**航空公司智能航班登机亭值机系统设计**

**结题报告**

**主 持 人：王光耀**

**小组成员：杨成宇**

**指导教师：王娜**

**学校：徐州市矿大实验学校**

# 项目说明

## 1.1项目介绍

本报告显示了该项目的开发过程。该软件是一个航空公司的智能航班登机亭，包括一个为希望获得登机牌的用户开发的简短而高效的登机系统。

当用户使用这个软件时，他们可以通过三种不同的方式登录，选择他们喜欢坐的座位，可以选择不同的餐食来满足他们，最终确定他们的选择并为之付费，然后获得自己的登机牌。

在软件开发过程注重合作，团队成员共同设计了该系统，尽力满足用户的需求，并对软件进行简短和有效的检查。

该报告将介绍我们的项目管理、需求、分析和设计，以及作为我们开发过程的一部分的过程和测试。

## 1.2项目成员和分工

每个成员都参加了项目的所有部分，项目成员分工明确。

王光耀：负责后端代码和生成Javadocs和自述文件，进行存储文件设计，实现用于与XML文件交互的相关方法，完成控制类的编码工作、存储文件、xml文件的设计和编写。

杨成宇：编写部分单元测试，设计和绘制整个原型，撰写并整理最终报告，完成大部分GUI，完善返回、弹出提醒、提示，完成用户手册。

# 项目管理

## 2.1项目计划

首先，我们项目制订了详细的计划，每一阶段都有冲刺的具体目标和假设。对整个时间计划和任务分配，我们使用传统的任务图表和网络来协助实现目标。每周都有会议，利用腾讯会议分配任务。在每次迭代中，可能会遇到各种各样的问题和难题，所以我们在会议上充分讨论，强化合作。当成员完成自己的任务时，通过微信及时展示自己的成果，以便更好地推进工作。

## 2.2风险处理

在实施过程中，我们通过加强合作来管控风险。当某一成员不能按时完成自己的任务，或者在截止日期前可能有自己无法解决的一些问题时，另外一个成员就会抽出时间来协助。通过充分的合作，使得计划能够如期完成。对于解决产品风险，我们使用多次迭代和多次评估来优化我们的产品。每次迭代后都有简短的讨论，通过腾讯会议总结我们上次迭代中的不足，提高下一次迭代的效率。如果在迭代中发生了亟待解决的问题，我们将在下一次迭代中解决这个问题，而不是打破当前迭代的现有顺序。

# 项目实施

## 3.1需求查找

开始开发之前仔细阅读项目要求并且结合实际的生活经历，确定了设计值机系统的主题。与此同时，根据我们在日常生活中的注册经验，一起讨论和编写用户画像。

通过互联网上查找了许多航空公司的智能航班登机亭，并下载了东方航空公司等航空公司的客户端应用程序，了解了它们的界面设计和功能。同时，我们也通过问卷调查了一些普通用户的注册需求。通过这些搜索和访谈，我们能够更清楚地了解软件的功能需求和设计。

## 3.2用户需求

在用户画像方面，我们生成大量用户画像，通过讨论删除一批，然后评估优先级。对于用户画像的汇编，我们的原则是创新和实用。

我们对用户需求的优先级有自己的理解。不需要根据优先级来完成用户需求。在每次迭代中，我们首先选择优先级较高的用户需求，然后根据具体情况选择一些权重较小的用户需求，以此提高开发的效率。

