



北京交通大学
BEIJING JIAOTONG UNIVERSITY

系统规划与可行性分析报告

航空乘客信息管理系统

学期：2024-2025 第二学期

编制日期：2025 年 05 月 10 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. 项目概述	2
2.1 要求	2
2.1.1 功能	2
2.1.2 性能	2
2.1.3 系统的输出	3
2.1.4 系统的输入	3
2.1.5 处理流程和数据流程	3
2.1.6 可靠性和安全性需求	4
2.1.7 完成期限	4
2.2 项目基本目标	4
2.3 条件、假定和限制	5
2.3.1 所建议系统的运行寿命的最小值	5
2.3.2 进行系统方案选择比较的时间	5
2.3.3 经费、投资的来源和限制	5
2.3.4 硬件、软件、运行环境和开发环境方面的条件和限制	5
2.3.5 可利用的信息和资源	6
2.3.6 系统投入使用的最晚时间	6
2.4 进行可行性分析的方法	6
2.5 评价尺度	6
3. 对现有系统的分析	7
3.1 数据流程	7
3.2 工作负荷	7
3.3 费用支出	7

3.4 人员	8
3.5 设备	8
3.6 局限性	8
4. 所建议的系统	9
4.1 对所建立的系统的说明	9
4.2 处理流程和数据流程	9
4.3 改进之处	9
4.4 影响	10
4.4.1 对设备的影响	10
4.4.2 对软件的影响	10
4.4.3 对用户单位机构的影响	10
4.4.4 对系统运行过程的影响	11
4.4.6 对地点和设施的影响	11
4.4.7 对经费开支的影响	11
5. 可行性分析	11
5.1 技术条件可行性分析	11
5.2 经济可行性分析	12
5.2.1 支出	12
5.2.2 收益	12
5.2.3 投资回报周期	13
6. 社会因素方面的可行性	13
6.1 法律方面的可行性	13
6.2 操作方面的可行性	13
7. 可行性的结论	14

1. 引言

1.1 编写目的

本报告旨在对“航空乘客信息管理系统”（Airline Passenger Information Management System, APIMS）的建设进行全面的系统规划和可行性分析。通过对系统的业务需求、技术方案、经济效益、以及潜在风险进行评估，为项目决策提供依据，明确项目范围、目标和实施策略，确保系统能够成功开发并有效满足航空公司的业务发展需求，提升运营效率和客户服务水平。

1.2 项目背景

随着航空业的快速发展和市场竞争的加剧，高效、准确的乘客信息管理和票务处理能力成为航空公司提升服务质量和核心竞争力的关键。传统的乘客信息管理方式可能存在效率低下、数据分散、信息更新不及时等问题，难以满足日益增长的业务需求和客户期望。本项目“航空乘客信息管理系统”正是在此背景下提出，旨在通过现代信息技术手段，构建一个集中的、自动化的乘客信息与票务管理平台，以优化业务流程，提升服务体验。该系统是基于前期完成的数据库系统原理相关实验中关于航空乘客管理的业务场景和数据库设计进行的深化和扩展。

1.3 定义

APIMS: Airline Passenger Information Management System (航空乘客信息管理系统) 的缩写，指代本报告所规划和分析的数据库应用系统。

