SocketLab：小说阅读器

姓名：夏海淞

学号：18307130090

概述

在本次Socket编程实验中，我实现了一个简单的小说阅读器，功能包括注册登录、翻页/翻章、跳页、添加/删除书签、下载等功能。其中前端（客户端）部分由PySide6模块实现，后端（服务端）逻辑部分使用Socket模块和基于JSON的私有协议和前端进行交互，数据部分使用SQLite完成数据存储。

文件清单及用途

文件名	用途
protocol_example	私有协议示例（JSON格式）
res	资源文件（服务端存储）
backend.db	SQLite数据文件
AddBookmarkDialog.ui/.py	添加书签视图的UI配置文件及其生成的py文件
backend_database.py	包含直接操作数据库的类
backend_server.py	直接处理前端请求的后端文件
ChapterView.ui/.py	选择书籍具体章节视图的UI配置文件及其生成的py文件
ContentView.ui/.py	阅读章节具体内容视图的UI配置文件及其生成的py文件
frontend_gui.py	前端逻辑代码
LoginView.ui/.py	登录视图的UI配置文件及其生成的py文件
MainWindow.ui/.py	前端主窗口的UI配置文件及其生成的py文件
RegisterView.ui/.py	注册视图的UI配置文件及其生成的py文件
threadpool.py	后端线程池的py文件
UserBookmarkView.ui/.py	用户书签视图的UI配置文件及其生成的py文件
WelcomeView.py	登录后选择具体书籍视图的UI配置文件及其生成的py文件
view_relation_graph.jpg	前端界面关系图
books/users.json	对后端进行数据注入的JSON配置文件
style.qss	前端样式配置文件

运行环境和运行方法

运行环境

语言：Python 3.7+

第三方Python包：PySide6

操作系统：Windows 10/Ubuntu 18.04

运行方法

前端部分

```
>python3 frontend_gui.py server_address server_port
```

后端部分

```
>python3 backend_server.py server_address server_port
```

前端架构介绍

前端采用类似移动应用的设计架构，由一个持续显示的主窗口和若干个互相切换的子视图组成。当前台子视图由 `A` 切换至 `B` 时，`A` 先被销毁，`B` 随后被初始化。这样设计有效地节省了内存开销。

主窗口与子视图的关系和交互

主窗口和子视图在前端逻辑代码中均表现为类，前台子视图是主窗口的成员变量。主窗口负责确定窗口的大小，根据不同的子视图在工具栏中显示相关操作按钮（如前台展示内容视图时，工具栏显示翻页按钮）；子视图主要负责内容的显示，内容中同样包含一些操作按钮（如前台展示注册登录）。

由于前台子视图是主窗口的成员变量，两者在交互上存在障碍，尤其是当子视图给主窗口发送消息的时候。为了在主窗口和前台子视图之间传递消息，前端利用了PySide6提供的信号（Signal）与槽（Slot）机制。

子视图对应的类维护了若干个不同的信号（Signal）变量，当子视图在主窗口对应的类中被初始化时，主窗口监听这些信号变量，并将其与自身设置的槽（Slot）函数绑定。当子视图满足特定条件（例如某个按钮被按下，文本框中内容发生变化等），某个对应的信号就会发送预先设定的消息，主窗口的槽函数就会执行。这样就完成了子视图对主窗口的特定操作。

