

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

« » травня 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
на здобуття освітнього ступеня бакалавр

зі спеціальності 122 - Комп'ютерних наук,
освітньо-професійної програми «Інформатика»
на тему: «Інформаційна система багатоагентного збору даних щодо змісту
навчальних дисциплін»
здобувача групи ІНпн – 01с Кузьменка Сергія Сергійовича

Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело.

Сергій КУЗЬМЕНО

_____ (підпис)

Керівник,
старший викладач кафедри
комп'ютерних наук, к.т.н., доцент

Борис КУЗІКОВ

_____ (підпис)

Суми – 2024

Сумський державний університет
Центр заочної, дистанційної та вечірньої форм навчання
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття освітнього ступеня бакалавра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІНпн-01с Кузьменко Сергія Сергійовича

1. Тема роботи: «Інформаційна технологія прогнозування курсу валют»
затверджую наказом по СумДУ від «26» квітень 2024 р. № 0438-VI
2. Термін здачі здобувачем кваліфікаційної роботи до 05 червня 2024 року
3. Вхідні дані до кваліфікаційної роботи _____
4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)
1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.
2) Огляд технологій, що використовують збір інформації з різних джерел. *3) Розробка web-додатку для аналізу змісту навчальних дисциплін.* *4) Аналіз результатів.*
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
6. Консультанти до проекту (роботи), із значенням розділів проекту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «02» листопад 2023 р.

Завдання прийняв до виконання _____
(підпис)

Керівник _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>	01.11.2023	
2	<i>Огляд технологій, що використовують збір інформації з різних джерел</i>	24.11.2023	
3	<i>Розробка web-додатку для аналізу змісту навчальних дисциплін</i>	05.05.2024	
4	<i>Аналіз отриманих результатів</i>	11.05.2024	
5	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>	15.05.2024	

Здобувач вищої освіти _____
(підпис)

Керівник _____
(підпис)

АНОТАЦІЯ

Записка: 61 стр., 50 рис., 1 додаток, 20 використаних джерел.

Обґрунтування актуальності теми роботи – Актуальність теми "Інформаційна система багатоагентного збору даних щодо змісту навчальних дисциплін" полягає в необхідності вдосконалення навчального процесу через ефективний збір, аналіз та використання інформації про зміст предметів. Це дозволить індивідуалізувати навчання, оптимізувати ресурси навчальних закладів, вдосконалювати програми та підвищувати якість освіти відповідно до потреб сучасного ринку праці та технологічних вимог.

Об'єкт дослідження — Багатоагентний збір інформації із відкритих джерел.

Предмет дослідження — Практичні та теоретичні аспекти багатоагентного збору інформації із відкритих джерел щодо змісту навчальних дисциплін.

Мета роботи — створення інформаційної системи для багатоагентного агрегації змісту силабусів на основі даних із відкритих джерел.

Методи дослідження — аналіз інформації на основі вхідних даних за допомогою багатоагентної системи, а також існуючих методів та технологій, пов'язаних із збором даних щодо навчальних дисциплін.

Результати — розроблено інформаційну систему, яка за вхідними даними, а саме назвою двох університетів та назвою дисципліни знаходить силабуси та дає користувачеві представлення про вміст для подальшого аналізу.

ІНФОРМАЦІЙНА СИСТЕМА, БАГАТОАГЕНТНИЙ ЗБІР ДАНИХ,
СИЛАБУС, JAVA, SPRING, HIBERNATE.

ЗМІСТ

ВСТУП.....	5
1 АНАЛІТИЧНИЙ ОГЛЯД.....	7
1.1 Аналіз існуючих рішень.....	7
1.2 Представлення, та пошук силабусів.....	13
1.3 Постановка задачі.....	16
2 ВИБІР МЕТОДУ РОЗВ’ЯЗАННЯ ЗАДАЧІ	17
2.1 Вибір середовища розробки	17
2.2 Патерн MVC	19
2.3 Фреймворк Spring MVC	20
2.4 Hibernate & MySQL	21
2.5 Selenium	23
2.6 Інструменти та утиліти для скачування файлів та роботи з внутрішнім контентом	24
3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ	29
3.1 Формування вхідних даних	29
3.2 Програмна реалізація	29
3.2.1 Maven dependency.....	29
3.2.2 Налаштування Entity	30
3.2.3 Налаштування конфігураційних файлів.....	32
3.2.4 DAO та Service.....	33
3.2.5 MasController	34
3.2.6 Агент СумДУ	37
3.2.7 Агент Google.....	38
3.3 Аналіз результатів	39
ВИСНОВКИ	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	43
ДОДАТОК	45

ВСТУП

Актуальність. В університетах часто постає питання про мобільність студентів, однією з ключових труднощів у цьому контексті є відсутність спеціалізованого сервісу чи додатку, який би дозволяв порівнювати силабуси однакових дисциплін в різних вищих навчальних закладах. Ця проблема стає значущою перепорою на шляху академічної мобільності та може суттєво ускладнити процес обміну студентами між університетами. Для того щоб завадити цій проблемі потрібно враховувати що університети можуть мати різні підходи до вивчення та викладання дисциплін і це ускладнює процес визначення еквівалентності курсів для студентів, які хочуть змінити своє місце навчання. Відсутність необхідного програмного забезпечення (ПЗ) може призвести до неточностей в процесі визначення кредитів та взаємозамінності курсів між університетами. Також постає питання зручності для викладачів університатів, потрібен спосіб швидко і зручно порівнювати дисципліни в ЗВО. Сервіси для перегляду списків університетів зі всього світу: QS World University Rankings, Academic Ranking of World Universities (ARWU), Times Higher Education (THE) і інші. Вони можуть надати інформацію про рейтинг університетів та їхні програми. Але це теж все робиться вручну та потребує аналізу курсів, які можуть бути схожими по назві але відрізнятися матеріалом який викладається. Також є проект під назвою “Цифровий університет – Відкрита українська ініціатива”[15], який має на меті створити цифровий вищий навчальний заклад, і для цього спочатку потрібно зрозуміти, які дисципліни викладаються у університетів партнерів.

Об’єкт дослідження. Багатоагентний збір інформації із відкритих джерел.

Предмет дослідження. Практичні та теоретичні аспекти багатоагентного збору інформації із відкритих джерел щодо змісту навчальних дисциплін.

Метою роботи є створення інформаційної системи для багатоагентного агрегації змісту силабусів на основі даних із відкритих джерел.

Гіпотеза. Потреба в інструменті, що дозволить швидко та ефективно порівняння силабусів, стає критичною в умовах зростаючого інтересу до глобальної освіти та студентської мобільності. Зазначена проблема визначає необхідність створення інноваційного підходу, який сприятиме зручності та прозорості у процесах переходу між університетами та обміну знаннями між академічними установами.

Новизна. Створення інноваційного підходу до цієї проблеми вимагає не лише технічної ефективності, але й урахування вимог різних систем навчання та регіональних особливостей. Інтегрований підхід до порівняння силабусів має стати кроком у напрямку глобальної освітньої взаємодії, забезпечуючи студентам, викладачам та адміністраторам університетів інструмент, який полегшить вибір та організацію академічного обміну.

Отже, виходячи з актуальності даної роботи, головною метою нашої роботи є створення інформаційної системи багатоагентного збору даних щодо змісту навчальних дисциплін.

Для досягнення мети сформульовані наступні задачі роботи:

- Проаналізувати існуючі рішення на тему інформаційної системи багатоагентного збору даних.
- Дослідити представлення силабусів.
- Знайти та дослідити інструменти та утиліти для пошуку силабусів, та для збору даних.
- Спроекувати web-додаток, та реалізувати його функціонал.
- Проаналізувати результати.

Структура. Дана робота складається зі вступу, аналітичного огляду, постановки задачі, вибір методу розв'язання поставленої задачі, опису програмного забезпечення інформаційної системи, висновків, списку використаних джерел та додатків.

1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Аналіз існуючих рішень

На сьогоднішній день жоден проект не може обійтися без аналізу існуючих рішень, адже цей етап є необхідним для створення успішного проекту. Потрібно врахувати всі переваги та недоліки всіх подібних рішень, та проаналізувати їхній функціонал, щоб отримати представлення як саме має виглядати наш проект.

Провівши аналіз мережі інтернет в пошуках аналогів для нашої програми, ми прийшли до того, що існують сайти з списками топових вузів, які також мають опис дисциплін які в них викладаються найпопулярніші з них:

1. Times Higher Education [1]
2. QS World University Rankings [2]
3. Academic Ranking of World Universities (ARWU) [3]

Ці сайти дають нам представлення про рейтинги всіх університетів світу з деякими описами їхніх структур та напрямків навчання.

1) Times Higher Education

Як бачимо на рис. 1.1 на головній сторінці цього сайту можна знайти багато цікавої інформації про вищу освіту. Переходимо на вкладку Rankings та тиснемо кнопку world і сайт нас переводить на банер з топом університетів – рис 1.2.

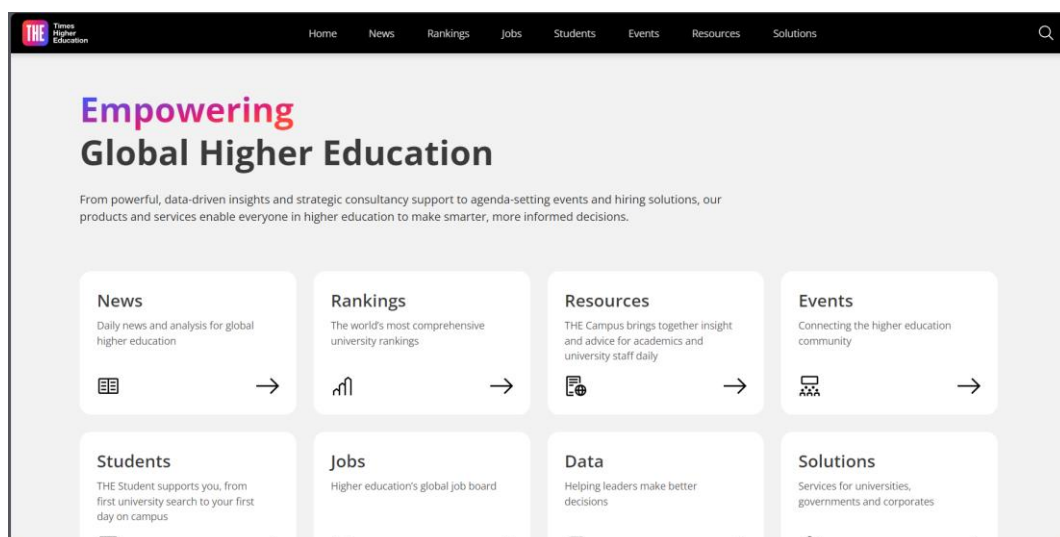


Рисунок 1.1 – Times Higher Education Main Page

На рис. 1.2 ми бачимо топ університетів всього світу зі вказаною кількістю всього студентів, студентів на штат, процент іноземних студентів та відношення статей.

