

**计算机与信息技术学院**

**课程设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
|  | 课程名称 | 信息系统课程设计 |  |
|  | 设计名称 | 停车场管理系统设计与实现 |  |
|  | 专业班级 | 软件工程2101 |  |
|  | 学 号 | 21852001 |  |
|  | 姓 名 | 张俊辉 |  |
|  | 任课教师 | 田晋坤 |  |
|  | 日 期 | 2024-11-01 |  |

**《信息系统课程设计》课程考核说明**

**一、考试对象**

2021级软件工程2101-2103班

**二、考核方式**

课程设计报告 + 答辩。

**三、考试内容**

使用Scrum方法，基于当前的主流开发技术，如：Java、Nodejs 、TypeScript、Vue3框架，Spring Boot框架等，完成某个信息系统的开发，题目自拟，如：毕业论文管理系统、固定资产管理系统、客户关系管理系统、销售管理系统、……：

**四、技术要求**

1、要求使用敏捷开发中的Scrum方法进行项目管理。

2、使用当前主流的开发技术，如：Java、Nodejs、Vue、Spring Boot等。

2、项目文件命名：英文名称，名称与多个英文单词之间用“-”连接，如李四的手机销售网站命名为：lm-phone-seller

3、单文件组件命名要符合Pascal命名规范，变量、文件及文件夹命名要符合规范，如路由文件夹用routes、视图文件夹用views，尽量做到见名知意。

**五、提交物**

项目源码、设计说明书电子版、设计说明书纸质打印版、答辩纪录。

**六、期末成绩构成**

由项目成绩和答辩成绩组成，其中项目成绩60分，答辩成绩40分

项目成绩评分标准如下：

|  |  |
| --- | --- |
| **综合/单项类评分** | **分值** |
| **1 提交物** | **10** |
| 1.1 内容完整性 | 5 |
| 1.2 目录结构 | 5 |
| **2 代码** | **20** |
| 2.1 项目、代码命名规范 | 5 |
| 2.2 代码编写规范性 | 5 |
| 2.3 代码可读性 | 5 |
| 2.4 技术全面 | 5 |
| **3 程序演示** | **30** |
| 3.1 结果正确，功能完整 | 15 |
| 3.2 页面设计美观 | 15 |

**《信息系统课程设计》课程考核成绩**

|  |  |
| --- | --- |
| 项目成绩（60分） |  |
| 答辩成绩（40分） |  |
| 期末总成绩 |  |
| **答辩评语：**  提交物内容（完整、较完整、基本完整、不完整），目录结构（正确、较正确、基本正确、不正确）；项目、代码命名（符合规范、较符合规范、基本符合规范、不规范），代码注释（完整、较完整、基本完整、不完整），可读性（好、较好、尚可、差）；技术应用方面（全面、较全面、基本合格、有欠缺）；程序功能（完整、较完整、基本完整、不完整），页面（美观、较美观、设计尚可、不美观），演示过程（顺畅、较顺畅、基本顺畅、不顺畅），问题回答（正确、较正确、基本正确、错误较多），思路（清晰、较清晰、基本清晰、不清晰），概念（清楚、较清楚、基本清楚、不清楚）。 | |

**目 录**

[第1章 产品需求 1](#_Toc180768412)

[1.1 背景和目的 1](#_Toc180768413)

[第2章 敏捷开发流程 3](#_Toc180768414)

[2.1 创建产品待办事项（Product Backlog） 3](#_Toc180768415)

[2.2 计划会议(Sprint Planning) 12](#_Toc180768416)

[2.3 第一次Sprint 14](#_Toc180768417)

[2.4 冲刺评审会议(Sprint Review) 23](#_Toc180768418)

[2.5 版本发布(Release) 26](#_Toc180768419)

[2.6 冲刺回顾会议(Sprint Retrospective) 26](#_Toc180768420)

[第3章 总结 29](#_Toc180768421)

[附件：答辩纪录 31](#_Toc180768422)

第1章 产品需求

## 1.1 背景和目的

### 1.1.1 产品背景

1. 城市化进程加速

随着全球化和城市化进程的不停加快，越来越多的人口涌入城市，城市人口密度不断增加。资源有限的城市环境带来了对停车资源的强烈需求，尤其是商业区和居民区的停车设施相对不足。在新兴城市中，汽车成为主要的交通工具，停车需求上升，原有的停车管理方式已经无法满足现代城市的需求。

2. 汽车保有量持续增加

汽车生产技术的不断进步和市场需求增长，推动了汽车生产及销量的持续上升。随着经济水平提升，家庭拥有多辆车的情况日益普遍，造成对个人和公共停车位的需求增加。政府对于汽车消费的鼓励政策，进一步刺激了汽车购买的热潮。相比公共交通，私家车在便利性和舒适性上的优势使更多人倾向选择驾车出行。

3. 交通拥堵与停车困难

鉴于汽车数量的急剧增多，城市道路的交通压力随之上升，频繁的交通堵塞使得在商业区和繁忙地区寻找车位变得更加困难。交通拥堵不仅影响出行效率，还造成了空气污染和资源浪费等环境问题。尤其在商业中心、医疗机构、学校等高人流区域，停车位供给不足，导致租车、购买、租赁停车服务的需求剧增。无序停车现象严重，进而造成公共交通停滞、事故频发以及资源占用。

4. 市场需求多样化

随着用户的消费意识提升，越来越多的用户渴望个性化、智能化的停车服务，如预约停车、自动缴费等。用户期望停车过程简单、快捷，能够实时获取车位信息，缩短寻找停车位的时间。商业运营者希望借助停车场管理系统提高停车场的利用率，避免资源浪费，同时通过数据分析来提升运营效率和收益。需要通过对车位流转率及用户行为的分析，优化停车场布局和服务，提升客户满意度。

### 1.1.2 产品目标

停车场管理系统的设计目标是为停车场管理过程提供一个全面的、数字化的解决方案，简化并优化从课题申报到最终论文提交和答辩的各个环节，提升管理效率，降低操作风险，并确保论文质量。具体目标如下：

1. 车辆与停车位管理

实现自动记录车牌、进场和离场的时间。使用高速摄像头或传感器，支持各种车型的识别，包括摩托车和小型货车。存储车辆入场、出场记录，包括车主信息、停车时长等。提供查询页面，支持通过车牌、时间、车主姓名等多种条件进行搜索。在每个车位部署传感器或摄像头，实时反馈每个车位的状态（空闲/占用）。通过移动端或网站提供实时车位信息，您可以轻松查询。在入口处和停车场内设置LED显示屏，指示空闲车位数量与位置。移动应用提供导航功能，用户可接收基于实时数据的最佳停车路线建议。

2. 计费系统与支付功能

支持夜间/节假日特殊计费规则，亦可针对特定用户（如常客）提供折扣。用户可以随时查询历史停车记录及费用明细。管理者可导出报表，查看不同时间段的收入和费用情况。支持各类支付方式，包括信用卡、支付宝、微信、现金等。移动应用可以实现扫码支付，简化用户流程。用户支付后可自动生成电子发票，供用户下载或发送至邮箱。系统保存支付记录，以便用户查询和管理。

3. 提高信息透明度与可控性

系统将确保信息透明化，所有提交、反馈、审核记录都可以通过系统进行管理和查询。通过明确的流程控制和权限设置，系统保证了每个用户角色在合适的时间看到合适的信息，减少了不必要的沟通障碍和信息丢失的风险。对于教务管理者来说，他们可以通过系统方便地汇总所有车辆的信息，并根据需要进行数据分析和问题预警。

4. 预约停车与用户管理

为新用户提供注册功能，包括邮箱、手机号码等信息收集。用户可修改个人信息，更改密码等功能。系统根据不同角色（如管理员、普通用户）设置不同的访问权限，确保数据安全。用户可以选择需要预约的日期、时间和停车时段。支持选择车位类型（如普通、VIP、长租等）并显示对应费用。用户提交预约后，系统立即发送确认短信或电子邮件。提前提醒用户即将到达的预约时间，并提供到达路线。提供多种数据报表，涵盖停车使用率、收入统计、车主行为分析等。数据可视化功能，通过图表展示，为决策提供依据。管理者可设置定期生成数据报告，如每周营业额、用户增长情况等，自动发送至指定邮箱。

