北 京 邮 电 大 学

本科毕业设计（论文）中期进展情况检查表

学院	计算机学院（国家示范性软件学院）	专业	计算机科学与技术	班级	2018211303
学生姓名	马嘉骥	学号	2018211149	班内序号	17
指导教师姓名	方维	所在单位	计算机学院（国家示范性软件学院）	职称	副教授
设计（论文）题目	（中文）基于漏洞知识图谱的可视化系统的设计与实现				
	（英文）Design and Implementation of Visualization System Based on Vulnerability Knowledge Graph				

目前已完成任务

一、 任务概述：

1. 项目背景：

近年来，随着互联网产业迅速发展，互联网安全漏洞问题显著性也急剧增加。本课题针对上述问题，提出一种漏洞知识图谱可视化系统。基于知识图谱、图数据库等技术，对互联网漏洞数据包括受影响产品、可利用代码及补丁等信息进行收集与分析，形成知识结构；对多个数据源抽取的知识进行融合、抽取漏洞的实体及关系、建立漏洞的图数据库、形成漏洞知识图谱；基于漏洞知识图谱搭建基于 B/S 架构的可视化系统，提供易于使用的交互接口进行展示、知识筛选等操作。

本系统采用自动化的方式，实现对漏洞信息的持续收集与整理，极大节省了人力资源的消耗。结合抽取关键信息，对漏洞间关联性进行分析、构建漏洞知识图谱，将离散的漏洞信息转化为相互联系的图结构，为开发者提供项目依赖安全性参考、为互联网安全研究人员提供数据与服务支撑，促进构建更高效安全的互联网环境。

2. 开发任务及要求：

1). 设计并实现漏洞数据采集系统；

2). 设计并实现基于规则或机器学习方法的漏洞知识图谱构建系统；

3). 设计并实现基于图数据的漏洞知识图谱存储系统；

4). 设计并实现基于上述漏洞知识图谱的可视化系统；

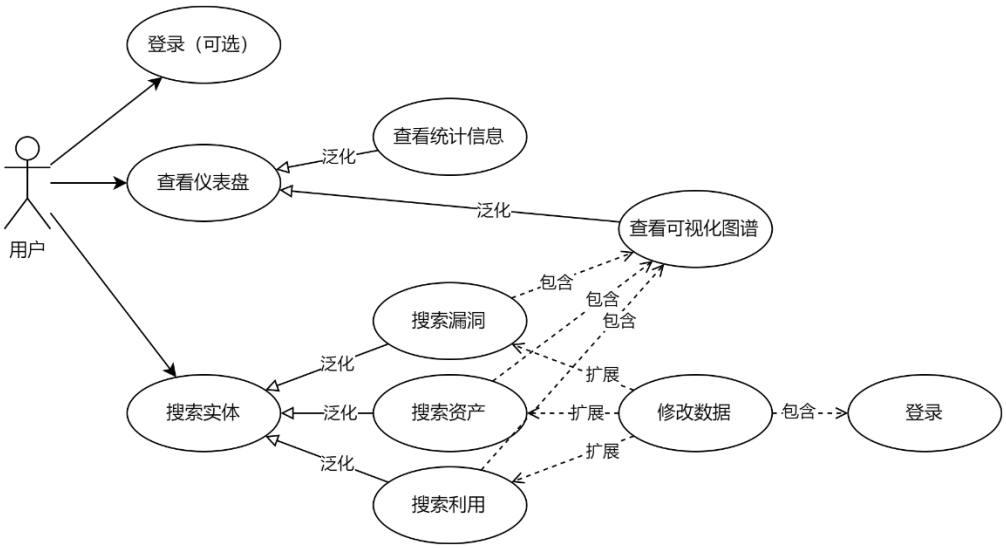
5). 对上述系统进行系统测试、排错、功能扩展、优化，编写文档记录。

二、 使用到的技术和工具：

数据获取	Scrapy、Pandas
图谱构建	Rule-based、Bi-LSTM-CRF
后端	Python-Flask
前端	Vue.js、Vuetify、D3.js
持久化	MongoDB、Neo4j

