# Стаття 2

# Ключові слова :

розробка онтологій, ontop, IDE, joint development , Protege, Protege Online

# Постановка проблеми

основна думка - виписати із усіх статтей якусь інфу про те, де використовуються онтології.   
виписати які треба зробити дії для того, аби створити інстанс в протеже і інших редакторах. і які редактори дозволяють редагувати онтологію онлайн.

Онтології викоистовуються для оптимізації процесів, іще чогось і іще чогось. У інтернет магазинах можуть викорстовуватися для опису продуктів для зручного їх пошуку. У вікі онтології використовуються для зберігання інформації про явища та його звязку з іншими явищами.

Онтологія складається із мета моделі , в яку входять класи та проперті (дата проперті. що визначають значення та обєктні проперті, що визначають звязки між обєктами) та об’єкти, що є інстансами класів із уже заповненими пропертями.

У випадку, коли для роботи треба заповнити онтологію інформаціюєю, тобто створити обєкти уже попередньо створених класів, робота із онтологією за допомогою таких інструментів як Protege та Protege Online може не бути оптимізованою та забирати більше часу аніж треба.

Саме в цьому і є проблема, яка дослдіжується в даній статті. Проблема оптимізації внесення інформації в попередньо розроблену онтологію для подальшого аналізу та пошуку знань.

# Аналіз останніх досліджень

На сьогодні протеже дозволяє використовувати плагіни. Серед них і OnTop - плагін, що дозволяє звязувати реляційну базу даних із онтлогією.

У разі, ж коли треба не лише звязати дані із онтологією, а перенести їх, є такі рішення (чи немає).

таким чином не можна ніяк занести дані в онтологію якимось шляхом окрім як вручну.

# постановка завдання

Дослідити альтернативні методи внесення інформації в онтологію.

підрахувати кількість кліків, потрібних для створення обєкту в протеже і в моїй системі.

# Результати дослідження

## овералл огляд і фічі

основна думка - описати фічі протеже і онлайн протеже і мого додатку   
у мене їх небагато, але вони дуже корисні.

у десктоп протеже є отакі фічі .....

у веб версії є отакі ....

але у мене немає отаких але є отакі....

Була зроблена спроба створити реляційне представлення онтології та створити систему, що дозволяє працювати із даними в онтології у вигляді таблиць. Також програмний продукт використовує ролі користувачів для підвищення безпеки доступу до даних.

Цими ролями є :

* r\_user (користувач із правом перегляду інформації);
* rw\_user (користувач із правом модифікації інформації);
* admin (адміністратор системи).

Користувачі, що мають лише роль “r\_user” можуть лише переглядати дані у вигляді таблиць і робити запити на мові SPARQL, що не модифікують дані.

Користувачі, що мають роль “rw\_user” можуть , окрім дій, що є доступними для ролі “r\_user”, також вводити і редагувати дані (створювати, редагувати і видаляти інстанси із онтологій).

Користувач із роллю “admin” має усі права, що й користувачі із попередніми ролями, плюс додаткові : завантажувати, вивантажувати та видаляти із системи файли із онтологіями (файли у форматі “.owl”).

Таким чнном ймовірність помилки, викликаної людським фактором знижається і підвищується безпека даних.

