**软件需求规格说明(SRS)**

目录

[软件需求规格说明(SRS) 1](#_Toc9406)

[1范围 3](#_Toc28449)

[1.1标识 3](#_Toc21877)

[1.2系统概述 4](#_Toc19628)

[1.3文档概述 4](#_Toc14430)

[1.4基线 5](#_Toc20854)

[2引用文件 5](#_Toc16662)

[3需求 5](#_Toc4909)

[3.1所需的状态和方式 5](#_Toc1268)

[3.2需求概述 6](#_Toc22181)

[3.2.1目标 6](#_Toc3825)

[3.2.2运行环境 7](#_Toc13988)

[3.2.3用户的特点 8](#_Toc957)

[3.2.4关键点 8](#_Toc28172)

[3.2.5约束条件 8](#_Toc8126)

[3.3需求规格 9](#_Toc18552)

[3.3.1软件系统总体功能/对象结构 9](#_Toc9675)

[3.3.2软件子系统功能/对象结构 13](#_Toc5244)

[3.3.3描述约定 13](#_Toc19672)

[3.4CSCI能力需求 14](#_Toc3370)

[3.4.1登录与注册模块 14](#_Toc27635)

[3.4.2车次查询模块 16](#_Toc7311)

[3.4.3票务信息模块 18](#_Toc24386)

[3.4.4订单管理模块 20](#_Toc10050)

[3.4.5个人信息管理模块 22](#_Toc13798)

[3.4.6车次管理模块 24](#_Toc20796)

[3.4.7用户管理模块 25](#_Toc30696)

[3.4.8数据可视化模块 25](#_Toc28298)

[3.5 CSCI外部接口需求 26](#_Toc3230)

[3.5.1接口标识和接口图 29](#_Toc9896)

[3.6CSCI内部接口需求 30](#_Toc19752)

[3.7CSCI内部数据需求 31](#_Toc18299)

[3.8适应性需求 32](#_Toc27644)

[3.9保密性需求 32](#_Toc27967)

[3.10保密性和私密性需求 33](#_Toc24977)

[3.11CSCI环境需求 33](#_Toc3829)

[3.12计算机资源需求 34](#_Toc20399)

[3.12.1计算机硬件需求 34](#_Toc15108)

[3.12.2计算机硬件资源利用需求 34](#_Toc6123)

[3.12.3计算机软件需求 35](#_Toc28211)

[3.12.4计算机通信需求 35](#_Toc1879)

[3.13软件质量因素 36](#_Toc32549)

[3.14设计和实现的约束 36](#_Toc18551)

[3.15数据 37](#_Toc11005)

[3.16操作 37](#_Toc5610)

[3.17故障处理 37](#_Toc29651)

[3.18算法说明 38](#_Toc15224)

[3.19有关人员需求 40](#_Toc18463)

[3.20有关培训需求 41](#_Toc8279)

[3.21有关后勤需求 41](#_Toc25168)

[3.22其他需求 42](#_Toc11505)

[3.23包装需求 43](#_Toc2587)

[3.24需求的优先次序和关键程度 43](#_Toc12138)

[4合格性规定 45](#_Toc27978)

[5需求可追踪性 46](#_Toc15793)

[5.1CSCI能力需求可追踪性 46](#_Toc13339)

[5.2适应性需求可追踪性 47](#_Toc18672)

[5.3保密性和私密性需求可追踪性 47](#_Toc21863)

[5.4计算机资源需求可追踪性 47](#_Toc30093)

[5.5软件质量需求可追踪性 47](#_Toc4393)

[5.6数据需求可追踪性 48](#_Toc9243)

[5.7操作需求可追踪性 48](#_Toc27681)

[5.8故障处理需求可追踪性 48](#_Toc15085)

[5.9人员需求可追踪性 48](#_Toc7586)

[5.10培训需求可追踪性 48](#_Toc32040)

[5.11后勤需求可追踪性 48](#_Toc10452)

[5.12其他需求可追踪性 48](#_Toc25482)

[5.13包装需求可追踪性 48](#_Toc25832)

[6尚未解决的问题 48](#_Toc19518)

[7注解 49](#_Toc5141)

[附录 50](#_Toc17013)

# 1范围

## 1.1标识

系统名称：火车售票系统

标识号：TRSPS-001

标题：火车售票系统软件需求规格说明

缩略词语：TRSPS SRS

版本号：2.2

发行号：2023-04-24

## 1.2系统概述

随着近几年物质生活水平的飞速发展，加上最近疫情形势的逐渐好转，人们对于外出旅行的需求愈发增长，这就需要一个功能更全面、更先进的售票系统来保证来服务于大众。

火车售票系统是一款在线售票软件，旨在为用户提供便捷的火车票购买服务。该系统主要功能可以分为管理员端和用户端，其中管理员端包括管理车次信息、管理用户信息和查看售票情况等功能，用户端包括在线浏览车次信息、选择座位、购买车票、支付、改签、退票、查看订单和管理自己的信息等功能，用户可以通过该系统方便地完成火车票购买流程，管理员也可通过该系统更快捷无误地管理各种信息。

该系统的一般特性包括：支持多种座位类型、购票支付功能、改签退票管理订单功能、用户个人信息管理、车次信息查询与显示、余票信息查询与显示、用户信息的管理与更改等。

该系统的开发、运行和维护历史：该系统的开发开始于2022年，历时3个月完成，由小组成员协同合作开发。该系统于2023年正式开始使用，目前已经经历了多次测试与调整，运行效果良好。维护方面，该系统同样由小组成员分工合作进行维护，及时修复漏洞和更新版本。

项目的需方为火车售票公司，用户为需要购买火车票的人群，开发方为小组成员全体，支持机构为山东大学。当前的运行现场为火车售票公司的办公场所，计划的运行现场包括火车售票公司的网站和移动客户端。

其他有关的文档包括：软件需求规格说明、软件设计文档、测试计划和测试报告、用户手册和运维手册等。这些文档都是对系统开发、运行和维护过程中不同阶段的需求、设计、测试、使用和维护等方面的描述和说明，对于系统的成功开发、有效运行和稳定维护非常重要。相关文档包括《可行性分析报告(FAR)-火车售票系统》。

## 1.3文档概述

本文档主要用于描述系统所需要满足的需求和功能，从用户角度（系统的外部行为），以及从开发者角度（一些内部特性）来阐述需求，它描述了系统的行为、特性或属性，是在开发过程中对系统的约束，并且为软件开发和测试提供指导和依据。该文档的主要内容包括但不限于以下几个方面：

系统概述：对火车售票系统的整体介绍，包括系统的目标、功能、特性和运行环境等。

功能需求：对系统所需要实现的各个功能点进行详细的描述和分析，包括输入、输出、处理、操作、界面等方面。

非功能需求：对系统的性能、安全、可靠性、可维护性、可用性等非功能需求进行详细的描述和分析。

约束和限制条件：对系统开发和运行的一些约束和限制条件进行描述，包括硬件、软件、数据、安全和法律等方面。

测试需求：对系统各个功能点的测试需求进行描述，包括测试方法、测试环境、测试数据、测试用例等方面。

用户需求：对用户的需求和期望进行描述和分析，包括用户界面、用户体验、用户文档等方面。

安全需求：对系统的安全机制进行详细的设计，包括访问控制、身份认证、数据加密、漏洞修复等方面，确保系统的安全和稳定性。

保密性或私密性要求旨在对文档内容进行保护，在使用本文档时，需要遵守相关的保密性或私密性要求，确保文档的安全和完整性，为系统的开发和维护提供有力的保障。因为该文档涉及到系统的功能和性能等重要信息，如果泄露出去，可能会对系统的安全和稳定性造成影响。因此，在使用该文档时，需要遵守一些保密性或私密性要求，包括以下几点：

严格控制文档的分发和访问权限，只有具有相关权限的人员，即现阶段小组全体开发成员才能够查看和修改该文档。在使用该文档时，严禁将文档内容外泄给未经授权的人员。

对文档中涉及到的机密信息，如系统的算法、接口和设计等，需要进行加密和控制，防止泄露。

对文档的版本管理进行严格控制，确保各个版本的完整性和正确性，避免因版本混淆而造成的错误或混乱。

对文档的存储和备份进行规范管理，确保文档的安全和可靠性。

在文档的撰写和审核过程中，需要加强对机密信息的保护和控制，确保文档中没有涉及到机密信息的泄露和误传。

## 1.4基线

编写火车售票系统的系统设计说明书所依据的设计基线是软件需求规格说明文档，确保了系统的设计符合用户需求和系统规格要求。在软件需求规格说明文档中，已经详细描述了系统所需实现的功能和性能等方面的需求，本系统设计说明书的编写就是在这个基础上进一步详细设计出系统的结构、模块、接口、算法、数据结构等方面的设计。同时，在本设计过程中考虑到系统的可维护性、可扩展性、可重用性等方面的因素，为系统的开发和维护提供了有力支持。本系统设计说明书依据的设计基线为：

GB T-8567-2006计算机软件文档编制规范----软件需求规格说明（SRS）

# 2引用文件

本章应列出本文档引用的所有文档的编号、标题、修订版本和发行日期，也应标识不能通过正常的供货渠道获得的所有文档的来源。

[1]普雷斯曼. 软件工程：实践者的研究方法（原书第6版）. 机械工业出版社. 2007.1

[2]麦斯阿塞克. 需求分析与系统设计（第3版）. 机械工业出版社. 2009.1

[3]Jim Palistrant. IBM RSA和UML可视化建模指南. 机械工业出版社. 2007.6

[4]徐峰. UML面向对象建模指南. 水利水电出版社. 2006.9

[5]陈樟洪，金发华. IBM Rational Software Architect建模. 电子工业出版社. 2008.6

[6]比约尼尔(Dines Bjorner). 软件工程卷3:领域、需求与软件设计. 清华大学出版社. 2010.1

[7]黄国光，周勇. 软件需求工程. 清华大学出版社. 2008.5

# 3需求

本章应分以下几条描述CSCI需求，也就是，构成CSCI验收条件的CSCI的特性。CSCI需求是为了满足分配给该CSCI的系统需求所形成的软件需求。给每个需求指定项目唯一标识符以支持测试和可追踪性。并以一种可以定义客观测试的方式来陈述需求。如果每个需求有关的合格性方法(见第4章)和对系统(若适用，子系统)需求的可追踪性(见5.a条)在相应的章中没有提供，则在此进行注解。描述的详细程度遵循以下规则：应包含构成CSCI验收条件的那些CSCI特性，需方愿意推迟到设计时留给开发方说明的那些特性。如果在给定条中没有需求的话，本条应如实陈述。如果某个需求在多条中出现，可以只陈述一次而在其他条直接引用。

## 3.1所需的状态和方式

以下是该火车售票系统的CSCI可能会用到的状态和方式：

状态：

车次状态：表示每个车次的状态，是否可以被查询到。

方式：查询车次信息的方式，如通过车次号、出发站和到达站等进行查询。

座位状态：表示每个座位的状态，如已售出、未售出等。

方式：预订和购买车票的方式，如选择座位、支付方式等。

订单状态：表示每个订单的状态，如已支付、未支付等。

方式：查询已购车票信息等。

管理员状态，用户状态：表示每个用户的状态，如在线，不在线等。

方式：通过登录进行登入时可以设置。

## 3.2需求概述

### 3.2.1目标

a.本系统的开发意图、应用目标及作用范围(现有产品存在的问题和建议产品所要解决的问题)。

火车售票系统的开发意图是为了提供一个便捷、高效、可靠的火车票务售卖平台，让用户可以通过互联网轻松地查询、订购、支付、退票、改签等，实现从传统人工售票向自助售票、网络售票的转变，提高火车售票业务的效率和用户体验。

该系统的应用目标是为广大旅客提供一个方便快捷的售票渠道，让他们可以通过互联网随时随地查询和购买火车票，避免了传统的人工售票所带来的长时间排队和耗时的问题。同时，火车售票系统也能够为火车站管理人员提供一个高效的票务管理工具，实现信息化、智能化管理。

该系统的作用范围包括火车票售卖业务的各个环节，如车票查询、订购、支付、退票、改签、订单管理、车票管理、用户管理等。它能够方便用户的购票需求，也能够提高火车站管理人员的工作效率，同时也能够为火车售票业务的发展提供技术支持和创新。

现有产品存在的问题包括购票流程复杂、购票效率低、安全性不足、用户体验差等，建议产品需要解决的问题包括：提供简单易懂的购票流程、优化购票流程，提高购票效率、提供安全的支付方式和用户信息保护措施、提供良好的用户体验和多样化的服务、提高售票业务的管理效率和准确性等。通过解决这些问题，可以提高系统的竞争力，满足用户的需求，促进火车售票业务的发展。

b.本系统的主要功能、处理流程、数据流程及简要说明。

火车售票系统的主要功能包括车票查询、车票订购、支付、退票、改签、订单管理、车票管理、用户管理等。

处理流程主要包括以下几个步骤：

* 用户输入出发地、目的地、日期、座位等信息进行车票查询；
* 系统从数据库中检索符合查询条件的车票信息，并按照票价和发车时间排序；
* 用户选择车票并进行订购；
* 用户选择支付方式进行支付；
* 系统完成支付后生成订单，并将订单信息保存到数据库中；
* 用户可以进行订单查询、退票、改签等操作；
* 系统提供车票管理功能，管理员可以进行车票的添加、修改、删除等操作；
* 系统提供用户管理功能，管理员可以进行用户的添加、修改、删除等操作。

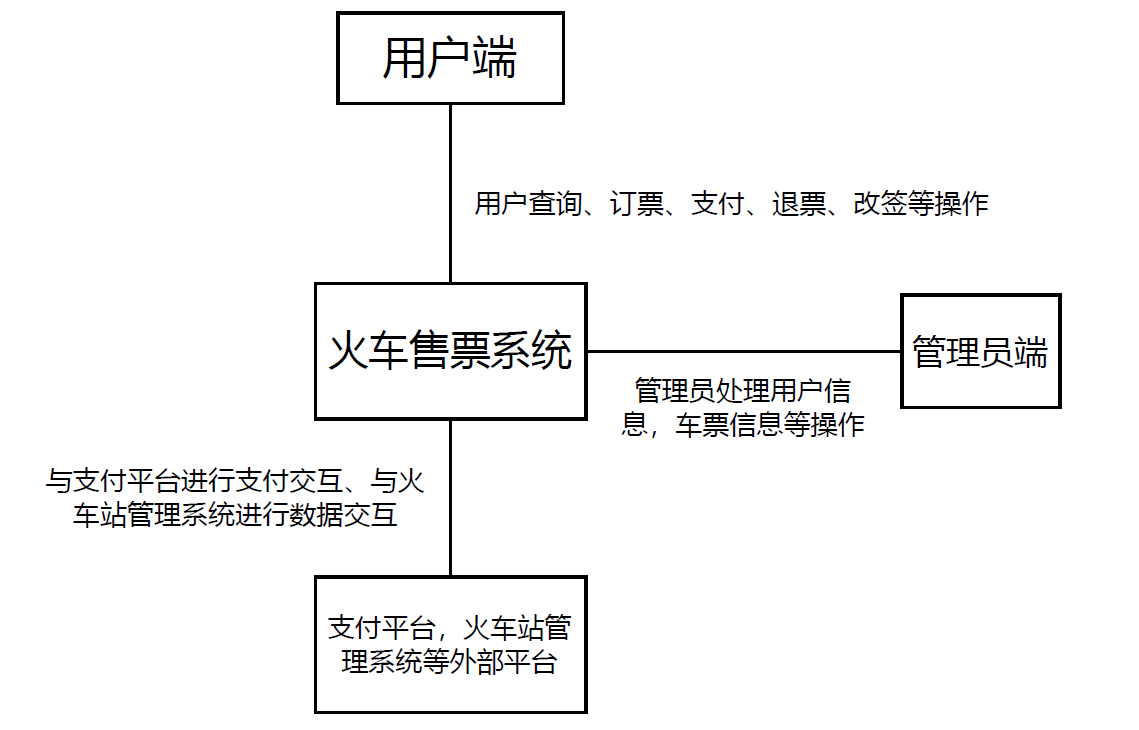
数据流程主要包括以下几个步骤：

* 用户输入查询条件，系统从数据库中检索车票信息并返回给用户；
* 用户选择车票进行订购，并填写相关信息；
* 系统根据用户的选择和填写信息生成订单，并将订单信息保存到数据库中；
* 用户进行支付，支付信息会被发送到支付平台进行处理；
* 支付平台处理完毕后，将支付结果返回给系统，并将支付信息保存到数据库中；
* 用户进行退票或改签操作，系统将订单信息更新并保存到数据库中。

简要说明：火车售票系统通过提供便捷、高效、可靠的售票渠道，实现用户在线购票、支付、退票、改签等操作，同时也为火车站管理人员提供了一个高效的票务管理工具，实现信息化、智能化管理。系统主要通过车票查询、订购、支付、退票、改签、订单管理、车票管理、用户管理等功能实现。处理流程包括用户输入查询条件、系统检索车票信息并返回给用户、用户订购车票、填写相关信息、系统生成订单并将订单信息保存到数据库中、用户进行支付、支付平台处理完毕后返回支付结果给系统、用户进行退票或改签操作等。数据流程包括用户输入查询条件、系统从数据库中检索车票信息并返回给用户、用户选择车票进行订购并填写相关信息、系统根据用户选择和填写信息生成订单并将订单信息保存到数据库中、用户进行支付、支付信息发送到支付平台进行处理、支付平台处理完毕后返回支付结果给系统并将支付信息保存到数据库中、用户进行退票或改签操作，系统将订单信息更新并保存到数据库中。

c.表示外部接口和数据流的系统高层次图。说明本系统与其他相关产品的关系，是独立产品还是一个较大产品的组成部分(可用方框图说明)。

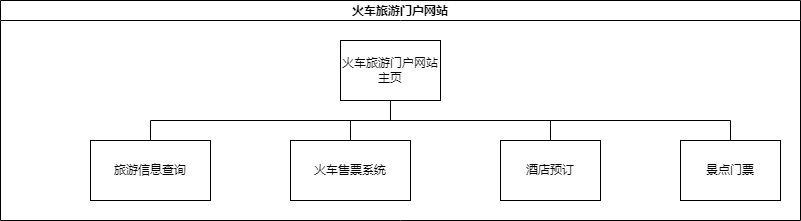
外部接口和数据流的系统高层次图：



火车售票系统是一个独立的产品，通过与支付平台进行支付交互、与火车站管理系统进行数据交互来实现其功能。它与其他相关产品的关系是通过数据交换来实现的，例如与支付平台交换支付信息，与火车站管理系统交换车票信息、订单信息等。

