



中山大學
SUN YAT-SEN UNIVERSITY

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： 开源软件分析与维护

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 20 日

一、作业要求

综合实践

实践任务:熟练掌握开源软件的编程语言,熟练掌握支撑开源软件开发的 **CASE 工具及其使用**。

实践方法:针对开源软件的编程语言,熟练掌握其使用及编程规范;基于开源软件的维护要求,选择并熟练

掌握相应的 **CASE 工具**。

实践要求:熟练掌握并能有效使用开源软件维护的编程语言及 **CASE 工具**。

实践结果:无。

综合实践

实践任务:编写所开发软件系统的程序代码。

实践方法:基于软件设计模型和文档,借助所选定的程序设计语言,利用编码、测试和调试等 **CASE 工具**,编写目标软件系统的源程序代码,并对代码进行单元测试和调试,以发现和解决代码中存在的缺陷。

实践要求:基于设计模型和文档编写代码,要对所编写的代码进行质量保证,以发现和解决代码中的缺陷。

实践结果:目标软件系统的源程序代码。

二、CASE 工具介绍

1. Enterprise Architect (EA) —— 系统建模工具

Enterprise Architect (简称 EA) 是由 Sparx Systems 推出的一款功能强大的统一建模工具 (UML CASE 工具)。它支持整个系统开发生命周期的建模和管理,涵盖了从需求分析到设计实现的全流程。主要功能包括:

- 支持 UML2.0 标准建模: 可创建类图、用例图、时序图、活动图等各种 UML 图形。
- 模型管理与版本控制: 支持包结构管理, 可与版本控制系统集成。
- 支持逆向工程和代码生成: 可根据已有代码自动生成模型, 也可根据模型生成代码骨架。
- 适用领域广泛: 不仅用于软件工程, 也适用于嵌入式系统、企业架构等领域。

EA 在本实验中用于设计自动化脚本项目系统的结构, 帮助理清项目功能模块之间的逻辑关系。

2. JIRA —— 敏捷项目管理工具

JIRA 是由 Atlassian 公司开发的业界主流项目管理平台, 广泛应用于敏捷开发、缺陷追踪和版本规划等任务中。其主要特点包括:

- Scrum 与 Kanban 支持: 可灵活切换任务面板和管理模式。
- 任务与缺陷管理: 能详细跟踪每一个任务或 Bug 的状态、负责人、时间节点。
- 与开发工具集成良好: 可与 GitHub、Bitbucket、Confluence 等工具深度集成。

- 可视化统计分析：通过燃尽图、版本进度图等方式展现项目健康状态。

本实验利用 JIRA 实现任务的创建、进度追踪、缺陷记录及版本发布安排，模拟真实开发团队协作流程。

3. Appium —— 移动端自动化测试工具

Appium 是一个跨平台的开源自动化测试框架，支持 Android 和 iOS 系统。使用 WebDriver 协议驱动移动端应用程序，无需对 App 进行重新编译。主要优点包括：

- 多平台兼容性：一个测试脚本可适用于多个平台。
- 与主流语言兼容：支持使用 Python、JavaScript、Java 等编写测试脚本。
- 元素定位灵活：结合 Appium Inspector 和 uiautomator2 可精确定位元素。
- 支持真机与模拟器测试：便于全面覆盖测试场景。

在本实验中，Appium 被用于编写自动化测试脚本，实现对 App 操作流程的回归测试，提高测试效率与准确性。

4. Git —— 分布式代码版本控制工具

Git 是当前最主流的版本控制系统，具备高效的分支管理能力，适合多人协同开发。其核心特性包括：

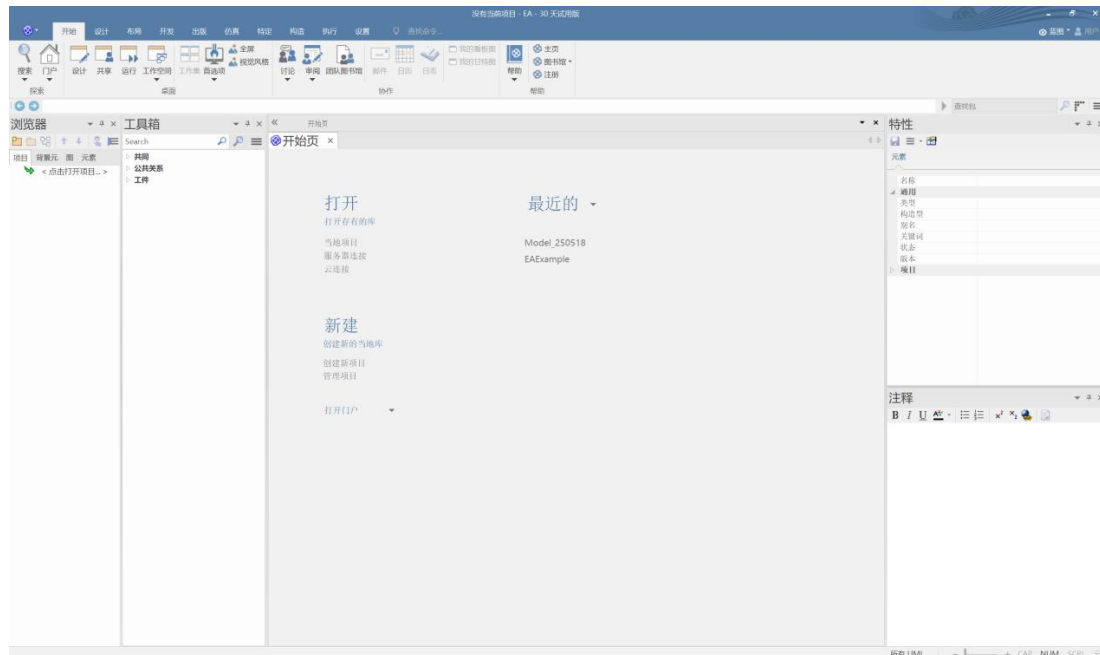
- 分布式结构：每个开发者本地都有完整的代码库副本。
- 分支合并机制强大：便于进行实验性开发。
- 与 GitHub 集成良好：支持线上协作与代码审查。
- 支持与 IDE（如 VSCode）无缝集成：大大提升日常开发效率。

在实验中，Git 被用于代码版本控制与管理，通过 GitHub 远程仓库进行团队协作开发和代码备份。

三、基于自动化脚本项目系统项目的 Enterprise Architect 设计建模

1. 下载安装

Enterprise Architect (EA) 作为一款强大的建模工具，支持多种建模语言（如 UML、BPMN 等），并可与版本控制、需求跟踪等功能无缝集成。通过官网下载或镜像站获取 EA 安装程序，根据引导完成安装。