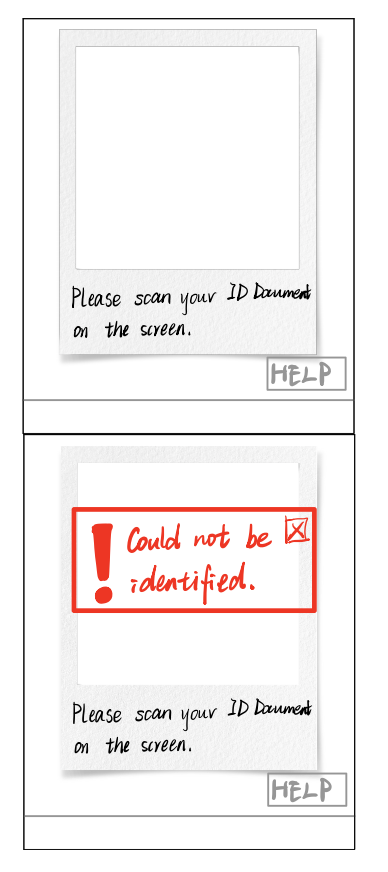
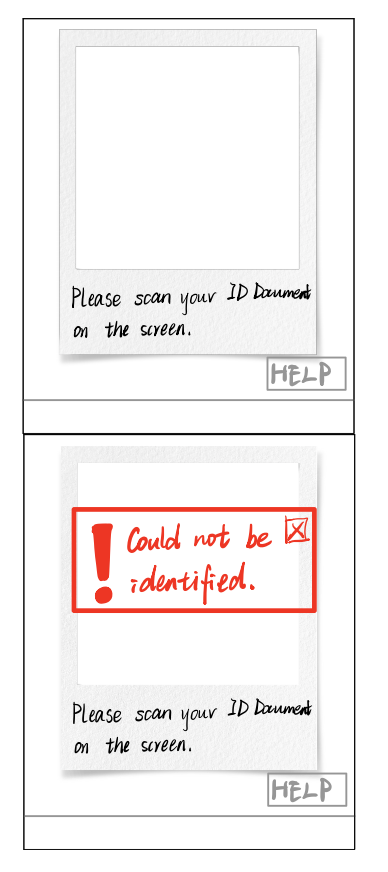
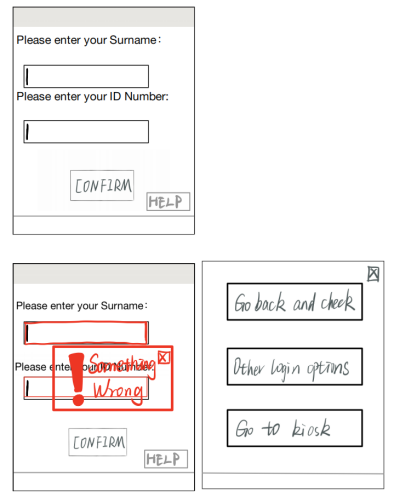
高优先级需求：登机系统登录、航班信息确认、座位选择、登机牌确认、待打印登机牌信息、待打印标签信息、取消操作、Java开发、简单文本文件格式的数据、检查乘客登机牌状态。

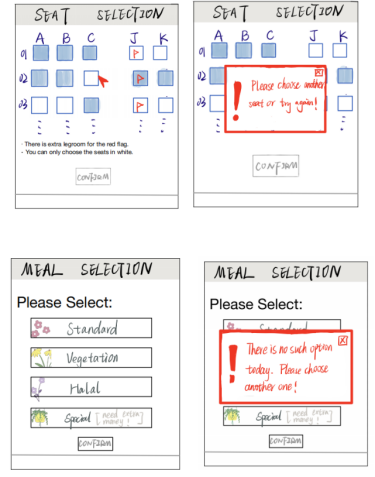
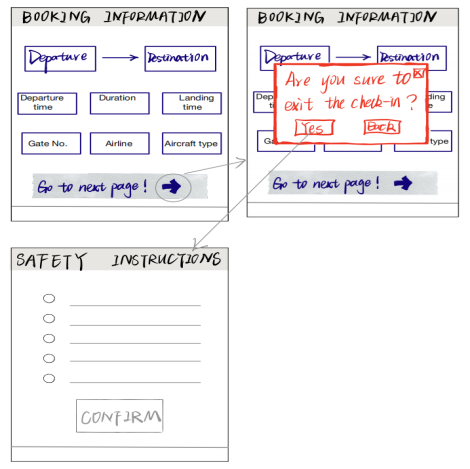
其他优先级需求：登录的替代方法、成功检索通知、机上餐食选择、打印机票信息、适应未来变化的能力、通知隐私条款、相关规定。

