**武汉大学计算机学院**

**本科生课程设计报告**

**珞做TodoList项目开发计划**

专 业 名 称 ：计算机科学与技术弘毅班

课 程 名 称 ：软件工程

团 队 名 称 ：珞做TodoList

指 导 教 师 一：伍春香 副教授

团 队 成 员 一：李品正（2020300004081）

团 队 成 员 二：贺镇涛（2021300004071）

团 队 成 员 三：洪志枫（2021300004026）

团 队 成 员 四：刘浩 （2021300004039）

二○二四年五月

**郑 重 声 明**

本团队呈交的设计报告，是在指导老师的指导下，独立进行实验工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本设计报告不包含他人享有著作权的内容。对本设计报告做出贡献的其他个人和集体，均已在文中以明确的方式标明。本设计报告的知识产权归属于培养单位。

团队成员签名：李品正 贺镇涛 洪志枫 刘浩 日期：2024.05.06

**1 简介**

1.1 项目概述

1.1.1 意图、范围和目标

本项目意在为用户提供一个方便管理任务和日程的工具，以帮助他们提高工作效率和组织能力，具体包括个人任务日程规划和团队的任务派遣和管理。

项目瞄准对时间规划有困难，或者需要参与团队协作，对任务规划有较高追求的用户。解决该范围内用户棘手的困难。

主要计划实现的的目标如下：

个人任务规划：用户可以创建个人任务清单，设定截止日期，优先级等。软件提供友好的时间规划辅助工具、跟踪提醒组件。

多个团队中的任务派遣和管理：支持团队协作，允许团队成员互相指派任务、查看任务进度和共享资源；支持团队内的权限管理（普通成员和团队管理者），支持多团队、多重身份应用场景。

完善的权限管理和信息安全保障：支持细粒度多角色的权限分配和控制、支持密文形式存储同步、支持外部临时访客。

1.1.2 设想和限制

无。

1.1.3 可交付项目

一个实现了预期功能的桌面应用。

1.1.4 时间表和预算概述

|  |  |
| --- | --- |
| 时间 | 描述 |
| 6月1日 | 系统交付测试 |
| 6月7日 | 系统最终交付 |

预算计划：

1. 人力成本：0元

2. 系统运行服务器：100元

3. 运行、维护与测试费用：0元

4. 计算机设备费用：0元

5. 其他费用：0元

1.2 项目管理计划的演化

无。

**2 参考材料**

[1]毛新军，董威. 软件工程：从理论到实践.北京：高等教育出版社.2022.

**3 定义和缩略语**

本文档中使用的术语包括：

1. **用户(普通用户)**：任何本系统的使用者都是用户，拥有基本的个人任务管理功能。任何用户均可创建或受邀加入/申请退出团队。团队内的普通用户可以提出协作任务请求。
2. **团队所有者**：本系统中团队最高权限的所有者，可以授予/修改/驳回团队内其他成员在本团队的权限，可以解散团队。创建团队的用户默认为团队所有者，也可转让给其他团队成员。
3. **管理员**：团队内的普通用户接受团队所有者授予的不同于普通用户的特别权限而成为管理员。特殊权限包括：邀请/移出团队普通成员、邀请/移除访客、受理用户的协作任务请求。
4. **访客**：团队外的成员收到群组所有者或管理员的邀请，获得限期内查看部分团队任务列表和接受协作/拒绝协作任务请求的权利。访客成员不能主动提出协作请求。
5. **操作人**：指操作任务条目的用户。
6. **参与人**：任务条目涉及的协作用户。协作任务的参与人必须在任务时间段内空闲。
7. **任务计划**：至少具有任务名称的数据条目，可以包括时间、参与人、提醒方式、反馈方式、执行周期、其他备注。
8. **协作任务与非协作任务**：协作任务指多人共同参与，在所有参与人的时间表上都占用时间段。非协作用户为单人任务。
9. **任务的分派**：订立好的任务被分派到参与人的设备上，建立提醒。

**4 项目组织**

4.1 外部接口

与本项目QA小组通过各种方式及时进行沟通。

4.2 内部结构

内部组织结构采用项目经理和成员的形式。项目经理是项目的领导者，负责规划、执行和监督项目的各个方面。项目团队成员是在项目中负责具体任务的人员，他们可能来自不同的部门或团队。