2. EA15 破解版

由于 EA 免费版只有 30 天的试用期，使用网上大神的时间重置脚本可以无限续杯。破解之前，提示只剩下 13 天：



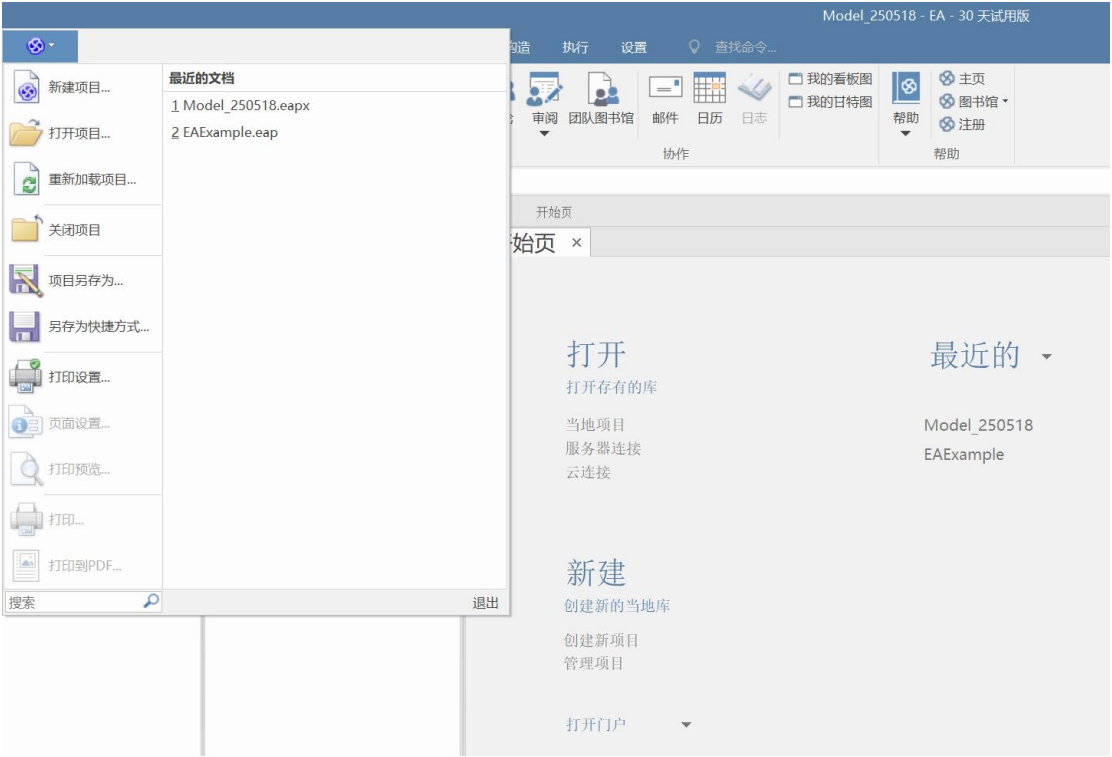
破解后，满血复活：



3. EA 工具的使用方法

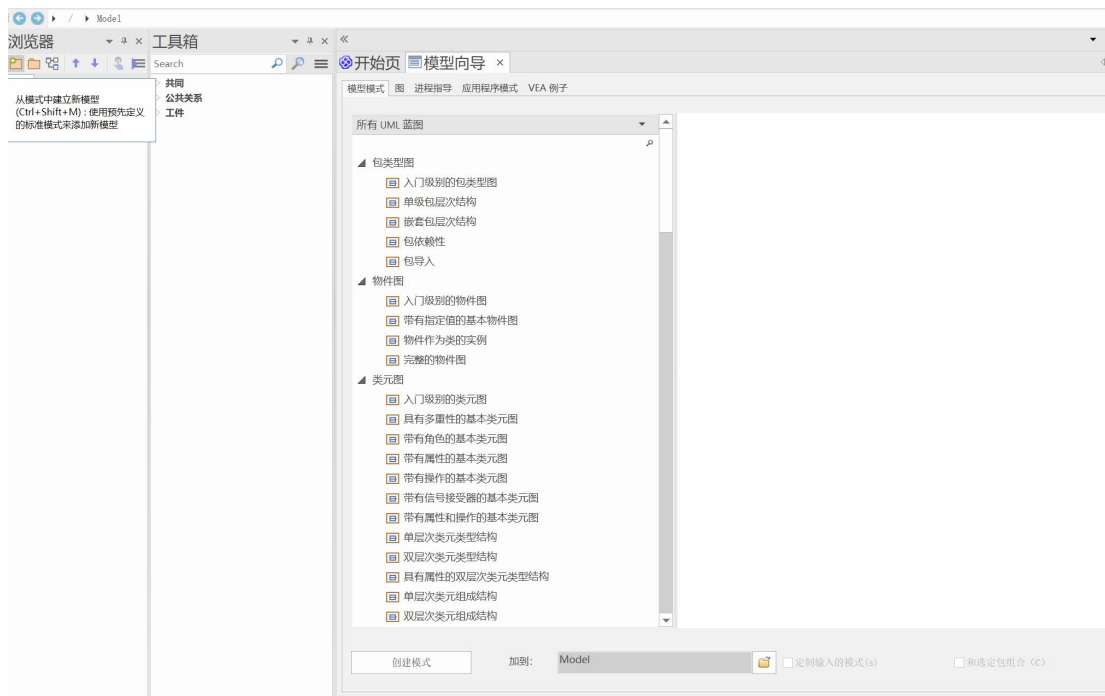
3.1 新建项目

打开 EA 后选择“新建项目”，设定项目名称与保存路径。项目文件保存为.eapx 格式，可后续编辑与版本控制



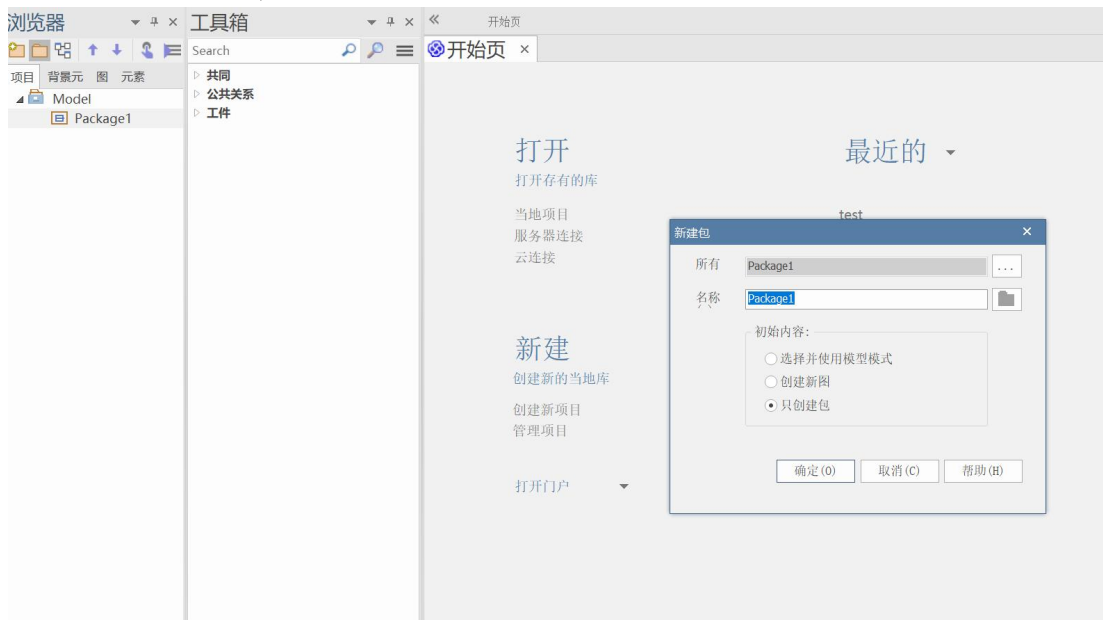
3.2 从模式中添加模型

EA 提供多个建模模板（如“软件开发”、“业务建模”），可直接加载标准模型结构，快速搭建系统蓝图。



