基于人脸识别的图书馆座位预约系统

王魏兴[1]

（1. 青岛科技大学 信息技术学院，山东青岛 266042）

摘　要：本文实现了基于人脸识别的图书馆座位预约系统，本系统基于 Spring Boot 框架和虹软离线识别 SDK 实现，有效解决了因为同学抢座占座导致的图书馆找座位难的问题。与现有的图书馆预约系统相比，本系统具有维护成本低，用户体验好等优点。通过座位与人脸绑定的形式，切实有效的解决了占座问题。本系统具有线上预约座位、人脸识别、预约超时提醒等功能。图书馆人脸识别配合预约系统管理自习室的位置安排，学生可以在线申请预约图书馆自习室的座位，刷脸进入自习室，统计学生的自习情况，对于经常存在的占位行为也会给予提醒，减少占座带来的纠纷问题。

关键词： 人脸识别；线上预约；图书馆管理

LIBRARY SEAT RESERVATION SYSTEM BASED ON FACE RECOGNITION

WANG Weixing[1]

(1. School of Information and Technology, Qingdao University of Science and Technology, Qingdao 266042, China)

Abstract: This article introduces an implementation method of a library seat reservation system based on face recognition. The purpose is to solve the hidden safety hazards caused by library crowding caused by queuing for seats. Compared with the existing reservation system, this system has a lower cost. Follow-up maintenance costs, better user experience, and at the same time solve the shortcomings of the existing system, make the use of library self-study resources more efficient. There is a phenomenon of occupying seats in the library, which wastes library resources. The system can reserve seats online to change the current situation of library congestion. The server can intelligently cancel overtime reservations to maximize resource utilization. The library's face recognition and the reservation system manage the location of the study room. Students can apply online to reserve a seat in the library study room, brush their faces and enter the study room, count the students' self-study status, and also give reminders for frequent seat-occupying behaviors. , To reduce disputes caused by seat occupation.

Keywords: face recognition; online booking; library management

1 绪论

1.1 研究的意义与背景

随着国内高校招生规模的不断扩大，学生对图书馆公共自习资源的需求也日益增长，现阶段，大多数高校的图书馆建设无法及时与招生规模同步，高校图书馆自习资源往往供不应求[6]。实际调查中发现，传统的自习资源管理存在诸多问题，如资源利用率低。此外，学生为了能在图书馆拥有一个自习位置，在图书馆开放前就已早早在门外排队，但当图书馆开放时，这种井然有序的状态可能被瞬间打破，大家推推嚷嚷，都想尽快进入图书馆，这无疑存在发生践踏事故的风险，此外，一些同学为确保自己的朋友拥有位置，可能会一次占取多个位置，这又带来了位置使用权的纠纷问题。

一些同学在占到位置以后，便不见了踪影，对于这种情况，一般有以下两种可能：

（1） 临时外出，稍后会回到位置自习；

（2） 在图书馆占一个位置，晚上闭馆再回来收拾自己的书籍。

对于情况（2），无疑是对资源的一种浪费，而传统的自习资源管理方式，主要是靠书库管理员巡视，清理占座书籍，但这很可能会导致情况（1）的同学失去自己的位置，存在“误清”的情况。

1.2 本文研究内容

国内大学校园招生规模的不断扩大，出现不同程度的图书馆公共资源的浪费，其情况每日逾下，为了更好的提高高校图书馆自习资源的利用率，同时减少安全隐患，本文设计并开发了一个适用于校园的智能化图书馆预约座位管理系统。本系统具有线上快速预约座位服务、管理自习座位功能，即可满足学生对上自习的需求同时也方便图书馆老师的管理。

本文研究的主要内容有：

（1）了解当前技术发展形式，熟悉并明确研究意义；

（2）对系统的需求以及方案进行分析：开发环境技术安排以及系统可行性分析；

（3）系统设计：设计流程图、系统结构图等；

（4）系统设计: 系统包括所有图书馆用户登录信息模块、管理员信息模块、图书馆座位信息模块等,实现对每个学生的所有用户登录以及图书馆的详细信息管理以及对用户预约信息进行查看、筛选等详细管理操作;

