**可行性分析(研究)报告(FAR)**

目录

[可行性分析(研究)报告(FAR) 1](#_Toc17415)

[1引言 3](#_Toc13890)

[1.1标识 3](#_Toc9018)

[1.2背景 3](#_Toc17293)

[1.3项目概述 4](#_Toc22096)

[1.4文档概述 4](#_Toc1665)

[2引用文件 4](#_Toc12660)

[3可行性分析的前提 4](#_Toc24376)

[3.1项目的要求 4](#_Toc26247)

[3.2项目的目标 4](#_Toc2423)

[3.3项目的环境、条件、假定和限制 5](#_Toc30202)

[3.4进行可行性分析的方法 5](#_Toc28842)

[4可选的方案 6](#_Toc16224)

[4.1原有方案的优缺点、局限性及存在的问题 6](#_Toc22689)

[4.2可重用的系统，与要求之间的差距 6](#_Toc30631)

[4.3可选择的系统方案1 7](#_Toc10598)

[4.4可选择的系统方案2 7](#_Toc14885)

[4.5选择最终方案的准则 8](#_Toc24233)

[5所建议的系统 8](#_Toc21194)

[5.1对所建议的系统的说明 8](#_Toc24460)

[5.2数据流程和处理流程 9](#_Toc4930)

[5.3与原系统的比较(若有原系统) 11](#_Toc27100)

[5.4影响(或要求) 12](#_Toc20940)

[5.4.1设备 12](#_Toc21491)

[5.4.2软件 12](#_Toc26631)

[5.4.3运行 12](#_Toc17118)

[5.4.4开发 12](#_Toc30995)

[5.4.5环境 13](#_Toc7801)

[5.4.6经费 13](#_Toc16810)

[5.5局限性 13](#_Toc7068)

[6经济可行性(成本----效益分析) 13](#_Toc2597)

[6.1投资 13](#_Toc13463)

[6.2预期的经济效益 14](#_Toc20875)

[6.2.1一次性收益 14](#_Toc27267)

[6.2.2非一次性收益 14](#_Toc16913)

[6.2.3不可定量的收益 14](#_Toc15929)

[6.2.4收益/投资比 14](#_Toc17084)

[6.2.5投资回收周期 15](#_Toc1520)

[6.3市场预测 15](#_Toc22283)

[7技术可行性(技术风险评价) 15](#_Toc11468)

[8法律可行性 17](#_Toc17601)

[9用户使用可行性 17](#_Toc5095)

[10其他与项目有关的问题 17](#_Toc31062)

[11注解 18](#_Toc18584)

[附录 18](#_Toc13204)

[补充信息 18](#_Toc8711)

# 1引言

## 1.1标识

本文档可以在任何环境下运行查看。

## 1.2背景

随着我国交通的发展，简单的火车售票管理已经不能满足人们的需求。如何利用先进的管理手段，提高火车售票的管理水平，是当今社会所面临的一个重要课题。要想提高火车站的售票管理水平，必须提高火车站的管理意识。只要以高标准，高质量为准的火车售票管理才能适应当下人们的生活。

面对信息时代带来的挑战，利用高科技手段来提高火车售票管理无疑是一条行之有效的途径。如今，火车售票管理的信息化科学化，已经成为现代交通的重要标志。随着公众文化水平的提高，采用文字、图片的火车售票方式已经成为了必然的发展方向。该售票系统针对市场需求设计出可靠实用的管理系统和方案，可以根据不同的 需求实际配置，适应各种类型的车站。

## 1.3项目概述

火车购票系统项目，有设计两个对象，一个是普通的用户，一个是管理员：

对于普通的用户来说，它的功能主要分为四个模块，分别是基础功能，信息管理，核心功能，图形化数据管理，对于每个模块中的不同功能的实现，我们又将其进行细分，基础功能包含登录功能，修改密码，忘记密码，注册功能。信息管理包含个人信息维护，乘车人维护和登录日志的维护。核心功能包含路线查询，退改签，购票功能和订单查询。图形化数据管理包含车票查询和销量查询。安全性功能包含路由维护和密码功能。

对于管理员来说，也是分为五个模块，分别是用户管理，车票管理，车次管理，座位管理和图形化数据管理。其中具体的设计细节在后面的详细设计中会详细说明。

## 1.4文档概述

本文档可用于用户对火车售票管理系统的设计方案、设计过程以及设计影响做出全面的了解，同时可以帮助用户结合生活理解火车售票管理系统的实现意义与其所带来的价值。除此之外，本文档内容可以更加明确用户对该系统的使用过程。此文档由开发人员编辑，可供使用人员进行查看。

# 2引用文件

本文档暂未引用任何文档，属自行编写修订。

# 3可行性分析的前提

## 3.1项目的要求

初步要求:

（1）项目最基本的功能：

要实现用户和管理员两类对象的基本功能。对于用户来说，它的功能主要分为四个模块，分别是基础功能，信息管理，核心功能，图形化数据管理。对于管理员来说，其主要功能也是五个模块：分别是用户管理，车票管理，车次管理，座位管理和图形化数据管理。

（2）性能：

其要具备数据管理系统的高效性能，能做到及时响应和处理用户提交的需求。

（3）界面：

界面要求简洁明了，不需要对用户有过多的教程便可以使用。

（4）数据传输和处理：

车票系统的数据有很强的实时性的要求，数据传输需要快速，并且后台能够大量地处理并且要求极强的鲁棒性。

（5）项目完成时间：

本学期16周以前。

## 3.2项目的目标

此项目的目标是完成上述项目要求，并且在此基础之上，我们进一步地可以细化该项目的功能，以实现该项目的实用性和用户体验等等。

## 3.3项目的环境、条件、假定和限制

（1）项目运行环境：

车票管理系统需要运行在网页上，我们要考虑web开发，其次我们要考虑其在常用手机系统之上的运行环境，我们要考虑将其封装成安卓和iOS端的应用。其次其可能需要运行在一些嵌入式系统之上，比如车站的现场买票机器需要安装此系统，所以我们开发的时候需要考虑它的各种运行环境。