4.3 角色和责任

|  |  |  |
| --- | --- | --- |
| 角色 | 人员 | 责任 |
| 项目经理 | 李品正 | 规划、执行和监督项目的各个方面 |
| 前端工程师 | 李品正 | 系统界面设计与实现 |
| 后端工程师 | 洪志枫 | 设计上层业务逻辑 |
| 后端工程师 | 贺镇涛 | 设计TodoList的信息管理数据库 |
| 后端工程师 | 刘浩 | 设计用户数据存储数据库 |

4.4 责任矩阵

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 李品正 | 贺镇涛 | 洪志枫 | 刘浩 |
| 架构设计 | A | S | R | S |
| 系统界面设计与实现 | A | R | S | S |
| 上层业务逻辑设计 | R | S | A | S |
| 信息管理数据库 | S | A | R | S |
| 用户数据存储数据库 | S | R | S | A |
| 组装联调 | S | A | S | R |
| 测试修改 | S | S | A | R |
| 验收测试（完善） | R | S | S | A |

A=Accoutable，负责人

R=Reviewer，评审人

S=Support，提供支持者

**5 管理过程计划**

5.1 启动计划

**1.需求分析**：

定义项目的目标和范围，对项目需求进行细致周密的分析，确定项目的功能

**2.市场调研**：

确定目标用户群体，并了解他们的需求和偏好。

研究市场上成熟的类似应用，分析其优势和劣势，以及市场定位。

**3.技术栈选择**：

选择合适的开发技术和框架，确定数据库选择。

**4.团队组建**：

确定项目团队成员，包括开发人员、设计师和测试人员等。

分配任务和角色，确保团队成员的职责清晰。

**5.项目计划**：

制定项目开发的时间表和里程碑，确保项目进度可控。

考虑到测试、反馈和修复的时间，合理安排开发周期。

**6.原型设计**：

确保原型包含所有主要功能，并与团队成员和目标用户进行确认和反馈。

**7.开发阶段**：

按照需求分析和原型设计开始开发。