三、 已经完成的工作：

1. 需求分析：



图：系统用例图

系统输入：来自多种数据源的漏洞相关信息数据。

系统输出：以图形化显示的知识图谱信息。

a） 功能性需求：

爬虫：自动化数据采集及处理。

持久化：结构化与非结构化数据存储。

数据处理：知识图谱生成。

后端：数据库交互、绘图数据生成。

前端：图展示。

b） 非功能性需求：

1> 性能：

系统具备处理大量图数据的能力，并能应对不断增长的数据量。

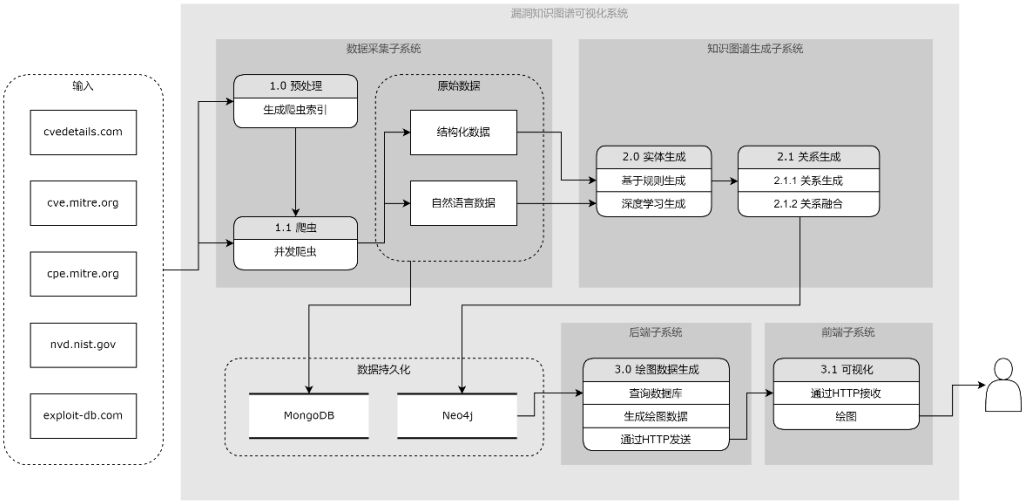
2> 兼容性：

可视化系统应在多种浏览器上运行。

2. 系统概要设计

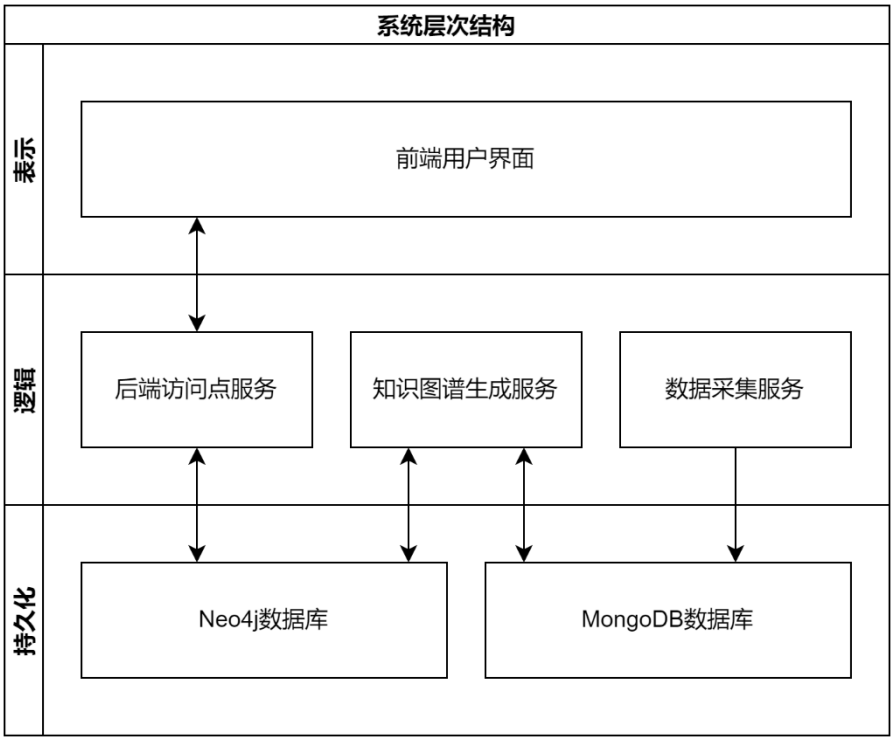
1） 系统总体设计

a） 系统数据流图



图：系统数据流图

b） 系统层次结构设计



图：系统层次结构设计

系统可以分为三层，表示层、服务层和持久化层。表示层包含前端用户界面。持久化层包含 Neo4j 与 MongoDB 两个数据库。逻辑层包含后端服务、知识图谱生成服务、数据采集服务。

数据采集服务通过生成索引、爬虫、数据清洗，向持久化层输入数据，知识图谱生成服务与持久化层进行交互，首先通过基于规则的实体构建与基于神经网络的命名实体识别，再经基于规则的关系生成、关系融合，将采集数据转化为知识图谱。后

