**火车票管理系统 软件设计说明**

**v0.4**

**2020.5**

**张洪鑫 周聪 黄臻**

**一、引言与项目背景**

**1.1 编写目的**

向运营本《火车票管理系统》的机关、单位与机构提供必要的技术细节支持，向使用本《火车票管理系统》的用户提供操作指南，向助教清晰地展示项目开发流程与细节。

**1.2 项目名称**

火车票管理系统

**1.3 任务提出者**

上海交通大学ACM班CS147数据结构（2020）课程大作业

**1.4 开发者**

上海交通大学ACM班2019级，张洪鑫、周聪、黄臻

**1.5 参考资料**

<https://github.com/oscardhc/TicketSystem-2020>（以下简称“**作业要求**”）

<http://www.cssmoban.com/cssthemes/8587.shtml>（网页前后端模版）

<https://www.runoob.com/>（JavaScript，jQuery, PHP, html, bootstrap参考资料）

<https://github.com/kuaibiancheng/ThunderClass>（《雷课堂》，清华大学，web socket部分）

<https://github.com/GZzzhsmart/development-document>（国家标准的软件开发文档）

<https://github.com/XunZhiyang/TTRS>

<https://github.com/spectrometerHBH/TTRs>

<https://github.com/ZYHowell/Ticket_Booking_System>

（本开发文档遵循国家GB8567——88开发文档规范。）

# 二、 总体设计

## 2.1 需求规定

“后端程序需要实现文档中要求的接口，在接受完整命令后立即输出结果并再次进入响应输入的状态。使用C++实现，且不能使用外部库或是STL容器。”

“前端部分通过网络后端程序通信，并向用户提供图形化操作界面，使得可以通过此界面实现接口中的任何操作。推荐实现跨平台的前端，如网页，Qt等。”

（引用自“作业要求”，接口要求详见“作业要求”）

## 2.2 运行环境

火车票管理单位侧：一台带公网ip的服务器，操作系统linux或windows皆可（demo以windows server 2016 Data Center为例）。如果需要stand alone的命令行管理，需要鼠标或者键盘等可输入外设。

对于windows server，需要部署的运行环境有：

* nginx
* php并开启socket服务
* Microsoft Visual C++ 2015-2019 Redistributable (x86)
* Microsoft Visual C++ 2015-2019 Redistributable (x64)

详细部署方法见操作手册。

用户侧（pc端）：一台pc，操作系统Linux或Windows皆可（demo以Windows 10 1903为例），仅要求可以通过浏览器（支持JavaScript，demo以Chrome 81.0为例）上网，需要鼠标或者键盘等可输入外设。

用户侧（移动端）：手机等可移动设备，操作系统Android或iOS皆可，仅要求可以通过浏览器上网，需要鼠标或者键盘等可输入外设。

## 2.3 基本设计理念和处理流程

### 2.3.1 设计理念

前端拟采用JavaScript，jQuery，html，bootstrap实现；

后端拟采用php + Nginx实现后端服务器，C++实现后端命令行管理系统与数据库存储。

二者通过web socket（暂定）进行通信。

### 2.3.2 主要功能部分

1. 网页客户端

2. 后端服务器部分

3. 后端逻辑部分

### 2.3.3 基本处理流程框架

1. 用户登陆网页客户端（html与JavaScript实现网页），通过与页面元素的交互向系统提出登陆、查询、增加、删除、修改记录的请求。

2. 请求由浏览器发送到后端服务器php（由php fast-cgi实现）接收到用户的请求。

3. php将用户的请求转换成大作业要求的指定输入通过web socket发送给8888监听端口。

4. 服务器上listener.exe在8888端口监听，接收到用户命令，作为参数调用xxx.exe。

5. xxx.exe得到用户指令进行核心业务逻辑的操作。（由xxx.cpp编译而来，其实也就是数据结构大作业的核心。）

6. xxx.exe处理完请求以后给一个要求的返回值（拟采用cout输出到std）。

7. socket.exe获取得到xxx.exe的cout，通过8888端口返回给后端php。

8. php将返回值转换成符合http前后端数据交流的规范形式（即数据json化）。

9. php把数据发送给前端JavaScript。

10. JavaScript前端接受到数据，逆json化，并按照一定格式显示在客户端的网页上。



socket

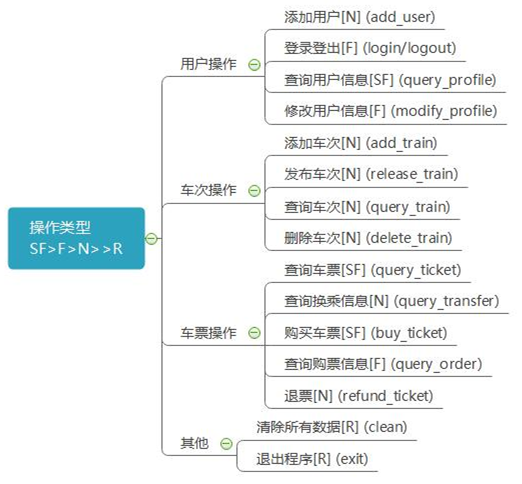
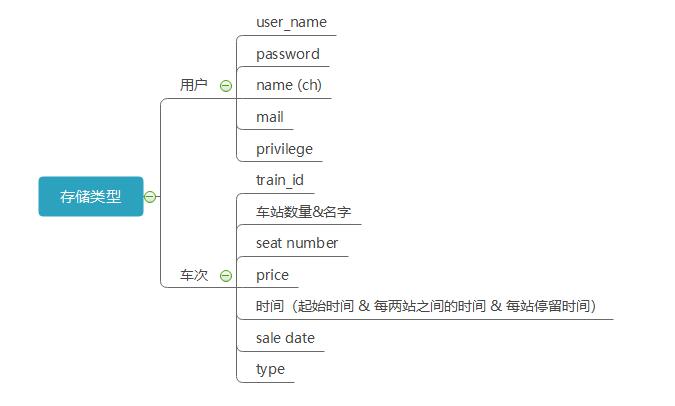
ajax

图1. 基本处理流程框架

\*步骤5与6可以单独运行，此时仅在服务器侧运行一个基于命令行的管理系统，用于维护与管理。

## 2.4 结构

### 2.4.1 操作类型与存储类型



### 2.4.x 中间件listener.exe设计

## 2.5 功能需求与程序的关系

## 2.6 需人工处理的过程

listener.exe需要服务器管理员手动运行，防止监听失效前后端断开与命令行的链接。