# 航空票务管理系统开发报告

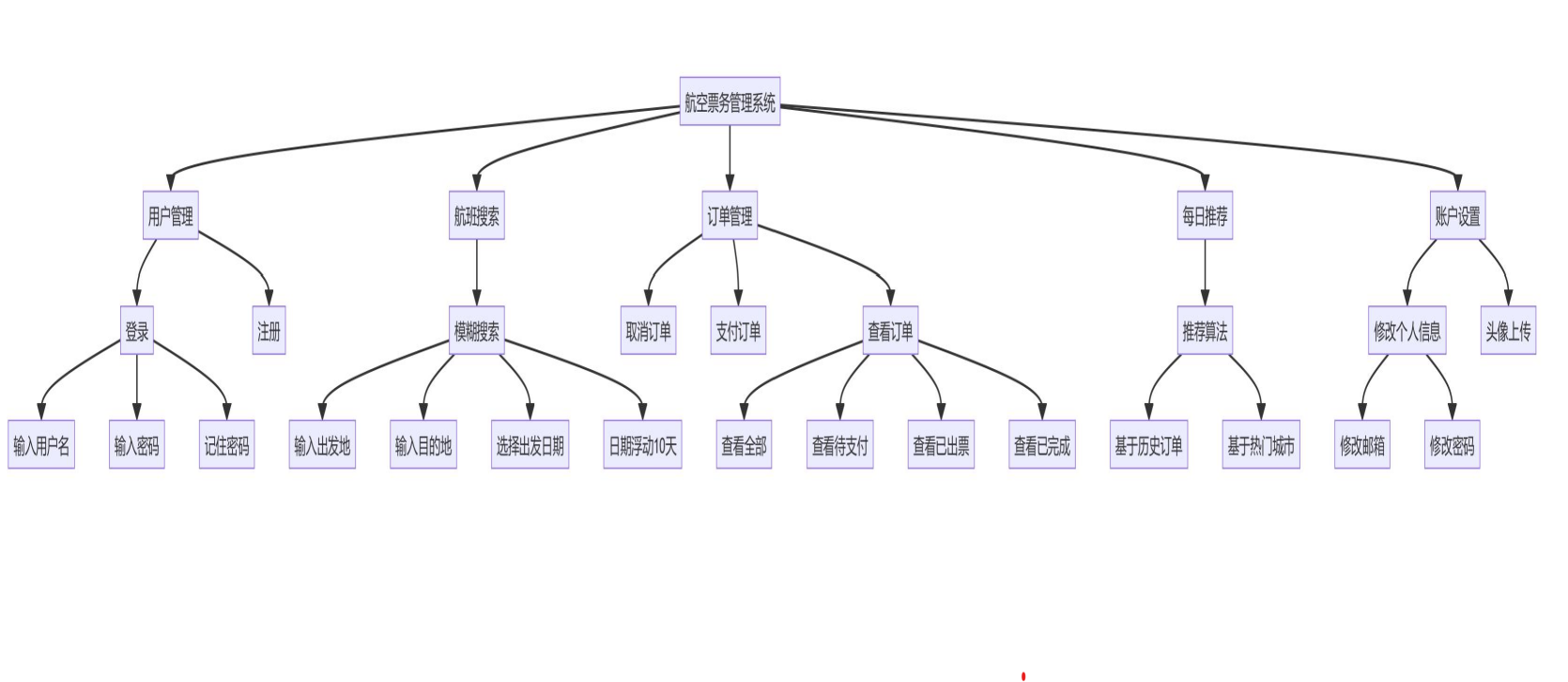
1. **项目概述**

**1.基本内容**

本项目旨在开发一个用户友好的航班预订和管理系统，支持用户登录、注册、订单搜索、订单管理、每日推荐、账户设置等功能。系统采用C++和Qt框架进行开发，数据库使用MySQL进行数据存储和管理。

1. **业务需求分析和基本模块功能**

**1.业务需求分析**

用户可以进行登录和注册，在登录成功后需要能根据始发地目的地和出发日期来搜索航班，同时支持在搜索界面下订单，然后能在订单管理界面实现订单的支付，出票和退票等功能，同时，我们还针对用户个性化设置了**航班推荐功能**，能根据用户和大众喜好为用户推荐航班，最后该系统还允许用户对自己的个人信息作出修改。