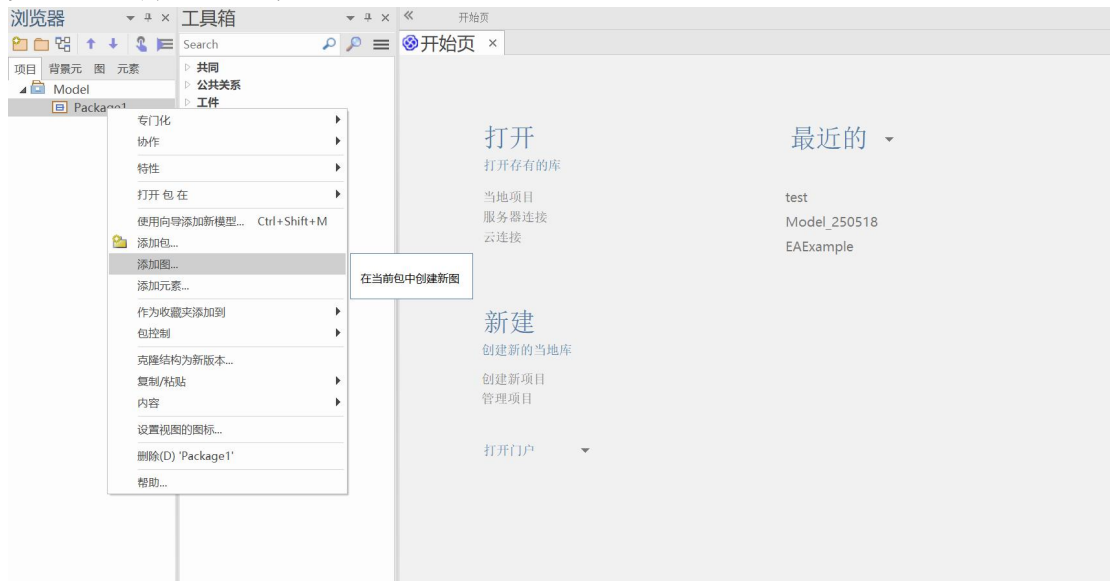
3.3 创建包

右键项目浏览器中新建“包”，用于模块化组织模型结构，例如创建“用户管理”、“测试用例”等功能包



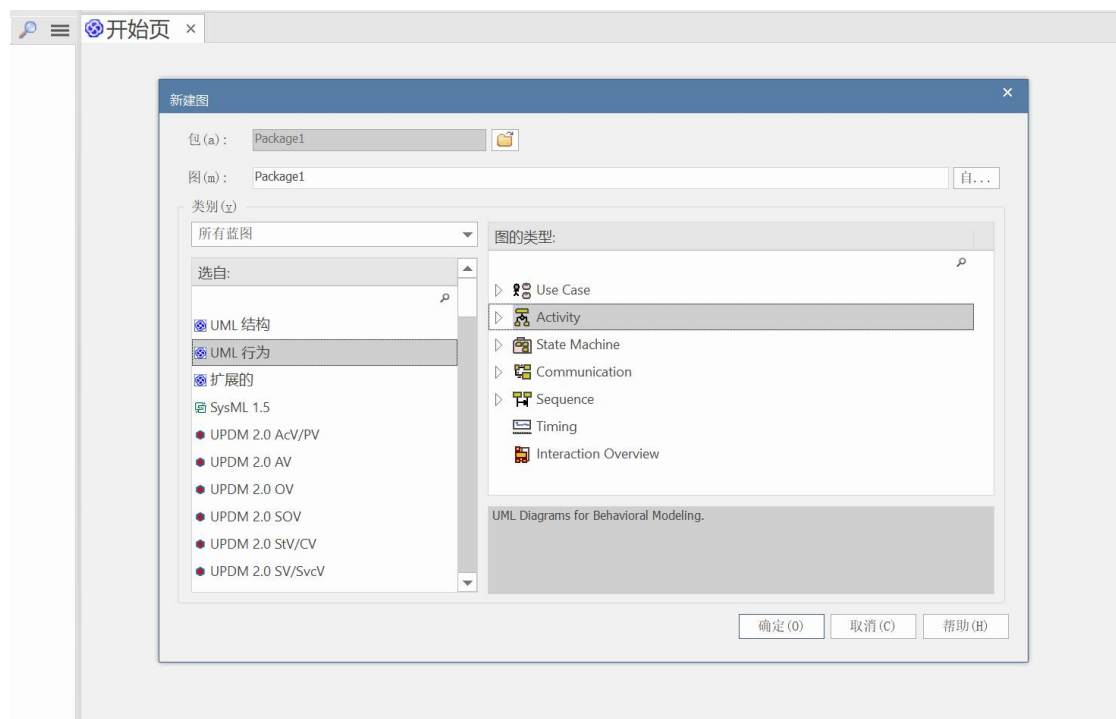
3.4 创建图

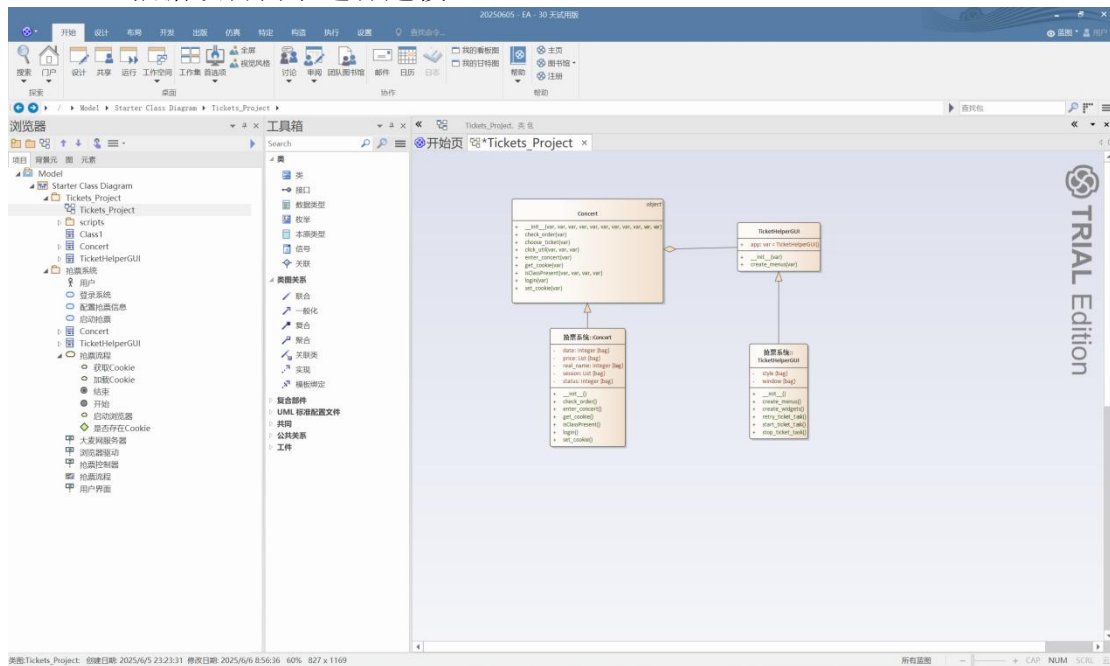
在包中右键选择“添加图表”，可以创建类图、流程图、活动图、用例图等。每类图表支持丰富元素与连接关系配置。



3.5 选择 UML

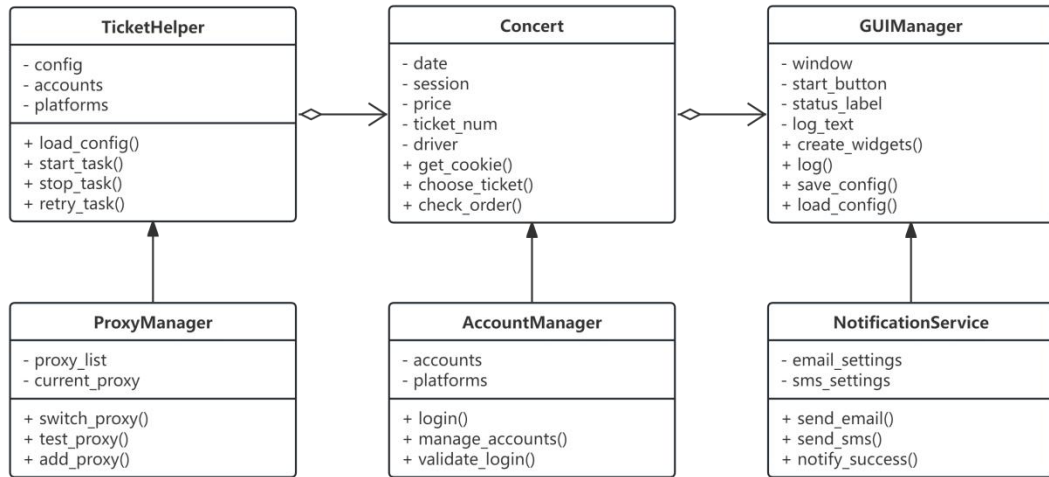
EA 内建完整 UML 标准图类型，涵盖软件设计各阶段，如类图支持继承、聚合、组合等关系表达。