端服务与持久化层、表示层进行交互，处理并传递图数据信息及用户控制信息。

c) 系统技术选型：

1> 前端用户界面：Vue.js+Vuetify（界面组件库）+Vuex（数据控制器）+Axios（异步网络通信）+D3.js（可视化）

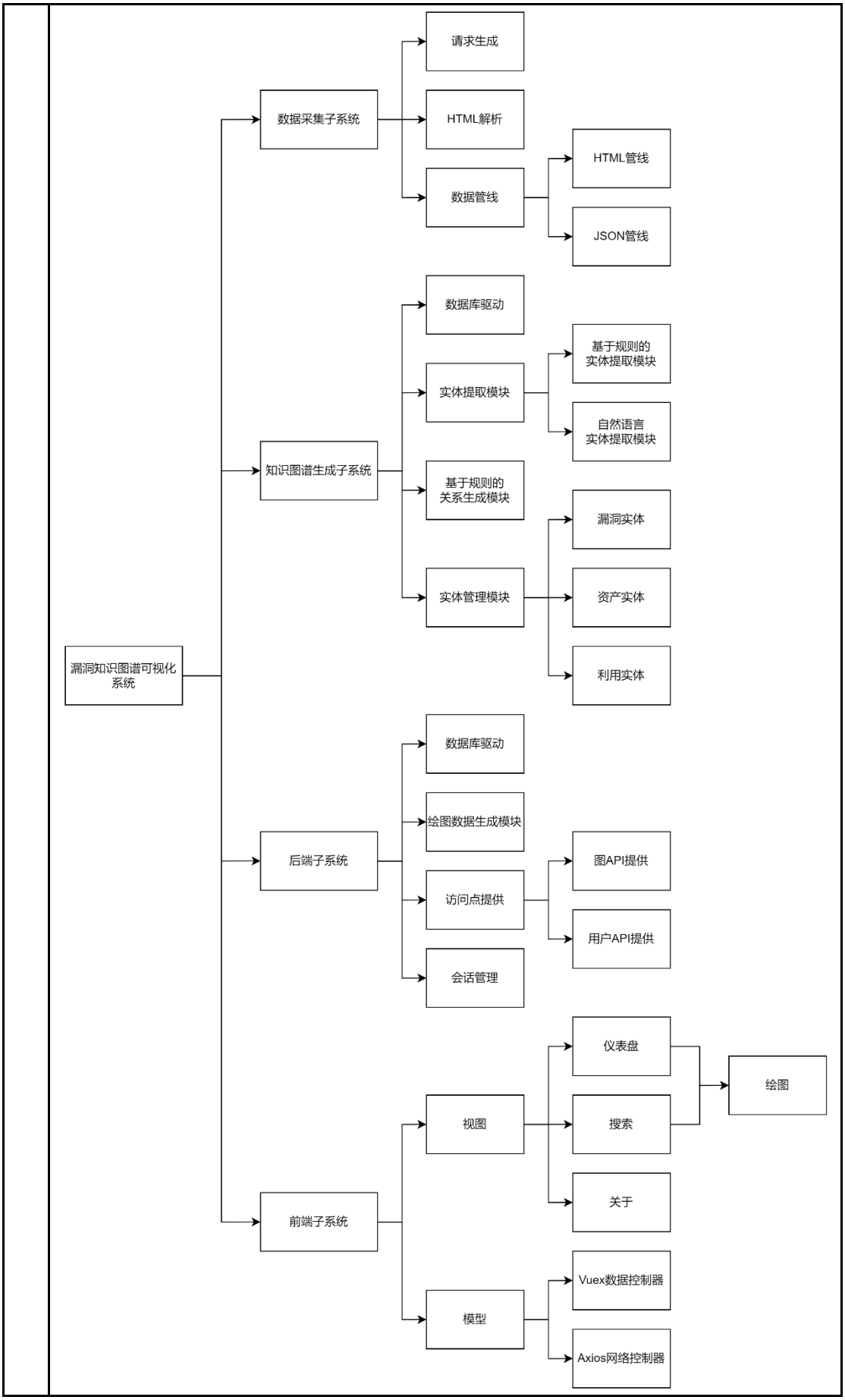
2> 后端 Web 服务：Python-Flask

3> 知识图谱生成服务：基于规则的实体生成与关系生成、Bi-LSTM-CRF

4> 数据采集服务：Scrapy、Pandas

5> 数据库：MongoDB、Neo4j

2） 系统功能模块设计


漏洞知识图谱可视化系统
数据采集子系统
请求生成
HTML解析
数据管线
HTML管线
JSON管线
知识图谱生成子系统
数据库驱动
实体提取模块
基于规则的实体提取模块
自然语言实体提取模块
基于规则的关系生成模块
实体管理模块
漏洞实体
资产实体
利用实体
后端子系统
数据库驱动
绘图数据生成模块
访问点提供
图API提供
用户API提供
会话管理
前端子系统
视图
仪表盘
搜索
关于
绘图
模型
Vuex数据控制器
Axios网络控制器

图：系统功能模块设计

系统可以分为 4 个主要子系统。

数据采集子系统：主要包含 Web 爬虫服务，包括请求生成器、HTML 解析器、数据管道三个模块，其中数据管道模块包含 HTML 管道和 JSON 管道两个子模块。请求生成器根据 csv 索引列表生成爬虫请求，HTML 解析器用于从不同页面中解析目标数据，数据管道将解析的数据格式化封装并存入数据库持久化，形成原始数据。