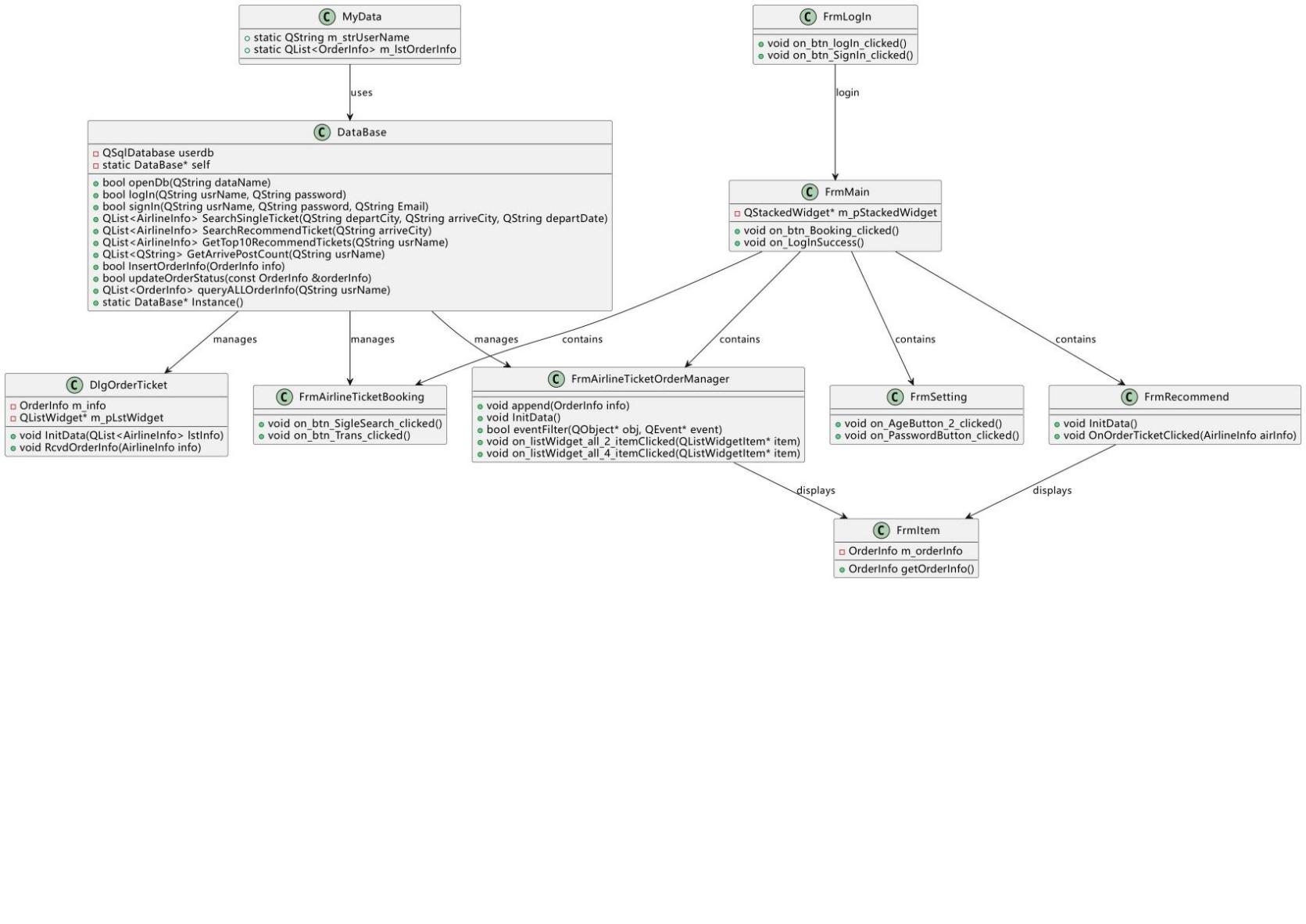
图 1 系统功能图

图 2 对象抽象和类层次图

**2.基本模块功能**

**（1）****登录和注册**

**功能描述**：用户可以通过用户名、密码和邮箱进行登录和注册。支持密码加密存储，确保用户数据安全。

**技术实现**：使用Qt的QLineEdit和QPushButton实现用户输入和交互。使用MySQL存储用户信息，密码采用SHA-256加密算法存储。登录时通过数据库查询验证用户信息。

**（2）订单搜索**

**功能描述**：用户可以根据出发地、目的地、出发日期等条件搜索航班。支持模糊搜索和高级搜索功能。

**技术实现**：使用Qt的QComboBox和QLineEdit实现搜索条件的输入。通过MySQL的LIKE语句实现模糊搜索，通过多条件组合查询实现高级搜索。搜索结果通过QTableView展示，支持分页显示。

**（3）订单管理**

**功能描述**：用户可以查看、修改和取消已预订的航班订单。支持订单状态的实时更新。

**技术实现**：使用Qt的QListWidget或QTableView展示订单列表。通过MySQL的事务机制确保订单状态的实时更新。提供修改和取消订单的按钮，点击后触发相应的数据库操作。

**（4）每日推荐**

**功能描述**：系统根据用户的历史订单和偏好，每日推荐合适的航班。支持推荐信息的展示和交互。

**技术实现**：使用简单的推荐算法，根据用户的历史订单生成推荐列表。推荐算法通过查询用户的历史订单，匹配相似的航班信息。推荐结果通过QListWidget展示，支持点击查看详情。

**（5）账户设置**

**功能描述**：用户可以修改个人信息，如邮箱、密码等。支持头像上传和显示。

**技术实现**：使用Qt的QLineEdit和QPushButton实现信息修改。头像上传通过QFileDialog选择本地文件，使用QPixmap显示头像。修改后的信息通过MySQL更新到数据库。

**（6）数据库连接和操作**

**功能描述**：使用MySQL数据库进行数据存储和管理。支持数据的增删改查操作。

**技术实现**：使用Qt的QSqlDatabase和QSqlQuery实现数据库连接和操作。数据库设计包括用户表、订单表、航班表等，支持复杂查询和事务处理。数据库操作通过封装的数据库类实现，确保代码的可维护性和可扩展性。

**三. 系统设计与实现**

**1.计算机体系设计（采用技术和平台）**

**开发语言**：C++

**开发框架**：Qt 6

**数据库**：MySQL 8.0

**操作系统**：Windows 10

**开发工具**：Qt Creator

**2.功能详细设计与数据库字典**

**（1）整体界面框架设计：**本项目的主体是基于stackwidget实现的，通过订单搜索，订单管理，每日推荐和用户设置分别在单独的一个widget中实现，通过点击相应button设置stackwidget的currentindex从而实现功能的切换。同时基于使用习惯，我们强制用户在使用这些功能前必须登录，未登陆前会隐藏stackwidget。