（2）项目的现有条件：

开发人员共有5名。

目前开发人员的技术栈只支持web开发，基于安卓和iOS端的应用开发日后也可以跟进并掌握。

目前只能配置本地数据库，可以根据项目以后的运行需求配置云端服务器以管理更大的数据量。

（3）项目的一些限制：

车票涉及到用户的交易，用户数据的安全性需要得到保证，如何保证用户数据的安全性是一个问题。

车座选座的时候我们需要支持多个用户同时访问选座界面，但是一个座位只能供一位用户选取，系统资源的临界性也是一个限制。

## 3.4进行可行性分析的方法

可行性分析的方法主要有两大类：

一、成本效益分析方法：

通过进行成本和效益的分析，可以判断项目的经济可行性，即投入产出的合理性，主要包括成本、直接收益和间接效益的分析。

（1）回收期分析法——只能作为辅助评价指标

回收期就是使累计的净现金流入等于最初的投资费用所需的时间。

优点：容易理解，计算方便。回收期越短，风险越小。

缺点：没有全面考虑投资方案总的可能的收益，只考虑收回之前的效果，不能反映投资回收之后的情况，即无法准确衡量方案在整个计算期内的经济效果。同时，还忽略了货币时间价值。

（2）净现值分析法——常用的项目评价方法

净现值就是未来报酬的总现值减去原先的投入。（现金流入量 > 现金流出量——>可采纳）

净现值法的决策规则是：在只有一个备选方案的采纳与否决决策中，净现值为正则采纳。在有多个备选方案的互斥选择决策中，应选用净现值是正值中的最大者。

二、技术及风险分析方法：

（1）技术分析

技术分析是要通过对技术设计方案或者演示模型的比较和分析，判断其技术的成熟性和适应性。这里最常用、有效的方法就是专家评定,即找相关行业的技术专家进行评审。

（2）风险分析

风险分析是对项目分别进行内部和外部的风险评估。最常用的方法就是定量分析法决策树。

# 4可选的方案

## 4.1原有方案的优缺点、局限性及存在的问题

优点：原方案的结构实现起来较为简单，较为方便，对技术栈的要求较低。

缺点：结构简单带来的功能上的不足，使得该系统的实用性不强。

局限和存在的问题:

一、用户层面：

（1）用户之间没有交互功能，即对于单一的用户只能对其账号进行购票操作，其不能对其他用户的账户进行操作。比如一个家庭当中，老人可能对购票系统的不熟悉导致其不会操作，这时需要其其他的家庭成员帮助其买票，这就需要原系统提供操作他人账号信息的部分功能，这是较大的局限。

（2）不同目的地之间并不是直接可达的，可能用户自己选取的路线车程与最优结果之间的差距较大，原系统没有提供给用户的购票建议，这样用户在查询其目的地的最优车票信息时较为麻烦。

二、管理员层面：

（1）在原有系统当中，我们对车次信息的导入是由管理员手动输入，其可能存在由于员工疏忽带来问题，我们需要尽可能地提升该系统的自动化程度。

（2）其没有可视化分析工具，使得数据的展示不是很清楚，使得原系统的数据利用较差，利用率不高。

## 4.2可重用的系统，与要求之间的差距

对于可重用的系统，我们要先了解其可重用的属性。所谓重用是指同一事物不做修改或稍加改动就在不同环境中多次重复使用。大量使用可重用的软件构件来开发软件，可以从下述两个方面提高软件的可维护性。

(1) 通常，可重用的软件构件在开发时都经过很严格的测试，可靠性比较高，且在每次重用过程中都会发现并清除一些错误，随着时间推移，这样的构件将变成实质上无错误的。因此，软件中使用的可重用构件越多，软件的可靠性越高，改正性维护需求就越少。

(2) 很容易修改可重用的软件构件使之再次应用在新环境中，因此，软件中使用的可重用构件越多，适应性和完善性维护也就越容易。

火车售票系统的结构在后端上是一套完整的逻辑，只需要给出接口，然后我们使用这些接口，完成这个系统所要完成的操作，其在任何环境下的运行都是采用同一套后端的体系。其在web，安卓，iOS上运行的最大不同是前端的开发。如果要提高该系统前端层面的可重用性，我们就需要在安卓和iOS端上进行封装，由于我们前端主体的开发价格是基于web体系的，我们在将其封装时需要封装对应的浏览器内核，这样与我们的要求就产生了较大的出入：

首先是我们要求该系统尽可能地简洁轻便，便于用户的安装，但是我们要多安装一个浏览器对其进行解析，就使得其在其他环境上的使用产生了冗余。

其次我们的要求是使得该系统的使用尽可能地流畅快速，但是基于上述方法的打包方式不能很好地调用其在本地环境下的系统功能。比如说我们的火车售票系统要使用导出发票文件的功能，如果基于浏览器的安卓或者是iOS端的应用，可能就会有卡顿，其是通过浏览器将该系统的行为翻译到安卓或者是iOS系统下的行为。

如果我们为了追求速度，我们就要舍弃一部分该系统的可重用性，我们就要基于安卓或者是iOS的生态将前端的架构重新编写，这样是更符合我们对该系统的要求。

## 4.3可选择的系统方案1

（1）用户登录注册：用户包括工作人员和乘客，有管理员、售票员、乘客3类用户；普通人可以通过注册功能成为新用户，用户通过登录可以使用系统提供的相应功能，如添加乘客，购买车票，查询订单等等。

（2）系统需要提供查询列车详细信息的功能：用户根据始发站和终点站，查询可以满足自己行程要求并且正常运行的列车，并且可以进一步查看开车时间，到达时间以及列车剩余座位的数量和票价；用户可以搜索具体的某一趟车次，可以得到该车次的具体路线信息以及发车时间。

（3）系统需要提供售票功能：售票员输入站点即可显示经过该站点的所有可售班次以及班次车辆的座位状态；一个售票员可同时售数张相同或不同站点，相同或不同票种（一等、二等、站票）的车票，可以实现累加本次售票款，直至下次新售票开始；还应实现辅助提示应找款以及退票的功能，售票员可以查询当天售票帐单，乘客也可以通过互联网购票。

（4）系统需要提供管理功能：包括用户管理、线路设置、站点设置、票价设置等。该系统主要分为四大部分，分别为用户、车票、车次和路线板块，每个板块可以进行查询与修改操作，管理员可以添加、修改和删除用户信息、线路信息、车票信息以及车次信息。

