



软件工程实验报告

实验课程: 软件工程

任课教师: 郑贵锋老师

实验题目: 软件代码标注、

UML 建模和软件需求规格说明书、

增加新需求

专业名称: 计算机科学与技术

学生姓名: 林宏宇

学生学号: 23320093

实验时间: 2025 年 5 月 23 日

项目 Github 链接: <https://github.com/fulL549/Tickets.git>

一、作业要求

此项实验报告包含了三篇综合实践的结果。

一、实践任务:标注代码。

实践方法:在软件开发环境(如 Eclipse)中标注代码,也可借助 CodePedia 等软件工具标注代码。

实践要求:对类、方法、语句块和语句等多个层次代码标注;注释要简洁和正确,确保质量;标注的代码量要有一定规模,建议在 1000—4000 行。

实践结果:具有标注的开源软件代码及其文件。

二、实践任务:分析开源软件的新需求。

实践方法:借助 UML 进行需求建模,遵循软件需求规格说明书的标准或模板撰写开源软件新增软件需求的文档。

实践要求:建立新增软件需求的用例交互模型、分析类模型和必要的状态模型,按照软件需求规格说明书的规范标准撰写相应的软件需求文档。

实践结果:开源软件新增需求的用例交互图、分析类图、状态图以及软件需求规格说明书。

三、实践任务:搜寻可支持新需求实现的开源软件或其他可重用软件资源。

综合实践

实践方法:分析开源软件的实现技术和运行环境,到开源软件托管平台中寻找合适的开源软件。

实践要求:深入理解软件设计需要满足的约束和限制,找到可有效支持新需求实现的可重用软件资源。

实践结果:无。

二、实验背景介绍

本实验报告是软件工程学科综合实践任务的项目作业,目的是通过综合实践,加强对学科的理解,提升对开源软件的理解和实际操作能力。

此次作业包含了代码标注、UML 需求建模和软件需求规格说明书、软件新需求实现三项功能,都是基于同一个项目(自动化测试),本人在作业过程中,通过实践任务要求也不断加深对项目和学科的认识理解,从 UML 建模明白框架、到代码标注理解细节、再到最后实现新需求。

三、实践项目

1. 项目名称:“抢票助手”——Android 移动端的自动化测试应用
2. 项目功能:该项目是一款自动化抢票软件,支持大麦网、淘票票、缤玩岛等多个票务平台,通过模拟人工操作实现自动登录、选择场次、提交订单等功能,提高用户抢票成功率。
3. 开源项目地址: <https://github.com/Guyungy/damaihelper.git>
4. 本次综合实践含有软件需求规格说明书,软件的具体介绍在说明书中提及

四、代码标注

1. 标注要求：

对类、方法、语句块和语句等多个层次代码标注，注释要简洁和正确

2. 层次说明：

模块级注释：说明文件作用和功能场景（自动化抢票模拟）

函数级注释：说明方法功能、参数意义及实现目标

语句块注释：标识逻辑结构，例如“配置”、“连接”、“操作”、“释放资源”