RANKING	SCORES					
Rank	Name Country/Region	No. of FTE Students	No. of students per staff	International Students	Female:Male Ratio	
1	University of Oxford United Kingdom	21,750	10.9	42%	49 : 51	
2	Stanford University United States	14,517	6.4	23%	47 : 53	
3	Massachusetts Institute of Technology United States	11,085	8.0	33%	41 : 59	
4	Harvard University United States	20,050	9.0	25%	51 : 49	
5	University of Cambridge	20,565	11.5	38%	48 : 52	

Рисунок 1.2 – World University Rankings

На рисунку 1.3 ми можемо бачити сторінку обраного університету - Oxford, з розбитими на категорії відомостями. Також ми бачимо коротки опис університету та точно адреса місцезнаходження університету.

United Kingdom
University of Oxford

1st World University Rankings 2024
4th World Reputation Rankings 2022
1st European Teaching Rankings 2019

Get free study advice | Admissions Support

University Offices, Wellington Square, Oxford, Oxfordshire, OX1 2JD, United Kingdom

OVERVIEW | RANKINGS | COMPARE | **SUBJECTS** | KEY STATS | JOBS | ENQUIRE

ABOUT UNIVERSITY OF OXFORD

The University of Oxford is the oldest university in the English-speaking world and the world's second oldest surviving university. While its exact founding date is unknown, there is evidence that teaching took place as far back as 1096.

Located in and around Oxford's medieval city centre, the university comprises 44 colleges and halls, and over 100 libraries, making it the largest library system in the UK.

Students number around 22,000 in total, just over half of whom are undergraduates while over 40 per cent are international, representing 140 countries between them.

Рисунок 1.3 – University of Oxford

Як бачимо цей сайт надає короткий опис, топ, порівнює кількість студентів та викладачів.

Наприклад на рисунку 1.4 ми можемо побачити перелік галузей які викладаються в цьому університеті.

<p>Social sciences</p> <p>Geography</p> <p>Communication & Media Studies</p> <p>Politics & International Studies (incl Development Studies)</p> <p>Sociology</p>	<p>Engineering & technology</p> <p>Chemical Engineering</p> <p>General Engineering</p> <p>Electrical & Electronic Engineering</p> <p>Mechanical & Aerospace Engineering</p> <p>Civil Engineering</p>	<p>Law</p> <p>Law</p>
<p>Physical sciences</p> <p>Chemistry</p> <p>Geology, Environmental, Earth & Marine Sciences</p> <p>Mathematics & Statistics</p> <p>Physics & Astronomy</p>	<p>Life sciences</p> <p>Biological Sciences</p>	<p>Business & economics</p> <p>Economics & Econometrics</p> <p>Business & Management</p> <p>Accounting & Finance</p>
	<p>Computer science</p> <p>Computer Science</p>	<p>Clinical, pre-clinical & health</p> <p>Medicine & Dentistry</p> <p>Other Health</p>
		<p>Psychology</p> <p>Psychology</p>

Рисунок 1.4 – Subjects taught at university of Oxford

Сайт має гарну базу вишів але не має всього потрібного нам функціоналу адже не має можливості переглянути вміст силабусів навчальних дисциплін.

2) QS World University Rankings

На головній сторінці цього сервісу (рис.1.5) ми також бачимо велику кількість різних опцій, і також бачимо рейтинг ЗВО 2024.

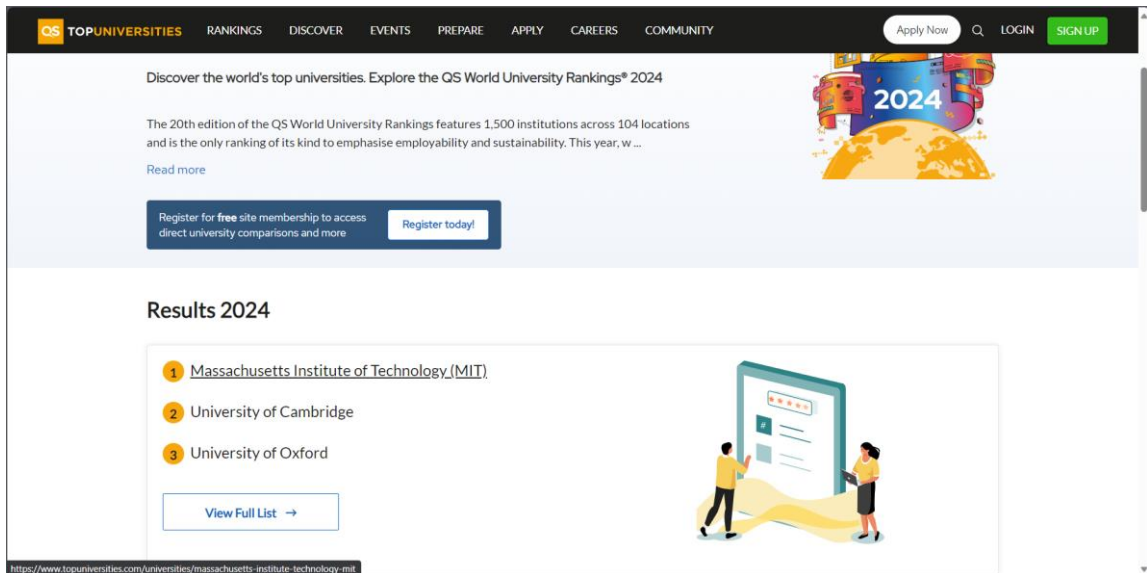


Рисунок 1.5 – QS World University Rankings Main Page

Перейшовши за посиланням на топ, ми бачимо на рисунку 1.6 зручний інтерфейс, фільтрацію та сам список вишів.

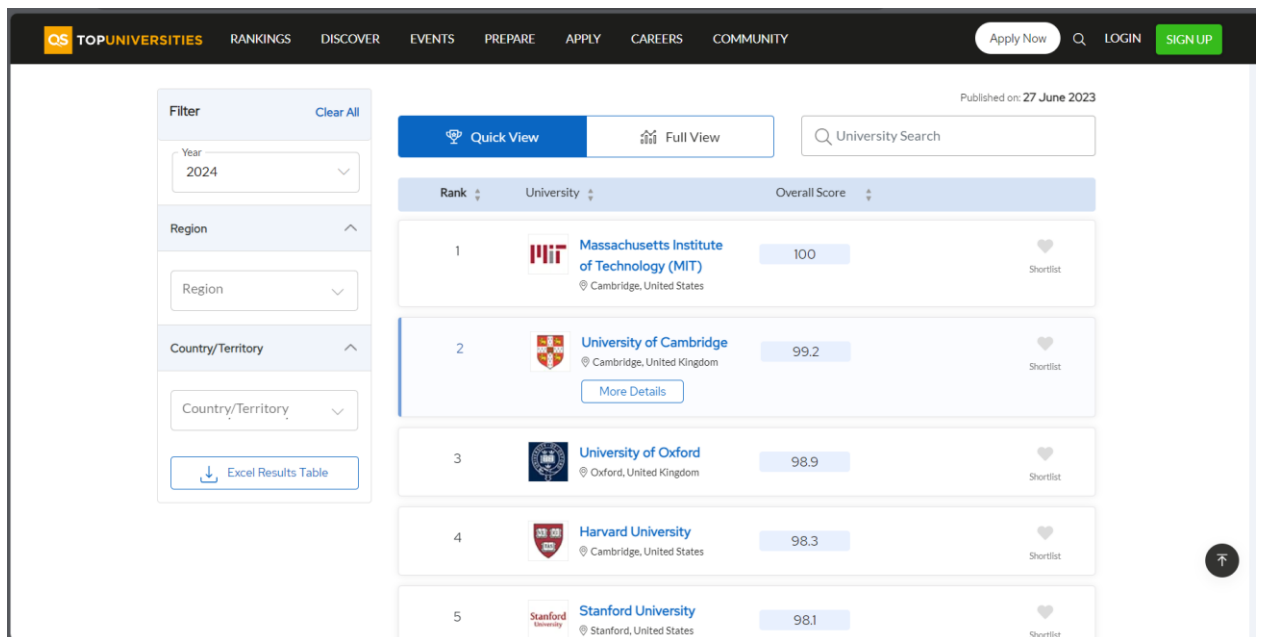


Рисунок 1.6 – QS World University Rankings 2024: Top global universities

Оберемо наприклад також Oxford і бачимо на рисунку 1.7 перелік освітніх програм які викладаються, які згруповані по факультетам.