（5）系统需要提供营收结算功能：包括售票员结算、汇总报表。对营收缴款进行管理，可以实时掌握车站营收情况，方便财务监督。能够快速、准确地生成各种营收结算报表。汇总报表主要包括：根据用户选择或者输入的各种条件（包括：时间，售票员，票号等）汇总车站运营数据，包括：售票汇总，退票汇总，预售票汇总，废票汇总，综合汇总，财务汇总，售票员结算单等等。

（6）系统需要提供综合查询功能：包括：汇总线路、站点、班线、班次等各种基础数据；统计营运数据，生成日报表、月报表和任意时段报表；以汇总表形式，显示出售票张数、售票金额以及售票情况。

## 4.4可选择的系统方案2

（1）用户登录注册

用户注册登陆后可以使用系统的所有功能，如添加乘客，购买车票，查询订单等等

（2）系统需要提供基础的列车信息查询：

根据车次查询列车是否正常运行，以及查看列车的基本信息（如列车类型，始发站，终点站，开车时间，到达时间，运行时间，车厢数等等）

根据车次，查询列车经停站信息（包含这趟列车每一个经停车站的车站名，到达时间，开车时间，运行时间等信息）

（3）系统需要提供根据车次查询列车详细信息的功能：

根据始发站和终点站，查询可以满足自己行程要求并且正常运行的列车（可以根据开车时或者运行时间进行排序），并且可以进一步查看中间经过的的车站信息，以及开车时间，到达时间等。

系统需要提供接续换乘一次的查询，根据输入的出发站和终点站，可以查询换乘一次满足条件的列车，并且可以根据开车时间或总运行时间进行排序。

（4）车票购买

在查询到符合自己出行条件的列车后，可以查询列车的剩余座位以及购买车票。

首先添加乘客（添加需要购买车票的乘客）——>进行座位选择（为每一位乘客选择座位)——>订单支付——>购票成功

接续换乘车票购买流程类似，只不过在选座时，需要选择两趟列车的座位。

（5）系统需要提供用户的个人信息修改功能以及修改密码功能。

（6）用户可以给除自己以外别的乘客购买车票，所以提供添加乘客的功能，每个用户下都可以添加多个乘客，从而为别的乘客购买车票。

（7）系统提供订单的查询功能，可以查询到与自己有关的所有订单，比如所有订单，未支付订单，未出行订单。

（8）未支付订单针对下单但是没有支付的订单，可以在规定的时间内进行支付操作，如果在规定时间内没有完成操作，则订单会作废，变成未完成支付的订单。

（9）未出行订单针对已经支付但是没有出行的订单，可以在未出行订单中查看自己的出行计划。未出行订单可以进行改签操作，改签相同出发站和终点站的其他列车。

（10）未出行订单还可以进行退票操作，从而取消订单。

## 4.5选择最终方案的准则

以上是一些我们所考虑的火车售票系统的方案，对于最终方案的选取，我们会综合考虑每个方案的框架和每个模块的功能，尽可能地满足所建议系统功能的全面性。

# 5所建议的系统

## 5.1对所建议的系统的说明

技术栈说明(初步确定,根据开发实际情况再修改)：

本系统是一个基于Java EE开发的Web项目：火车售票系统。本项目采用前后端分离框架，项目用到的技术主要包括：

前端：Web的vue框架+ElementPlus+ECharts

后端：Springboot框架+Maven

数据库：Mysql

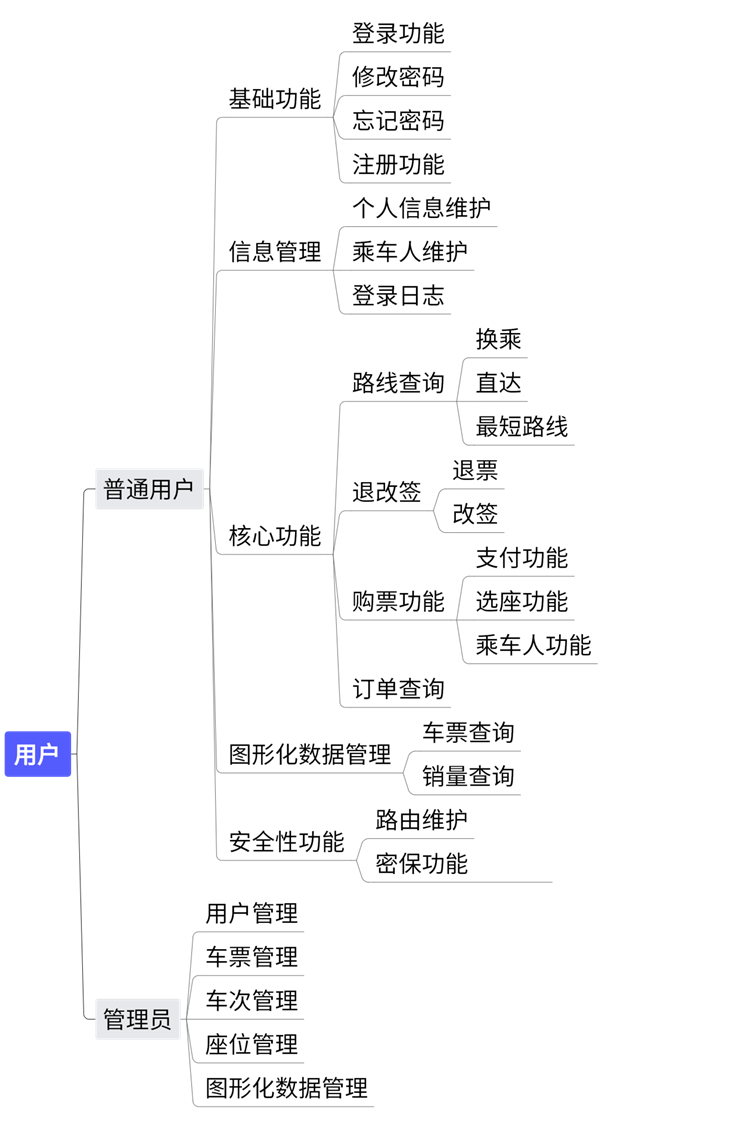
服务器：采用的springboot框架内置了Tomcat服务器。若用户数量增多随着，要 处理更大的数据量，应当部署性能更高的服务器。