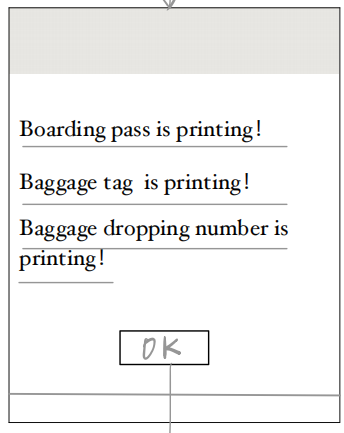
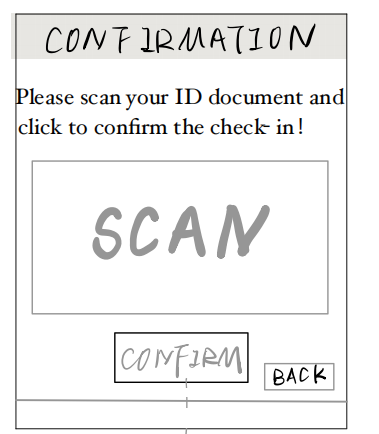
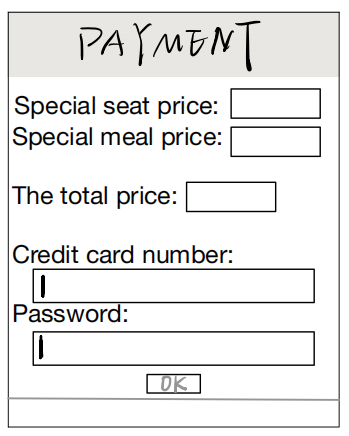
可以提供额外的座位服务，特殊的机上食品服务，基本的输入限制和错误检查。

## 3.3原型

## 1.1 capture_20220524210830309 capture_20220524211456624 capture_20220524211901875

31393935333136383b31393937373434373bbcfdcdb7



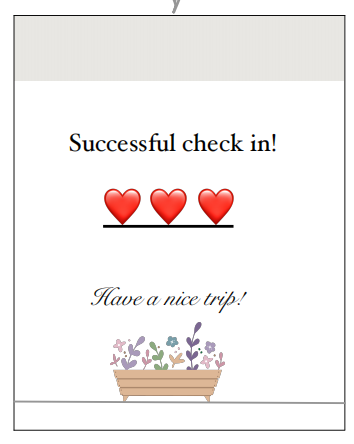


图1：原型图

## 3.4迭代计划和时间规划

|  |  |
| --- | --- |
| **时间** | 结果 |
| 1月11日至30日 | 建立小组，讨论项目。收集相关软件设计信息，了解部分操作签到系统的运行情况 |
| 2月1日至2月28日 | 用户需求分析，产品积压和原型。 |
| 3月1日至9月30日 | 迭代1：登录界面、航班信息用户页面设计、数据格式和存储文件设计、准备相关的查询和更新界面，以及四个高优先级功能（未与用户页面集成）。  迭代2：我们将在最后一次迭代中集成功能和GUI设计，以获得更完整的软件界面。同时，我们根据积压的情况增加了三个新的功能，并将它们集成到新的构建中。  三个功能：显示相关规定；基本限制和错误检查；附加服务 |
| 10月1日至11月1日 | 迭代3：在本次更新迭代中，我们完成了大多数低优先级的软件功能和页面显示，如打印界面、确认界面和结束界面等。同时，我们还完成了与弹出窗口相关的各种功能，如确认、提醒错误等，并标准化了输入格式。 |
| 11月2日至12月31日 | 在这次更新中，我们完善了姓氏记录和扫描确认的功能，几乎在每个步骤中都添加了返回功能，开始测试并组织我们的报告。 |

图2：规划图

## 3.5进行调整以适应需求的变化

在设计过程中，遇到了很多问题。例如，我们在迭代过程中意识到我们开始使用的函数是有缺陷的，于是我们首先细化在迭代中要完成的其他不受影响的任务，然后再处理函数问题。由于我们在一开始就确定了特定的用户画像，并分析了他们的特征，这一准备工作使我们能够在随后的设计过程中作出更清晰、更精确的调整。