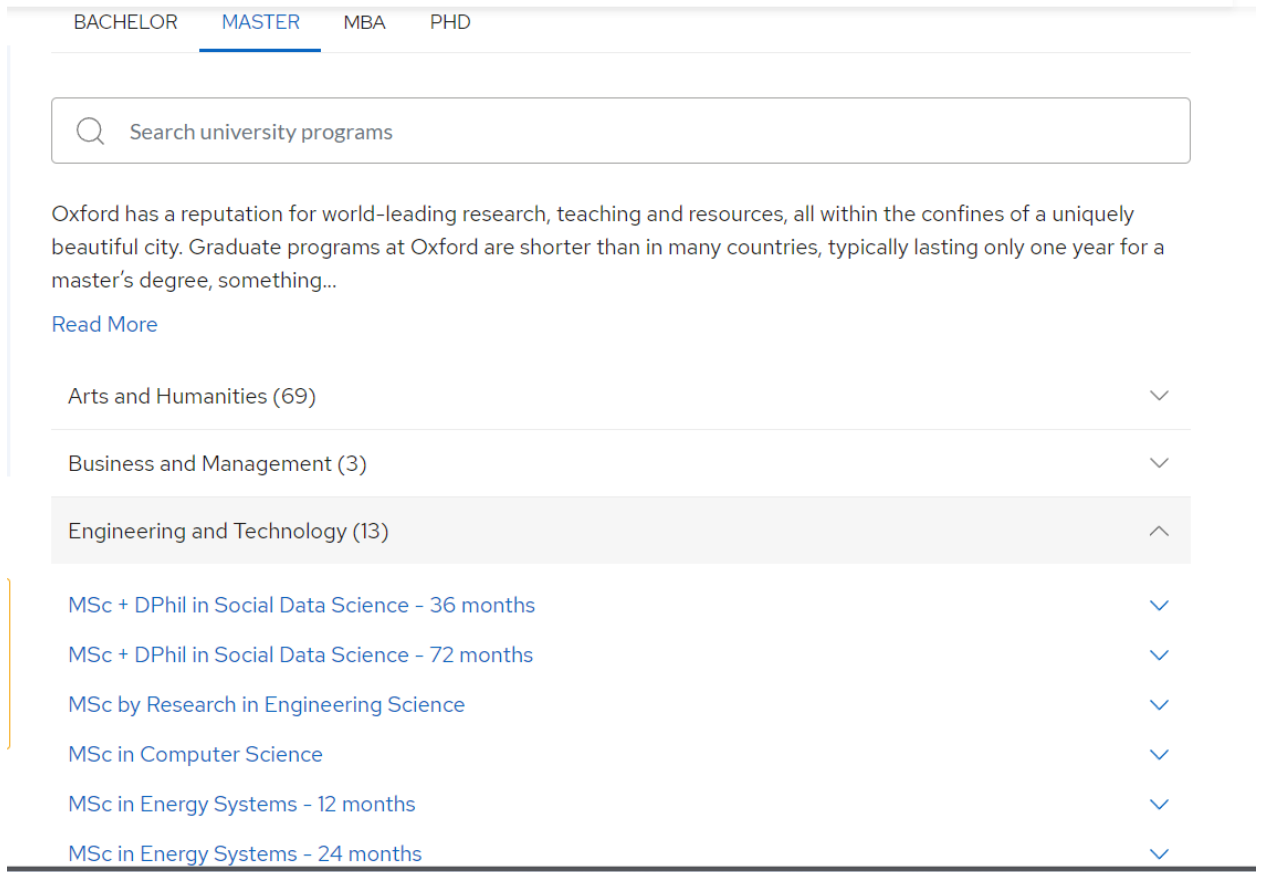


Рисунок 1.7 – Вкладка сайту Available programs

Але в цього сайту аналогічна проблема з попереднім і на ньому також не можна порівняти силабуси цих ЗВО та їх силабуси в автоматизованому форматі, але це можна зробити вручну хоча це і не досить зручно та забере багато часу.

3) Academic Ranking of World Universities (ARWU)

На рис. 1.8 показана головна сторінка, яка одразу рекламує нам рейтинг університетів світу, та дозволяє знайти ЗВО у пошуковому рядку.

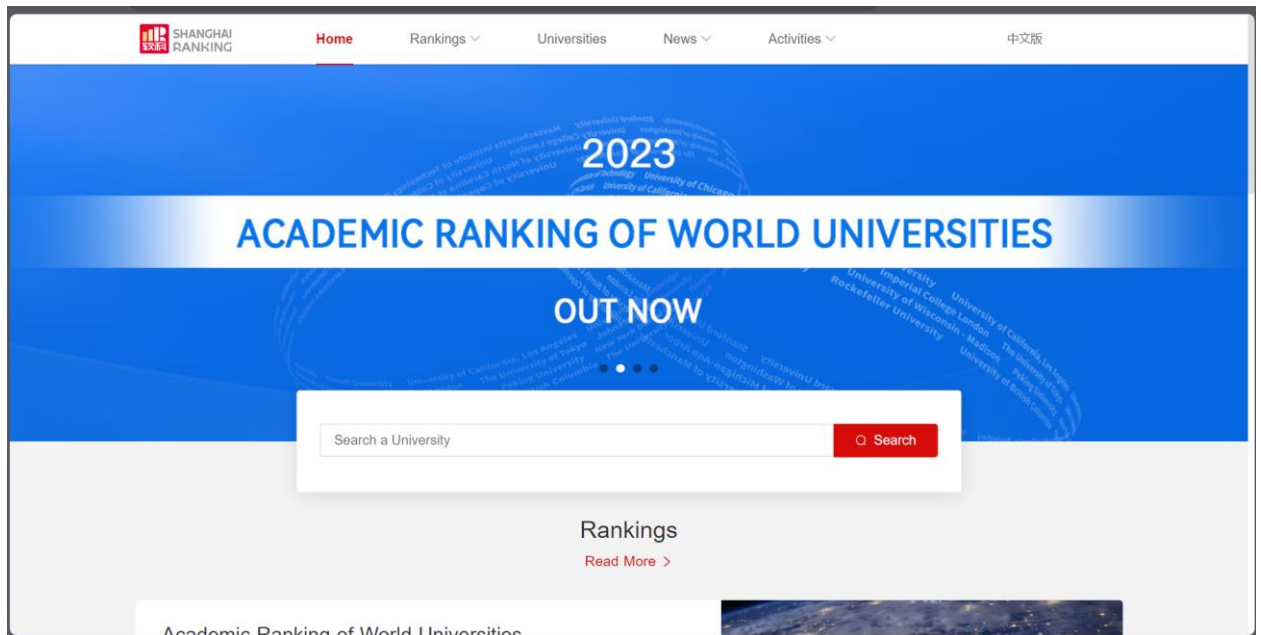


Рисунок 1.8 – Shanghai Ranking Main Page

На рисунку 1.9 можемо бачити сам рейтинг, також для прикладу обираємо Oxford і бачимо нижче на рисунку 1.10 що сайт також надає інформацію про предмети які є в університеті.

World Rank	Institution	Country/Region	National/Regional Rank	Total Score	Alumni
1	Harvard University	USA	1	100.0	100.0
2	Stanford University	USA	2	74.8	45.6
3	Massachusetts Institute of Technology (MIT)	USA	3	69.1	72.5
4	University of Cambridge	UK	1	67.9	78.9
5	University of California, Berkeley	USA	4	63.4	65.8
6	Princeton University	USA	5	60.1	61.7
7	University of Oxford	UK	2	59.5	49.9
8	Columbia University	USA	6	55.3	58.5

Рисунок 1.9 – Academic Ranking of World Universities

The screenshot displays the Shanghai Ranking website for the University of Oxford. The page features the university's crest and name, along with key information such as its location in Northern Europe, its founding year of 1096, and its address at Wellington Square. A navigation menu at the top includes 'Home', 'Rankings', 'Universities', 'News', and 'Activities'. A prominent section highlights the university's 'Academic Ranking of World Universities' score of 7 for the years 2021, 2022, and 2023. Below this, the 'Global Ranking of Academic Subjects' is presented in two tables. The first table, 'All Subjects', lists subjects and their ranks, while the second table, 'Best Ranked Subjects', lists the top subjects and their ranks.

Subject	Rank
Mathematics	3
Physics	9
Chemistry	39
Earth Sciences	16
Geography	2
Ecology	1
Oceanography	76-100
Atmospheric Science	12

Subject	Rank
Ecology	1
Geography	2
Mathematics	3
Sociology	3
Public Health	4
Political Sciences	4
Computer Science & Engineering	6
Clinical Medicine	6

Рисунок 1.10 – Available programs


Але також як і в попередніх прикладах цей сервіс має той же недолік.

Отже виходячи з вищесказаного ми можемо зробити висновок, що дані сервіси не у повній мірі задовольняють наші потреби, отже створення власної інформаційної системи багатоагентного збору даних щодо змісту дисциплін в контексті академічної мобільності є актуальним варіантом.

1.2 Представлення, та пошук силабусів


Для прикладу розглянемо дисципліну “Мова програмування Java”, для того що зрозуміти як працює пошук, знайдемо кілька силабусів з різних ЗВО. Для цього вводимо в пошуковий рядок браузеру Google Chrome запит “силабус з дисципліни мова програмування Java”.

Як бачимо на рисунку 1.11 браузер після запиту видає результат у вигляді сторінки на яких знаходяться силабуси з різних університетів.

 КПІ ім. Ігоря Сікорського
<https://acts.kpi.ua/syllabuses/uk/SyllabusContent?curriculumId=2620>


Силабус: Мова програмування Java (ICT)

Інтернет **МОВА ПРОГРАМУВАННЯ java** - Робоча **програма** навчальної дисципліни
(Силабус) Реквізити навчальної дисципліни

 khadi.kharkov.ua
<https://mf.khadi.kharkov.ua/fileadmin/F-MECHANIC...> · Файл PDF


СИЛАБУС МОВА ПРОГРАМУВАННЯ JAVA / JAVA ...

Інтернет **СИЛАБУС МОВА ПРОГРАМУВАННЯ JAVA / JAVA PROGRAMMING LANGUAGE SYLLABUS** освітній ступінь бакалавр / bachelor галузь знань 12 Інформаційні ...

 Український державний університет науки і технологій
https://nmetau.edu.ua/file/f16_mova_programuvannya... · Файл PDF

Український державний університет науки і технологій ...

Інтернет **СИЛАБУС** навчальної дисципліни «**Мова програмування Java**» Назва дисципліни **Мова програмування Java** Шифр та назва спеціальності 121 – Інженерія ...

 Факультет електроніки та комп'ютерних технологій
<https://electronics.lnu.edu.ua/.../Java126Horon.pdf> · Файл PDF

Силабус з навчальної дисципліни «Мова програмування Java», ...

Інтернет **Силабус** з навчальної дисципліни «**Мова програмування Java**», що викладається в межах ОПП «Інформаційні системи та технології» першого (бакалаврського)...