使用敏捷开发方法，确保及时调整和迭代。

**8.测试与反馈**：

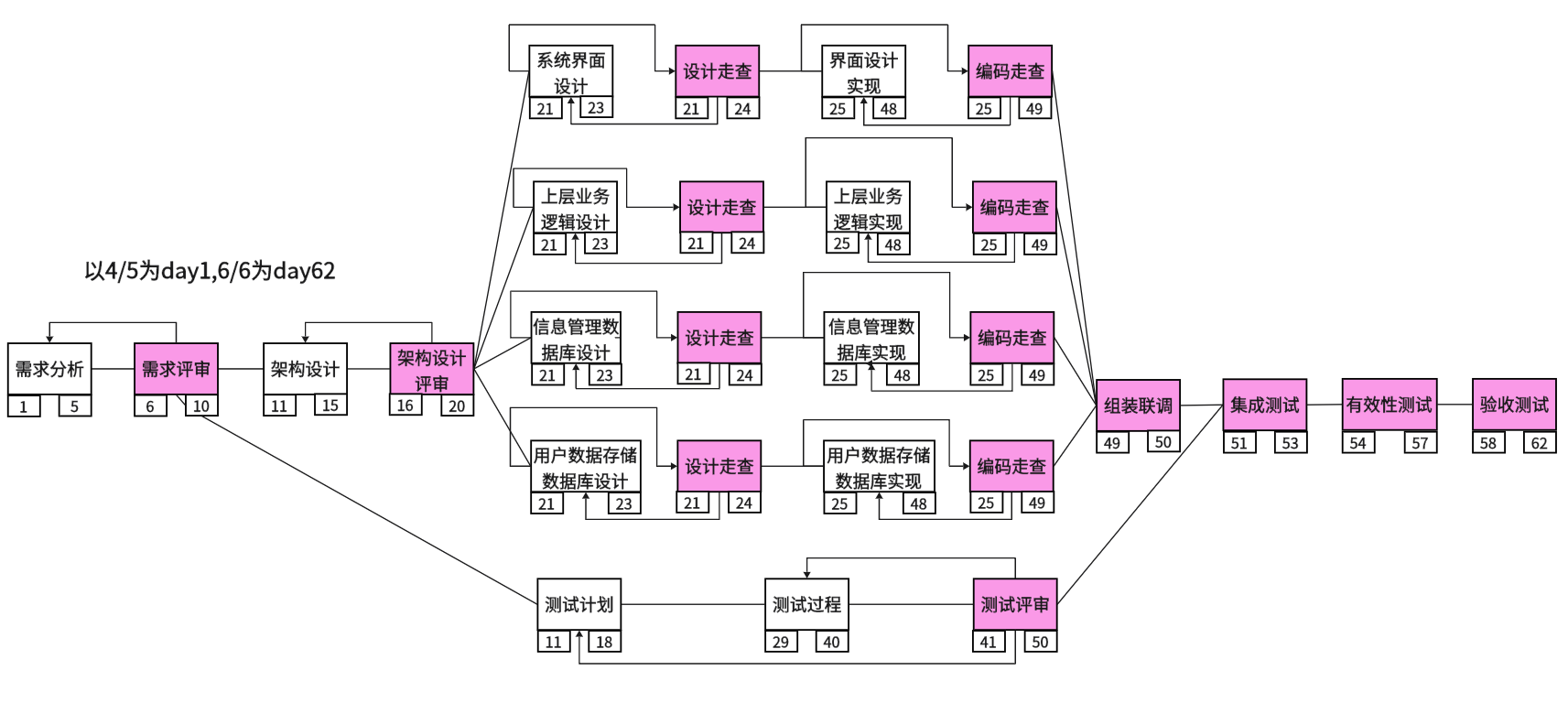
进行单元测试、集成测试和用户验收测试，确保应用质量。

收集用户反馈，不断改进和优化应用。

5.2 工作计划

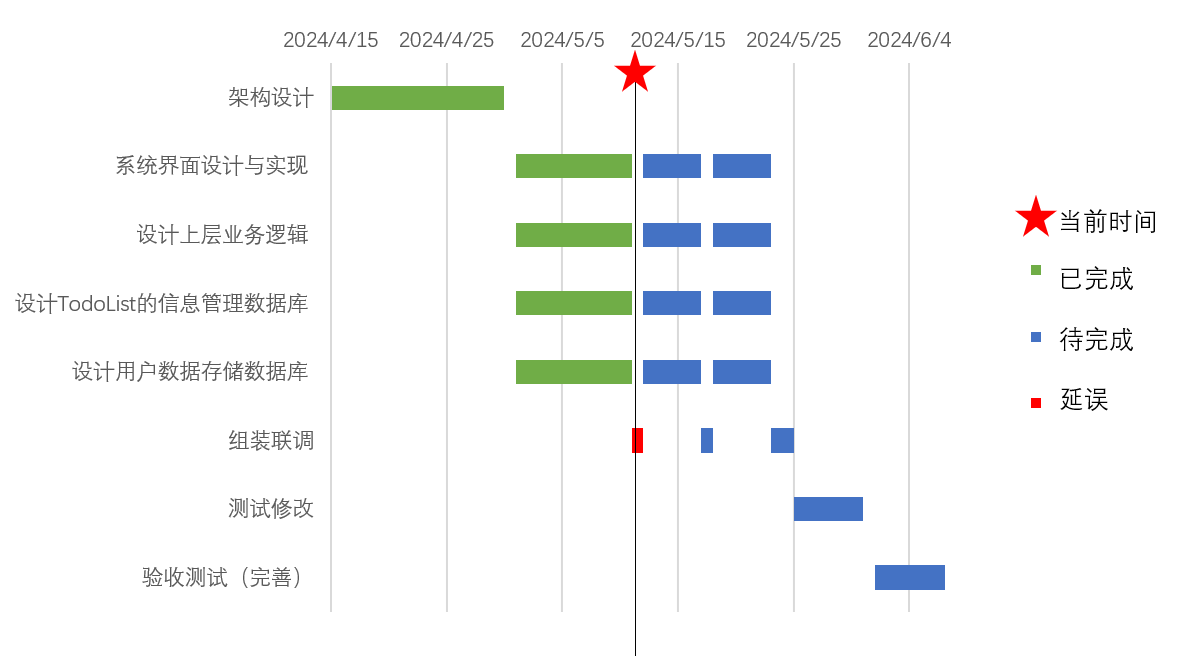
5.2.1 工作活动

本项目的工作计划任务网络图如下：



组员首先对项目需求进行充分分析并评审，然后基于需求进行架构设计和测试设计；架构设计通过评审后，四名组员分别负责系统界面、上层业务逻辑、信息管理数据库、用户数据存储数据库的设计和实现，期间组员之间要对自己和其他同学的模块进行设计走查和编码走查，同时也要根据各个部分的设计进行测试设计和测试评审；各个模块实现过程中要进行多轮组装联调，最终联调和测试评审通过后依次进行集成测试、有效性测试，根据测试的结果和反馈对项目结构进行修改，修改完成后进行验收测试

5.2.2 时间安排

本项目的时间安排甘特图如下：

**项目时间安排甘特图**

项目自4月15日正式开始，首先用5月份之前的15天完成架构设计，定义各模块应实现的功能和模块间的接口。

随后，组员按照责任矩阵的分工分别负责系统界面设计与实现、上层业务逻辑实现、Todolist任务信息管理数据库、用户数据和权限数据库，分模块开发。模块开发分为三个增量阶段，各阶段分别分配10天、5天、5天工时，按照需求优先级列表先后增量开发。每完成一个阶段工作后用1天工时组装联调，最后一个阶段开发完成后有2天的组装联调时间。

5月25日~5月31日，项目在组内测试修改完善。6月1日，小组将软件系统提交给QA小组进行验收测试。6月1日~6月7日，本项目开发小组按照验收小组的验收意见继续修改完善，并在6月7日前完成最后的交付测试。

5.2.3 资源分配

**人力资源分配：**

|  |  |
| --- | --- |
| 李品正 | 规划、执行和监督项目的各个方面 |
| 李品正 | 系统界面设计与实现 |
| 洪志枫 | 设计上层业务逻辑 |
| 贺镇涛 | 设计TodoList的信息管理数据库 |
| 刘浩 | 设计用户数据存储数据库 |

