和支持：为用户提供系统使用培训和技术支持，以便用户能够顺利使用系统。

以上工作范围是考研信息查询系统开发过程中的主要工作内容，其中每个环节都需要严格执行，确保系统的质量和稳定性。

2.5 应交付成果

2.5.1 需完成的软件

可进行前后端交互的查询系统，完整的通讯模块和对应的数据库系统。

2.5.2 需提交用户的文档

软件需求说明书、用户测试说明、软件使用说明。

2.5.3 须提交内部的文档

可行性分析报告、项目开发计划、软件需求说明书、概要设计说明书、详细设计说明书、测试计划、测试分析报告、开发进度报告、软件问题报告、源程序。

2.5.4 应当提供的服务

结合需求分析结果可以总结出我们应当提供的服务如下：

1、提供全面、准确、及时的考研信息，提供多条件筛选考研信息服务

2、管理员可后台监测数据库的被查询情况，及时监控异常交互查询，查看用户反馈信息

3、用户可通过反馈入口反馈使用中遇到的问题

2.6 项目开发环境

硬件环境：电脑。

软件环境：Windows10/Linux，Git，IntelliJ IDEA，Visual Studio Code，PyCharm Community Edition 2022.1.1，Python3，MySQL

编程语言和框架：Python，Tkinter，selenium

版本管理和协作工具：Git

测试和调试工具：IntelliJ IDEA，Visual Studio Code，PyCharm Community Edition 2022.1.1，Jupyter

文档和报告工具：Office、LaTeX等文档工具，以及UML图等报告工具。

2.7 项目验收方式与依据

在发布前首先进行小组内评审，代码需与文档说明保持一致，代码书写风格统一，采用标准规范且没有以下错误：由于软件缺陷造成丢失数据，不符合设计要求，响应时间太长无法接受等问题。

## 3 项目团队组织

3.1 组织结构

为了完成本系统，我们组建了一个临时项目团队——玄不救非氪不改命小组，其中有项目经理、软件需求师、软件架构师、软件工程师、数据库工程师、UI设计师、质量控制员、文档人员、测试人员，如图所示。

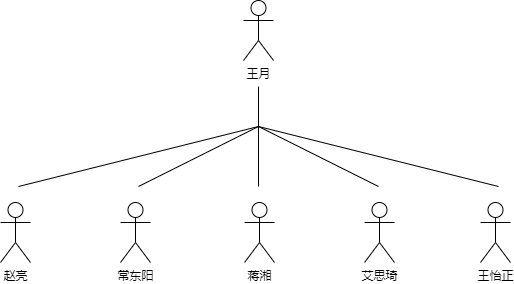


图4 团队组织结构图

3.2 人员职责

根据团队内每个成员的能力特长和技术能力偏好，我们对每个人进行了职责划分。

表1 人员职责表

|  |  |  |
| --- | --- | --- |
| **姓 名** | **角 色** | **工 作 描 述** |
| 王月 | 组长 | 项目经理、软件架构师、软件工程师、UI设计师 |
| 赵亮 | 组员 | 软件架构师、软件工程师、测试人员 |
| 常东阳 | 组员 | 软件需求师、软件工程师、质量控制员 |
| 蒋湘 | 组员 | 软件需求师、软件工程师、测试人员 |
| 艾思琦 | 组员 | 数据库工程师、软件工程师、质量控制员 |
| 王怡正 | 组员 | 软件工程师、UI设计师、文档人员 |

3.3 协作与沟通

3.3.1 内部协作

我们将采用Git实现多人合作开发和代码管理，同时对项目开源：

<https://github.com/KyoiLin/SEWork-23-spring>

3.3.2 外部沟通

我们计划招募数名计划备考、正在备考、考研结束的同学对我们的项目进行测试和评估，从而完善我们的功能。为了保证测评获得的建议的广泛程度，这一测评计划采取线上线下结合的形式，尽力囊括不同考研目的和需求、不同学校、不同专业、不同年级的同学（后续可能会招募社会考生进行测评）。

## 4 实施计划

4.1 过程模型的选择

结合我们的需求以及小组成员的能力和开发经验，我们选择使用RAD模型。

由于我们设计的系统规模不大且技术风险较低，大部分成员都有类似项目的开发经验且对开发环境比较熟悉，且由于项目组成员具有一定的类似系统的开发经验，开发过程中有很多可复用的构件，选用RAD模型进行开发十分贴合我们的开发需要。