语句级注释：解释关键字段和方法调用的具体作用

3. 代码标注样例：

此处给出 appium 驱动模块的代码标注，整个项目的代码标注在代码中体现。

```
"""
Appium 模拟器模块
用于模拟手机端操作，实现自动化抢票
"""

from appium import webdriver

def start_simulation(account_info):
    """
    启动 Appium 模拟

    通过 Appium 模拟手机操作，实现在大麦 APP 上的自动化抢票

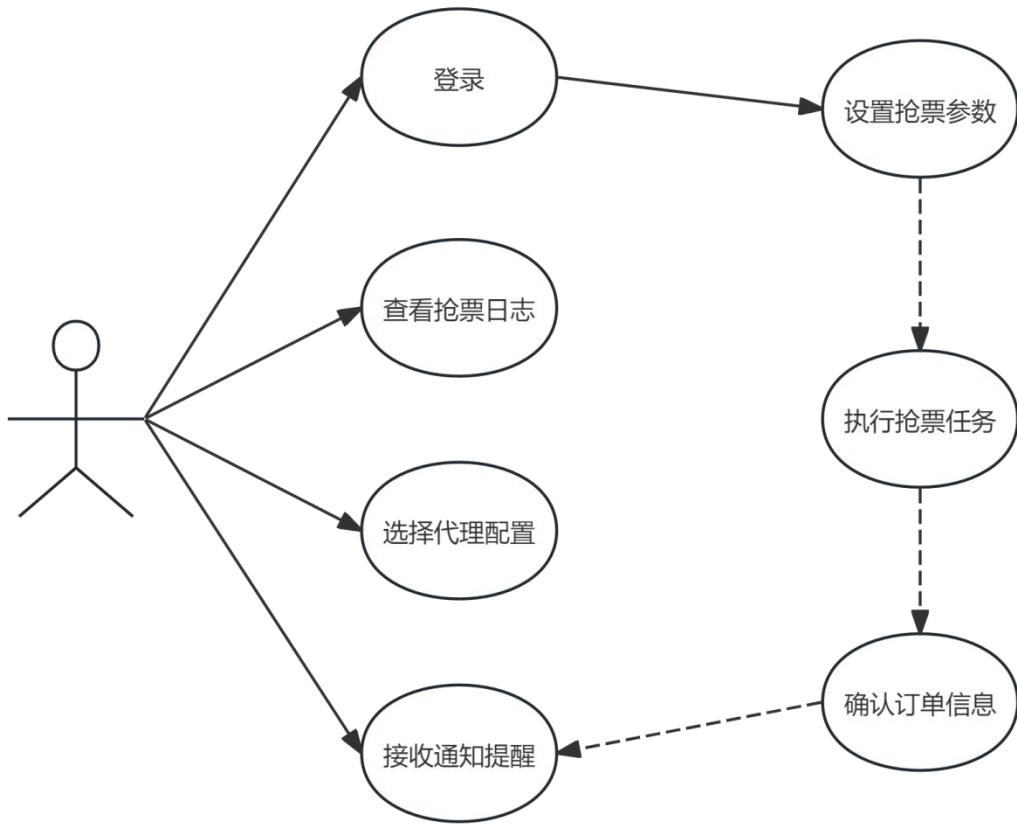
    参数：
        account_info: 账户信息字典，包含用户名、密码等信息
    """

    # 初始化 Appium 驱动
    desired_caps = {
        "platformName": "Android",           # 平台名称
        "deviceName": "KZYSWOJ7QW4TNRU4",     # 您的设备 ID
        "udid": "KZYSWOJ7QW4TNRU4",          # 确保使用正确的设备
        "appPackage": "com.damai.android",    # 大麦APP 包名
        "appActivity": ".activity.MainActivity", # APP 入口 Activity
        "noReset": True                      # 保留 APP 数据和登录状态
    }
    driver = webdriver.Remote('http://localhost:4723/wd/hub', desired_caps)

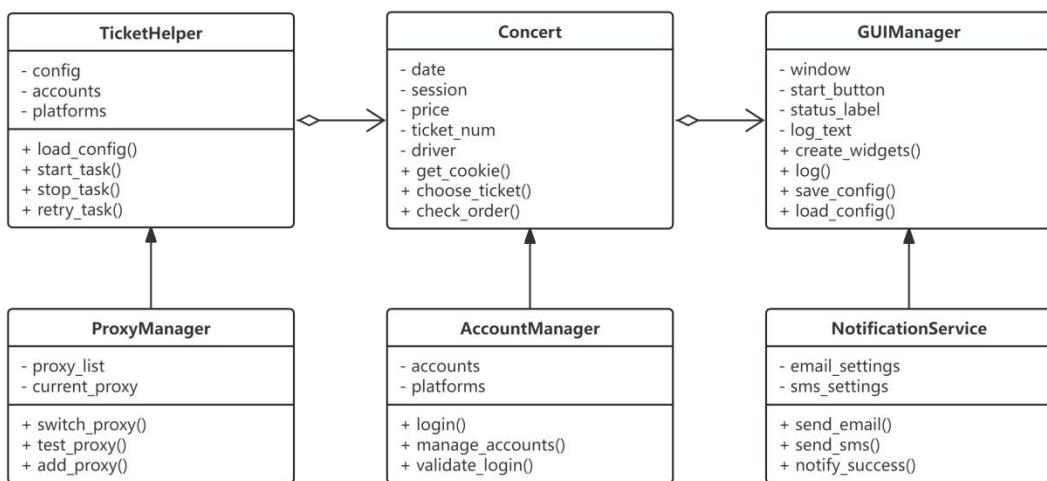
    # 执行模拟操作
    # 例如模拟点击、滑动、输入等
    driver.find_element_by_id("com.damai.android:id/login_button").click()
    driver.quit() # 关闭驱动
```