1. **多界面、组件ui设计互联：**如设计了FrmItem、Form，其是一个自定义的QWidget，它可以接收一个 OrderInfo 对象，并使用这个对象来设置界面元素的值，显示多个订单的状态及预定等。FrmMain主窗口使用 QStackedWidget 来管理不同的功能模块，以及使用 Qt 的信号和槽机制来实现不同组件之间的通信，如FrmLogIn 有一个登录的信号，当用户成功登录后，FrmMain 类连接了这个信号到 on\_LogInSuccess 在登录成功后执行特定的操作、在FrmAirlineTicketBooking有一个信号，当用户完成机票预订后传递一个 OrderInfo对象，FrmAirlineTicketOrderManager 类的 append 槽函数连接到这个信号，用于将新订单添加到订单管理界面中等.

**（3）数据库设计：**本项目的数据库主要涉及四张表：

**airdata(所有航班信息表)**：用于存储近几个月全国各大航空公司开设的航班信息，包括航班编号，飞机型号，机场信息，始发和到达时间，登机口，准点率等信息。

**citycode(城市编号表)**：包含所有航班设计的城市名及其编号。

**orderinfo(用户订单表)**：用于存储用户订单信息，除了airdata所包含的信息外还存储了订单的状态（待支付，已出票，已完成）。

**user\_table(用户信息表)**：用于存储用户信息，包括姓名，密码，邮箱信息。

此外本项目专门设计了一个database类，用于实现项目中涉及的各种数据库（增删查改）操作。

**（4）登录和注册**

本项目的登录和注册的界面搭建分别由类FrmLogIn和FrmSignIn实现，登录和注册逻辑均在database中实现，其中值得注意的，我们对用户注册时输入的密码进行了SHA-256

盐值加密处理（加密算法也在database中实现），其在数据库中显示的是处理后的密码，同时在登录时也是将用户输入的密码与数据库中经过还原的密码进行对比来判断是否登录成功。除此之外，我们在登陆界面还设置了一个记住密码的checkbox，用户在勾选登录再退出后，密码会直接在登录界面显示，便于下次直接登录。

**（5）订单搜索功能**

由于航班信息过于复杂，因此本文设计了一个航班结构体AirlineInfo来存储航班信息，具体到订单搜索界面，我们设计了两个下拉框，支持用户选择citycode中的所有城市，同时支持下拉选择和键盘输入，同时设计了一个changebutton实现始发地和目的地的一键切换。搜索算法则在database中实现，为了避免航班过少而导致搜索结果无法满足用户需求，我们通过MySQL的LIKE语句实现了模糊搜索，当用户选定出发日期后，我们会为用户显示选定日期前后十天的符合条件的航班，为用户提供多样化的选择，同时用户可以在搜索结果页面直接下订单，稍后再在订单管理界面进行支付。

**（6）订单管理功能**

本文设计了一个订单结构体OrderInfo用于存储用户的订单信息，该界面是基于tabwidget实现的，总共分为“全部订单”“待支付”“已出票”“已完成”四个板块，全部订单板块用于显示用户已经支付过的所有订单，用户可以在待支付板块支付订单，支付过的订单会在已出票界面显示同时支持在已出票界面退票，当航班完成时间超过系统时间后，该订单会在已完成界面显示。

**（7）每日推荐功能**

该功能是本项目的一大特色，本项目考虑到用户有旅游的需求但是可能又不知道去哪里玩，因此设计了一个推荐算法来为用户个性化推荐航班。该算法在database中实现，其具体实现逻辑是：首先查询用户的历史订单，选出其最常出发的两个城市，若用户暂无历史订单，则会从10个热门城市中随机推荐两个作为出发地，然后我们会遍历orderinfo表，统计所有用户的订单数据，得出国内旅客最常去的10个旅游城市作为目的地，若不足十个则从国内著名大城市中补全。最后结合用户历史出发地和热门目的地查询航班信息，为用户推荐热门航班，从而实现为每个用户个性化推荐航班。

**（8）账户设置功能**

该部分的主要功能是显式显示用户的邮箱信息，隐式显示用户的密码，初始状态下这两项内容都不可编辑，只有点击对应的修改按钮后才可对内容进行编辑，再次点击按钮保存即可实现对账户信息的更改。具体的数据库操作由database实现。