RAD模型能给我带来一些额外的益处：由于小组成员工程能力比较强，RAD模型多小组并行开发的模式在某种程度上给予每个成员自由发挥的空间，明确目标并规范接口后，各小组的工作相互透明，在各小组负责的范围内每个成员可以选择自己最熟悉的方式进行功能的实现。不过这种模式仅适合较小的项目开发，当项目较大时可能会带来管理上的困难，由于本次大作业的项目并不算较大型的项目，以上条件是成立的。

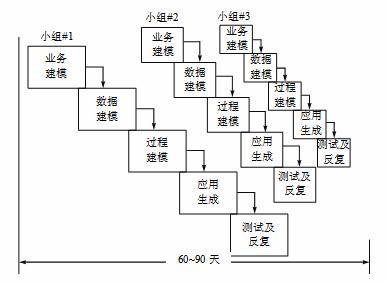


图5 RAD模型

同时，考虑到项目组成员近期事务繁忙，这就要求我们提高开发速度并保证开发质量，RAD模型十分适合这一需求。

再者，我们计划在开发过程中招募数名计划备考、正在备考、考研结束的同学对我们的系统进行测试，使用RAD模型可以让测试人员对项目中细分的部分进行反馈，可以对功能进行针对性地优化。同样的，当我们的项目出现较大变更时，RAD模型也能让我们及时进行调整，调整时比较灵活。

由于RAD模型属于线性模型，而在实际开发过程中，为了保证开发的质量，我们还参考了一些迭代和增量的思想。各小组在开发过程中会同步进行多次版本迭代以保证功能的可行性和稳健性并尽量提升性能，在前期版本进行测试后也许会发现新的问题或需求，此时则利用增量的思想进行系统功能的补充和完善并更新版本。

4.2 风险评估及对策

1、数据来源不可靠风险：我们对一些同学对考研信息获取的需求进行分析得知，目前网络上有各种各样的考研信息，一些个人或机构整理的信息不一定可靠，官方信息比较可靠但是通常比较宽泛，难以满足同学们针对学科或院校的信息需求，同时很多同学在前期报考时通常会综合考虑多所院校，在整理和横向对比这些院校的信息时往往感到十分迷茫或觉得十分繁琐。针对这一问题，我们只把官方信息纳入信息获取的范畴，同时对各种各样繁杂的信息进行整合，满足同学们进行横向对比的需要。

2、信息时效性风险：由于考研系统需要处理大量的信息，例如成绩查询、考试通知等，且在分数线公布后、调剂报名前后，各院校的信息通常会增多，因此信息时效性风险可能会导致考生获得错误的或落后的信息。对策可以是对系统数据进行及时更新和审核，确保信息准确无误。

3、服务中断风险：系统出现故障或攻击可能会导致系统服务中断，给考生造成不便。对策可以是建立系统备份和灾难恢复机制，保证系统稳定运行。

4、安全漏洞风险：由于考研系统需要与外部网络进行通信，因此可能会存在安全漏洞，被黑客攻击导致系统被入侵。对策可以是定期对系统进行安全漏洞扫描和测试，及时修复和升级系统。

以上内容可以整理为如下表格。

表2 风险及对策表

|  |  |  |  |
| --- | --- | --- | --- |
| **风险排序** | **风险项名称** | **风险描述** | **风险缓解方案** |
| 1 | 数据来源不可靠 | 网络上充斥着各种各样的考研相关信息 | 整合筛选官方信息 |
| 2 | 信息时效问题 | 考研信息通常具有极强的时效性 | 根据需要调整更新频率 |
| 3 | 系统不稳定 | 系统故障等问题 | 建立系统备份和灾难恢复机制 |
| 4 | 安全漏洞 | 系统需与外部通信可能带来安全隐患 | 定期对系统进行安全漏洞扫描和测试，及时修复和升级系统 |

4.3 工作流程

4.3.1 需求分析和可行性分析

需求分析和可行性分析是项目设计中十分重要的一环，项目启动后我们计划马上进行可行性分析，根据可行性分析的结果确定项目大致的框架，之后根据这一框架进行需求分析。

当项目启动、可行性分析完成、开发角色确定后，我们计划首先让设计开发人员对业务流程、管理方式进行分析，同时利用互联网、线下调研等渠道收集、整理资料，确定用户需求后，开始对软件必须完成的功能进行定义，并在此基础上进行数据定义。