五、UML 需求建模

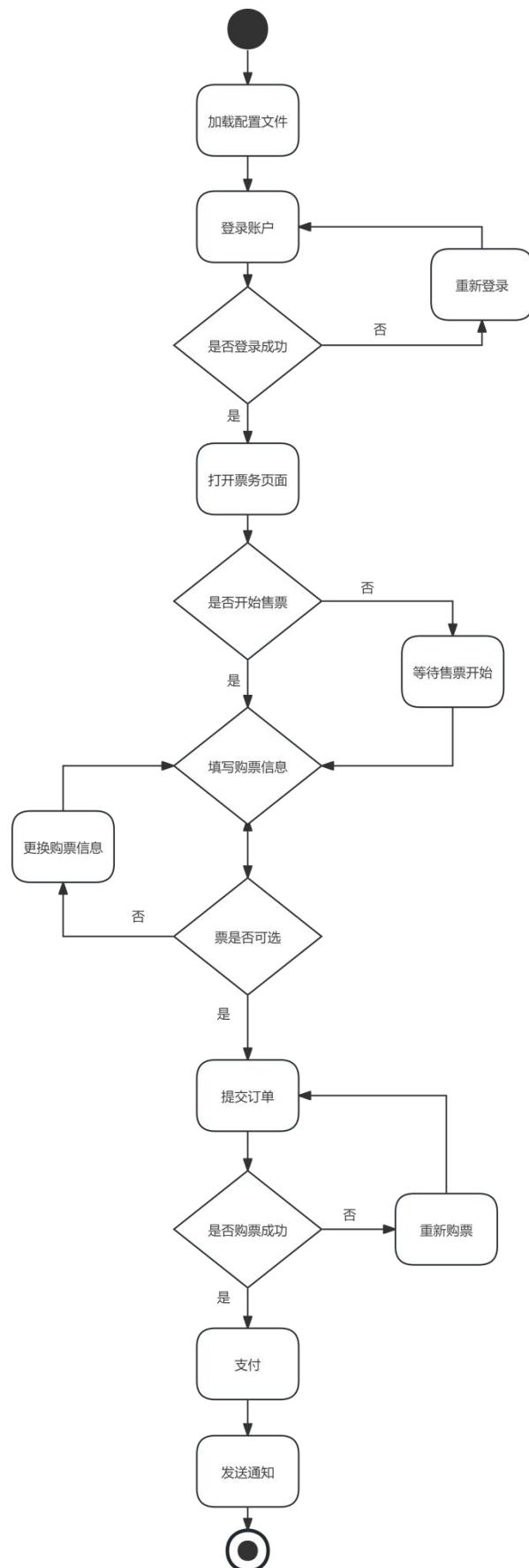
1. 用例图



2. 类图



3. 流程图



六、软件需求规格说明书(SRS)

抢票助手软件需求规格说明书

1. 引言

1.1 目的

本文档旨在详细描述“抢票助手”软件系统的需求，为开发团队提供明确的功能规范和技术要求，同时作为用户和开发者之间的沟通桥梁。

1.2 范围

“抢票助手”是一款自动化抢票软件，支持大麦网、淘票票、缤玩岛等多个票务平台，通过模拟人工操作实现自动登录、选择场次、提交订单等功能，提高用户抢票成功率。

1.3 定义、缩略语与缩写

SRS: Software Requirements Specification, 软件需求规格说明书

GUI: Graphical User Interface, 图形用户界面

OCR: Optical Character Recognition, 光学字符识别

API: Application Programming Interface, 应用程序接口

1.4 参考资料

B 站《自动化测试》BV1Fj42197Bc

https://blog.csdn.net/2401_85696278/article/details/144667919

2. 总体描述

2.1 产品概述

抢票助手是一款针对在线票务网站的自动化抢票工具，可以帮助用户在热门演出开售时自动执行抢票操作，提高抢票成功率。系统支持多账户、多平台操作，并提供代理 IP 切换、验证码识别等功能，降低被票务平台反爬虫系统拦截的风险。