系统内部实现用户，管理员和系统的数据交互。

此外，火车售票系统可以作为一个较大产品的组成部分，例如火车旅游门户网站等，通过提供在线购票等功能来为用户提供更完整的服务。这可以通过下面的方框图来说明：



在这个方框图中，火车售票系统是火车旅游门户网站的一个组成部分，与其他模块（例如旅游信息、酒店预订、景点门票等）一起为用户提供完整的旅游服务。

### 3.2.2运行环境

理想状况下:

硬件环境要求：

服务器：推荐使用多核CPU、大内存、高速磁盘的服务器，以保证系统的高并发处理能力和数据存储能力。

客户端设备：火车售票系统支持通过Web浏览器、移动App等多种方式进行访问，因此客户端设备可以是PC、笔记本电脑、智能手机、平板电脑等。

支持环境要求：

操作系统：火车售票系统支持在Linux、Windows等多种操作系统下运行。

数据库：推荐使用高性能、高可用的关系型数据库，如MySQL、Oracle等。

Web服务器：推荐使用高性能、高可用的Web服务器，如Nginx、Apache等。

中间件：火车售票系统需要使用Java或其他编程语言开发，因此需要支持Java或其他编程语言的中间件，如Tomcat、JBOSS等。

支付平台接口：火车售票系统需要与支付平台进行交互，因此需要支持支付平台提供的API或SDK。

火车站管理系统接口：火车售票系统需要与火车站管理系统进行数据交互，因此需要支持火车站管理系统提供的API或SDK。

以上是火车售票系统的运行环境的基本要求，根据实际情况可能会有所不同。

我们开发小组目前的任务是使用笔记本进行开发浏览器应用，后台数据的处理放在数据库上,所以不会向上面所述规模那么大。

### 3.2.3用户的特点

火车售票系统的主要用户是火车乘客和管理员。其中，火车乘客作为系统的最终用户，可以通过火车售票系统完成火车票的查询、订购、支付等操作。管理员作为系统的操作员，可以通过火车售票系统完成售票、退票、改签等操作，同时还可以管理车票信息、班次信息、车站信息等。

从使用系统的特点来说，火车售票系统的用户通常具有以下特点：

1.对时间的敏感性：火车乘客需要在有限的时间内查询、订购和支付火车票，因此系统需要提供快速、准确的响应。

2.对信息的准确性要求高：火车乘客需要获取准确的车次信息、车票价格、余票数量等信息，售票员需要准确地售票、退票、改签等操作，因此系统需要确保信息的准确性。

3.移动设备的普及性：随着智能手机、平板电脑等移动设备的普及，越来越多的用户使用移动设备进行火车票的查询、订购、支付等操作，因此系统需要支持多种移动设备的访问。

4.复杂的业务流程：火车售票涉及到复杂的业务流程，包括车次查询、车票预订、票款支付、售票、退票、改签等环节，系统需要支持这些业务流程的顺畅进行。

5.用户数量庞大：火车售票系统是一个高并发、大规模的系统，需要支持大量用户同时访问，因此系统需要具备高性能、高可用、可伸缩等特点。

### 3.2.4关键点

火车售票系统的软件需求规格说明书中的关键点包括：

关键功能：火车售票系统的关键功能包括车次查询、车票预订、票款支付、售票、退票、改签等功能。这些功能是用户使用系统的核心，系统需要确保这些功能的准确性、高效性和可靠性。

关键算法：火车售票系统的关键算法主要包括车票价格计算、余票数量更新、票款支付等算法。这些算法需要确保数据的准确性，同时还需要保证算法的高效性和可靠性。

关键技术：火车售票系统涉及到的关键技术包括数据库技术、网络技术、安全技术等。其中，数据库技术用于存储车次信息、车票信息、用户信息等数据，网络技术用于实现用户与系统之间的交互，安全技术用于保护用户信息和系统数据的安全。

数据库设计：火车售票系统需要设计数据库，包括车次信息、车票信息、用户信息等数据的存储和管理。数据库的设计需要考虑数据的完整性、一致性和可靠性。

系统安全性：火车售票系统需要保证系统的安全性，包括用户信息的安全、支付信息的安全、系统数据的安全等。系统需要采用安全技术，如SSL协议、数字证书等，保证用户信息和系统数据的安全。同时，系统还需要采用安全的编码规范和防范措施，防止恶意攻击和安全漏洞的发生。

性能要求：火车售票系统需要具备高性能和高可用性。系统需要支持高并发访问，能够承受大量用户同时访问。同时，系统还需要具备可伸缩性，能够根据业务需求调整系统资源，以满足用户的需求。

### 3.2.5约束条件

1. 经费限制：系统开发的经费限制是必须考虑的约束条件，需要根据预算进行合理的分配和利用，确保项目能够按时完成。
2. 开发期限：系统开发的时间限制是一个重要的约束条件。需要根据实际需求和项目进度，合理制定开发计划和时间表，确保项目能够按时完成。
3. 所采用的方法与技术：系统开发需要采用一定的方法和技术，需要考虑到团队的技术水平和能力。同时，还需要考虑到系统的性能要求、安全要求、易用性等因素，以选择最合适的技术和方法。
4. 政治、社会、文化、法律等方面的约束：开发过程中需要遵守相关的政治、社会、文化、法律等法规和规定，确保项目的合法性和规范性。例如，需要保护用户信息，遵守相关的隐私法律；同时还需要遵守相关的安全法规，确保系统的安全性。
5. 人力资源：开发团队的人力资源是约束条件之一。需要根据项目需求，招募和培养合适的人才，以保证项目能够按时、高效地完成。
6. 可行性：进行火车售票系统开发之前，需要进行可行性研究，评估项目的技术可行性、商业可行性和风险可行性等方面的问题，确保项目的可行性和成功的实现。

## 3.3需求规格

### 3.3.1软件系统总体功能/对象结构

这个模型包含以下几个元素：

起始状态（S）：在这个状态下，系统处于空闲状态，等待用户的操作。

输入库所（I）：当用户提交购票请求时，将会产生一个标记进入该库所。这个库所包含三个子库所，分别是：

起点站子库所（SI）：表示用户所选的列车的起点站。

终点站子库所（DI）：表示用户所选的列车的终点站。

座位子库所（CI）：表示用户所选的列车座位的类型和数量。

车次信息库所（MI）：这个库所包含了所有列车的信息，包括起点站、终点站、发车时间、到达时间和座位类型等。

查询所（T1）：当用户提交购票请求后，系统会从输入库所读取用户的选择，然后在车次信息库所中查询符合条件的车次，如果有可用车次，则从车次信息库所移动一个标记到查询所。

预定库所（O）：当系统在车次信息库所中找到符合条件的车次时，将会进入预定库所，这个库所包含两个子库所：

座位信息子库所（CI）：表示所选的列车座位的类型和数量。

票据信息子库所（BI）：表示该预定的票据信息。

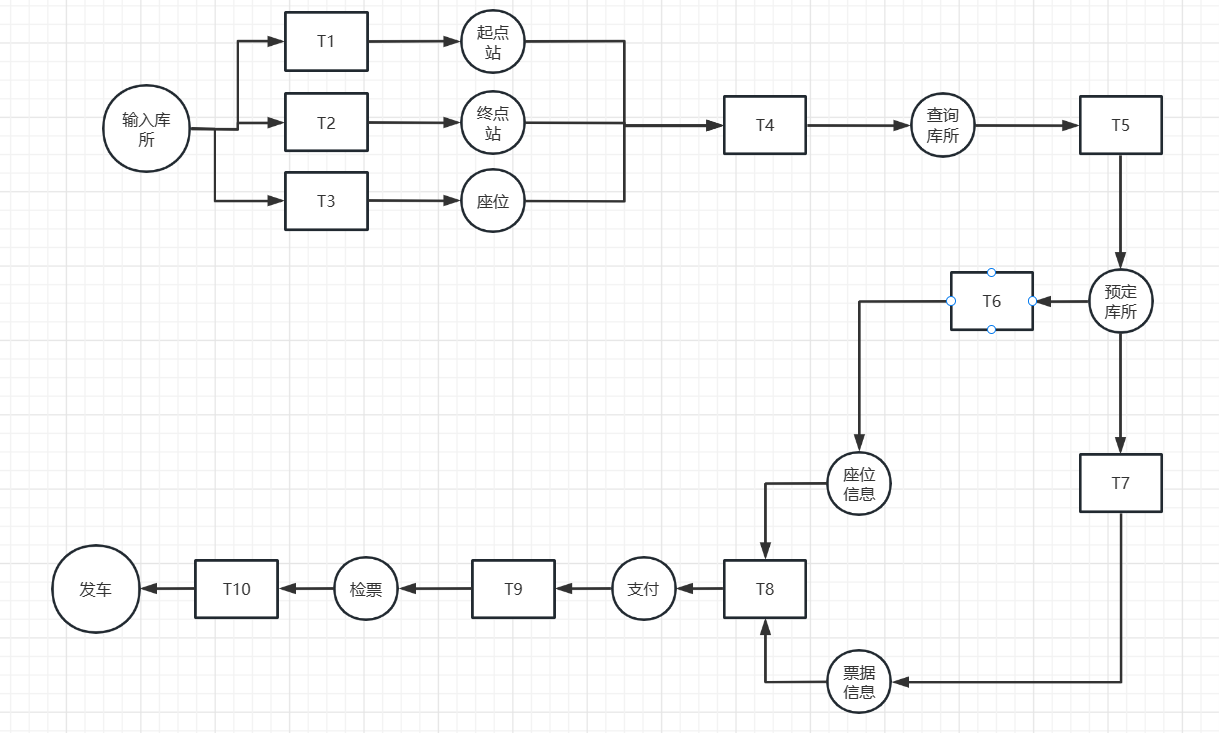
支付所（T2）：当用户确认购票后，系统将从预定库所移动一个标记到支付所，用户将支付相应的费用。

检票所（T3）：在购票成功后，用户需要前往检票口进行检票。当系统检测到有足够的座位并确认购票成功后，将会从支付所移动一个标记到这个检票所。

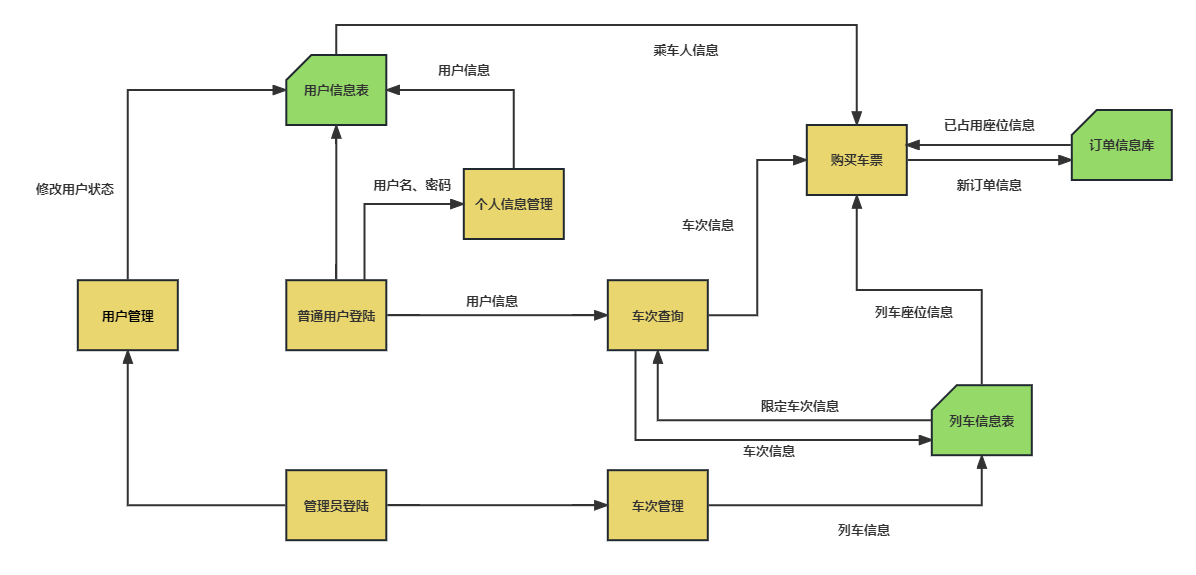
发车所（T4）：当所有乘客都检票完毕后，系统会通知列车开始发车，并将标记从检票所移动到发车所。

完成状态（E）：在发车之后，系统将进入这个状态，表示该购票流程已经结束。

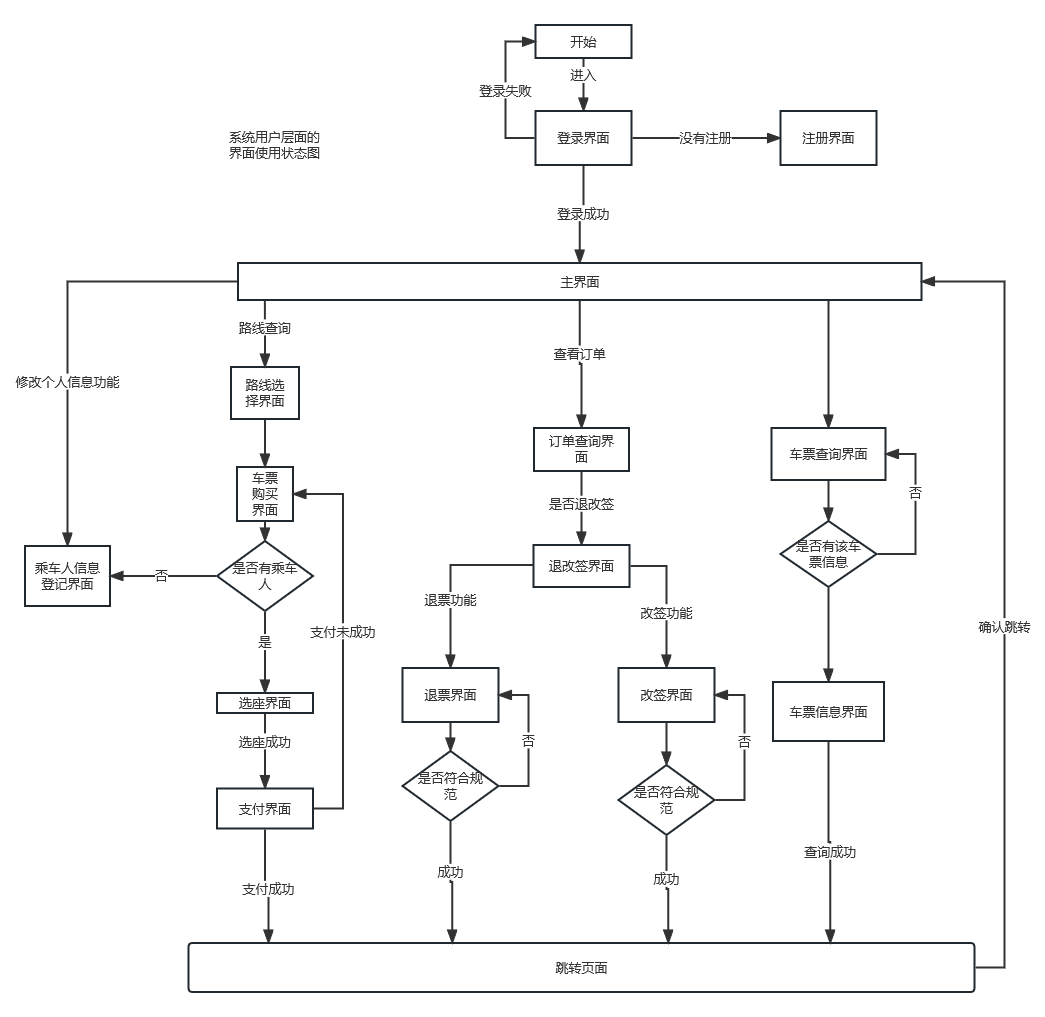
这个Petri网模型描述了一个完整的火车售票系统Web项目的流程，包括查询车次信息、预定座位、支付费用、检票和发车等步骤。

