

# 需求规格说明文档

车一晗 181250009

黄婉红 181840096

纳思彧 181250107

王博 181250133

2021 年 3 月 18 日

## 摘要

本文档为 Heap 小组在 2021 年春学期《软件工程与计算三》课程作业迭代一中为项目所撰写的需求规格说明文档。

## 目录

1 引言.....	4
1.1 编写目的.....	4
1.2 对象与范围.....	4
1.3 参考文献.....	4
1.4 名词与术语.....	4
2 总体描述.....	5
2.1 项目前景.....	5
2.1.1 背景与机遇.....	5
2.1.2 业务需求.....	5
2.2 项目功能.....	5
2.3 用户特征.....	5
2.4 约束.....	6
2.5 假设和依赖.....	6
3 详细需求描述.....	6
3.1 对外接口需求.....	6
3.1.1 用户界面.....	6
3.1.2 通信接口.....	7
3.2 功能需求.....	7
3.2.1 用户新增节点、关系.....	7
3.2.1.1 特征描述.....	7
3.2.1.2 刺激/响应序列.....	8
3.2.1.3 相关功能需求.....	8
3.2.2 用户编辑节点、关系.....	8
3.2.2.1 特征描述.....	8
3.2.2.2 刺激/响应序列.....	8
3.2.2.3 相关功能需求.....	9
3.2.3 用户删除节点、关系.....	9
3.2.3.1 特征描述.....	9
3.2.3.2 刺激/响应序列.....	9

3.2.3.3 相关功能需求.....	10
3.2.4 用户使用工作域.....	10
3.2.4.1 特征描述.....	10
3.2.4.2 刺激/响应序列.....	10
3.2.4.3 相关功能需求.....	11
3.2.5 导入和导出.....	11
3.2.5.1 特征描述.....	11
3.2.5.2 刺激/响应序列.....	11
3.2.5.3 相关功能需求.....	12
3.3 非功能需求.....	12
3.3.1 安全性.....	12
3.3.2 性能.....	12
3.3.3 可靠性.....	12
3.3.4 易用性.....	12
3.3.5 可移植性.....	13
3.3.6 约束.....	13
3.4 数据需求.....	13
3.4.1 数据定义.....	13
3.4.2 默认数据.....	13
3.4.3 数据格式要求.....	13
3.4.4 其他需求.....	14

# 1 引言

## 1.1 编写目的

本文档描述了 COIN 知识关系图谱系统的功能需求与非功能性需求。开发小组的软件系统实现与验证工作都以此文档为依据。

注：除特殊说明之外，本文档所包含的需求都是高优先级需求。

## 1.2 对象与范围

本文档的读者是 Heap 团队内部的开发和管理人员。

COIN 知识关系图谱旨在展示在线编辑可视化的“节点-关系-节点”的关系图，用户能够自主通过在线编辑或导入文件格式的方式编辑知识图谱，同时能够将生成的知识图谱导出为图片或其他文字格式。

## 1.3 参考文献

《项目计划文档》，Heap;

《软件架构设计文档》，Heap;

《项目启动文档》，Heap

《软件工程与计算（卷二）》 骆斌 丁二玉 刘钦

《软件工程与计算（卷三）》 骆斌 刘嘉 张瑾玉 黄蕾

## 1.4 名词与术语

COIN: COConstructing and vlsualizing kNowledge graph 知识图谱可视化系统

## 2 总体描述

### 2.1 项目前景

#### 2.1.1 背景与机遇

在众多知识表示方式中，知识图谱作为一种语义网络拥有极强的表达能力和建模灵活性：知识图谱是一种语义表示，可以对现实世界中的实体、概念、属性以及它们之间的关系进行建模；其次，知识图谱是其衍生技术的数据交换标准，其本身是一种数据建模的“协议”，相关技术涵盖知识抽取、知识集成、知识管理和知识应用等各个环节。通过构建知识图谱，可以极大化的辅助相关知识的理解。