功能性说明：

针对购票系统的这个项目，我们设计了两个对象，一个是普通的用户，一个是管理员。

对于普通的用户来说，它的功能主要分为五个模块，分别是基础功能，信息管理，核心功能，图形化数据管理，对于每个模块中的不同功能的实现，我们又将其进行细分，基础功能包含登录功能，修改密码，忘记密码，注册功能。信息管理包含个人信息维护，乘车人维护和登录日志的维护。核心功能包含路线查询，退改签，购票功能和订单查询。图形化数据管理包含车票查询和销量查询。安全性功能包含路由维护和密码功能。

对于管理员来说，也是分为五个模块，分别是用户管理，车票管理，车次管理，座位管理和图形化数据管理。火车售票系统的功能示意图如下：



## 5.2数据流程和处理流程

用户端操作处理流程：

一 、用户登录：

1.用户进入登陆界面，用例开始；

2.用户输入账号、密码，并向系统发送登陆请求；

3.系统检查用户信息，如果输入的信息错误则跳转到扩展路径A；

4.系统显示登陆成功

扩展路径A：1.系统要求用户重新输入相关信息或取消登陆；

2.用户选择取消登陆，跳转到扩展路径A的第五步；

3.用户重新登陆，跳转到基本路径2；

二、车票查询：

1.用户选择查询直达车次或换乘车次 ；

2.用户输入个人要求；

3.用户向系统发送查询请求；

4.系统检查请求并显示票务信息

三、订单管理：

用户进入订单管理界面，用例开始；

用户选择进行退票/改签操作的订单，向系统发送退票/改签请求；

3.系统接收退票/改签请求，查看是否满足退票/改签条件，若满足条件执行4，不满足执行5；

4.系统进行退票/改签处理，提示用户成功退票/改签；

5.系统提示用户无法退票/改签。

四、票务相关操作

1.用户查询车次后，选定车次、日期，选择购票功能，用例开始；

2.系统检查用户要求是否完善，车次及日期是否选择，检查错误则跳转到1；

3.切换界面到购票界面，选择乘车人以及座位类型进行购票；

4.进入支付界面，使用支付宝进行支付

五、个人信息管理

用户向系统发出管理请求，用例开始

2、系统显示相应界面

3、用户选择想要管理的个人信息

4、系统显示相关的个人信息

5、用户发送修改好的个人信息

6、系统显示管理成功，用例结束

管理员端操作流程：

一、用户管理

管理员进入用户管理界面后；

可查看所有用户的列表，展示用户的基础信息；

可以选择对用户进行拉黑操作，拉黑后用户进入黑名单。

二、车次管理

1、管理员向线上火车票预订系统发出请求

2、系统要求管理员输入需统计的车票信息

3、管理员输入需统计的车票信息

4、系统检查车票信息，显示统计成功，如果输入的需统计的车票信息出现

错误，转到 A ；

5、用例结束。

扩 展 点:A

<1>、系统要求管理员重新输入需统计的车票信息

<2>、管理员重新输入需统计的车票信息，转到 4

三、数据可视化/车票管理

1、管理员向线上火车票预订系统发出请求

2、系统要求管理员输入车次信息

3、管理员输入车次信息

4、系统检查车次信息，显示添加成功，如果输入的车次信息出现错误，转到A

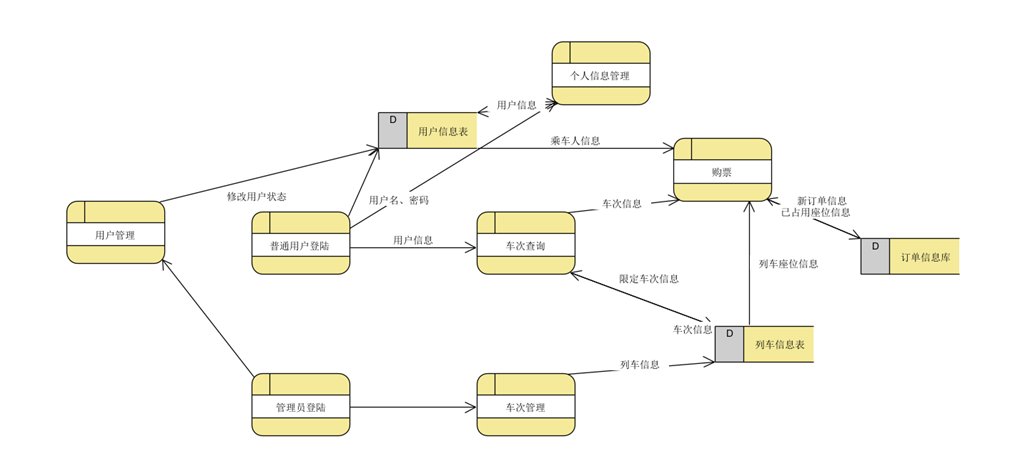
5、用例结束

扩 展 点:A

<1>、系统要求管理员重新输入车次信息

<2>、管理员重新输入车次信息，转到 4

数据流程示意图：



## 5.3与原系统的比较(若有原系统)

新系统与原系统相比,有一下优点：

在原系统基础之上加入了新的数据库支持，使用先进的数据库技术和数据管理技术：Mysql+MyBatis+Redis，使数据的准确性和安全性得到了很大的提高，且在用户的并行操作和用户管理方面有了极大的改善。

与原系统比较，新系统增强了用户业务办理的灵活性，提高了业务办理效率。新系统对于每一用户可以添加任意数量的乘车人信息，满足单一账号多用户购票，实现了用户之间的交互，使得业务办理灵活便捷。

用户核心功能扩展，实现了直达车次查询、换乘车次查询和最短路径查询，帮助用户选择最优路线，为用户提供全方位的优质服务。

后台管理功能升级：Excel导入车次信息功能可以大大减少工作人员的工作负担，节约人力资源开销，并且保证车次信息的准确导入，避免出现工作人员手动导入错误的现象；有效的提高了整体的工作效率和准确度。加入可视化图表展示售票相关信息的功能，可以更加直观的向后台工作人员展示售票情况，便于相关工作的开展。

安全性得到提高，在前端将用户信息隐藏，无法通过前端请求的反馈获得用户个人信息，如用户密码等；密码在后端进行加密保存；实现路由守卫，用户必须进行登陆校验才能进入相关界面；