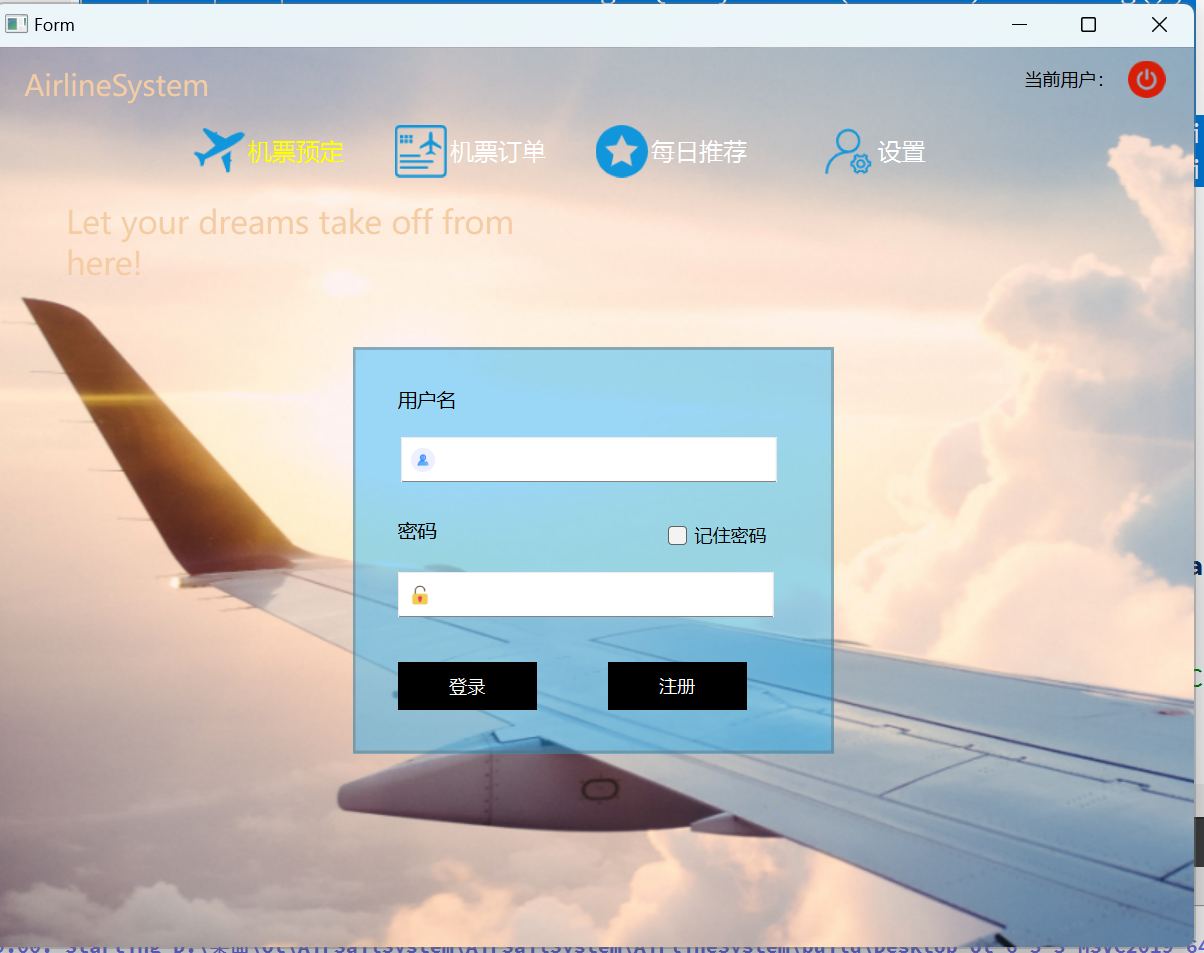
**2.编码实现**

本报告主要展示推荐算法的实现，具体的实现逻辑已经上文介绍，以下是代码截图：

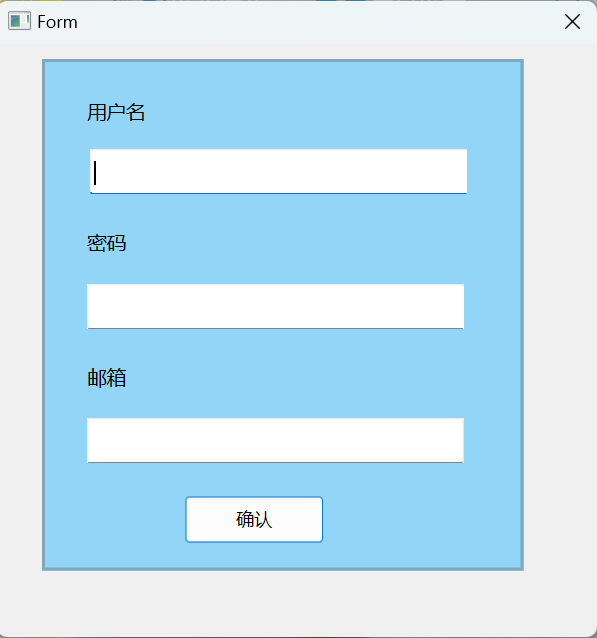


**图 3推荐算法编码实现**