我们的代码已经在多次迭代中被重构了好几次。为了响应变化，目前的设计是通用的、适应性强的和可扩展的。

# 4.分析与设计

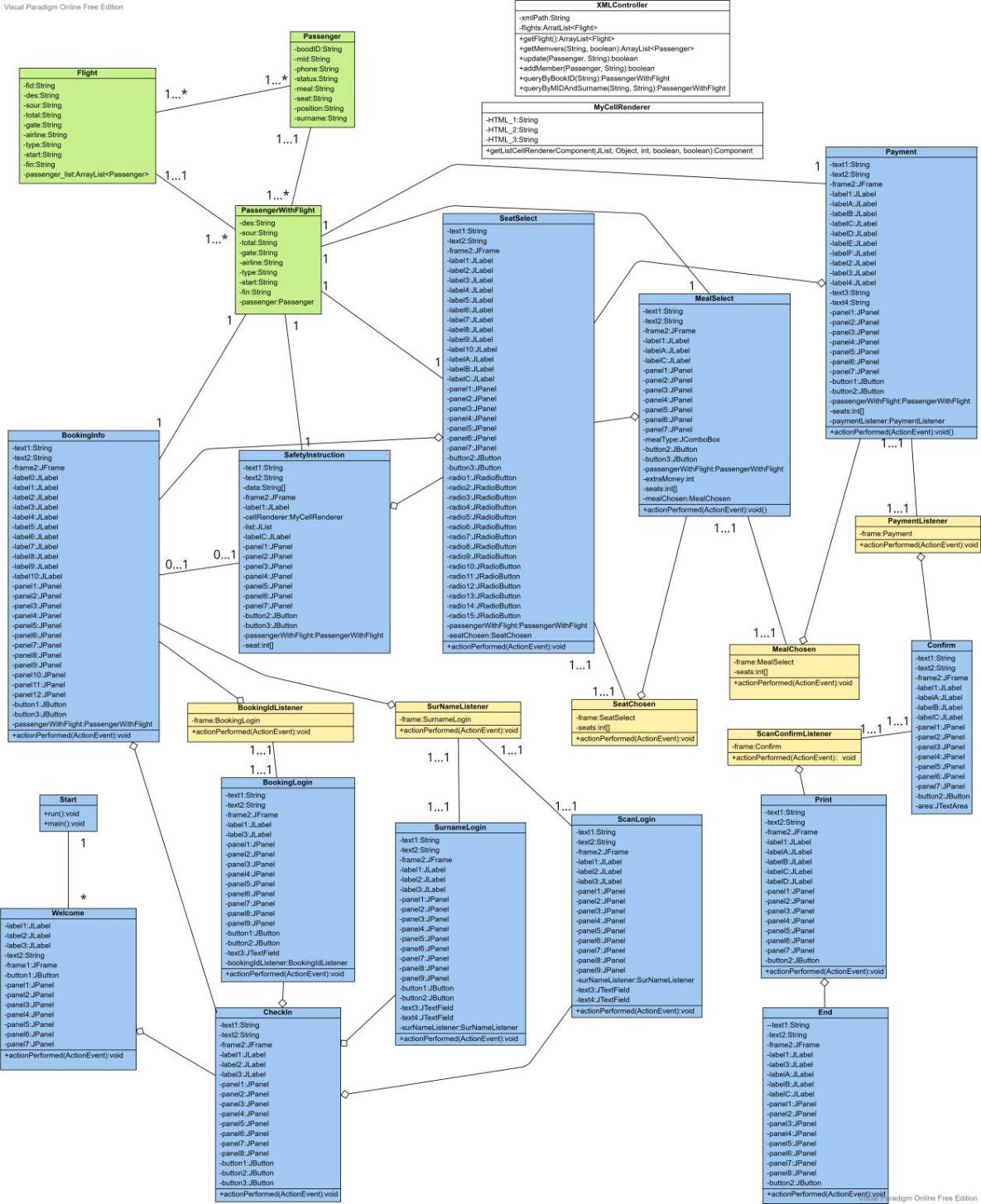
****

图3：Class diagram

在上图中，实体类为绿色。它们被用于建模长期存在的数据。Flight包含其乘客的列表和有关该航班的详细信息。Passenger代表那些已预订机票的人。PassengerWithFlight包含乘客信息和预订的航班的身份证。它将乘客和航班之间的多对多关系划分为两个一对多关系，并促进更新操作。

控制类用黄色表示，负责让用户从接口输入数据并处理它们。

边界类用蓝色表示，它们是程序中的用户界面。为了使它们连接起来，我们向一些按钮添加了操作监听器，以便它们可以在单击后生成下一个边界类的实例。对于需要用户输入数据的接口，我们将每个接口连接到一个负责处理这些数据的专门控制类。在处理之后，控制类将像边界类那样生成下一个接口。

大多数边界类和所有控制类都实现了执行监听器，但目的不同。操作控制类的执行方法用于处理数据并将其传递给下一个接口，而边界类的那些用于激活“返回”按钮，以防止主进程单向，使程序容忍用户的错误操作。