容错性高，用户错误的操作之后会进行相关提示，例如时间选择的不合理等，且不直接删除用户填入的错误信息，给了用户部分修改的权力，无需从头修改；但是应更多地使用选择的模式，来限制用户的输入范围，降低错误发生的概率。

## 5.4影响(或要求)

### 5.4.1设备

对设备的影响主要是：随着新系统性能的提高，用户使用数量增多，系统需要处理的数据量增大，为了维持系统运行的稳定性和安全性，需要更换更先进性能更好的设备。主要需要更换的设备是服务器、网络设备等。

拟更换(购买或租用)以下服务器：

业务服务器用Nginx代理。

存储服务器：(暂定,根据实际情况更改)

CPU：Intel(R) Xeon(R) Platinum 8163

CPU @ 2.50GHz

RAM：8.0 G

云计算服务器：

CPU：Intel(R) Xeon(R) Silver 4210 CPU

@ 2.20GHz RAM：192GB

GPU：GeForce RTX 4080 \* 8

### 5.4.2软件

对软件的影响：需要落实是否有符合本报告列出的正版的软件环境，如果没有则需要购买。

例如本系统使用的后端开发平台：IntelliJ IDEA是JetBreain的一款产品，该产品具有完整功能的企业开发版需要1300元/年的使用费用；并且需要配置合适版本的java、Maven等用于后端开发；前端使用的集成开发环境是Visual Studio Code，也需要配置，并且配置相应的第三方插件。

### 5.4.3运行

系统设计的主要目的是提高效益。所以在系统设计和建设初期应着手参考各方面的标准和规范，并且应遵从该规范各项技术规定，并做好系统的标准化设计和管理工作。系统考虑今后发展的需要，因而必须为在系统产品系列、容量与处理能力等方面的扩充和换代的可能，这种扩充不仅充分保护了原有投资，而且具有较高的综合性能价格比。

### 5.4.4开发

新系统的开发环境与原有系统完全不同，因此需要搭建新的开发环境，购买新的开发软件和硬件，配置新的开发工具。新系统采用前后端分离架构，因此开发环境包括前端的集成开发环境：Visual Studio Code和后端集成开发环境：IntelliJ IDEA 2022.2.3(企业版)；数据库使用Mysql数据库并配合MyBatis Plus+Redis提高数据库的性能，简化开发的复杂度；API设计/开发/测试使用Apifox一体化协作平台(Postman+Swagger+Mock+JMeter);拟使用的业务服务器是Ngnix代理服务器，运行在Linux操作系统之上；文件服务器采用云端服务器。除此之外前端开发采用vue框架+ElementPlus+ECharts；后端采用Springboot框架+Maven。

### 5.4.5环境

由于使用者人数众多，水平不一，该系统设计时要考虑到使用者在使用时的操作简单，查询方便，软件设计采用模块方式，使用时用户基本不用担心应用还有其他功能，让用户一眼看出应用具备功能。

### 5.4.6经费

新系统增加了项目开发运维所需的经费。

对经费的影响主要体现在以下两方面：(1)前期开发所用的经费，包括人员酬劳和开发设备使用费用等；(2)系统硬件设备投资和运维费用：包括服务器、网络设备等。

前期开发以使用的集成开发环境为例：新系统使用的集成开发环境是IntelliJ IDEA(企业版)，该集成开发环境的使用费用为一个账户1300元/年，若需要更新版本还需要进一步投入资金；

系统硬件设备以项目使用的服务器为例：新系统采用的应用服务器是Nginx代理，租赁该服务器的最低费用是500元/月；文件服务器采用云端服务器，费用按量计算；随着新系统的使用，服务器的费用大幅提升。

## 5.5局限性

该项目目前的定位是一款web开发项目，目标实现火车售票系统的网页客户端，但在安卓、iOS移动端的开发尚未开展，网页客户端的便捷性明显不及移动端，所以该项目在用户使用的便捷性上有所欠缺，我们团队也希望能够开展移动端的开发工作来完善我们的火车售票系统。

# 6经济可行性(成本----效益分析)

## 6.1投资

基本建设投资：

系统硬件设备投资：包括服务器、网络设备、售票终端、打印机、扫码枪等硬件设备的采购投资。

系统开发投资：包括人员酬劳，开发设备，前期资料收集等先期投资。

建设工程投资：包括售票大厅、显示屏、音响设备、门禁系统等售票系统相关基本建设工程的投资。

一次性投资：

项目前期调研、立项、方案设计等费用。

售票系统推广费用：包括宣传推广、广告投放等费用。

人员培训费用：包括操作人员、维护人员等人员培训费用。

非一次性投资：

运营成本：包括系统维护、设备维修、软件更新，人力消耗的工资，奖金，差旅费等成本。

财务成本：包括融资成本、利息、贷款等成本。

税费成本：包括营业税、增值税、所得税等税费成本。

其中，基本建设投资是建立火车售票系统的核心投资，一次性投资用于项目启动和推广，非一次性投资则是用于系统的运营和维护。

包括基本建设投资(如开发环境、设备、软件和资料等)，其他一次性和非一次性投资(如技术管理费、培训费、管理费、人员工资、奖金和差旅费等)。

## 6.2预期的经济效益

成本控制方面：火车售票系统可以实现自动化的售票、退票、改签等操作，减少了人工操作成本，同时还可以实现电子化的票务管理，降低了票据印刷、存储、运输等方面的成本。

提高客户满意度：火车售票系统可以提供更快、更方便的售票服务，例如预订电子票、选择座位、在线支付等。这可以提高客户满意度并促进客户保留和口碑宣传。减少了后期的成本消耗，而且可以变相增加流量，提高效益。