**技术资源分配：**

项目所需的软件资源皆可从武汉大学正版软件平台获得支持。

**设备资源分配：**

团队成员的个人PC设备可作为开发设备，另外需要租赁系统运行服务器。

5.2.4 预算分配

预算计划：

1. 人力成本：0元

2. 系统运行服务器：100元

3. 运行、维护与测试费用：0元

4. 计算机设备费用：0元

5. 其他费用：0元

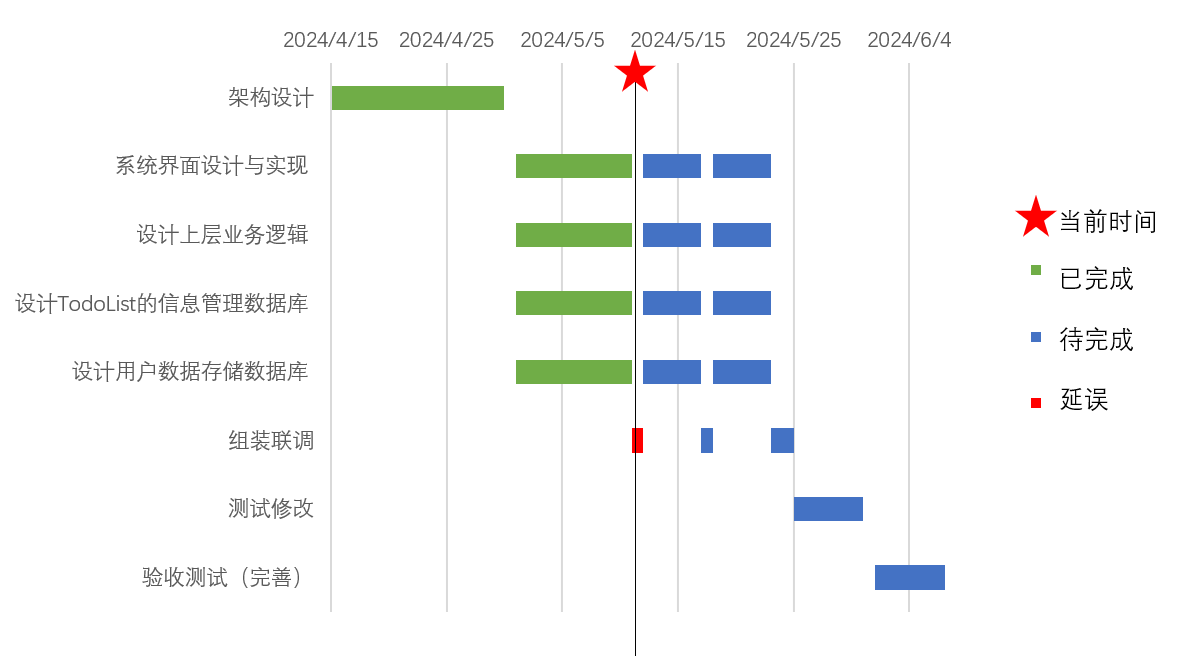
5.3 控制计划

5.3.1 需求控制计划

除了《需求规格说明书》中确定的部份外，项目还考虑：

1. 需求验证：需求是否实现根据测试结果判定，依据是需求规格书、验收标准和测试计划。验证分为组内验证和QA小组验证，分别由开发小组组长监督、开发小组和QA小组组长共同监督；验证结果即使记录成文档（验收结果）。
2. 需求变更控制：需求变更必须通过成文的变更请求表格提交，与课程教师和QA小组沟通、小组全体成员同意后方可变更。变更请求应当妥善保存归档，通知QA小组、老师和助教。
3. 需求跟踪和报告：每周对照需求说明文档，记录各需求项的完成情况，并在项目团队会议上分享和讨论。

5.3.2 时间控制计划



**项目时间安排甘特图**

本项目的时间安排根据上文5.2.2节进行。每完成一轮增量之后，应当全组组装联调、检查完成情况。若相应内容未完成或未达标，则需要考虑调整任务量，如适当削减可选功能、确保在5月25日产生可供测试的软件系统。

**【即：组装联调和每个任务完成的时刻都是检查点】**

在后续测试修改、验收测试的过程中，应当注意版本控制，保留好可供运行的系统版本。如果时间富余，此时也可以继续开发维护。但需要保证最终提交的版本通过了所有测试。

5.3.3 预算控制计划

本项目不适用预算控制计划。

5.3.4 质量控制计划

项目在过程中、过程后都采取质量控制手段。除测试修改阶段外，在开发过程中的每一次组装联调时都进行软件质量检查。质量控制也依照测试验收标准进行。