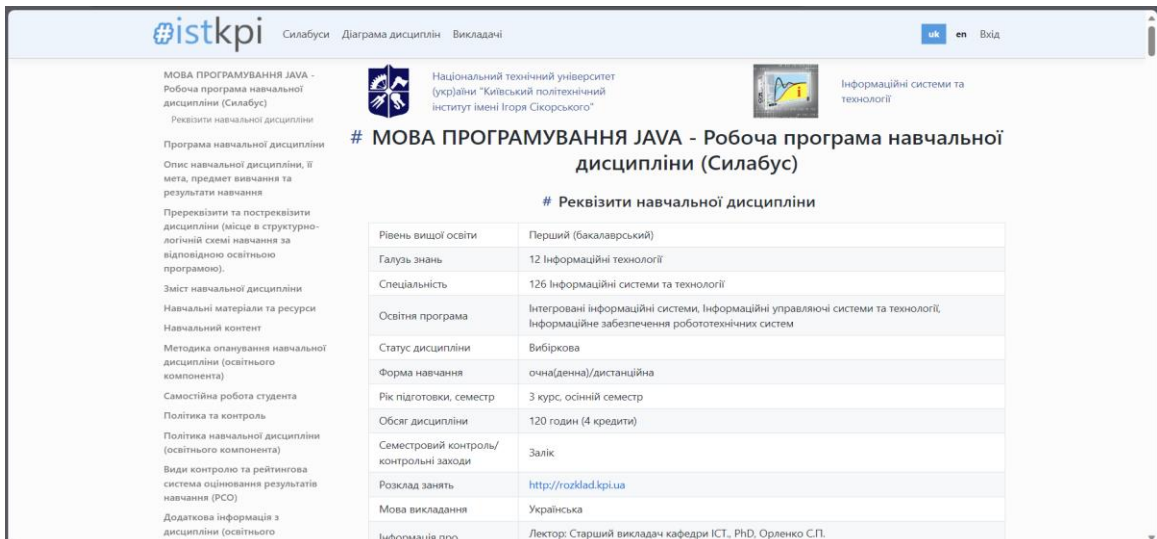
 pnu.edu.ua
<https://kkite.pnu.edu.ua/wp-content/uploads/sites...> · Файл PDF

123 Комп'ютерна інженерія

Інтернет **Силабус** навчальної дисципліни “Об’єктно-орієнтоване **програмування (Java)**” складений відповідно до освітньо-професійної програми підготовки бакалавра ...

Рисунок 1.11 – Відповідь на запит Мова програмування Java

Наприклад сайт КПІ повертає HTML сторінку див. рис. 1.12.



Силабуси | Діаграма дисциплін | Викладані

uk en Вхід

МОВА ПРОГРАМУВАННЯ JAVA - Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Програма навчальної дисципліни

Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Пререквізити та поостреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою).

Зміст навчальної дисципліни

Навчальні матеріали та ресурси

Навчальний контент

Методика опанування навчальної дисципліни (освітнього компонента)

Самостійна робота студента

Політика та контроль

Політика навчальної дисципліни (освітнього компонента)

Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Додаткова інформація з дисципліни (освітнього компонента)

Национальний технічний університет (українці) "Київський політехнічний інститут імені Ігоря Сікорського"

Інформаційні системи та технології

МОВА ПРОГРАМУВАННЯ JAVA - Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	126 Інформаційні системи та технології
Освітня програма	Інтегровані інформаційні системи, Інформаційні управлюючі системи та технології, Інформаційне забезпечення робототехнічних систем
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)/дистанційна
Рік підготовки, семестр	3 курс, осінній семестр
Обсяг дисципліни	120 годин (4 кредити)
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	http://rozklad.kpi.ua
Мова викладання	Українська
Інформація про	Лектор: Старший викладач кафедри ICT, PhD, Орленко С.П.

Рисунок 1.12 – HTML сторінка силабусу КПІ

А сайт ХНАДУ повертає PDF файл – рис. 1.13

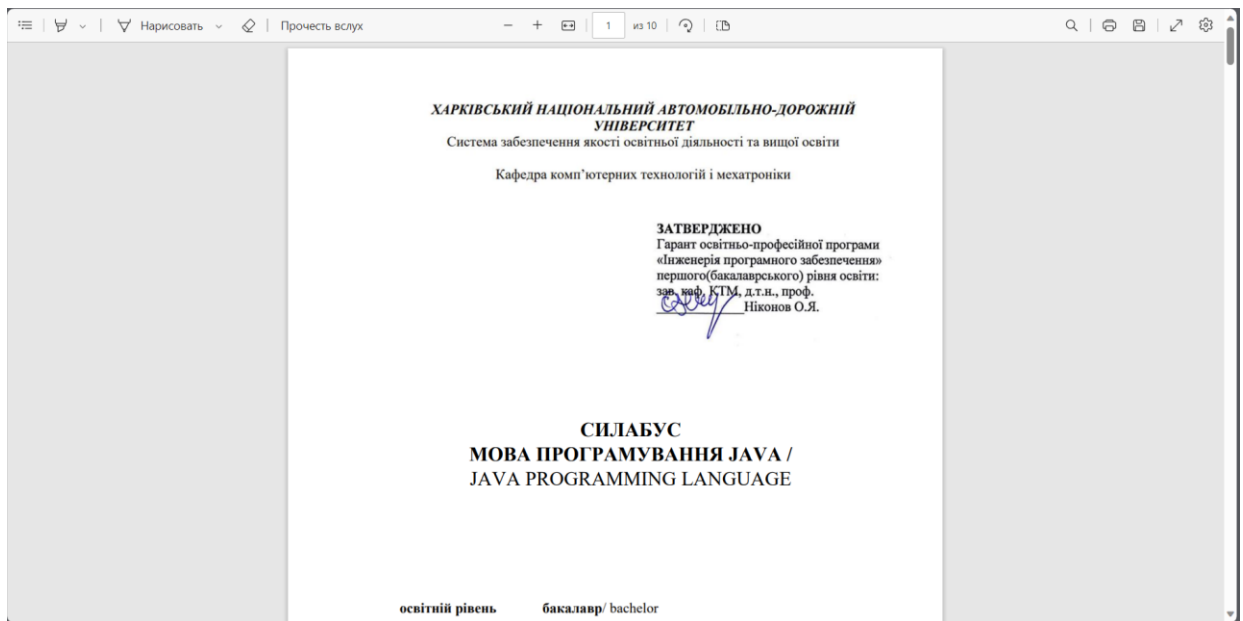


Рисунок 1.13 – Представлення силабусу у форматі PDF

А наприклад сайт Національного університету водного господарства та природокористування не є напряду силабусом, але має посилання на силабус всередині див. рисунки 1.14 та 1.15.

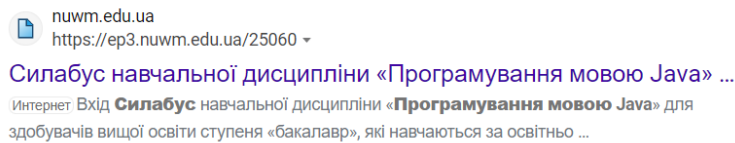


Рисунок 1.14 – Сайт який має посилання в середині

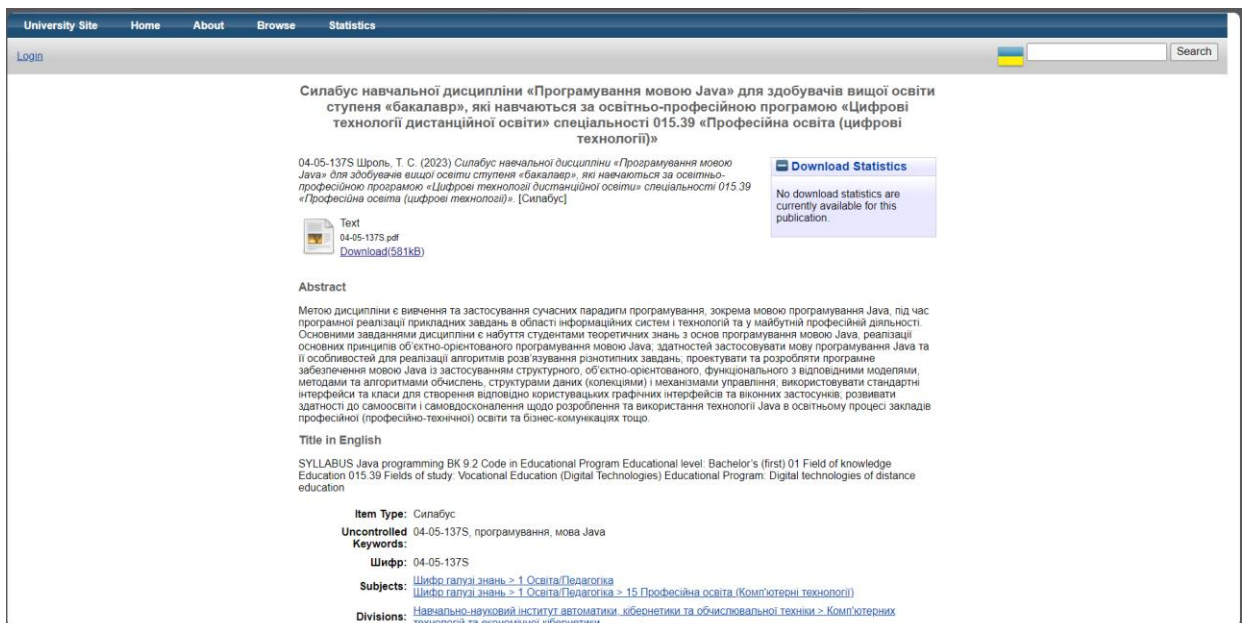


Рисунок 1.15 – Сторінка з посиланням на завантаження диплому

Тому це також потрібно враховувати, щоб знайти потрібну інформацію.

Цей приклад показує всю суть вислову багатоагентний, адже представлені джерела інформації у різному форматі.

1.3 Постановка задачі

Програма, що розробляється призначена для використання в галузі вищої освіти, переважно нею будуть користуватися викладачі та студенти, у меншій степені абітурієнти. Студенти які потребують академічної мобільності зможуть аналізувати підходящі для себе університети. Викладачі зможуть порівняти програму дисциплін яка викладається в вишах. Абітурієнти ж зможуть знайти для себе ЗВО залежно від дисциплін, які хочуть вивчати.

Маючи назву університету та приблизну назву дисципліни, знайти в інтернеті силабус та скачати його, видобути з нього текстові дані, адже це може бути як word документ так і pdf, або html сторінка. Ці данні будуть застосовані для аналізу або порівняння з інформацією з іншого силабуса.

