

## 实践报告（计算机实习）

课程编号：3152100601    实践课程名称：计算机实习    学年：2    学期：秋

<b>学生姓名</b>	吴郑啸	<b>学号</b>	2020213235
<b>指导教师姓名</b>	计算机学院 刁婷;软件学院 孙鹏飞,杨金翠,李峥,李晶。	<b>起止时间</b>	2021 年 8 月 30 号~9 月 12 号
<b>项目名称</b>	航空售票系统		
<b>项目内容 (200字左右)</b>	<p>本组严格按照课件的要求开发了一个航空售票系统，使用了 SQLite 进行数据管理、MFC 进行更友好的图形界面绘制，实现了管理员和购票用户的各种功能。两者均有登录，信息修改功能，与账户有关的航班的信息展示，按要求排序，按起飞地点筛选功能。用户部分实现了机票购买、订单管理，管理员部分实现了航班添加和订单总览等功能。各类输入都有相应的错误检测，如添加的航班飞行员冲突、用户购票冲突，信息修改时输入的信息合法性等。本项目清晰地定义了各功能的接口并分源文件编写，提高了团队效率，增强了可维护性。</p>		
<b>简要个人成果 (200字左右)</b>	<p>作为本组组长，我在立项时即编写了 API 文档，用于组内异步的开发和调试，使得各模块在主程序还只有雏形时即可充分调试纠错。我不仅需要完成自己的 SQL 数据库接口部分和与 MFC 的连接工作，还积极与组员沟通，解答大家的疑惑，或是根据要求新增相应的接口。作为数据库部分的负责人，设计一个合理的数据库表单也是重点所在。同时，我还负责将各个组员的模块进行整理，统合到整个工程当中，并努力解决 CLI 与 GUI 思路不同造成组员编写的差异。</p>		
<p><b>评语</b></p>			
<p><b>成绩（百分制）：</b></p> <p style="text-align: right; margin-top: 20px;"><b>指导教师签字：</b></p> <p style="text-align: right; margin-top: 10px;">年      月      日</p>			

# 计算机实习

2021-2022 学年第 1 学期

## 项目总结报告

(每人一份)

项目完成人：

姓名： 吴郑啸 学号： 2020213235

日 期： 2021 年 09 月 12 日

## 一 . 项目开发及运行环境（说明本项目需要的环境）

系统环境: Windows 10 Pro x64 21H1;

IDE 环境: Visual Studio 社区版 2019, MSVC v142, 安装了“使用 C++ 的桌面开发”及其中的“适用于最新 v142 生成工具的 C++ MFC (x86 和 x64)”模块;

附加库: SQLite 3.36.0; (已包含在项目中, 无需额外添加)

主要开发语言: Visual C++, SQL;

SQLite 可视化分析工具 (可选): DB Browser for SQLite;

## 二 . 项目内容（说明个人负责的内容部分，如：问题分析、设计方案、算法、设计图、功能分析、系统设计等）

由于本组采用按层级递进分配的方式，部分组员的负责内容可能涉及诸多模块，全部写下过于繁琐，在此请允许我以用户的订单查询、管理部分为例。

### 1. 数据库环境与表单设计

问题分析:

需要一个能便利地存储和查询用户信息、管理员信息、航班信息和订单信息的数据存储模块，并且与本项目契合度较高。

设计方案:

使用 SQLite 进行数据库管理。在 Visual Studio 的项目属性中添加相应的位置即可完成开发环境配置（该项目中已包含）。数据库文件存储位置:

MFCApplication1\bupt44Database.db; 若直接运行可执行文件，需将该数据库放置于与可执行文件相同的位置。

数据库包含表单: Admin, User, Flight, Orders;

- Admin: UID 用户的内部编号, ID 用户名, airline 所属航空公司, name 姓名, email 邮箱, password 密码;
- User: UID 用户的内部编号, ID 用户名, name 姓名, gender 性别, phone 电话, password 密码, email 邮箱;
- Flight: UID 统一标识符 (航班的内部编号), flightNumber 航班号, airline 航空公司, departure 出发地, land 目的地, takeOffTime 起飞时间, estTime 预计飞行时间, seat 座位信息 (序列化存储), type 机型, onTimePerf 准点率, pilot1 飞行员 1, pilot2 飞行员 2, price1 经济舱价格, price2 头等舱价格, remain1 经济舱总座数, remain2 头等总座数;
- Orders: OUID 订单统一标识符 (订单的内部编号), UUID 用户的内部编号, flightNumber 航班号, number 单订单票数, tier 舱位等级, isPaid 是否支付, airline 航空公司, takeOffTime 起飞时间, estTime 预计飞行时间, price 票价, date 购买日期, fdate 出发日期;

