一、 程序功能简介

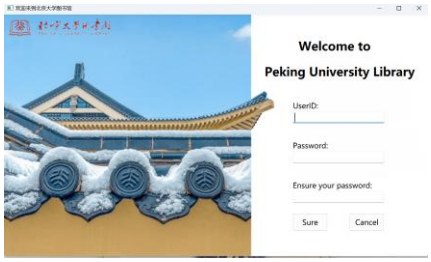
该项目实现了用户的注册、登录功能以及北京大学图书馆各楼层的简要座位预约功能，包括预约、退约、查看预约记录，此外还有一个简易公告栏可以显示近期的新闻，程序基本上贯彻前后端分离的思想。

二、 项目各模块与类设计细节

1. 前端（可视化界面）

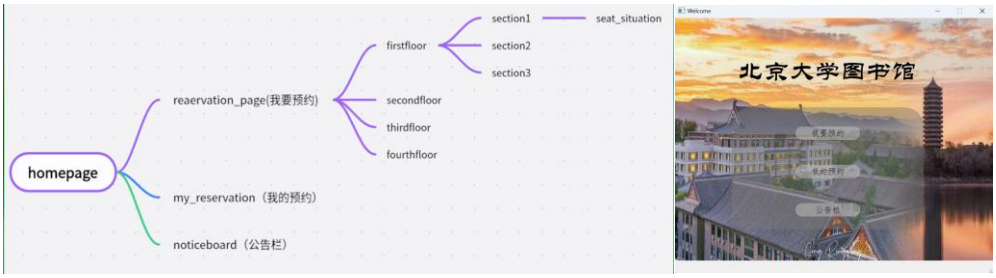
（1）登录注册界面

登录注册界面独立于其他界面存在，为一个 MainWindow，命名 mainwindow 类和 sign_up 类，输入账号与密码正确后可进入主页面，注册界面为其子窗口，注册界面为模态窗口，账号不与已有账号重复，密码长度在 7-14 位时即可成功注册，账号密码储存在 seating_system.db 文件中，无法成功注册或登录的情况均会有弹窗报错，登录成功后就会进入主页面，此窗口消亡。



（2）主页面

主页面及其下属界面关系如下：

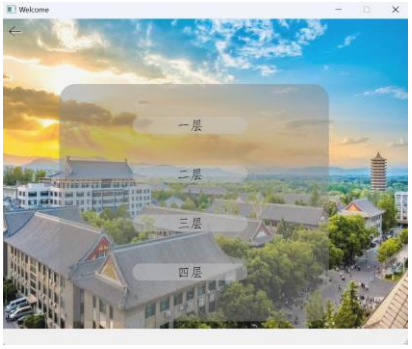


以上界面全部属于同一个 MainWindow，或作为子窗口弹出，主页面命名 homepage 类，前三层的界面切换是同一个 MainWindow 的 centralWidget 的更换，转换比较流畅，下属界面左上角均有返回键，点击返回键会发送一个“go_back”信号到 homepage，homepage 处理信号找到对应的 Widget 进行更换，从而实现返回上一个界面的操作；余下界面均以子窗口形式弹出，实现相应功能。

（3）楼层选择界面与楼层界面

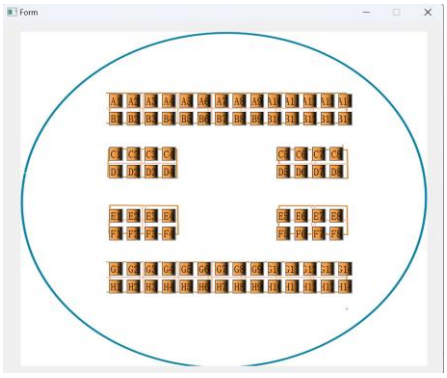
楼层选择界面与楼层界面设计简单，点击相应按钮跳转到相应楼层或区域，

点击左上角返回键返回上一界面，楼层选择界面命名 reservation_page 类，楼层界面统一采用“f2_plan”形式命名类，其中第一层作为代表层，类命名为 first_floor,图片使用的是北京大学图书馆官方发布的平面图。