2.2 产品功能

多平台支持：大麦网、淘票票、缤玩岛等票务平台

多账户抢票：同时使用多个账户增加抢票成功率

自动登录：支持扫码、短信等多种登录方式

票务选择：自动选择日期、场次、票价

实名观影人选择：按配置自动选择实名观影人

定时抢票：在指定时间自动开始抢票

代理 IP 池：防止 IP 被封或限制

验证码识别：OCR 自动识别验证码

抢票日志：详细记录抢票过程

通知提醒：抢票成功后通过邮件、短信通知

2.3 用户特征

普通用户：希望提高抢票成功率的个人用户，具备基本电脑操作能力

高级用户：需要同时购买多个场次或为多人抢票的用户，对抢票流程有较深理解

2.4 约束

浏览器兼容性：需要使用 Chrome 浏览器及对应版本的 ChromeDriver

网络环境：要求稳定的网络连接，最好是无线网络

性能限制：同时运行多个账户会增加系统资源消耗

验证码识别准确率：OCR 识别准确率会影响抢票成功率

平台反爬虫机制：各票务平台的反爬虫策略不断更新可能影响软件使用

2.5 假设和依赖

用户已完成票务平台的实名认证并添加观影人信息

用户账户中有足够余额支付订单

票务平台 API 和页面结构相对稳定

用户理解并遵守票务平台的使用条款

3. 系统功能需求

3.1 用户界面功能

3.1.1 配置管理界面

描述：提供图形化界面，用于配置抢票参数、账户信息等

输入：票务 URL、抢票时间、场次选择、票价选择、实名观影人、代理设置等

处理：保存用户配置到 JSON 文件

输出：配置保存成功提示，显示当前配置状态

3.1.2 抢票任务控制界面

描述：提供开始、停止、重试抢票操作的按钮和状态展示

输入：用户点击操作按钮

处理：执行相应的抢票流程控制

输出：任务状态、进度条、详细日志

3.1.3 日志展示界面

描述：实时显示抢票过程中的关键信息和状态

输入：系统自动记录的操作日志

处理：格式化日志信息并更新显示

输出：按时间顺序排列的日志信息

3.2 抢票核心功能

3.2.1 自动登录功能

描述：自动完成账户登录过程，支持多种登录方式

输入：账户信息、登录方式

处理：模拟用户登录操作，处理验证码，保存 Cookie

输出：登录状态、Cookie 信息

3.2.2 场次票价选择功能

描述：根据配置自动选择日期、场次和票价

输入：配置文件中的日期序号、场次序号、票价序号优先级

处理：按照优先级尝试选择可用场次和票价

输出：选择结果、可用票价信息

3.2.3 实名信息确认功能

描述：自动选择观影人和确认实名信息

输入：配置文件中的观影人序号

处理：选择对应的观影人，确认订单信息

输出：观影人选择状态、订单确认状态

3.2.4 订单提交功能

描述：自动提交订单并跳转至支付页面

输入：确认的订单信息

处理：提交表单数据，处理确认页面
输出：订单提交状态、跳转支付页面

3.3 辅助功能

3.3.1 代理 IP 管理功能

描述：管理和切换代理 IP，降低被封风险
输入：代理 IP 列表、代理设置
处理：测试代理可用性，轮换使用代理 IP
输出：当前使用的代理信息、代理状态

3.3.2 验证码识别功能

描述：自动识别并填写验证码
输入：验证码图片
处理：使用 OCR 技术识别验证码文本
输出：识别结果、填写状态

3.3.3 通知提醒功能

描述：抢票成功后通过多种方式通知用户
输入：抢票结果、通知设置
处理：生成通知内容，调用通知 API
输出：通知发送状态

4. 非功能需求

4.1 性能需求

响应时间：抢票流程中的页面操作响应时间不超过 0.5 秒
并发能力：支持同时运行至少 5 个账户的抢票任务
资源占用：单个账户抢票任务内存占用不超过 200MB

4.2 安全需求

账户信息保护：本地加密存储账户信息，不上传至远程服务器
防止恶意使用：限制单个 IP 的并发抢票数量，避免对票务平台造成过大压力
数据传输安全：与票务平台之间的数据传输使用 HTTPS 加密