#### 2.1.2 业务需求

BR1：系统正式上线半个月后，至少获取 1000 次有效访问。

### 2.2 项目功能

SF1：对知识图谱进行在线编辑（增、删、改）

SF2：能够导入特定格式文件生成图谱

SF3：能够将图谱导出图片

### 2.3 用户特征

用户	系统的访问者，希望通过导入或在线编辑的方式使用知识图谱，考虑到用户计算机水平层次不齐，因此界面应该尽可能简单，做到对用户易用。
----	---

## 2.4 约束

CON1: 系统使用 Web 界面。

CON2: 系统使用迭代式开发。

CON3: 在开发中，开发者要提交计划、软件需求规格说明文档、设计描述文档和测试报告

## 2.5 假设和依赖

AE1: 用户至少运行在 100M 带宽的网络环境中，网络通畅。

AE2: 不会有高并发场景。

# 3 详细需求描述

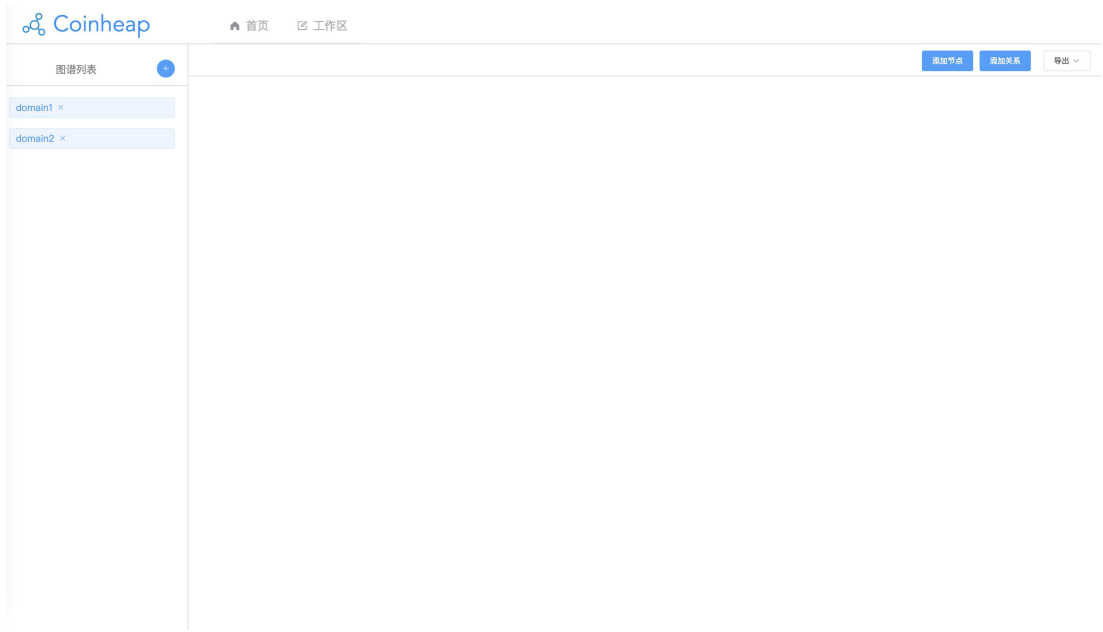
## 3.1 对外接口需求

### 3.1.1 用户界面

首页：导入文件、创建新图谱



工作区：创建节点、建立关系、导出等



### 3.1.2 通信接口

CI：客户端与服务器使用 HTTP 的方式进行通信。

## 3.2 功能需求

### 3.2.1 用户新增节点、关系

#### 3.2.1.1 特征描述

用户选择添加节点，输入节点相关信息，或单击对已有的节点创建关系连接

优先级：高

### 3.2.1.2 刺激/响应序列

刺激：用户选择新增节点

响应：系统返回节点信息一览，请求用户填写

刺激：用户输入节点信息

响应：系统保存节点信息

刺激：用户选择两个节点创建连接

响应：系统为选择节点创建关系

### 3.2.1.3 相关功能需求

名称	描述
EntityController.createNode	通过输入节点信息来新增节点
RelationshipController.createLink	通过输入节点起始终止 id 来新增关系

## 3.2.2 用户编辑节点、关系

### 3.2.2.1 特征描述

用户点击节点，编辑节点相关信息，或单击连接，编辑连接信息

优先级：高

### 3.2.2.2 刺激/响应序列

刺激：用户点击节点或关系

响应：系统弹出编辑按钮

刺激：用户点击编辑



响应：系统请求对应信息键入

刺激：用户确认信息键入

响应：系统为选择的节点或关系更新信息

3.2.2.3 相关功能需求

名称	描述
EntityController.updateNode	通过编辑节点信息来编辑节点
RelationshipController.updateLink	通过修改关系信息来编辑关系