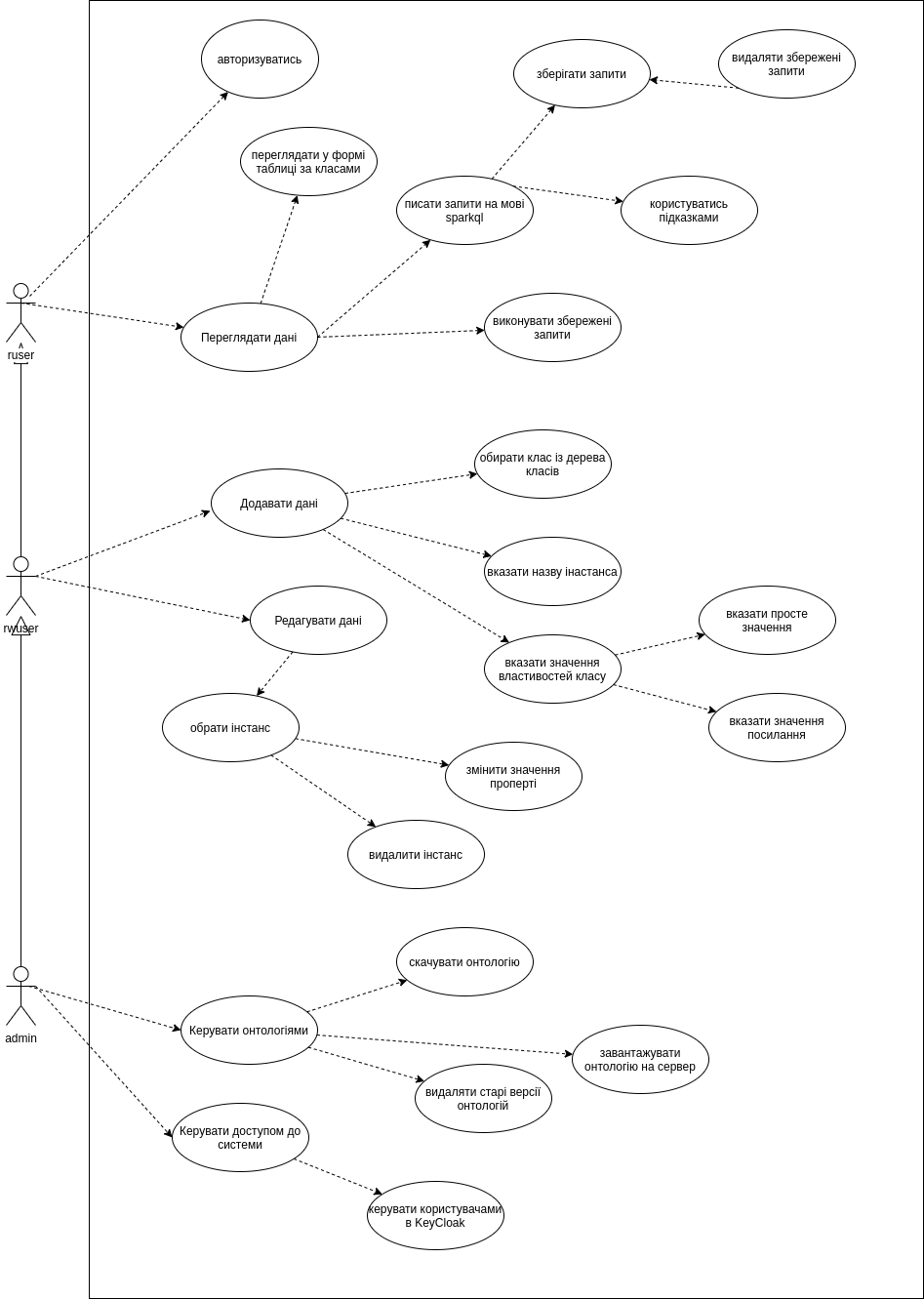


Рисунок 4.3 - Use-Cases розробленої системи

На основі розробленої діаграми прецедентів, був спроектований інтерфейс. Таким чином веб-додаток складається із декількох сторінок:

* головна;
* сторінка із деревом класів (“Class Tree”);
* сторінка для розробки запитів (“Query Page”);
* сторінка для менеджменту онтологій (“Admin Page”).

Сторінка із деревом класів дозволяє переглядати класи, що є наявними у онтології, а також інстанси із онтології за класами або усі підряд. Ця сторінка також дозволяє створювати, редагувати та видаляти інстанси. Цю сторінку користувачі із різними ролями бачать по-різному. Так, наприклад, користувач, у якого є лише роль “r\_user”, не зможе зробити ніяких дій, що змінять дані в онтології. Він зможе лише переглядати дані.

Сторінка для розробки запитів складається із 2 панелей: для введення запиту і для перегляду відповіді. Користувач, що має тільки роль “r\_user” зможе виконувати лише запити, що повертають дані, але не модифікують їх.

Сторінка для менеджменту онтологій доступна лише для користувачів із роллю “admin”. Тут є список усіх онтологій, що присутні у системі. Користувач може їх вивантажувати на робочу машину, редагувати і завантажувати назад. Також адміністратор може видаляти із системи старі онтології або версії онтологій.