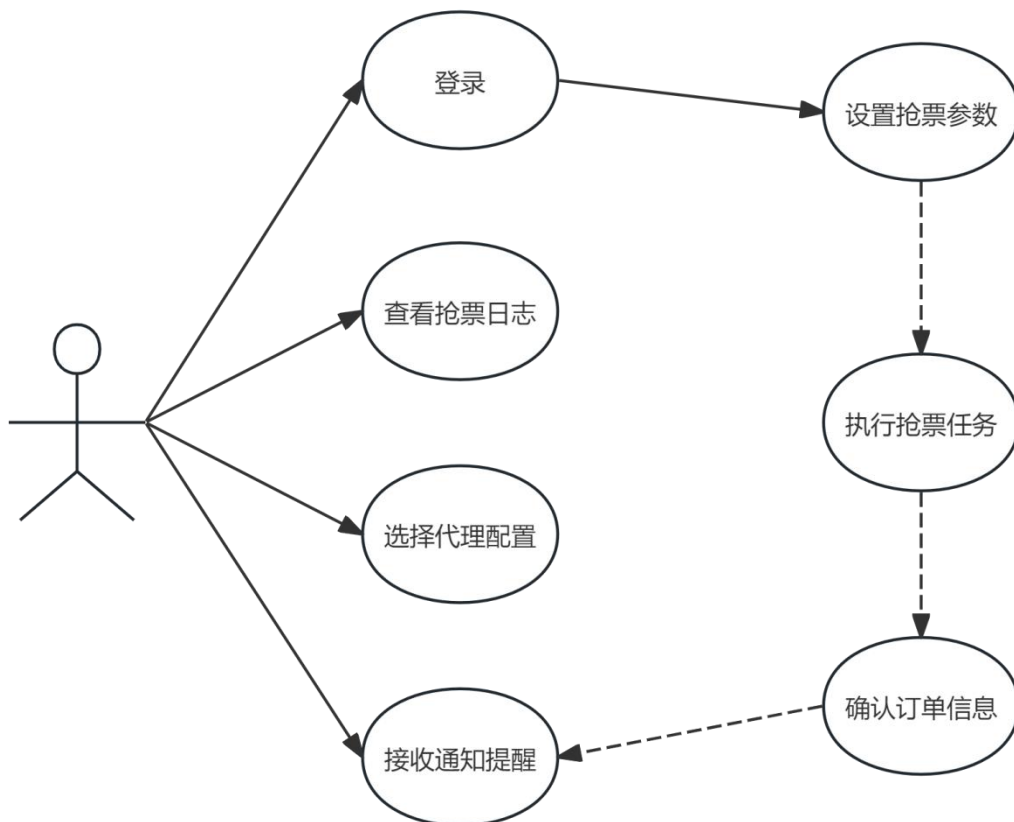


四、EA 项目建模

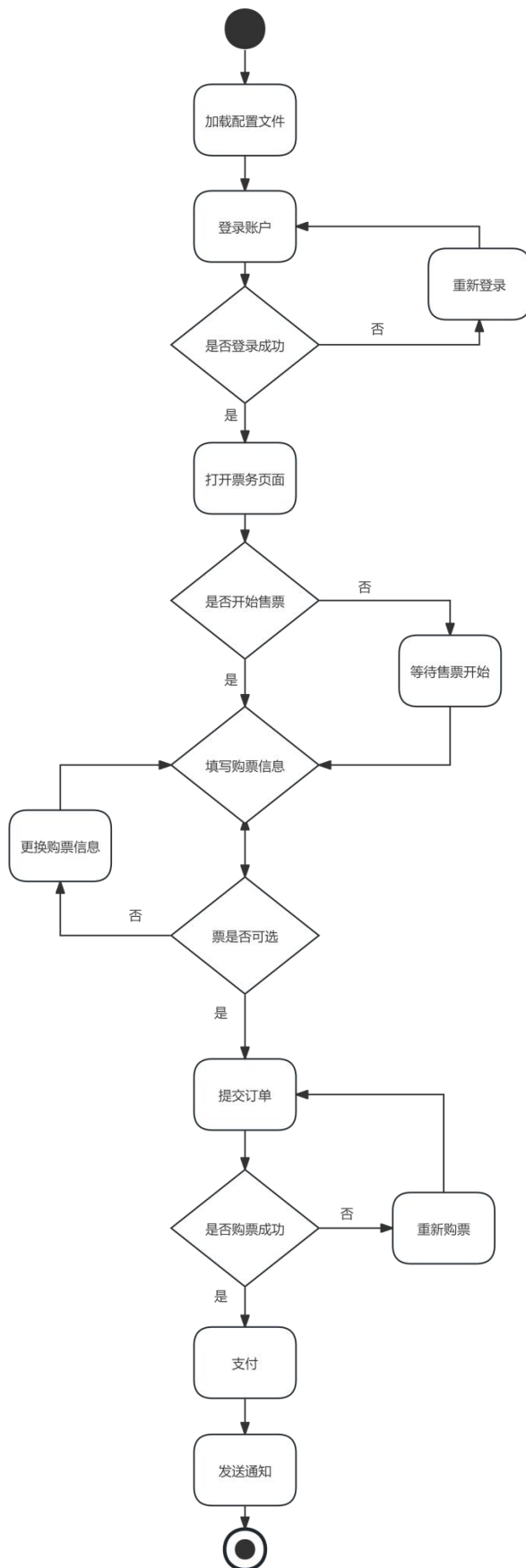
1. 类图



2. 用例图



3. 流程图



五、JIRA

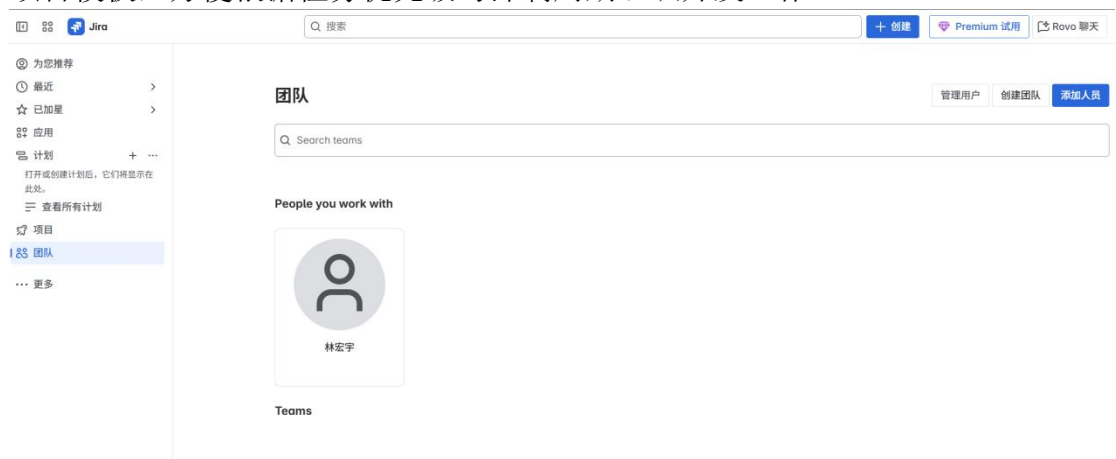
1. Atlassian 的 JIRA 工具网站地址

<https://www.atlassian.com/>



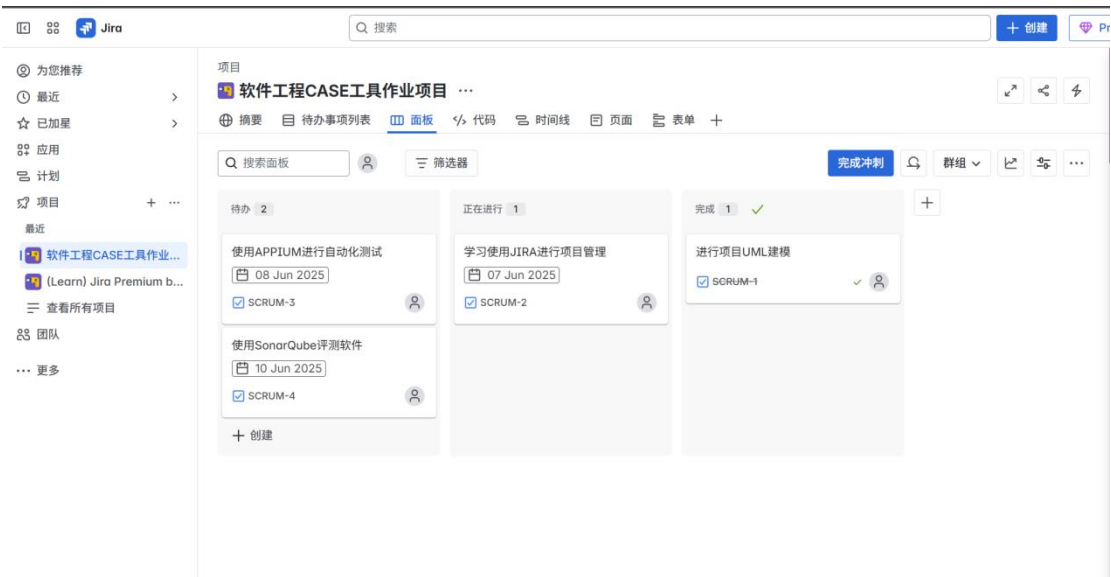
2. 创建账户和域名进行使用，可选择个人开发或团队开发

注册 Atlassian 账户后可创建个人/团队域名，创建“Scrum”或“看板(Kanban)”项目模板，方便根据任务优先级与冲刺周期组织开发工作。



3. 面板代办事务管理

JIRA 的看板视图通过“待办 - 进行中 - 已完成”三个阶段展示任务状态，有助于团队成员实时了解进度



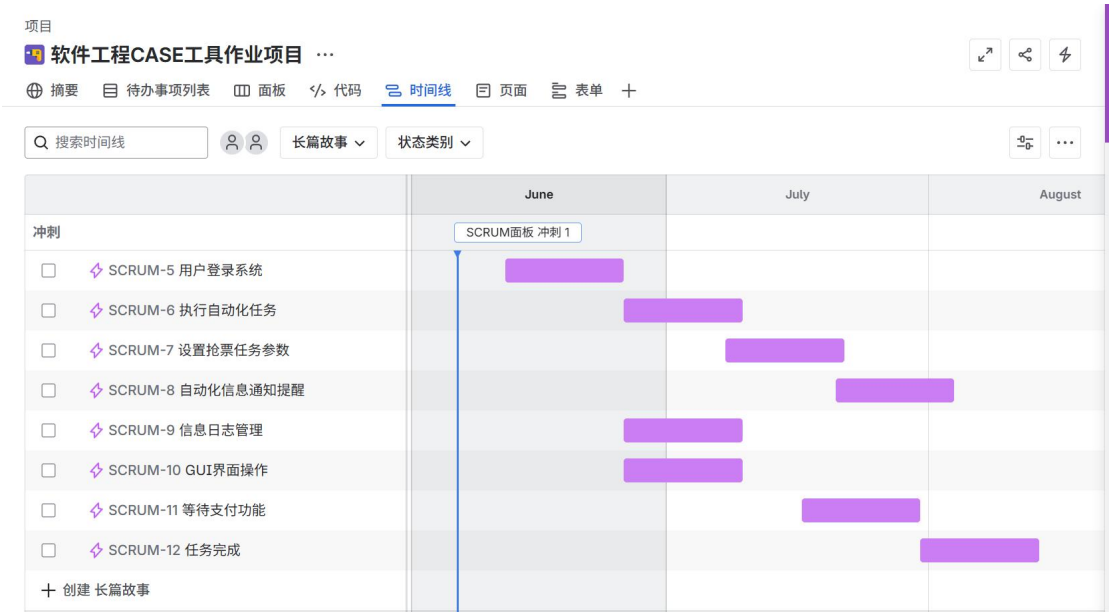
4. 为代办添加详细信息

为每个代办事项添加详细描述、优先级、负责人、标签等信息，便于精准追踪和责任划分。



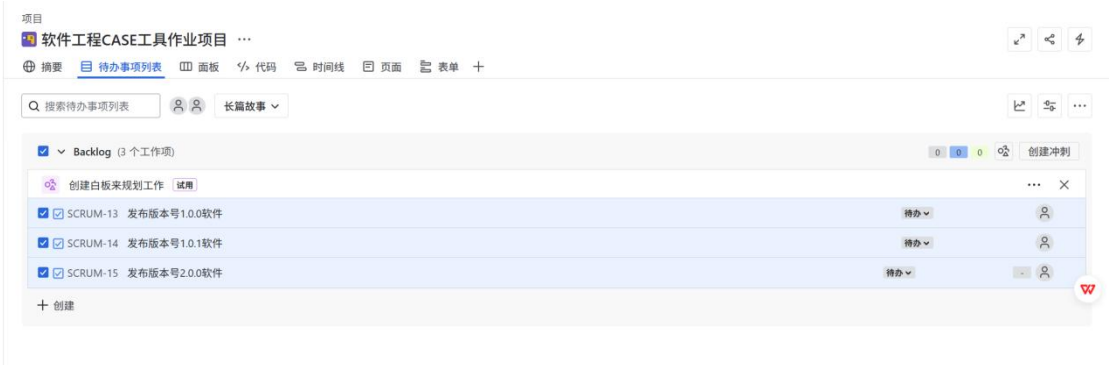
5. 时间线功能

JIRA 支持甘特图视图，帮助规划各模块开发与交付节奏，提高时间管理效率。



6. 使用项目管理工具来管理版本号

支持按版本编号发布功能更新，并记录每次发布关联的任务与缺陷，确保变更可溯。



7. 还可以使用团队开发模式，进行任务分配管理

团队成员可在任务中评论、上传附件，实现协同开发与透明沟通。

8. 使用进度跟踪，实时查看项目剩余进度

通过燃尽图、冲刺报告等可视化工具分析进展与预测完成时间。

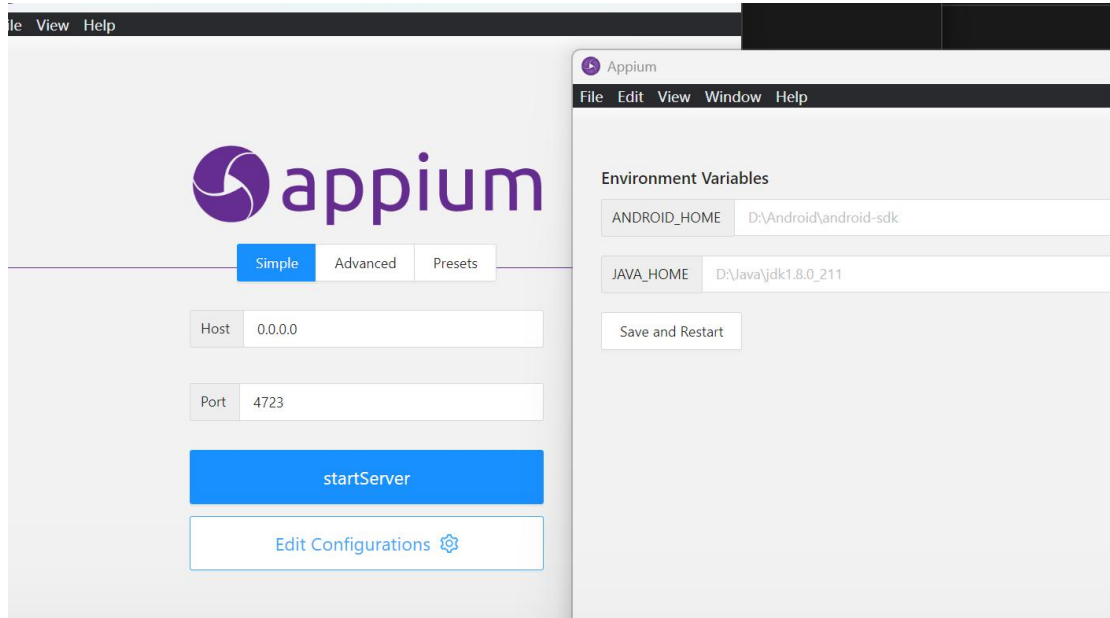
9. 将项目管理工具连接 Github 代码管理工具

JIRA 可与 Github 账户绑定，关联提交记录至 JIRA 任务卡，实现“代码即文档”的一体化管理。

六、基于自动化脚本项目的 Appium 自动化测试

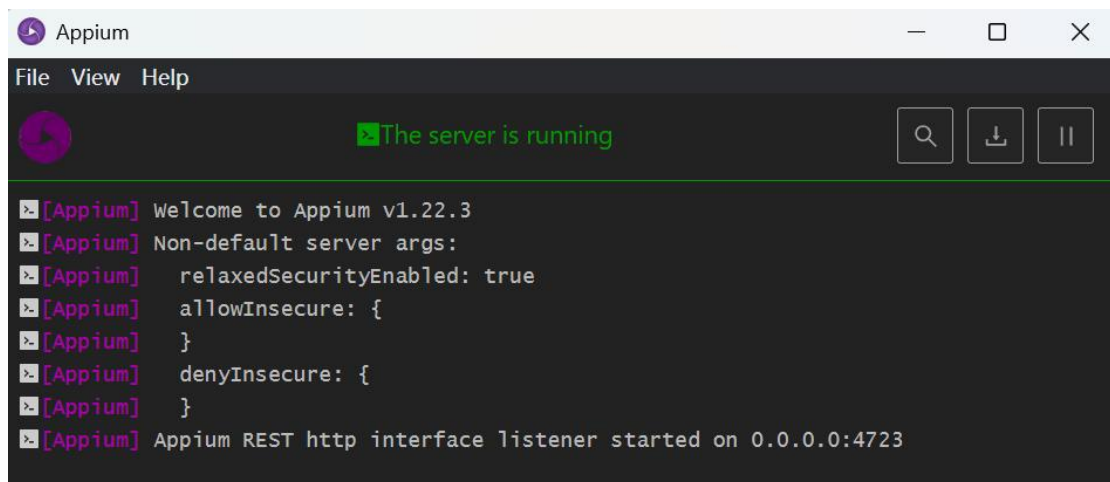
1. 下载安装 appium

下载并检查配置



2. 启动服务

启动 Appium Server 后会监听 `http://localhost:4723/wd/hub` 接口，等待测试脚本连接。



3. 连接安卓移动设备

检查是否成功连接



七、Appium inspector 自动化测试工具

1. 下载 appium inspector

下载地址