# 第2章 敏捷开发流程

## 2.1 创建产品待办事项（Product Backlog）

### 2.1.1 需求收集

在开发毕业论文管理系统的过程中，需求收集是项目初期的关键步骤之一。有效的需求收集确保系统设计符合所有相关方的期望，并能解决实际业务中存在的痛点。该系统的需求收集主要从以下几个关键用户群体进行，包括终端用户、管理人员、外包服务商以及技术。

#### 1）收集到的需求

1. 终端用户需求

私家车车主、商务人士是停车场管理系统的主要用户之一。需求收集的过程中，通过社会调查，确定他们在停车场管理中的主要需求：

便捷的车位预订：希望能通过手机或网页快速找到并预订车位。

简约的支付流程：在线支付或非接触支付功能，提高交易效率。

实时信息：渴望及时获取车位状态、停车费用及距离等信息。

高品质服务：期望有高标准的停车场（如安全、整洁、便捷的位置）以维持职业形象。

费用透明：希望在预约时能清楚了解费用信息，避免后续争议。

导航与信息整合：希望停车信息能结合导航系统，提供顺畅的到达体验。

灵活停靠时间：需求适应性强，停车时间多样。

2. 管理人员需求

监控与管理系统：需求实时监控停车场的使用情况，掌握停车位的占用率和流动情况。

用户反馈管理：希望有系统自动汇总用户反馈和投诉，方便后续处理。

数据分析与优化：需要利用数据分析工具优化停车资源配置，降低运营成本。

应急处理机制：希望在出现突发事件时，系统能够提供快速的信息传递和处理流程。

3. 外包服务商需求

系统集成需求：需要将停车管理系统与其他技术（如车牌识别、支付系统等）进行无缝集成。

数据分享与安全：希望确保与停车场管理者的数据共享和安全性，满足各方需求。

技术支持与培训：需要提供对停车场工作人员的培训，确保系统的顺利运行。

4. 技术需求

除了用户需求，技术需求也是需求收集过程中需要考虑的一部分：

系统可扩展性：系统需要能够处理多个用户的并发访问，并能够根据学生和导师的增加灵活扩展。

数据安全与隐私保护：系统必须遵循严格的安全标准，确保学生的论文数据、导师的评审意见以及所有学术相关的资料不会被未经授权的人访问或泄露。

跨平台兼容：系统需要能够兼容多种设备，包括桌面端和移动端，以便用户能够随时随地访问系统。

2）使用GitHub管理需求

### 2.1.2 产品设计

#### 1）获得产品功能

基于需求收集和用户反馈，停车场管理系统的产品功能设计将涵盖从用户停车到最终驶离缴费的整个流程，确保不同角色（终端用户、管理人员、外包服务商、技术人员）的功能需求得到满足。以下是各个功能模块的设计及其功能描述。

1. 登录功能

功能描述：

管理人员、外包服务商、技术人员等用户通过输入凭证（如用户名和密码）来访问系统包含输入框供用户输入用户名和密码。提供“登录”按钮以提交凭证。可选链接：忘记密码、注册新账户等。

用户角色：管理人员、外包服务商、技术人员

关键功能点：

对输入的密码进行格式校验（如长度、特殊字符等）

2. 控制台数据可视化信息显示

功能描述：

在控制台中显示可视化信息，展示合作单位数量统计、停车场数量、车辆数量、收益总额等，下方可以使用Echats图表库展示最近七日支付订单及订单支付类型。

用户角色：管理人员、外包服务商、技术人员

关键功能点：

收入统计分析

3. 显示车辆停车信息

功能描述：

实现停车场对应的信息与合作企业的管理，包括基本的停车场名称、免费时

长，单元费用、最大收费金额、计时单元等；可以实现停车场的新增、修改、搜

索和删除等功能。

用户角色：管理人员

关键功能点：

信息的显示

4. 车辆管理功能描述：

实现车辆对应信息的管理，可以选择合作单位/停车场进行搜索。包括序号、

所属单位、停车场、车牌号、车主姓名、性别、状态、类别、创建时间等；可以

实现对车辆信息的新增、修改、搜索和删除等功能。

用户角色：管理人员、终端用户

关键功能点：

将车辆信息上传到数据库

5. 用户管理功能

功能描述：

实现系统各角色信息和权限的管理。包括序号、角色标识、角色名称、备注、

创建时间等。可以实现角色的新增、修改和删除功能，还可以通过“菜单权限”

更改各角色的权限。

用户角色：终端用户、管理人员

关键功能点：

查看系统的各项功能用户

6. 订单列表功能

功能描述：

可以显示各条停车记录的订单信息。包括序号、订单号、付款金额、车牌号、

支付类型、有效期、备注等。可以对这些信息进行搜索，还可使用“导出数据”