****

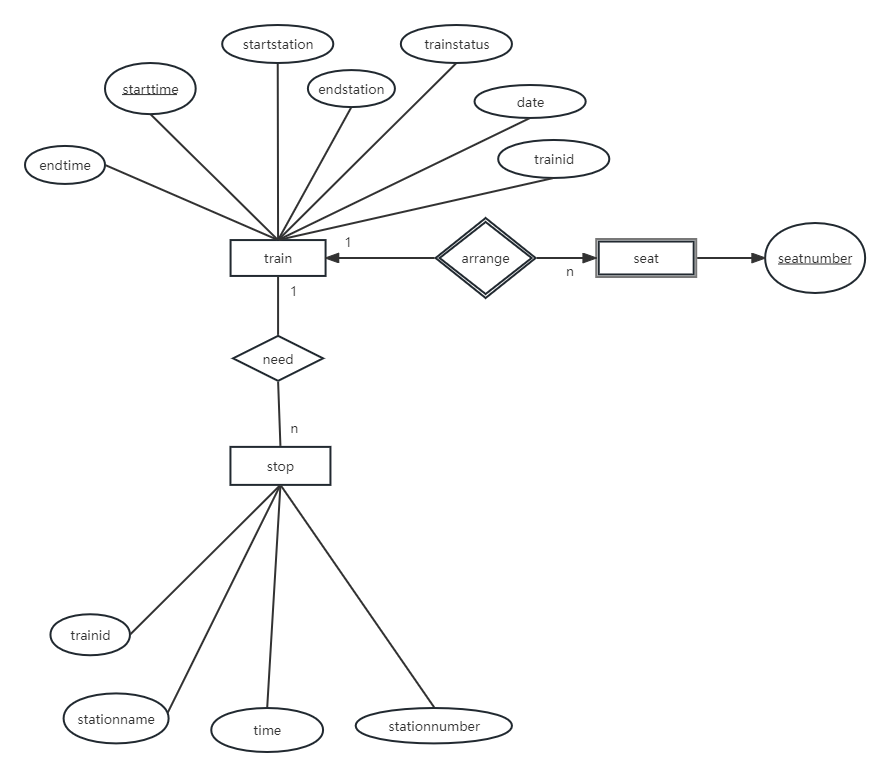
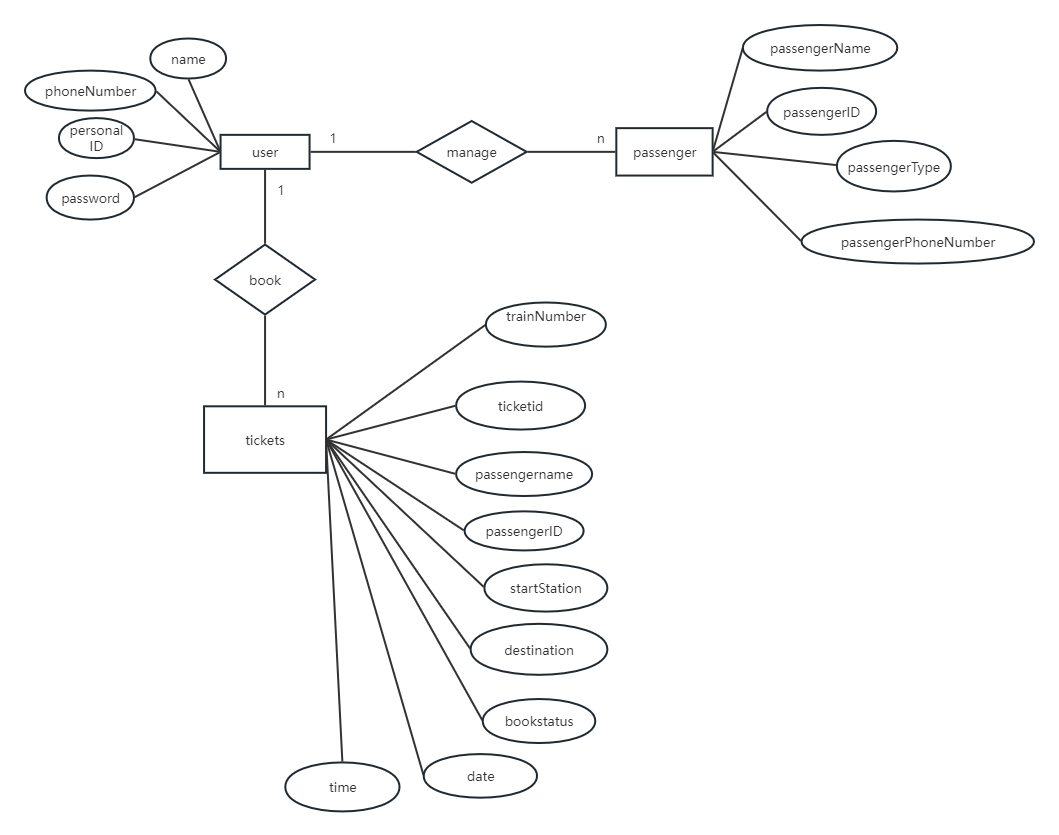
**数据流图：**

****

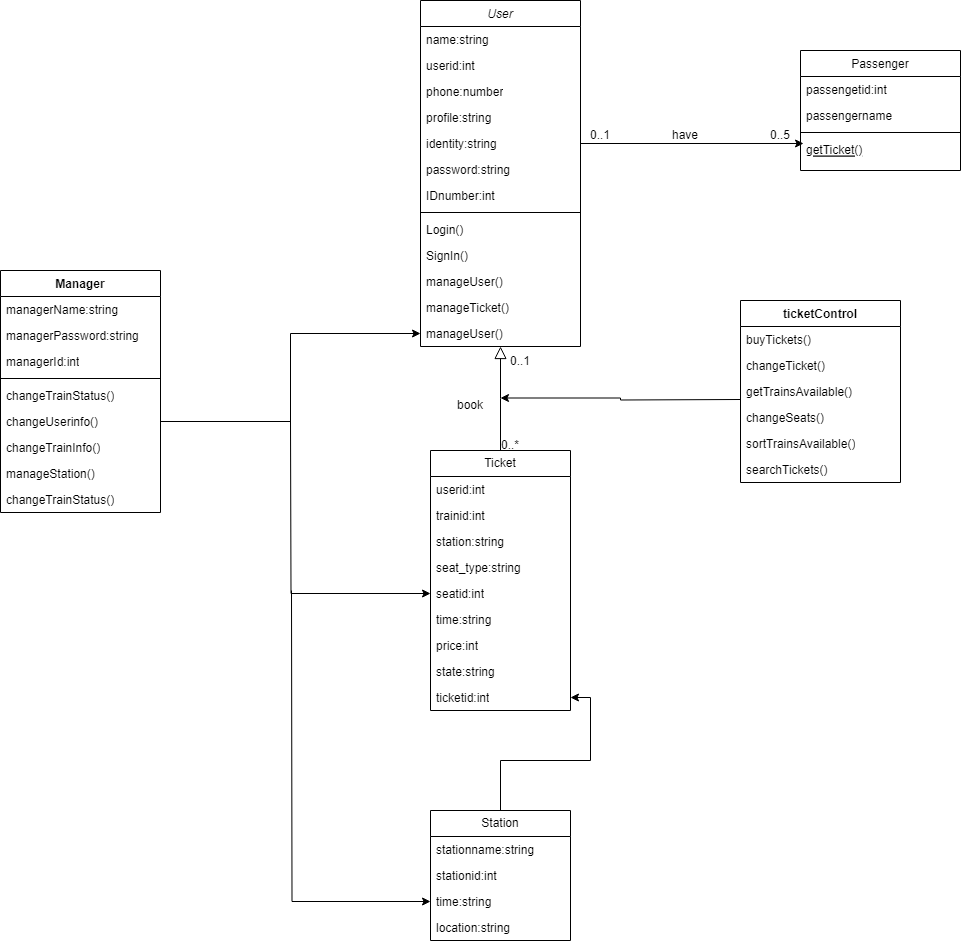
**状态图：**

****

1. **R图：**

****

**UML:**

**解释：该处所有的功能，流程，结构图仅代表当前初期讨论结果，并非最终实际，该软件的大体框架和功能已经决定，后期开发时可能存在细节方面的出入。**

### 3.3.2软件子系统功能/对象结构

该处和3.3.1类似，由于软件功能并不复杂，因此可以将所有的描述建模绘制成一张图，就不用拆分。

### 3.3.3描述约定

在火车售票系统的开发中，通常使用以下约定描述：

1. 数学符号：在火车售票系统的算法和数据处理中，通常会使用一些数学符号进行描述和计算。例如，+ 表示加法，- 表示减法，\* 表示乘法，/ 表示除法等。
2. 度量单位：火车售票系统中会涉及到一些度量单位，例如时间、距离、重量等。常用的时间单位包括秒、分钟、小时等；距离单位包括米、千米等；重量单位包括克、千克等。
3. 数据格式：在火车售票系统中，需要对数据进行格式化，以便于处理和存储。常用的数据格式包括文本格式、数字格式、日期格式等。
4. 代码规范：在火车售票系统的开发中，为了提高代码的可读性和可维护性，通常会采用一些代码规范。例如，采用驼峰命名法、缩进格式、代码注释等。

以上约定描述可以帮助开发人员理解和规范代码，提高系统的可靠性和可维护性。

## 3.4CSCI能力需求

### 3.4.1登录与注册模块

#### 3.4.1.1用户注册

a.说明：

用户注册功能是系统的基本功能，旨在为用户提供个性化的服务和安全的购票体验。该功能将确保只有注册用户才能购买车票。

b.输入：

用户在注册时需要输入以下信息：

(1)姓名：用户的真实姓名，因为火车售票系统需要采用实名制，绑定用户的身份证号码。

(2)电子邮件地址：用户的电子邮箱可用于注册或登录时的身份认证。

(3)用户名：用于登录和显示的名字。

(4)密码：用于保护用户账户的私密信息。

(5)手机号码：用于验证用户身份和接收通知短信等。同时,为了防止恶意用户注册大量账户，手机号码也需要进行验证，例如通过短信验证码等方式进行验证。

c.处理：

用户注册功能需要进行以下处理：

检查输入信息的有效性和合法性:

用户输入的信息需要进行有效性和合法性的检查，例如：

用户名不能包含特殊字符，长度不能超过限制;

密码需要符合一定的复杂度要求，例如长度、大小写字母、数字和特殊字符的组合;

手机号码需要符合一定的格式要求，如手机号码的位数应当是11位且不能是空号；

电子邮件地址需要符合一定的格式要求；

(2)检查输入的手机号码和电子邮件地址是否唯一：

在用户进行注册的时候，会输入手机号码和电子邮件地址作为用户的标识，并进行实名认证，所以需要判断新注册用户输入的电话号码或者电子邮件地址是否已经被使用过，如果已经使用过则不可以再次使用相同的进行注册。

(3)检查用户是否已经注册：

在用户注册时，需要检查用户是否已经注册过，避免重复注册。可以在数据库中查询用户信息进行验证。

(4)检查密码是否符合安全要求：

在用户进行注册时，密码需要符合一定的复杂度要求，例如长度应当不少于8位且不超过16位、应当是大小写字母、数字和特殊字符的组合;

(5)将用户信息保存在数据库中：

注册成功后，需要将用户信息保存在数据库中，包括用户名、密码、手机号码、电子邮件地址等信息。登录成功后，需要将用户登录信息保存在会话中，以便后续的功能调用。

(6)对用户密码进行加密存储：

为了用户的信息安全，我们在数据库中存储用户密码的时候应当使用一定的加密手段进行加密存储，这样即使用户信息不小心泄露，也不会对用户造成实质性的损失。

d.输出：

用户注册功能将输出以下信息：

(1)用户注册成功：系统将返回注册成功的信息，并要求用户进一步完善个人信息。

(2)用户注册失败：系统将返回注册失败的信息，并告知用户注册失败的原因，例如用户名已存在、密码过于简单等。

#### 3.4.1.2用户登录

a.说明

用户登录是系统的基本功能，旨在为用户提供个性化的服务和安全的购票体验。该功能将确保只有已经注册过的用户才能购买车票。

b.输入：

用户登录时需要输入以下信息：

(1)用户名：在注册时设置的用户名。

(2)密码：在注册时设置的密码。需要注意的是，为了保护用户的账户安全，密码应该要求具有一定的复杂度，例如至少包含8个字符，包括大写字母、小写字母、下划线和数字等多种字符。

(3)手机号码：为了防止恶意用户注册大量账户，手机号码也需要进行验证，例如通过短信验证码等方式进行验证。系统会向用户输入的手机号码发送一个验证码来检验用户的身份。

(4)验证码：系统发送给用户的验证码用于验证用户的身份。

c.处理：

用户登录功能需要进行以下处理：

(1)检查输入信息的有效性和合法性：

用户输入的信息需要进行有效性和合法性的检查，例如：

用户名不能包含特殊字符，长度不能超过限制；

密码需要符合一定的复杂度要求，例如长度、大小写字母、数字和特殊字符的组合；

手机号码需要符合一定的格式要求，例如电话号码的长度应该是11位；

(2)检查用户输入的用户名、密码或短信验证码是否正确：

在用户登录时，需要检查用户输入的用户名、密码或者输入的短信验证码是否正确，用户名和密码需要与数据库中的用户信息进行对照，判断用户名和密码是否正确，如果不正确应当返回登录失败。短信验证码应当与系统刚刚发送给用户的短信验证码一致。

(3)检查用户是否已经登录：

在用户登录时，需要检查用户是否已经登录过，如果已经登录则应当提醒用户登录失败，避免重复登录造成用户信息或者系统的混乱。可以在数据库中查询用户登录信息进行验证。

(4)将用户登录信息保存在数据库中：

登录成功后，需要将用户的登录信息保存在数据库中，包括用户名、密码、登录时间、登录地址、登录状态等信息。且登录成功后，需要将用户登录信息保存在会话中，以便后续的功能调用。

(5)返回用户的个人信息并生成且返回一个新的Token或者Session会话：

登录成功后，需要将用户登录信息保存在Session会话中返回给用户或者生成并返回一个新的Token给用户，以便后续的功能调用。

d.输出

用户登录功能将输出以下信息：

(1)用户登录成功或失败的信息：

当用户登录成功时，系统将返回成功的信息，并允许用户进入系统内部。例如：

登录成功！欢迎使用火车售票系统！

当用户登录失败时，系统将返回失败的信息，并告知用户登录失败的原因。例如：

登录失败！用户名或密码错误，请重新输入！

登录失败！该用户已登录！

(2)用户信息：

在登录成功后，系统将返回用户信息，例如：用户名、手机号码、上次登录时间、用户token等，用户信息将被保存在会话中，以便后续的功能调用。

在输出信息时，系统应该尽可能地清晰明了，确保用户可以理解和掌握相应的信息。此外，系统应该对输出信息进行统计和分析，以便监控和优化系统的运行情况。例如，系统可以记录用户登录的次数、成功率和登录时长等信息，并在后续的系统优化中使用这些信息进行决策。

#### 3.4.1.3用户登出

a.说明：

用户在成功登录后，点击页面上的注销登出连接实现登出系统，系统将删除所有与该用户相关的登录信息。

b.输入：

该功能需求没有输入。

c.处理：

在用户点击登出后，系统保存该用户的登录时间信息和登出时间信息后，删除用户登录的Cookies等信息，实现用户再次浏览页面时为未登录状态。

d.输出：

如果登出成功，返回一个导航页面：返回主页。如果登出失败，返回错误提示页面。

### 3.4.2车次查询模块

#### 3.4.2.1直达查询

a.说明：

直达车次查询是火车售票系统中的一个重要功能。该功能可帮助用户快速便捷地查询车次、票价和座位等信息，提高用户购票效率，减少用户购票时间和成本。

车次直达查询功能可在系统主界面上显示，用户可以通过输入起始站点、终点站点和出发日期等查询条件，快速查询并购买直达车次的火车票。该功能还可以提供车票余量、车票价格等相关信息供用户选择。

b.输入：

进行直达车次查询需要输入以下信息：

用户输入：用户需要输入起始站点、终点站点和出发日期等查询条件。

接口控制文件：该功能需要使用站点列表、车次列表、车次详细信息等接口控制文件。

c.处理：

直达车此查询功能需要进行以下处理：

(1)输入数据的有效性检查：系统需要检查用户输入的查询条件是否合法，例如是否存在该站点、是否存在该日期等。

(2)操作顺序：系统需要按照用户输入的查询条件，查询符合条件的车次信息。

(3)异常情况的响应：系统需要对可能出现的异常情况进行处理，例如网络通信故障、系统故障等，避免影响用户正常查询。

(4)受操作影响的参数：该功能可能会影响车票余量、车票价格等相关信息。

(5)输入转换：系统需要将用户输入的查询条件转换为可供查询的数据格式。

(6)输出数据的有效性检查：系统需要检查查询结果的有效性，例如查询结果是否为空、是否符合查询条件等。

d.输出：

直达车此查询功能将输出以下信息：

查询结果：系统需要将符合查询条件的车次信息输出给用户，包括车次号、车次类型、发车时间、到达时间、剩余票量、票价、剩余座位信息等。

接口控制文件：该功能需要将查询结果保存到车次列表、车次详细信息等接口控制文件中。

#### 3.4.2.2换乘查询

a.说明：

换乘车次查询功能是指用户输入起始站点、终点站点和出发日期等信息，系统能够根据用户需求，查询出一条或多条合适的换乘路线，同时提供每条路线中涉及的所有车次信息，以方便用户选择最优换乘方案。

b.输入：

进行换乘车次查询时用户需要输入以下信息：

(1)用户需要提供起始站点、终点站点详细地址或者站点名称。

(2)对于站点名称，系统需要支持模糊查询，以便用户能够找到所需的站点。

(3)对于起点和终点之间的距离，需要有一定的限制范围，以便保证查询结果的准确性。

(4)用户需要指定出发日期、出发时间和到达时间，系统需要能够按照用户需求进行查询。

(5)对于查询结果的数量，需要有一定的限制，以便避免返回过多的信息，影响查询效率。

c.处理：

换乘车次查询功能需要进行以下处理：

(1)检查输入的站点是否存在。

(2)检查输入的日期是否有效。

(3)查询车次、票价和座位信息。

(4)系统需要根据用户提供的起点和终点信息，查询出所有可能的直达路线和换乘路线。

(5)系统需要根据用户指定的出发时间和到达时间，筛选出符合时间要求的路线。

(6)系统需要计算出每条路线的总距离、总时间以及需要换乘的次数。

(7)对于每个车站的换乘，系统需要查询并显示换乘站点和换乘时间，并计算换乘所需要的时间。

(8)系统需要对查询结果进行排序，以便用户能够快速找到最优的换乘方案。

d.输出：

换乘车次查询功能将输出以下信息：

(1)系统需要将所有符合用户需求的换乘路线展示给用户，包括每个路线的起点、终点、总距离、总时间、换乘站点和时间等详细信息。

(2)系统需要能够将查询结果按照距离、时间、价格等因素进行排序，并给出最优换乘方案的推荐。

(3)对于每个车站的换乘，系统需要在查询结果中标注换乘站点和换乘时间，以便用户能够更加方便地了解换乘的细节。

(4)对于查询结果的数量，需要有一定的限制，以便用户能够快速找到最优的换乘方案，同时不影响查询效率。

e.异常处理：

(1)对于输入信息不全或者不合法的情况，系统需要给出相应的提示，并要求用户重新输入。

(2)对于无法查询到符合用户需求的换乘路线的情况，系统需要给出相应的提示，同时推荐相关的直达路线。

(3)对于网络故障或者系统故障的情况，系统需要给出相应的提示，并尽快恢复网络故障和系统故障。

### 3.4.3票务信息模块

#### 3.4.3.1余票查询

a.说明：

余票查询功能的目的是让管理员能够查询某一列车的余票信息。管理员可以通过输入车次号、起点站、终点站、出发时间、到达时间等信息来查询该车次的余票情况。该功能能够准确地查询到该车次的余票数量，帮助管理员更好地管理车票销售和库存，提高系统的服务质量。

b.输入：

该功能的输入包括车次号、起点站、终点站、出发时间、到达时间等信息。其中，起点站和终点站需要用户手动输入，而出发时间和到达时间可以从系统中读取。管理员还可以设置其他查询条件，如列车类型、座位等级、余票数量等。查询条件需要在系统中进行验证，确保输入的数据正确有效。

c.处理：

余票查询功能的处理包括以下内容：

(1)根据输入的查询条件，在车票库存中查询符合条件的车票信息。

(2)对查询结果进行排序，方便用户选择。

(3)统计查询结果中的余票数量，并将结果返回给管理员。

(4)对查询结果进行排序，确保管理员能够看到符合条件的车票信息。

(5)在查询过程中需要对输入的数据进行验证，以确保数据的准确性和有效性。

(6)如果查询不到符合条件的车票，则向管理员返回相应的提示信息。

d.输出：

余票查询功能的输出主要包括查询到的车票余量信息。管理员可以通过系统返回的信息了解到该车次的余票数量，以便更好地管理车票销售和库存。系统返回的信息应包括车次号、出发时间、到达时间、座位等级、余票数量等。在输出结果时，需要确保结果的准确性和有效性。查询结果为空时，给出相应提示。

e.异常情况：

在进行余票查询时，可能会出现查询结果为空、查询条件无效、查询超时等异常情况。针对这些异常情况，系统需要给出相应的提示信息，让管理员能够更好地理解查询失败的原因。在查询超时时，系统应该提示管理员重新尝试查询。