### 2. 订单模块

问题分析:

提供用户订单支付、取消、查询等功能的 SQLite 数据库接口, 并将相关数据输出至用户图形界面。

设计方案:

在相应界面(如订单查询 CListDlg.cpp)调用 execsqlite.cpp 下的相关接口, 获得本用户相关的订单, 并进行相应操作。

#### execsqlite3.cpp

**sqlite3\* openDb(char\* fname)**

传入数据库文件路径, 打开数据库并返回数据库指针类型;

**int loginUser(sqlite3\* db, char\* ID, char\* password)**

传入数据库指针 db、用户名和密码, 与数据库内容匹配, 返回是否成功登录;

**int getOrder(sqlite3\* db, order\* dest, int UUID)**

传入数据库指针 db、需要被覆盖的订单信息结构体数组地址 dest 和用户内部编号 UUID, 将获取的用户订单信息覆盖至 dest 位置的数组, 并通过整形返回运行情况 (0 表示正常退出);

**int changePaid(sqlite3\* db, int OUID)**

传入数据库指针 db 和订单内部编号 OUID, 将相应订单修改为已支付状态, 并通过整形返回运行情况;

**int addOrder(sqlite3\* db, order orderInfo)**

传入数据库指针 db 和新增的订单信息 orderInfo, 添加新的订单;

**int deleteOrder(sqlite3\* db, int OUID)**

传入数据库指针 db 和订单内部编号 OUID, 取消相应的订单, 并通过整形返回运行情况;

每一个函数大致包含以下流程:

- 1) 通过输入的数据和需实现的功能生成 SQL 语句, 并存入 sql 字符串中;
- 2) 使用 sqlite3\_exec 执行 SQL 语句;
- 3) (可选) 通过对应的回调函数处理执行结果, 一般通过 dest 指针和 callbackCount 来确定该次调用该写入内存中哪个数组的第几个地址;
- 4) 分析运行结果, 判断是否成功。

#### CListDlg.cpp

**OnInitUpdate()**

窗口初始化函数, 初始化表格, 通过 getOrder 获取订单信息, 调用 changeList 刷新表格显示;

**changeList()**

根据现有的订单信息刷新表格显示;

**OnBnClickedButton1()**

支付订单按钮处理事件, 调用 changePaid 更改数据库状态, 若成功则刷新显示;

**OnBnClickedButton2()**

取消订单按钮处理事件, 调用 deleteOrder 更改数据库状态, 若成功则

刷新显示；

OnColumnclickList1(NMHDR\* pNMHDR, LRESULT\* pResult)

表格列点击处理事件，调用 orderSort.cpp 下的接口进行排序；为了切合题意，我们没有使用数据库的索引功能，而是在程序部分使用冒泡排序算法实现排序；

### 三 . 项目运行结果及分析（运行截图及说明）

在数据库中，所有可用的用户账号有：

	UID	ID	name	gender	phone	password	email
	...	过滤	过滤	过滤	过滤	过滤	过滤
1	3	LPQ2002	LiuPeiqing		0 18888888888	lpq020304	lpq02@163.com
2	4	YG2002	YuGuo		0 18876543210	yg020304	yg03@163.com
3	5	HZC2002	HuZhuocheng		0 16666666666	hzc020304	hzc02@163.com

管理员账号：

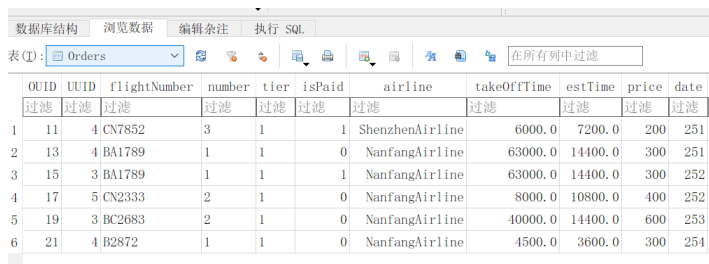
	UID	ID	airline	name	email	password
	...	过滤	过滤	过滤	过滤	过滤
1	1	WZX2001	NanfngAirline	WuZhengxiao	wzx03@163.com	wzx020304