До ключових функціональних особливостей продукту, що розробляється слід віднести:

1. Можливість виконання пошуку силабусів за ключовими словами, такими як назва ЗВО чи назва дисципліни.
2. Фільтрація результатів пошуку.
3. Фільтрація за назвою країни.
4. Релевантність пошукової видачі: система повинна повертати результати, які найточніше відповідають запиту користувача.
5. Система повинна бути здатна взаємодіяти з іншими інформаційними системами для отримання додаткової інформації, наприклад, із системами управління навчальним процесом в ЗВО.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Проектування системи

Спочатку потрібно спроектувати систему для нашого додатку, розглянемо дві UML-діаграми: діаграму послідовностей (Sequence Diagram Рис. 2.1) та діаграму діяльності (Activity Diagram Рис. 2.2).

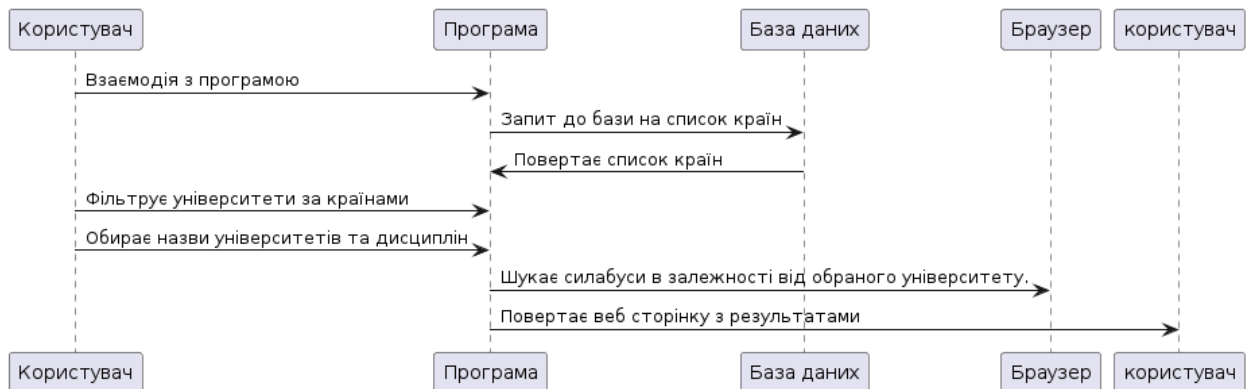


Рисунок 2.1 – Діаграма послідовності багатоагентної системи щодо змісту навчальних дисциплін

Як бачимо на діаграмі послідовності, в нас є певний користувач який взаємодіє з системою, а система в свою чергу звертається до допоміжних компонентів, таких як база даних чи браузер. Діаграма послідовності допомагає нам поверхово візуалізувати процеси, полегшити виявлення недоліків у дизайні та допоможе в документуванні проекту.

Для візуалізації потоку використання нашого майбутнього додатку ми створили діаграму діяльності, яка є розширеною версією блок-схеми, на ній ми можемо бачити послідовну активність від дії до дії.

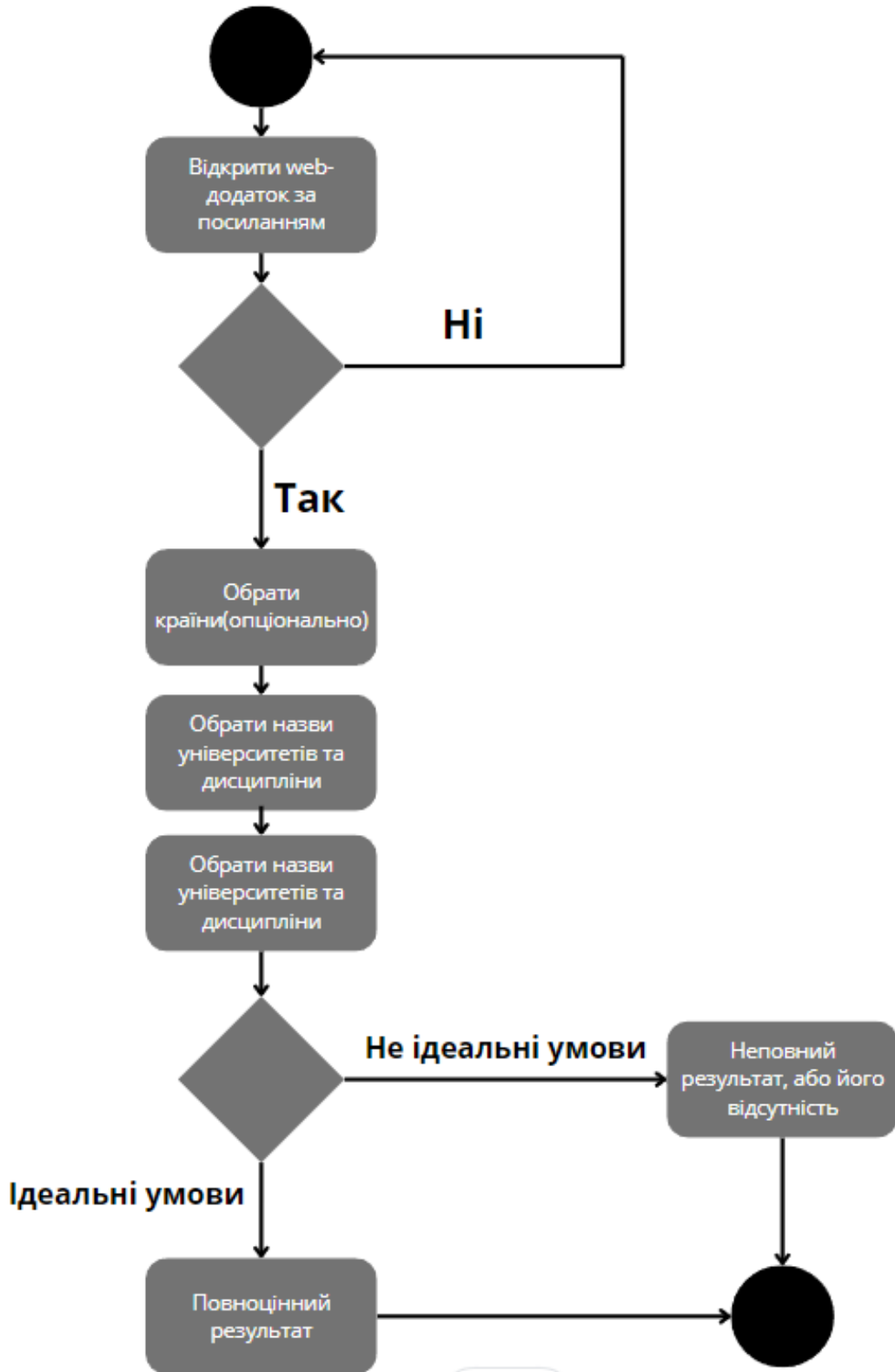


Рисунок 2.2 – Діаграма діяльності багатоагентної системи щодо змісту навчальних дисциплін

2.2 Патерн MVC

Наш додаток буде побудований на фреймворку Spring, який базується на патерні MVC (Рис. 2.3). Отже MVC представляє собою тріаду компонентів Model-View-Controller – це допомагає розділяти нам логіку нашого додатку.

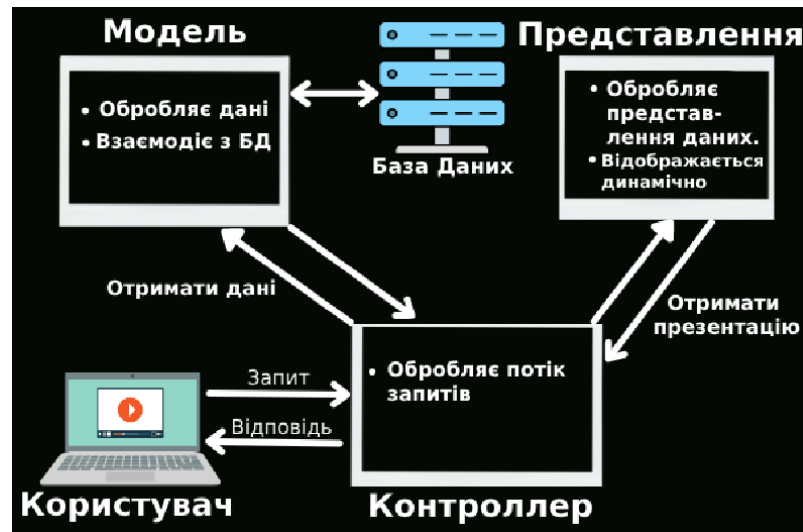


Рисунок 2.3 – Шаблон проектування MVC[8]

Модель представляє собою данні додатку, а також методи для роботи з ними. Представлення відповідає за відображення даних моделі. Контроллер оброблює запити користувача, змінює модель і в результаті передає дані моделі в відображення. UML-діаграму нашого проекту за патерном MVC можна побачити на рисунку 2.4.

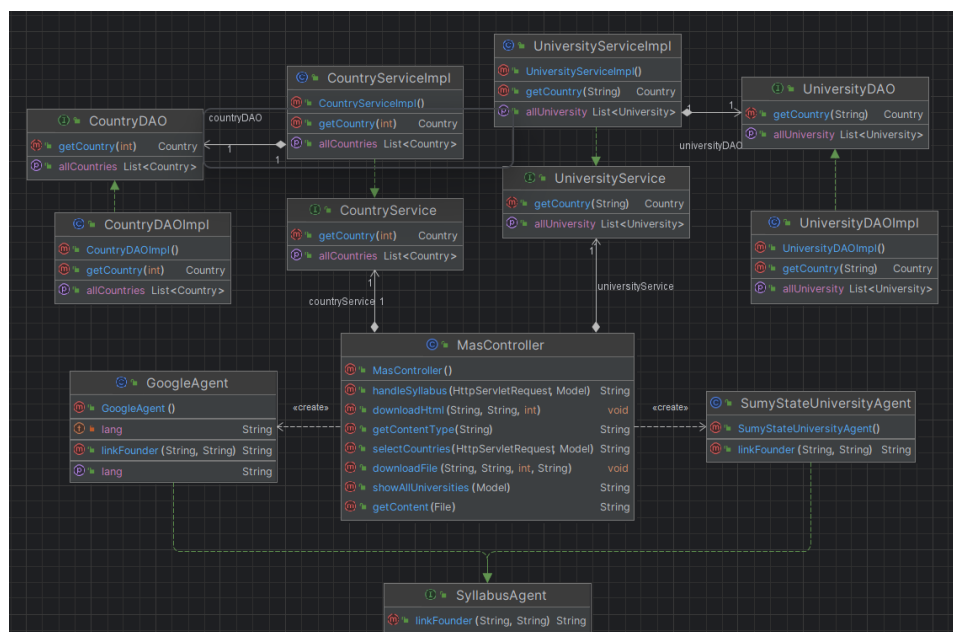


Рисунок 2.4 – UML-діаграма багатоагентної системи.