（4）座位选择界面

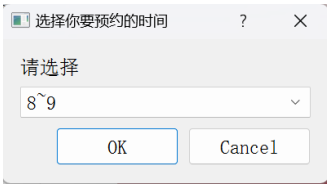
座位选择界面统一以“f1_part1”形式命名类，代表第几层第几个区域的座位分布，由于组内成员绘图能力较差，因此较为简陋，其上每个按钮代表一个座位，按钮上的编码即为座位编码，所有座位编码都会存储在 seating_system.db 文件中，点击按钮会弹出专属于该座位的预约界面。



（一层阳光大厅）

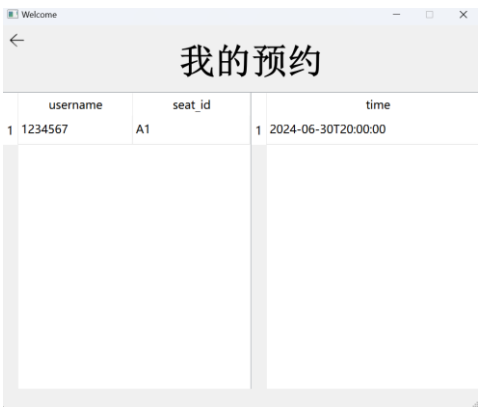
（5）预约界面

预约界面命名 seat_situation 类，预约界面左侧为当前位置信息，空白表示时间已过，蓝色表示该时段座位空闲，红色表示该时段已被预约，状态每隔十秒更新一次，通过后端函数对状态进行查询进而更新；当前时间为国际标准时间，点击“预约”或“提前退约”按钮后会弹出如下窗口，时间不符合规定时均会弹出相应警告，相应预约仍然会储存在 seating_system.db 文件中。



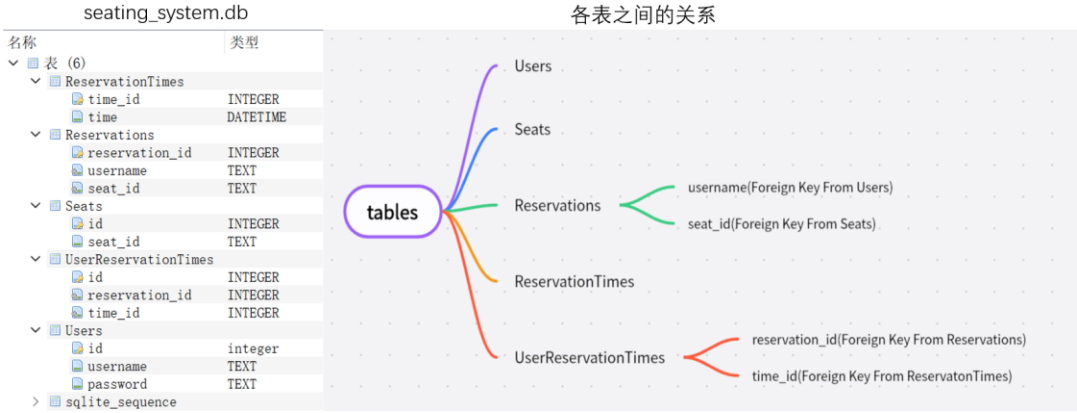
（6）公告栏与预约记录查看界面

公告栏界面命名 noticeboard 类，公告栏中每个按钮点击后可以查看对应新闻；我的预约命名 my_reservation 类，三个信息“username”“seat_id”“time”相对应，并有唯一的 Reservation_id 储存在 seating_system.db 文件中。



2. 后端（数据管理程序）

后端程序以 Sqlite 类型的数据库文件 seating_system.db 为核心，编写了 database.h/cpp，其中共有九个查询、修改相关的函数，seating_system.db 文件结构如下：



所有函数均对注入攻击有所预防，出现错误回滚事务，运行结束提交事务。

（1）void sqlite_Init()

初始化函数，初始化并打开数据库 seating_system.db。

（2）bool isUsernameAvailable(const QString& username)