<https://appium.github.io/appium-inspector/latest/quickstart/installation/#>

2. Appium inspector 使用命令

与所有 Appium 插件一样，可以使用 Appium 命令行安装和激活 Inspector 插件。

1. 安装插件：

```
appium plugin install --source=npm appium-inspector-plugin
```

笔记

Appium 3 还将支持该 `appium plugin install inspector` 命令

1. 启动已激活插件的 Appium 服务器：

```
appium --use-plugins=inspector --allow-cors
```

1. 在您的 Web 浏览器中打开检查器 URL：

```
http://localhost:4723/inspector
```

信息

确保上述主机 URL 和端口与 Appium 服务器的 URL 和端口匹配。请注意，服务器的基本路径值将被忽略 - 插件始终使用该 `/inspector` 路径。

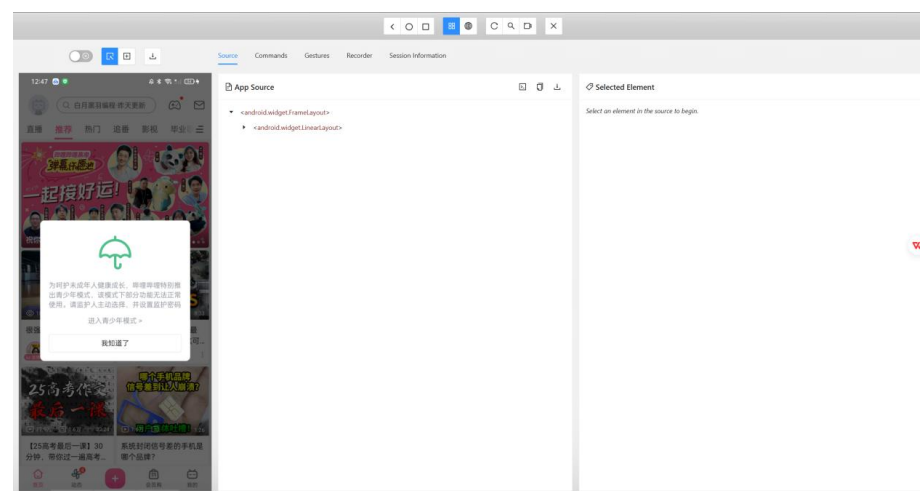
3. 下载 uiautomator2

```
管理员: Windows PowerShell
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

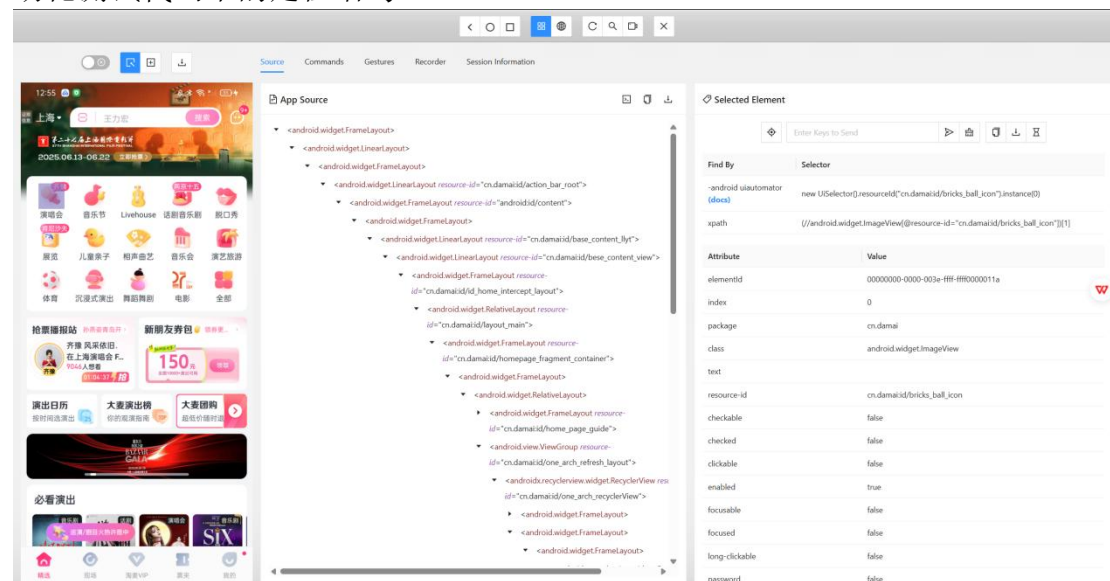
安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS C:\WINDOWS\system32> appium driver install uiautomator2
√ Checking if 'appium-uiautomator2-driver' is compatible
√ Installing 'uiautomator2'
i Driver uiautomator2@4.2.3 successfully installed
+ automationName: UiAutomator2
+ platformNames: ["Android"]
PS C:\WINDOWS\system32>
PS C:\WINDOWS\system32>
```

点击“Start Session”，连接成功后进入被测 APP，Inspector 会展示应用当前页面的 UI 层级结构。



选中某一元素，可查看其属性（resource-id、text、xpath 等），用于编写自动化测试代码中的定位语句