不满足软件质量标准的模块不允许合并到可发行版本上；需要打回修改到满足质量标准为止。涉及的时间控制参照5.3.2。

5.3.5 报表计划

每周的例会上应共同汇总和讨论项目进展报告，包括已完成工作、进行中工作、计划工作和问题/风险汇报。

报表的重点是需求跟踪、变更控制、突发风险。

5.3.6 度量收集计划

项目需要收集的绩效指标(KPIs)包括需求任务点的完成数目、代码行数、代码验收质量。

项目需要收集的非绩效指标包括：任务点开发耗时、编译用时、测试用时、联调用时。这些数据用于后续的开发决策。

5.4 风险管理计划

风险信息表

|  |  |
| --- | --- |
| **项目名称** | 珞做TodoList |
| **风险类型** | 人员风险 |
| **影响值** | 3 |
| **风险因素** | 小组成员缺乏开发经验与技术 |
| **发生概率** | 30% |
| **影响** | 不能按时完成分配的阶段性开发任务 |
| **监控方法** | 分析任务所需工具技术，成员是否具备工具使用经验与技术，督促成员及时学习缺乏的经验技术 |
| **应急计划** | 其他小组成员帮助完成开发任务以顺利推进项目开发 |
| **估算资源** | 2人 |

|  |  |
| --- | --- |
| **项目名称** | 珞做TodoList |
| **风险类型** | 人员风险 |
| **影响值** | 4 |
| **风险因素** | 小组成员怠工或繁忙，导致计划和进度失效 |
| **发生概率** | 20% |
| **影响** | 不能按时完成分配的阶段性开发任务 |
| **监控方法** | 阶段任务执行时，定期汇报任务进度，并进行评审 |
| **应急计划** | 其他小组成员帮助完成开发任务以顺利推进项目开发 |
| **估算资源** | 3人 |