与上理由相同，此处仅以用户的订单功能部分为例。

使用 YG2002 的账号选择日期登录后，在左侧菜单找到订单管理，即可列出已有订单，与数据库内容比对，一致（UUID 为 4 的订单均正常显示）；

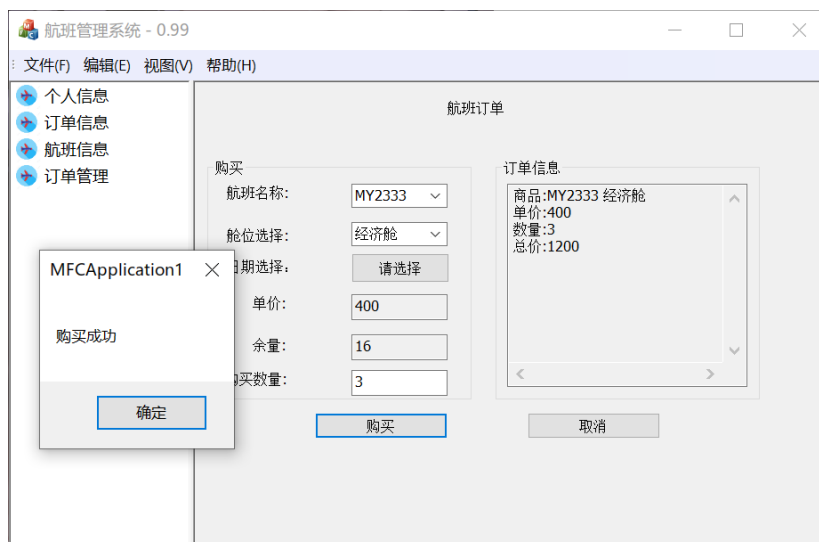


航班号	起飞时间	持续时间	舱位类型	单位票价	数量	支付状态	购买日期	出发日期	座位号	总价
CN7852	1:40	2:0	经济舱	200	3	已支付	9月8日	10月1日	Y51C,Y51...	600
BA1789	17:30	4:0	经济舱	300	1	未支付	9月8日	10月1日	Y93E	300
B2872	1:15	1:0	经济舱	300	1	未支付	9月11日	3月2日	Y71C	300



	OUID	UUID	flightNumber	number	tier	isPaid	airline	takeOffTime	estTime	price	date
	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤
1	11	4	CN7852	3	1	1	ShenzhenAirline	6000.0	7200.0	200	251
2	13	4	BA1789	1	1	0	NanfngAirline	63000.0	14400.0	300	251
3	15	3	BA1789	1	1	1	NanfngAirline	63000.0	14400.0	300	252
4	17	5	CN2333	2	1	0	NanfngAirline	8000.0	10800.0	400	252
5	19	3	BC2683	2	1	0	NanfngAirline	40000.0	14400.0	600	253
6	21	4	B2872	1	1	0	NanfngAirline	4500.0	3600.0	300	254

在“订单信息”中购买航班，正确选择信息并购买后，数据库内容正确更新；



DB Browser for SQLite - C:\Users\wx14\Desktop\BUPT2109\_Final\MFCApplication1\bupt44Database.db

文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)

新建数据库(N) 打开数据库(O) 写入更改(W) 倒退更改(R) 打开工程(P) 保存工程(V)

数据库结构 浏览数据 编辑备注 执行 SQL

表(T): Orders

	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤
	UUID	flightNumber	number	tier	isPaid	airline	takeOffTime	estTime	price	date	
1	11	4 CN7852	3	1	1	ShenzhenAirline	6000.0	7200.0	200	251	
2	13	4 BA1789	1	1	0	NanfongAirline	63000.0	14400.0	300	251	
3	15	3 BA1789	1	1	1	NanfongAirline	63000.0	14400.0	300	252	
4	17	5 CN2333	2	1	0	NanfongAirline	8000.0	10800.0	400	252	
5	19	3 BC2683	2	1	0	NanfongAirline	40000.0	14400.0	600	253	
6	21	4 B2872	1	1	0	NanfongAirline	4500.0	3600.0	300	254	
	22	4 MY2333	3	1	0	ShenzhenAirline	2000.0	7200.0	400	255	

在订单管理菜单选择新的订单并点击支付订单, 可看到数据库和界面内容正确更新;

航班管理系统 - 0.99

文件(F) 编辑(E) 视图(V) 帮助(H)