4.3 可靠性需求

系统可用性：软件运行期间的稳定性不低于 99%
错误处理：对常见错误（网络波动、页面变化等）有自动恢复机制
日志记录：详细记录关键操作和错误信息，便于问题诊断

4.4 可用性需求

操作简单：提供直观的图形界面，降低学习成本
配置灵活：支持多种参数配置，适应不同场景
友好提示：对操作中的关键步骤和状态变化给予清晰提示

4.5 可维护性需求

模块化设计：系统分为 UI 层、业务逻辑层、数据访问层
代码规范：遵循 PEP 8 Python 编码规范
文档完善：提供详细的代码注释和开发文档

5. 其他需求

5.1 法律合规要求

遵守《网络安全法》等相关法律法规

不得用于商业用途或违反票务平台使用条款
用户在使用软件前需同意免责声明

5.2 用户文档

用户手册：详细的软件安装和使用指南

常见问题(FAQ)：解决常见问题的指导

视频教程：主要功能的操作演示

6. 附录

6.1 配置文件格式说明

详细描述 config.json 文件中各配置项的含义和格式要求。

6.2 项目 UML

6.3 项目代码

七、环境配置、安装

1. JAVA

下载

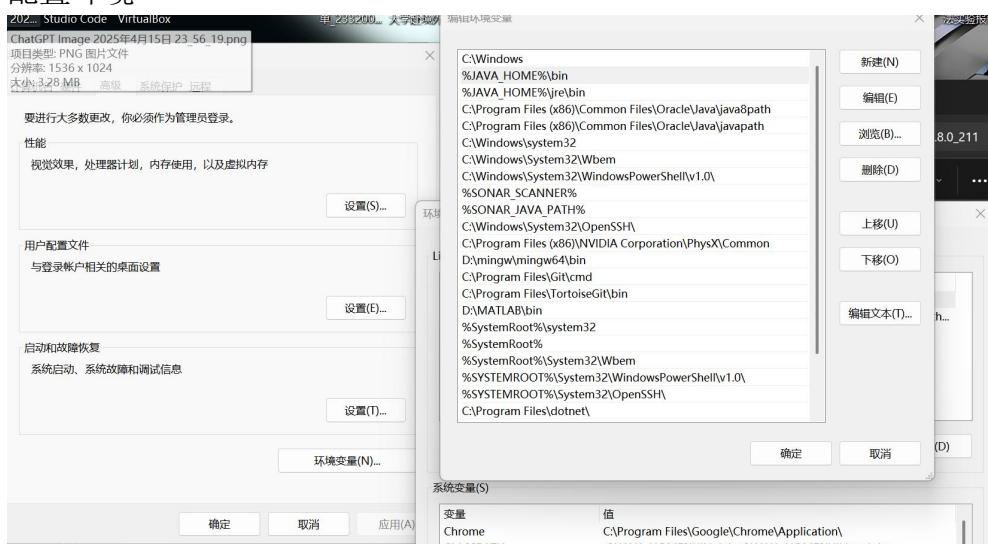


Java Mission Control 分析和诊断工具套件现在作为 JDK 的一部分提供

在我们的安装过程中，不会收集个人信息。

[有关我们所收集信息的详细信息](#)