（5）对系统的操作逻辑、界面以及功能等关键部件进行了测试,减少并及时改正其中的技术漏洞等,同时在本次测试中注意对系统的功能进行完善。

2 系统分析

2.1 系统功能分析

目前国内的大学规模越来越大，有限的图书馆自习资源的扩充与招生人数指数型增长之间的矛盾越来越大。另外，新生在线下寻找相应座位时，会因为对周围环境生疏浪费不少时间。采用本文介绍的图书馆座位预约系统，可以让所有可利用资源在微信小程序中一览无余，尽最大可能缩小在未知情况下的对于座位的寻找范围，同时可以根据喜好对相关座位进行查找等操作[7]。另外，大规模的线下环境，对于图书馆的管理也不容易，插队，拥挤，踩踏，抢座，占座，各种不良现象都很难避免，给图书馆的管理带来困难。借助本系统，学生根据自己的需要主动预约位置，一个位置只能被一个同学预约，避免了拥挤、不必要的纠纷，同时，系统采用一种“信誉”机制，尽可能杜绝占座现象，提高有限资源的利用率。

综合上述问题，将本系统由三部分组成[7]：

（1）管理员系统，供图书馆管理员维护系统数据、查看预约记录使用，应具有基本信息维护、批量数据导入功能，此外，预约、签到、鉴权等操作所依赖的功能也由管理员端提供，即管理员系统与系统后台整合，作为一个整体；

（2）预约系统，供学生预约位置使用，应具有位置查询、预约、取消预约等功能；

（3）签到系统，供预约后签到使用，采用人脸识别技术，应具有实时人脸检测与识别、活体检测等功能。

2.2 可行性分析

本系统基于 SpringBoot 技术栈完成设计搭建。SpringBoot 框架方便对于关系对象、方法、关系视图和依赖等数据进行妥善的维护,并大大简化了我们的系统代码。MyBatis 框架可以对所用的数据库部分完成持久的连接。MySQL 数据库是一个轻量型关系数据库，数据处理能力较为强大，足已完成本系统所需要的数据管理。Redis 是一种非关系型数据库，其数据存储在内存中，同时也具有持久化能力，得益于内存的高速访问，其作为数据库缓存可以极大提高数据查询效率，确保系统可以承受更多的访问请求。

