



北京交通大学
BEIJING JIAOTONG UNIVERSITY

航空乘客信息管理系统 需求规格说明书

学期：2024-2025 第二学期

编制日期：2025 年 05 月 11 日

编制人：江家玮

学号：22281188

班级：计科 2204

目录

1. 引言	1
1.1 编写目的	1
1.2 项目背景	1
1.3 定义	1
2. 数据库应用系统需求分析	2
2.1 系统业务需求、用户标识与用例图	2
2.2 业务场景分析与泳道图	5
2.3 数据需求分析（数据字典）	6
2.4 系统数据处理需求与数据流图	7
2.4.1 顶层数据流图（0 层图）	7
2.4.2 底层数据流图	8
2.5 系统非功能性需求	9

1. 引言

1.1 编写目的

本需求规格说明书旨在详细定义“航空乘客信息管理系统 (APIMS)”的功能需求、数据需求、非功能性需求以及相关的业务流程。本文档将作为系统设计、开发、测试和验收的主要依据，确保用户及其他相关方对系统需求有一致和清晰的理解。通过明确系统边界、用户角色、核心功能和性能指标，为后续的数据库设计与系统实现奠定坚实基础。

1.2 项目背景

随着航空运输业的迅速发展，高效、准确的乘客信息管理对于提升航空公司运营效率和服务质量至关重要。传统的管理方式往往面临数据分散、处理效率低下、信息更新不及时等挑战。本项目旨在开发一个现代化的“航空乘客信息管理系统”，以满足航空公司在乘客信息登记、查询、修改、删除以及相关业务处理方面的核心需求。该系统基于前期“数据库系统原理”课程实验的成果，进行系统化的需求梳理与规格定义。

1.3 定义

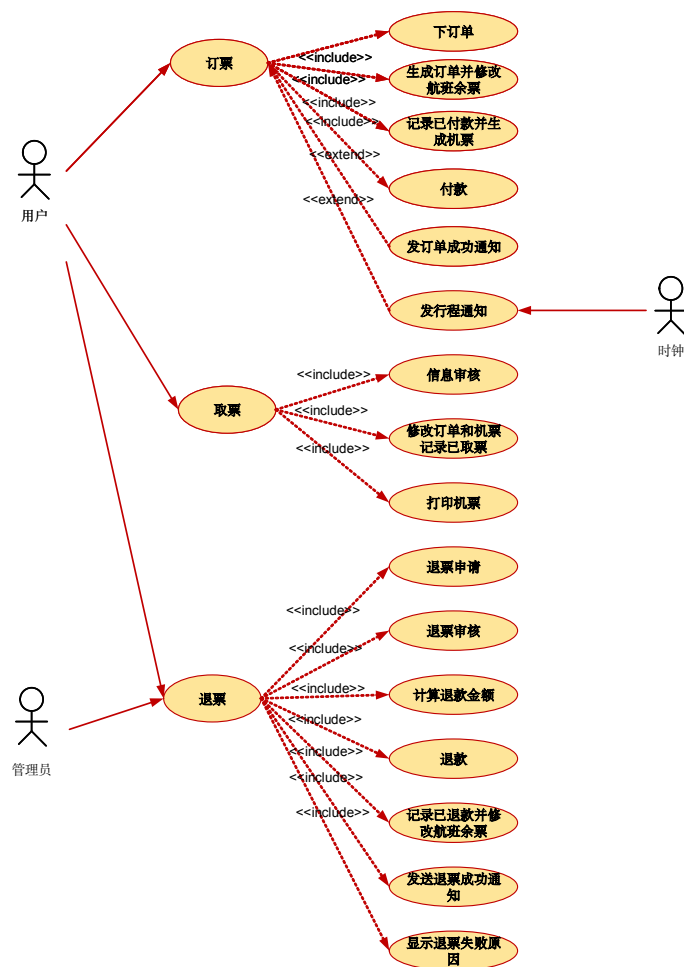
- APIMS: Airline Passenger Information Management System (航空乘客信息管理系统) 的英文缩写。
- 乘客 (Passenger): 指购买或计划购买航空公司机票并接受航空运输服务的个人。
- 票务代理 (Ticket Agent): 指航空公司内部或授权的，负责处理乘客票务相关事宜（如信息录入、查询、机票预订、改签等）的工作人员。
- 航班管理员 (Flight Manager): 指负责航空公司航班计划、航班信息录入与更新、航班状态监控等工作的管理人员。