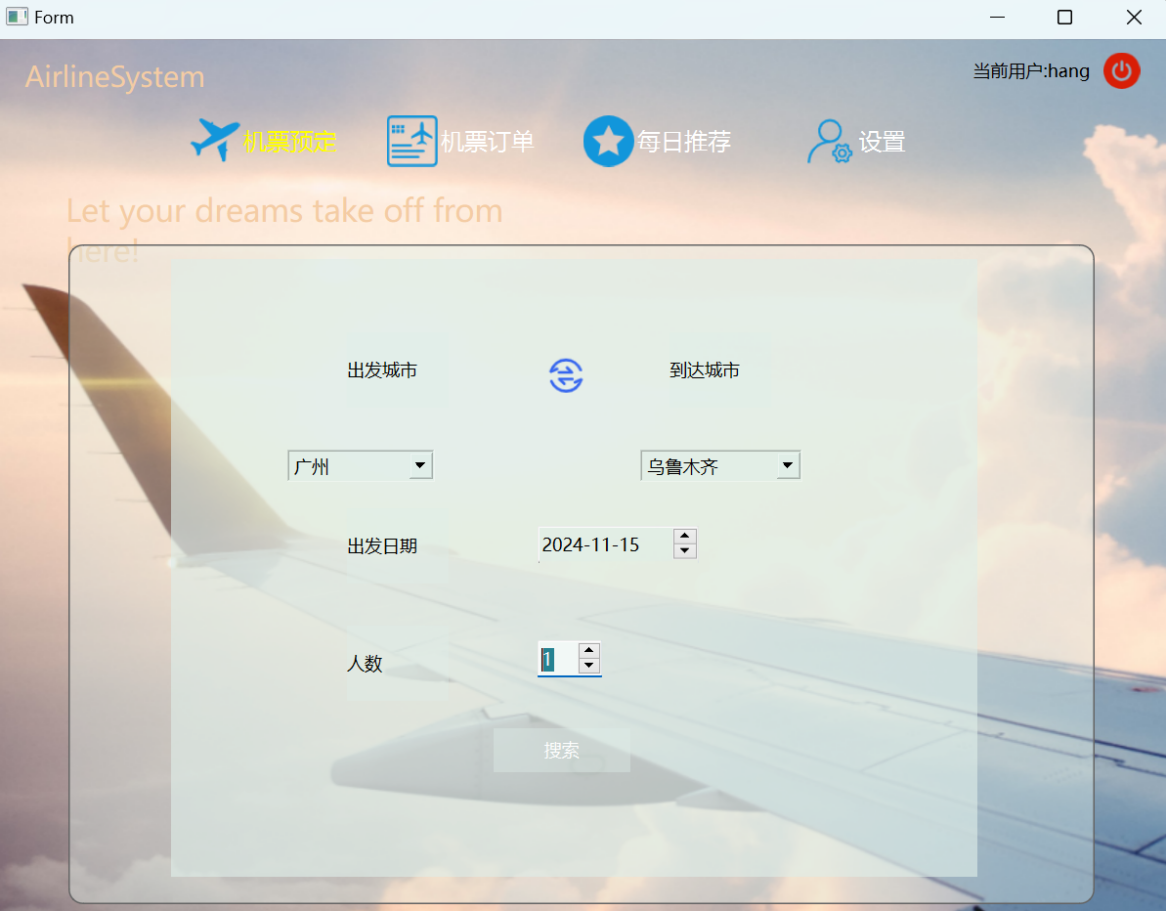
**四．系统测试**



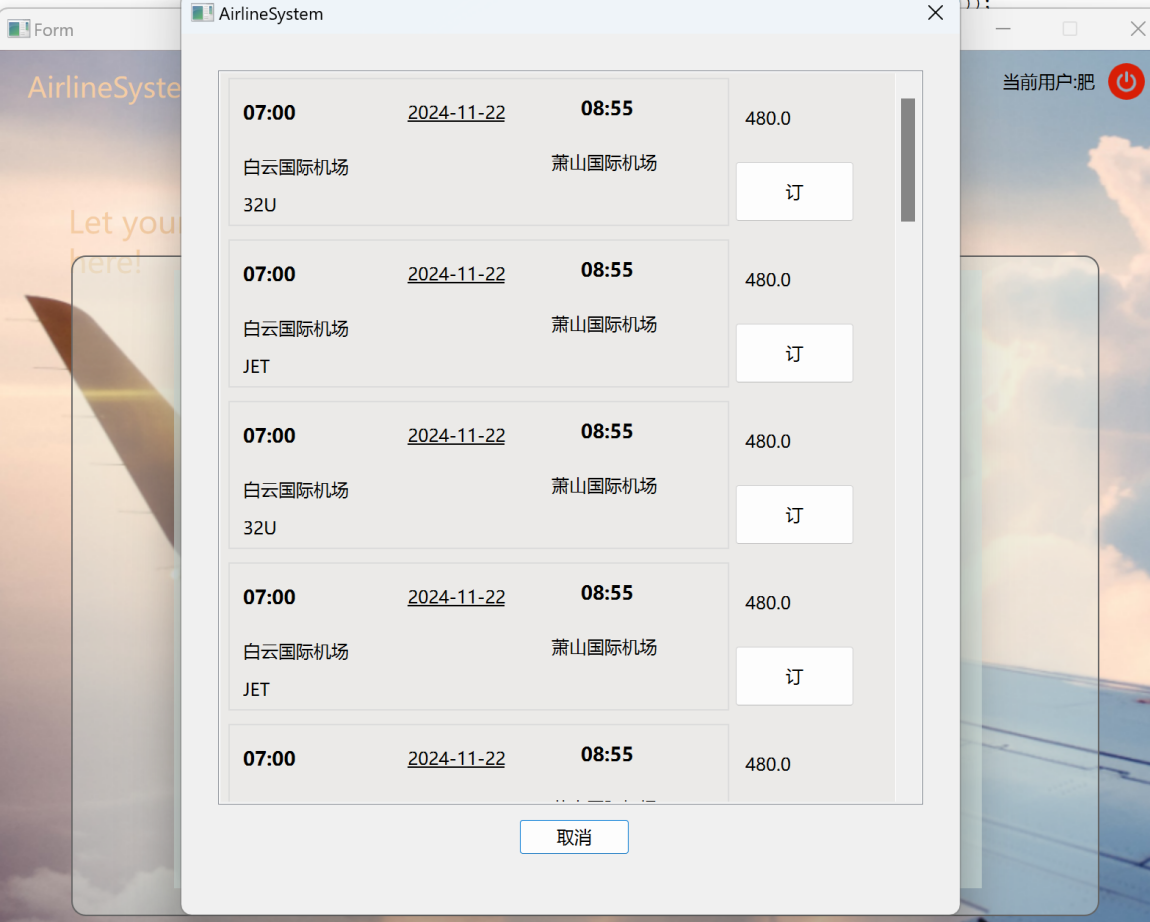
**图 4登录界面测试**



**图 5注册界面测试**



**图 6搜索界面测试**



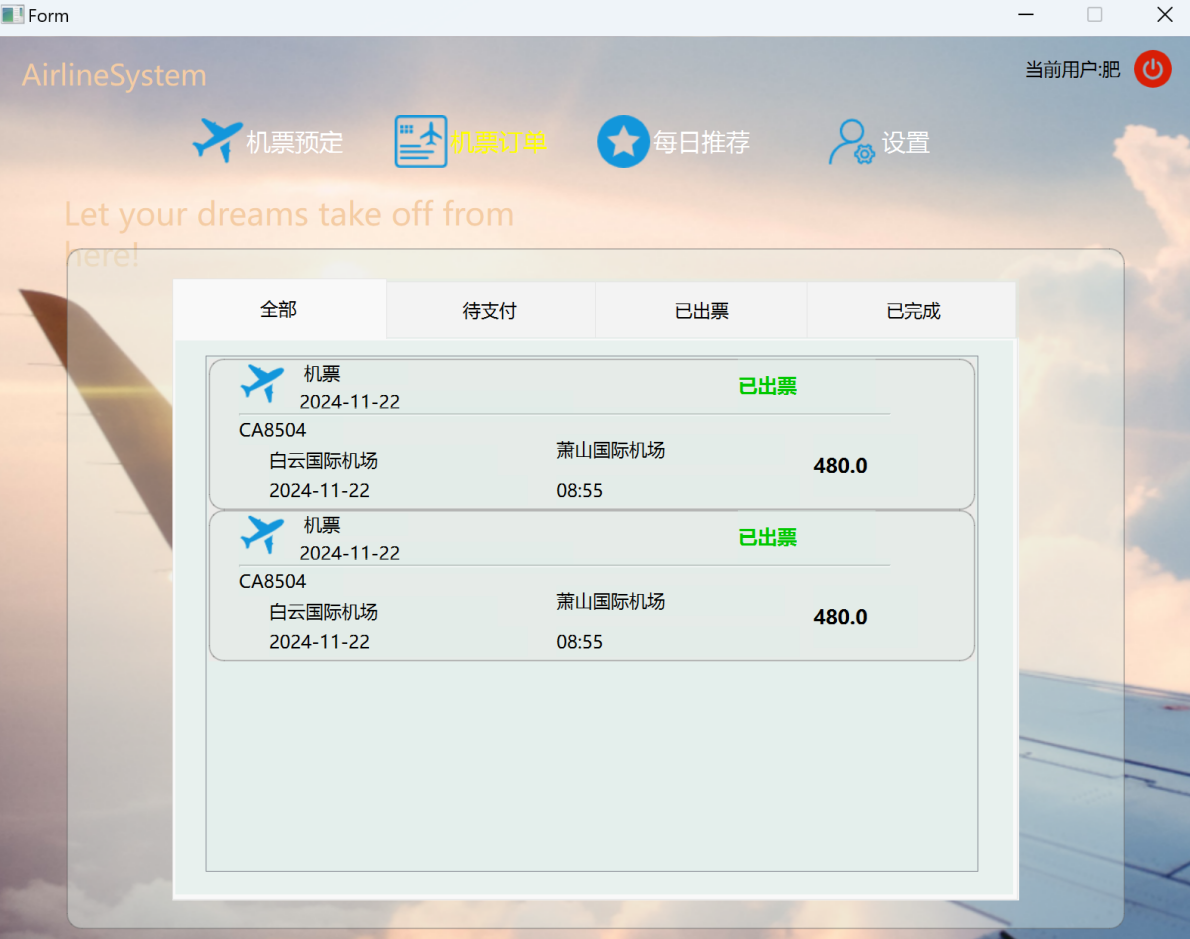