本系统相对于传统的预约系统，具有以下优点[8]：

1、大量减少了线下监管的人力物力。

2、方便同学入馆自习。

3、相比扫码签到，维护成本低。

4、人脸识别、活体检测技术的引入，解决了扫码签到的弊端。

从技术层面及中国经济政策层面上来看，本系统是可行的。

3 系统架构设计

3.1 系统架构设计

本系统采用较为主流的技术框架实现，并结合人脸识别等前沿技术，系统实现整体架构图如图 1。

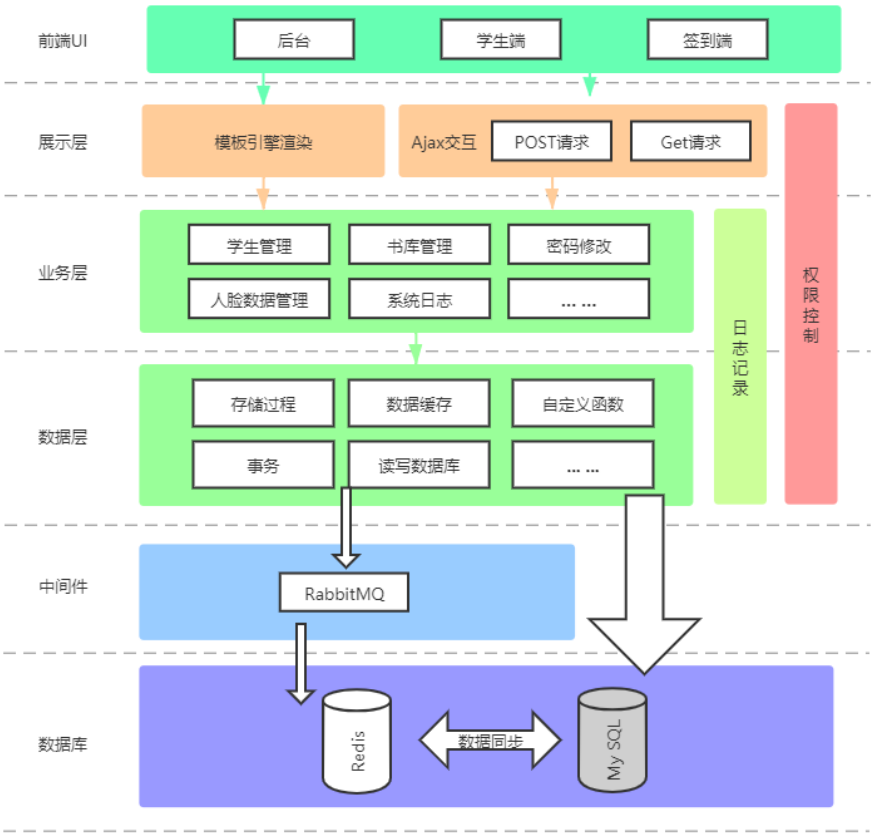

图 1 系统架构图

Fig.1 System architecture diagram

3.2 系统技术选型

本文采用 SpringBoot 框架作为系统后台开发框架，其具有诸多优势，如非常简洁的安全策略、集成支持关系数据库和非关系数据库、支持运行期内嵌容器、自动管理依赖、自带应用监控等。前端采用 LayUI、JQuery 等技术，可以在较短时间内快速开发出界面美观的 Web 页面。为实现跨平台，同时方便用户使用，采用微信小程序作为学生端，用于预约位置。

对于数据的存储，本文采用 MySQL 关系型数据库，为提高系统性能，方便后台开发，采用 ORM 框架 MyBatis 管理数据库连接。

除以上必须基本框架外，本文还采用了 Redis 非关系型数据库作为 MySQL 的缓存，使用 RabbitMQ 异步向数据库写入数据，以此提高系统承受高并发访问的能力。

4 系统设计

4.1 数据库设计

数据库管理系统作为一个项目或系统的重要数据来源和核心必不可少,起着举足轻重的数据基础作用,需要对所有的数据进行统一的设计和管理。经过数据库的需求结构分析、概念结构设计、逻辑结构设计、数据库的物理结构设计、数据库的实施和管理以及数据库的运行和数据系统维护等阶段即可设计出项目或系统所需的整个数据库[2]。