Перш ніж роботу із системою зможуть почати користувачі із ролями “r\_user” та “rw\_user” - адміністратор має імпортувати у систему онтології, із якими інші користувачі зможуть працювати (вводити дані та переглядати їх).

Оскільки система задля простоти використання була реалізована лише із основними функціями для редагування інстансів у онтології - у систему має бути імпортована онтологія із готовими класами і їх властивостями.

Це може зробити лише адміністратор на сторінці із назвою “Admin page” (рисунок 5.13).

На цій сторінці є таблиця із усіма онтологіями у системі і їх версіями, а також форма для завантаження нових онтологій або версій. Кожен файл, що є у таблиці, можна скачати на локальну машину, відредагувати (наприклад, додати класи, використовуючи професійне середовище розробки - Protege) і завантажити назад. Таким чином, у системі реалізована функція контролю версій. Непотрібні онтології або старі версії онтологій можуть бути видалені із системи.

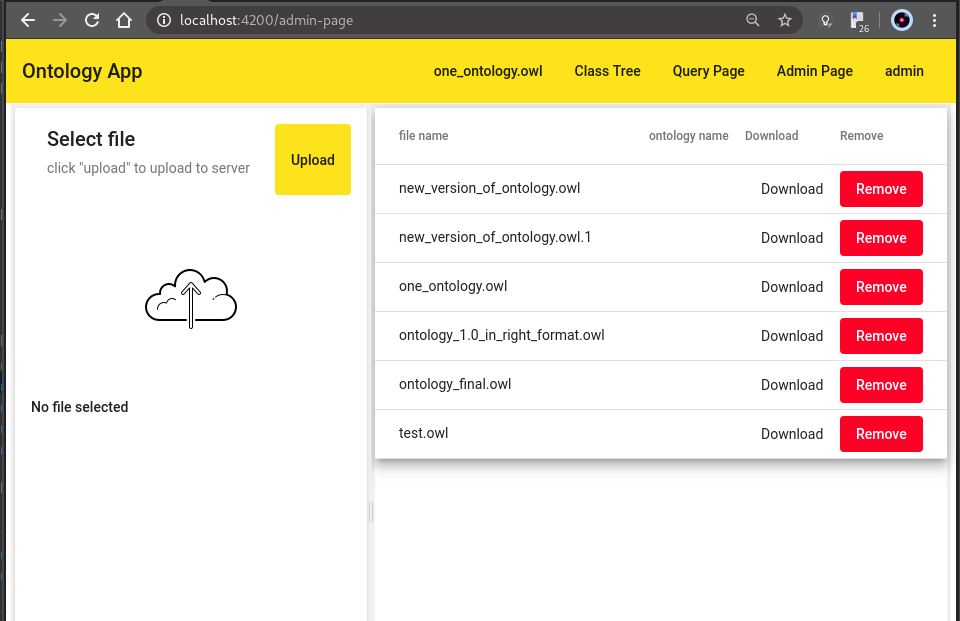


Рисунок 5.13 - Сторінка менеджменту онтологій

основна думка, хоч і різонери не підтримуються, можна перевірити наприклад правильність даних кверею. може бути користним для аналітиків, що використовують цю мову і не повинні нічого редагувати

Користувач системи із роллю, що передбачає лише перегляд даних після авторизації може переглядати дані у формі таблиць, користуючись сторінкою із деревом класів. Також може користуватись сторінкою для виконання запитів на мові SPARQL.

Також система дозволяє виконувати прості запити на мові запитів SPARQL. Підтримки різонерів іще немає.

Тому нажаль пошук знань у даній системі провести неможливо. Але якщо завдання передбачає створення великої кількості обєктів у онтології (наприклад занесення каталогу товарів електронного магазину) таку роботу можна простіше виконати паралельно в даній системі.