4.3.2 系统设计

这一部分需要在需求分析给出的系统必要功能的基础上完成对整个系统的分析设计，对概念模型、存储模式、完整性控制、存储权限等进行定义，对系统功能各模块进行详细设计并定义数据库总体结构以及编码命名规范。这一部分结束后需给出系统架构。

4.3.3 详细设计

这一部分需要在系统设计的基础上进一步细分、完善每一个功能的设计，同时进行界面设计，针对较复杂的功能或核心功能，这一部分需给出程序框图方便后续开发。

4.3.4 编码及测试

这一部分需需要完成程序设计和系统测试。根据之前的功能设计进行编码实现，其中包括获取数据、数据库建立、服务器和客户端的搭建、服务器和客户端功能的实现。同时为了避免错误累积，我们选择采用边开发边测试的基本模式，对每个模块都要求进行单独测试，对系统处理逻辑、错误分析和诊断等进行大规模测试，保证系统的稳健性和代码的质量。

4.3.5 文档、产品部署

项目进行过半时即可开始各类文档的编写，同时可以进行用户测试。

4.3.6 项目总结

项目结束后会对项目研发、部署等开发过程中遇到的问题、经验等进行交流学习，有助于项目经验的积累，帮助项目组每个成员进行总结和提高。

4.4 工作任务分解

在分工上我们首先考虑每个人的能力特长，再根据每个人的时间安排以及项目计划进行详细分工，追求尽量发挥每个人的特长，提高开发效率。

表3 分工表

|  |  |  |
| --- | --- | --- |
| **工 作 内 容** | **负 责 人** | **参 加 人 员** |
| 项目可行性分析 | 王月 | 全体人员 |
| 项目开发报告 | 王月 | 王月、王怡正 |
| 需求分析 | 蒋湘 | 全体人员 |
| 系统分析 | 王月 | 全体人员 |
| 详细设计 | 王月 | 全体人员 |
| 数据获取（数据更新） | 赵亮 | 赵亮 |
| 数据库设计与建立 | 艾思琦 | 艾思琦 |
| 查询功能设计与实现 | 王月 | 王月、赵亮 |
| 管理员功能设计与实现 | 常东阳 | 常东阳、蒋湘、王月 |
| 服务器端的设计与实现 | 赵亮 | 赵亮、蒋湘、王怡正 |
| 客户端的设计与实现 | 王月 | 王月、艾思琦、常东阳 |
| 界面设计与实现 | 王怡正 | 王月、王怡正 |
| 测试 | 王月 | 全体人员 |
| 文档编写 | 王月 | 王月、王怡正、艾思琦 |
| 用VISIO绘制数据流图、E-R图等各种图形 | 王月 | 王月、王怡正 |
| 后期维护 | 王月 | 全体人员 |

4.5 总体进度计划

通过甘特图规划的项目总体进度计划如图6所示。

由于我们选择了RAD模型作为过程模型，在实际进行开发和设计的过程中，很多步骤可以并行，同时，通过使用可复用构件可以极大缩短开发时间，提高开发效率。但是相对的，当一些功能模块之间存在逻辑上的相关性时，这些模块的开发和测试是一环扣一环的，这就需要上游开发人员尽可能保证质量，从而让下游开发人员在基于上一模块进行开发和测试时避免错误，同时，将任务拆解并细分小组进行开发意味着在最后进行模块整合时存在更大的工作量，需要更多的时间进行调试，测试阶段也需要投入更多时间来保证各小组的成果的质量。

结合以上分析，且由于这是我们第一次团队合作进行正式的软件项目开发，我们更希望开发过程中尽量避免问题，于是我们在需求分析、概要设计、详细设计以及测试四个阶段安排了更多的时间，尽量保证项目在一开始就有明确的目标和方向，保证项目的质量和稳定。