配置环境

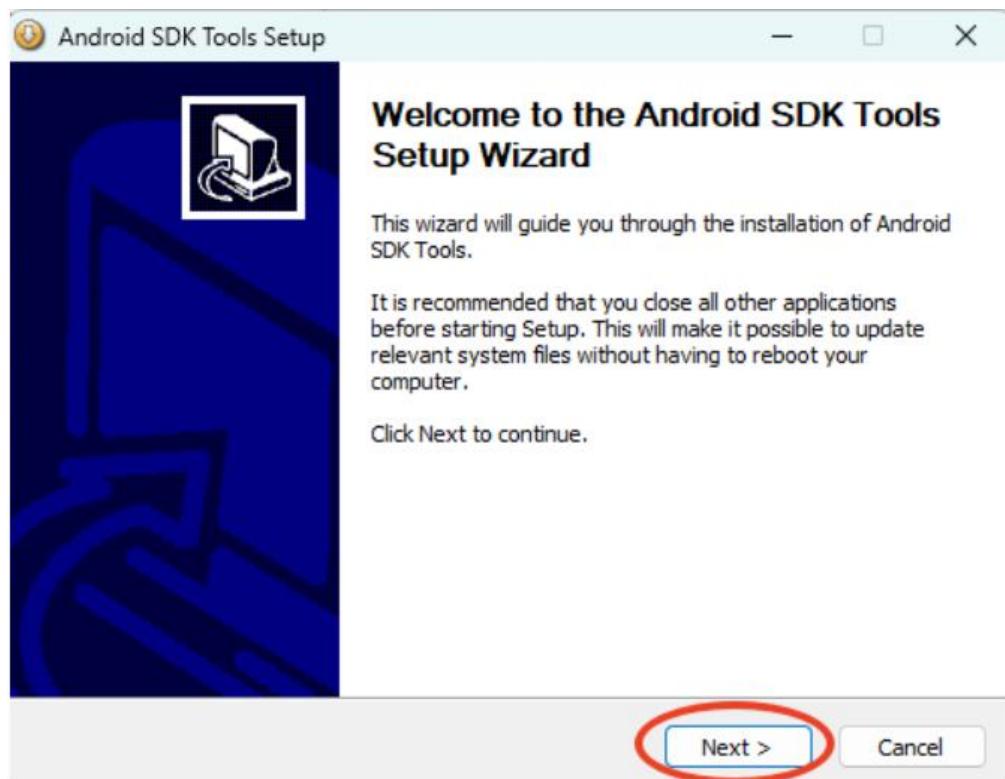


验证安装

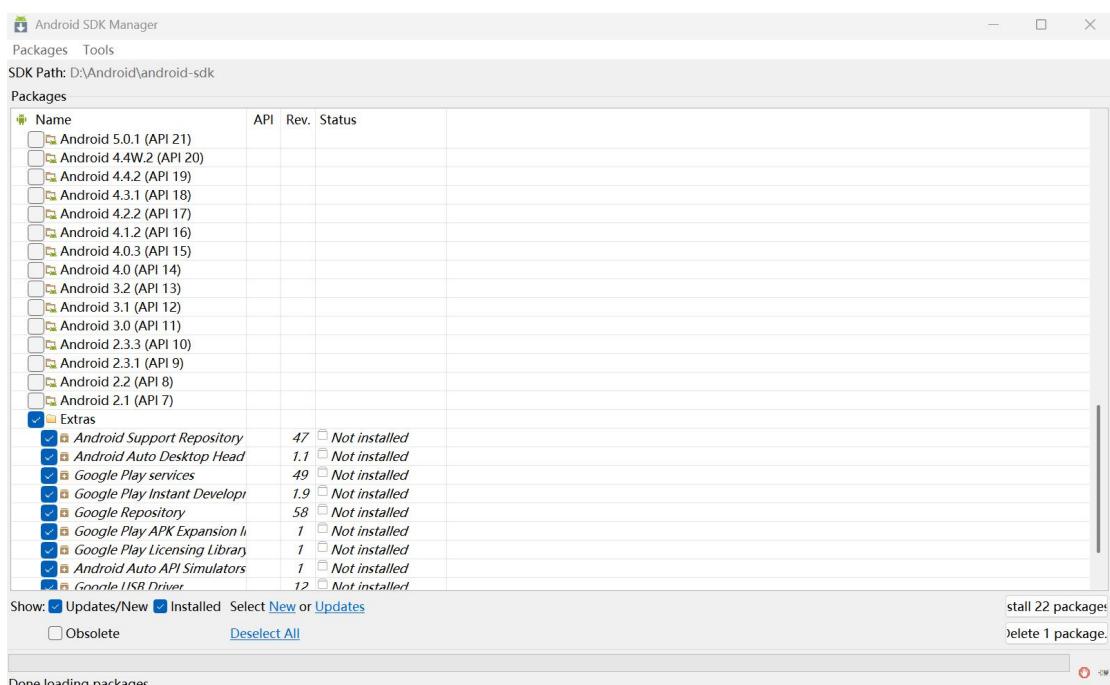
```
C:\Users\Lin Hongyu>java -version
java version "1.8.0_211"
Java(TM) SE Runtime Environment (build 1.8.0_211-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.211-b12, mixed mode)
```

2. Android

下载



安装工具



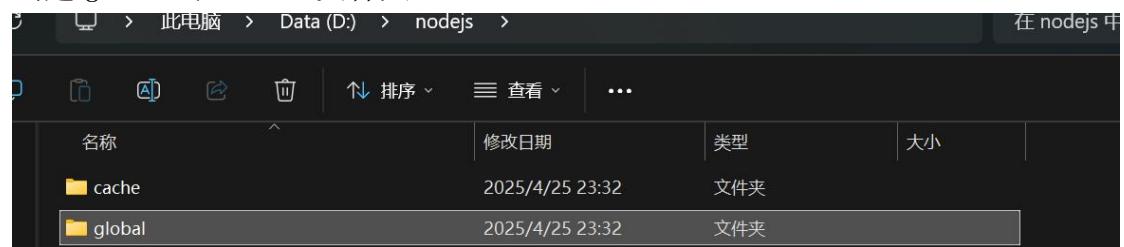
3. 下载 Node.js

node.js 上一次作业已经安装了
验证安装

```
C:\Users\Lin Hongyu>node -v  
v22.14.0
```

```
C:\Users\Lin Hongyu>npm -v  
11.3.0
```

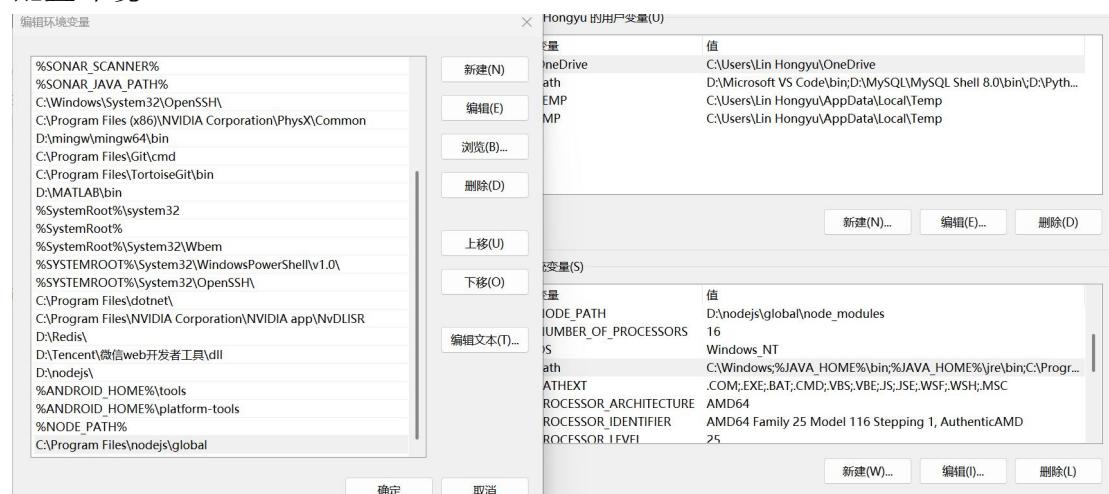
创建 global 和 cache 文件夹



配置 npm 仓库和缓存

```
C:\Windows\System32>npm config set registry https://registry.npmmirror.com/  
C:\Windows\System32>npm config set prefix "C:\Program Files\nodejs\global"  
C:\Windows\System32>  
C:\Windows\System32>npm config set cache "C:\Program Files\nodejs\cache"  
C:\Windows\System32>npm config ls  
; "builtin" config from D:\nodejs\node_modules\npm\npmrc  
; prefix = "C:\\\\Users\\\\Lin Hongyu\\\\AppData\\\\Roaming\\\\npm" ; overridden by user  
; "user" config from C:\\Users\\Lin Hongyu\\.npmrc  
cache = "C:\\\\Program Files\\\\nodejs\\\\cache"  
prefix = "C:\\\\Program Files\\\\nodejs\\\\global"  
registry = "https://registry.npmmirror.com/"
```