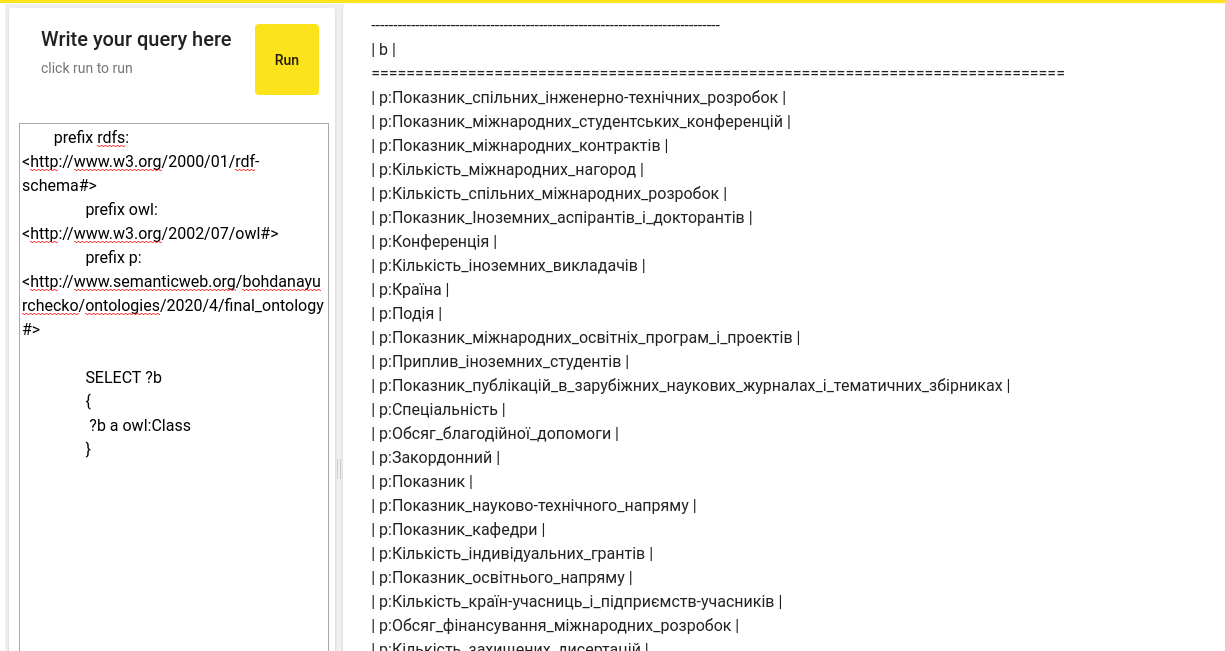


Рисунок 5.31 - Результати виконання запиту

## Опис сторінки із деревом класів

головна думка - у мене створення інстансу проходить на одній формі. форма будується динамічно. на основі інформації про клас.

Після того, як адміністратор системи завантажив у систему хоча б одну онтологію із класами і їх властивостями, користувачі із роллю “rw\_user” можуть почати свою роботу - вносити інформацію у систему.

У даному списку містяться лише актуальні версії онтологій. Це означає, що, наприклад, якщо онтологія із назвою “test.owl” була декілька разів завантажена адміністратором і у списку усіх файлів вона з'являється декілька разів із номерами 1,2,3, і так далі, то у даному списку буде лише остання версія, що не має порядкового номеру.

Таким чином людина. що не знайома із протеже та стандартом owl зможе без проблем працювати із даними в онтології так, наче це була б звичайна реляційна база даних.

Після наповнення онтології даними можна її викачати із системи і використовувати в інших системах для аналізу і пошуку знань уже із використанням різонерів.

Справа на сторінці для роботи із даними знаходиться дерево класів. Це дерево класів онтології, вибраної користувачем. При натиску по класу зявляється таблиця із обєктами даного класу. Обєктом (інстансом) даного класу вважається будь який інстанс у якого є .....

Для того, яби створити новий інстанс даного класу , треба натиснути кнопку New....

При натиску система читає мета дані онтології (дерево класів і проперті, що асоційовані із цими класами) і тоді на основі інформації про клас будує форму для стоврення обєкту даного класу. У формі також проводиться валідація (чи поле мандаторне чи ні) , для обджект пропертей створюється поле із випадаючим списком із усіма обєктами, що можуть бути обрані (список складається на основі інформації про пропертю і тип інстансів, на які вона може ссилатися).

Таким чином на одній формі можна заповнити усі поля обєкту. Це означає, що кількість мануальних дій. потрібних для створення і заповнення даних про один інстанс в онтології зменшується. І створення об’єктів в онтології стає більш прозорим.

Після того, як була вказана онтологія, з якою хоче працювати користувач, - він може приступити до роботи - занесенню даних.