提高数据质量和安全性：火车售票系统可以收集、存储和管理大量票务数据。这可以提高数据质量和可靠性，并促进数据安全和隐私保护。

改进票务管理：火车售票系统可以帮助管理人员实时监控和管理售票业务，包括票据发行、座位管理和票务结算等。这有助于提高管理效率和监督票务流程。

### 6.2.1一次性收益

系统开发费用：为了建立火车售票系统，必须支付一次性费用来开发、测试和实施系统，这可能包括软件开发、硬件采购、数据迁移等费用。

通过优化系统效率，数据存储方式的改进所节省的开支，多余设备出售的回收收入等，也意味着收益的提升。

### 6.2.2非一次性收益

可以规定从每一次售票中获得一定比例的提成，作为系统收益的一部分。

可以要求使用者/组织隔一定时间缴纳一定的维护费用，以租借而不是买断式提供系统的使用权。

可以利用上述的维护费用检查系统的可改进之处，不断优化系统，减小开支以达到扩大收益的目的。

### 6.2.3不可定量的收益

系统的口碑，若系统人性化并且易于使用，容易在用户之间积累较好的口碑，无形中提升了用户的使用量，从而提升了系统的非一次性收益。

提高效率和减少人力成本：通过自动化的售票系统，可以减少人工售票的工作量，提高售票效率，从而减少人力成本。

数据分析和营销：火车售票系统可以收集大量的数据，包括乘客偏好、乘车时间和路线等信息。这些数据可以用于进一步的分析和营销，以优化运营和提高收益。

提高安全性和可靠性：通过自动化的售票系统，可以提高乘车过程的安全性和可靠性，从而减少人为失误和安全事故的风险。降低了可能的额外支出。

### 6.2.4收益/投资比

收益和投资分别为上述相关的收入和开支的比值，对于一个软件来说，该比率可以用来衡量一个项目的经济效益，帮助投资者或企业管理者决策是否进行这个项目。因此我们可以通过提高收益或者降低投资的方式提高该比值。但不意味着偷工减料，而是要通过合理的分析手段进行系统的优化升级，以提供给顾客更好的体验。从而形成良性循环，促进软件的完善与发展。

### 6.2.5投资回收周期

火车售票系统的投资回收周期是指从开始投资到收回全部投资的时间。通常情况下，投资回收周期较短的项目更为有利，因为它们能够更快地产生收益，并使投资者更快地回收投资成本。但是具体回收周期的长短取决于多种因素，如投资额、项目规模、运营成本、市场需求等等。对于火车售票系统这种大型项目，由于要形成一个功能完善的系统，相关投资应该会比较大，开发周期也比较长，因此投资回收周期可能会比较长，通常需要数年或更长时间才能回收全部投资。

## 6.3市场预测

火车售票系统是一种电子商务系统，可以使乘客通过互联网或其他电子设备购买火车票，这种系统的市场前景非常广阔。主要体现在以下几个方面：

1. 市场规模：根据研究，火车售票系统市场在2020年的规模约为200亿美元，预计到2027年将增长至400亿美元左右。

2. 增长趋势：随着互联网和移动设备的普及，越来越多的人开始使用火车售票系统。这种趋势预计将在未来几年持续增长，尤其是在新兴市场的发展。

3. 竞争情况：目前，火车售票系统市场存在许多竞争对手，包括在线旅游代理商，铁路公司自己的网站以及其他出行的交通方式的发展。然而，该市场的前景仍然十分乐观，因为越来越多的人开始使用互联网进行购票，且火车仍然是较为快速和安全的出行工具。

# 7技术可行性(技术风险评价)

项目的开发系统前端采用：Web的vue框架+ElementPlus+ECharts；后端采用Springboot和Maven；数据库采用：MySQL，开发人员总共五名，经过两年多的计算机专业的学习，拥有一定的项目开发经验和扎实的计算机基础知识，也有着丰富的课外拓展实践的心得体会。本次软件工程项目的初期开发，不需要实验室硬件设备，只需要使用开发人员自己的电脑，以及需要很好的团队合作，合理的分工与配合，在项目的初期开发中所需的技术条件开发人员也都具备。

在项目的后期投入实践中，由于火车售票系统是一个庞杂的系统，可能会面临开发人员不足够，缺少设备以及更多的调研时间的问题，还可能会面临实际投入使用的一系列资金不足的问题，具体的补救措施为：可以邀请更多的志同道合的开发人员加入项目的开发，不仅仅局限于小组成员；还可以咨询感兴趣地承包方进行投资以及完成将火车售票系统真正投入使用的一系列工作。可以利用课余时间去火车站或者在手机app上进行调研与考察，评估现阶段的火车售票系统的缺点与不足，在自己的项目进行后续开发时进行弥补。

在项目的初期开发中，开发人员都有着良好的计算机项目研究研发基础，在技术水平上是合格的，并且开发人员可以进行良好的团队协作，预期可以在规定开发时间内完成项目的研发。在项目后期的经济问题上，由于火车售票系统是一个庞大、共通且不可或缺的平台，若能良好地开发并运行，并且寻找到感兴趣的承包商进行项目投资，它带来的效益是远大与成本的，所以在经济投资上可行的。

项目可能存在的风险有许多，如需求方面：需求描述不清晰、需求范围不确定、需求目标不明确、需求变更需求理解偏差；架构设计方面：架构不满足建设需求、设计缺陷、缺乏拓展性；进度计划方面：进度延期、计划调整；人员相关方面：开发人员离职、人员不足、能力不足；还有项目预算不足等等；还可能存在的不可控风险有近期的关于疫情的人员流动的政策调整、市场变化等。

面对风险的预案：

1、风险识别

目的在已经划定的边界范围之内，把所有的威胁或机会一个一个找出来，整理在一张表格上，列成风险记录单。

2、风险评估

当识别出来风险后，接下来要评估风险记录单中的每个风险，仔细地分析风险记录单中每个风险发生的概率和造成的影响，尽量做到量化。当评估风险的时候，对所有的风险进行优先级划分，确定哪些风险是要优先处理的。

3、为风险设定规避计划和应急计划

全方位思考把所有可能的应对方法都考虑到，然后比较不同应对方法之间的优缺点，最终挑选一个或几个最适合的应对方法并制订实施计划。

4、沟通

开发人员应该积极沟通，在项目管理中，需要在项目生命周期早期，针对项目相关方多样性的信息需求，制定有效的沟通计划，建立及时、有效的沟通机制。风险管理是贯穿整个项目周期的，需要项目团队全员的参与。

风险列举,分级,对应预案:

1.安全风险：火车售票系统需要处理大量的个人信息以及支付信息,如果系统受到黑客攻击或遭到数据泄露，将会对用户造成不可挽回的损失。

风险分级：高风险

应对预案：对数据进行加密,建立安全风险管理体系，加强系统安全性能，定期进行安全性评估和漏洞扫描，及时更新安全补丁，确保数据的保密性、完整性和可用性。同时，在用户注册和交易过程中加强身份验证和风险控制。