知识图谱生成子系统：包含数据库驱动、信息提取器、基于规则的关系生成器、实体管理四个模块，其中信息提取器包含基于规则的结构化数据提取器和基于神经网络的自然语言数据提取器，实体管理包括漏洞实体、资产实体、利用实体三种。数据库驱动实例化一个单例数据库对象，并向其他模块暴露对数据库的访问方法。信息提取器用于从原始数据中抽取实体，提取相关属性，实例化实体类。关系生成器用于生成实体对象之间的关系，并将其存入图数据库中。实体管理则管理图数据库中已有的实体和关系，在处理新的数据时防止重复生成实体和关系。

后端子系统：包含数据库驱动、绘图信息生成器、Web 服务访问点、会话管理四个模块。数据库驱动实例化一个单例数据库对象，并向其他模块暴露对数据库的访问方法，并将其挂载到请求处理的上下文对象中。绘图信息生成器用于将图数据库中存储的信息根据用户请求生成 D3.js 可以处理的格式化节点及关系数据。Web 服务访问点通过 Flask 提供一系列遵循 RESTful 格式的 API，主要包含图处理 API 及用户 API。图处理 API 提供图数据交互，满足可视化展示、搜索、信息修改等功能。用户 API 提供用户管理相关功能。会话管理用于维护前后端间通信的会话，为用户系统提供保证。

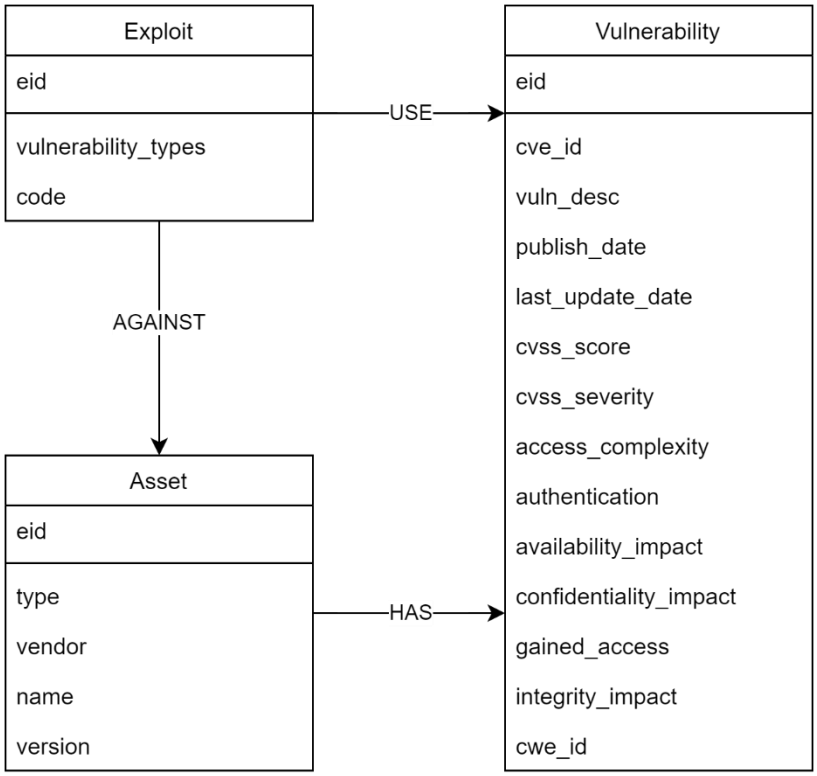
前端子系统：包含视图与模型两个模块。视图模块包含仪表盘、搜索、关于三个子模块，其中仪表盘子模块为用户提供知识图谱信息概览，包含列表数据展现及可视化数据统计。搜索模块支持用户在图谱中搜索感兴趣的内容，并将搜索结果以可视化图的形式展现。此外还提供数据修改等功能。关于子模块向用户展示本系统相关信息及操作指南。模型模块包含 Vuex 数据控制器与 Axios 网络控制器两个子模块。Vuex 数据控制器子模块是前端子系统中数据仓库，为视图模块响应式显示提供数据源，还包含在前端对数据进行操作处理功能。Axios 网络控制器子模块封装 httpClient 对象，提供向后端通信的方法。

3） 数据库设计

cve_item_html
cve_id
content

cve_item_json
cve_id
content cve_id vuln_desc publish_date last_update_date cvss_score cvss_severity access_complexity authentication availability_impact confidentiality_impact gained_access integrity_impact cwe_id affected_products[ ] reference[ ]

图：MongoDB 数据模式



图：Neo4j 数据模式

4） 系统接口设计（暂定）：

localhost:5000/api/graph/<op>	图数据接口
localhost:5000/api/user/<op>	用户接口
localhost:5000/<path>	前端路径