该函数用于在注册时检测用户名是否已经存在，存在则返回 True，否则返回 False。

（3）bool sign_in_serach(const QString &username,const QString &password)

该函数用于在登陆时检查用户名与密码是否正确，正确则返回 True，否则返回 False。

（4）int sign_up_data_upload(const QString &username, const QString &password, const QString &surepsword)

该函数用于注册时对用户名与密码进行判别后再将数据上传到数据库中，返回值为 3，提示“空用户名”错误；返回值为 5，提示“用户名已存在”错误；返回值为 4，提示“密码长度不合格”错误；返回值为 2，提示“两次输入的密码不一致”错误；返回值为 1，提示“上传错误”警告；返回值为 0，提示注册成功。

（5）void insertReservationForUser(const QString &username, const QString &seat_id,const QDateTime &reservationTimes)

该函数用于在预约界面预约时上传预约信息,包括用户名、座位号、时间，并生成唯一一个预约号作为标识。

（6）bool isReservationMade(const QString &seat_id, const QDateTime &time)

该函数用于查询某座位某时间是否有预约,主要在预约界面更新座位状态时使用。

（7）bool deleteReservation(const QString & username,const QString &seat_id, const QDateTime &time)

该函数在提前退约时使用，以 Reservation_id 作为中间媒介删除该条预约记录。

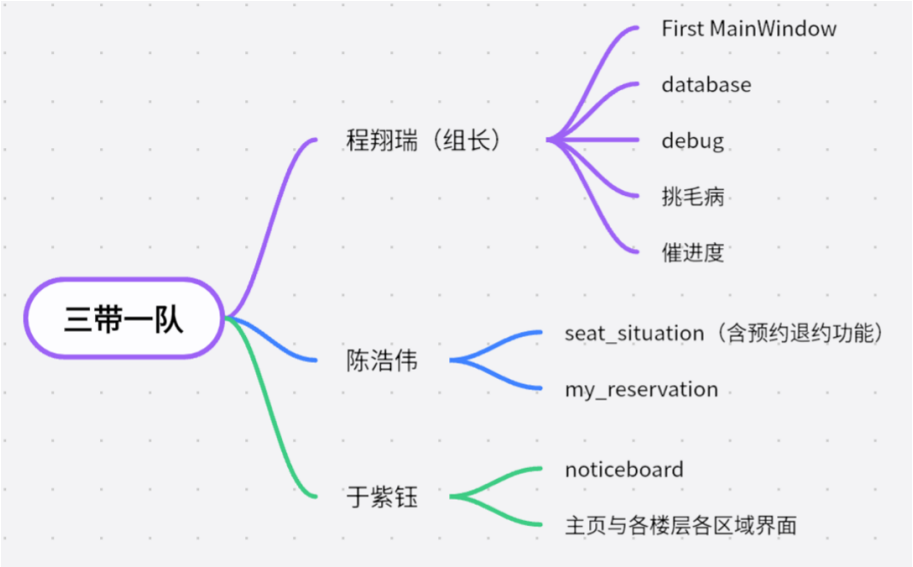
（8）void add_seat(const QString &seat_id)

该函数用于将座位编号上传到数据库中。

（9）void checkCoinsistency()

辅助函数，检查外键连接的对象是否一致。

三、 小组成员分工情况



四、 项目总结与反思

本次项目我们从期中季和期末季图书馆一座难求的现象获得灵感，设计这个简单的座位预约系统，并借鉴了学长们的优秀作业中前后端分离的思想，但是在编写“my_reservation”部分时，由于交流不足，前后端混编，未完全分离开，以致该部分逻辑比较混乱。在编写代码时我们采用了统一的命名方法，但是一位组员在命名图片时使用了拼音，带来了理解上的困难，并对一些 ui 文件组件的命名未严格按照标准，主要是因为线上交流效率差，且组员们对 github 的使用并不熟悉，组员们思想上的分歧未能及时解决。

该项目仅用期中十天时间赶出，在期末周结束后组员全部返乡招生，未对项目进行更新，且仍然存在一些顽固 bug，整体上并未到达我们最初的预期，虽然我们参加了路演，但是我们对此半成品的提交深感惭愧。