功能将数据导出成表格。

用户角色：管理人员

关键功能点：

数据可以导出

将数据库数据转换成表格

1. 合作单位

功能描述：

这个页面可以用来展示使用该系统的停车场的信息。包括序号、单位代码、

单位名称、单位全称、创建时间等。有增加、删除、修改等操作

用户角色：外包服务商、技术人员

关键功能点：

进行合作单位的显示

8. 系统日志功能

功能描述：

这个页面用于记录系统的日志，主要是用户的登录日志。可以记录序号、用

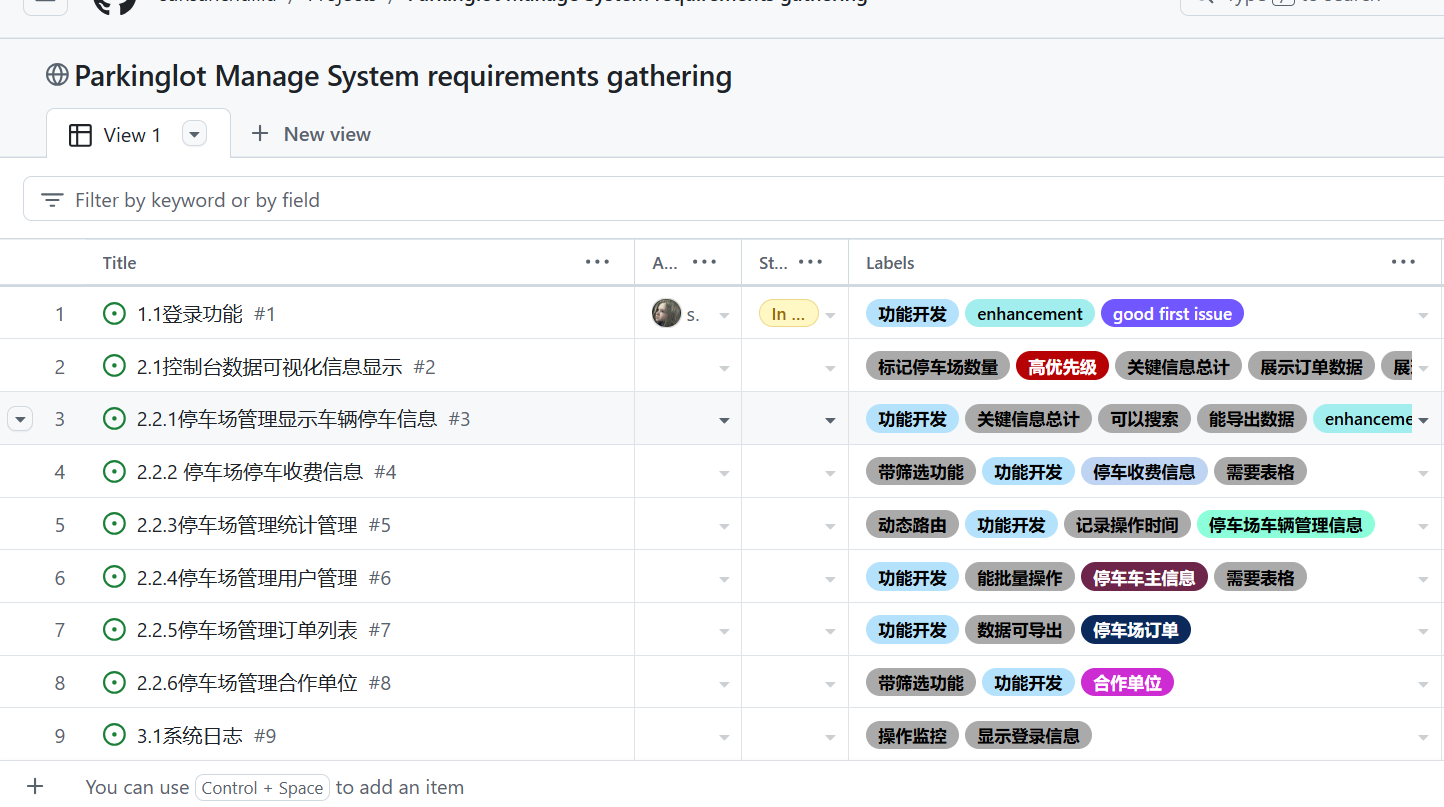
户名、用户操作、系统响应时间和操作时间。能够对这些信息进行搜索操作。

用户角色：系统管理员

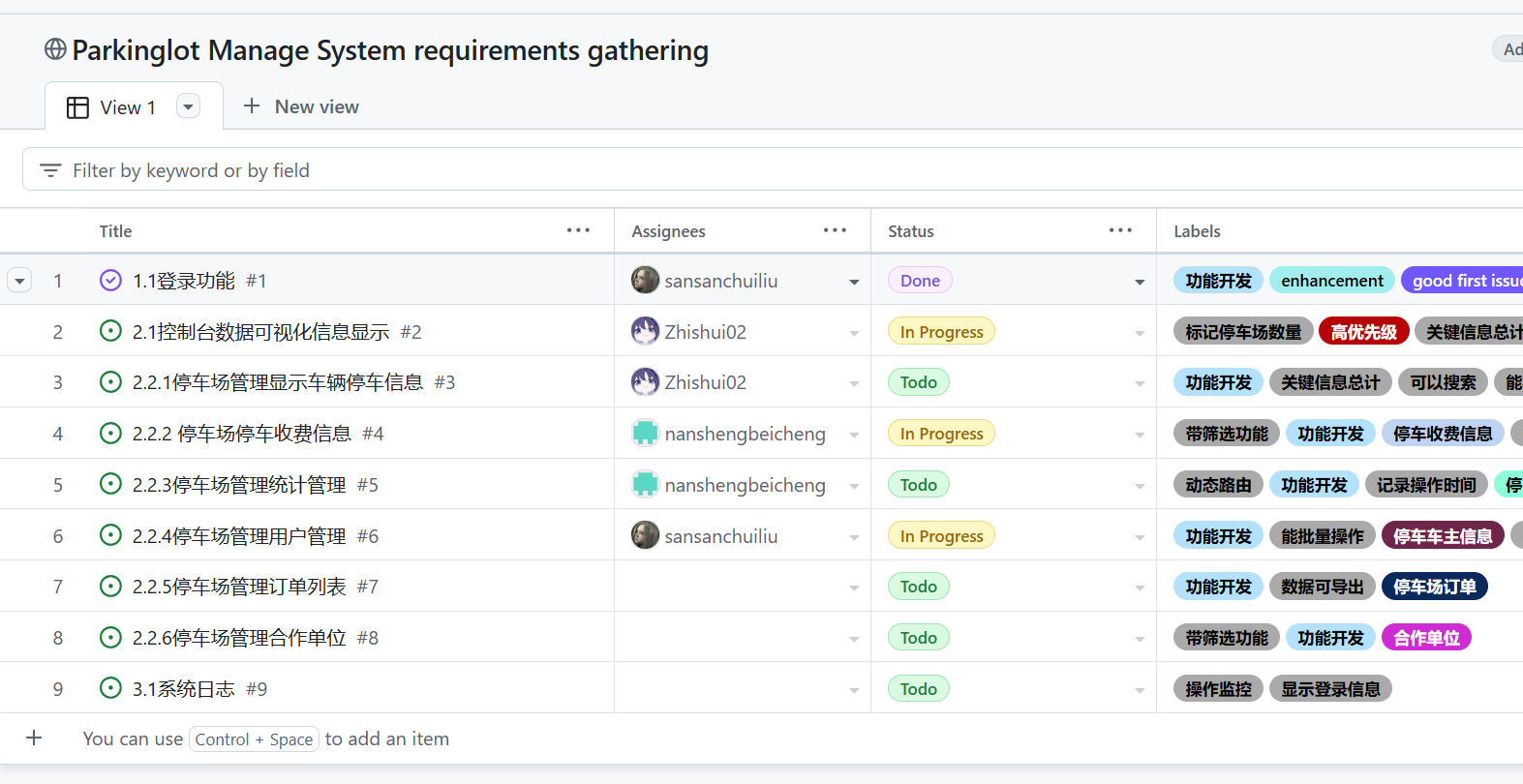
关键功能点：

在每次操作结束之后及时更新

#### 2）使用GitHub管理产品功能



### 2.1.3 优先级划分



#### 高优先级（P1 - 关键功能）

这些功能是系统的核心功能，直接影响到停车场管理的工作流程，是系统上线时必须具备的功能。

1. 控制台数据可视化信息显示功能

优先级：P1

原因：这是整个停车场系统管理流程的终端信息显示，在这里可以查看各个模块的信息，方便管理人员对不同功能模块信息的管理。

2. 停车场管理统计管理功能

优先级：P1

原因：作为停车场管理最关键的功能之一，该功能是为了帮助管理人员更好地了解停车场的运营情况，做出相应的管理决策。

3.停车场管理显示车辆停车信息功能

优先级：P1

原因：停车场管理系统中的“车辆停车信息显示功能”是一个核心功能模块，旨在为用户和管理人员提供关于车辆停车状态的实时和历史信息。

#### 中优先级（P2 - 重要但可延后）

这些功能也是重要的，但在系统上线初期可以稍微延后，或可以在完成高优先级功能后逐步实现。

4. 停车场管理用户管理功能

优先级：P2

原因：停车场管理系统中的用户管理功能是确保停车场运营和服务质量的重要组成部分，在后续开发中完善该功能并不影响核心功能的运行。

5. 停车场停车收费信息功能

优先级：P2

原因：该功能旨为计算并显示停车车辆的收费信息，但不影响主要流程的运行，因此可以在后续开发中添加。

6. 停车场管理合作单位功能

优先级：P2

原因：用以促进停车资源的共享、提高管理效率以及增值服务的提供，作为后续扩展功能，并不需要优先开发。

#### 低优先级（P3 - 增强功能）

这些功能虽然有助于提高系统的整体体验和管理效率，但可以作为后续的系统优化内容，不影响系统的核心运作。

7. 登录、注册功能

优先级：P3

原因：登录、注册功能并不参与停车场管理系统核心功能流程开发，因此在系统基本开发完成时，在添加该功能模块即可。

8. 停车场管理订单列表功能

优先级：P3

原因：该功能可以帮助用户方便地查看和管理自己的停车订单，同时也为管理员提供清晰的订单处理与分析工具，作为扩展功能后续再通过灵活的权限配置功能进行优化。

9. 系统日志

优先级：P3

原因：通过系统日志，管理员可以追踪系统的运行状态、用户活动以及安全事件，确保系统的透明性、安全性和可维护性。在系统全部功能开发完成后，再继续优化完善。

### 2.1.4 提出冲刺里程碑草案

基于当前的系统开发进度和产品目标，系统开发将分为 5 个 Sprint，每个 Sprint 有明确的开发目标和功能交付标准。以下是针对每个 Sprint 的里程碑草案。