配置环境



检查全局配置是否完成

```
管理员: 命令提示符
Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation。保留所有权利。
C:\Windows\System32>npm install express -g
added 66 packages in 4s
14 packages are looking for funding
  run `npm fund` for details
```

安装淘宝镜像

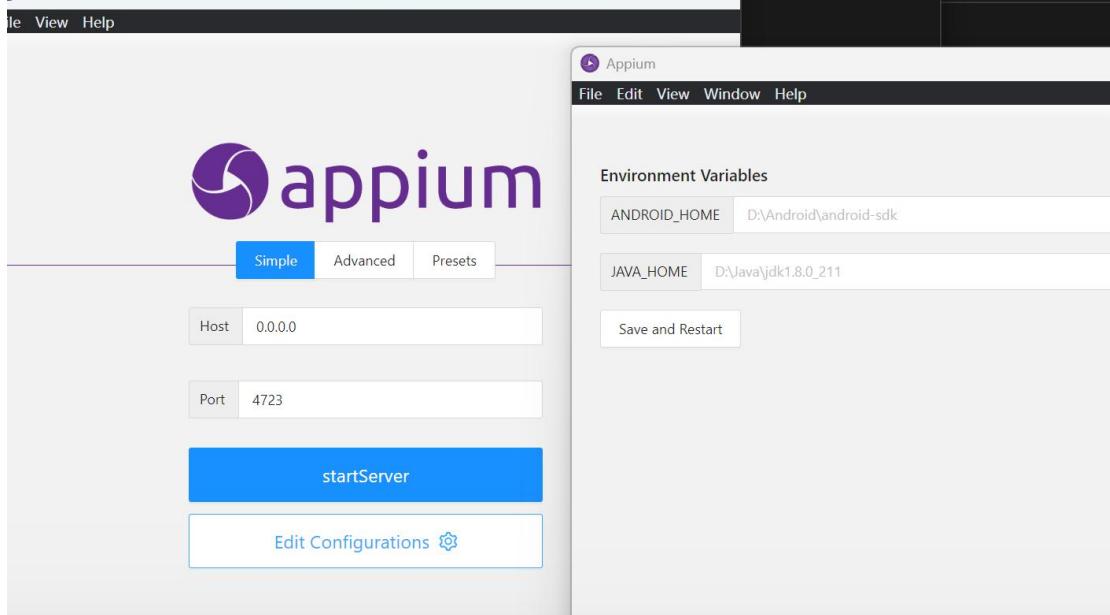
```
管理员: 命令提示符
Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation。保留所有权利。
C:\Windows\System32>npm config set registry https://registry.npmmirror.com
C:\Windows\System32>npm config get registry
https://registry.npmmirror.com
```

4. 安装 python 依赖

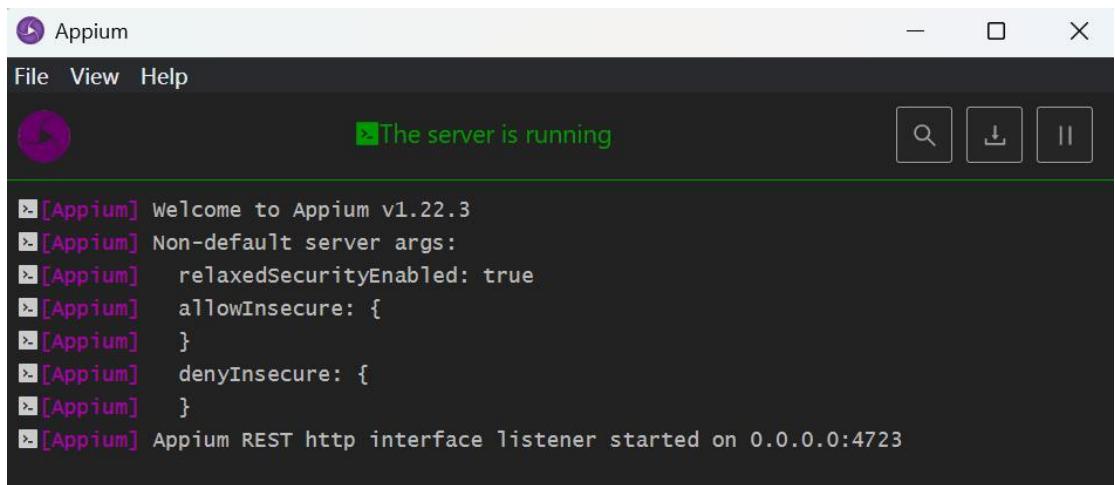
```
pip install nose
pip install selenium
pip install Appium-Python-Client
pip install apscheduler
pip install pytesseract
```