3. 系统详细设计

1） 开发环境配置：

操作系统：Windows 11 Pro，Ubuntu 20.04 LTS

数据采集子系统：Python 3.8，scrapy 2.6

图谱生成子系统：Python 3.8，py2neo 2021.2.3，pymongo 3.12，uuid 1.3

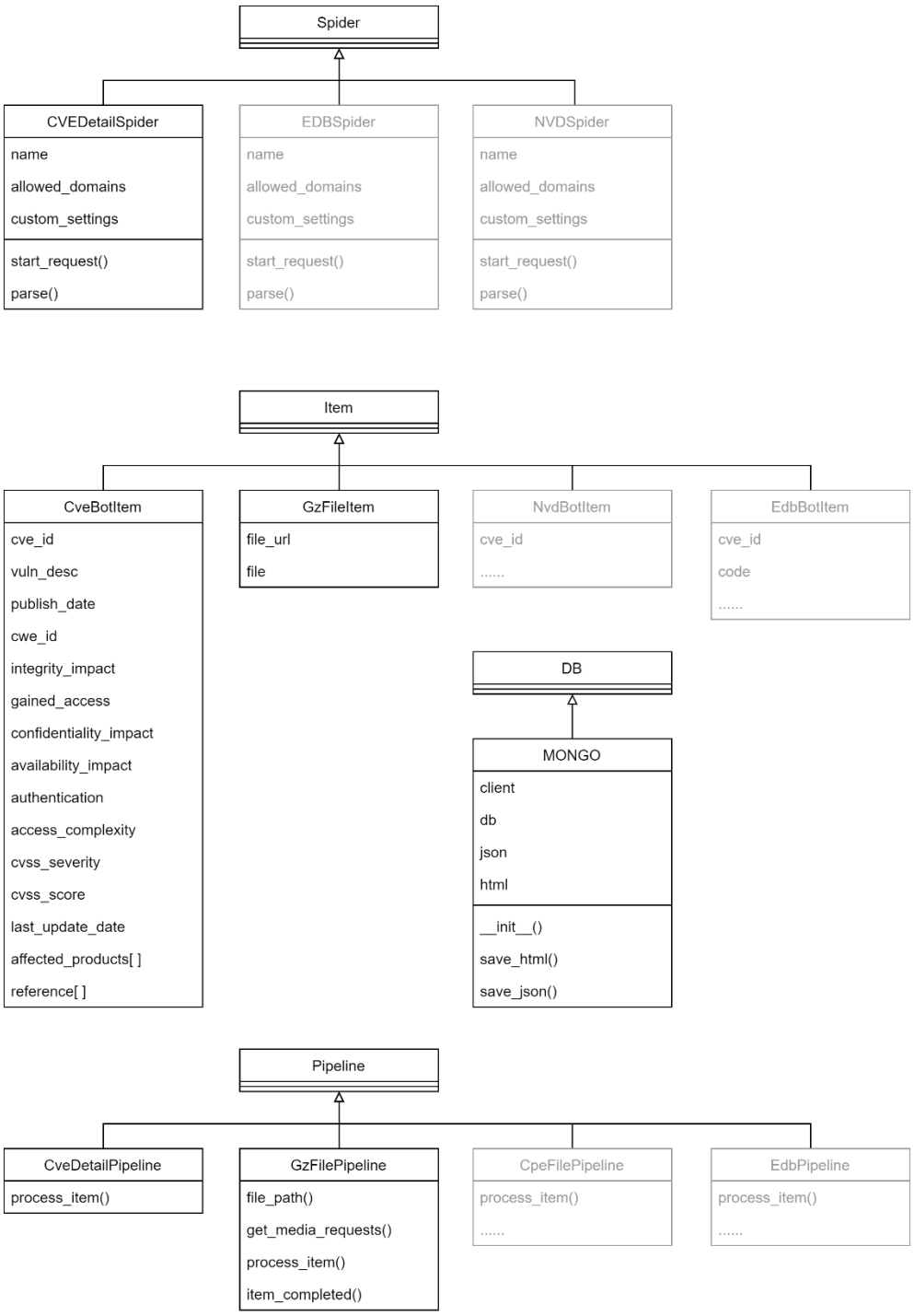
后端子系统：Python 3.8，flask 2.0，flask-cors 3.0，neo4j 4.4

前端子系统：JavaScript ES6，vue 2.6，d3 7.3，vue-axios 3.4，vuetify 2.6，vuex 3.4，force-graph 1.42