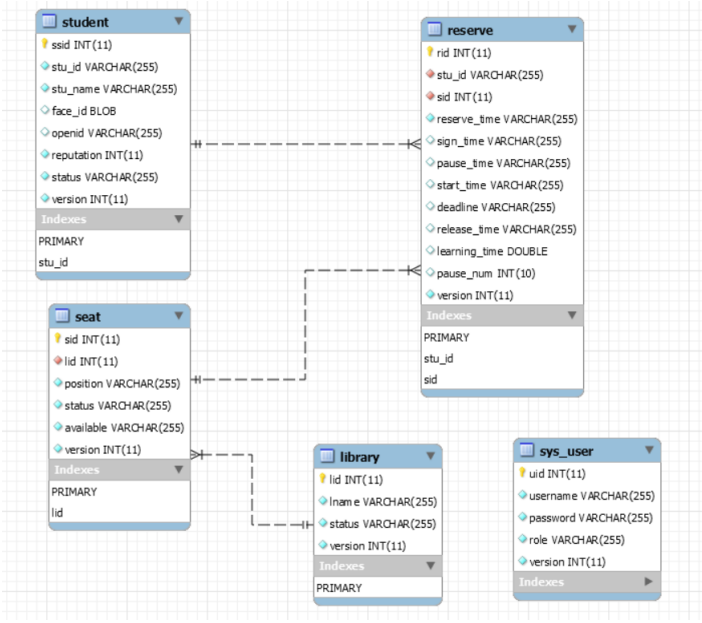

图 2 数据库 E-R 图

Fig. 2 Database E-R Diagram

4.2 系统流程设计

根据系统各部分功能需求，得到系统各部分核心工作流程图。管理员系统人脸识别流程图如图 2，签到系统工作流程图如图 3，预约系统工作流程图如图 4。

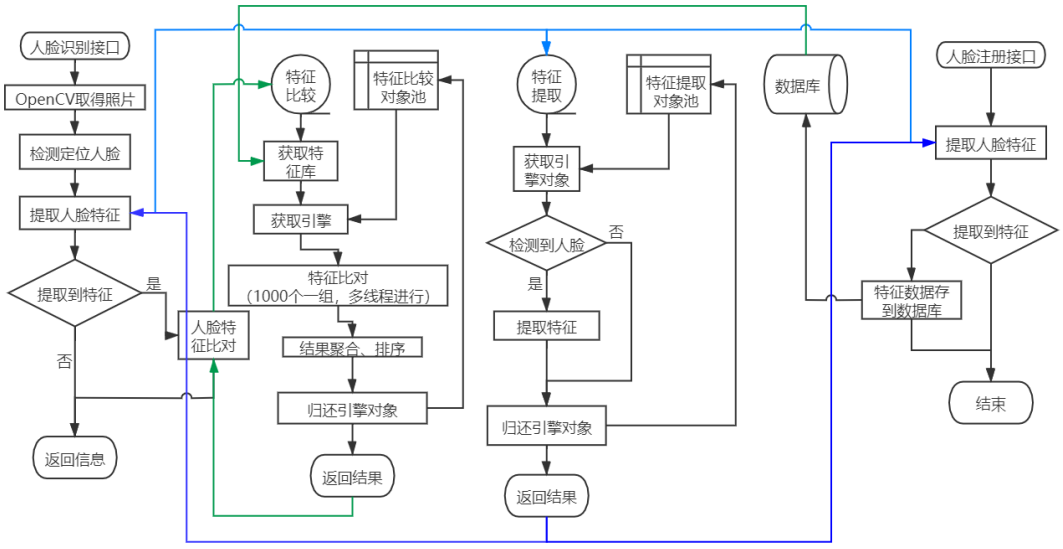

图 3 后台人脸识别流程图

Fig. 3 Background face recognition flow chart

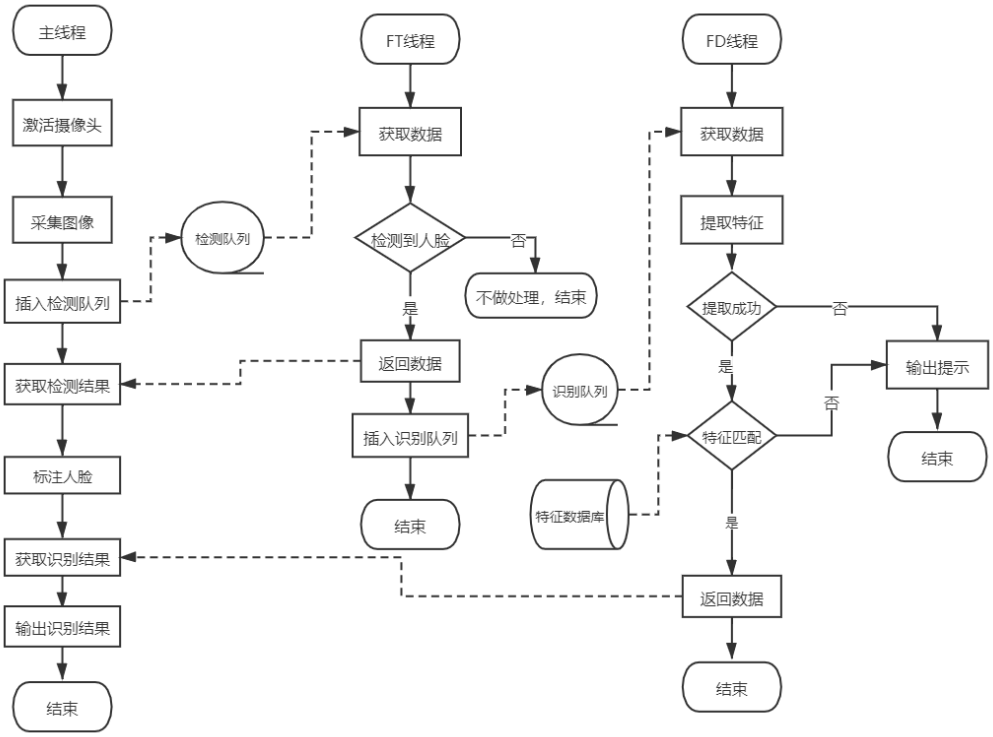

图 4 签到系统流程图

Fig.4 Sign-in system flow chart

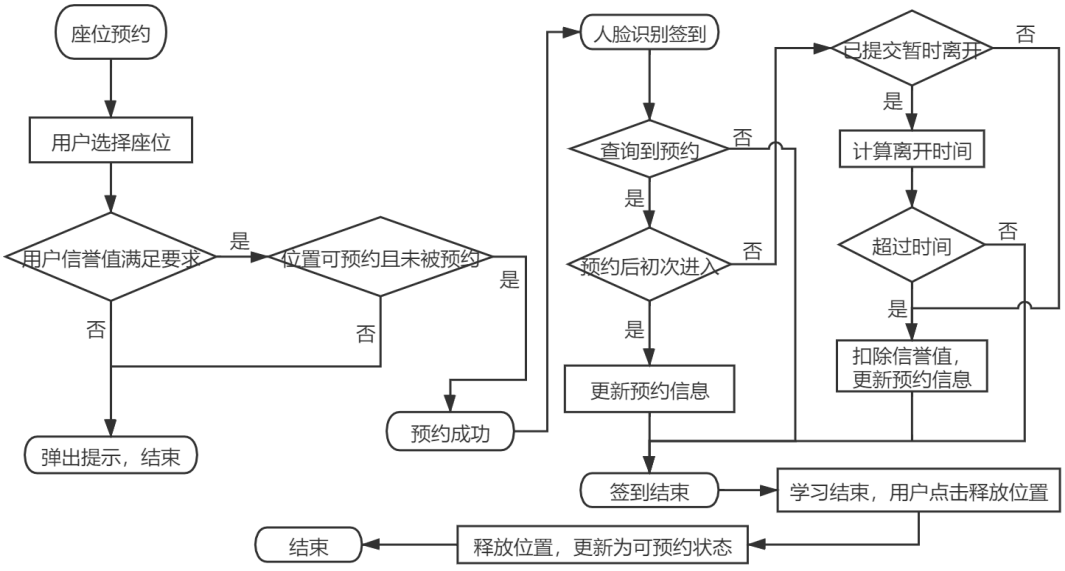

图 5 预约系统流程图

Fig.5 Appointment system flow chart

5 技术要点

5.1 小程序页面间通信