6. 常用定位方式

6.1 按 ID 定位

```
driver.find_element(By.ID, "element_id")
driver.find_elements(By.ID, "element_id") # 多个元素
```

6.2 按 Xpath 定位

```
driver.find_element(By.XPATH, "//android.widget.TextView[@text='登录']")
"//android.widget.EditText[contains(@resource-id, 'username')]"
```

6.3 按 Class Name 定位 (类名)

```
driver.find_elements(By.CLASS_NAME, "android.widget.TextView")
```

6.4 按 Accessibility ID 定位

```
driver.find_element(By.ACCESSIBILITY_ID, "Login")
```

6.5 按 Android UIAutomator 定位

```
driver.find_element(AppiumBy.ANDROID_UIAUTOMATOR,
    'new UiSelector().text("登录")')
'new UiSelector().className("android.widget.Button").text("提交")'
```

7. 常用 appium 操作函数

| 功能 | 函数示例 |
|-----------------|--|
| 点击元素 | <code>element.click()</code> |
| 清除输入框 | <code>element.clear()</code> |
| 输入文本 | <code>element.send_keys("text")</code> |
| 获取文本 | <code>element.text</code> |
| 获取属性 | <code>element.get_attribute("resource-id")</code> |
| 判断元素存在 | <code>len(driver.find_elements(...)) > 0</code> |
| 判断是否显示 | <code>element.is_displayed()</code> |
| 获取页面源代码 | <code>driver.page_source</code> |
| 获取当前页面 Activity | <code>driver.current_activity</code> |
| 获取当前上下文 | <code>driver.context</code> |