2.3 Фреймворк Spring MVC

Spring Framework – це відкрита програмна платформа для Java, що включає контейнери з підтримкою інверсії керування. Його основні функції доступні для всіх Java-застосунків, і він має додаткові можливості для розробки веб-додатків на Java EE. Spring Framework не прив'язаний до конкретної програмної моделі, що робить його популярним серед Java-розробників як альтернатива або доповнення до моделі Enterprise JavaBean (EJB).

У нашому проекті ми застосовуємо розширення цього фреймворку, а саме – Spring Web MVC, він базується на принципах моделі-перегляд-контролера (MVC), з центральним елементом - DispatcherServlet, який маршрутизує запити до відповідних обробників. Цей фреймворк надає широкий набір можливостей, включаючи роздільну здатність перегляду, локалізацію, теми оформлення та завантаження файлів. Обробники створюються за допомогою анотацій, таких як `@Controller` та `@RequestMapping`, що дозволяє гнучко обробляти запити. З виходом версії Spring 3.0, механізм контролера також підтримує створення веб-сайтів та RESTful додатків через анотації, такі як `@PathVariable`.

У Spring Web MVC можна використовувати будь-який об'єкт як команду або об'єкт підтримки форми, не обов'язково реалізуючи специфічний інтерфейс чи базовий клас. Зв'язування даних дуже гнучке, приймаючи невідповідності типів як помилки перевірки, що дозволяє їх обробляти програмно. Це уникає необхідності дублювання властивостей бізнес-об'єктів для обробки недійсних даних.

Управління переглядами дозволяє контролеру визначати, які дані передавати у представлення та як обробляти вихідний формат. Можливість налаштування імен перегляду дозволяє просто налаштувати цей процес. Модель (M у MVC) використовує інтерфейс Map, що дозволяє абстрагувати технологію перегляду, та інтегруватися з різними технологіями візуалізації або генерувати різноманітний вміст, такий як XML, JSON та інші.

2.4 Hibernate & MySQL

Для розробки мого проекту я обрав Hibernate як ORM (Object-Relational Mapping) і MySQL як систему керування базами даних. Цей вибір заснований на кількох ключових перевагах, які ці технології пропонують.

Hibernate - це фреймворк для роботи з базами даних дозволяє мені працювати з об'єктами в мові програмування Java безпосередньо, не звертаючись безпосередньо до бази даних. Ось декілька переваг, які вирішили мій вибір на користь Hibernate:

1. Універсальність: Hibernate підтримує різні бази даних, що дозволяє мені з легкістю мігрувати між різними реляційними СУБД без значних змін у моєму коді.

2. Зручність використання: Hibernate забезпечує простий і зрозумілий інтерфейс для взаємодії з базою даних, що спрощує розробку та підтримку коду.

3. Керування об'єктами: Завдяки Hibernate я можу працювати з об'єктами, зручно використовуючи реляційні дані. Це спрощує розробку складних систем та зменшує кількість SQL-коду.

В нашому проекті використовується фільтрація університетів, для цього ми обираємо дві країни в яких є університети, і отже у нас формується зв'язок один до багатьох, що означає що в одній країні може бути кілька університетів.

Для проекту була створена ER-діаграма(Рис. 2.5) з урахуванням кожної країни і всіх університетів в них.

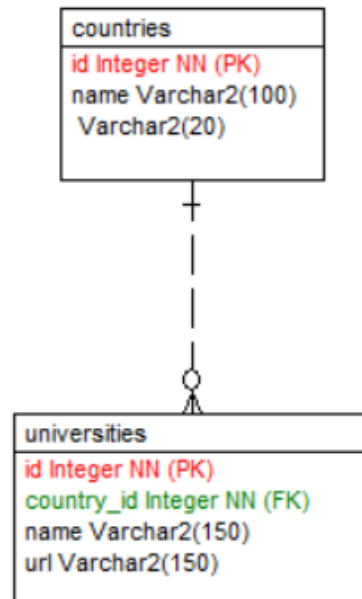


Рисунок 2.5 – ER-діаграма зберігання університетів у базі

Щодо MySQL, ось деякі переваги, які зробили його моїм вибором для бази даних у проекті:

1. Відкритість та спільнота: MySQL - це вільна та відкрита СУБД з широкою спільнотою користувачів та активною підтримкою. Це дозволяє мені швидко отримувати допомогу та вирішувати проблеми, які виникають під час розробки.
2. Швидкодія: MySQL відомий своєю високою швидкодією та ефективністю роботи з даними, що особливо важливо для проектів з великим обсягом інформації.
3. Масштабованість: MySQL легко масштабується з ростом обсягу даних та навантаження, що дозволяє моєму проекту розвиватися без проблем з продуктивністю.

Представлення в базі даних MySQL можна побачити на рис. 2.6

Таблиця	Дія	Рядки	Тип	Зіставлення	Розмір	Фрагментовані
<input type="checkbox"/> countries	☆ [іконки]	246	MyISAM	latin1_swedish_ci	25.4 КБ	-
<input type="checkbox"/> universities	☆ [іконки]	16,773	MyISAM	utf8_general_ci	1.5 МБ	-
2 таблиці	Всього	17,019	InnoDB	utf8mb4_unicode_ci	1.6 МБ	0 Б

Рисунок 2.6 – Представлення MySQL DB.

2.5 Selenium

Selenium WebDriver: Це основна частина, що надає API для написання скриптів автоматизації для різних браузерів. Ви можете використовувати WebDriver для взаємодії з елементами сторінки, надсилання клавішних комбінацій, заповнення форм та іншого.

У нашому проекті він застосовується в агентах пошуку інформації, для того щоб почати працювати з selenium[10] – потрібно створити web-driver(Рис. 2.7)

```
System.setProperty("webdriver.chrome.driver", "D:\\
ChromeOptions options = new ChromeOptions();
options.addArguments("--headless");

WebDriver driver = new ChromeDriver();
```

Рисунок 2.7 – Створення Chrome драйверу

Це дуже потужний інструмент для автоматичного управління браузером, він дозволяє наприклад здійснювати автоматичний скролінг екрану(Рис. 2.8), та шукати елемент по заданому шляху чи атрибуту(Рис. 2.9).

```
while (System.currentTimeMillis() - startTime < duration) {
    Thread.sleep( millis: 100);
    js.executeScript( s: "window.scrollTo(0, document.body.scrollHeight)");
```

Рисунок 2.8 – Код для автоматичного скролінгу сторінки до низу

```
try{
    element = driver.findElement(By.xpath(xpathExpression));
}catch (NoSuchElementException e){
    element=null;
```

Рисунок 2.9 – Пошук елемента на сторінці за xpath

Для того щоб отримати xpath в нашому проекті я застосовую розбиття на ключові слова в циклі(Рис. 2.10)

```
String xpathExpression = "//a";

for (int i=0;i< words.length;++i) {
    xpathExpression += "[contains(text(),' + words[i] + ')]";
}
```

Рисунок 2.10 – розбиття ключових слів і побудова xpath

2.6 Інструменти та утиліти для скачування файлів та роботи з внутрішнім контентом

jsoup: Java HTML Parser [4]

jsoup — це бібліотека Java для роботи з реальним HTML. Він надає дуже зручний API для отримання URL-адрес, вилучення та маніпулювання даними за допомогою найкращих практик HTML5 DOM та селекторів CSS.

Для прикладу спарсимо вміст тегу <body> силабуса дисципліни “Мова програмування Java”

Для початку в залежностях Maven підключимо бібліотек – рис. 2.11.

```
<dependencies>
  <dependency>
    <groupId>org.jsoup</groupId>
    <artifactId>jsoup</artifactId>
    <version>1.16.1</version>
  </dependency>
</dependencies>
```

Рисунок 2.11 – Підключення Jsoup

Демонстрація прикладу коду(Рис. 2.12a)

```
Connection connection =
Jsoup.connect("https://acts.kpi.ua/syllabuses/uk/SyllabusContent?curriculumId=2376");
connection.ignoreHttpErrors(true);
connection.ignoreContentType(true);
connection.sslSocketFactory(sslContext.getSocketFactory());

var document = connection.get();
var body = document.selectFirst("body");

try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {
    writer.write(body.text());
} catch (Exception e) {
    e.printStackTrace();
}
```

Рисунок 2.12a – Приклад використання Jsoup

За допомогою цього коду можна дістати текст зі сторінки силабусу Київського політехнічного університету ім. Ігоря Сікорського з дисципліни “Мова програмування Java” та записати його наприклад в текстовий файл, або базу даних.

На виході отримуємо текст силабуса - рис. 2.126.