CRUD: 指数据库操作中的创建 (Create)、读取 (Read)、更新 (Update)、删除 (Delete) 四种基本操作。

三层架构: 指表示层（用户界面）、业务逻辑层（应用服务器）和数据访问层（数据库服务器）的系统架构模式。

RESTful API: 一种基于 HTTP 协议的软件架构风格，用于设计网络应用程序接口，强调资源、标准 HTTP 方法和无状态通信。

并发用户数：系统在同一时刻能够支持的并行操作的用户数量。

核心业务响应时间：系统完成关键业务操作（如查询、保存）所需的时间。

2. 项目概述

2.1 要求

2.1.1 功能

系统核心功能围绕航空公司的乘客信息管理展开，必须支持对乘客数据的完整 CRUD（创建、读取、更新、删除）操作。具体而言，系统应允许授权用户（如票务代理）添加新乘客信息，包括姓名、身份证号、联系电话、电子邮件（可选）和常旅客编号（可选）等。用户能够根据乘客 ID、姓名或身份证号等条件查询现有乘客信息，并在界面上清晰展示。对于已存在的乘客记录，系统需支持信息的修改和更新。同时，在符合业务规则的前提下（例如，乘客无未处理票务），应能删除乘客信息。此外，系统还需支持用户角色管理和权限控制，如 Lab4 报告中设计的票务代理、数据分析师、航班管理员等角色，确保不同用户只能访问其职责范围内的功能和数据。未来可考虑扩展至航班信息管理、预订管理等模块。

2.1.2 性能

系统性能需满足航空公司的日常运营需求。在并发用户数方面，针对票务代理等内部用户的核心业务操作，峰值并发用户数预计为 50 至 100 人；若未来扩展至乘客在线服务，则需支持更高的并发量。总用户数方面，内部员工用户预计在 100 至 300 人，若扩展，注册乘客用户可能达到数十万甚至百万级别。核心业务响应时间上，乘客信息查询的平均响应时间应控制在 2 秒以内，而添加或修改乘客信息等操作，95% 应在 3 秒内完成。数据存储需具备良好的可扩展性，以应对日益增长的数据量。系统可用性目标为核心模块达到 99.9%。

2.1.3 系统的输出

系统主要的输出包括动态展示的乘客信息列表，该列表通常以表格形式呈现，包含乘客 ID、姓名、身份证号、电话、邮箱、常旅客号等关键信息，以及相应的操作按钮（如编辑、删除）。当用户执行添加、修改或删除操作后，系统应能输出明确的操作结果反馈信息，如“乘客添加成功！”或“删除失败：身份证号已存在”等提示。对于数据分析需求，系统未来可输出各类统计报表或特定数据视图，例如航班座位可用情况、各航空公司航班数量统计等，以 JSON 或可导出文件格式供进一步分析。

2.1.4 系统的输入

系统的主要输入来源于用户的操作。在添加新乘客或修改乘客信息时，用户需要通过前端表单输入乘客的详细资料，如姓名（文本）、身份证号（文本，有特定格式）、联系电话（文本）、电子邮件（文本，可选）、常旅客编号（文本，可选）。查询操作时，用户可能输入乘客 ID、姓名等作为查询条件。删除操作时，输入为指定待删除记录的标识。所有输入数据在提交给后端前，前端会进行初步的格式校验（如必填项、数据格式），后端则进行更严格的业务逻辑校验。

2.1.5 处理流程和数据流程

系统采用经典的三层架构进行处理。用户通过前端表示层（Web 浏览器中的 `passengers.html`）与系统交互。当用户发起操作请求（如保存乘客信息）时，前端 JavaScript 收集表单数据，通过 Fetch API 以 JSON 格式向后端业务逻辑层/应用层（由 Flask 实现的 `app.py`）定义的 RESTful API 接口（如 `/api/passengers`）发送 HTTP 请求（如 POST、PUT）。后端应用服务器接收到请求后，进行数据验证和业务逻辑处理，然后通过 `mysql.connector` 库调用数据访问层，构造并执行相应的 SQL 语句（如 INSERT、UPDATE）与 MySQL 数据库进行交互。数据库执行操作后返回结果，后端应用服务器根据执行结果（成功、失败、错误信息）生成 JSON 响应返回给前端。前端接收到响应后，动态更新页面内容（如刷新乘客列表、显示反馈信息）。

2.1.6 可靠性和安全性需求

系统可靠性方面，核心业务模块的可用性目标设定为 99.9%，这意味着系统需具备较高的稳定性和容错能力，能应对常见的软硬件故障，并通过合理的数据库备份与恢复机制保障数据安全。安全性方面，用户口令必须采用密文存储，并使用如 Argon2 等不可逆加密算法（如 Lab4 中所述）配合盐值进行哈希处理，以防密码泄露。系统需实现基于角色的访问控制（RBAC），为不同用户（如票务代理、航班管理员）授予与其职责相符的最小权限，限制对敏感数据的访问和操作。前后端数据传输应考虑使用 HTTPS 加密。同时，需防范常见的 Web 安全漏洞，如 SQL 注入（通过参数化查询实现）、XSS 攻击等。