八、Appium 自动化测试代码

1. 自动进入指定 app 页面设置

该段代码实现自动启动大麦 APP，并初始化 driver 进行后续 UI 操作。

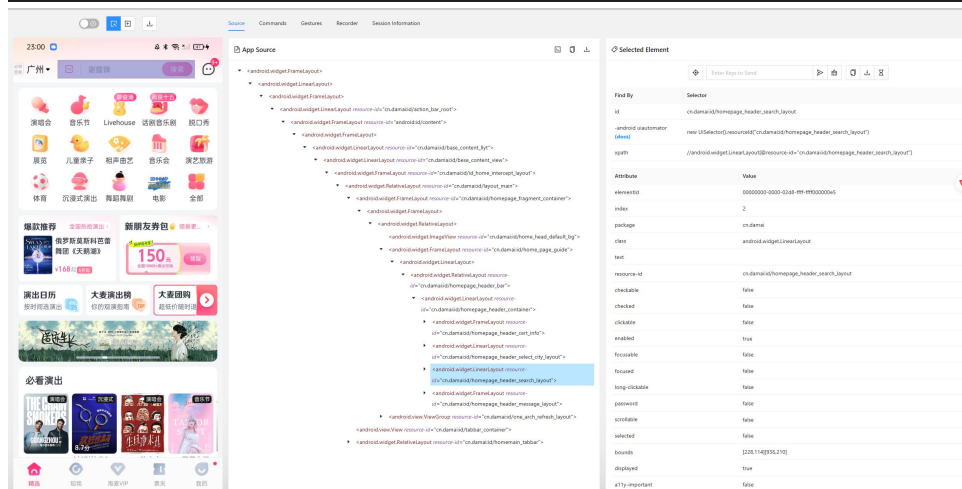
```
# 使用 UiAutomator2Options 创建选项
desired_caps = {
    'platformName': 'Android', # 被测手机是安卓
    'platformVersion': '11', # 手机安卓版本，如果是鸿蒙系统，依次尝试 12、11、10 这些版本号
    'deviceName': 'KZYSW0J7QW4TNRU4', # 设备名，安卓手机可以随意填写
    'appPackage': 'cn.damai', # 启动 APP Package 名称
    'appActivity': '.homepage.MainActivity', # 启动 Activity 名称
    'unicodeKeyboard': False, # 禁用 Appium 的 Unicode 输入法
    'resetKeyboard': False, # 禁用键盘重置
    'noReset': True, # 不要重置 App
    'newCommandTimeout': 6000,
    'automationName': 'UiAutomator2',
    'skipServerInstallation': False, # 跳过 UIAutomator2 服务器安装
    'skipDeviceInitialization': False, # 跳过设备初始化
    'ignoreHiddenApiPolicyError': True, # 忽略隐藏 API 策略错误
    'autoGrantPermissions': True, # 自动授予应用权限
    # 'app': r'd:\apk\bili.apk',
}
```

```
print(u'###通过 Appium 连接手机，进入大麦网###')
# 创建 options 对象并加载所有 capabilities
options = UiAutomator2Options().load_capabilities(self.desired_caps)
# 连接 Appium Server，初始化自动化环境
self.driver = webdriver.Remote('http://localhost:4723/wd/hub', options=options)
#self.driver = webdriver.Remote('http://localhost:4723', options=options)

# 设置缺省等待时间
self.driver.implicitly_wait(5)
```

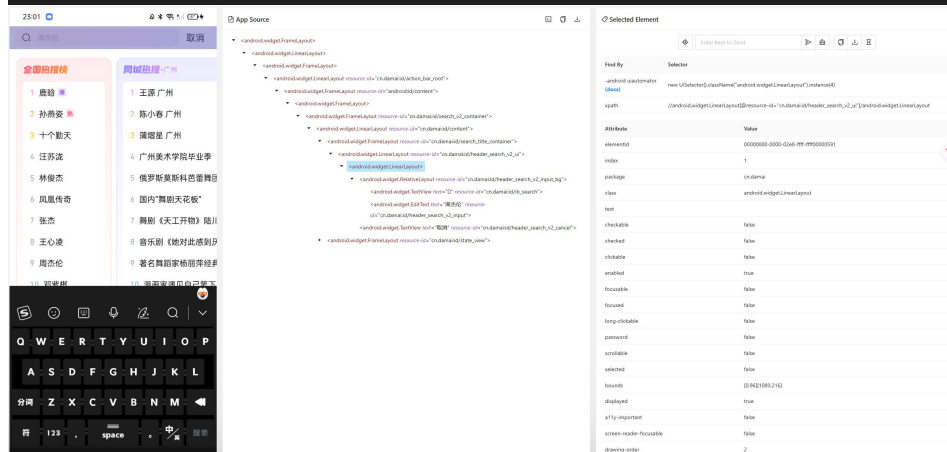
2. 查找搜索框的定位元素编写代码

```
driver.find_element(By.ID, "homepage_header_search_layout").click()
```



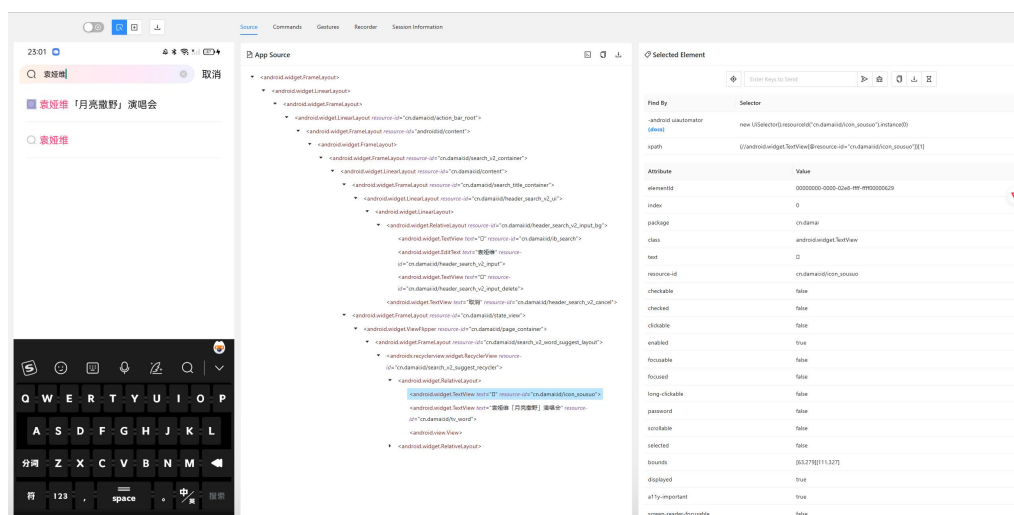
3. 点击搜索框输入内容“xxx”

```
driver.find_element(By.ID, "header_search_v2_input").send_keys("袁娅维")
```



4. 根据搜索内容进行点击

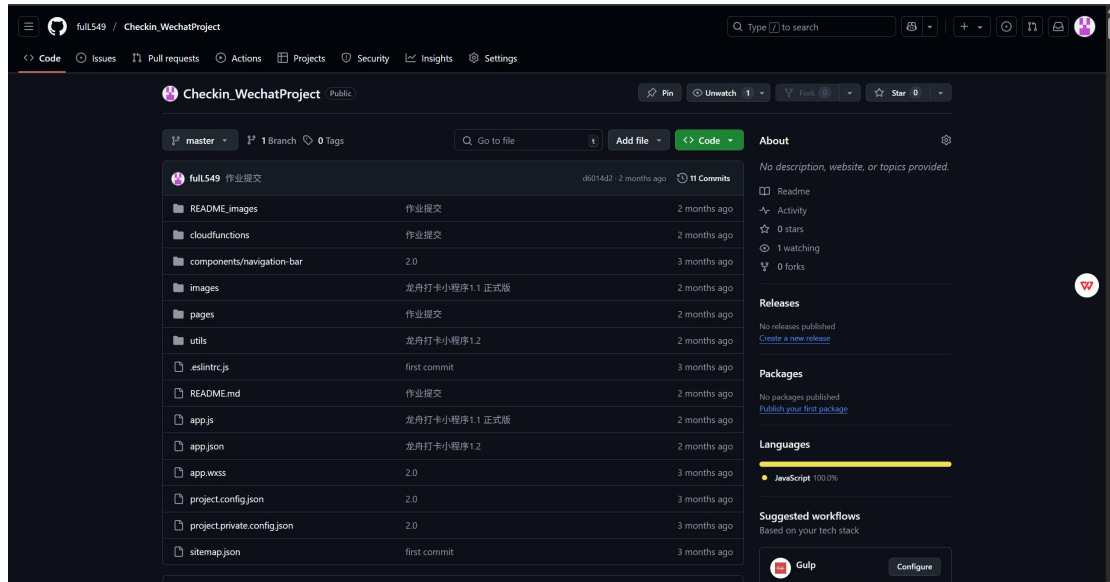
```
driver.find_element(By.ID, "icon_sousuo").click()
```