- 数据分析师 (Data Analyst): 指负责从系统中提取数据、进行统计分析, 并为管理层提供决策支持的专业人员。
- 系统管理员 (System Administrator): 指负责 APIMS 系统日常运行维护、用户账户及权限管理、数据备份与恢复等工作的技术人员。
- CRUD: 数据库操作中创建(Create)、读取(Read)、更新(Update)和删除>Delete)四种基本原子操作的简称。

2. 数据库应用系统需求分析

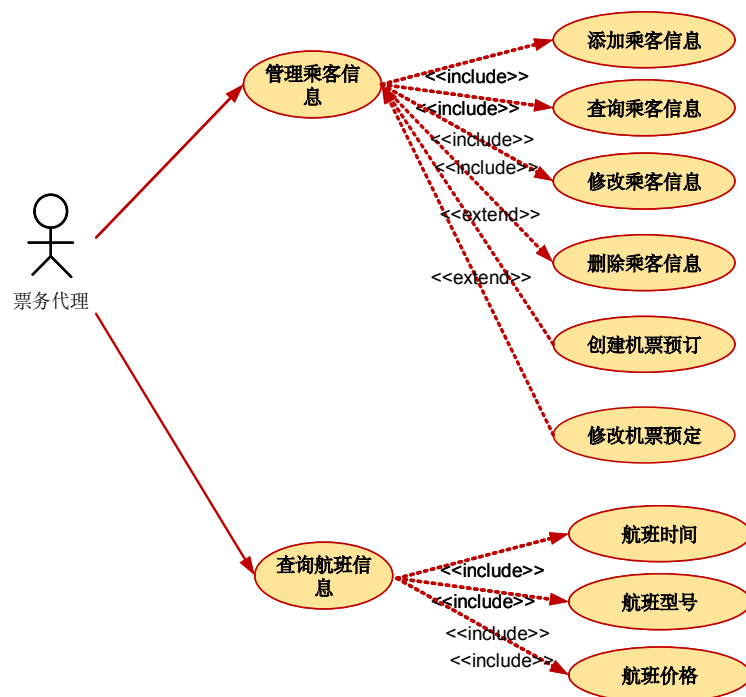
2.1 系统业务需求、用户标识与用例图

系统核心业务是围绕航空公司的乘客信息进行管理。

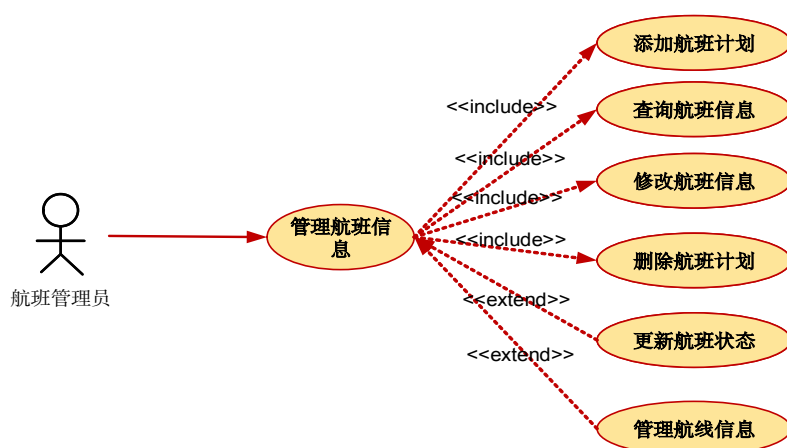


主要参与者 (Actors)有:

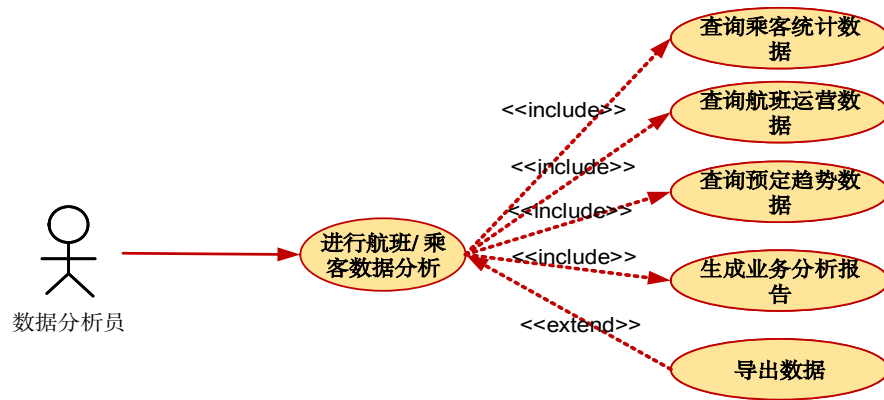
- 票务代理 (Ticket Agent): 负责直接与乘客打交道,进行乘客信息录入、查询、修改、删除,以及机票预订等操作。