Результат:

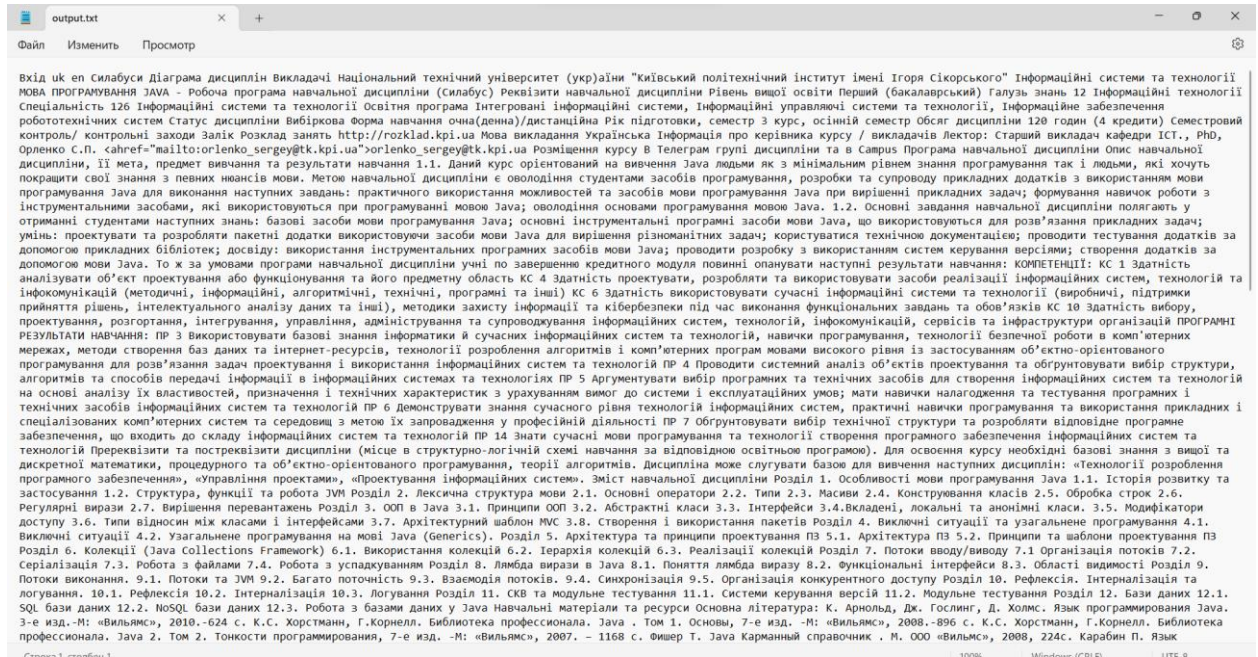


Рисунок 2.126 – Результат використання Jsoup

Apache HttpClient[5]

Apache HttpClient - це бібліотека для виконання HTTP-запитів у Java. Вона дозволяє здійснювати взаємодію з веб-серверами, включаючи відправку запитів і отримання відповідей.

Також отримуємо структуру силабуса із попереднього прикладу. Підключаємо залежності на рис. 2.13.

```
<dependencies>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.13</version>
  </dependency>
</dependencies>
```

Рисунок 2.13 – Підключення HttpClient

Демонстрація прикладу коду(Рис. 2.14а)

```

        HttpGet httpGet = new
HttpGet ("https://acts.kpi.ua/syllabuses/uk/SyllabusContent?curriculumId=2376
");
        HttpResponse response = httpClient.execute(httpGet);

        HttpEntity entity = response.getEntity();
        String responseBody = EntityUtils.toString(entity);

        System.out.println(responseBody);

```

Рисунок 2.14а – Приклад HttpClient

В змінну типу `HttpGet` надаємо `url` до нашого силабуса та через `response` ми виконуємо (`execute`) `http` запит `get`. За допомогою `HttpEntity entity` отримуємо сутність із відповіді, та надаємо їй рядкове представлення і виводимо в консоль, хоча теж могли записати в файл.

Результат на рисунку 2.14б.

```

<head>
<meta charset="utf-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Силабус: Мова програмування Java (ICT)</title>
<meta name="description" content="Силабус: Мова програмування Java (ICT)" />
<meta name="keywords" content="Силабус, Мова програмування Java, ICT" />

<link rel="stylesheet" href="/syllabuses/css/site.min.css?v=71RACm9BYWjzoTK7gEY28vF0r0yKMFzaWNwNICETAeM" />

<!-- Global site tag (gtag.js) - Google Analytics -->
<script async src="https://www.google-analytics.com/gtag/js?id=6-BQZLZNT8TM"></script>
<script>
var app_culture = "uk";
var base_url = "/syllabuses/";
window.dataLayer = window.dataLayer || [];
function gtag() { dataLayer.push(arguments); }
gtag('js', new Date());
gtag('config', '6-BQZLZNT8TM');
</script>
</head>

<body>
<header>
<nav
class="navbar navbar-expand-sm navbar-toggleable-sm navbar-light bg-info border-bottom box-shadow mb-2 py-1">
<div class="container">

```

Рисунок 2.14б – Результат HttpClient

В результаті ми можемо побачити повну `html` структуру сторінки силабуса.

Google Custom Search JSON API[6]

Custom JSON Search API дозволяє розробляти веб-сайти та програми для програмного вилучення та відображення результатів пошуку з програмованої пошукової системи. За допомогою цього API ви можете використовувати

запити RESTful для отримання результатів веб-пошуку або пошуку зображень JSON.

Для роботи з цим API потрібно:

Записати в константи API ключ та сх пошукової системи, створюємо в рядковій змінній наш запит та кодуємо його в UTF-8.

Зчитуємо порядково відповідь браузера та записуємо в змінну response, для наглядності виведемо знайдений результат у текстовий файл.

Apache Tika [7]

Інструментарій Apache Tika™ виявляє і витягує метадані і текст з більш ніж тисячі різних типів файлів (таких як PPT, XLS і PDF). Усі ці типи файлів можна аналізувати через єдиний інтерфейс, що робить Tika корисною для індексації пошуковими системами, аналізу вмісту, перекладу тощо.

Для прикладу спробуємо “витягнути” текст з силабуса в PDF форматі.

Підключаємо залежності – рис. 2.15.

```
<dependencies>
  <dependency>
    <groupId>org.apache.tika</groupId>
    <artifactId>tika-core</artifactId>
    <version>2.9.0</version>
  </dependency>

  <dependency>
    <groupId>org.apache.tika</groupId>
    <artifactId>tika-parsers-standard-package</artifactId>
    <version>2.9.0</version>
  </dependency>
</dependencies>
```

Рисунок 2.15 – Підключення в Maven залежності Apache Tika

Демонстрація прикладу коду Рис(2.16)

```
Parser parser = new AutoDetectParser(new PDFParser());
BodyContentHandler handler = new BodyContentHandler(10 * 1024 * 1024);
Metadata metadata = new Metadata();
ParseContext context = new ParseContext();

metadata.set(Metadata.CONTENT_ENCODING, StandardCharsets.UTF_8.name());
context.set(Parser.class, parser);
parser.parse(stream, handler, metadata, context);

System.out.println("Вміст файлу:");
System.out.println(handler.toString());

System.out.println("Метадані:");
for (String name : metadata.names()) {
    System.out.println(name + ": " + metadata.get(name));
}
stream.close();
```

Рисунок 2.16 – Приклад використання ApacheTika

Створюємо PDF парсер та за допомогою BodyContentHandler встановлюємо розміри вихідної інформації. Створюємо метадату, щоб надати вірний тип кодування, у нашому випадку UTF-8. Включаємо в context наш parser, та викликаємо наші методи передаючи потрібні параметри. Виводимо все в консоль.

Результат показано на рис. 2.17(a-b).

```

Силабус навчальної дисципліни «Програмування мовою Java» для здобувачів вищої освіти ступеня «бакалавр», які навчаються за освітньо-професійною програмою «Цифрові технології дистанційної освіти» спеціальності 015.39 «Професійна освіта (цифрові технології)». Рівень: НУВГП, 2022. 10 с.

ОПП на сайті університету: https://ep3.nuwm.edu.ua/21062/

Розробник силабусу:
Шріль Тетяна Степанівна, кандидат педагогічних наук, доцент кафедри комп'ютерних технологій та економічної кібернетики

Керівник освітньої програми «Цифрові технології дистанційної освіти»:
----- Парфенюк О.В., канд. пед. наук, доцент кафедри комп'ютерних технологій та економічної кібернетики

Силабус схвалений на засіданні кафедри комп'ютерних технологій та економічної кібернетики
Протокол № 4 від "25" листопада 2022 року

Завідувач кафедри комп'ютерних технологій та економічної кібернетики:
----- Гришук П.М., д-р екон. наук, професор

```

Рисунок 2.17(a) – Результати роботи Apache Tika

```

ЗАГАЛЬНА ІНФОРМАЦІЯ*
Ступінь вищої освіти Бакалавр
Освітня програма Цифрові технології дистанційної освіти
Спеціальність 015.39 Професійна освіта (цифрові технології)
Рік навчання, семестр 3

5
Кількість кредитів 5
Лекції: 24 год.
Лабораторні заняття: 30 год.
Самостійна робота: 96 год.
Курсова робота: Ні
Форма навчання Денна
Форма підсумкового контролю Залік
Мова викладання Українська

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА*
ПРОФІЛ ЛЕКТОРА
Лектор Шріль Тетяна Степанівна, кандидат педагогічних наук,
доцент, доцент кафедри комп'ютерних технологій та економічної кібернетики
Вікісайт http://wiki.nuwm.edu.ua/index.php/Шріль\_Тетяна\_Степанівна
ORCID https://orcid.org/0000-0002-8694-631X
Як комунікувати t.s.shrol@nuwm.edu.ua

Актуальні оголошення на сторінці дисципліни в системі MOODLE
ПРО ДИСЦИПЛІНУ

```

Рисунок 2.17(b) – Результати роботи Apache Tika

3 ІНФОРМАЦІЙНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ

3.1 Формування вхідних даних

Формування вхідних даних має на меті забезпечити системі важливу інформацію для подальшого аналізу та обробки. У цьому випадку, ми фільтруємо дані за країною, обираючи два університети та приблизну назву дисципліни, яка може бути важливою для оцінки або порівняння студентів.

Наприклад:

- Країни
- Університет 1
- Університет 2
- Приблизна назва дисципліни

Ця інформація допоможе нам у дальшому аналізі та порівнянні даних стосовно студентів, що навчаються в цих університетах за вказаною дисципліною.

3.2 Програмна реалізація

3.2.1 Maven dependency

Як збірник проекту ми використовуємо Maven, і щоб наш додаток працював потрібно підключити необхідні залежності у pom.xml файл:

Перша залежність це spring-webmvc – вона потрібна для реалізації веб додатків за допомогою патерну MVC.

```
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-webmvc</artifactId>
  <version>5.2.9.RELEASE</version>
</dependency>
```

Друга залежність це jstl – вона дозволяє створювати JSP сторінки в яких можна використовувати Java код, в результаті ми отримуємо динамічну HTML сторінку.