**图 7搜索结果测试**



**图 8订单管理界面测试1**



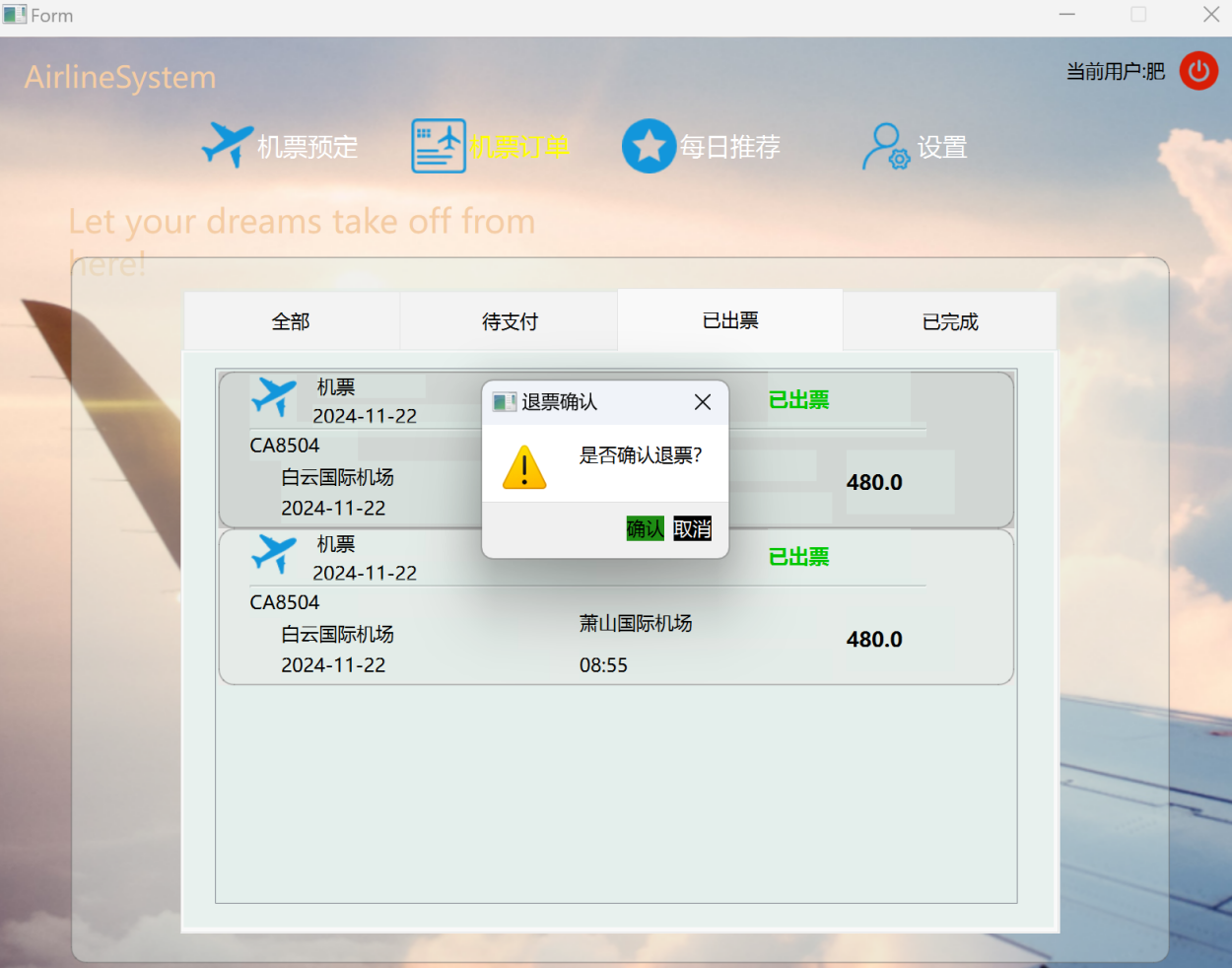
**图9订单管理界面测试2**



**图10订单管理界面测试3**



**图11订单管理界面测试4**



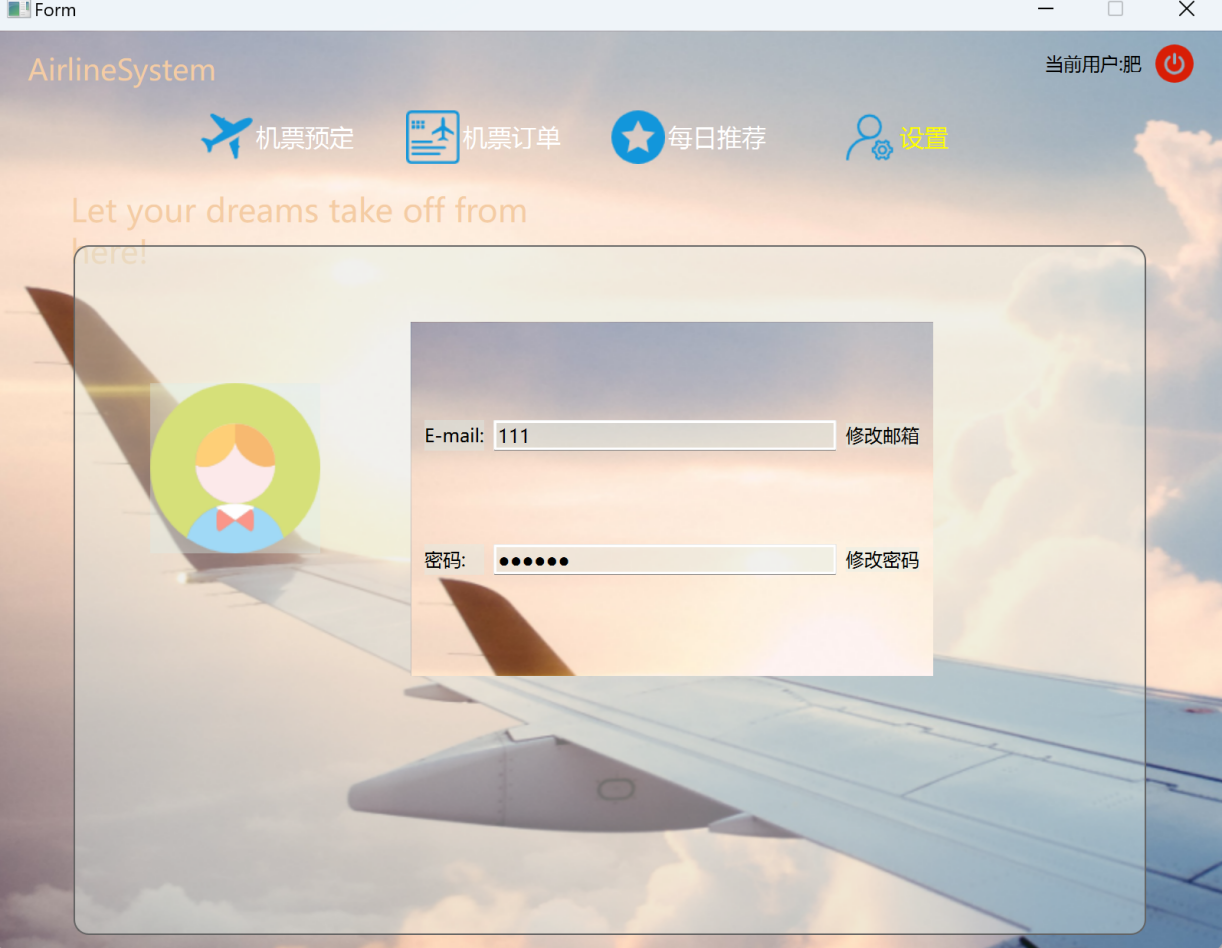