|  |  |
| --- | --- |
| **项目名称** | 珞做TodoList |
| **风险类型** | 工程规模风险 |
| **影响值** | 3 |
| **风险因素** | 软件项目规模估算可能偏差很大 |
| **发生概率** | 40% |
| **影响** | 如果项目规模被低估，项目的工作量可能会被低估，导致项目进度延误 |
| **监控方法** | 定期对项目的进度和成本进行跟踪和比较，与原始的估算进行对比。如果发现实际进度和成本与最初的估算存在较大偏差，就需要采取修正措施 |
| **应急计划** | 增加资源投入、重新分配任务、调整项目范围或时间表 |
| **估算资源** | 2人 |

|  |  |
| --- | --- |
| **项目名称** | 珞做TodoList |
| **风险类型** | 开发环境风险 |
| **影响值** | 4 |
| **风险因素** | 团队成员之间的开发环境配置不一致 |
| **发生概率** | 20% |
| **影响** | 可能导致代码编译、测试和部署的不一致性，从而增加了集成和部署过程中出现问题的风险 |
| **监控方法** | 定义统一的开发环境配置标准，包括操作系统版本、开发工具版本等，确保所有团队成员都清楚这些配置标准 |
| **应急计划** | 建立备份和还原策略是必要的，以确保在出现问题时可以迅速恢复到稳定的环境 |
| **估算资源** | 4人 |

5.5 项目收尾计划

6月5日前完成项目收尾工作，包括清理和整理项目文档、代码、测试数据等，总结并交付项目成果，评估项目绩效，收集项目开发过程中的经验与教训。

**6 技术过程计划**

6.1 过程模型

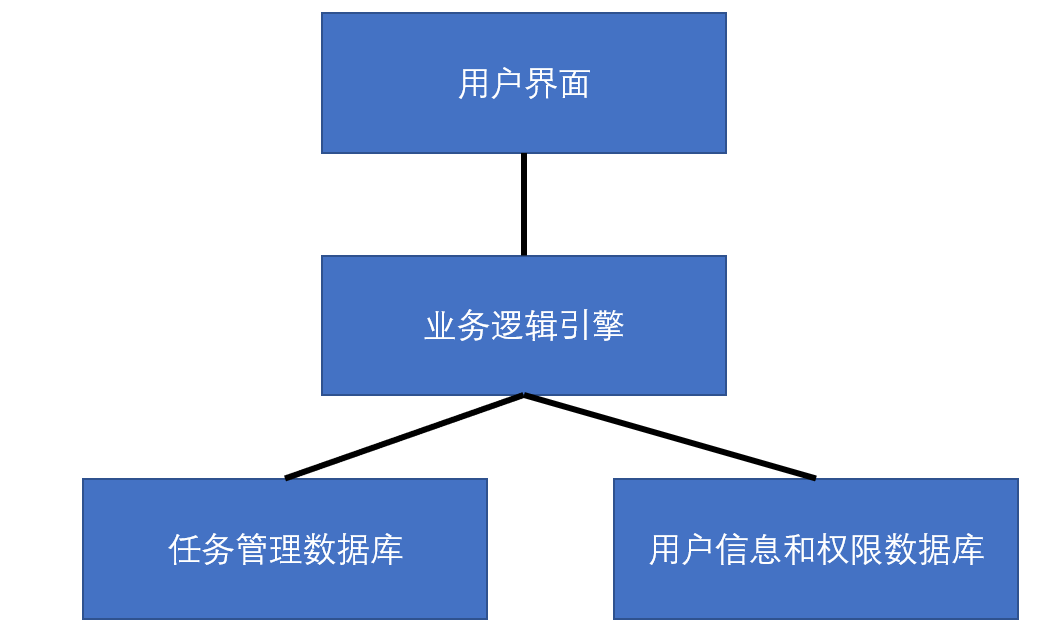
本项目的开发采用增量模型。由于架构设计过程中已经定义好了四个功能模块和每个模块的三个需求层次，因此每次增量开发内容即可，无需重新做需求分析。