#### 3.4.3.2购买车票

a.说明：

购买车票功能的目标是让用户能够方便地购买到合适的车票，以满足用户的出行需求。用户可以根据自己的出行计划，在系统中查询到符合条件的车次信息，并在系统中购买到对应的车票。

b.输入：

购买车票时用户需要输入以下信息：

输入数据：

车次信息：用户选择的车次号、座位信息；

乘车人信息：用户需要输入乘车人的姓名、身份证号码等信息，以便系统能够生成相应的车票，并进行乘车人身份验证。

支付方式：用户需要选择支付方式，并输入相应的支付信息，以便系统能够完成支付流程。

接口说明：

在购买车票功能中，需要调用以下接口：

车次查询接口：用于查询符合条件的车次信息。

乘车人身份验证接口：用于验证乘车人身份信息。

支付接口：用于完成支付流程。

c.处理：

购票功能需要进行以下处理：

(1)输入数据的有效性检查

在购买车票功能中，需要对输入数据进行有效性检查，以确保输入的数据符合要求。例如，需要检查乘车人的身份证号码是否合法，车票数量是否超过了可售数量等。

(2)操作顺序和时间设定

在购买车票功能中，需要根据用户的输入数据，按照一定的顺序和时间设定进行相应的操作。例如，需要先查询符合条件的车次信息，再进行乘车人身份验证，最后进行支付流程。

(3)选择支付方式

系统需要支持多种支付方式，并允许用户选择自己喜欢的支付方式。

(4)异常情况的响应

在购买车票功能中，需要对异常情况进行响应。例如，当车票数量不足时，需要提示用户购票失败，并返回可售车票数量。

(5)受操作影响的参数

在购买车票功能中，需要记录和更新受操作影响的参数，例如车票数量、车次信息等。

(6)用于把输入转换成相应输出的方法

在购买车票功能中，需要对输入数据进行处理，并将其转换为相应的输出数据。

d.输出：

(1) 购票成功页面

如果用户购票成功，系统需要返回购票成功的页面，并展示订单信息和支付方式等相关信息。

(2)购票失败页面

如果用户购票失败，系统需要返回购票失败的页面，并提示用户错误信息，例如余票不足、输入信息错误等。

e.异常条件：

(1)如果用户输入信息不合法，系统需要返回错误信息并提示用户重新输入；

(2)如果支付失败，系统需要返回错误信息并提示用户更换支付方式或稍后再试；

(3)如果系统出现故障，需要及时通知管理员并处理问题。

### 3.4.4订单管理模块

#### 3.4.4.1改签车票

a.说明：

改签车票是指旅客在原定车次出发前，按照规定的流程将原购车票作废，再重新购买新的车票的服务。改签车票是为了满足旅客的灵活出行需求，方便旅客更改出行计划而设计的服务。旅客可以通过网站、手机APP、自助售票机等多种渠道进行改签操作，方便快捷。

b.输入：

改签车票需要输入以下数据：

1.原定车票信息：包括原始订单号、原始车次、座位号等信息。

2.新车次信息：包括新的订单号、出发时间、车次、座位类型等信息。

支付信息：包括支付方式、支付金额等信息。

c.处理：

改签车票需要进行以下操作：

1.检查原始车票是否可以改签：系统需要检查原始车票是否符合规定的改签条件，如改签时间、车票类型等。

2.计算差价：如果新的车票价格高于原票价，需要计算差价并提示旅客支付差价。

3.修改订单信息：修改旅客订单信息，包括车次、座位等信息。

4.生成新车票：系统生成新的车票信息，包括新的订单号、车票号、座位号等。

5.取消原车票：系统取消旅客原有的车票信息，并将原车票退款到旅客的支付账户中。

d.输出：

改签车票需要输出以下数据：

1.新车票信息：包括新的订单号、车票号、座位号等信息。

2.支付信息：包括支付方式、支付金额等信息。

3.退款信息：如果新车票价格低于原票价，需要将差价退还到旅客的支付账户中。

4.错误信息：如果操作失败，需要返回错误信息给旅客，如改签时间不符合规定、车票已经超过改签期限等

#### 3.4.4.2取消订单

a.说明：

取消购买车票是指用户在购买车票后，需要对已购买的车票进行取消操作。该功能的目标是让用户能够方便、快捷地取消购买的车票，同时确保系统数据的准确性。取消购买车票是火车售票系统中常见的需求，用户可能会因为各种原因需要取消已经购买的车票。

b.输入：

进行取消订单操作时用户需要输入以下信息：

用户输入：用户需要提供自己的身份信息，以及已购车票的相关信息，包括车次、座位类型、出发时间等。

接口输入：取消购买车票功能需要调用其他模块的接口，例如支付模块的退款接口。

c.处理：

取消购买车票功能的处理过程包括以下内容：

(1)身份验证：系统需要验证用户的身份信息，确保用户有权限取消该车票。

(2)信息检验：检查用户是否已经登录，检查订单状态是否为未出行状态，将订单状态更改为已取消状态，将座位释放为可用状态。

(3)取消操作：系统需要将已购车票从订单列表中删除，并将相应座位释放出来供其他用户购买。

(4)退款操作：如果用户已经支付了该车票的费用，系统需要调用支付模块的退款接口，将费用返还给用户。

(5)操作日志：系统需要记录该操作的详细信息，包括取消时间、取消原因、用户身份信息等，以便后续查询和统计。

d.输出：

取消购买车票功能的输出包括以下内容：

(1)用户提示：系统需要向用户提示取消操作是否成功，并给出相应的提示信息。

(2)日志输出：系统需要将操作日志记录到系统日志中，以便后续查询和统计。

(3)接口输出：如果需要调用其他模块的接口，系统需要将接口输出信息记录到相应的日志中，以便后续排查问题。

e. 异常情况处理

在取消车票的过程中，可能会出现以下错误情况：

(1)输入的订单号不存在，订单状态不允许取消，系统故障导致取消失败。对于这些错误情况，系统应该给出相应的错误提示信息，并引导用户进行相应的操作。

(2)身份验证失败：如果用户提供的身份信息验证失败，系统需要向用户提示身份验证失败并终止操作。

(3)订单不存在：如果用户输入的车票信息不存在于订单列表中，系统需要向用户提示该车票不存在并终止操作。

(4)取消失败：如果取消操作失败，例如该车票已经被改签或退票，系统需要向用户提示取消失败并给出相应的提示信息。

退款失败：如果退款操作失败，例如支付模块的退款接口返回错误，系统需要向用户提示退款失败并给出相应的提示信息。

#### 3.4.4.3查看订单

a.说明：

查看订单是火车售票系统的一项基本功能，它允许用户查询其已购买的车票订单信息，包括订单号、乘车人信息、车次信息、座位信息、出发时间、票价等。用户可以通过查看订单功能方便快捷地了解自己已购买的车票信息，方便用户做好出行准备。

b.输入：

查询已订车票时需要输入查询条件，例如出发日期、订单状态等。

(1)订单号：用户输入所查询订单的订单号。

(2)乘车人信息：为了确保查询的订单信息是用户指定乘车人的，系统需要用户输入其售票时所填写的乘车人信息。

(3)时间范围：用户可以选择查看订单的时间范围，例如最近一个月的订单、近三个月的订单等。

其他筛选条件：如出发地、目的地、乘车日期等。

c.处理：

查询已订车票时需要根据查询条件查询订单信息，并按照时间顺序排序。

(1)验证乘车人和订单号的合法性：系统会校验输入的乘车人ID和订单号是否合法，如果不合法，系统会提示用户重新输入。

(2)查询订单信息：系统通过订单号和乘车人ID在数据库中查询该订单的详细信息，包括订单号、乘车人信息、车次信息、座位信息、出发时间、票价等。

(3)处理筛选条件：如果用户选择了时间范围或者其他筛选条件，系统会根据用户选择的条件对订单信息进行筛选处理，只显示符合条件的订单信息。

异常情况处理：如果查询出错或者没有查到订单信息，系统会提示用户相应的错误信息，并给出解决方法。

d.输出：

该功能将输出以下信息：用户的订单记录及其详细信息。

(1)订单信息：系统会将查询到的订单信息以列表或表格的形式展示给用户，包括订单号、乘车人信息、车次信息、座位信息、出发时间、票价等。

(2)接口说明或接口控制文件的参考资料：为了方便用户了解订单信息的详细内容，系统还会提供相关接口说明或接口控制文件的参考资料。

e. 错误处理

在查询订单的过程中，可能会出现以下错误情况：

用户未登录 查询结果为空。对于这些错误情况，系统应该给出相应的错误提示信息，并引导用户进行相应的操作。

### 3.4.5个人信息管理模块

#### 3.4.5.1乘车人信息修改

a.说明：

乘客的个人信息是火车售票系统中非常重要的一部分，为了保证乘客信息的准确性和及时性，火车售票系统需要提供修改乘车人信息的功能。该需求的目的是为了让用户能够在系统中方便地修改乘车人信息，以便后续购票操作的顺利进行。

b.输入：

修改乘车人信息

(1) 用户需选择要修改的乘车人，并输入要修改的信息。

(2) 对于证件号码，系统需检查其格式是否符合规范，例如身份证号码应为18位，且需要进行合法性校验。

(3) 用户需输入正确的联系电话，以便系统能够及时联系到乘客。

c.处理：

修改乘车人信息

(1)系统需要对用户输入的修改信息进行校验，确保输入的信息格式正确，信息完整。

(2)操作的顺序：用户先登录系统，进入乘车人管理页面，选择添加新乘车人，填写基本信息和特殊需求，提交保存。

(3)异常情况的响应：当输入数据有误时，系统应该提示用户重新填写并提供错误信息，确保数据的正确性。

(4)受操作影响的参数：添加乘车人后，用户可以在订票时选择已添加的乘车人信息，提高了订票的效率。

(5)用于把输入转换成相应输出的方法：用户填写乘车人信息后，系统将保存该信息并生成一个唯一的乘车人ID，方便用户选择该乘车人信息进行订票。

(6)数据库更新：系统需要更新数据库中对应乘车人的信息，确保信息能够及时准确地更新。

d.输出：

修改乘车人信息

(1) 系统需要提示用户修改乘车人信息成功，并显示修改后的乘车人信息。

(2)用户可以在系统中查看已修改的乘车人信息。

e.异常情况处理：

修改乘车人信息

(1) 当用户输入的证件号码格式不正确或者证件号码已存在时，系统需提示用户重新输入正确的证件号码。

(2) 当用户输入的信息不完整或者格式不正确时，系统需提示用户输入正确的信息。

#### 3.4.5.2添加乘车人

a.说明：

添加乘车人是一个重要的功能，目标是方便用户管理和预订火车票时选择已添加的乘车人信息，提高用户体验，减少用户填写信息的工作量，同时提高信息的准确性和安全性。

b.输入：

添加乘车人功能的输入包括以下信息：

(1) 用户需输入乘车人的姓名、性别、证件类型、证件号码、乘客类型、联系电话等基本信息。

(2) 对于证件号码，系统需检查其格式是否符合规范，例如身份证号码应为18位，且需要进行合法性校验。

(3) 对于乘客类型，系统需提供成人、儿童、学生等选项，用户需选择相应的类型。

乘车人的特殊需求，如带小孩、残疾人、老年人等特殊群体的需求。

乘车人的订票信息，如订票时间、车次、座位等级等。

c.处理：

(1)输入数据的有效性检查：对输入的数据进行格式检查，确保数据的正确性和完整性。

(2)操作的顺序：用户先登录系统，进入乘车人管理页面，选择添加新乘车人，填写基本信息和特殊需求，提交保存。

(3)异常情况的响应：当输入数据有误时，系统应该提示用户重新填写并提供错误信息，确保数据的正确性。

(4)受操作影响的参数：添加乘车人后，用户可以在订票时选择已添加的乘车人信息，提高了订票的效率。

(5)用于把输入转换成相应输出的方法：用户填写乘车人信息后，系统将保存该信息并生成一个唯一的乘车人ID，方便用户选择该乘车人信息进行订票。

(6)输出数据的有效性检查：系统保存乘车人信息后，应对数据进行有效性检查，确保数据的正确性和完整性。

d.输出：

(1)系统需要提示用户添加乘车人信息成功，并显示添加的乘车人信息。

(2)用户可以在乘车人管理页面查看已添加的乘车人信息。

(3)添加乘车人后，用户可以在订票时选择已添加的乘车人信息，提高了订票的效率。

(4)用户可以在订票页面选择已添加的乘车人信息进行订票。

e.安全性：

为了保证用户信息的安全性，系统应该采取以下措施：

(1)数据加密：对用户输入的敏感信息进行加密处理，确保用户信息的安全。

(2)用户权限控制：对用户进行身份认证，只有认证通过的用户才能访问和管理自己的乘车人信息。

(3)安全审核：系统应定期对用户信息进行安全审核，确保用户信息不被泄露和非法使用。

### 3.4.6车次管理模块

#### 3.4.6.1车次信息修改

a.说明：

该功能的目的是修改现有车次的信息。

b.输入：

(1)必须输入要修改的车次号。

(2)系统应该能够根据车次号自动检索出车次的信息，然后可以对需要修改的信息进行修改。

(3)对于输入数据的有效性和合法性进行检查。

c.处理：

(1)根据输入的车次号，获取要修改的车次信息。

(2)对于输入数据不合法的情况，需要进行错误处理并向管理员显示错误信息。

(3)修改车次信息后，系统应该自动将新的信息保存到数据库中。

d.输出：

系统应该能够输出修改后的车次信息，以供管理员查看和核对。

#### 3.4.6.2添加车次信息

a.说明：

该功能的目的是添加新的车次信息。

b.输入：

(1)必须输入车次号、出发地、目的地、出发时间、到达时间、列车类型、列车座位数、车票价格等基本信息。

(2)对输入数据的有效性和合法性进行检查。

(3)系统应该可以支持通过文件上传和手动输入两种方式添加车次信息。

c.处理：

(1)将输入的车次信息与现有的车次信息进行比较，以确保新添加的车次不会与现有车次冲突。

(2)对于输入数据不合法的情况，需要进行错误处理并向管理员显示错误信息。

d.输出：

系统应该能够输出新添加的车次信息，以供管理员查看和核对。

#### 3.4.6.3删除车次信息

a.说明：

该功能的目的是删除现有的车次信息。

b.输入：

(1)必须输入要删除的车次号。

(2)系统应该能够根据车次号自动检索出车次的信息。

c.处理：

(1)根据输入的车次号，获取要删除的车次信息。

(2)系统应该提示管理员确认删除操作，并在管理员确认后将车次信息从数据库中删除。

d.输出：

系统应该输出删除成功的提示信息。

### 3.4.7用户管理模块

#### 3.4.7.1查看注册用户名单

a.说明：

该功能的目的是提供管理员查看已注册用户的列表。

b.输入：

用户名单查询条件，例如按照注册时间、用户等级、用户名等条件进行查询。

c.处理：

(1)接收管理员登录信息并验证身份。

(2)接收并验证用户名单查询条件。

(3)根据查询条件从用户数据库中获取用户列表。

(4)根据查询条件对用户列表进行排序和过滤。

(5)将查询结果展示给管理员。

d.输出：

满足查询条件的用户列表，包括用户名、注册时间、用户等级等信息。

若查询结果为空，则返回相应提示信息。

若查询条件无效，则返回相应提示信息。

e.异常处理：

若管理员登录信息无效，则返回相应提示信息。

若查询条件无效，则返回相应提示信息。

### 3.4.8数据可视化模块

#### 3.4.8.1售票情况展示

a.说明：

售票情况展示的目标是为管理员提供实时的、直观的售票情况数据，帮助其更好地管理售票系统。背景是在日常运营中，售票情况随时都会发生变化，管理员需要能够随时了解售票情况以便及时做出调整。

b.输入：

1）已售出的票数，剩余的票数，销售额等相关数据，这些数据需要由售票系统实时更新并传输给展示界面。

2）数据的度量单位应该明确，比如票数的单位可以是张，销售额的单位可以是元等。

3）应该指明数据的有效输入范围，比如售票数量不能超过总票数。

c.处理：

1）对输入数据的有效性检查，如售票数量不能超过总票数，输入数据必须符合规定的度量单位等。

2）确定数据的处理方式和时间设定，比如售票数量的变化需要及时更新到展示界面。

3）应对异常情况做出响应，比如数据传输异常等，需要有相应的错误处理措施。

4）展示界面应该能够受到操作的影响，比如管理员的查看操作会导致相应的数据更新。

5）需要选择合适的方法将输入数据转换为相应的图表输出，如饼状图、柱状图等。

6）输出数据的有效性检查，比如图表的准确性和可读性。

d.输出：

1）详细说明展示的所有输出数据，包括图表展示的内容、度量单位、时间关系等。

2）有关接口说明或接口控制文件的参考资料。

展示的图表应该具备以下要求：

1）易于理解：图表展示应该是清晰、易懂的，让管理员可以迅速地了解售票情况。

2）准确性：图表展示的数据应该是准确的，管理员应该能够信任这些数据。

3）实时性：展示的数据应该是实时的，可以随时更新，反映当前的售票情况。

4）互动性：展示界面应该具有一定的互动性，管理员可以进行交互式操作，比如放大、缩小、移动等

5）美观性：展示界面应当较为美观，给用户带来较好得用户体验。

## 3.5 CSCI外部接口需求

a.用户接口

用户接口是指与系统进行交互的所有手段和方式，包括系统的操作界面、菜单、输入/输出数据格式等。在设计用户接口时，需要考虑用户的使用习惯和操作习惯，使得用户可以方便、快捷地完成所需操作，从而提高系统的易用性和用户满意度。

本系统的用户接口要求符合现代化的UI设计理念，具有良好的可用性和可访问性。以下是本系统的用户接口需求：

1.操作界面：本系统应提供一个友好的操作界面，包括主界面和子界面。主界面应包括以下模块：车票查询、车票购买、订单管理、个人信息、退出登录等。子界面应根据用户的操作需要提供相应的功能模块。