Даний функціонал знаходиться на сторінці “Class Tree” - сторінці із деревом класів. На цю сторінку можна перейти, натиснувши кнопку “Class Tree” на головній панелі сайту.

Дана сторінка складається із двох частин : панелі із деревом класів онтології і панелі для відображення інстансів онтології (рисунок 5.19).

Кореневим класом кожної онтології є “Root class”. Якщо у класу є підкласи, біля його назви є стрілочка-кнопка, за допомогою якої можна розгорнути клас для перегляду його підкласів. Таким чином, можна переглянути всі класи, що є в обраній онтології (рисунок 5.20).

Таблиця із інстансами за замовчуванням показує лише 2 стовпчики, що є стандартними для кожного класу і інстансу - назву, що є частиною URI, та стовпчик із доступними діями.

Були реалізовані такі режими перегляду інстанів - із використанням наслідування і без.

Один із них дозволяє переглядати лише інстанси саме обраного класу. Інший що дозволяє переглядати інстанси обраного класу так само, як і інстанси його підкласів.

Система також дозволяє налаштувати проперті, відображені в таблиці (також використовуючи інформацію про класи і їх проперті). За замовчуванням таблиця має лише 2 стовпчики - назву інстансу, що є унікальним ідентифікатором в онтології і доступні дії. Для кожного класу також є набір унікальних властивостей, що доступні лише для нього.

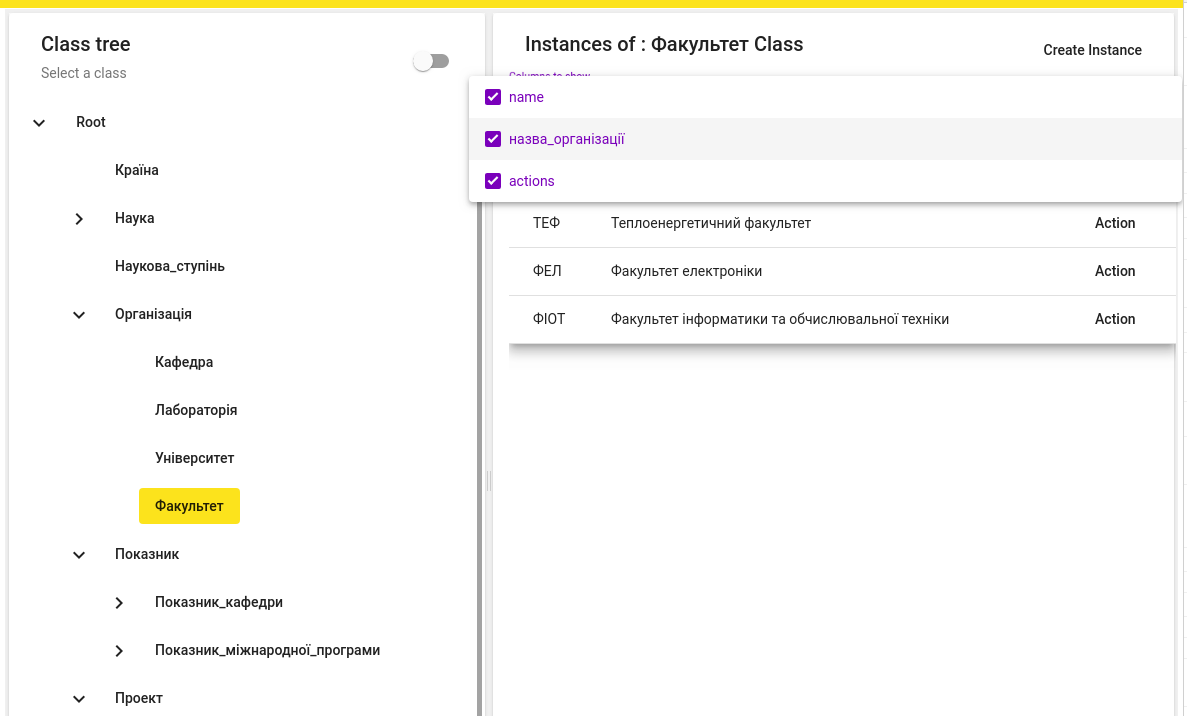


Рисунок 5.24 - Відображення додаткових стовпців у для таблиць інстансів

Коли на панелі класів є обраний якийсь клас - на панелі із таблицею інстансів з'являється кнопка для створення нового інстансу обраного класу. Ця кнопка доступна лише для ролей “admin” та “rw\_user” і лише якщо обраний не кореневий клас (інстанс, якого створити неможливо).