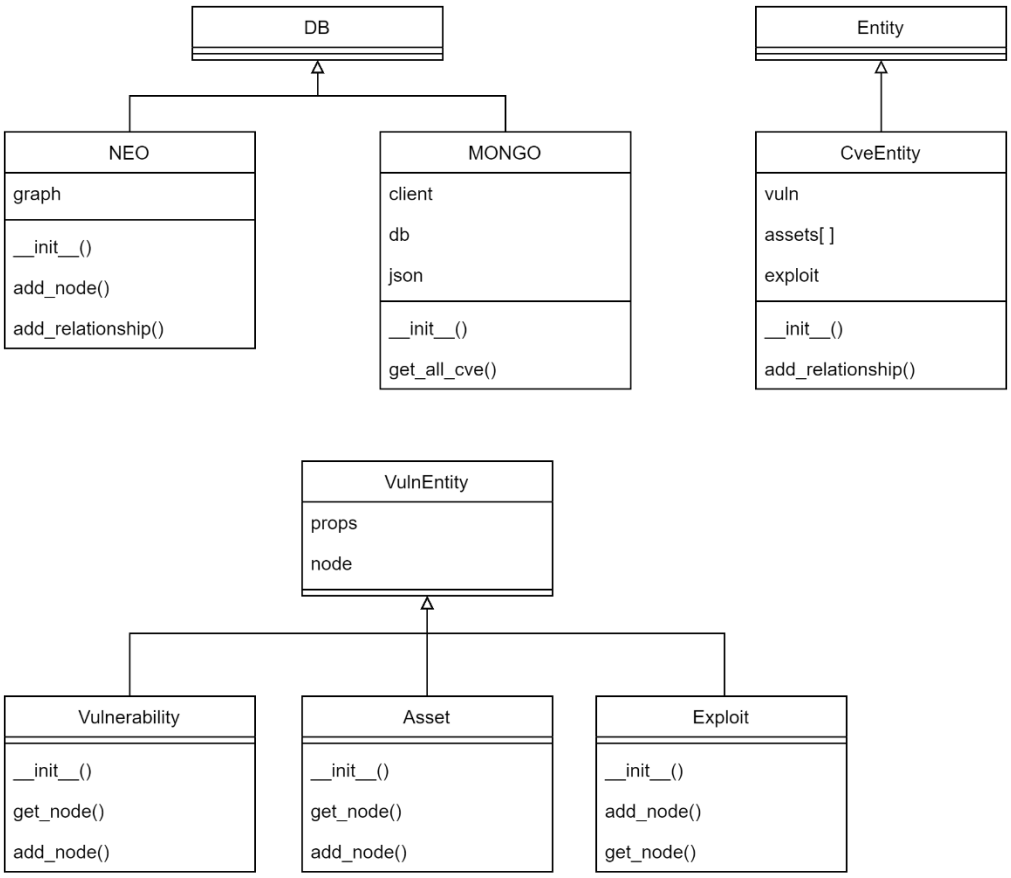
持久化子系统：MongoDB 5.0，Neo4j 4.4

2） 各核心功能模块的详细设计与实现

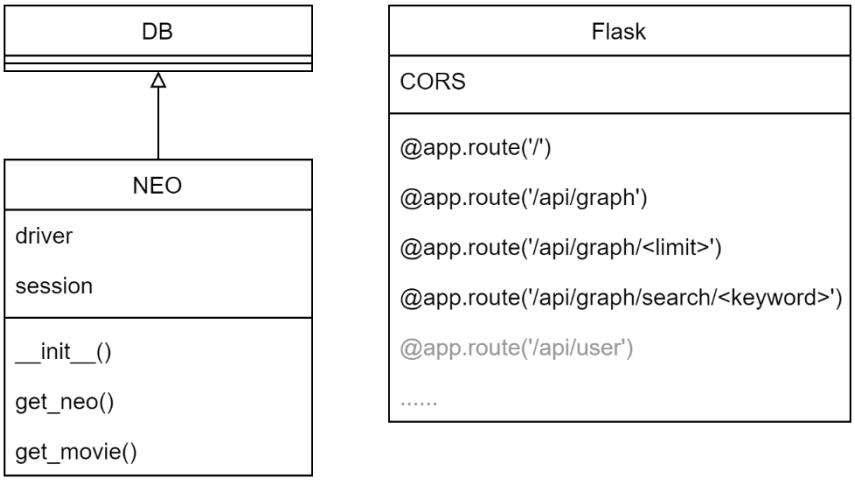
a) 类图



图：数据采集服务类图

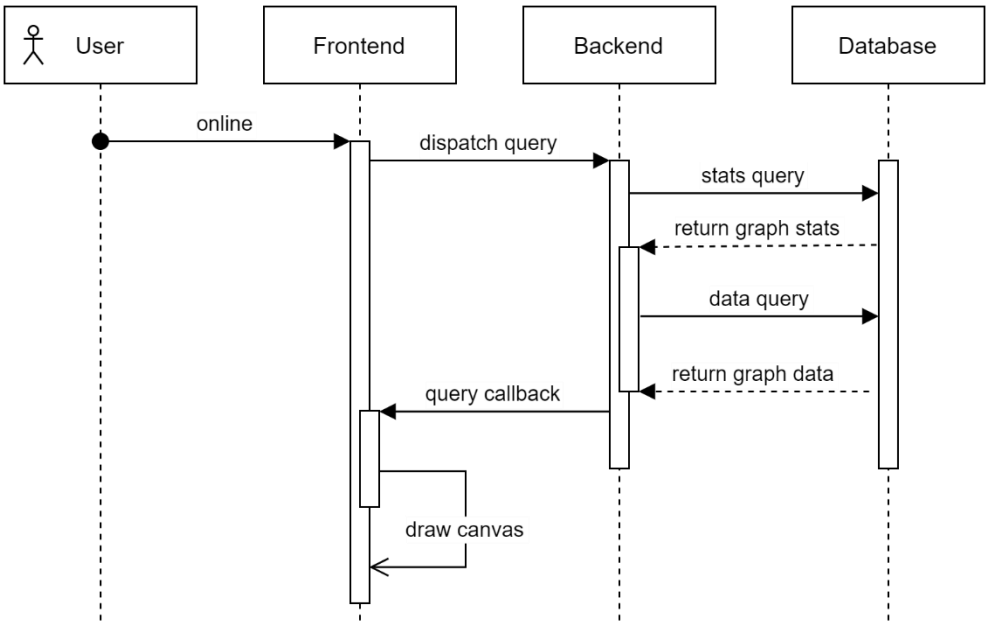


图：知识图谱生成服务类图



图：后端服务类图

b） 功能时序图



图：用户请求时序图

四、 已经完成的工作测试：

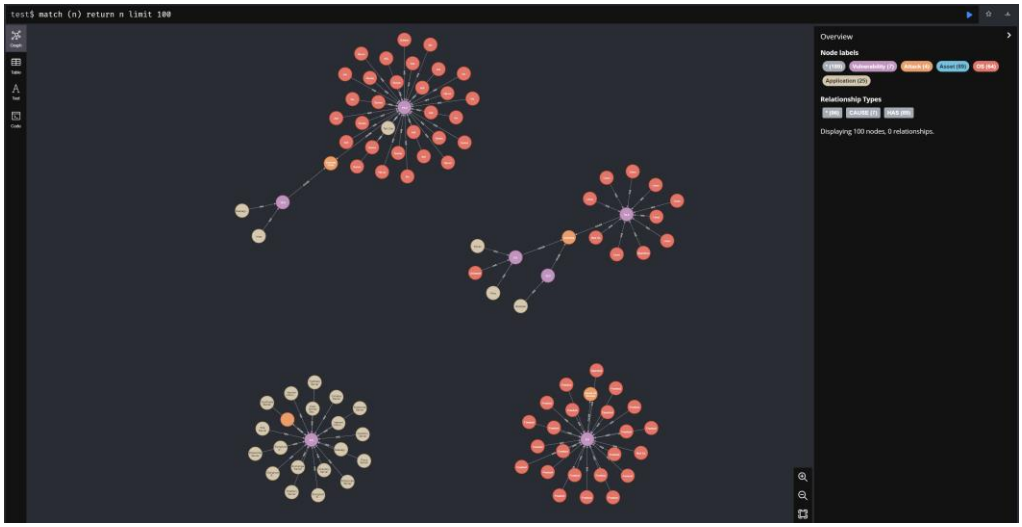
1. 数据采集模块获取数据：

   通过 Scrapy 爬虫 cvedetails.com、cpe.mitre.org、cve.mitre.org，获得约 16 万条 cve 信息及其 html 页面，清洗整理获得超 400 万个 cve 属性。


```
_id: ObjectId("620d45286e07ac4435f26c8f")
cve_id: "CVE-1999-0001"
content: Object
    cve_id: "CVE-1999-0001"
    vuln_desc: "ip_input.c in BSD-derived TCP/IP implementations allows remote attacke..."
    publish_date: "1999-12-30"
    last_update_date: "2010-12-16"
    cvss_score: 5
    cvss_severity: "Medium"
    confidentiality_impact: Object
        text: "None"
        desc: "There is no impact to the confidentiality of the system."
    integrity_impact: Object
        text: "None"
        desc: "There is no impact to the integrity of the system."
    availability_impact: Object
        text: "Partial"
        desc: "There is reduced performance or interruptions in resource availability..."
    access_complexity: Object
        text: "Low"
        desc: "Specialized access conditions or extenuating circumstances do not exis..."
    authentication: Object
        text: "Not required"
        desc: "(Authentication is not required to exploit the vulnerability.)"
    gained_access: "None"
    vulnerability_types: "Denial Of Service"
    cwe_id: 20
    affected_products: Array
    references: Array
        0: "http://www.osvdb.org/5707"
        1: "https://www.openbsd.org/errata23.html#tcpfix"
```