火车售票系统至少包含一下子界面：

(1)登录界面：用户可以在登录界面输入用户名和密码以登录系统。

(2)车票查询界面：用户可以在车票查询界面选择查询条件，并查询满足条件的车票信息。

(3)车票购买界面：用户可以在车票购买界面选择座位类型和数量，并进行车票购买操作。

(4)订单管理界面：用户可以在订单管理界面查看已预订的车票订单，并进行取消订单操作。

(5)支付界面：用户可以在支付界面选择支付方式，并完成车票的支付操作。

(6)售后服务界面：用户可以在售后服务界面进行更改车票信息、取消订单等操作。

2.提供菜单：本系统应提供简单明了的菜单，使用户可以方便地进行各项操作。

3.输入/输出数据格式：本系统应支持各种数据格式的输入和输出，如文本、图片、音频等。

4.响应时间：本系统应具有快速响应的特性，能够在较短时间内处理用户的操作请求，并返回相应的结果。

5.吞吐时间：本系统应具有高吞吐量，能够同时处理多个用户的请求，保证系统的高效性和可靠性。

6.容量：本系统应支持大容量的数据处理，能够处理大量的用户数据和系统日志等。

7.容错处理：本系统应具有良好的容错处理能力，能够自动检测和纠正用户输入的错误信息，避免对系统的影响。

b.硬件接口：

硬件接口是指系统与外部硬件设备进行通信和交互的接口。：

火车售票系统应该能够运行在常见的计算机硬件上，本系统需要满足以下硬件接口需求：

1.至少需要4GB内存和500GB硬盘存储空间；

2.支持至少1024\*768分辨

3.必须具有至少一个网络接口，以便与火车售票系统服务器进行通信；

4.支持USB和HDMI接口，以便连接打印机和显示器；

5.支持硬件加密和解密，以保护系统的安全性；

6.支持多种操作系统，例如Windows，Linux和MacOS等。

7.服务器：本系统应运行在一台或多台服务器上，需要具备足够的存储空间、内存、带宽等资源。

8.数据库：本系统应使用可靠的数据库管理系统，如MySQL、Oracle等，以确保数据的安全性。

9.打印机：本系统需要支持打印机，以方便用户打印车票等相关信息。

此外，为了满足客户需求，可以考虑提供以下硬件功能：

1.支持NFC芯片，以便顾客可以通过触碰芯片来快速地获取信息；

2.支持语音输入和输出，以便顾客可以使用语音交互的方式进行购票；

3.支持触摸屏，以便顾客可以使用手指进行交互操作；

4.支持条形码和二维码扫描仪，以便顾客可以使用手机进行扫码购票；

5.支持打印机自动售票功能，以便顾客可以直接在售票机上打印车票并付款。

总的来说，硬件接口的要求与顾客需求密切相关，需要考虑到顾客使用习惯和便利性。因此，在设计火车售票系统时，需要综合考虑硬件性能、可扩展性、兼容性和易用性等方面的因素。

在硬件接口方面，系统需要能够与各种不同类型的传感器和执行器进行通信。这些传感器和执行器包括但不限于温度传感器、压力传感器、加速度传感器、电机、阀门等。

为了与这些传感器和执行器进行通信，系统需要提供多种接口，包括模拟输入接口、数字输入接口、模拟输出接口和数字输出接口。模拟输入接口用于读取模拟传感器数据，数字输入接口用于读取数字传感器数据，模拟输出接口用于控制模拟执行器，数字输出接口用于控制数字执行器。

此外，系统还需要提供通信接口，以便与其他设备进行通信。通信接口可以是以太网接口、RS232接口、RS485接口、CAN总线接口、USB接口等。

最后，在设计硬件接口时，还需要考虑到电源和地线的布局和连接方式，以确保系统的稳定性和可靠性。

c. 软件接口需求

软件接口是指系统与其他软件系统进行通信和交互的接口。本系统需要满足以下软件接口需求：

1.Web服务接口：本系统需要使用Web服务接口进行与其他系统的交互，包括查询车票信息、订票、支付、退票等功能。

2.支付接口：本系统需要与第三方支付平台进行接口对接，以支持在线支付功能。

3.数据库接口：本系统需要与数据库进行交互，包括存储和获取用户信息、车票信息、订单信息等。因此，本系统需要满足与数据库系统的接口需求，包括连接数据库、执行SQL语句等。

4.系统接口：本系统需要与操作系统进行交互，以获取系统资源和执行操作。具体包括获取系统时间、检测网络连接状态、创建和终止进程等。因此，本系统需要满足与操作系统的接口需求，包括调用系统API、使用系统库等。

5.短信接口：本系统需要使用短信接口进行用户验证和通知，例如注册时的手机验证码、订单确认等。因此，本系统需要满足与短信接口的接口需求，包括发送短信、接收短信等。

6.邮件接口：本系统需要使用邮件接口进行用户通知，例如密码重置、订单确认等。因此，本系统需要满足与邮件接口的接口需求，包括发送邮件、接收邮件等。

7.文件接口：本系统需要进行文件读写操作，例如存储和获取用户上传的头像、保存系统日志等。因此，本系统需要满足与文件系统的接口需求，包括创建文件、读写文件等。

8.网络接口：本系统需要使用网络接口进行通信，例如与其他系统进行数据交互、获取火车票实时信息等。因此，本系统需要满足与网络接口的接口需求，包括发送和接收网络数据包、设置网络参数等。

以上是本系统的软件接口需求，系统需要满足各种接口的要求，以保证系统的稳定性、可靠性和安全性。

d.通信接口需求：

通信接口是指系统与外部设备或其他系统进行通信和交互的接口。本系统应满足以下通信接口需求：

1.网络接口：本系统需要通过网络接口实现与其他系统进行通信和数据交换，包括查询车票信息、订票、支付、退票等功能。网络接口需支持HTTP/HTTPS协议，能够保证数据的安全传输和稳定性。

2.短信接口：本系统需要通过短信接口实现与用户的信息互动，例如订票成功后发送短信通知用户取票信息等。短信接口需支持国内主流运营商，能够保证短信发送的稳定性和及时性。

3.邮件接口：本系统需要通过邮件接口实现与用户的信息互动，例如订票成功后发送邮件通知用户取票信息等。邮件接口需支持常用邮件协议，能够保证邮件发送的稳定性和及时性。

4.与第三方支付平台的接口：本系统需要接入第三方支付平台，与其进行支付信息的交换，需要支持HTTP或HTTPS通信协议，使用POST或GET方法发送支付请求并接收响应。

5.与列车信息系统的接口：本系统需要与列车信息系统进行交互，获取列车的实时运行情况以及票价信息等，需要支持HTTP或HTTPS通信协议，使用GET方法获取信息。

6.与用户手机客户端的接口：本系统需要提供手机客户端的接口，与客户端进行通信，需要支持HTTP或HTTPS通信协议，使用POST或GET方法发送请求并接收响应。

7.接口安全性：为确保通信接口的安全性，本系统需要支持数据加密和身份认证等功能。同时，系统需要对非法访问和攻击进行有效的防范和响应。

### 3.5.1接口标识和接口图

本系统与以下实体存在接口关系：

1.Web服务接口：用户通过Web页面或手机客户端与火车售票系统进行交互，包括查询车票信息、订票、支付、退票等功能。

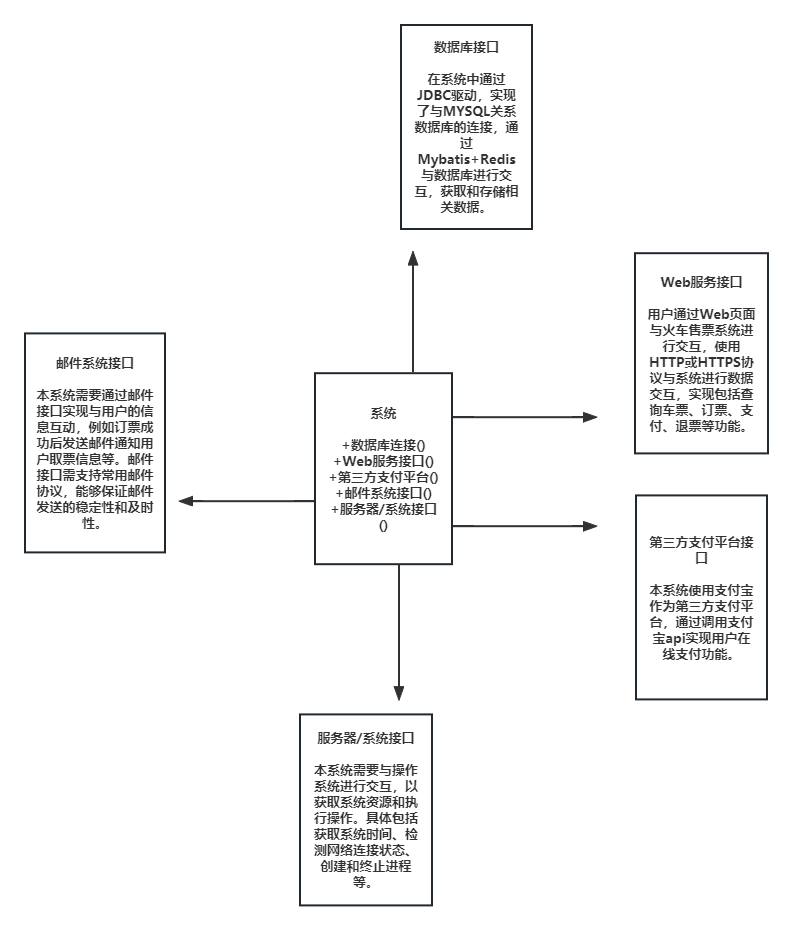
2.第三方支付平台：支付交易数据的传输。本系统需要与第三方支付平台进行接口对接，以支持在线支付功能。

3.数据库接口：本系统需要与数据库进行交互，包括存储和获取用户信息、车票信息、订单信息等。因此，本系统需要满足与数据库系统的接口需求，包括连接数据库、执行SQL语句等。

4.邮件系统接口：本系统需要通过邮件接口实现与用户的信息互动，例如订票成功后发送邮件通知用户取票信息等。邮件接口需支持常用邮件协议，能够保证邮件发送的稳定性和及时性。

5.服务器/系统接口：本系统需要与操作系统进行交互，以获取系统资源和执行操作。具体包括获取系统时间、检测网络连接状态、创建和终止进程等。因此，本系统需要满足与操作系统的接口需求，包括调用系统API、使用系统库等。

接口图如下：



其中，用户通过Web页面或手机客户端与本系统进行交互，本系统通过HTTP或HTTPS协议与第三方支付平台进行支付交易数据的传输，通过HTTP或HTTPS协议与列车信息系统进行列车运行情况和票价数据的传输。

## 3.6CSCI内部接口需求

CSCI内部接口需求是指在CSCI内部，不同模块或组件之间的交互和通信需求。这些模块或组件可以是由同一团队或不同团队编写的，它们需要通过接口来实现互操作。这些接口可以是函数调用、消息传递、共享内存等方式。

在设计CSCI内部接口时，应该考虑以下几个方面的需求：

1.数据共享需求：CSCI内部的不同模块或组件可能需要共享数据，这就需要设计数据共享的接口。这些接口需要定义数据的格式、访问方式和访问控制策略等。

2.模块间通信需求：不同模块或组件之间可能需要进行通信，这需要设计相应的消息传递接口。这些接口需要定义消息的格式、传递方式和处理策略等。

3.函数调用需求：CSCI内部的不同模块或组件可能需要调用对方的函数，这需要设计相应的函数调用接口。这些接口需要定义函数的参数和返回值、调用方式和错误处理策略等。

4.共享资源需求：CSCI内部的不同模块或组件可能需要共享某些资源，比如共享内存、共享文件等。这需要设计相应的共享资源接口，这些接口需要定义资源的访问方式和访问控制策略等。

5.错误处理需求：CSCI内部的不同模块或组件之间可能会发生错误，需要设计相应的错误处理接口。这些接口需要定义错误的类型、处理方式和错误信息等。

在设计CSCI内部接口时，需要考虑接口的可扩展性和可维护性，使得系统可以容易地进行扩展和维护。同时，需要遵循相应的编码标准和规范，确保接口的可靠性和稳定性。

总之，CSCI内部接口的设计是整个系统设计中非常重要的一环，良好的接口设计能够提高系统的可靠性、可维护性和可扩展性。

本系统所有具体的内部接口都留待设计时决定。详细内容可参考《详细设计》文档。

## 3.7CSCI内部数据需求

CSCI内部数据需求是指对于系统内部的数据，包括数据库和数据文件，所需的需求。这些需求可能影响到系统的性能、可靠性、安全性等方面，因此必须特别关注。

1.首先，对于系统的数据库需求，需要考虑以下几个方面：

(1)数据库类型和结构：确定系统需要使用的数据库类型和数据结构，以满足系统的需求。例如，如果系统需要高效地存储和查询大量数据，那么选择关系数据库可能更合适；如果系统需要更高的灵活性和可扩展性，则可以考虑使用NoSQL数据库。

(2)数据库性能：需要确定系统所需的数据库性能指标，以确保系统能够满足其预期的负载。这些指标可能包括数据库的响应时间、并发性能、读写性能等。

(3)数据库安全：需要确保数据库中的数据是安全的，并且只有经过授权的用户才能访问。因此，需要在设计中考虑数据库的访问控制、身份验证和加密等安全措施。

2.除了数据库，还需要考虑系统所需的数据文件的需求。这些文件可能包括系统配置文件、日志文件、备份文件等。在确定系统所需的数据文件时，需要考虑以下几个方面：

(1)文件格式和结构：需要确定所需的数据文件格式和结构，以确保它们能够被系统正确地读取和写入。

(2)文件访问权限：需要确保系统只有经过授权的用户才能访问所需的数据文件，以保护数据的安全。

(3)文件备份和恢复：需要确保系统的数据文件可以被定期备份，并且在需要时能够进行恢复。

3.此外，在CSCI内部数据需求中还需要考虑一些其他方面的需求。例如，系统需要存储的数据量、数据的类型和格式、数据存储的位置和备份策略等。这些需求可能会影响到系统的性能、可靠性和安全性，因此必须认真考虑和规划。

总之，对于CSCI内部数据需求，需要考虑到数据库、数据文件以及其他方面的需求，以确保系统能够高效、可靠、安全地管理和存储数据。

本系统所有具体的有关内部数据的决策都留待设计时决定，具体内容详见《详细设计》文档。

## 3.8适应性需求

适应性需求是指系统在不同的环境或场景下需要满足的特定需求。在火车售票系统中，适应性需求可以包括以下内容：

1.数据依赖：系统需要获取并处理各种车票信息、用户信息、订单信息等数据，这些数据可能会受到不同的数据来源、格式、质量等方面的影响。因此，系统需要具备处理各种数据的能力，包括数据解析、数据清洗、数据验证等。

2.运行参数：系统在不同的运行环境下可能需要调整一些参数，以保证系统的正常运行和性能表现。这些参数可能包括数据库连接池大小、线程池大小、缓存大小、日志级别等。

3.可配置性：为了适应不同的用户需求和运行环境，系统需要具备可配置性。例如，管理员可以配置系统的运行模式、访问控制策略、通知方式等，以便根据不同的需求进行定制化配置。

4.可扩展性：系统需要支持功能的可扩展性，以便将来根据业务需要进行功能扩展。例如，支持新的支付方式、提供新的查询接口、添加新的车站等。

5.可定制性：系统需要支持界面的可定制化。例如，用户可以自定义界面的主题、布局、语言等，以便根据个人喜好进行定制化设置。

6.跨平台性：系统需要支持在不同的操作系统、浏览器和设备上运行，以便满足不同用户的需求和使用习惯。

## 3.9保密性需求

保密性需求是软件需求规格说明书中非常重要的一部分，它指出了系统必须满足的安全性和保密性需求。保密性需求的主要目的是确保系统能够防止非授权访问和数据泄露，从而保护用户的隐私和机密信息，减少潜在的危险。

保密性需求的要求包括但不限于以下几个方面：

1.系统应该具有严格的身份验证和访问控制机制，以确保只有授权用户才能访问敏感信息。系统管理员应该能够管理用户的权限和角色，以控制他们可以访问的数据和功能。

2.系统必须采用加密算法，以保护数据在传输和存储过程中的安全性。系统应该支持常见的加密算法和安全协议，如SSL、TLS等，以确保通信过程中的数据安全。

3.系统应该能够记录和监控所有的系统活动和日志，并能够及时检测和响应潜在的安全漏洞和攻击。系统应该具有防火墙和入侵检测系统等安全机制，以防止未经授权的访问和攻击。

4.系统应该能够对敏感数据和信息进行备份和恢复，以确保在系统发生故障或遭受攻击时能够及时恢复数据和系统功能。备份和恢复的数据和信息必须存储在安全的位置，并且必须采用加密和访问控制等安全机制进行保护。

5.系统应该具有合适的物理安全措施，如门禁、监控、防盗等，以防止未经授权的物理访问和损坏。

6.系统必须遵守相关的安全法规和标准，如ISO 27001等，以确保系统满足业界标准和法规的要求。

在设计和开发系统时，保密性需求必须作为系统的关键需求之一来考虑。为了满足这些需求，开发人员需要采取一系列的安全措施，并且对系统进行全面的安全测试和评估，以确保系统的安全性和保密性。同时，系统管理员也需要对系统进行定期的安全检查和维护，以及对用户进行安全培训和指导，以提高用户的安全意识和保密意识。只有这样，我们才能够建立一个安全、可靠、保密的火车售票系统，保障用户的安全和隐私。

## 3.10保密性和私密性需求

保密性和私密性是信息系统设计中非常重要的方面，特别是对于需要保护敏感信息的系统而言。本系统作为一款火车售票系统，也需要保障其保密性和私密性。

1.首先，本系统需要运行在一个保密性和私密性环境下，以保障敏感信息不被非法获取和使用。系统需要在可控的环境下运行，例如，系统应该安装在安全室内，只有授权的人员可以访问和操作系统，系统应该在安全的网络下运行，严格控制网络访问权限，保证系统数据不被未授权的人员获取。

2.其次，本系统需要提供保密性和私密性的类型和程度，以保障用户的个人信息、支付信息等不被泄露。例如，用户的个人信息需要进行加密存储，支付信息需要通过第三方支付平台进行加密传输，确保数据在传输和存储过程中不被窃取或篡改。