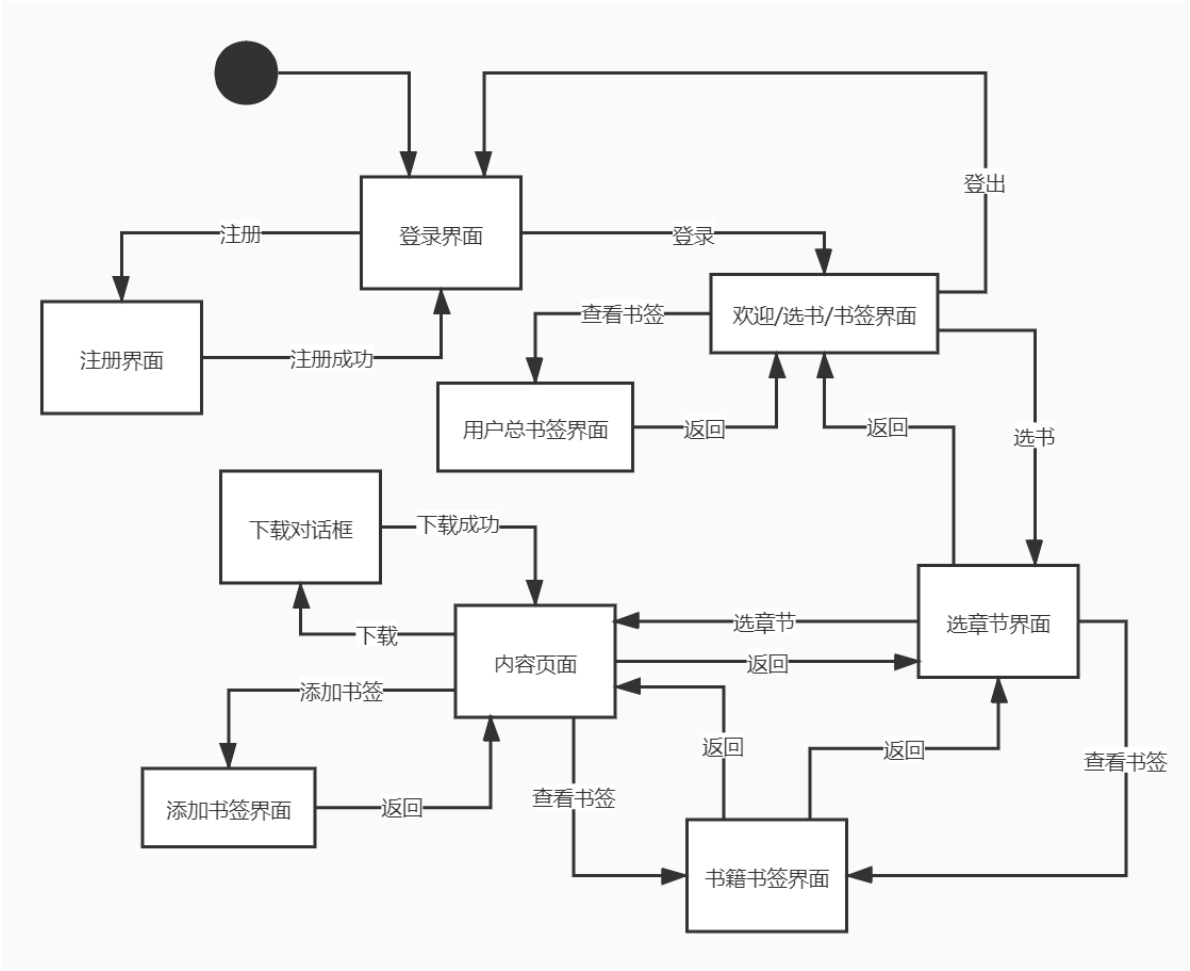
与前台子视图创建时的操作对应，当前台子视图被销毁时，同样需要对其所有信号变量和对应槽函数逐一进行解绑。