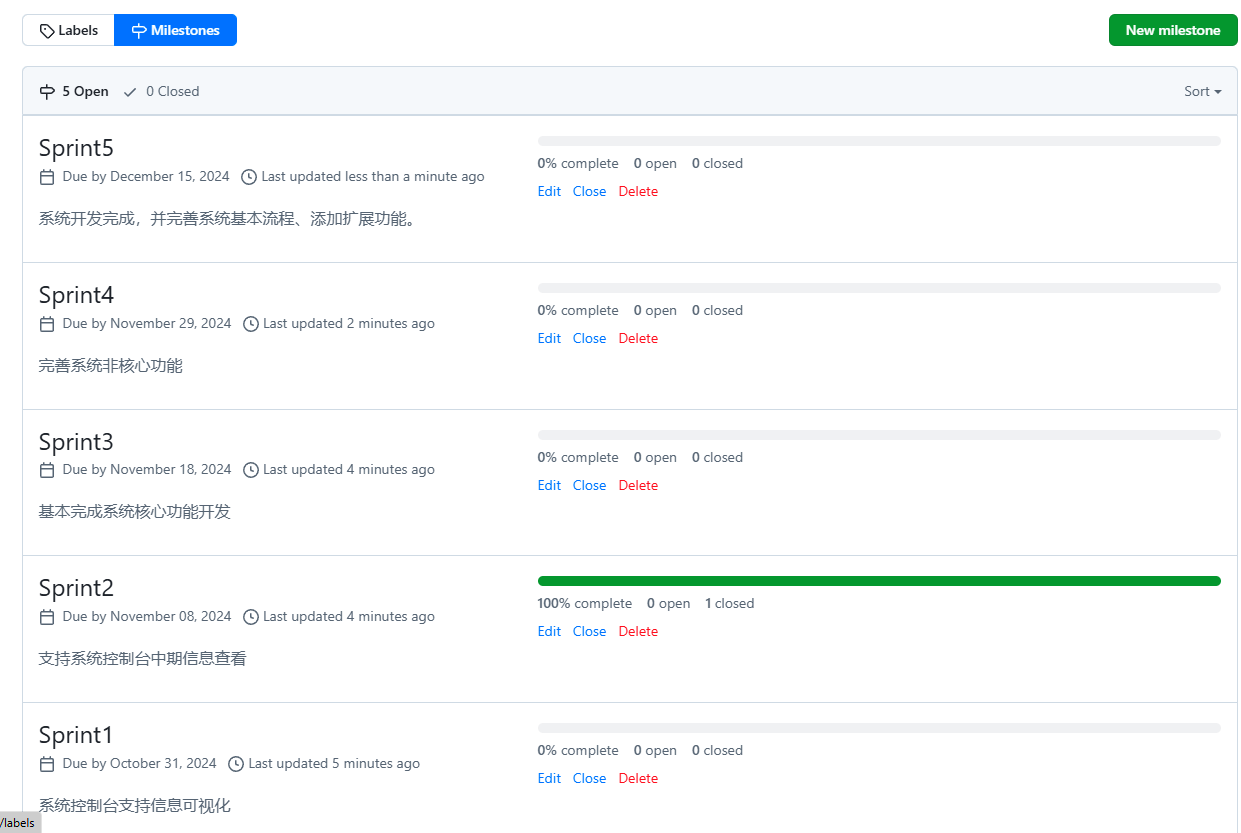
Sprint 1：重点实现控制台可视化功能，确保可以清晰了解并管理停车场相关管理信息，预计完成日期为2024年10月31日。

Sprint 2：实现控制台可视化功能的支持，包括停车场管理显示车辆停车信息停车场管理统计管理等**，**目标完成日期为2024年11月08日。

Sprint 3：实现中优先级功能，目标完成日期为2024年11月18日。

Sprint 4：完善订单展示、系统日志功能，确保系统能够管理和展示停车场相关信息，目标完成日期为2024年11月29日。

Sprint 5：扩展控制台信息可视化展示功能，最终完善系统，目标完成日期为2024年12月15日。



## 2.2 计划会议(Sprint Planning)

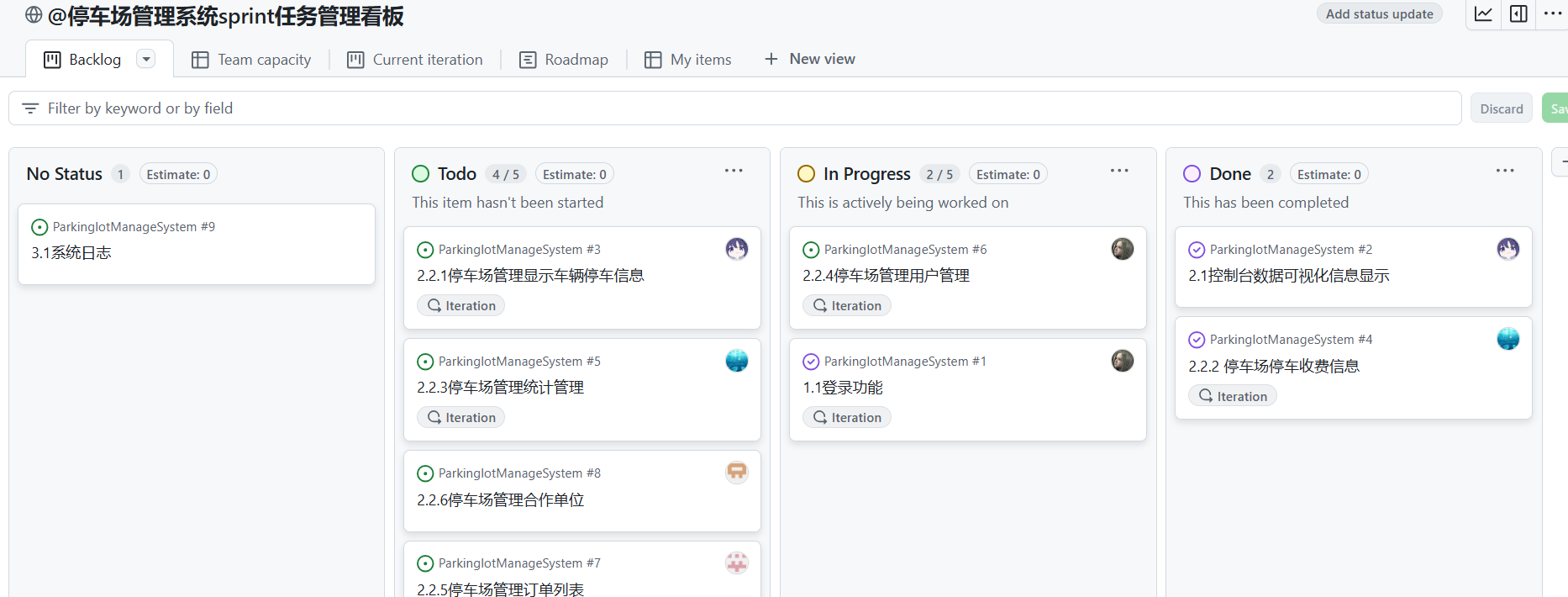
### 2.2.1 会议纪要

|  |
| --- |
| 项目名称：停车场管理系统 会议日期：2024年10月29日 参会人员： 产品负责人：张羽辰，熊志坤 开发团队：张俊辉，张治华，费亮 敏捷教练：田晋坤 会议主题：第一次 Sprint 计划会议 - 实现控制台功能 1.会议目标： 确定本Sprint的开发目标和用户故事，评估上个Sprint的完成情况及反馈 讨论当前开发中的挑战和风险，制定团队的工作计划和分配任务 2. 工作范围 产品负责人张羽辰、熊志坤介绍了本次 Sprint 的工作范围，选取了产品待办事项中的开题功能，并详细描述了以下需求： 日志查询（停车次数、收入等） 数据可视化（图表展示） 报表导出功能（如Excel、PDF） 该功能是停车场管理系统的必经步骤，因此本次 Sprint 将集中力量完成这一功能。 3.任务分解 开发数据统计接口 实现数据可视化功能（使用图表库） 实现系统设置界面（如收费标准、开放时间等） 开发停车位状态实时监控（例如通过传感器或手动更新） |