Після кліку по цій кнопці з'являється діалог для створення інстансу із усіма потрібними полями. Діалог складається із одного головного поля - назви інстансу та полів для усіх властивостей, що є у класу, інстанс якого користувач намагається створити.

Якщо клас має додаткові властивості, поля для них з’являться у формі створення інстансу. Наприклад, клас показник кафедри має такі властивості : посилання на кафедру, значення (число), коментар (вільний текст), рік (дата). Якщо властивість є типу “посилання”, то поле дозволить користувачеві обрати інстанс із можливих. На даному скріншоті видно, кафедру можна вибрати серед усіх кафедр, що є в онтології (для прикладу у систему була додана лише одна кафедра - АПЕПС) (рисунок 5.28).

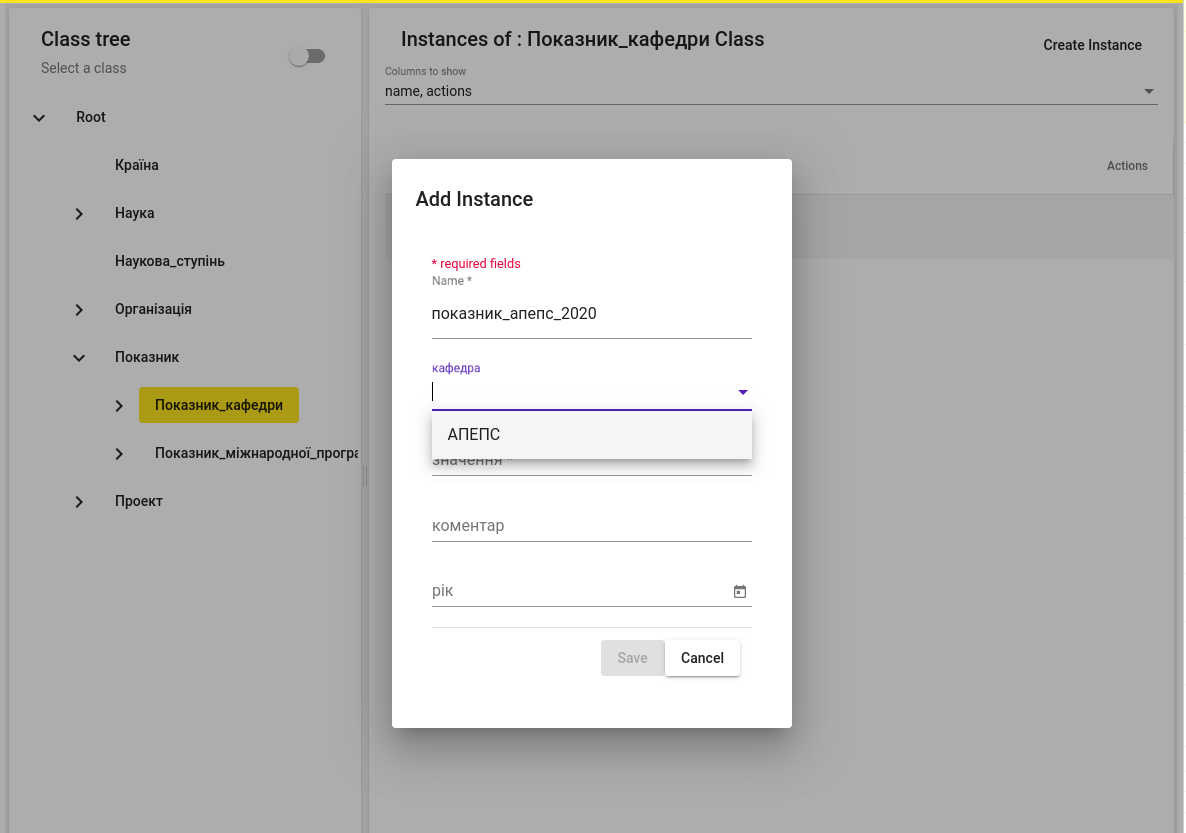


Рисунок 5.28 - Вказування значення властивості типу посилання

Усі створені інстанси можна редагувати або видаляти (ці дії теж доступні лише для користувачів із ролями “admin” та “rw\_user”) (рисунок 5.29).