子视图关系图

子视图之间的关系较为复杂。为了更好地说明子视图之间的关系，采用以下流程图的形式进行描述。

下图中，黑色圆点表示程序开始；矩形框表示一个子视图，矩形框之间的箭头表示转移方向，转移线上的文字表示触发转移的条件。

由于程序可以在任意一个子视图阶段结束，因此关于退出程序的状态及其转移并未画出。



后端部分

Socket参数

Socket模块使用了TCP协议进行数据传输。

后端在运行时给定的服务端地址的某一端口进行监听，最大连接数设为5。

为了提高后端处理性能，后端使用了PPT中提供的线程池代码，实现了多线程的功能。线程池中的线程数设为4。

私有协议

前端和后端的通过应用层的私有协议进行交互。私有协议的设计参考了JSON的格式。

下面首先介绍私有协议的基本格式，随后抽取部分典型格式进行具体描述。

概述

私有协议的基本格式如下：

请求样例

```
{
    "type":"head_of_protocol"       // represent the type of requests
    "data":{                        // represent the data part of requests
        "some_key":"some_value"
        /*...*/
    }
}
```

响应样例

```
{
    "type":"head_of_protocol"                 // represent the type of responses, same as the relevant requests
    "data":{                                  // represent the data part of responses
        "status":"ok/error"                   // ok/error means the request is legal/illegal
        "type":"head_of_protocol/error_reason"// if 'status'='ok', 'type'=type of requests;
                                              // if 'status'='error', 'type'=reasons of error
        ["content":{"requested_data"}]        // represent the requested data (OPTIONAL)
    }
}
```

修改密码

在实现修改密码的前后端交互时，前端提供的信息有：

- 请求类型
- 用户名
- 旧密码
- 新密码

后端返回的信息有：

- 响应类型
- 响应状态（合法/非法）
- 非法错误类型

请求样例

```
{
  "type": "change_psw",
  "data": {
    "uid": "xhs7700",
    "old_psw": "12345678",
    "new_psw": "123456789"
  }
}
```

响应样例

```
{
  "type": "change_psw",
  "data": {
    "status":"ok",
    "type":"change_psw"
  }
}

{
  "type": "change_psw",
  "data": {
    "status":"error",
    "type":"wrong password/same password/user not exist"
  }
}
```

获取书签

在实现获取书签的前后端交互时，前端提供的信息有：

- 请求类型
- 用户ID
- 书籍ID

后端返回的信息有：

- 响应类型
- 响应状态（合法/非法）
- 书签信息（列表）：
  - 书签ID
  - 书签名字
  - 章节号
  - 页码

请求样例

```
{
  "type": "get_bookmark_user_book",
  "data": {
    "uid": "xhs7700",
    "book_id": "2"
  }
}
```

响应样例

```
{
  "type": "get_bookmark_user_book",
  "data": {
    "status": "ok",
```

```
    "type": "get_bookmark_user_book",
    "content": [
      {
        "bookmark_id": "1",
        "name": "bookmark 1",
        "chapter_id": "2",
        "page_num": "3"
      }
    ]
  }
}

{
  "type": "get_bookmark_user_book",
  "data": {
    "status": "error",
    "type": "user_id not exist/book_id not exist",
    "content": []
  }
}
```

数据部分

用户数据表：`users`

数据表`users`有4个字段：`uid`，`psw`，`email`和`log_state`，分别表示用户名、密码、邮箱和登录状态，其中用户名为主键。在这4个字段中，除了`email`允许为空之外，其余字段均不能为空。

邮箱字段的设置是考虑到未来引入邮箱验证的扩展功能的需要，而登录状态字段的设置是为了防止重复登录或重复登出。

书籍数据表：`books`

数据表`books`有三个字段：`book_id`，`path`和`name`，分别表示书籍编号、书籍资源路径和书籍名称。其中最终书籍编号为主键，所有字段均不能为空。

章节数据表：`chapters`

数据表`chapters`有三个字段：`book_id`，`chapter_id`和`path`，分别表示书籍编号、章节编号和章节资源路径。其中书籍编号和章节编号构成的有序对为主键，所有字段均不能为空。

书签数据表：`bookmarks`

数据表bookmarks有六个字段：`bookmark_id`，`uid`，`book_id`，`name`，`chapter_id`和`page_num`
，分别表示书签编号、书签用户名、书籍编号、书签名字、书签指向的章节编号和页码。其中书签编号为主键，所有字段均不能为空。