### 2.2.2 会议纪要的管理



### 2.2.3 建立Sprint任务管理看板

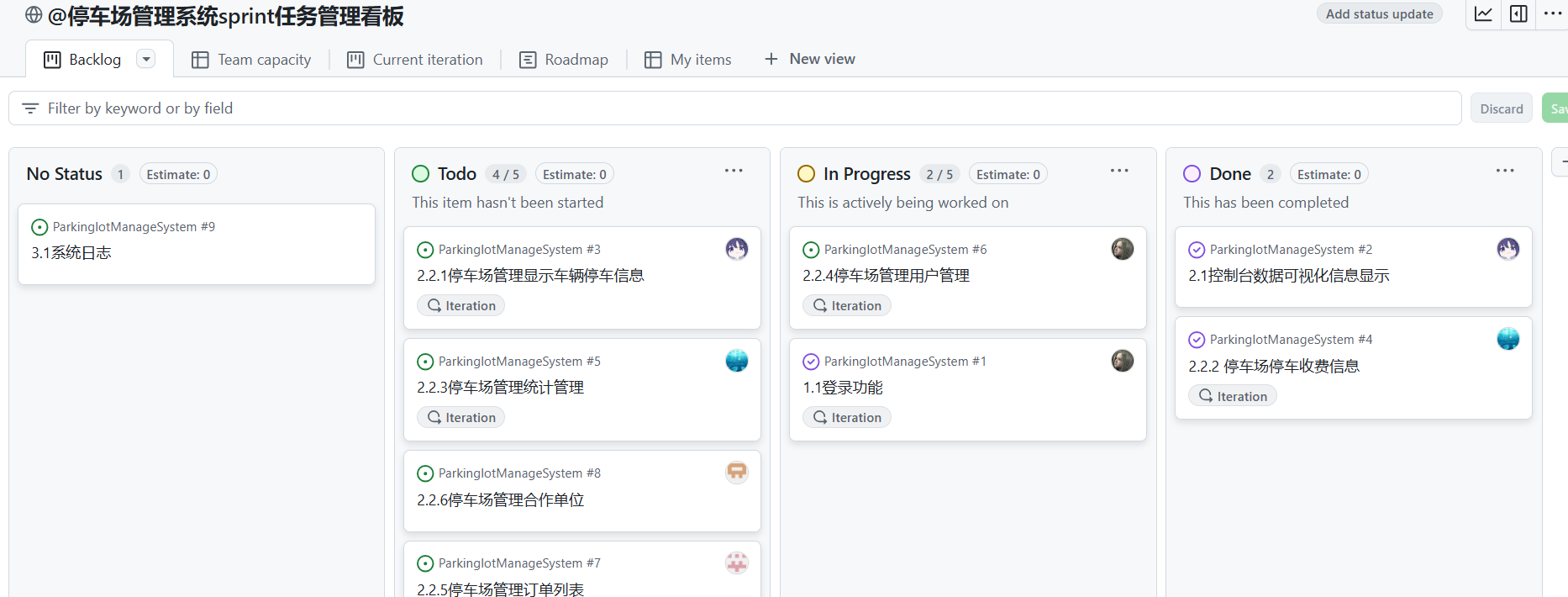


## 2.3 第一次Sprint

### 2.3.1 站立会会议纪要

|  |
| --- |
| 1. 昨天的进展：  昨天我完成了停车场管理系统中停车位信息提交页面的设计工作，页面包括了基本的表单字段（车牌号码、停车时段、车主姓名、联系方式、附件上传等），并且我已经确保表单的布局在各个设备上都能正常显示。同时，我还为上传车辆照片的功能添加了基础的 UI 元素，文件类型的限制也已经实现。  2. 今天的计划：  今天我会继续推进停车场管理系统的功能开发，确保用户在提交停车信息时，所有必填项得到正确的验证，防止提交空白或错误的信息。此外，我还计划实现车牌信息上传时的大小和格式验证功能，比如限制图像格式为 JPG 或 PNG，文件大小不能超过 5MB。今天我会尽量完成这些信息的前端验证工作。  3. 遇到的问题：  目前遇到的一个小问题是，在实现车牌信息上传功能时，上传的图像无法正确显示预览。这可能是因为前端对上传的文件格式处理不当。今天我会先在本地调试这个问题，如果需要的话，我会和张治华（后端开发）对接，看看接口返回的数据格式是否存在问题。 |