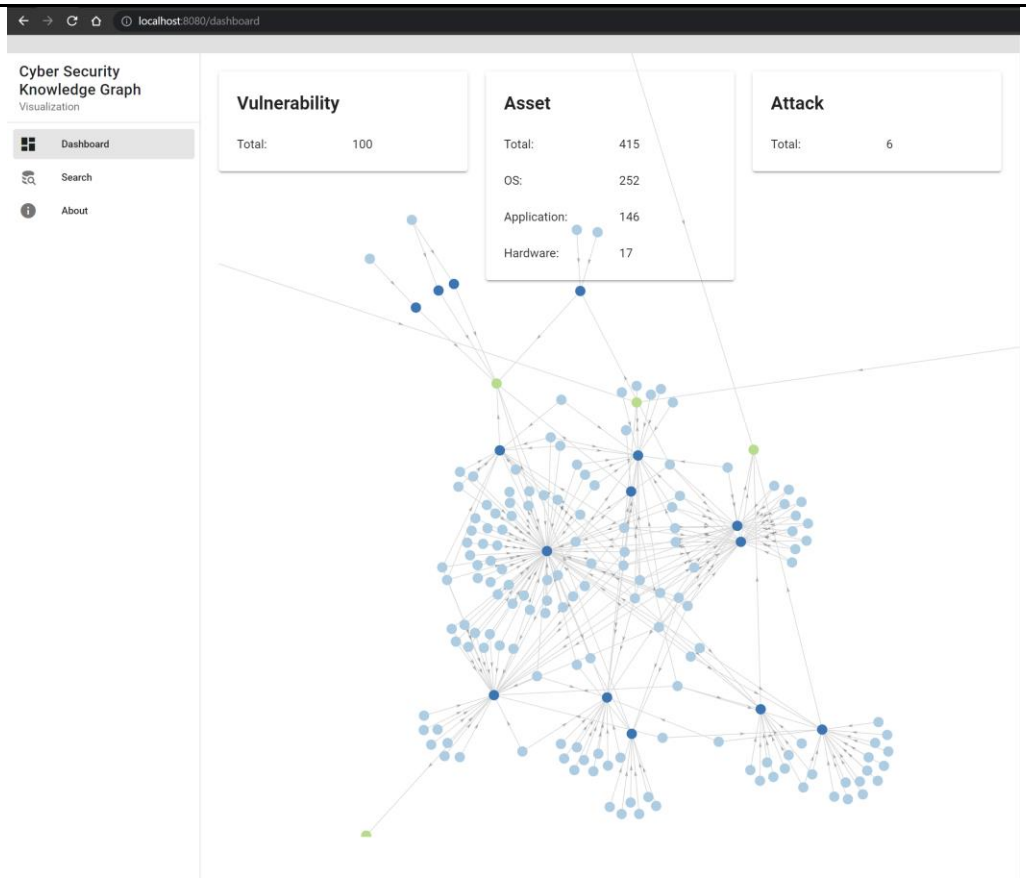
2. 知识图谱生成测试（Neo4j Browser）：

通过 MongoDB 的 cve 信息，使用前 100 条目，编写基于规则的实体与关系生成模块，进行生成漏洞、资产、利用实体测试。



3. 前端可视化展示测试：

使用 Vue.js 与 Flask 搭建前后端，Vuetify 作为前端组件框架，d3.js 作为绘图库，实现前端可视化展示与根据 cve-id 字符串进行简单搜索功能。



是否符合任务书要求进度 是

<table>
<tr><td rowspan="2">尚需完成的任务</td><td colspan="2">数据采集子系统：
1. 根据数据库索引信息比对进行增量爬虫
2. 异构数据源采集

知识图谱生成子系统：
1. 完善网络安全实体与属性的设计，完善提取规则
2. 实现基于 Bi-LSTM-CRF 模型的深度学习命名实体识别模块
3. 完善实体融合模块的设计与实现

后端系统：
1. 实现用户认证、连接管理系统
2. 拓展图数据 API 功能，延伸可视化系统能力
3. Neo4j 数据库访问性能优化

前端系统：
1. 完善 UI 设计
2. 完善可视化展示功能、可视化数据修改功能
3. 增加用户系统</td></tr>
<tr><td colspan="2">能否按期完成设计（论文） 是</td></tr>
<tr><td>存在问题</td><td>存在问题</td><td>1. Flask 连接 Neo4j 驱动系统启动初次查询响应慢。
2. 图数据库结点量为百万级，需要性能优化。</td></tr>
</table>

<table>
<tr><td>和解决办法</td><td>拟采取的办法</td><td colspan="3">1. 经查证此为 Neo4j 在 Windows 平台上的 bug，后期服务器部署将采用 Linux 系统。
2. 在数据库中使用多种索引加快查询速度；在后端限制单次查询最大结点数量；在前端限制同时展示最大结点数量，为用户提供更细分的查询逻辑，使用 LocalStorage 缓存数据。</td></tr>
<tr><td colspan="2">指导教师签字</td><td>方雅</td><td>日期</td><td>2022 年 4 月 13 日</td></tr>
<tr><td colspan="2">检查小组评分及意见</td><td colspan="3">评分：23 （总分：25 ）
通过中期检查，按反馈意见修改。

组长签字：方雅 2022年 4 月15日</td></tr>
</table>

注：可根据长度加页。