白色语言中的类是实用程序类。XMLContorller类包含用于与XML文件交互的所有方法，包括查询和更新。每次用户登录时，程序都会查询XML文件中的数据，并检查它是否与用户的输入相匹配。在主进程结束时，程序将记录所使用的已做出的选择，并更新XML文件。此外，我们还为机场工作人员提供了一种额外的查询方法，可以打印一个航班和乘客的所有信息，使他们更容易查看。MyCellRenderer被用来渲染一些接口的元素。

## 4.1设计原则

设计的总体原则是整个体系结构的合理便捷。子系统彼此之间是相对独立的。减少类之间的含义，以实现松散耦合。同时，子系统内部的连接尽可能紧密，相关的方法归为一类，因为它们很可能使用相同的属性。例如，xmlReader类包含用于查询和修改数据的方法。此外，如果一个方法的函数是复杂的，而代码又很长，我们分割出不同的部分。

为了使代码简洁和减少重复，我们将所有的布局样式放入MyVFlowLayout类中，可以多次使用。我们还采用了适当的封装来限制对这些对象的直接访问。所有属性都被标记为私有属性，它们只能通过我们指定的方式进行联系，因此可以防止任意篡改。

由于它是一个敏捷模型，考虑到该软件是一个不断更新和迭代的系统，并且涉及到与支付、背景、读卡等系统的集成，我们假设了相应的接口，为它们留出了一些位置，并考虑了它们需要的数据。

# 过程和测试

## 5.1过程

在实现的过程中，我们将设计映射到代码中。在这个阶段，我们发现了一些我们以前没有考虑过的东西和我们不确定如何实现的某些功能。我们还发现，该设计中的一些点需要进行修改。为了解决这些问题，我们不定期举行会议，从而达成新的共识，并将任务分配到下一次迭代中。同时，每次会议开始时都谈论自己最近的成果，对项目的进展有一致的了解，顺利进行模块对接。在每次迭代结束时，我们再次重新浏览新编写的代码并对其进行重构，从而确保没有不必要的代码。

整个实现和测试任务是灵活的。首先，确保代码执行时没有出现严重错误，然后实现各种函数并不断地细化它们，最后考虑可能会发生哪些异常。我们经常交流正在编码的部分，一方面，可以记录进度，另一方面及时存储了相关数据，以便可以随时找到以前的版本。

|  |  |  |
| --- | --- | --- |
| 迭代 | 构建 | 内容 |
| 1 | V1.0 | * 简单的登录界面 * 已购票的乘客可以通过输入预订号码来通过认证 |
| 1 | v1.1 | * 实施这三个签到通道 * 添加警告消息 |
| 2 | V2.0 | * 简单的座位选择，餐食选择 * 总价 |
| 2 | V2.1 | * 检查空座位 * 补充警告按摩 |
| 2 | V2.2 | * 改善餐食选择 * 添加返回，退出按钮 |
| 2 | V2.3 | * 验证帐卡号和支付密码 |
| 3 | V3.0 | * 确认登机信息、打印页面、结束页 * 修复剩余错误 |
| 3 | V3.1 | * 美化界面 * 修复剩余错误 |
| 4 | V4.0 | * 修复剩余的错误并优化一些功能 |

图4：构建计划图表

## 5.2测试策略

相对较小的项目，几乎所有的测试都是手动的。

检查用户界面是否按照我们的预期显示。组件位于它们正确的位置。

检查乘客航班信息显示正确，是否调用相关方法。

在乘客选择座位和用餐后，价格是否正确。

测试取消选择、返回最后一页、退出系统等操作处理正确。

在乘客确认所有信息后，检查签到状态数据的变化。

成功标准：没有未解决的高优先级缺陷。

## 5.3试验技术

验证测试。验证测试的目的是演示该软件满足其要求。我们使用的方法是分区测试，一种黑盒测试。它可以帮助我们找到有关功能需求的问题。具体的工具是JUnit。它在测试用户输入方面扮演着重要的角色，比如预订号码、姓氏和ID等等。以身份测试为例。成员ID由18位数字组成，其中第7至第14位数字为用户的出生日期。测试被分为9个分区。从分区1到分区3用于测试ID长度。分区4到分区6用于测试第11和12位数字，表示用户出生月份。分区7到分区9用于测试第13位和第14位数字，这些数字代表用户出生的特定日期。