- 航班管理员 (Flight Manager): 负责航班信息的管理,包括航班的添加、修改、删除以及航班状态的更新。

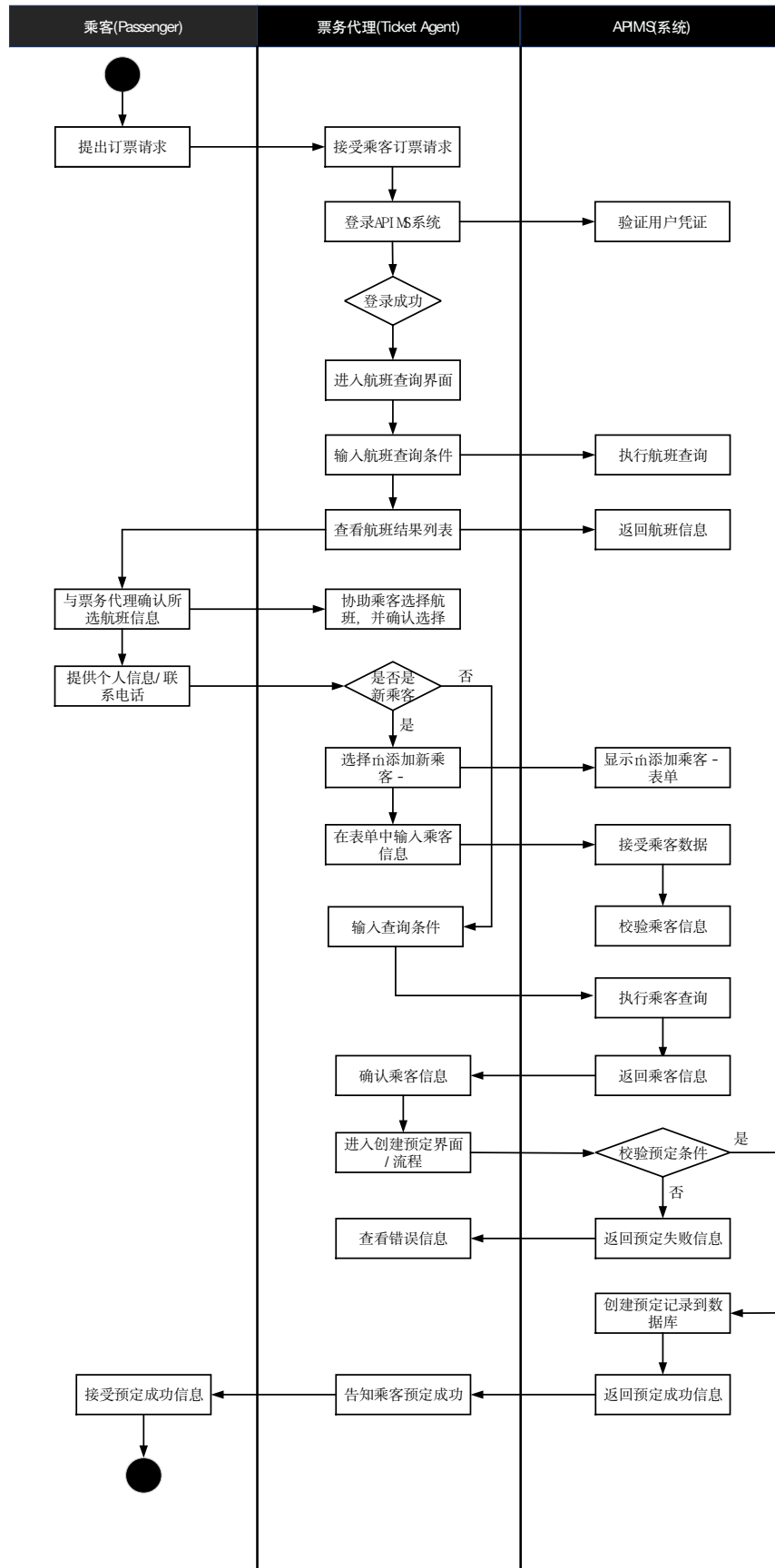


- 数据分析师 (Data Analyst): 负责查询和分析系统中的数据,如乘客统计、航班预订情况等,为决策提供支持。



➤ 系统管理员 (System Administrator): 隐含角色

2.2 业务场景分析与泳道图



选择业务场景：票务代理为新乘客办理机票预订并登记信息。

此场景涉及“乘客”（作为外部信息提供者）、“票务代理”和“APIMS 系统”三方。此泳道图展示了各方在此业务场景中的职责分配和交互顺序。

2.3 数据需求分析（数据字典）

数据项名称 (逻辑名)	英文标识符	数据类型	长度/精度	允许空	约束/说明	来源/去向
乘客 ID	passenger_id	INT		否	主键 (PK), 自增 (通常)	系统生成 / 乘客表
姓名	name	VARCHAR	100	否		用户输入 / 乘客表
身份证号	id_card_number	VARCHAR	18	否	唯一 (UNIQUE), 特定格式 (需应用层校验)	用户输入 / 乘客表
联系电话	phone_number	VARCHAR	20	否	特定格式 (需应用层校验)	用户输入 / 乘客表
电子邮件	email	VARCHAR	100	是	唯一 (UNIQUE, if not null), 邮件格式 (需应用层校验)	用户输入 / 乘客表
常旅客编号	frequent_flyer_number	VARCHAR	50	是	唯一 (UNIQUE, if not null)	用户输入 / 乘客表
航空公司 ID	airline_id	INT		否	主键 (PK) / 外键 (FK) in other tables	系统定义 / 航空公司表
航空公司名称	airline_name	VARCHAR	100	否	唯一 (UNIQUE)	用户输入 / 航空公司表
航班 ID	flight_id	INT		否	主键 (PK)	系统生成 / 航班表
起飞时间	departure_time	DATETIME		否		用户输入 / 航班表
到达时间	arrival_time	DATETIME		否	需晚于起飞时间 (CHECK)	用户输入 / 航班表
预订 ID	booking_id	INT		否	主键 (PK)	系统生成 / 预订表
预订日期	booking_date	DATETIME		否		系统生成 / 预订表
票价	price	DECIMAL	10,2	否	必须为正 (CHECK)	用户输入 / 系统计算 /

						预订表
支付状态	payment_status	VARCHAR	20	否	预定义值 (如 'Pending', 'Paid', 'Cancelled') (CHECK)	用户操作 / 系统更新 / 预订表
用户名	username	VARCHAR	50	否	唯一 (UNIQUE)	用户输入 / 用户表
密码哈希	password_hash	VARCHAR	255	否	存储加密后的密码	系统生成 / 用户表
盐值	salt	VARCHAR	64	否	用于密码哈希	系统生成 / 用户表
角色名称	role_name	VARCHAR	50	否		系统定义 / 角色表

数据存储 (持久化) 需求:

所有业务数据, 包括乘客信息、航班信息、预订信息、用户信息、角色权限信息等, 都需要进行持久化存储。系统将采用 MySQL。数据应存储在服务器端的数据库中, 确保数据的安全、一致和可恢复性。需要定期进行数据备份。