2.1.7 完成期限

考虑到系统的核心功能已在实验项目中得到初步实现，若要将其完善为一个可供航空公司内部初步使用的版本，预计需要 3 至 6 个月的开发、测试和部署时间。此期限包括了进一步的需求细化、UI/UX 优化、更全面的错误处理、安全性加固、以及系统文档编写和用户培训。

2.2 项目基本目标

本项目的基本目标是开发一个稳定、高效、安全的航空乘客信息管理系统。通过该系统，旨在实现对乘客信息的集中化、规范化管理，显著提高票务代理等内部员工处理乘客相关业务的效率和准确性。系统致力于优化乘客信息的增删改查流程，提供友好的用户操作界面，减少人工操作的繁琐和错误。长远来看，该系统将为航空公司积累宝贵的客户数据资产，为提升客户服务质量、开展精准营销、以及支持管理层决策提供数据基础，从而增强航空公司的整体竞争力。

2.3 条件、假定和限制

2.3.1 所建议系统的运行寿命的最小值

所建议的“航空乘客信息管理系统”在成功部署后，其最小预期运行寿命为 5 年。在此期间，系统将通过持续的维护和必要的升级来适应业务发展和技术进步带来的变化，确保其能稳定服务于航空公司的核心业务需求。

2.3.2 进行系统方案选择比较的时间

由于本项目是在已有实验项目的基础上进行深化和完善，核心的技术选型（如 Python Flask 后端、MySQL 数据库、HTML/JS/Bootstrap 前端）已经基本确定并经过初步验证。因此，本阶段不再进行大规模的系统方案选择比较。时间将主要用于对现有方案的优化、细节补充以及新增功能的规划上。

2.3.3 经费、投资的来源和限制

项目的经费主要来源于航空公司的年度 IT 预算或专项信息化建设资金。初期投资预估范围在人民币 21.5 万元至 67 万元之间，主要涵盖硬件采购或云服务租赁、必要的商业软件许可、以及开发团队的人力成本。年度运营成本预估在 3 万元至 10 万元。经费的使用将受到公司财务审批流程的限制，并确保投资回报与公司的战略目标相符。

2.3.4 硬件、软件、运行环境和开发环境方面的条件和限制

硬件方面，系统运行需要应用服务器、数据库服务器和 Web 服务器（可合并部署或采用云实例），以及票务代理等用户终端设备。软件方面，服务器需安装相应的操作系统（如 Linux），数据库管理系统选用 MySQL，应用后端基于 Python 和 Flask 框架，前端采用 HTML5、CSS3、JavaScript 和 Bootstrap。开发环境需配置 Python 解释器、Flask 库、mysql-connector-python、文本编辑器或 IDE（如 VS Code）、以及版本控制系统（如 Git）。限制条件可能包括公司现有的

IT 基础设施标准、对开源软件使用的政策以及网络安全策略等。

2.3.5 可利用的信息和资源

可利用的信息和资源包括：前期实验项目中已完成的系统原型代码、数据库设计文档（表结构、约束、示例数据）、以及初步的功能测试结果。技术上，可以充分利用 Python、Flask、MySQL 等成熟开源技术的官方文档、社区支持和丰富的第三方库。团队成员在前期实验中积累的开发经验也是宝贵的资源。此外，可以参考行业内其他航空公司类似系统的公开信息或最佳实践。

2.3.6 系统投入使用的最晚时间

考虑到项目的重要性以及对业务效率提升的迫切需求，系统核心功能模块计划在项目启动后的第 6 个月月底前完成开发和测试，并开始小范围试运行。最晚在项目启动后的第 8 个月，系统应正式投入使用，覆盖所有相关的票务代理和管理人员。