九、Git 代码管理

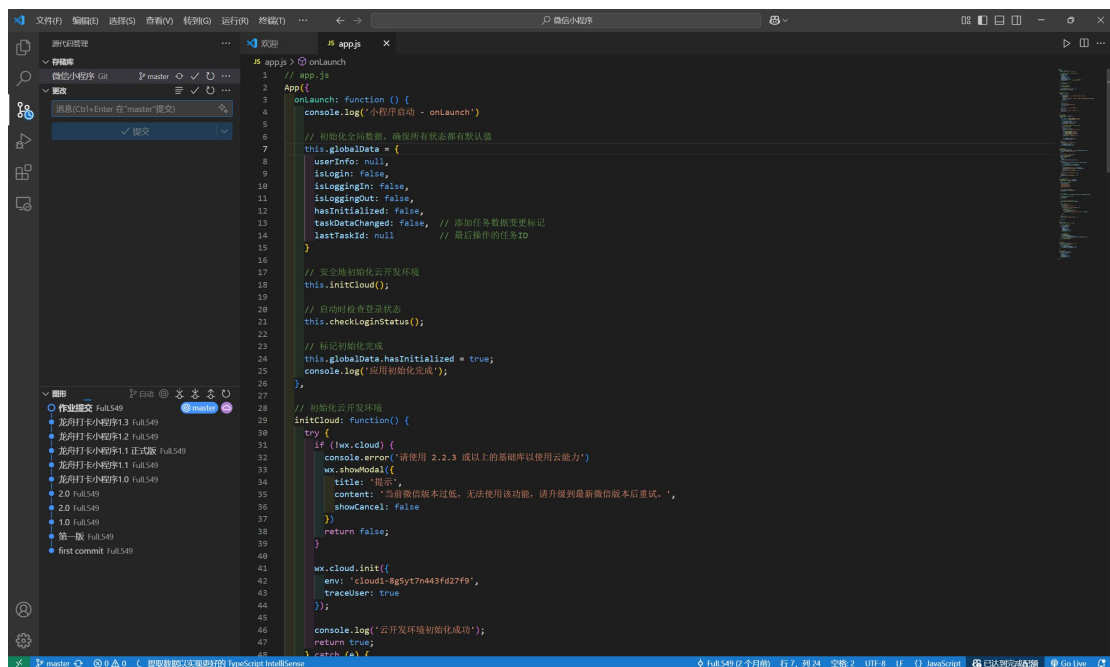
1. 使用 github 建立代码仓库进行项目管理

在 GitHub 平台创建远程仓库，实现代码版本控制。每次提交都具有唯一标识，便于回滚、比较、审查历史变更



2. 配合 vscode 进行代码管理

使用 VSCode 编辑器内置的 Git 工具，可快速完成 pull、commit、push 等操作，结合分支管理机制，实现多人协作与功能隔离开发



十、总结与思考

1. 通过本次开源软件综合实践，我深入掌握了几种主流的 CASE 工具在软件开发各环节的具体使用方法，显著提升了系统建模、测试自动化和项目管理的能力。整个实践过程从理论知识到实操应用，经历了如下几个方面的成长：

2. 从模型驱动开发到代码落地

通过使用 Enterprise Architect 工具进行类图、用例图、流程图的建模，我更加清晰地理解了面向对象系统的设计逻辑，也学会了将需求转化为可视化的建模语言。建模不仅有助于明确模块功能划分，还大大减少了后续实现中的歧义。

3. 实现了项目的敏捷式管理

使用 JIRA 工具让我熟悉了敏捷开发的流程与看板管理方式。任务分解、缺陷跟踪、版本迭代的流程大大提高了团队开发的效率。对于个人开发者而言，借助 JIRA 也能更有计划地推进开发任务，避免开发进度失控。

4. 自动化测试带来的质量保障

Appium 自动化测试脚本的编写极大提升了软件测试的效率。自动化测试不仅可以节省大量人力资源，而且能保证每次回归测试的完整性和一致性。对 Appium Inspector 和 uiautomator2 的使用也增强了我对 UI 自动化测试技术的理解。

5. 开源工具链构建开发闭环

从 EA 的需求建模、JIRA 的项目管理、Appium 的自动化测试到 Git 的代码管理，这些 CASE 工具构成了一套完整的开源开发工具链。在此过程中，我不再局限于单一开发技术，而是从更宏观的角度掌握了软件工程的流程管理和协同开发机制。

6. 对未来工作的启发

本次实验提升了我对工业级软件工程实践的理解，为将来进入企业从事软件开发、测试或项目管理工作打下了坚实基础。CASE 工具的使用能力也是企业用人单位重点考察的能力之一，通过本实验，我对它们的掌握更加系统和扎实。

十一、参考文献

1. https://blog.csdn.net/hhyyqq/article/details/141282529?ops_request_misc=%257B%2522request%255Fid%2522%253A%25229f5b73d6a091bf7856f8ace531132219%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=9f5b73d6a091bf7856f8ace531132219&biz_id=0&utm_medium=distribute.pc_chrome_plugin_search_result.none-task-blog-2~all~first_rank_ecpm_v1~rank_v3l_ecpm-4-141282529-null-null.nonecase&utm_term=win%E4%B8%8B%E8%BD%BDuwsig&spm=1018.2226.3001.4187

2. <https://www.piis.cn/zhishi/2453.html>

3. [https://blog.csdn.net/weixin_42292991/article/details/95976016/?ops_request_misc=&request_id=&biz_id=102&utm_term=chdir\(\):%20No%20such%20file%20or%20direc&utm_medium=distribute.pc_search_result.none-task-](https://blog.csdn.net/weixin_42292991/article/details/95976016/?ops_request_misc=&request_id=&biz_id=102&utm_term=chdir():%20No%20such%20file%20or%20direc&utm_medium=distribute.pc_search_result.none-task-)

blog-2~all~sobaiduweb~default-0-95976016.142^v102^pc_search_result_base3&spm=1018.2226.3001.4187

4. <https://github.com/3BPM/shshop>

5. https://blog.csdn.net/lmdsoft/article/details/82585126?ops_request_misc=&request_id=&biz_id=102&utm_term=sonar%20scanner%E6%89%AB%E6%8F%8F%E7%BB%93%E6%9E%9C%E5%88%86%E6%9E%90&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-9-82585126.nonecase&spm=1018.2226.3001.4187

6. https://blog.csdn.net/intelrain/article/details/95043570?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522027523848925426d4aadd82b6e40eala%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=027523848925426d4aadd82b6e40eala&biz_id=0&utm_medium=distribute.pc_chrome_plugin_search_result.none-task-blog-2~all~sobaiduweb~default-2-95043570-null-null.nonecase&utm_term=sonar%E6%89%AB%E6%8F%8F%E5%A4%B1%E8%B4%A5&spm=1018.2226.3001.4187

7. https://blog.csdn.net/Hatsune__Miku__/article/details/132866003

8. <https://blog.csdn.net/bfqsl988/article/details/80236753>

9. https://blog.csdn.net/weixin_48704094/article/details/132224975?ops_request_misc=&request_id=&biz_id=102&utm_term=sonarqube%E9%85%8D%E7%BD%AE%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8Fjre&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduweb~default-1-132224975.142^v102^pc_search_result_base3&spm=1018.2226.3001.4187

10. <https://so.csdn.net/so/search?q=sonarqube%E9%85%8D%E7%BD%AE%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8Fjre&urw=>

11. https://so.csdn.net/so/search?from=chrome_plugin&q=Sonarqube%E8%BF%9E%E6%8E%A5%E6%95%B0%E6%8D%AE%E5%BA%93%E5%90%8E%E6%8A%A5%E9%94%99

12. https://blog.csdn.net/qq_41631176/article/details/84952820?spm=1001.2101.3001.6661.1&utm_medium=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPaidSort-1-84952820-blog-90613736.235%5Ev43%5Epc_blog_bottom_relevance_base7&depth_1-utm_source=distribute.pc_relevant_t0.none-task-blog-2%7Edefault%7ECTRLIST%7EPaidSort-1-84952820-blog-90613736.235%5Ev43%5Epc_blog_bottom_relevance_base7&utm_relevant_index=1