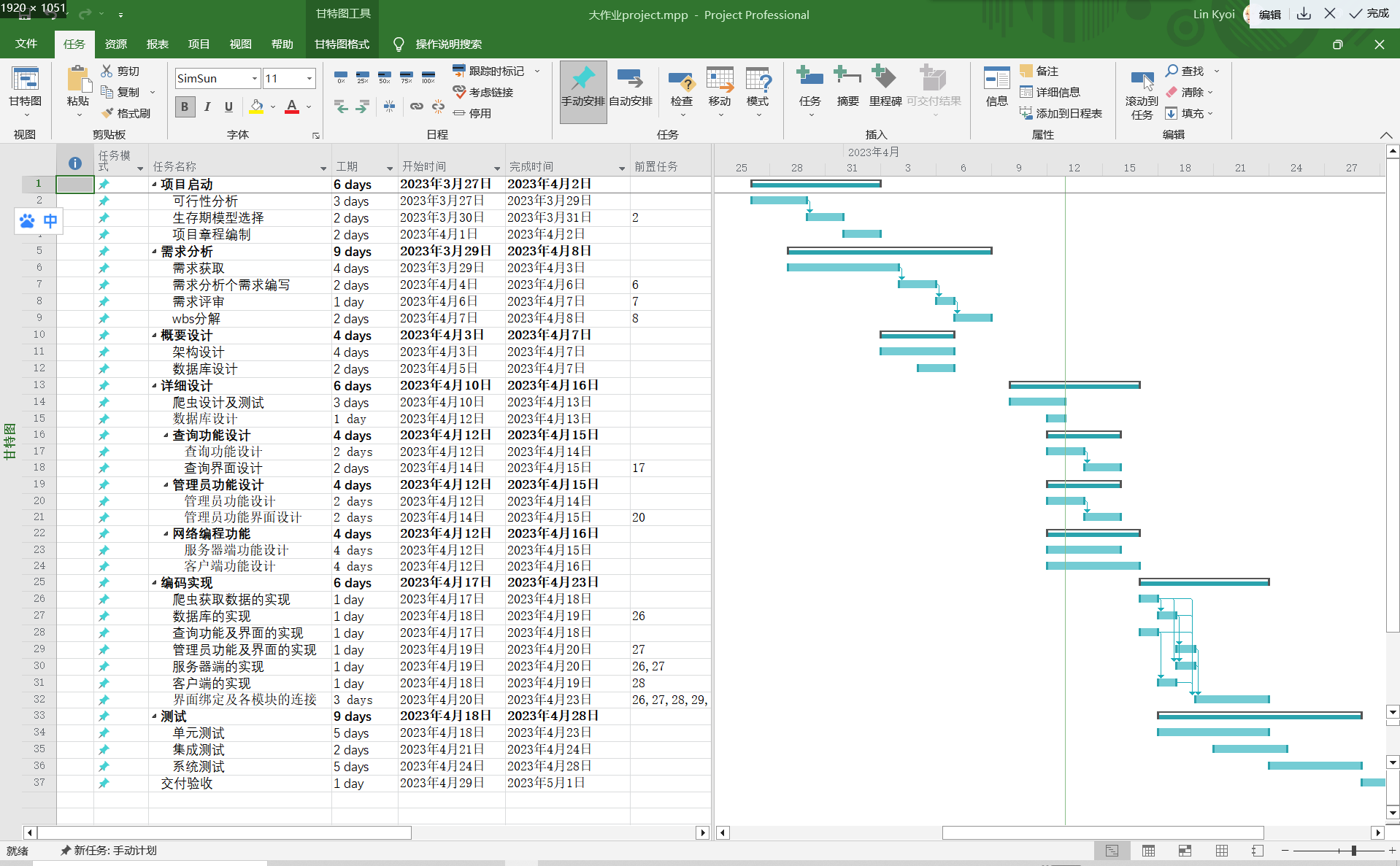


图6 项目甘特图

## 5 支持条件

5.1 内部支持

硬件方面，如果部署在实际情况，需要考虑服务器等因素，由于本项目计划进行实际场景的模拟，所以只需设备可连接局域网即可。

软件方面，需要Python3.6运行环境及相关开发工具，数据库采用MySQL，项目可运行在Windows10和Linux下。

开发人员需要熟悉Python或SQL语句。

5.2 客户支持

参与前期调研和后期用户测试。

## 6 关键问题

经过分析，我们认为影响本计划完成的主要问题有：

1、用户需求不清晰，存在误解及二义性。

2、第一次团队合作开发软件，开发人员缺乏合作经验和开发经验。

3、概要设计过于模糊导致后续设计出现歧义和混乱。

4、团队成员能力参差不齐导致开发进度不一致，部分功能难以及时交付。

5、接口等规范考虑不周导致后续模块整合时出现问题。

6、时间有限，没有足够的开发时间。