5. 下载 APPium

下载并检查配置



启动



The server is running

```
[Appium] Welcome to Appium v1.22.3
[Appium] Non-default server args:
[Appium]   relaxedSecurityEnabled: true
[Appium]   allowInsecure: {
[Appium]     }
[Appium]   denyInsecure: {
[Appium]     }
[Appium] Appium REST http interface listener started on 0.0.0.0:4723
```

6. 连接安卓移动设备

检查是否成功连接



```
命令提示符
Microsoft Windows [版本 10.0.26100.3775]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\Lin Hongyu>adb devices -l
List of devices attached
KZYSWOJ7QW4TNRU4      device product:PCDM10 model:PCDM10 device:OP4699 transport_id:8
```

八、完善软件需求

1. 现状：

开源软件许久未经过维修，功能实现不符合当下实际应用

开源软件的框架仍然可正常使用，有学习价值

需要完善原项目以符合个人实际使用

大麦等 APP 无法使用网页端进行抢票，必须使用移动端

提高自动化执行速度，真正替代手动

2. 完善目标：

先落地，能够模仿手动操作过程

查缺漏，应对机器人测试等问题

做完善，提高执行速度，更好的交互界面

3. 实现过程：

自动化测试规则学习

学习使用界面查看工具，用于查找 id、操作等

在案例基础之上拓展功能，适配自己使用的自动化测试过程

4. 软件新增结果和说明在代码和 README 文档中体现

九、项目结构分析

```
damaihelper-main/
├── config/                                # 配置文件目录
│   ├── config.json                         # 主要配置文件
│   └── config_clean.json                   # 无注释版配置文件
├── scripts/                               # 脚本模块目录
│   ├── main.py                             # 入口模块
│   ├── appium_simulator.py                # Appium 模拟器模块
│   ├── selenium_driver.py                 # Selenium 驱动模块
│   ├── multi_account_manager.py          # 多账户管理模块
│   ├── scheduler.py                       # 任务调度器模块
│   └── captcha_solver.py                 # 验证码识别模块
├── logs/                                   # 日志目录
│   ├── confirm.html                        # 确认页面模板
│   ├── GUI.py                            # 图形用户界面程序
│   └── ticket_script.py                  # 主抢票脚本
└── chromedriver.exe                      # Chrome 浏览器驱动
```

十、总结与思考

1. 通过本次软件工程综合实践，我深刻体会到了从需求分析到代码实现，再到软件完善的完整工程流程，也切实提升了我在开源软件理解与改造方面的能力。整个过程不仅是对课堂所学理论知识的巩固，更是一次系统的实战演练。
2. 首先，在代码标注阶段，我逐行分析了“抢票助手”项目的核心模块，特别是Appium模拟器与自动化任务调度相关的脚本。在标注过程中，我不仅理解了各方法和语句块的具体功能，还通过添加模块级和函数级注释进一步梳理了系统结构和逻辑流程。这一过程极大提高了我对他人代码的阅读和维护能力，也让我体会到了高质量注释在实际开发中的重要性。
3. 其次，在UML建模与需求文档编写阶段，我第一次完整地从用户视角出发，设计并建模了软件新增需求，包含用例图、分析类图与状态图。在撰写《软件需求规格说明书》的过程中，我参考标准模板对软件功能与非功能需求进行了详细定义，也深入理解了如何将用户需求转化为清晰、可实现的技术规范。这个过程让我认识到：规范化的需求分析是软件项目成功的基础。
4. 在开源资源分析与功能拓展阶段，我结合自己的实际使用场景（移动端抢票），对原项目的架构、可用性进行了深入分析，并尝试改进其在模拟速度、验证码识别等方面的性能瓶颈。通过调研现有OCR、代理IP库等模块，并在项目中测试集成，我初步具备了对已有系统进行“软重构”和“场景适配”的能力。
5. 综合来看，这次实践让我不仅熟悉了如何与开源软件打交道，更重要的是理解了软件工程的“完整闭环”——需求、设计、实现、验证、演进。我也深刻体会到文档编写、代码注释、设计建模等“看似非技术”的环节对工程质量的巨大

影响。未来在学习或工作中，我将更加重视软件规范性与协作性，努力提升自己成为一个既懂编码、也懂工程流程的综合型开发者。

十一、 参考文献

参考资料在软件需求文档中提及