2.4 进行可行性分析的方法

本次可行性分析主要采用综合评估的方法，结合了技术评估、经济评估和操作适用性评估。技术评估基于项目前期实验所采用的技术栈的成熟度、可扩展性、安全性以及团队的掌握程度进行分析。经济评估则通过估算项目的初期投资成本、年度运营成本，并分析其可能带来的直接和间接收益，进行成本效益的初步衡量。操作适用性评估则从用户角度出发，考虑系统的易用性、对现有工作流程的改善程度以及用户培训的需求。分析过程将充分参考项目已有的设计文档和实验报告。

2.5 评价尺度

评价系统方案可行性的主要尺度包括：首先是功能满足度，即系统是否能够完整实现规划的核心业务功能，并满足用户需求；其次是技术可行性，评估所选技术路线的成熟度、稳定性、可扩展性和安全性；再次是经济合理性，通过成本效益分析，判断项目的投入产出是否符合预期，是否具有经济价值；然后是操作

可行性，考量系统是否易于用户学习和使用，能否顺利融入现有业务流程；最后是风险可控性，评估项目在开发和实施过程中可能遇到的风险及其应对措施。

3. 对现有系统的分析

3.1 数据流程

在引入本系统之前，乘客信息的处理可能涉及多个环节和媒介。例如，客户通过电话或柜台订票，票务代理可能首先在纸质表格或本地电子表格中记录客户的基本信息和预订需求。这些信息随后可能需要手动输入到另一个不成体系的预订记录系统，或者仅仅以文件形式存档。数据在不同部门或人员之间流转可能依赖邮件、打印件或口头传递，缺乏统一和实时的共享机制。这种数据流程容易导致信息冗余、不一致和更新延迟。

3.2 工作负荷

由于缺乏集中的自动化工具，票务代理在处理乘客信息和预订时，工作负荷较大。查询特定乘客信息可能需要翻阅大量纸质文件或在多个电子表格中搜索，耗时费力。信息的录入、修改和核对也主要依靠人工，不仅效率低下，而且容易出错。在业务高峰期，如节假日，工作负荷会急剧增加，可能导致客户等待时间过长，影响服务质量。数据统计和报表制作也极为不便，需要人工汇总，难以快速响应管理层的决策需求。

3.3 费用支出

现有系统或手动流程的费用支出主要体现在人力成本上。由于效率低下，需要较多的人员来处理同等数量的业务。此外，纸张、打印耗材等办公费用也是一部分支出。因人工操作失误导致的客户投诉、赔偿或机票作废等，也会间接增加运营成本。如果存在分散的、功能单一的旧软件系统，其维护费用和可能的许可费用也是成本的一部分。

3.4 人员

相关业务人员（如票务代理、客户服务代表）可能需要花费大量时间在繁琐的数据录入、查找和核对工作上，难以投入更多精力在高价值的客户沟通和个性化服务上。员工可能因为工具落后、工作效率不高而产生挫败感。对于新员工的培训，由于流程不规范、信息不集中，上手周期也可能较长。

3.5 设备

现有设备可能主要是员工的个人电脑，用于处理电子表格或访问简单的内部应用。可能缺乏集中的服务器来存储和管理数据，也缺乏统一的数据库系统。网络环境可能仅满足基本的办公需求，未针对高并发的业务系统进行优化。

3.6 局限性

现有系统的主要局限性在于：

第一，信息分散且不一致，乘客数据可能存储在不同的地方，导致数据冗余和冲突，难以形成统一的客户视图。

第二，操作效率低下，大量手动操作和重复劳动浪费了人力资源，延长了业务处理时间。

第三，数据安全性不足，纸质文件易丢失损坏，电子表格权限控制薄弱，敏感乘客信息存在泄露风险。

第四，缺乏有效的数据分析能力，难以从现有数据中提取有价值的洞见来支持业务决策和优化。

第五，难以适应业务发展，无法快速响应新的市场需求或扩展新的服务功能。这些局限性严重制约了航空公司的服务质量提升和运营效率的优化。

4. 所建议的系统

4.1 对所建立的系统的说明

所建议建立的系统是“航空乘客信息管理系统 (APIMS)”。这是一个基于 Web 的数据库应用系统，旨在为航空公司提供一个集中、高效、安全的平台，用于管理乘客的个人信息、联系方式、常旅客信息以及相关的票务操作记录。该系统将通过现代化的技术手段，优化乘客信息的录入、查询、修改和删除流程，提升票务代理等内部用户的工作效率，并为未来的客户关系管理、数据分析和在线服务扩展奠定坚实基础。