个人信息 订单信息 航班信息 订单管理

订单管理

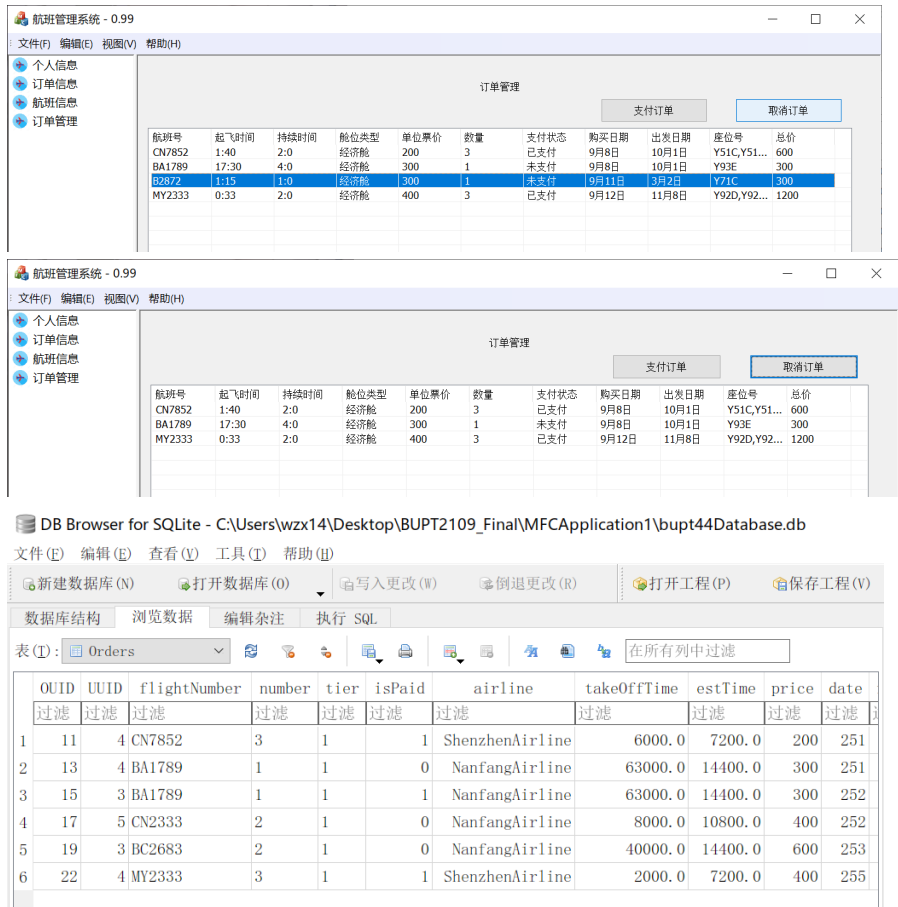
支付订单 取消订单

航班号	起飞时间	持续时间	舱位类型	单位票价	数量	支付状态	购买日期	出发日期	座位号	总价
CN7852	1:40	2:0	经济舱	200	3	已支付	9月8日	10月1日	Y51C,Y51...	600
BA1789	17:30	4:0	经济舱	300	1	未支付	9月8日	10月1日	Y93E	300
B2872	1:15	1:0	经济舱	300	1	未支付	9月11日	3月2日	Y71C	300
MY2333	0:33	2:0	经济舱	400	3	已支付	9月12日	11月8日	Y92D,Y92...	1200

(T): Orders

	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤	过滤
	UUID	flightNumber	number	tier	isPaid	airline	takeOffTime	estTime	price	date	
11	4 CN7852	3	1	1	1	ShenzhenAirline	6000.0	7200.0	200	251	
13	4 BA1789	1	1	0	0	NanfongAirline	63000.0	14400.0	300	251	
15	3 BA1789	1	1	1	1	NanfongAirline	63000.0	14400.0	300	252	
17	5 CN2333	2	1	0	0	NanfongAirline	8000.0	10800.0	400	252	
19	3 BC2683	2	1	0	0	NanfongAirline	40000.0	14400.0	600	253	
21	4 B2872	1	1	0	0	NanfongAirline	4500.0	3600.0	300	254	
22	4 MY2333	3	1	1	1	ShenzhenAirline	2000.0	7200.0	400	255	

任意选择未支付订单并点击取消订单按钮，界面和数据库中相应订单正确消失；



篇幅所限，难以分析所有和数据库有关的功能，其他部分同理。

## 四 . 项目心得及体会

虽然身为组长，但这次其实是我第一次参与开发不小的项目，也是我做过的东西里面使用的技术栈的数量和难度比较高的一次（原先还打算做成基于 TCP Socket 的 C/S 架构，但日程所限只能作罢）。每位组员都积极地适应不太一样的技术栈和开发环境，为本项目的开发添砖加瓦。

作为项目分工和统合的负责人，虽然在最初就意识到了分模块和低耦合的重要性，为此早早地写了需要实现和参考的 API 文档，但最终开发的成品依然与文档中的计划有不小的差距——一部分是因为逻辑改了，更多的是开发中按照计划的 API 实现会更加麻烦。这就导致后来的整合中多了很多不必要的工作。同时，外部库使用经验的不足也使得我在环境配置上也花了不少的时间。本次作业让我接触并入门了项目开发的一些方法，理解了模块化和解耦的重要性，同时也增加了算法题之外的实际开发的经验。

我相信，本次小学期一定能给本组成员带来收获，为我以后的项目开发带来很多帮助。