2.技术风险：火车售票系统需要大量的硬件和软件设备来支持其运作，如服务器、网络设备、数据库、应用程序等，如果其中任何一项出现故障或问题，将会对整个系统造成严重影响。

风险分级：高风险

应对预案：建立技术风险管理体系，加强系统监控和预警机制，定期进行备份和恢复测试，确保系统的可靠性和稳定性。同时，在系统运行过程中加强维护和更新，防止技术设备过时或失效。

3.法律风险：火车售票系统需要遵守相关法律法规，如用户数据保护法、电子商务法、消费者权益保护法等，如果违反相关规定，将会面临严重的法律风险。

风险分级：高风险

应对预案：建立法律风险管理体系，加强对相关法律法规的学习和了解，确保系统的合法合规。同时，在系统设计和运行过程中，加强用户知情权的保护，规范交易行为，保障消费者权益。

4.经营风险：火车售票系统面临着来自市场竞争、用户需求变化等方面的风险，如果不能及时适应市场变化，将会影响其持续发展。

风险分级：中风险

应对预案：建立经营风险管理体系，加强市场调研和用户需求分析，及时调整和优化系统功能和服务，提升用户体验和满意度。同时，加强与相关行业和企业的合作，开拓新的业务领域，扩大市场份额。

5.竞争风险：可能会出现市场上其他类似系统的竞争，导致用户数量减少、市场份额下降等问题。

风险分级：中/低风险

应对预案：增强市场宣传,提升系统的用户友好性,加强可用性,提高用户粘性,扩大市场份额,稳定用户群体。

# 8法律可行性

本系统作为软件工程这门课程设计，没有签订任何合同，不存在合同责任；所有的东西都是自己的，也没有挪用别人的成果，不存在侵犯专利权，版权问题。

法律方面的可行性问题很多，如合同责任、侵犯专利权、侵犯版权等方面的陷阱，软件人员通常是不熟悉的，有可能在系统开发时会造成侵权等一系列问题，例如系统开发时使用开源软件可能导致的侵权问题有：

1.无视开源协议使用开源软件导致侵权

开源软件并不是想用就用，不受任何限制。因为每一个开源软件中都附带了开源协议，并且在企业使用开源软件时，即视为已经接受了开源协议。如果企业的技术人员在使用开源软件为企业进行软件开发的过程中，忽视了开源软件所附带的开源协议，随意的使用开源代码，那么企业的最终的软件产品可能会因为违反开源协议而出现知识产权侵权问题。

例如GPL协议要求，只要在一个软件中使用GPL协议的产品，则该软件产品必须也采用GPL协议，既必须也是开源和免费。即企业的软件若使用了带有GPL协议的开源代码，但是在软件开发完成后不对软件开源，那么就可能违反GPL协议的要求，导致侵犯著作权人的权益。

2.开源许可协议之间不兼容导致侵权

再以GPL许可协议为例，GPL协议要求“用户必须以该许可协议的许可条款发布整个程序的源代码”，而MPL协议要求修改作品“受到本许可协议的条款管辖”。

可以看出这两个条款实际上是相互矛盾的，所以企业在使用同时附带GPL和MPL开源协议的开源软件时，就可能因为开源协议的冲突，面临必须会违反其中一个协议的困境。

# 9用户使用可行性

对于用户单位，在该软件的操作对象上应该区分为管理员和普通用户，管理员应周期性的对用户使用软件的数据安全性做出检查，保障用户的基本权益；同时，管理维护软件的人员应具备较高的软件调试水平和软件信息安全保护水平。

# 10其他与项目有关的问题

长期以来火车站管理工作一直是使用传统的人工方式或者是不完善的网络辅助方式，这样的管理方式存在着:效率低，大量文件和数据难于查找，更新维护时间过长。随着互联网技术的不断发展， 用信息技术改造传统行业， 是国家实现铁路现代化战略任务的迫切要求。 铁路信息化是铁路信息化的重要标志，将信息技术运用到铁路生产经营与各项管理决策中， 提高市场竞争力和经济效益，所以，开发出一款基于 web 的火车站售票管理系统就显得尤为重要了。

# 11注解

目前暂无注解,待后续添加。

# 附录

暂无附录,待项目开发时进行补充。

# 补充信息

分析各个软件开发模型的优缺点，并说明如何应用于自己的项目。

1.传统瀑布模型

传统瀑布模型是一种顺序型的开发模型，其开发过程依次包括需求分析、设计、编码、测试和维护等阶段。在这个模型中，每个阶段必须完成后才能进入下一个阶段。这种模型适用于需求明确、稳定的项目。

优点：具有明确的开发流程和阶段，易于管理和控制进度。在项目需求明确的情况下，能够保证项目的顺利开发。

缺点：瀑布模型是一种刚性模型，无法很好地应对需求不确定性的情况。缺乏灵活性，不能反映实际的代码开发方式，必须在项目开始前说明全部需求。同时，每个阶段必须等待前一个阶段完成后才能开始，会延长项目开发周期。

适用于：需求比较稳定、质量需求高，开发人员技术力量比较弱或者缺乏经验的项目。

2.原型模型

软件开发人员在与用户进行需求分析时，以比较小的代价快速建立一个能够反映用户主要需求的原型系统，在实践过程中提出改进意见，开发人员根据用户的意见，对原型进行补充和完善，然后再由用户试用、评价、提出建议，重复这一过程，直到用户对开发的原型系统满意为止。

优点：原型模型遵循了人们认识事物的规律，因而更容易为人们所普遍接受，可以更快地反馈用户需求，减少需求不确定性。同时，它可以更快地构建出基本的系统框架，提高开发效率。

缺点：原型模型可能会出现需求不清晰、进度控制困难等问题。项目可能难以规划和管理。

适用于：需求不明确或需要快速迭代开发的项目，或项目组成员（包括分析员、设计员、程序员和用户）不能很好交流或通信有困难的情况。

3.增量模型

软件开发的不同阶段是按软件产品所具有的功能划分，先开发主要功能或用户最需要的功能，然后，随着时间推进，不断增加新的辅助功能或次要功能，最终开发出一个强大的、功能完善的、高质量的、稳定的产品。