4.2 处理流程和数据流程

建议的系统将采用成熟的三层架构。用户通过 Web 浏览器访问前端表示层，该层负责用户界面的展示和交互（使用 HTML, CSS, JavaScript, Bootstrap 实现）。当用户执行操作（如添加乘客）时，前端将收集数据并通过 RESTful API（JSON 格式）向部署在应用服务器上的业务逻辑层发起请求。业务逻辑层（使用 Python Flask 框架实现）负责处理业务规则、数据校验，并构造 SQL 语句。随后，业务逻辑层通过数据库连接器（如 mysql.connector）与数据访问层（MySQL 数据库服务器）通信，执行数据持久化操作（如 INSERT, SELECT, UPDATE, DELETE）。数据库服务器处理请求后返回结果给业务逻辑层，业务逻辑层再将处理结果封装成 JSON 格式返回给前端表示层，前端据此更新界面，向用户反馈操作状态。这一流程确保了数据处理的模块化和各层职责的清晰分离。

4.3 改进之处

相较于可能存在的传统手动或半自动化流程，APIMS 将带来显著的改进。首先，它实现了乘客信息的集中统一管理，解决了数据分散、不一致的问题，为所有授权用户提供了一个单一、准确的数据源。其次，通过自动化的数据录入、查询和校验功能，大幅提升了业务处理效率，减少了人工操作的错误率和工作量。

再次，系统加强了数据安全性，通过用户认证、权限控制以及安全的密码存储机制，保护了敏感乘客信息的安全。此外，规范化的数据存储为后续的数据分析和报表生成提供了可能，有助于管理层进行数据驱动的决策。最后，系统基于 Web 技术，具有良好的可访问性和易用性，便于员工快速上手。

4.4 影响

4.4.1 对设备的影响

部署 APIMS 需要新的硬件设备或对现有设备进行升级。这主要包括应用服务器和数据库服务器，可以选择物理服务器或云服务器实例，其配置需满足系统的性能和并发需求。对于客户端，票务代理等用户需要配备能够流畅运行现代 Web 浏览器的个人电脑。网络基础设施也需要保证足够的带宽和稳定性，以支持系统内外的数据流畅访问。

4.4.2 对软件的影响

系统运行需要操作系统（如 Linux 用于服务器）、MySQL 数据库管理系统、Python 运行环境及 Flask 等相关库。客户端需要标准的 Web 浏览器（如 Chrome, Firefox, Edge）。可能还需要版本控制软件（如 Git）用于代码管理和部署。如果现有系统中存在需要集成的旧系统，可能还需要开发相应的接口软件。

4.4.3 对用户单位机构的影响

APIMS 的引入将对航空公司的相关业务部门（如信息技术部、市场与销售部、运营与客户服务部）的工作流程和角色职责产生积极影响。票务代理的工作将更加高效和规范，可以从繁琐的数据处理中解放出来，更专注于客户服务。IT 部门需要承担系统的开发、部署和日常运维工作。数据管理将更加集中，可能需要明确数据所有权和维护职责。部分岗位的工作内容可能会调整，以适应新的系统和流程。

4.4.4 对系统运行过程的影响

系统投入使用后，原有的乘客信息管理和票务处理流程将被新的、基于 APIMS 的流程所取代。数据录入、查询、修改等操作都将通过统一的系统界面进行。这将要求员工具备基本的计算机操作技能和系统使用能力，因此需要进行相应的用户培训。新的运行过程将更加标准化和透明化，有利于追溯操作记录 and 进行绩效评估。

4.4.6 对地点和设施的影响

由于 APIMS 是一个基于 Web 的系统，用户只要有网络连接和授权，原则上可以在任何合适的办公地点访问系统，对物理地点和设施的特定要求不高。服务器设备可以放置在公司现有的数据中心或选择云服务提供商的设施。主要的设施影响可能在于确保工作区域有稳定的网络接入和电源供应。