与 WEB 页面不同，微信小程序页面之间跳转需要调用微信提供的 API，其官方文档中并未给出页面间数据交互的参数及样例，微信小程序其本质是对传统 WEB 网页的封装，故页面间通信可以采用传统 URL 携带参数方式，只需要在跳转前在目标地址添加需要传递的

数据，即可在跳转后的页面中获取、解析、使用数据。

5.2 管理员系统人脸识别性能调优

管理员系统人脸识别主要用于提取人脸特征数据，完成人脸注册功能。为加快开发过程，本系统人脸识别技术使用虹软离线识别 SDK 支撑，人脸识别是相对耗时的操作，为确保后台能够及时响应用户请求，人脸识别部分采用多线程设计，异步执行。

考虑到可能存在多个用户同时进行人脸注册操作，采用线程池技术，提前建立多个人脸检测、识别线程，当接收到请求时，向线程池“租借”一个线程，使用完成后再“归还”。线程池的大小可根据服务器 CPU 核心数量进行调整，一个大小为 n 的线程池，可以同时处理 n 个人脸注册请求。其中，n 的取值应小于服务器 CPU 核心数，以确保服务器有空余核心去处理普通访问请求，防止发生处理机争抢，进而导致频繁发生作业调度，引发程序性能下降。

5.3 签到系统线程间通信

签到系统涉及较多人脸识别功能，其中人脸特征提取所需时间相比人脸检测所需时间较长，为确保系统预览画面的流畅，采用多线程进行程序设计，将人脸检测和人脸识别分别放到两个子线程中执行。