3.另外，本系统必须经受的保密性和私密性的风险也需要进行评估，以便制定相应的安全措施。例如，系统面临的风险包括未经授权的访问、恶意软件攻击、数据泄露等。系统需要采取一系列的安全措施，例如网络安全措施、身份认证措施、加密措施等，以减少系统面临的风险。

4.此外，本系统必须遵循的保密性和私密性政策也需要进行明确。例如，系统需要遵循国家相关法律法规的规定，保护用户个人信息，禁止泄露和滥用信息。同时，系统还需要制定相应的操作规范和安全策略，指导运维人员的操作行为。

5.最后，本系统需要提供保密性和私密性审核，以确保系统的安全性和合规性。例如，系统需要进行定期的安全审计和漏洞扫描，发现安全漏洞及时修复，确保系统安全运行。

综上所述，本系统需要提供多种保密性和私密性保护措施，以确保系统数据的安全性和完整性，保护用户的个人信息和支付信息不被泄露。同时，系统需要遵循相关政策法规，严格控制系统访问权限，确保系统数据不被未授权的人员获取。系统需要定期进行安全审核和漏洞扫描，及时发现并修复安全漏洞，确保系统安全运行。

## 3.11CSCI环境需求

CSCI环境需求是指软件系统所必须运行的硬件、操作系统、网络和其他相关环境的要求。这些要求对于保证系统的正常运行和性能至关重要。本系统的CSCI环境需求如下：

1.硬件需求：本系统需要运行在一台至少具有2GHz处理器、4GB内存和500GB硬盘的服务器上，以保证系统的稳定性和高性能。此外，需要安装至少一台打印机和一个扫描仪，以满足打印和扫描的需求。

2.操作系统需求：本系统需要在Windows Server 2016或以上版本的操作系统上运行，以确保系统能够充分利用硬件资源和提供稳定的操作环境。

3.数据库需求：本系统需要使用MySQL或Oracle等关系型数据库管理系统，以提供高效的数据存储和检索功能。

4.网络需求：本系统需要在支持TCP/IP协议的局域网中运行，以便用户能够通过网络访问系统。此外，系统需要具备足够的网络带宽，以支持高并发访问和数据传输。

5.安全需求：本系统需要在安全的网络环境中运行，以确保用户信息和系统数据的安全性和保密性。因此，需要使用防火墙、加密技术和访问控制等安全措施，以防止未经授权的访问和数据泄露。

6.备份和恢复需求：本系统需要定期进行数据备份，并能够在系统故障或数据丢失的情况下快速进行数据恢复。因此，需要建立完善的备份和恢复机制，并定期测试其有效性。

7.其他需求：为了确保系统的正常运行，需要安装并配置适当的软件环境和开发工具，如Java运行环境、Web服务器和代码管理工具等。

总之，本系统的CSCI环境需求对硬件、操作系统、数据库、网络、安全、备份和恢复等方面提出了一系列要求，需要进行充分的规划和准备，以确保系统能够稳定高效地运行。

## 3.12计算机资源需求

### 3.12.1计算机硬件需求

本条应描述cSc1使用的计算机硬件需求，(若适用)包括：各类设备的数量、处理器、存储器、输入/输出设备、辅助存储器、通信/网络设备和其他所需的设备的类型、大小、容量及其他所要求的特征。

处理器: （intel、amd等桌面级CPU）intel 双核高主频64bit，建议使用i5系列或i7系列，建议cpu主频高于2.8GHz ，中央处理器必须支持 SSE2 (Streaming SIMD Extensions 2)；

存储器：硬盘；

内存：最少2GB（推荐4GB以上）；

输入/输出设备：计算机输入/输出设备，键盘鼠标麦克风等；

辅助存储器：可以使用U盘；

通信设备：有可以连接互联网的网卡；

显卡：显存 1G 及以上 位宽 256 bit及以上 ，显存类型 为DDR3以上 ，建议使用NVIDIA品牌显卡(推荐使用solidworks认证的专业显卡：如：NVIDIA Quadro 系列、ATI FireGL系列 或 Firepro系列)；

其他：建议选择一台具有高分辨率显示器、多个 USB 端口和 HDMI 端口的电脑，以便连接外部设备和显示器。

### 3.12.2计算机硬件资源利用需求

本条应描述CSCI计算机硬件资源利用方面的需求，如：最大许可使用的处理器能力、存储器容量、输入/输出设备能力、辅助存储器容量、通信/网络设备能力。描述(如每个计算机硬件资源能力的百分比)还包括测量资源利用的条件。

处理器强大：软件开发通常需要处理大量的数据和程序代码。因此，单核心处理器已经不能满足我们的需要。软件工程师需要考虑到多核心的处理器。像i7或者更高的CPU可以给开发团队更好的表现。二核心或四核心的处理器可以满足日常需求；

大内存：软件开发通常需要占用大量的内存。软件工程师需要一台足够大的内存的电脑。RAM至少需要有16GB以上的内存，以确保系统没有卡顿和延迟存储器：硬盘（不低于7200转）；

存储空间：软件开发者需要存储大量的代码，数据和程序。因此，软件工程师需要一个有足够大的硬盘容量的电脑。至少需要256GB以上的SSD或更大的硬盘；

显示器：分辨率不低于1400X900；

### 3.12.3计算机软件需求

本条应描述CSCI必须使用或引人CSCI的计算机软件的需求，例如包括：操作系统、数据库管理系统、通信/网络软件、实用软件、输入和设备模拟器、测试软件、生产用软件。必须提供每个软件项的正确名称、版本、文档引用。

操作系统：建议选择 Linux操作系统，因为它有很适合软件开发的环境；

数据库管理系统：

Oracle数据库被认为是业界比较成功的关系型数据库管理系统。Oracle的数据库产品被认为是运行稳定、功能齐全、性能超群的贵族产品。对于数据量大、事务处理繁忙、安全性要求高的企业。Oracle无疑是比较理想的选择。随着Intemet的普及，Oracle适时地将自己的产品紧密地和网络计算结合起来，成为在Intemet应用领域数据库厂商的佼佼者。

Oracle数据库可以运行在Unix，Windows等主流操作系统平台，完全支持所有的工业标准，并获得最高级别的ISO标准安全性认证。Oracle采用完全开放策略，可以使客户选择最适合的解决方案，同时对开发商提供全力支持。

Apache Cassandra是一套开源分布式数据库管理系统，由Facebook开发，用于储存特别大的数据。Cassandra是一个混合型的非关系的数据库，类似于Google的BigTable。Cassandra的主要特点就是它不是一个数据库，而是由一堆数据库节点共同构成的一个分布式网络服务，对Cassandra 的一个写操作，会被复制到其它节点上去，对Cassandra的读操作，也会被路由到某个节点上面去读取。对于一个Cassandra群集来说，扩展性能是比较简单的事情，只管在群集里面添加节点就可以了；

实用、测试、生产用软件：

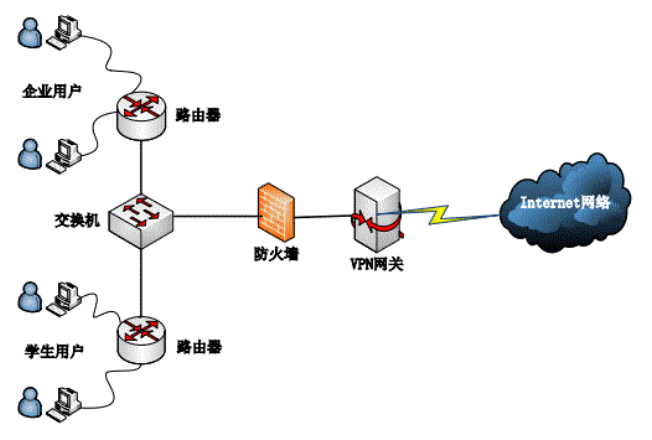
IntelliJ IDEA2023：

是java编程语言的集成开发环境。IntelliJ在业界被公认为最好的Java开发工具，尤其在智能代码助手、代码自动提示、重构、JavaEE支持、各类版本工具(git、svn等)、JUnit、CVS整合、代码分析、 创新的GUI设计等方面的功能可以说是超常的。IDEA是JetBrains公司的产品，这家公司总部位于捷克共和国的首都布拉格，开发人员以严谨著称的东欧程序员为主。它的旗舰版还支持HTML，CSS，PHP，MySQL，Python等。

，Java18（Java开发的JDK）、eclipse4.5版本（开放源代码的、基于Java的可扩展开发平台）。

### 3.12.4计算机通信需求

本条应描述CSCI必须使用的计算机通信方面的需求，例如包括：连接的地理位置、配置和网络拓扑结构、传输技术、数据传输速率、网关、要求的系统使用时间、传送/接收数据的类型和容量、传送/接收/响应的时间限制、数据的峰值、诊断功能。



网络通信全部依靠通信公司提供的电信网络，系统响应时间应在1s之内。

## 3.13软件质量因素

(若有)本条应描述合同中标识的或从更高层次规格说明派生出来的对CSCI的软件质量方面的需求，例如包括有关CSCI的功能性(实现全部所需功能的能力)、可靠性(产生正确、一致结果的能力)、可维护性(易于更正的能力)、可用性(需要时进行访间和操作的能力)、灵活性(易于适应需求变化的能力)、可移植性(易于修改以适应新环境的能力)、可重用性(可被多个应用使用的能力)、可测试性(易于充分测试的能力)、易用性(易于学习和使用的能力)以及其他属性的定量需求。

1)功能性。系统应该很好地实现在前述的需求中所提出的功能需求，使系统可以在要求的环境中正常的供用户使用。

1. 可靠性。系统能够在长时间、一定用户量同时访问的情况下，保证正确的处理信息数据，并产生正确、始终一致的结果，而不会随时间和访问量的变化而变化。
2. 可维护性。系统需要能在发生错误的时候，保证维护人员能较快地发现错误发生的原因，确定错误发生的位置，并迅速而正确地对错误加以改正。
3. 可用性。系统需要有很高的可用性，可以很好的供相应的用户使用。
4. 可移植性。软件在设计、编码和架构上需要满足一定的可移植性。当硬件环境，如服务器，软件环境，如数据库系统发生变化时，可以较好的移植到新的使用环境之中。
5. 可测试性。系统需要提供简便、充足的测试接口，以满足对系统在功能和性能上的特性的测试需求。UI

7）易用性。系统通过 UI设计，页面架构等方面，力争为用户提供一个简洁直观、通俗易懂的使用接口，使得用户不需要过多的时间就可以熟练使用系统中提供的各项功能。

## 3.14设计和实现的约束

(若有)本条应描述约束CSCI设计和实现的那些需求。这些需求可引用适当的标准和规范。

例如需求包括：

a.特殊CSCI体系结构的使用或体系结构方面的需求，例如：需要的数据库和其他软件配置项；标准部件、现有的部件的使用；需方提供的资源(设备、信息、软件)的使用；

b.特殊设计或实现标准的使用；特殊数据标准的使用；特殊编程语言的使用；

c.为支持在技术、风险或任务等方面预期的增长和变更区域，必须提供的灵活性和可扩展性.

（1）本系统采用前后端分离框架，前端使用vue+ElementPlus+Echarts框架，后端使用springboot+maven框架；

（2）前端的功能主要是可视化的数据展示，通过与后端的数据交互，来展示相关的数据信息，同时与用户来打交道，实现客户的需求，并将用户的需求及其数据传递到后端。

（3）后端主要是用户需求的处理和对数据库的管理。后端分为五个包，common是相关的配置类，controller是控制类，是和前端交互的类，service.impl是服务的主要实现类，从控制类中调用服务的主要实现类来实现相关的功能。Mapper是与数据库交互的类，其中就是调用数据库的接口类。entiy是实体对象类，主要用来对数据进行封装，很好的体现了面向对象封装的概念。

（4）根据12306的体系风格，采用CS的结构体系会更好的帮助我们完成购票系统的实现，所以我们采用的CS的体系结构风格；

（5）对于火车购票系统来说，我们需要采用较为简洁并且直接的用户界面，对于系统的目标人群，年龄覆盖范围较广，这就需要我们设计的系统操作简单。为了更好的体现系统不同模块之间的关系，我们设计使用elementplus中的典型框架，即上侧标题栏和左侧导航栏，相结合的方式，在左侧导航栏中，我们将按照系统的功能模块进行设计导航，并在功能模块之间实现小的子导航栏。在上侧的导航栏中，主要去显示一些系统的基本信息，和用户的一些基本信息，例如说用户的头像，用户名称等等。在中间的主要的面板中，根据当前的路由来展示不同的内容；

（6）我们设计采用前后端分离的技术，所以后端我们采用了当前最流行的springboot的典型结构，同时使用maven的框架，我们针对不同的功能情况，我们在大的方面设计了五个包，一个是common的包，是我们进行相关配置的包，一个是controller包，专门负责与前端的交互，一个是entiy主要存放实体对象的包，一个包是mapper包，专门进行与数据库交互的包，一个是service包，用来专门实现用户的功能的包。针对不同的对象，我们在不同的包里面设计了不同的类来进行实现。有专门的针对用户的类，也有专门针对车票的类，也有专门针对管理员的类，这样就可以实现分类管理。

## 3.15数据

说明本系统的输入、输出数据及数据管理能力方面的要求(处理量、数据量)。

本系统在用户访问数量方面，允许大量用户同时进行购票、改签以及退票操作；

允许同时导入多车次信息；同时允许多管理员同时插入车次及购票信息；

在数据库方面，本系统可以将原数据库中的各个对象都移入新数据库中，并且会比较新旧数据库的内容差别；在数据库的移植过程中，保证数据存储无误；数据库升级后，应该将新数据库的数据和功能于原数据库进行比对，检查变化情况。

## 3.16操作

说明本系统在常规操作、特殊操作以及初始化操作、恢复操作等方面的要求。

常规操作：由一般用户即可进行的操作，必须保证用户在进行该操作时不会出错；

特殊操作：对于服务器的操作和数据库的操作，应由专业的管理人员进行完成，保证特殊操作的正确性和稳定性，尽可能降低对用户操作使用的影响；

初始化操作：初始化操作应由系统架构师来完成，同时应该具有较强的健壮性：在不同的版本和不同环境下不需要再单独地进行初始化操作；

恢复操作：恢复操作也应该由专业的系统管理人员来完成，主要包含在系统出错后的数据回滚、恢复等操作。

## 3.17故障处理

说明本系统在发生可能的软硬件故障时，对故障处理的要求。包括：

a.说明属于软件系统的问题；

b.给出发生错误时的错误信息；

c.说明发生错误时可能采取的补救措施。

（1）软件故障：若用户存在某些不当操作或者未在开发者考虑范围内，则会产生错误的程序结果；返回给用户的信息会存在不合逻辑不合常理的现象；对于此类故障，应该设置用户的系统使用反馈机制，及时的接收用户的反馈，积极改正程序的问题；

（2）硬件问题：若存在服务器中断的现象，则应该在软件中提示“系统正在维护”的字样，同时可以添加适当的原因解释和预计恢复时间；为了快速恢复系统的运行，可以设置双服务器系统，即备用服务器系统。

## 3.18算法说明

最优路径算法：在火车售票管理系统中，需要计算最优的车票价格和时间，因此可以使用最优路径算法，如Dijkstra算法或A\*算法来计算乘客最短时间和最佳价格的路线。

算法的主要流程和公式如下：

Dijkstra算法：

初始化：将起点的顶点距离值设置为0，其他顶点距离值设置为无穷大。

重复以下步骤，直到所有顶点都被访问：

从未标记的顶点中选择距离起点最近的顶点。

对于该顶点的每个未标记的邻居，计算从起点到该邻居的距离，并更新距离值。

标记该顶点为已访问。

最后，从起点到目标顶点的最短路径就可以通过回溯从目标顶点的前驱节点来确 定。

具体公式为：

距离 = min(距离，初始节点到当前节点的距离 + 当前节点到下一个节点的距离)

优先级 = 距离

（按照用户设定来设定优先级，比如说价格）

A\*算法：

初始化：将起点的f值设置为h值，其他顶点的f值设置为无穷大。

重复以下步骤，直到目标节点被访问或者无法继续搜索：

选择f值最小的未访问节点。

如果该节点为目标节点，则搜索结束。

标记该节点为已访问。

对于该节点的每个未访问邻居，计算从起点到该邻居的实际距离和估计距离， 更新f值和前驱节点。

最后，从起点到目标节点的最短路径可以通过回溯从目标节点的前驱节点来确定。

具体公式为：

f值 = g值 + h值

g值 = 初始节点到当前节点的实际距离

h值 = 当前节点到目标节点的估计距离，可以使用曼哈顿距离、欧几里得距离等 几何距离，这里使用路程的距离，或者可以用耗时等效代替路程

数据挖掘算法：火车售票管理系统可以使用数据挖掘算法来分析历史数据，了解乘客的出行习惯和购票趋势，以便更好地管理车票销售和优化座位安排。

算法的主要流程和公式如下：

数据挖掘算法：

支持度 = 包含项集的事务数 / 总事务数

置信度 = 支持同时包含A和B的事务数 / 支持包含A的事务数

数据挖掘算法：

频繁项集挖掘：

定义一个支持度阈值，找出所有支持度不小于该阈值的项集。

从单个元素开始，迭代地生成更大的项集并计算其支持度，直到无法生成更大 的项集。

关联规则挖掘：

定义一个置信度阈值，找出所有置信度不小于该阈值的关联规则。

对于每个频繁项集，生成其所有可能的关联规则，并计算其置信度。

人工智能算法：火车售票管理系统可以使用人工智能算法，如机器学习算法和深度学习算法，来预测乘客需求和优化票价设置。

乘客需求预测：可以使用机器学习算法，如回归分析、决策树、随机森林等，来预测不同时间段的乘客需求。首先需要收集历史售票数据、天气数据、节假日等数据，然后使用机器学习算法建立预测模型。模型训练完成后，可以根据预测结果提前安排列车车次、座位数量等，以满足不同时间段的乘客需求。

票价优化：可以使用深度学习算法，如神经网络、卷积神经网络等，来优化票价设置。首先需要收集历史售票数据、乘客类型、出行时间、目的地等数据，然后使用深度学习算法建立优化模型。模型训练完成后，可以根据乘客类型、出行时间、目的地等因素，自动调整票价，以最大化收益和客户满意度。

推荐系统：可以使用机器学习算法，如协同过滤、基于内容的推荐等，来推荐最适合乘客的车次、座位等。首先需要收集乘客历史购票数据、乘客偏好、产品信息等数据，然后使用机器学习算法建立推荐模型。模型训练完成后，可以根据乘客的历史购票数据和偏好，推荐最适合的车次、座位等，提高乘客的购票体验。