### 2.3.2 任务状态跟踪



### 2.3.3 一次任务处理的流程

#### 1）认领 Issue

在 Sprint 计划会议后，认领了 GitHub 上的一个 Issue，任务是停车场停车收费信息的验证。具体流程如下：

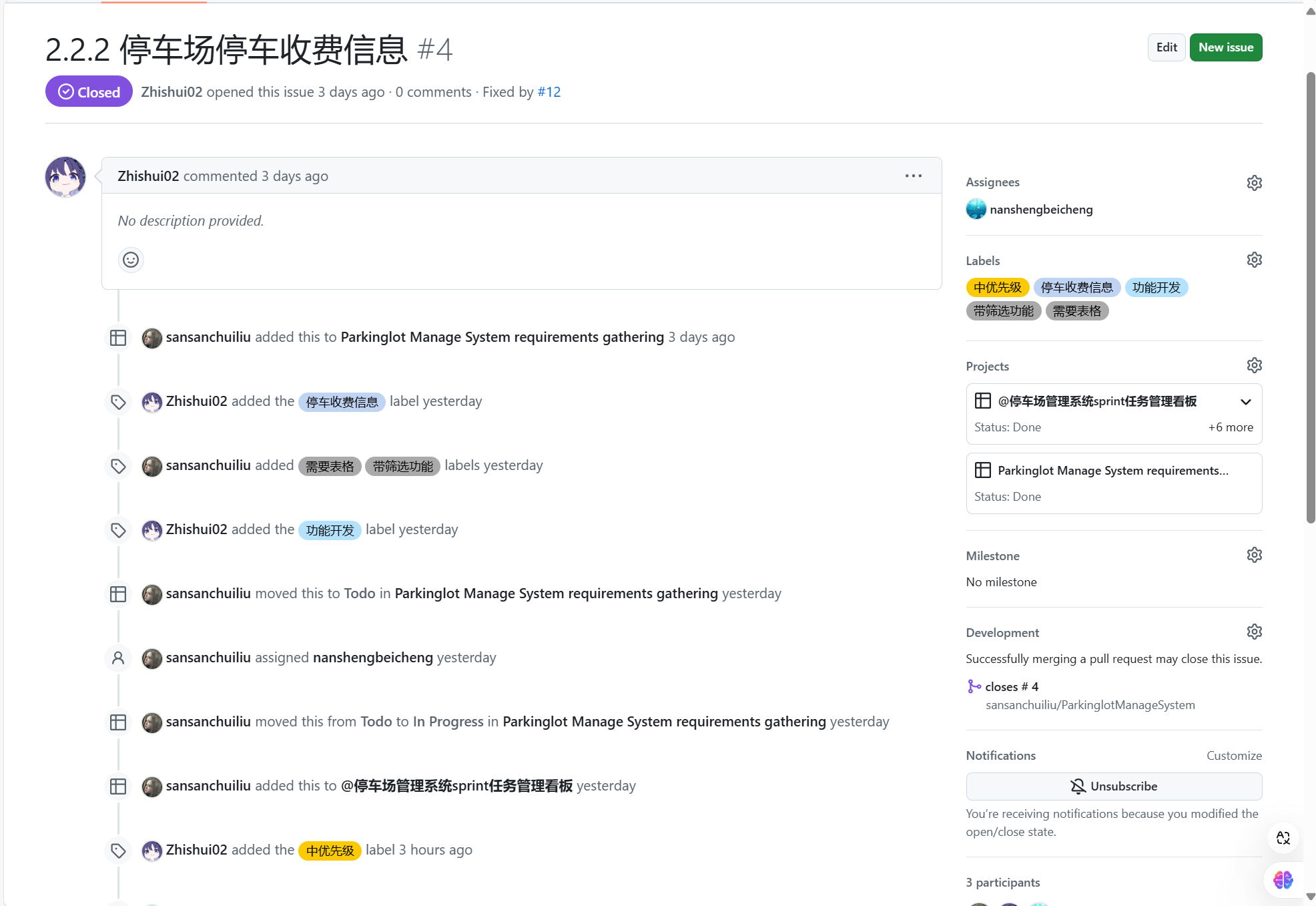
1. 在 GitHub Issues 页面查看任务：

在 GitHub 项目的 "Issues" 页面，找到了相关的 Issue，标题为“停车场停车收费信息”（Issue #4）。

在 Issue 页面，详细描述了任务的内容，包括需要验证的字段（如课题名称、研究背景、研究目的等），以及附件的格式和大小限制。

2. 将 Issue 分配给自己：

点击 "Assign yourself" 按钮，将该 Issue 分配给自己。此时任务已经显示为“张俊辉”负责，其他团队成员可以看到他已认领该任务。



#### 2）创建分支

开始为任务创建一个新的 Git 分支，以便在分支上进行独立的开发工作。

1. 进入本地开发环境：

通过 Git 在本地克隆的项目目录中，执行命令，拉取最新的代码：

|  |
| --- |
| git pull origin main |

2. 创建新的功能分支：

他根据项目中的命名规范创建了一个新的分支，将 Issue 编号作为分支名的一部分：

|  |
| --- |
| git checkout -b feature/issue-4-form-validation |

3. 与 GitHub Issue 关联：

虽然分支的命名不能直接关联 Issue，可以在后续提交代码时，通过提交信息关联 Issue。

#### 3）编写代码和测试用例

1. 编写停车收费逻辑

|  |
| --- |
| //导入datetime模块，用于处理日期和时间  import datetime #  // 初始化Vehicle类，接受车牌号作为参数  class Vehicle:  def \_\_init\_\_(self, plate\_number):  self.plate\_number = plate\_number  self.entry\_time = None // 入场时间初始化为None  // 初始化ParkingLot类  class ParkingLot:  def \_\_init\_\_(self):  self.vehicles = {} // 用于存储在停车场中的车辆，键为车牌号，值为Vehicle实例  self.rates\_per\_hour = 5 // 每小时收费5元  def enter\_vehicle(self, plate\_number):  //创建新的Vehicle实例  vehicle = Vehicle(plate\_number)  vehicle.entry\_time = datetime.datetime.now() // 记录当前时间为入场时间  self.vehicles[plate\_number] = vehicle // 将车辆加入停车场  print(f"车辆 {plate\_number} 入场，入场时间: {vehicle.entry\_time}") // 输出入场信息  // 车辆出场的方法  def exit\_vehicle(self, plate\_number):  if plate\_number in self.vehicles:  vehicle = self.vehicles.pop(plate\_number) // 从停车场中移除该车辆  exit\_time = datetime.datetime.now() // 记录当前时间为出场时间  duration = (exit\_time - vehicle.entry\_time).seconds / 3600 # 计算停车时长（小时）  cost = self.calculate\_cost(duration) // 根据时长计算停车费用  print(f"车辆 {plate\_number} 出场，出场时间: {exit\_time}，停车费用: {cost}元") // 输出出场信息及费用  else:  print(f"车辆 {plate\_number} 不在停车场中。") // 如果车辆不在停车场，输出提示信息  //根据停车时间计算费用的方法  def calculate\_cost(self, hours):  return max(0, hours) \* self.rates\_per\_hour // 确保费用不为负  def current\_vehicle\_count(self): |

2. 编写测试用例

|  |
| --- |
| import org.junit.jupiter.api.Test;  // 导入静态方法用于断言  import static org.junit.jupiter.api.Assertions.assertEquals;  // 定义ParkingLotTest类用于测试ParkingLot类的功能  public class ParkingLotTest {  // 测试1小时内停车的费用计算  @Test  public void testCalculateFeeForFreeParking() {  ParkingLot parkingLot = new ParkingLot(); // 创建ParkingLot类的实例  double fee = parkingLot.calculateFee(1); // 计算1小时停车的费用  assertEquals(0.0, fee, "1小时内停车应免费"); // 断言费用应为0.0  }  // 测试2小时停车的费用计算  @Test  public void testCalculateFeeForTwoHours() {  ParkingLot parkingLot = new ParkingLot(); // 创建ParkingLot类的实例  double fee = parkingLot.calculateFee(2); // 计算2小时停车的费用  assertEquals(5.0, fee, "2小时停车应收费5.0"); // 断言费用应为5.0  }  // 测试3小时停车的费用计算  @Test  public void testCalculateFeeForThreeHours() {  ParkingLot parkingLot = new ParkingLot(); // 创建ParkingLot类的实例  double fee = parkingLot.calculateFee(3); // 计算3小时停车的费用  assertEquals(10.0, fee, "3小时停车应收费10.0"); // 断言费用应为10.0  }  // 测试未停车的情况  @Test  public void testCalculateFeeForNoParking() {  ParkingLot parkingLot = new ParkingLot(); // 创建ParkingLot类的实例  double fee = parkingLot.calculateFee(0); // 计算未停车的费用  assertEquals(0.0, fee, "未停车应免费"); // 断言费用应为0.0  }  } |

3. 本地测试

|  |
| --- |
| npm run test -- --watchAll |

#### 4）提交代码

代码开发和测试完成后，准备将代码提交到 GitHub 上的远程分支。

1. 查看本地修改：

首先检查本地的改动：

|  |
| --- |
| git status |

2. 添加修改到暂存区：

他将修改后的文件添加到暂存区：

|  |
| --- |
| git add . |

3. 提交代码并关联 Issue：

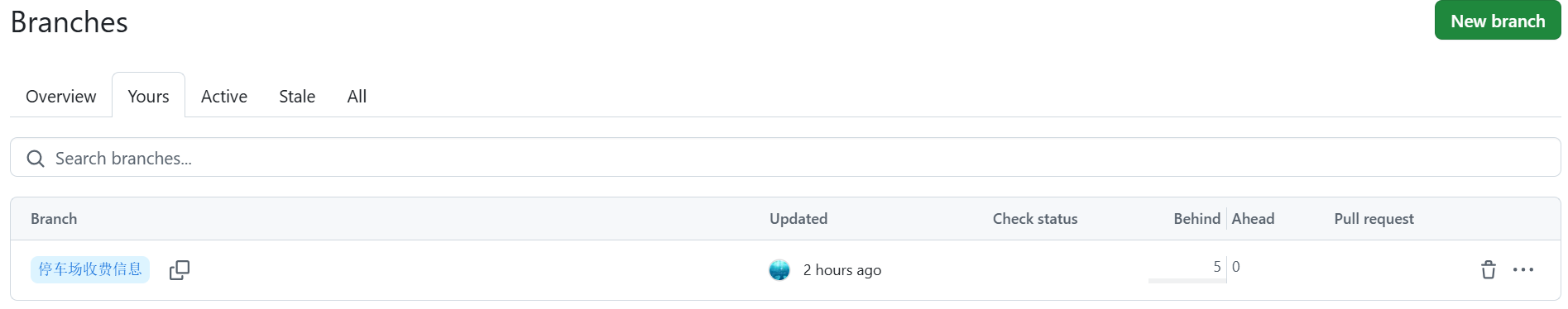
提交代码时，在提交信息中引用了 Issue 编号，以便 GitHub 自动关联此提交与 Issue：

|  |
| --- |
| git commit -m "Implemented form validation for opening report submission. Closes #4" |

4. 推送代码到远程分支：

将代码推送到远程分支：

|  |
| --- |
| git push origin feature/issue-4-opening-report-submission |



#### 5）提交 Pull Request（PR）

代码推送到远程分支后，通过 GitHub 提交一个 Pull Request 来请求代码合并到主分支。

1. 在 GitHub 上创建 Pull Request：

进入项目的 GitHub 仓库，点击 "Compare & pull request" 按钮，进入 PR 提交页面。

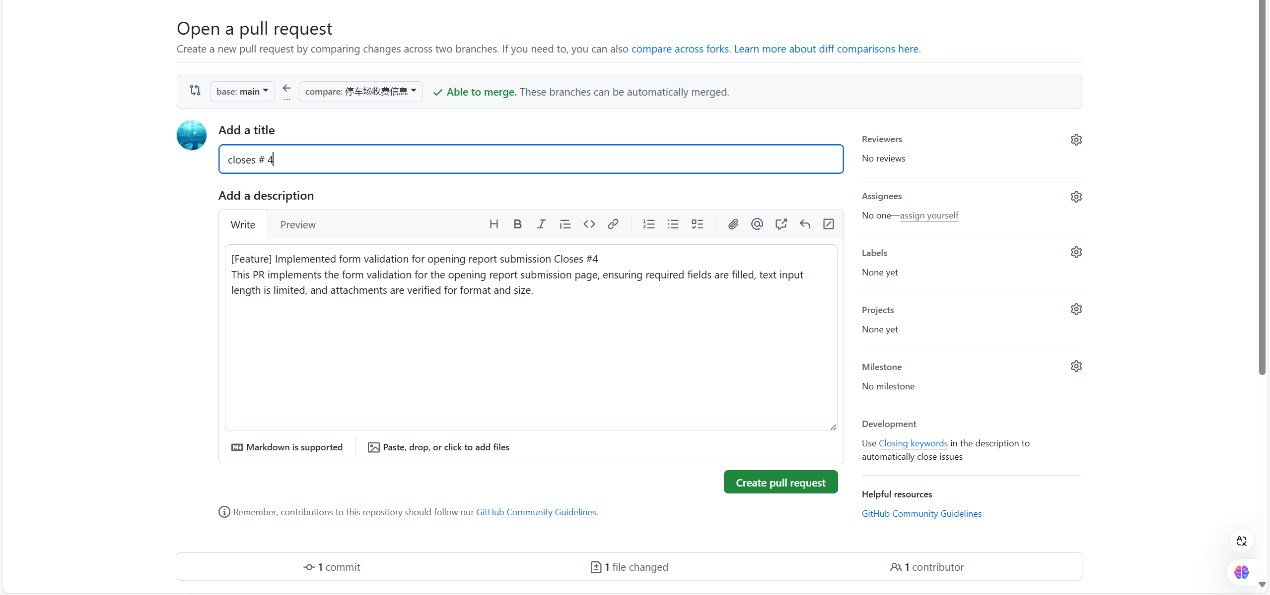
填写 PR 的标题和描述：

|  |
| --- |
| [Feature] Verification for parking fee management has been implemented  Closes #4  This PR opens validation for parking charges, making sure to fill in the required fields, limiting the text input length, and verifying the format and size of the attachments. |