优点：增量模型可以更早地进行测试和验收，减少项目出错的风险。同时，项目成果逐步呈现，更容易与用户沟通。有利于增加客户对系统的信心，降低系统失败风险，提高系统可靠性，稳定性和可维护性。

缺点：增量粒度难以选择，把每个新的增量构建集成到现有软件体系结构中时，必须不破坏原来已经开发出的产品。容易退化为边做边改模型，从而是软件过程的控制失去整体性。

适用于：进行已有产品升级或新版本开发，完成期限严格要求的产品，对所开发的领域比较熟悉而且已有原型系统。

4.螺旋模型：

支持用户需求的动态变化并能够进行风险管理。第一周期开始利用需求分析技术理解应用领域，获取初步用户需求，制定项目开发计划（即整个软件生命周期计划）。经过一个周期后，根据用户和开发人员对上一周期工作成果评价和评审，修改、完善需求，明确下一周期软件开发的目标、约束条件，并据此制定新一轮的软件开发计划。

优点：支持用户需求的动态变化, 具有良好的可扩充性和可修改性。也支持软件系统的可维护性，每次维护过程只是沿螺旋模型继续多走一两个周期。原型易于用户和开发人员共同理解需求，还可作为继续开发的基础，并为用户参与所有关键决策提供了方便。螺旋模型为项目管理人员及时调整管理决策提供了方便，进而可降低开发风险。

缺点：建设周期长，而软件技术发展比较快，所以经常出现软件开发完毕后，和当前的技术水平有了较大的差距，无法满足当前用户需求。

适用于：大规模、复杂、风险高的软件开发项目，特别是需要长时间开发的大型项目。它适用于需要多次迭代开发、测试和验证的项目，可以通过早期风险分析和风险控制，降低项目失败的风险。此外，它还适用于需要不断变更需求的项目，可以在每个迭代周期中及时调整项目需求和目标。

5.敏捷开发模型：

敏捷开发作为一种新的软件工程方法，与传统方法相比更加注重人的因素。不再把开发者当作一个物化的，投入多少时间可以完成相应数量代码的代码开发机器；而是注重开发者之间的互动以及开发者和用户之间的互动，同时因为增加了交流和协作使得开发的项目更加灵活和易于修改。

敏捷开发分为四个阶段：

策划阶段

首先在策划阶段，用户和开发这进行交流，开发者总结出一系列“用户故事”，描述软件某一部分功能。之后客户对这些功能进行优先级排序，xp团队评估每一个故事的成本。之后客户和xp团队共同决定在开发的下一个版本中将会新增哪些功能。而在版本不断的迭代的过程中，会进行很多次这样的策划过程，每一次客户都可以根据已有的功能来决定是否要新增一些功能，以及要新增哪些功能。

设计阶段

在设计阶段，开发人员会根据用户故事，提出这些用户故事的实现方案。设计的过程中主要遵循简洁的原则，也就是尽量使用简介的表述而不是复杂的表述。而设计的另一个方面则是重构，重构是一种通过不改变代码的外部功能的情况下改变软件模块的内部结构从而优化软件系统的功能的过程。这是一种改进代码的设计。

在设计的这两个层面中，我们可以看到在xp开发过程中，设计和开发是同步进行的。我们在不断实现开始设计的过程中，同时要对到吗进行优化也就是重新设计。这样，大大的增强了整个软件开发的适应性，而不是始终刻板的实现最开始的第一版设计。

编码阶段

xp开发的第一件任务不是直接对初步的设计和用户故事进行编码，而是针对这些设计全力开发单元测试。完成了单元测试也就确定了开发者要实现的所有功能。这样开发者就只需要全力通过单元测试，而不必在实现什么功能上再浪费不必要的时间和精力。这正体现了敏捷开发的以测试驱动的特点。

而在敏捷开发中，很重要的一个提高效率的方式就是结对编程。在结对编程的过程中，两个开发者共用一台电脑，并各有分工。其中一个人进行实际的编码实现，另一个人在旁边考虑代码在宏观上该如何实现，比如针对什么功能应该使用什么样的算法。这样，在编码者工作遇到问题时，两个人交换位置。这时在旁边思考的人更有可能可以解决这一问题。事实上，结对编程的形式不必拘泥于什么规矩。关键在于，两个人共同开发的过程中，两个人的交流可以使得大部分的问题可以在第一时间解决。并且，因为两个人中只有一个人在进行编程这一项比较疲惫的工作，另一个人较为轻松，这样可以保证开发效率一直保持在一个比较高的状态。

测试阶段

每一个模块都通过自己的单元测试之后，开发者会将所有的模块集成到一起进行测试。这样可以及时发现每一模块在最近一次改动之中出现的问题同时避免一些兼容性问题。每一次改动一点小问题要比等到最后一次集中修改所有问题要容易得多。

与传统软件开发过程相比：

优势：

敏捷开发的高适应性，以人为本的特性，和轻量型的开发方法即以测试为驱动取代了以文档为驱动，这三个主要的特点，也就是敏捷开发相对与传统开发方式的主要有点。因为他更加的灵活并且更加充分的利用了每个开发者的优势，调动了每个人的工作热情。

劣势：

与传统开发方式相比，敏捷开发的最主要的劣势在于敏捷开发欢迎新的需求，而在每次新的需求产生时都可能引起整个系统的大幅度的修改。因为开发者在开发上一个版本的时候，完全没有考虑以后的优化将要如何进行。这样的开发方式实际的软件开发过程中，并不一定总是有效的。

而另一个方面，敏捷开发因为缺乏很多在敏捷开发中被认为“不重要”的文档，这样在一个大型项目比如一个操作系统开发的时候，由于其项目周期很长，所以很难保证开发的人员不更换，而没有文档就会造成在交接的过程中出现很大的困难。

适用范围：项目经常发生变更，高风险的项目实施，开发人员可以参与决策。

在小组的火车票管理系统的开发过程中，我们每一个人可以分不同的工进行合作，主要体现在根据每个人的能力侧重点不同，去分配给每一个人侧重、优势方面的任务，这样可以提高每个开发人员的参与度，突出每个成员的独特能力，同时可以大大提高软件开发的效率。采用敏捷开发为主，辅助以其它开发方式，例如螺旋模型中的迭代，瀑布模型的移植等。综合考虑组内成员的能力，加强沟通，以达到项目要求。