4.4.7 对经费开支的影响

APIMS 的建设和运营将带来新的经费开支。初期投资包括硬件采购/租赁、软件许可（若有）、系统开发和部署的人力成本等。运营成本则包括服务器维护、可能的云服务费用、软件年度支持费用、以及 IT 人员的维护成本。但同时，通过提升效率、减少错误、优化决策，系统有望降低长期的运营成本，并带来间接的经济效益，从而对整体经费开支产生积极的净效应。

5. 可行性分析

5.1 技术条件可行性分析

从技术角度评估，“航空乘客信息管理系统”的建设是完全可行的。项目拟采用的技术栈，包括后端的 Python Flask 框架、前端的 HTML/CSS/JavaScript 及 Bootstrap、以及 MySQL 数据库，均为当前业界广泛应用且技术成熟的解决方案。这些技术拥有庞大的开发者社区、丰富的学习资源和完善的生态系统，便于开发

和问题解决。项目的前期实验 (Lab5) 已经验证了基于此技术栈构建核心 CRUD 功能的可行性。系统架构设计为三层结构，清晰明了，易于开发和维护。Flask 框架的轻量级和灵活性使其适合快速开发 API 和 Web 应用，而 MySQL 作为关系型数据库，能够提供稳定可靠的数据存储和高效查询能力。对于性能要求，通过合理的数据库设计（如索引优化）、高效的 SQL 编写以及未来可能的架构优化（如负载均衡、缓存机制），可以满足预期的并发用户数和响应时间。安全性方面，可以采用参数化查询防止 SQL 注入，对用户输入进行严格校验，密码采用哈希加盐存储，并实施基于角色的权限控制，确保系统和数据的安全。系统的可扩展性也较好，未来可以方便地增加新功能模块或与其它系统集成。

5.2 经济可行性分析

5.2.1 支出

项目的经济支出主要包括一次性的初期投资和持续的年度运营成本。初期投资估算在人民币 21.5 万元至 67 万元之间，具体构成包括：硬件设备（服务器、网络设备等）的采购或云服务初装费，约为 5 万至 20 万元；商业软件许可费用（如操作系统、数据库商业版等，若选择付费版本），约为 1 万至 5 万元；系统开发的人力成本，按照 3-6 个月开发周期、5 人团队估算，约为 15 万至 40 万元；以及员工培训等其他费用，约为 0.5 万至 2 万元。年度运营成本预估在人民币 3 万元至 10 万元，主要包括硬件维护与折旧、软件许可续费、服务器托管或云服务年度费用、以及 IT 支持人员的部分薪资和日常运维开销。

5.2.2 收益

APIMS 的收益可以从直接和间接两个层面来看。直接收益可能体现在：由于票务处理效率的提升，单位时间内能够完成更多的销售，从而增加销售收入；如果未来系统扩展支持在线直销，则可以减少对第三方分销渠道的依赖，降低佣金支出。间接收益则更为广泛和深远，包括：通过自动化流程和减少人工错误，显著降低运营差错成本和人力成本；提升客户服务响应速度和准确性，从而提高客户满意度和忠诚度，减少客户流失，并通过良好口碑吸引新客户；系统积累的乘

客数据和航班数据,经过分析可以为航线优化、市场营销策略制定提供数据支持,提升决策质量,进而提高航线的盈利能力和市场竞争力;同时,标准化的系统操作也能提升员工的工作效率和满意度。

5.2.3 投资回报周期

精确计算投资回报周期 (ROI Period) 较为复杂,因为它依赖于收益的具体量化和时间的推移。考虑到系统带来的效率提升、成本降低以及潜在的销售增长,预计系统在投入使用后的 2 到 4 年内可能收回初期投资。初期的主要回报将体现在运营效率的提升和差错成本的减少上。随着系统的稳定运行和数据的不断积累,其在辅助决策和提升客户价值方面的作用将更加凸显,从而带来更长期的经济效益。对于一个中型航空公司而言,这样的投资回报周期通常是可以接受的。