2. 关联 Issue：

在 Pull Request 的描述中，再次提到了与 Issue 的关联：

|  |
| --- |
| closes # 4 |



3. 请求代码审查：

将 Pull Request 提交给团队成员进行代码审查。选择后端开发人员和测试人员作为审查者。



#### 6）等待代码审查和合并

提交 Pull Request 后，等待团队成员的代码审查反馈。审查者会检查代码的正确性和风格，如果没有问题，他们会批准并合并 Pull Request。

1. 代码审查反馈：

如果审查者发现需要改进的地方，会收到评论并根据反馈修改代码，然后再次提交。

2. 合并到主分支：

如果 Pull Request 通过审查，代码将被合并到主分支，相关的 Issue 也会自动关闭。

3. 删除分支：

在 PR 合并后，可以安全地删除本地和远程的功能分支，以保持代码库的整洁：

|  |
| --- |
| git branch -d feature/issue-4-opening-report-submission  git push origin --feature/issue-4-opening-report-submission |

## 2.4 冲刺评审会议(Sprint Review)

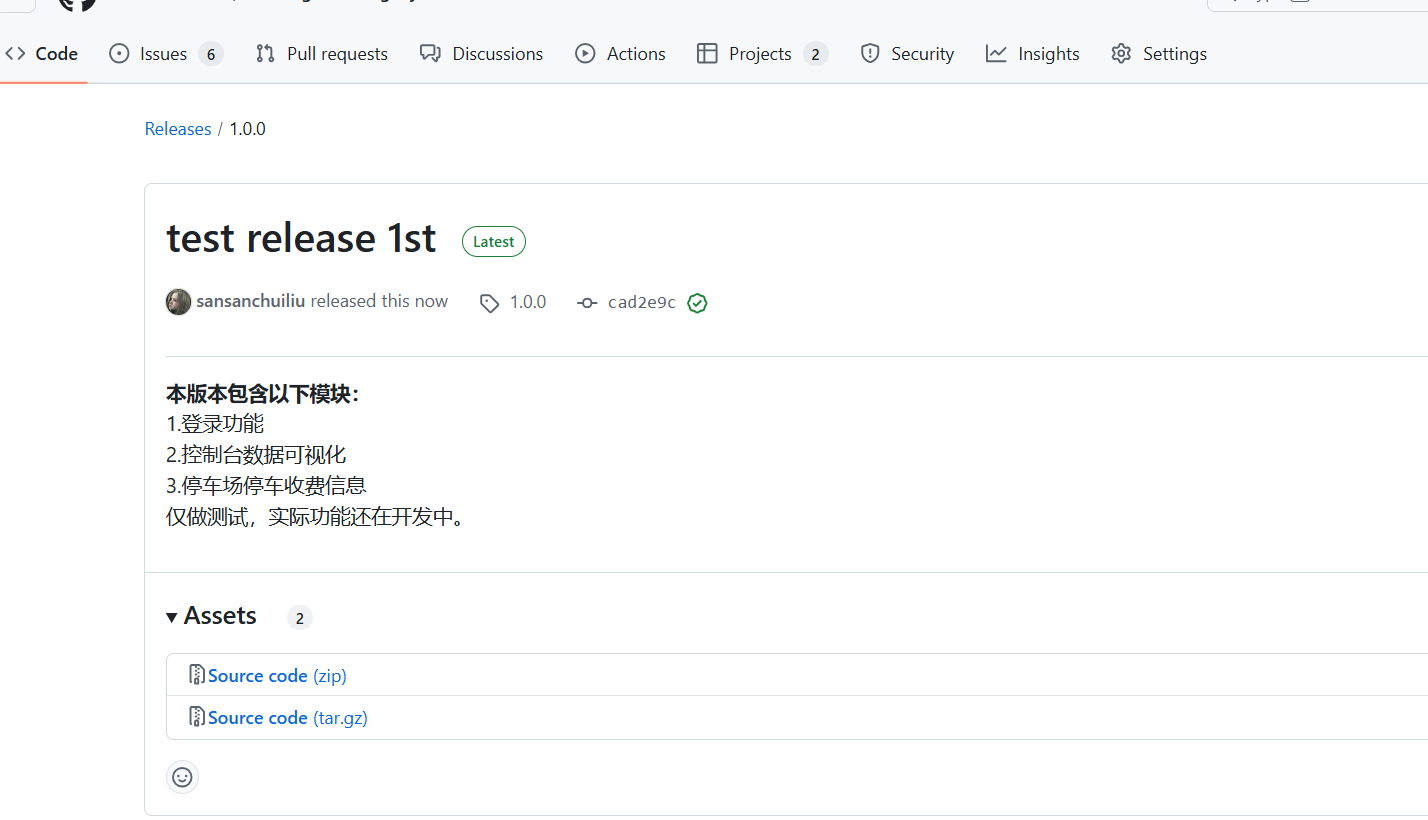
### 2.4.1 评审会会议纪要

|  |
| --- |
| 项目名称：停车场管理系统 会议日期：2024年10月30日 参会人员： 产品负责人（Product Owner）：张羽辰、熊志坤 开发团队（前端开发、后端开发、测试人员）：张俊辉、张治华、费亮 敏捷教练（Scrum Master）：田晋坤 利益相关者（管理代表、操作人员代表）：王老板、李经理 1.会议目的  本次会议旨在对第一次 Sprint 进行评审，展示已完成的功能模块，确保团队在此次冲刺中达成预期目标，并收集利益相关者的反馈。Sprint 1 的核心目标是实现停车场管理系统中的车位管理功能。 2.已完成的功能展示  产品负责人张羽辰、熊志坤对 Sprint 1 的目标和完成情况进行了总体介绍，确认以下功能已完成并通过测试：  1）车位状态监测功能：  功能描述：通过使用地磁传感器、超声波传感器等设备，实时监测每个车位的占用状态。系统将车位状态分为“已占用”和“空闲”两种，管理员可以在管理系统的界面上以直观的图形（例如，绿色表示空闲，红色表示已占用）或列表形式查看车位情况。 展示：张羽辰展示了车位状态监测的界面，包括车位状态的实时更新和图形化显示。通过模拟操作，展示管理员如何查看车位的占用情况。  2）车位引导功能：  功能描述：为进入停车场的车辆提供便捷的车位引导服务。在停车场入口或主要通道设置引导显示屏，以展示各个区域的剩余车位数量；此外，在停车场内部，通过指示灯和引导路线指示牌，帮助车辆快速找到空闲车位。例如，在岔路口的指示灯会根据车位占用情况显示箭头方向，引导车辆驶向空闲车位较多的区域。 展示：张羽辰展示了车位引导界面，包括引导显示屏的实时数据和内部指示灯的操作效果，演示车辆如何通过引导系统快速找到空闲车位。 3.利益相关者的反馈  王老板（管理代表）和李经理（操作人员代表）根据演示和使用体验，给出了以下反馈：  正面反馈：  王老板表示系统的车位状态监测功能非常直观，能够实时查看每个车位的占用情况，方便管理员快速做出决策，提高了停车场的管理效率。 李经理指出，车位引导功能非常实用，能够有效指引车辆找到空闲车位，减少了车辆在停车场内的徘徊时间，提升了用户体验。  改进建议：  王老板建议增加一个“车位使用历史”功能，允许管理员查看每个车位的使用记录，以便于分析车位的使用情况和高峰时段。 李经理建议在引导系统中增加语音提示功能，以便在停车场内部为司机提供更加人性化的导航服务，进一步提升停车体验。 4.未完成的功能  产品负责人张羽辰指出，由于接口文档准备不足，部分与后端对接的任务--数据统计与报表功能被推迟到了下一个 Sprint。这是本次 Sprint 中未完成的部分，团队将在接下来的开发中优先处理。 5. 改进计划与行动项  1）增加车牌识别功能：开发团队将在下一个 Sprint 中实现自动识别进入和离开停车场的车辆车牌号码，以便快速准确地记录车辆的进出时间  2）部署视频监控系统：开发团队将在下一个 Sprint 中安装多台摄像头，以实现对停车场的全方位视频监控。摄像头将实时录制停车场的状况，并将视频数据存储在本地服务器或云端，便于查询和回放。例如，在发生车辆刮擦、盗窃等事件时，管理员可以通过查看监控视频来确认事件经过和责任方。  3）优化报警系统：开发团队将在下一个 Sprint 中完善报警系统。当系统检测到异常情况（如非法闯入、车位被占用超过规定时间等）时，将自动触发报警。报警方式包括声音警报、短信通知管理员，或推送通知至相关工作人员的手机应用上，以便及时处理问题。 |