需要注意的是，以上算法实现需要有足够的数据支持，同时需要进行模型训练和调优，才能得到较好的效果。建议在实施前，进行充分的数据分析和算法评估，以确保算法的可行性和有效性。

优化算法：火车售票管理系统可以使用优化算法，如遗传算法和模拟退火算法，来优化座位安排和票价设置，以最大程度地利用可用资源并提高收益。

遗传算法实现

1.定义适应度函数：适应度函数用于评估每个个体（座位安排和票价设置方案）的 优劣程度。适应度函数应该考虑以下几个方面：

座位利用率：座位利用率越高，收益越大。

票价收益：高价位的票价应该与航线的需求量相符合，以最大化收益。

竞争对手：考虑竞争对手的票价和座位数量，以便制定有效的票价策略。

2.初始化种群：随机生成一定数量的座位安排和票价设置方案，作为初始种群。

3.选择：通过选择算子，从当前种群中选择一部分个体用于繁殖后代。选择的过程 应该考虑个体的适应度，适应度高的个体有更大的概率被选择。

4.交叉：在选定的个体中，使用交叉算子进行交叉，生成新的后代。交叉的过程应 该考虑座位安排和票价设置的相关性，以保证生成的后代仍然是可行的解。

5.变异：对新生成的后代使用变异算子进行变异，以增加种群的多样性。

6.替换：将新生成的后代替换掉原来的一部分个体，以保证种群大小不变。

7.终止条件：当达到预设的迭代次数或者达到适应度的最优解时，终止遗传算法。

模拟退火算法实现

1.定义能量函数：能量函数用于评估座位安排和票价设置的优劣程度。能量函数应 该考虑以下几个方面：

座位利用率：座位利用率越高，能量值越小。

票价收益：高价位的票价应该与航线的需求量相符合，以最大化收益。

竞争对手：考虑竞争对手的票价和座位数量，以便制定有效的票价策略。

2.初始化状态：随机生成一个座位安排和票价设置方案，作为初始状态。

3.设定初始温度和冷却速率：温度越高，接受劣解的概率越大；冷却速率越慢，搜 索的时间越长。

4.在当前温度下，随机生成一个新状态。

5.计算能量差：计算新状态与当前状态的能量差。

6.根据概率接受新状态：如果新状态的能量差小于等于零，则接受新状态；否则， 根据一定概率接受新状态。

7.降温：将温度降低一定比例，如0.9。

8.终止条件：当温度降低到一定程度，或者达到一定迭代次数时，终止模拟退火算法。

## 3.19有关人员需求

使用或支持火车售票系统（CSCI）的人员有关的需求，包括人员数量、技能等级、责任期、培训需求等其他信息：

1.人员数量和技能等级

火车售票系统需要如下人员：

售票员：负责处理客户购票需求，需要了解火车票销售业务以及相关政策。需求人 员数量：根据车站规模和客流量确定。

客服人员：专门解答用户疑问，提供售后支持。需求人员数量：3-5人，具备良 好的沟通技巧和客户服务意识。

技术支持人员：负责软件维护、更新和故障排除。需求人员数量：5-10人，具 备计算机科学或相关领域的专业知识。以下可能是我们需要的一些技术人员的类别和他 们需要的技能等级：开发人员可能需要具备Java、C++、Python等多种编程语言的经验， 测试人员可能需要具备自动化测试和手动测试的技能，项目经理可能需要有项目管理和 沟通技巧等方面的经验。

2.职责期：

根据公司规定和合同约定，人员的职责期可长期或短期。售票员和客服人员职责期一般为1年，可在年度评估后续约；技术支持人员职责期为2年，可在合同到期后续约。在项目运营良好的情况下考虑给员工发放的福利。

每个参与此项目的工作人员可能需要参与项目的不同阶段，例如开发人员可能需要在整个项目期间参与，而测试人员可能需要在测试阶段参与。在项目的开发阶段一定要做好每个工作人员的工作分配。此外在进行该项目的开发工作时，我们可能会进行人员的调整。可能会有新员工加入项目的开发当中来。这就需要为新员工提供培训和支持，以使他们能够更好地适应项目需求。

3.培训需求：

新员工在入职前需要接受相关培训，以确保具备完成工作的基本技能。培训内容包括：

售票员：火车票销售流程、票务政策、客户服务技巧等。

客服人员：客户服务理念、沟通技巧、问题解答技巧等。

技术支持人员：系统架构、故障排除、软件更新等。

4.人力行为工程需求：

同时存在的用户数量的需求：

火车售票系统是一个大型的数据管理系统，其会有大量用户进行使用。该软件系统可能需要支持数百万个用户的账户管理，在同一时间也需要满足至少有数十万的用户在线登录的需求，在用户访问同一班次的列车时还需要确保数据的实时性，即一个座位被一个用户选中之后短时间内其他用户不可以选中该座位，除非第一次选中的用户放弃了该位置，此时在后续用户重新进入该系统时该系统才会重新开发对于该位置的选取和选择权。

为降低人为错误的发生，火车售票系统应考虑以下人力行为工程需求：

火车售票系统会有大量用户进行操作，所以需要考虑人为错误，如输入无效数据或 在错误的位置点击按钮，以及考虑如何防止或最小化这些错误的影响。在正常和极端条 件下可预测的人为错误：如输入错误、误操作等。

人为错误造成严重影响的特定区域：如购票金额错误、购票时间错误等。

错误消息的颜色和持续时间：火车售票管理系统系统可能需要设计清晰的错误消 息，以便用户可以快速识别和解决问题。使用醒目的颜色（如红色）提示错误，持续一 定的错误提示时间以确保用户可充分理解错误内容。

可以提供给用户的操作引导：火车售票管理系统可能需要提供用户手册、在线帮助和培训视频等支持内容，以帮助用户了解和使用该系统。

关键指示器或关键的物理位置：该软件系统可能需要设计易于使用的用户界 面， 以便用户可以快速找到所需的功能和信息。如购票按钮、退票按钮等，应设置在易 于操作的位置。

听觉信号的使用：该软件系统可能需要使用声音和其他听觉信号来提供反馈和提 示，以帮助用户了解系统状态和事件。如在关键操作（如购票成功、退票成功）时发出 提示音，提醒用户操作结果。

## 3.20有关培训需求

在我们的火车售票系统项目中，需要包含培训软件的CSCI。培训软件应该具备以下功能：

1.培训计划管理：该功能应该允许管理员创建和管理员工的培训计划。管理员可以根据员工的职位和工作内容，制定相应的培训计划，包括培训课程、培训时间、培训地点、培训方式等。首先，管理员需要在系统中维护员工信息，包括姓名、职位、工作地点、联系方式等。这些信息将作为制定培训计划的依据，以便管理员能够根据员工的工作内容和职位要求，为其制定相应的培训计划。此外，管理员需要对员工的培训需求进行分析，以便制定相应的培训计划。分析员工的现有知识和技能水平，确定员工需要掌握的新知识和技能，并制定培训目标和计划。另外，管理员需要管理培训资源，包括培训课程、培训讲师、培训场地等。管理员需要选择合适的培训课程和培训讲师，为员工提供优质的培训服务，同时需要预定合适的培训场地，以确保培训计划的顺利实施。最后，管理员需要根据员工的培训需求和培训资源，制定相应的培训计划。培训计划应包括课程名称、授课讲师、培训时间、培训地点、培训方式、培训费用等信息，以便员工能够清楚地了解培训计划的具体内容和安排。并且需要将制定好的培训计划发布到系统中，并通知员工参加培训。管理员可以通过系统中的通知功能，向员工发送培训通知，以便员工能够及时了解培训计划的具体安排和要求。

2.培训课程管理：该功能应该允许管理员创建和管理培训课程。管理员可以添加课程名称、描述、培训内容、培训方式、考核方式等信息。课程分类管理：管理员可以对课程进行分类管理，便于员工根据自己的需求选择相应的课程。例如，管理员可以将课程按照不同的职位要求进行分类，或者按照不同的技能水平要求进行分类。课程评价管理：管理员可以设置课程评价功能，让员工在完成课程后进行评价。管理员可以根据员工的评价结果，对课程进行优化和改进，提高培训效果。培训计划集成：管理员可以将课程集成到培训计划中，让员工根据需要选择相应的课程进行学习。管理员可以根据员工的职位和工作内容，为员工制定相应的课程学习计划。

3.培训教材管理：该功能应该允许管理员上传和管理培训教材。管理员可以添加教材名称、描述、教材文件等信息。管理员可以将教材按照不同的版本进行管理，让员工能够选择最适合自己的教材版本。管理员可以设置教材下载功能，让员工能够方便地下载所需的教材。管理员可以设置下载权限，保证教材的安全性和保密性。管理员还可以设置教材评价功能，让员工在学习教材后进行评价。管理员可以根据员工的评价结果，对教材进行优化和改进，提高培训效果。

4.考核与培训记录管理：该功能应该允许管理员创建和管理员工的考核信息。管理员可以添加考核名称、描述、考核方式、考核持续时间、考核成绩等信息。除此之外还可以在人员管理系统中记录员工的培训记录，包括参加的培训课程、培训时间、培训成绩等信息。这些记录可以作为员工培训评估和绩效考核的依据。最后需要管理员对培训计划和课程进行评估。管理员可以根据员工的反馈和考核成绩，评估培训的效果和质量，及时调整培训计划和课程。

## 

## 3.21有关后勤需求

1.系统维护：该功能应该加入可供维护的功能，比如说日志系统，系统运行状态的后台界面。允许管理员进行系统维护，包括系统备份、系统更新、系统修复等操作。系统需要定期进行数据的备份，以确保数据的准确性，该系统需要时刻进行日志的记录和更新，当系统崩溃时可以进行数据的恢复或者是工作状态的恢复。管理员应该能够对系统的运行状态进行实时的监控，及时发现并解决系统故障。

2.软件支持：该功能应该提供其他的软件支持，比如应该支持应该提供用户问题解答，以便用户能够快速解决任何与软件使用相关的问题。这可以通过提供常见问题解答（FAQ）以及在线支持论坛、电子邮件和电话支持等方式实现。此外，软件支持还应该具备多语言支持，以便全球用户能够顺畅地使用软件。其次软件支持应该提供及时的软件更新，确保软件能够保持最新、最先进的状态。这些更新可以涉及功能增强、错误修复、安全修补程序和新功能的添加。管理员应该确保软件更新可用并通知用户。再者软件支持应该提供快速的软件修复，以便管理员能够快速解决软件中出现的任何问题。这可以通过提供在线支持论坛、电子邮件和电话支持等方式实现。管理员需要及时响应用户反馈，以便能够快速识别和解决问题。另外软件支持还应该提供用户培训和文档，以便用户能够快速掌握软件的使用方法和功能。这可以通过提供在线视频、文档和培训课程等方式实现。管理员应该确保培训和文档的质量和及时性。还有可以考虑的需求是软件支持应该提供反馈机制，以便用户能够向管理员提供反馈和建议。这可以通过在线支持论坛、电子邮件和电话支持等方式实现。管理员应该及时回复用户的反馈和建议，以便能够改进软件并满足用户的需求。管理员还可以跟进用户反馈和建议的统计数据，以便了解用户的需求和痛点，并在软件的下一次更新中加以改进。

3.系统运输方式：该系统的运输方式主要通过网络数据传输的方式进行。不过，网络数据传输需要保证网络的可靠性和数据的安全性。管理员需要确保网络连接的可靠性和稳定性，防止数据传输过程中出现数据丢失、数据包损坏等问题。其次我们需要考虑传输工作的效率问题：为了减少数据传输的时间和网络带宽，系统可以采用数据压缩技术。数据压缩可以通过减少数据的存储空间来缩小数据的大小，从而减少数据传输时间和网络带宽。压缩数据的同时也需要确保数据的完整性和准确性，以便系统能够正确地解压缩数据。另外该系统涉及交易这一模块，要严格确保交易时的安全性能，以免敏感数据被未经授权的人员访问和窃取。系统可以采用数据加密技术来保护数据的安全性。数据加密可以通过使用密钥来将数据转换为加密形式，从而保护数据免受未经授权的访问。管理员需要确保密钥的保密性，并定期更改密钥以提高数据的安全性。此外网络安全管理人员也可以找机构进行证书的发放，使用更加安全的证书进行交易链接。

4.供应系统的需求：管理员需要考虑售票系统的数据供应的来源，这些数据应该来源于其他的数据管理系统，管理员应当根据这些基础的数据更新本系统上的数据，以满足实时性的需求，需要的数据可能来自铁路管理部门的数据管理系统。为了确保售票系统中的数据及时更新，管理员需要建立数据更新机制。数据更新机制应该定期检查数据管理系统中的数据是否发生变化，并将变化的数据更新到售票系统中。管理员需要确保数据更新机制的准确性和及时性，以便系统能够及时反映列车信息和票价信息的变化。该系统可能需要登记列车的信息，比如列车的型号，时速，列车车厢个数，列车内餐厅所提供的食物等等。售票系统需要准确地显示票价信息，包括不同列车类型、不同座位类型、不同时间段的票价信息。管理员需要根据市场的实时燃油费信息或者是铁路保养费用，及时更新票件信息，确保票价信息的准确性和及时性，以便乘客能够准确地了解票价和购买车票。

5.对现有设施和设备的影响：该条分析需要评估系统对现有设施的影响，包括车站、餐厅等设施，该系统虽然是软件系统，但是也是需要硬件实体承载其工作运行，我们可能会在车站安装能运行该系统的设备。对设备的具体影响有：系统需要在车站安装设备，以便乘客能够使用售票系统。管理员需要评估系统对车站设施的影响，包括设备的安装位置、设备的数量、设备的维护和保养等问题。另外该系统需要一些别的设备，比如客户有开发票的需求，或者打印车票的需求。所以我们需要提供包括计算机、服务器、打印机、扫描器等设备。管理员需要评估系统对硬件设备的影响，包括设备的数量、设备的配置、设备的安装位置等问题。该系统可以提供电子车票或者纸质车票，所以售票系统可能需要提供多种车票验证设备，包括纸质和电子车票的检验。以便乘客进入列车。管理员需要评估系统对车票验证设备的影响，包括设备的类型、设备的数量、设备的安装位置等问题。售票系统需要与其他系统进行集成，包括铁路管理部门的数据管理系统、车站系统、餐厅系统等。管理员需要评估系统对其他系统的影响，包括集成的难度、集成的方式、集成的时间等问题。

以上是后勤方面的CSCI需求的基本功能要求，具体实现还需要根据实际情况进行设计。

## 3.22其他需求

这里仅仅提及可能对火车售票系统需要加以补充的需求的概括性描述，不做具体的展开性介绍。

1.配置性需求：描述软件系统的可配置性和可定制性，包括用户可以自定义的配置项和参数等。

2.扩展性需求：描述软件系统的可扩展性和可扩展性要求，包括系统需要支持的扩展方式和接口。

3.可移植性需求：描述软件系统的可移植性和支持的平台范围，包括系统需要支持的操作系统、编程语言等。

4.可重用性需求：描述软件系统的可重用性和支持的重用方式，包括系统需要支持的重用接口和文档等。

5.可靠性需求：描述软件系统的可靠性和可靠性要求，包括系统需要实现的故障恢复方式和机制等。

6.维护性需求：描述软件系统的可维护性和可维护性要求，包括系统需要实现的代码规范、注释、文档等。

7.可用性需求：描述软件系统的可用性和可用性要求，包括系统需要满足的用户界面设计规范、交互方式、易用性等。

8.测试性需求：描述软件系统的可测试性和测试要求，包括系统需要支持的测试方式、测试工具等。

9.管理性需求：描述软件系统的可管理性和管理要求，包括系统需要支持的监控方式、日志记录、统计分析等。

10.追踪性需求：描述软件系统的可追踪性和追踪要求，包括系统需要支持的需求跟踪、变更管理等。

## 3.23包装需求

本文档描述了火车售票管理系统（以下简称为“系统”）在包装、加标签和处理方面的需求。这些需求旨在确保系统的交付和部署顺利进行，符合相关规范和标准。

1. 包装

1.1. 系统交付时，所有软件组件应打包成一个易于部署的安装程序或压缩文件。

1.2. 安装程序或压缩文件的名称应明确表示系统名称、版本号和发布日期。

1.3. 如有硬件设备相关（如打印机、读卡器等），应提供设备驱动程序和安装指南。

1.4. 包装应确保软件在传输过程中受到充分保护，以防止损坏或数据丢失。

2. 加标签

2.1. 系统中的所有文件和文件夹应具有描述性名称和版本信息，以便于识别和跟 踪。

2.2. 如适用，源代码文件应包含文件头，其中包含作者、创建日期、版权信息和文 件描述。

2.3. 为方便用户识别和操作，界面元素（如按钮、菜单项等）应有清晰的标签。

3. 处理

3.1. 系统应提供详细的安装、配置和操作手册，以指导用户进行正确的处理。

3.2. 在系统交付前，应进行充分的测试以确保满足功能和性能需求。

3.3. 为确保系统的稳定运行，应提供软件更新和补丁的支持。

3.4. 如有故障或问题发生，应提供有效的技术支持和客户服务。

4. 规范和标准

本系统应遵循以下适用的规范和标准：

ISO/IEC 12207:2008（信息技术 - 系统和软件工程 - 软件生命周期过程）

ISO/IEC 25010:2011（系统和软件工程 - 系统和软件质量模型）

## 3.24需求的优先次序和关键程度

本系统中需要特别指明优先顺序的需求主要有如下几点：（由高到低）

|  |  |  |
| --- | --- | --- |
| 需求（优先顺序从上往下递减） | 参考要素 | 权值 |
| 保密性需求 | 交易的安全 | 10 |
| CSCI环境需求 | 系统运行基础 | 9 |
| 计算机资源需求 | 系统资源配置 | 9 |
| CSCI外部接口需求 | 系统内部功能 | 8 |
| CSCI内部接口需求 | 系统内部功能 | 8 |
| CSCI内部数据需求 | 系统内部功能 | 8 |
| 软件质量需求 | 系统完成质量 | 8 |
| 包装需求 | 涉及系统安装 | 7 |
| 适应性需求 | 系统的可移植性 | 7 |
| CSCI能力需求 | 人员的技术要求 | 6 |
| 有关人员需求 | 人员水平和工作要求 | 5 |
| 有关培训需求 | 人员的技术培养 | 5 |
| 有关后勤需求 | 后期系统的维护 | 5 |