**图12订单管理界面测试5**



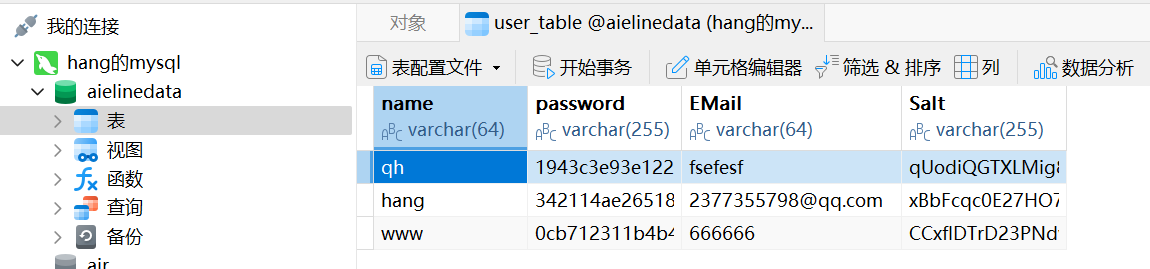
**图13航班推荐测试**



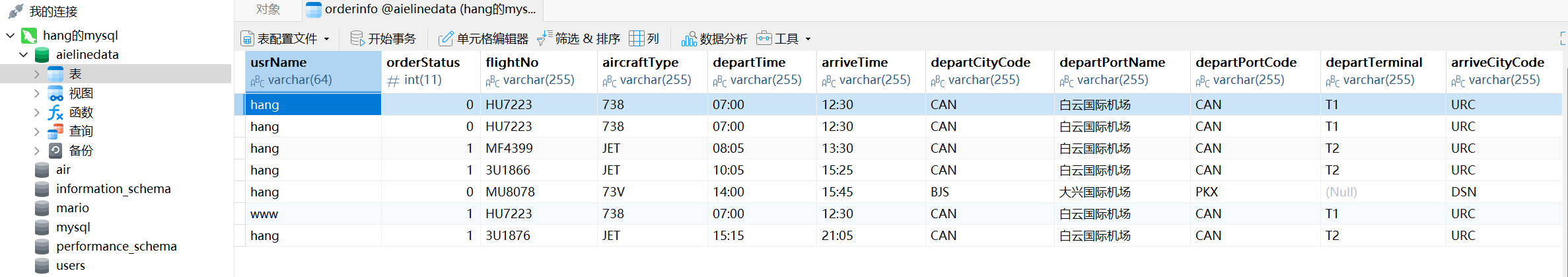
**图14航班推荐测试**



**图15账户设置测试**

****

**图16密码加密展示**



**图17用户订单展示**

**五．项目展望**

**未来改进方向**

1. **增加移动端支持**：开发移动端应用，提升用户体验。
2. **优化推荐算法**：引入机器学习算法，提升推荐准确性。
3. **增强安全性**：增加多因素认证，提升系统安全性。