系统主线程负责进行图像采集、信息反馈提示等操作，人脸检测线程进行人脸位置的探测，给出人脸区域边框坐标及当前人脸 ID 编号，人脸识别线程进行人脸的特征比对，得出最终识别结果，对于同一人脸 ID 的数据，仅需识别一次即可，无需重复识别。

以上三个线程间需要进行通信，本系统采用信号量机制协调线程间的同步，互斥访问共享数据队列。其中，绝大多数数据交换发生在人脸检测线程和人脸识别线程之间，进而可以将该问题抽象为生产者—消费者模型，人脸检测线程生产数据，人脸识别线程消费数据。其核心代码如下所示。

```
1. semaphore mutex = 1;  // 互斥变量
2. semaphore empty = k;  // 缓冲区大小
3. semaphore full = 0;  // 已占用的缓冲区的大小
4.
5. producer() {
6.      P(empty);
7.      P(mutex); // 进入临界区
8.      // 生产数据...
9.      V(mutex); // 退出临界区
10.     V(full);
11.}
12.consumer() {
13.     P(full);
14.     P(mutex); // 进入临界区
15.     // 消费数据...
16.     V(mutex); // 退出临界区
17.     V(empty);
18.}
```

5.4 接口幂等性

幂等性是离散数学的一个概念，其公式为：

$$f\left(f\left(f(x)\right)\right)=f(x)$$

由公式可得，一个幂等操作的特点是其任意多次执行所产生的影响均与一次执行的影响相同。即幂等结口对于相同数据，无论执行多少次，都应只进行一次处理。

为更好地实现“用户信誉值”功能，后台签到接口在重复发生访问时，会扣除当前签到用户的信誉值。为了避免因网络波动等原因导致后台在哟此签到过程中接收到多次签到请求，进而导致重复签到错误扣除信誉值，签到接口应实现幂等性。

本系统接口幂等性借助 Redis 和 Token 机制实现。在访问签到接口之前，先访问 token 请求接口，获取一个 token，该 token 使用 Redis 进行缓存并设置过期时间。请求签到接口时，除签到必要数据外，还要携带 token 数据，后台接收到请求后，先验证 Redis 缓存中是否存在 token，如果存在，则将缓存删除后执行业务操作，否则不进行操作并返回“已经完成签到”的提示。其流程图如图 5。

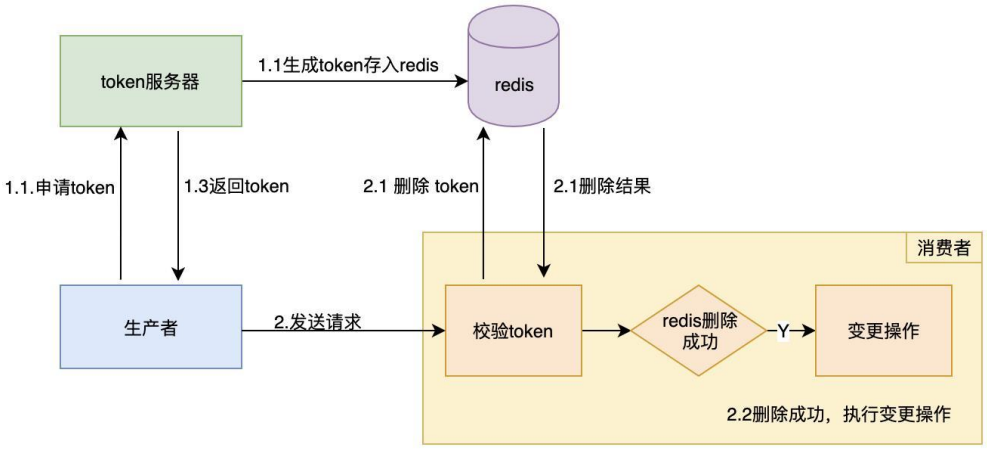

图 6 接口幂等性流程图

Fig. 6 Interface idempotence flowchart

5.5 并发处理

根据调查数据分析可得，图书馆在书库开放前，会存在高并发访问，为确保数据可靠，防止发生“超占”现象，即一个位置同时被两个人同时预约，应当采取恰当的数据校验处理手段。为确保系统性能，应当从最底层数据库入手，结合数据库设计，实现乐观锁，在保障数据可靠的同时保证性能。其原理如图 6。在进行数据更新时，将当前数据版本号加一，与更新内容一起提交到服务器，服务器接收到请求后，先将版本号与当前数据库中的数据版本号比较，如果当前版本号加一等于提交的版本号，则进行更新操作，否则说明以已经有请求修改更新过数据，则忽略数据，返回失败提示给用户。