### 2.4.2 会议纪要的管理



## 2.5 版本发布(Release)



## 2.6 冲刺回顾会议(Sprint Retrospective)

### 2.6.1 回顾会会议纪要

|  |
| --- |
| 项目名称：停车场管理系统  会议日期：2024年10月31日  参会人员：  产品负责人（Product Owner）：张羽辰、熊志坤  开发团队（前端开发、后端开发、测试人员）：张俊辉、张治华、费亮  敏捷教练（Scrum Master）：田晋坤  1. 会议目的  本次会议成功评审了第一次 Sprint 的完成情况，获得了利益相关者的反馈，并制定了明确的改进计划和行动项。希望在下一个 Sprint 中能够继续推进停车场管理系统的各项功能，实现更好的用户体验和管理效率。  2. Sprint 回顾讨论  2.1 做得好的方面  1. 需求分析清晰：  在会议中，能够明确各种用户和利益相关者的需求，确保系统设计能够满足实际需求。  2. 技术方案讨论：  针对系统的技术架构、数据库设计、接口设计等方面进行深入讨论，确保技术方案的可行性和扩展性。  3. 用户体验设计：  充分考虑用户体验，讨论停车场管理系统的界面设计、人机交互等，以提升用户的使用便利性。  4. 数据安全和隐私保护：  针对用户数据的安全性和隐私保护进行充分讨论，制定相应的安全策略。  5. 培训与支持：  讨论如何为停车场的工作人员提供培训和支持，确保他们能有效使用管理系统。  6. 预算和资源分配：  合理讨论预算，确保项目在资金和资源上的有效分配。  7. 项目时间表：  设定明确的项目进度和阶段性目标，确保项目按期推进。   1. 风险管理：   识别潜在的项目风险，制定应急预案和应对措施。   1. 持续沟通机制：   建立会议后的沟通机制，确保团队成员之间信息畅通   1. 反馈机制   讨论如何在系统上线后收集用户反馈，以便及时进行改进。  2.2 需要改进的方面   1. 缺少车牌识别功能：   开发团队将在下一个 Sprint 中实现自动识别进入和离开停车场的车辆车牌号码，以便快速准确地记录车辆的进出时间  2. 缺乏监控系统：  开发团队将在下一个 Sprint 中安装多台摄像头，以实现对停车场的全方位视频监控。摄像头将实时录制停车场的状况，并将视频数据存储在本地服务器或云端，便于查询和回放。例如，在发生车辆刮擦、盗窃等事件时，管理员可以通过查看监控视频来确认事件经过和责任方。  3. 未完善报警系统：  开发团队将在下一个 Sprint 中完善报警系统。当系统检测到异常情况（如非法闯入、车位被占用超过规定时间等）时，将自动触发报警。报警方式包括声音警报、短信通知管理员，或推送通知至相关工作人员的手机应用上，以便及时处理问题。  3. 改进措施  在下一次 Sprint 开始前，产品负责人和后端开发团队将加强接口文档的准备。计划在 Sprint 开始前进行接口文档的审查，确保所有接口都经过详细设计并记录在案，以避免开发过程中反复沟通。  4. 总结  在本次 Sprint 回顾会议中，团队深入探讨了 Sprint 中的优秀表现以及待改进的方面。团队计划在下一个 Sprint 中实施这些改进措施，以提高开发效率和沟通质量，确保 Sprint 的顺利推进以及功能的及时交付。 |

### 2.6.2 会议纪要的管理



# 第3章 总结

在本次《信息系统课程设计》课程中，我深入学习和掌握了敏捷开发的基本方法，特别是在 Scrum 框架下，团队协作、任务分解与迭代开发等实践技巧对我而言尤为重要。通过理论学习与实践结合，我对敏捷开发的核心理念有了更深的理解，并在实际项目中将这些知识有效地应用了出来。

在项目实施的过程中，我逐步意识到了需求收集和需求分析的重要性。我们与客户和stakeholders 的密切沟通，使我们能够在每个 Sprint 开始前更准确地把握需求，并在 Sprint 结束时按时交付高质量的产品增量。这种交付模式不仅提高了客户的满意度，还为项目的顺利推进打下了坚实的基础。

参与 Sprint 计划会议、每日站会、Sprint 评审会议和 Sprint 回顾会议的过程中，我逐渐提升了自身的项目管理能力。这些会议为我的团队提供了一个高效的沟通平台，让我学会了如何高效地进行任务规划、合理分配资源，并有效管理团队成员之间的沟通与协调。通过与团队其他成员的积极互动，我深刻认识到良好的沟通是项目成功的关键，而合理的任务分配和资源管理则是实现项目目标的重要保障。每个 Sprint 中的产品待办事项（Product Backlog）的管理，使我意识到，优先级的确定不仅要考虑需求的重要性，还需平衡项目的进度与资源的投入，这对项目的成功实施具有重要的指导意义。

此外，GitHub 平台的运用让我对代码管理和版本控制有了更深的认识。通过创建 Issues 我们能够有效地记录和跟踪项目中遇到的问题，同时通过管理项目看板，可以清晰地展示当前的任务进度和状态，保证团队成员对项目的整体把控。而发布 Releases 则使我们能够在每个关键节点上，向客户展示我们所完成的工作。通过这些实践，我逐步提升了团队协作和代码质量管理的能力。GitHub 的工作流（例如 PR 流程、代码审查等）的运用，不仅帮助我们在确保代码质量的同时提高了开发效率，也促进了团队成员之间的知识共享和技术交流。

总结而言，本次课程设计不仅让我掌握了信息系统开发的流程和技巧，更重要的是提升了我在团队中的角色意识和项目管理能力。通过这次实践，我不仅理解了敏捷开发以及 Scrum 框架的具体应用，也学会了如何在团队中发挥自己的作用。这些宝贵的经验与技能，将为我未来的学习和工作奠定坚实的基础，产生深远的影响。展望未来，我将继续应用和深化这些知识，以更好地应对日益复杂的信息系统开发挑战。

# 附件：答辩纪录

**《信息系统课程设计》课程答辩记录**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 专业班级 | 软件2101 | 姓名 | 张俊辉 | 学号 | 21852001 |
| 答辩时间 | 2024-11-01 14：00-18：00 | | 答辩地点 | 思源楼422 | |
| 答辩题目 | 信息系统课程设计 | | | | |
| 答辩老师 | 田晋坤 | | | | |
| 提问及答辩记录：   1. 什么是 GitHub，它的主要功能是什么？   回答: GitHub 是一个基于云的版本控制和协作平台，主要用于软件开发。它允许开发者托管、管理和分享代码，其主要功能包括：  1）版本控制：使用 Git 来追踪代码的更改。  2）协作：允许多个开发者同时在同一个项目上工作，通过拉取请求（Pull Requests）进行代码审查和合并。  3）问题跟踪：提供问题跟踪功能，帮助团队管理任务和Bug。  4）项目管理：支持项目板（Projects）来组织和管理工作流。  2.如何使用 GitHub 进行代码审查？  1）首先，开发者在本地创建一个新分支并提交更改。  2）然后，他们在 GitHub 上创建一个拉取请求（Pull Request），选择目标分支进行合并。  3）在拉取请求页面，团队成员可以查看更改的代码、发表评论和提建议。  4）审查完成后，拉取请求的发起者可以根据反馈进行更改，然后再次提交。  5）一旦审查通过，合并按钮会变为可用，审查人可以选择合并拉取请求  3. 常见的敏捷开发框架有哪些？  1）Scrum  2）Kanban  3）Extreme Programming | | | | | |