2.4 系统数据处理需求与数据流图

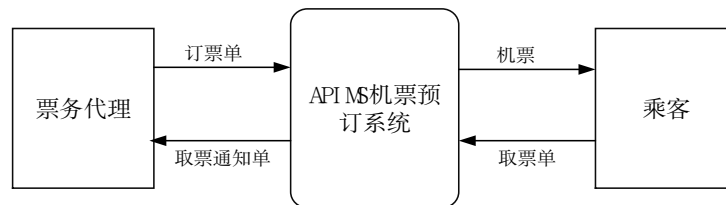
本章节旨在通过数据流图 (DFD) 的方式, 展现“航空乘客信息管理系统 (APIMS)”中的主要数据处理需求、数据流动路径、数据存储以及与外部实体 (如票务代理、乘客) 的交互。数据流图从高层次的上下文视角逐步细化, 以帮助理解系统的的核心。

2.4.1 顶层数据流图 (0 层图)

主要的外部实体及其与系统的数据流如下:

- 票务代理 (Ticket Agent): 作为系统的主要操作用户, 票务代理向“APIMS 机票预订系统”提交“订票单”信息。这股数据流代表了票务代理为乘客创建或管理预订时输入系统的所有必要数据, 例如乘客个人信息、选择的航班、日期、特殊需求等。同时, 系统在处理完相关业务 (如成功出票、取消确认等) 后, 会向票务代理输出“取票通知单”或类似的操作结果反馈。

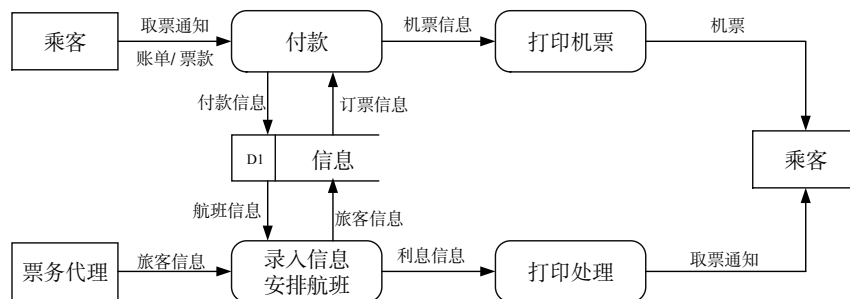
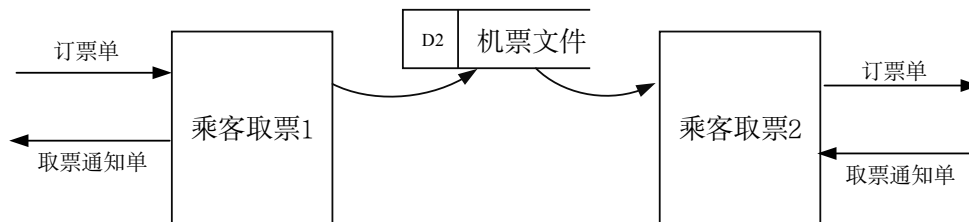
- 乘客 (Passenger): 乘客是航空服务的最终使用者,也是信息的重要来源和去向。乘客会向“APIMS 机票预订系统”提供其个人信息和出行需求,这体现在票务代理提交的“订票单”中,或者在未来可能的乘客自助服务中直接输入。系统处理完成后,会将最终的“机票”信息(电子票或凭证)和“取票单”(或行程确认单)传递给乘客。



2.4.2 底层数据流图

此 1 层图包含以下主要处理、数据存储和数据流:

- 外部实体:
 - ✓ 乘客 (Passenger): 乘客是流程的起点和终点。他们提供“账单/票款”(作为支付输入)和个人信息,并最终接收“机票”和可能的“取票通知”。
 - ✓ 票务代理 (Ticket Agent): 票务代理接收来自乘客的“旅客信息”,并将其输入到系统中进行航班安排。
- 数据存储:
 - ✓ D1 信息 (D1 Information): 这是一个核心的数据存储,用于存放与订票相关的重要信息。它接收来自“付款”处理的“付款信息”和“订票信息”,并向“录入信息安排航班”处理提供“航班信息”、“旅客信息”和“利息信息”。
 - ✓ D2 机票文件 (D2 Ticket File): 这是一个专门用于存储已生成机票信息的数据存储。它从“乘客取票 1”(一个处理环节)接收处理后的订票单信息并存储机票文件,然后将这些机票文件提供给“乘客取票 2”处理环节。



2.5 系统非功能性需求

➤ 性能需求:

并发用户数: 核心业务（票务代理日常操作）峰值并发用户数至少支持 50-100 个。系统需具备扩展能力以应对未来用户增长。

响应时间:

- ✓ 乘客信息列表加载及简单查询：平均响应时间应为 2 秒。
- ✓ 添加、修改、删除乘客等核心写操作：95% 的操作应在 3 秒内完成。
- ✓ 用户登录验证：应在 1 秒内完成。

数据吞吐量: 系统应能有效处理高峰期的数据录入和查询请求，具体指标需根据航空公司日均业务量进一步确定。

资源利用率: 在满足上述性能指标的前提下，服务器 CPU、内存、网络带宽等资源利用率应保持在合理水平（如平均低于 70%），为突发流量留有余地。

➤ 安全性需求:

用户认证： 所有用户必须通过用户名和密码进行身份验证才能访问系统。密码在传输过程中应加密（如使用 HTTPS），在数据库中必须使用强哈希算法加盐存储，防止明文泄露。

授权与访问控制： 系统应实现基于角色的访问控制。不同角色的用户（如票务代理、航班管理员、数据分析师、系统管理员）拥有不同的操作权限，严格限制对数据的访问和修改范围，遵循最小权限原则。

操作审计： 对关键操作（如用户登录、重要数据修改、权限变更）应记录审计日志，便于追踪和安全事件分析。

数据备份与恢复： 必须建立定期的数据备份机制（如每日全量备份，更频繁的增量备份），并制定有效的数据恢复预案，确保在发生灾难时能够快速恢复数据，减少损失。

➤ **完整性需求 (数据层面):**

实体完整性： 所有核心数据表（如乘客表、航班表、用户表）必须定义主键，确保每条记录的唯一性。主键值不能为空。

参照完整性： 表之间的外键关系必须正确定义，并设置合理的参照操作（如 ON DELETE RESTRICT/CASCADE, ON UPDATE CASCADE），确保关联数据的一致性，防止孤立数据的产生。

用户定义完整性：

- ✓ **唯一性约束：** 如乘客身份证号、电子邮件（若非空）、常旅客编号（若非空）、用户名、航空公司名称、票号等必须唯一。
- ✓ **非空约束：** 关键业务字段（如乘客姓名、身份证号、联系电话、航班起飞时间等）不允许为空。
- ✓ **检查约束 (CHECK)：** 对特定字段的值域进行限制，如票价必须为正数，支付状态必须是预定义集合中的一个，航班到达时间必须晚于起飞时间等。

- ✓ **数据类型与格式：** 所有数据项必须符合其定义的数据类型和格式要求（如日期格式、电话号码格式、身份证号格式等），应用层面需进行校验。

➤ **易用性需求：**

用户界面应简洁、直观，符合用户操作习惯。

常用功能应易于查找和操作，减少用户记忆负担。

系统应提供清晰的操作提示和错误反馈信息。

对于复杂操作，可提供向导或帮助文档。

➤ **可维护性需求：**

系统代码应结构清晰、模块化设计，具有良好的可读性和可维护性。

提供必要的开发文档和注释。

配置信息应易于修改和管理。

➤ **可扩展性需求：**

系统架构应具备良好的水平和垂直扩展能力，以适应未来业务量的增长和新功能的增加。

API 接口设计应遵循标准，便于与其他系统集成。

➤ **可靠性/可用性需求：**

核心业务模块的系统可用性应达到 99.9%(年度停机时间不超过约 8.76 小时)。

系统应能从常见的软硬件故障中快速恢复。