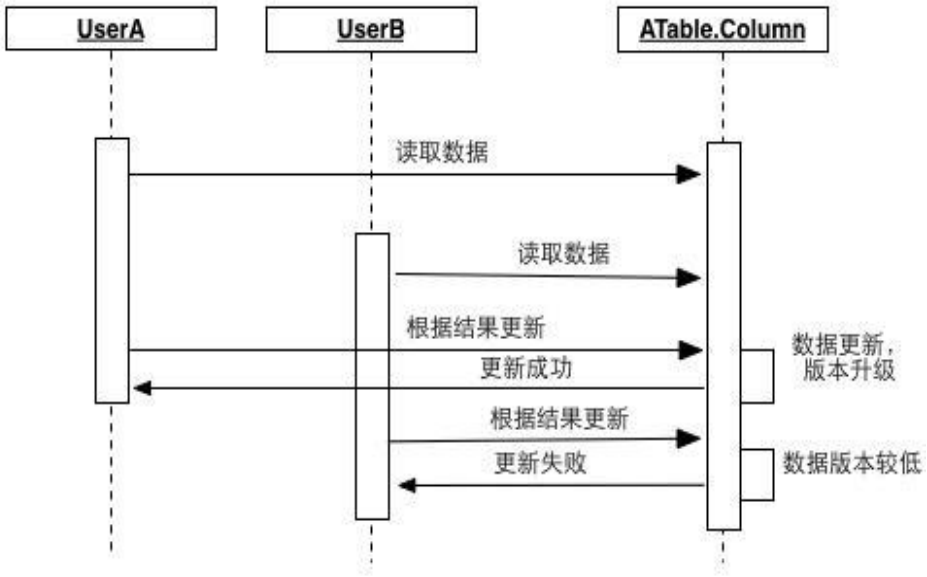

图 7 乐观锁原理图

Fig. 7 Optimistic Locking Schematic

5.6 引入数据库缓存带来的问题及解决方案

引入数据库缓存后，一定程度上可以提高系统对并发访问的承受能力，但同时因为缓存的引入，导致出现一些新的问题。由于缓存涉及到数据库数据，因此这些问题如果处理不当，除了影响系统性能外，也将对数据库中的数据可靠性造成影响。

引入缓存后带来的问题如下：

（1） 缓存雪崩问题；

（2） 数据一致性问题；

（3） 缓存穿透问题；

（4） 缓存击穿问题。

“缓存雪崩”是指缓存在同一时间集体失效，导致大量数据库访问，突如其来的大量访问可能会冲垮数据库，导致系统服务不可用，为防止缓存雪崩发生，在设置缓存过期时间时，应当随机设置，避免过期时间相同，对于一些热点数据，可以设置缓存数据永不过期。

“数据一致”即缓存与数据库中的数据要保持一致。缓存一致性可能会在更新数据、删除数据时被破坏，一般是由于更新了一端数据，而另一端未被更新。例如，更新数据库数据后没有更新缓存，缓存因为还没有过期，所以在查询数据时不会触发缓存更新，查询到的数据将依然是旧数据，最终导致业务逻辑出现异常。对于该问题的解决，可以在更新数据库前，先删除对应缓存，删除成功后再更新数据库，以便在下次查询数据时被动触发缓存更新操作，确保数据一致。

“缓存穿透”指一个缓存刚好过期时，有大量访问请求到来，导致数据库服务负载过大。对于该问题的处理，可以采用显式锁解决。当有缓存数据过期时，会触发查询数据库的操作，若对数据库查询操作上锁，则只有一个请求会进入查询数据库操作，其他请求阻塞等待，查询完成后，更新缓存，手动释放锁，即完成了对缓存的更新后。此时，被阻塞的请求便可从缓存中获取数据，而不是去查询数据库。