# Висновки

Використання розробленої системи значно зменшує час, потрібний для ознайомлення із онтологіями і степінь входження. Інтерфейс є простим. Система дозволяє працювати одночасно над однією онтологією в онлайн режимі із різних пристроїв. Також система дозволяє збільшити надійність даних, адже онтологія зберігається на сервері а не на персональному компютері.

Оскільки в системі передбачені різні ролі і степені доступу . система дозволяє підвищити степінь надійності та попередити помилки.

# Джерела інформації (30 штук)

https://github.com/ontop

https://www.w3.org/2001/sw/wiki/Ontop

https://link.springer.com/chapter/10.1007/978-3-642-41335-3\_35

https://link.springer.com/chapter/10.1007/978-3-662-45495-4\_6

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.401.7458&rep=rep1&type=pdf

https://vern.pw/1608881033.pdf

https://link.springer.com/chapter/10.1007/978-3-540-39964-3\_54

https://link.springer.com/chapter/10.1007/11595014\_22

https://corais.org/sites/default/files/ontology\_development\_101\_aguide\_to\_creating\_your\_first\_ontology.pdf

отут порівняння різних тулів для роботи із онтологіями

https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.3953&rep=rep1&type=pdf

https://www.researchgate.net/profile/Thabet\_Slimani1/publication/288258366\_Ontology\_Development\_A\_Comparing\_Study\_on\_Tools\_Languages\_and\_Formalisms/links/56d1369608ae85c823487dc6.pdf

https://watermark.silverchair.com/btg194.pdf?token=AQECAHi208BE49Ooan9kkhW\_Ercy7Dm3ZL\_9Cf3qfKAc485ysgAAAr4wggK6BgkqhkiG9w0BBwagggKrMIICpwIBADCCAqAGCSqGSIb3DQEHATAeBglghkgBZQMEAS4wEQQMMDX9t7F8ZendkVjIAgEQgIICcfrNN05TQWILVLLeW2dr\_cS07TfSN0qn7iUSFn3MGPtnkGrY-kzCvevVULRAQXOEBlW8fGMC7dIoyh6Z5Rgo0h3VDLBbZGBLlX3a93UJyZZn3N3XmAK-vIafqtrAlDYuuTtey53LlSvZ7T0Hf0HLWSJ3SsWsimBiL9UkHduzvFZMGQneh2NpKLp6Rd1vQ6djTicAYEwoV6HuOHDbyGpMT1\_UyNsn1Dd1fYlRGt3Fdzp-fVH\_FHb-2z7rteBvPapKCh30Zu8PuTM-Nl2SfkAEqksQ3hq4C3U5-CHJQ2SyGQyRLvrDt3hBDtgv5LRe-VL1nh4AuAWuaNbkkvT24lBljgrylh1FqW3Kdl-mgzkVNSmBTtkAY513rcR7rHbWwOfjDNLk-qR8cYG8Zof9E78\_Rrw1kWmsEueZV9itqMrA5K-A1B47EJFzfY1j4ByPvDo1VAy6Tp8uC6rGCJLsFYuFX2uC2SCg9XXUALr7xMf6f4nhinSVTUCv3fENa-G\_ga1nT9exGcmFwY84LsQtubuwogYgRKO9fOqcRgqOmpXM6VkrVkLRFrh483O2hXw-AL92V4835OYuuSEbVKiIjmgwjZZuaVDvumkfHr8RenjxuOQ8O6CcVZWT2FwNOm2H6PHFSxd6auNpIvSXwaw-ipxKm0aIyPC34gcXxTOZXz21jyW-0CHRXrtoqM4fiBREyHnYo51BePdNP-\_dSwZ\_JeDB4JCXVZFex7gurBnTHZUeMj6koYzbHzQ3p\_KNtRVZ\_SspFpqlK\_LqsgbAsc03VnykDDPVpdtQ6s3L7905aOIuavaminejkWkGQ5XC7TKWIMdtiPw