6. 社会因素方面的可行性

6.1 法律方面的可行性

在系统设计和运营过程中,必须严格遵守国家及地区关于个人信息保护的相关法律法规,例如中国的《个人信息保护法》等。系统需要明确乘客数据的收集、存储、使用和销毁规则,确保获取用户数据的合法性,并采取必要的技术和管理措施保护乘客隐私不被泄露或滥用。在用户注册和信息收集环节,应提供清晰的隐私政策,并获得用户的明确同意。对于数据的跨境传输,也需遵守相关规定。只要在开发和后续运营中充分考虑并落实这些法律要求,系统在法律方面的风险是可控的。

6.2 操作方面的可行性

从操作层面来看,该系统的可行性较高。系统设计采用用户友好的 Web 界面(借助 Bootstrap 框架),力求操作直观简便,降低用户学习成本。票务代理等主要用户群体具备一定的计算机操作基础,经过适当的培训后应能较快掌握系统的使用方法。系统将优化现有的工作流程,使其更加标准化和高效,预计能够

得到用户的积极配合。在系统上线初期，提供详尽的用户手册、操作视频和现场支持将有助于用户平稳过渡。通过收集用户反馈并进行持续改进，可以进一步提升系统的易用性和用户满意度，确保系统能够顺利融入日常业务操作中。

7. 可行性的结论

综合以上对“航空乘客信息管理系统 (APIMS)”的技术条件、经济效益、法律合规性以及操作适用性等方面的分析，可以得出以下结论：该项目的建设是可行的。

在技术上，所选用的技术栈成熟稳定，能够满足系统功能和性能的基本要求，并具备良好的可扩展性和安全性，开发团队基于前期实验已具备一定的技术基础。

在经济上，虽然项目需要一定的初期投资和持续的运营成本，但通过提升运营效率、降低差错成本、改善客户服务以及辅助管理决策，系统预期能带来显著的直接和间接收益，具有良好的投资回报前景。

在社会因素方面，只要严格遵守个人信息保护等相关法律法规，并重视用户培训和系统易用性设计，系统在法律和操作层面上的风险是可控的，能够被用户所接受并有效应用于实际工作中。

因此，建议批准“航空乘客信息管理系统”项目的立项，并投入资源进行后续的详细设计、开发、测试和部署工作，以期通过该系统提升航空公司的整体运营水平和市场竞争力。

8. 专家论证

为了进一步验证“航空乘客信息管理系统”方案的可行性，我邀请了两位同学作为业务专家，对本方案进行了论证。

8.1 专家 1

姓名：王祺

论证日期：2025 年 5 月 10 日

专家论证建议：

王祺同学提出了以下建议：

数据校验与容错处理：建议加强前端和后端的数据校验，特别是对身份证号、电话号码、邮箱等格式的检查要更加严格。对于重要的操作（比如删除乘客信息），建议增加二次确认机制，并且记录操作日志，方便后续追溯和恢复数据。另外，后端在处理 API 请求时，应该更加细致地捕捉异常并反馈给前端，帮助开发人员更清晰地知道问题所在。

专家论证结论：

王祺同学认为，系统整体设计合理，技术方案主流且可行。如果加强数据校验，并逐步增加数据分析功能，系统会有更好的前景和实际应用价值。

结论：方案基本可行，建议根据上述意见进行优化后实施。

8.2 专家 2

姓名： 杜玮林

论证日期： 2025 年 5 月 10 日

专家论证建议：

杜玮林同学对方案提出了以下建议：

业务流程集成度：目前系统主要集中在乘客信息管理。建议将来考虑将系统与航空公司其他核心业务系统（如航班管理、票价管理、支付系统等）进行集成。通过 API 接口实现数据共享和流程联动，可以进一步提高整体运营效率。

专家论证结论：

杜玮林同学认为，系统在解决核心问题是可行的，技术架构也有良好的基础。

结论：方案还行，但在项目实施过程中，需要特别关注与其他业务系统的集成