“缓存击穿”指一个请求频繁查询一个数据库中不存在的数据，可能会导致数据库服务负载过高。对于该问题的解决，思路非常简单，只需要对查询结果为空的键值也做缓存，即可避免对数据库的访问，防止持续高并发访问导致数据库服务崩溃。

5.7 算法设计

不同书库位置分布排列不同，为提高系统可维护性，降低耦合度，位置分布图需要小程序从后台请求位置分布数据，动态渲染。后台数据库中存储的位置信息是位置坐标，其格式不方便小程序渲染平面图使用，为了方便小程序使用，在后台返回位置数据前，需要先对数据进行相应的格式转换，本文给出一个转换的方法：

（1） 根据查询到的数据长度建立二维数组，并以 0,0 元素为坐标原点；

（2） 遍历数组，借助封装类添加相应的附加信息，直至所有数据遍历完成；

（3） 返回处理后的数据。

本算法时间复杂度为$O(N^2)$，空间复杂度为$O(N)$，因为书库中位置数据量较少，数据规模较小，从时间复杂度和空间复杂度综合分析，本算法可行，对于 500 个位置数据，可在极短时间内完成处理。

6 结语

传统人工方式管理图书馆自习资源，存在效率低下、工作量繁杂的问题，一些高校引进了通过扫码完成签到的位置预约系统，但二维码容易磨损，维护成本高，后续又发展为采用扫码枪扫码，降低了维护成本，但采用扫码签到的方式存在“代签到”的弊端，无法进一步提高资源利用率。

本文介绍的方法采用人脸识别进行签到，可以与图书馆门禁结合，在进入图书馆的同时完成签到，节约学生的时间，借助活体检测技术，必须真人到场才能完成签到，可以有效解决“代签到”问题。

基于以上优点，本系统相较于传统无预约、手工预约及扫码预约均具有更好的应用前景。不过，由于时间原因，本系统尚有以下问题需要进一步考虑，这也是所有信息系统，尤其是各类作为预约系统的共同难点：

（1） 投资成本与维护成本之间的博弈；

（2） 特殊时段仍存在排队问题。

参考文献：

[1]杜波伊斯.MySQL 技术内幕[M].第 4 版.人民邮电出版社,2011 年 07 月.

[2]赵松涛.中文版 SQLServer2005 应用及实例集锦[M].北京:人民邮电出版社,2005

[3]罗辉，web 商务系统的设计与实现[J],电脑与信息技术，2008，(1):165-168

[4]徐敬.J2EE 架构下基于 Web 的信息管理系统框架研究[J]. 硅谷， 2015(1):84-85.

[5]王英龙. 张伟. 杨美红. 软件测试技术[M]. 清华大学出版社，2009.

[6]王璐.大学校园占座理象的管埋学分析 [J].杜科纵横,2008(23):82-83.

[7]王守红.高校图书馆自习座位管理系统设计 [J]大学图书馆学报,2010(2):38-40.

[8]孙发.吴代莉.曾为众.图书馆自习室管理系统的设计与实现[J] 现代图书情报技术.2010 (5):93-98.

[9]谢红.王炳江.基于 VB 的图书馆阅览座位管理系统的开发及应用[J].图书馆论坛,2010(10):58-60.

[10]RogerS-Pressman.SoftwareEngineering[M].NewDelhi:TataMcGraw-Hill PublishingCompany Ltd,2006.

[11]CKirkrgaard,A Ller.StaticanalysisforjavaservletsandJSP[M].InternationalConference on StaticAnalysis,2006.

[12]A Steelman,JMurach.Murachs Java Servletsand JSP,2nd Edition[M].Mike Murach&Associates,2008.

[13]AR Hevner ,ST Marcb. The Information Systems Research Cycle[J]. IEEE Computer, 2003,9 (11):111-113.

[14]R Sandhu,V Bhamidpati,Q Munawer. The ARBAC 97 model for role-based administration of roles[J]. ACM Transactions on Information and System Security,1999,8(1):105.

[15]王金.非关系型数据库的设计与实现[J].《计算机导刊》,2018,1(2):56-58.