6.2 方法、工具和技术

本项目按照模块化的思想分模块实现功能。前端用户界面考虑使用VUE和Electron，后端主要使用Python和SQLite等技术。

6.3 基础结构计划

本项目将产品系统划分为四个模块：(1)前端用户界面；(2)业务逻辑引擎；（3）Todolist任务管理数据库；（4）用户信息和用户权限数据库。



6.4 产品验收计划

通过了组内测试的产品于6月1日提交给QA小组进行验收测试；6月1日至6月7日根据QA小组的意见继续修改完善。

具体产品验收计划按照QA小组提供的验收计划文档执行。

**7 支撑过程计划**

7.1 配置管理计划

7.1.1 配置管理工具

使用 Git 作为版本控制系统，GitHub 作为代码托管平台。

使用项目管理工具（如Trello、Jira等）来追踪任务和变更请求。

7.1.2 配置项标识

对所有文档、程序代码、数据、标准和规约等配置项进行标识，为每个配置项生成一个唯一和直观的标识，对配置项属性进行准确和详细的描述，标识模板如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 项目名 | 配置项类型 | 配置项名称 | 版本号 | 修订号 |

7.1.3 配置控制

提交变更请求时，由组长审查，并根据需要与小组成员讨论。变更经过批准后，由提交请求的小组成员实施，并记录变更内容和版本信息。

7.1.4 配置状态跟踪

使用版本控制系统跟踪配置项的变更历史和版本信息，在代码库中标记每个版本，并在发布时记录版本信息。

7.1.5 配置审核

定期审查代码库和变更历史，确保配置项的完整性和一致性。

7.2 测试计划

7.2.1 引言

测试计划旨在确保 TodoList 项目的软件质量得以保障，通过有效的测试活动来发现和修复潜在的缺陷，以确保项目的成功交付。

7.2.2 测试目标

确保 TodoList 应用程序的功能完整性和稳定性，验证用户界面的易用性和用户体验，发现和修复潜在的性能问题和安全漏洞。

7.2.3 测试策略

进行单元测试、集成测试和系统测试，以覆盖不同的测试层面。使用自动化测试工具和手动测试方法相结合，提高测试效率和覆盖率。针对主要功能和关键路径进行重点测试，以确保其正确性和可靠性。

7.2.4 缺陷跟踪和修复

使用缺陷跟踪系统（如Jira、Bugzilla等）来记录发现的缺陷，并及时修复缺陷，并在修复后重新进行测试和验证。

7.2.5 测试报告和总结

生成测试报告，记录测试活动的结果和发现的问题，总结测试过程中的经验教训和改进建议，以便将其应用于未来的测试活动。

7.3 归档计划

确定需要归档的文档和资料，如需求文档、设计文档、测试文档、用户手册等。

使用统一的命名和存储规范，将归档文件整理并存档到适当的位置，以便日后查阅和使用。

确保备份归档文件，并定期更新备份。

7.4 质量保证计划

确定质量标准和指标，以确保TodoList应用程序符合预期的质量水平。

制定测试策略和计划，包括单元测试、集成测试、系统测试和用户验收测试。

使用质量度量和评估方法，对项目进行定期的质量审查和评估，以及及时采取纠正措施。

7.5 评审和审计计划

进行设计审查和需求审查，以确保项目符合标准和规范。

每周日下午对配置管理、代码质量、安全性等方面进行一次审计。

记录评审和审计的结果，并跟踪和解决发现的问题。

7.6 问题解决计划

确设立问题跟踪系统，用于记录和跟踪项目中的问题和缺陷。

确定问题解决的流程和责任人，包括问题的提交、分配、解决和验证。

确保及时响应和解决问题，以减少对项目进度和质量的不利影响。

7.7 转包商管理计划

无。

7.8 过程提升计划

定期进行过程评估和改进，识别和解决项目中存在的问题和瓶颈。

定期收集团队成员的反馈和建议，为项目的持续改进提供参考。

组织相互学习与指导，以提高团队成员的技能和知识水平，促进团队的成长和发展。

**8 附加计划**

无。

**教师评语评分**

评语：

评分：

评阅人： 年 月 日

（备注：对该实验报告给予优点和不足的评价，并给出百分制评分。）