3.2.3 用户删除节点、关系

3.2.3.1 特征描述

用户点击节点或关系，删除节点或关系，删除节点时，会一并删除和该节点有关联的关系

优先级：高

3.2.3.2 刺激/响应序列

刺激：用户点击节点或关系

响应：系统弹出删除按钮

刺激：用户点击删除

响应：系统删除该节点或关系

3.2.3.3 相关功能需求

名称	描述
EntityController.deleteNode	通过输入节点 id 删除节点及其附属关系
RelationshipController.deleteLink	通过输入关系 id 删除关系

3.2.4 用户使用工作域

3.2.4.1 特征描述

用户编辑知识图谱时均在工作域内进行，并可对工作域进行编辑

优先级：高

3.2.4.2 刺激/响应序列

- 刺激：用户新增工作域

响应：系统为用户新增工作域
- 刺激：用户编辑工作域

响应：系统弹出信息编辑，请求修改
- 刺激：用户删除工作域

响应：系统删除该域的所有节点和关系
- 刺激：用户使用工作域

响应：系统在界面上展示工作域的节点、关系信息

3.2.4.3 相关功能需求

名称	描述
EntityController.getNodesByDomainId	获取域内的所有节点信息
RelationshipController.getLinkByDomainId	获取域内知识图谱关系
DomainController.createDomain	新建工作域
DomainController.updateDomain	编辑工作域信息
DomainController.deleteDomain	删除域及域内的关系节点
DomainController.selectDomainById	根据 id 查找域
DomainController.selectAllDomain	返回所有域

3.2.5 导入和导出

3.2.5.1 特征描述

用户可通过导入规范格式文件自动生成图谱，或将已有图谱导出为指定格式文件

优先级：高

3.2.5.2 刺激/响应序列

刺激：用户选择导入 csv 文件

响应：系统自动创建工作域，并按照文件内容填充关系

刺激：用户选择导出 xml 文件

响应：系统将图谱导出为同目录下的 xml 文件

### 3.2.5.3 相关功能需求

名称	描述
FileController.getCsvFile	导入 csv 文件生成知识图谱
FileController.exportGraphXML	导出已有图谱为 xml 格式文件

## 3.3 非功能需求

### 3.3.1 安全性

Safety1: 系统要每隔 10s 对与其建立链接的网络环境进行探测并反馈报告

Safety2: 系统可以在 0.1s 内识别并拒绝非法访问

### 3.3.2 性能

Performance1: 系统的吞吐量可达到 50GB/s

Performance2: 系统的响应时间不超过 10ms

Performance3: 系统的超时错误率不超过 0.1%

### 3.3.3 可靠性

Reliability1: 如果在同客户交互时, 网络出现故障, 系统不能出现故障

Reliability2: 数据库的数据实时备份, 在丢失或破损后可自动恢复

### 3.3.4 易用性

Usability1: 无需用户使用手册或专业培训即可直接使用系统

Usability2: 查询任何一条数据不会跳转超过 3 个页面, 鼠标点击不会超过 5 次

### 3.3.5 可移植性

Portability1：系统或部件可以在 4 个人月内从 windows10 系统迁移至 centos7 操作系统

Portability2：数据库可在 5 个人月内从 MySQL 数据库迁移至 Oracle 数据库

### 3.3.6 约束

C1：系统采用分层模型开发

C2：系统前端使用 Vue 框架和 JavaScript 语言进行开发

C3：系统后端使用 Java 语言进行开发

## 3.4 数据需求

### 3.4.1 数据定义

DR1：系统需要存储用户创建的节点的详细关系信息

DR2：系统需要存储用户三个月内的活动记录

DR3：系统删除的所有数据仍需继续存储三个月的时间，以保证历史数据显示的正确性

### 3.4.2 默认数据

系统从其他数据源导入新数据中的某一项为空时，默认为 null

### 3.4.3 数据格式要求

Format1：日期格式统一精确到年份-月-日 yyyy-mm-dd

Format2：节点、关系的识别 id 统一为长整型数字

Format3：颜色格式为类似#FFFFFF 的字符串格式

Format4: 节点、关系的类型格式为-1~10 内的整数

Format5: csv 文件的格式为三列的 “起始节点名,终止节点名,关系名”

#### 3.4.4 其他需求

系统安装部署时，需同时导入一批数据，保证系统安装后即可供用户使用。同时该系统投入使用时，需对运维人员进行 1 个月的专业培训