关键程度：

高：下面给出关键程度最高的几个需求，因为首先我们需要保证火车售票系统的可运行性，而以下这些关键程度较高的的需求是该系统稳定运行的关键，如该系统所要求，所需要的环境，这个是最基本也是最需要高完成度的需求，没有其运行的环境的配置该软件系统是没办法运行的。此外我们需要满足其计算机资源的需求，要考虑系统的运行峰值的状况，考虑运行的最大限度和最高要求，在用户使用率较低时如何进行资源的合理分配。当然最重要的就是保密性需求，因为该系统涉及到用户的个人信息的使用和处理，还有涉及到金钱流动上的交易问题，所以在这两方面的数据上一定要做好保密性。

下面给出一些考量因素：保密性需求：该需求是最关键的需求之一，因为系统涉及到用户的个人信息和金钱交易，必须要保证数据的安全和保密性。因此，系统必须有足够的安全措施，如加密用户数据和交易信息，限制访问和授权等等。CSCI环境需求指的是系统所运行的环境，包括操作系统、数据库、网络等等。系统必须能够在规定的环境下运行并提供所需的功能，所以必须准确地定义和满足这些环境需求。计算机资源需求指的是系统所需的计算机资源，包括处理器、内存、硬盘等等。必须确保系统在运行时能够满足最大负载的需求，并且在系统使用率较低时能够合理分配资源。

1.保密性需求

2.CSCI环境需求

3.计算机资源需求

中：下面给出几个关键程度适中的需求，我们在考虑这些需求时也是需要着重考虑的，但是与关键程度高的需求产生冲突时，我们要尽量满足关键程度较高的需求。比如我们的接口需要提供的功能可能会破坏保密性的需求，因此我们可能要放弃那个功能，从而保证我们的保密性是完全满足需求的。下面给出一些考量因素：CSCI外部接口需求指的是系统与外部系统或服务进行交互的接口需求，如支付网关、用户认证系统等等。必须确保系统能够与这些外部系统进行正确的数据交换和集成。CSCI内部接口需求指的是系统内部各个模块之间的接口需求，必须确保系统内部各个模块之间的数据交换和协作顺畅。CSCI内部数据需求指的是系统内部所需的数据和数据结构，必须确保系统能够正确地管理和存储数据，以及提供数据访问和查询功能。软件质量需求指的是系统的软件质量要求，如性能、可用性、可靠性、可维护性等等。必须确保系统能够满足这些质量要求，提供高质量的服务。包装需求指的是系统的软件包装和发布要求，包括安装程序、文档、版本控制等等。必须确保系统能够方便地部署和发布，同时提供详细的文档和版本控制。适应性需求指的是系统的适应性和可扩展性要求，必须确保系统能够适应不同的用户需求和环境，以及能够方便地扩展和升级。

4.CSCI外部接口需求

5.CSCI内部接口需求

6.CSCI内部数据需求

7.软件质量需求

8.包装需求

9.适应性需求

低：下面的需求就是关键程度较低的，下面的需求在考虑上程度较上面提到的需求低，但是也是我们要考虑的需求，因为人员的能力问题关系到项目的落实和完成的质量，但是这些问题都可以较为轻松地解决。

下面给出一些考量因素：CSCI能力需求：该需求指的是系统所需的技术能力和人员能力要求，必须确保系统开发人员具备必要的技能和经验，能够完成系统的开发和维护工作。有关人员需求：该需求指的是系统所需的人员资源，包括开发人员、测试人员、运维人员等等。必须确保有足够的人员资源来支持系统开发和维护工作。有关培训需求：该需求指的是系统用户和管理员的培训需求，必须确保系统用户和管理员能够掌握系统的使用和维护技能，以确保系统的正常运行。有关后勤需求：该需求指的是系统开发和维护所需的后勤支持，如办公场地、设备维护、物资采购等等。必须确保有足够的后勤支持来支持系统的开发和维护工作。

10.CSCI能力需求

11.有关人员需求

12.有关培训需求

13.有关后勤需求

# 4合格性规定

对于第3章中每个需求，指定所使用的方法，以确保需求得到满足，合格性方法包括：

a.演示：运行依赖于可见的功能操作的CSCI或部分CSCI,不需要使用仪器、专用测试设备或进行事后分析；

b.测试：使用仪器或其他专用测试设备运行CSCI或部分CSCI,以便采集数据供事后分析使用；

c.分析：对从其他合格性方法中获得的积累数据进行处理，例如测试结果的归约、解释或推断；

d.审查：对CSCI代码、文档等进行可视化检查；

e.特殊的合格性方法。任何应用到CSCI的特殊合格性方法，如：专用工具、技术、过程、设施、验收限制。

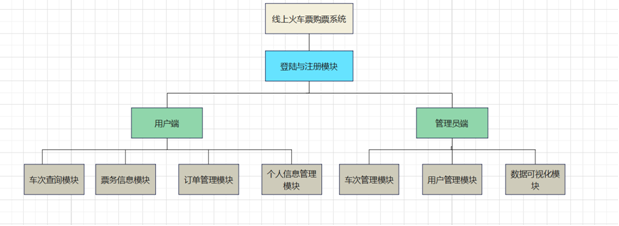
|  |  |  |  |
| --- | --- | --- | --- |
| 需求ID | 需求描述 | 需求实现 | 合格性检测方法 |
| REQ-001 | CSCI能力需求 | 登陆与注册模块 | a b d |
| REQ-002 | 车次查询模块 | a b d |
| REQ-003 | 票务信息模块 | a b d |
| REQ-004 | 订单管理模块 | a b d |
| REQ-005 | 个人信息管理模块 | a b d |
| REQ-006 | 车次管理模块 | a b d |
| REQ-007 | 用户管理模块 | a b d |
| REQ-008 | 数据可视化模块 | a b d |
| REQ-009 | CSCI外部接口需求 | 用户接口 | b c d |
| REQ-010 | 硬件接口 | b c d |
| REQ-011 | 软件接口 | b c d |
| REQ-012 | 通信接口 | b c d |
| REQ-013 | CSCI内部接口需求 | 内部接口 | b c d |
| REQ-014 | CSCI内部数据需求 | 内部数据 | a c e |
| REQ-014 | 适应性需求 | 适应性 | b |
| REQ-015 | 保密性需求 | 保密性 | e |
| REQ-016 | 保密性和私密性需求 | 保密性和私密性 | e |
| REQ-017 | CSCI环境需求 | 环境要求 | b e |
| REQ-018 | 计算机资源需求 | 计算机硬件 | b |
| REQ-019 | 计算机硬件资源利用需求 | b |
| REQ-020 | 计算机软件需求 | b |
| REQ-021 | 计算机通信需求 | b |
| REQ-022 | 软件质量因素 | 软件质量需求 | a b c d |
| REQ-023 | 操作 | 操作需求 | a b d |
| REQ-024 | 故障处理 | 故障处理 | a c |
| REQ-025 | 人员需求 | 人员需求 | e |
| REQ-026 | 培训需求 | 培训需求 | e |
| REQ-027 | 后勤需求 | 后勤需求 | e |
| REQ-028 | 其他需求 | 其他需求 | e |
| REQ-029 | 包装需求 | 包装需求 | e |

# 5需求可追踪性

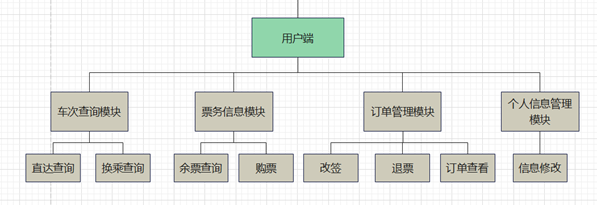
要确保每一个需求都能被跟踪到，以便在软件开发的各个阶段进行追踪和验证。

## 5.1CSCI能力需求可追踪性

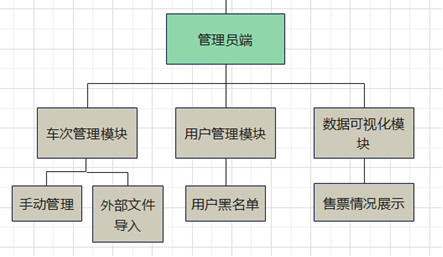
该火车票购票系统分为用户端和管理员端两个部分，用户端包括登陆与注册模块、车次查询模块、票务信息模块（提供票价、余票信息，购票功能）、订单管理模块（提供改签以及退票功能）、个人信息管理模块；管理员端包括车次管理模块（手动添加车次、从外部文件导入车次信息功能）、用户管理模块、数据可视化查看模块。下图为主要模块结构：



用户系统中模块的功能细化如下：



管理员系统中模块的功能细化如下：



## 5.2适应性需求可追踪性

为了满足对于不同浏览器的适应性需求，系统会在详细设计以后最后测试阶段对于不同的浏览器实现分类浏览。

## 5.3保密性和私密性需求可追踪性

为了满足不同用户以及管理员的保密性和私密性需求，系统会在登录的一开始对于用户身份进行识别，对于不用级别的用户以及不同权限的管理员实现浏览内容的分级，以及实现不同功能与服务的提供。

## 5.4计算机资源需求可追踪性

|  |  |  |
| --- | --- | --- |
|  | Web服务器 | 客户机 |
| 用途 | 用于假设系统所使用的Web服务器，承担了响应系统用户访问请求的责任，为用户提供Web服务 | 用户向服务器请求服务，实现用户与系统的交互 |
| 处理器 | 生产产商：Intel  核心数量：八核  主频：cpu主频2.8GHz  中央处理器：支持SSE2(Streaming SIMD Extensions 2) | 无特别要求 |
| 存储器 | 内存：4GB | 无特别要求 |
| 输入/输出设备 | 计算机输入/输出设备，键盘鼠标麦克风等 | 无特别要求 |
| 辅助存储器 | U盘 | 无特别要求 |
| 通信设备 | 可以连接互联网的网卡 | 无特别要求 |
| 显卡 | 显存1G，位宽256bit，现存类型为DDR3，NVIDIA系列 | 无特别要求 |
| 其他 | 选择一台具有高分辨率的显示器、多个USB端口和HDMI端口的电脑，以便连接外部设备和显示器 | 无特别要求 |

## 5.5软件质量需求可追踪性

|  |  |
| --- | --- |
| 质量需求内容 | 可追踪性 |
| 功能性 | CSCI能力需求 |
| 可靠性 | 故障处理 |
| 可维护性 | 故障处理 |
| 可用性 | 操作 |
| 可移植性 | 系统详细设计 |
| 可测试性 | 系统测试 |
| 易用性 | 操作 |

## 5.6数据需求可追踪性

为了满足系统同时间的大量的用户上线并进行各种操作，并要求系统能处理多个用户同时提交单次数据量较大的数据信息，同时保证高速率别的数据输出量，系统将在详细设计中通过数据库以及系统架构实现上述的要求

## 5.7操作需求可追踪性

对于常规操作应该由系统保证没有出错，对于特殊操作以及初始化操作均会由数据库内部功能实现，对于恢复操作，除了会由数据库的负责人员承担一部分之外，系统本身也会内置使用日志，并进行数据的备份，方便进行恢复。

## 5.8故障处理需求可追踪性

对于各种故障以及处理方式在3.17中做过详细说明，需要对于各种硬件或软件的错误信息进行记录与分析，进行后继的完善与总结。

## 5.9人员需求可追踪性

对于人员需求则是由各个阶段不同的负责人来负责，例如对于售票员、客服人员以及技术支持人员都要有相关的负责人来管理。对于各个阶段所需要的人员以及分工进行细化与落实。

## 5.10培训需求可追踪性

培训需求主要由系统开发以及测试人员负责，即现阶段的小组全体开发研究成员，对于用户以及其他开发人员进行培训，通过对于培训教程、方案、计划等的规划进行进一步的实现。

## 5.11后勤需求可追踪性

通过开发研究人员对于各项系统以及设施设备的维护的实践与规划，以及各项软件和供应系统的管理和跟进来实现。

## 5.12其他需求可追踪性

通过开发研究人员进一步对于系统功能的讨论与研究，在系统详细设计时，综合考虑到各种例如扩展性、配置性等各个方面的需求的实现。

## 5.13包装需求可追踪性

在进行包装、加标签、处理以及规范和标准化时，由专门的负责人员进行细化分工，严格按照标准与要求进行实现。

# 6尚未解决的问题

在设计和开发火车售票系统项目的过程中，可能会涉及到一些软件需求中尚未解决的遗留问题。这些问题可能会对系统的实现、性能、安全性、可靠性等方面产生影响，所以在此部分对这些问题进行记录和描述。以下是一些在软件需求中尚未解决的遗留问题：

系统性能问题：火车售票系统项目需要支持大量用户同时进行订票和查询等操作，因此系统的性能是至关重要的。目前还需要进一步研究和优化系统的性能，例如通过分布式架构、缓存等技术手段来提升系统的响应速度和吞吐量。

安全性问题：火车售票系统的设计中涉及到用户的个人信息和支付信息等敏感数据，因此系统的安全性是至关重要的。目前还需要进一步研究和加强系统的安全机制，例如通过加密、身份认证、访问控制等技术手段来保护用户数据的安全和隐私。

用户体验问题：由于对于火车票的售卖与购买需求的日益增长，火车售票系统需要提供更为良好的用户体验，包括简单易用的界面、快速响应的操作等方面。目前还需要进一步研究和优化系统的用户体验，例如通过设计更加人性化的界面、采用更快速的响应方式等技术手段来提升用户体验。

功能需求问题：火车售票系统需要提供完善的功能，目前该系统已经实现一系列的操作，包括订票、退票、改签、查询等方面，但是在后续的设计中还需要进一步研究和完善系统的功能需求，例如添加更多的支付方式、支持更多的车站和线路等方面。

可靠性问题：火车售票系统需要保证系统的可靠性，避免出现意外崩溃和数据丢失等问题。在后续设计中还需要进一步研究和加强系统的可靠性，例如可以通过数据备份、故障恢复等技术手段来保证系统的可靠性。

火车售票系统APP开发问题：目前我们研发的火车售票系统是一个Web项目，尚未实现APP的研发与投入使用。由于现在人们大多使用手机APP进行火车购票等各种操作，所以为了以后为用户提供更便捷、更全面的服务，在后续的设计中应该考虑进行APP的开发与设计。而火车售票系统APP需要兼容多个平台和设备，例如Android、iOS等操作系统和多种型号的手机和平板电脑等设备。所以在后续设计中还需要进一步研究和优化系统的兼容性，例如通过适配多种分辨率和屏幕大小等技术手段来提升系统的兼容性。

火车售票系统APP的用户体验和界面设计问题：虽然我们已经有了火车售票系统APP初步的设计想法，但是还需要更深入的思考和讨论，以确保我们的设计能够最大程度地满足用户的需求，并提供最佳的用户体验。我们需要考虑如何使用户能够更轻松快捷地浏览车次信息、选择座位、购票并支付，以及完成一系列的相关其他操作，例如改签与退票等，同时也需要考虑如何提供给用户足够的反馈和支持，以帮助他们解决任何出现的问题。

火车售票系统APP的性能和稳定性问题：在后续研发出火车售票系统相关的APP后，我们还需要确保APP在各种情况下都能够稳定运行，包括在网络连接不稳定或信号弱的情况下，并且我们还应该考虑如何优化APP的性能，以确保它能够快速响应用户的操作，并提供最佳的用户体验。所以在后续的APP开发与设计我们需要测试APP在不同设备上的性能，并进行优化，以确保它在各种情况下都能够正常运行。

火车售票系统APP的安全性和隐私保护的问题：在后续火车售票系统APP投入使用后，我们就需要确保APP用户的个人信息和支付信息得到充分的保护，防止任何未经授权的访问或泄露。我们需要采取一系列措施来确保APP的安全性，包括数据加密、身份验证和访问控制等。同时，我们还需要遵守相关的隐私保护法律法规，保护用户的隐私权。

火车售票系统APP后续的维护和更新相关的问题：我们需要确保APP的代码结构和文档具有足够的可维护性，以便在需要进行修复或更新时能够更快、更方便地进行操作。我们需要考虑如何优化代码结构，使其易于理解和维护，并提供足够的文档和注释，以便小组开发研究人员能够更轻松地进行操作。此外，我们还需要考虑如何支持APP的后续更新和升级，以确保它能够适应未来的需求和变化。

# 7注解

该部分旨在提供一些有助于理解和解释本文档中使用的术语和定义的背景信息。我们将提供与火车售票系统相关的一些基本信息和概念，以帮助读者更好地理解本文档中的内容：

背景信息：火车售票系统是一款用于火车票预订和购买等各种功能的Web项目。该系统将提供用户友好的界面，方便用户查询列车时刻表、票价、预订火车票以及支付购票费用、退票改签等功能，并且能够方便管理员更快捷方便地查看和管理车次信息以及用户信息等各种信息，方便管理员的维护与使用。火车售票系统的设计旨在为用户提供便捷的购票体验，为管理员提供更全面的统一管理途径，并提供高效的售票服务。

本文档中涉及到的一些术语和定义如下：

火车售票系统：指本文档所描述的火车售票系统，是用于用户进行火车票预订和购买等功能，以及管理员进行车次信息和用户信息管理和维护的系统。

用户：指使用火车售票系统的个人或实体。

管理员：指具有一定权限的利用火车售票系统进行信息管理的个人或实体。

列车时刻表：指所有列车到达和离开车站的时间表。

票价：指乘坐火车所需的费用。

订单信息与服务：指用户在火车售票系统上进行的订单信息查看，包括订票、改签、退票以及查看历史订单信息的操作。

售票服务：指为用户提供便捷的购票体验，包括查询车票信息、预订车票、支付购票费用等服务。

本文档中使用的一些缩略语及其含义如下：

CSCI：计算机软件配置项。

API：指应用程序编程接口。

UI：指用户界面。

JDBC：全称是Java数据库连接(Java Database connect)，它是一套用于执行SQL语句的Java API。应用程序可通过这套API连接到关系数据库，并使用SQL语句来完成对数据库中数据的查询、更新和删除等操作

DBA：数据库管理员。

在本文档中，这些术语和定义将会频繁出现，并且对于理解和使用本文档非常重要。因此，我们对这些术语和定义进行了详细的解释，以确保读者能够充分理解这些术语和定义的含义。在本文档的各个章节中，这些术语和定义将被广泛使用，并将被用于描述系统的各个方面和功能，以及系统的需求和设计。

# 附录

附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理，附录可单独装订成册。附录应按字母顺序(A，B等)编排。

暂无，关于该文档写作周期中的其余文档任务参见该文件夹的另一个文档。