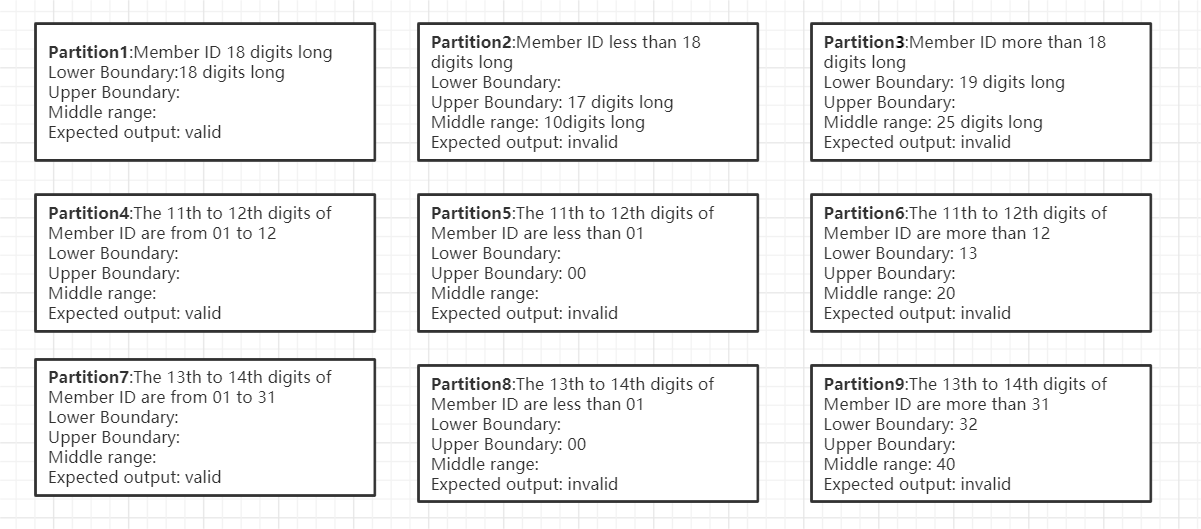


图5：分区

然后，我们在下表中显示了一个测试用例。

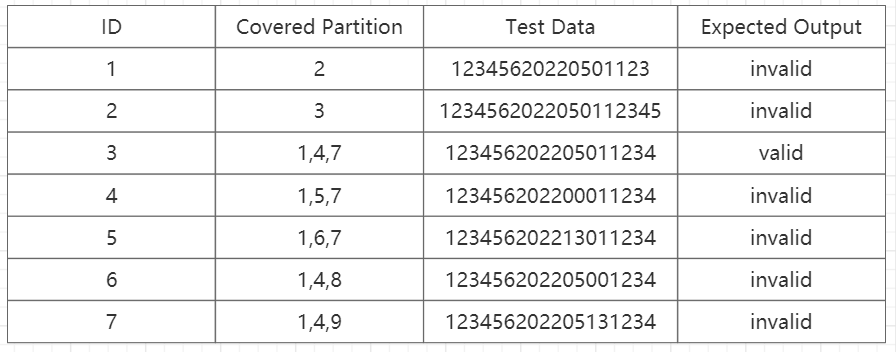


图6：案例和预期输出

然后，我们将使用测试驱动开发（TDD）来实现测试用例，代码在代码文件中。

缺陷测试。缺陷测试的目标是发现故障、错误和缺陷。我们主要使用白盒测试，以确保所有的语句和条件至少执行一次，以进行彻底的测试。

# 结论

该项目花了一年的时间，主要利用课余时间及假期完成。成员们积极参与了该软件的创建，这是我们在信息学课堂及竞赛上学到的知识的一种实践。通过本次项目的实施，我们建设了信息亭软件，同时对书本及课堂理论有了更深的理解和认识，激发了我们对信息领域的探索。我们对敏捷模型的理解更加具体和全面，同时也得到了指导教师的大力帮助。由于是第一次尝试系统地开发一个软件，所以我们的程序设计仍然存在许多不足之处。不过程序相对较小，目前在使用中没有严重的问题。虽然有很多缺陷，但我们的技能在不断解决问题的过程中得到了训练和提高，也更好的拓宽了我们的知识面，这是这个项目最大的价值。

# 附录

1. 参考文献
2. 伊恩·萨默维尔著的《软件工程》教科书。
3. 布雷特·麦克劳克林等人的“头部优先面向对象分析与设计”教科书。
4. <https://www.mountaingoatsoftware.com/agile>
5. 桑德拉·阿什莫尔的《敏捷的介绍》。