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>jstl</artifactId>
  <version>1.2</version>
</dependency>
```

Для роботи з базою даних нам знадобиться одразу три залежності. По-перше ви як ORM використовуємо Hibernate, а за базу даних у нас відповідає MySQL. Також ми підключаємо C3P0 пул щоб автоматично закривати сесії підключення до DB.

```
<dependency>
  <groupId>org.hibernate</groupId>
  <artifactId>hibernate-core</artifactId>
  <version>5.4.22.Final</version>
</dependency>

<dependency>
  <groupId>com.mysql</groupId>
  <artifactId>mysql-connector-j</artifactId>
  <version>8.3.0</version>
</dependency>

<dependency>
  <groupId>com.mchange</groupId>
  <artifactId>c3p0</artifactId>
  <version>0.9.5.2</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-orm</artifactId>
  <version>5.2.9.RELEASE</version>
</dependency>
```

Також нам знадобиться керування запитами та відповідями в наших контролерах, за це буде відповідати залежність servlet-api:

```
<dependency>
  <groupId>javax.servlet</groupId>
  <artifactId>servlet-api</artifactId>
  <version>2.5</version>
  <scope>provided</scope>
</dependency>
```

Для скрізної реалізації логування ми будемо використовувати AOP, для цього потрібно додати aspectjweaver:

```
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjweaver</artifactId>
  <version>1.9.6</version>
</dependency>
```

3.2.2 Налаштування Entity

У базі даних знаходяться список всіх закладів вищої освіти, та список всіх країн. Для того щоб зв'язати клас Country з таблицею в базі, ми

прописуємо анотацію `@Entity` та `@Table(name="countries")`. Повний опис класу виглядає так:

```
@Entity
@Table(name = "countries")
public class Country {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;
    @Column(name = "iso2")
    private String iso2;
    @Column(name = "iso3")
    private String iso3;

    @Column(name = "name")
    private String name;

    @OneToMany(mappedBy = "country", cascade = CascadeType.ALL, fetch =
FetchType.EAGER)
    private List<University> universities;
}
```

Анотація `@Id` позначає Primary key в таблиці країн, а `@Column` позначає її ім'я, таким чином ми зв'язуємо поля нашого класу з полями в таблиці бази даних. Також у класі `Country` є поле що приймає список університетів і реалізую відносини один до багатьох, адже у одного міста може бути багато університетів, про що й свідчить анотація `@OneToMany`. Також в класі присутні гетери та сетери на кожне поле та конструктор з параметрами і без.

Другий Entity, який зв'язано з базою це `University`:

```
@Entity
@Table(name = "webometric_universities")
public class University {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @ManyToOne(cascade = CascadeType.ALL)
    @JoinColumn(name = "country_id")
    private Country country;

    @Column(name = "name")
    private String name;

    @Column(name = "url")
    private String url;
}
```

Тут все аналогічно до Country тільки немає зворотного зв'язку ManyToOne.

3.2.3 Налаштування конфігураційних файлів

Ми повинні сконфігурувати xml файл, для роботи з бінами, які контролюють всі процеси фреймворка Spring, налаштовуємо applicationContext.xml

В цьому файлі ми використовуємо анотації для сканування, включаємо обробку анотацій для веб-контролерів, автоматично створюються проксі для аспектів, корекція JSP-сторінок, також конфігурується джерело даних – доступ до бази MySQL. Налаштовується сесійна фабрика Hibernate та менеджер транзакцій.

Налаштування web.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://xmlns.jcp.org/xml/ns/javaee"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd"
  id="WebApp_ID" version="3.1">

  <display-name>multi-agent-system</display-name>

  <absolute-ordering />

  <servlet>
    <servlet-name>dispatcher</servlet-name>
    <servlet-
class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param>
      <param-name>contextConfigLocation</param-name>
      <param-value>/WEB-INF/applicationContext.xml</param-value>
    </init-param>
    <load-on-startup>1</load-on-startup>
  </servlet>

  <servlet-mapping>
    <servlet-name>dispatcher</servlet-name>
    <url-pattern>/</url-pattern>
  </servlet-mapping>
</web-app>
```

В цьому файлі ми визначаємо DispatcherServlet, вказуємо місце розташування файлу applicationContext.xml. Вказуємо відповідність сервлета як /.

3.2.4 DAO та Service

Для зручної роботи з базою даних ми використаємо такий підхід як DAO (Data Access Object) – це допоміжний компонент, відповідальний за роботу з БД. Для цього ми спочатку реалізуємо інтерфейси CountryDAO та UniversityDAO.

CountryDAO:

```
public interface CountryDAO {
    public List<Country> getAllCountries();
    public Country getCountry(int id);
}
```

Цей інтерфейс пропонує нам два методи, один для отримання списку всіх країн, а інший для отримання країни по id.

UniversityDAO:

```
public interface UniversityDAO {
    public List<University> getAllUniversity();
}
```

В цьому інтерфейсі є метод для отримання списку всіх університетів.

Далі потрібно створити класи які будуть реалізувати ці інтерфейси, CountryDAOImpl та UniversityDAOImpl відповідно.

CountryDAOImpl:

```
@Repository
public class CountryDAOImpl implements CountryDAO{
    @Autowired
    private SessionFactory sessionFactory;
    @Override
    public List<Country> getAllCountries() {
        Session session = sessionFactory.getCurrentSession();
        List<Country> countries = session.createQuery("FROM Country",
Country.class).getResultList();
        return countries;
    }

    @Override
    public Country getCountry(int id) {
        Session session = sessionFactory.getCurrentSession();
        Country country = session.get(Country.class, id);
        return country;
    }
}
```

За допомогою анотації @Repository ми вказуємо на те що ця сутність має зв'язок з базою даних.

UniversityDAOImpl:

```
@Repository
public class UniversityDAOImpl implements UniversityDAO{
    @Autowired
    private SessionFactory sessionFactory;
    @Override
    public List<University> getAllUniversity() {
        Session session = sessionFactory.getCurrentSession();
        List<University> universities = session.createQuery("FROM
University", University.class).getResultList();
        return universities;
    }
}
```

Аналогічно до CountryDAOImpl налаштований і UniversityDAOImpl.

Також ми в нашому проекті застосовуємо сервіси, для цього потрібно створити інтерфейс який реалізує певні методи і імплементувати його в клас сервісу, який ми позначаємо анотацією як @Service наприклад Рис. 3.1

```
import java.util.List;
@Service
public class CountryServiceImpl implements CountryService{
    @Autowired
    CountryDAO countryDAO;
    1 usage
    @Override
    @Transactional
    public List<Country> getAllCountries() {
        return countryDAO.getAllCountries();
    }
}
```

Рисунок 3.1 – приклад сервісу для країн

Ми визначаємо змінну типу CountryDAO і звертаємося до цього об'єкту в методі getAllCountries.

3.2.5 MasController

Головний контролер у проекті керує URL шляхами для відображення JSP сторінок з генерованим на сервері вмістом. Контролер це компонент з анотацією @Controller, якому надсилає запити DispatcherServlet, і в залежності від оброблених даних на сервері ми повертаємо динамічно згенеровану сторінку.

MasController має два поля(Рис. 3.2) які позначені анотаціями @Autowired – це означає що вони будуть внесені автоматично при старті програми із контексту.

```
@Autowired
private UniversityService universityService;

@Autowired
private CountryService countryService;
```

Рисунок 3.2 – Поля контроллера

В контроллері описано три шляхи за допомогою анотацій @GetMapping та @RequestMapping. Для прикладу (Рис. 3.3) За кореневим шляхом проекту “/” ми показуємо в браузері головну сторінку де можна обрати країни для фільтрації університетів, це значно спрощує пошук.

```
@GetMapping("/")
public String showAllUniversities(Model model){
    List<Country> allCountries = countryService.getAllCountries();
    model.addAttribute("allCountries", allCountries);
    return "select-countries";
}
```

Рисунок 3.3 – Метод передачі всіх країн у модель

У MasController є не тільки методи зі шляхами, також є допоміжні методи: для завантаження html файлу(Рис. 3.4), для завантаження файлів інших форматів(Рис. 3.5), так як наша система багатоагентна потрібно врахувати різні джерела, які представляють силабуси, також є методи для отримання заголовків файлу, для порівняння розширень та контенту(Рис. 3.6), надається метод який отримує весь контент з файлу(Рис. 3.7).

```
public static void downloadHtml(String url, String saveFilePath, int num) throws IOException {
    Document doc = Jsoup.connect(url).get();
    try (BufferedWriter writer = new BufferedWriter(new FileWriter(fileName: saveFilePath+"\\syllabus"+num+".html"))) {
        writer.write(doc.html());
    }
    System.out.println("HTML-page successfully downloaded and saved: syllabus"+num+".html");
}
```

Рисунок 3.4 – Метод для завантаження HTML сторінок

Цей метод використовує бібліотеку Jsoup для отримувати вміст потрібних сторінок і записує їх у файл.

```
public static void downloadFile(String fileURL, String saveDir, int num, String ext) throws IOException {
    URL url = new URL(fileURL);
    HttpURLConnection httpConn = (HttpURLConnection) url.openConnection();
    int responseCode = httpConn.getResponseCode();

    if (responseCode == HttpURLConnection.HTTP_OK) {

        InputStream inputStream = httpConn.getInputStream();
        String fileName = "syllabus"+num+ext;

        String saveFilePath = saveDir + File.separator + fileName;
        FileOutputStream outputStream = new FileOutputStream(saveFilePath);
        int bytesRead;
        byte[] buffer = new byte[4096];
        while ((bytesRead = inputStream.read(buffer)) != -1) {
            outputStream.write(buffer, 0, bytesRead);
        }
        outputStream.close();
        inputStream.close();
        System.out.println("File successfully downloaded and saved: " + fileName);
    } else {
        System.out.println("Error. Code response: " + responseCode);
    }
    httpConn.disconnect();
}
```

Рисунок 3.5 – Завантажує файли інших форматів

Так як наша система працює з різним форматом файлів, було створено метод який міг би викачувати більшість видів файлів.

А для порівняння того якого формату файл ми скачуємо ви створили метод, який повертає заголовки файлів за допомогою HTTP методу HEAD:

```
public static String getContentType(String url) throws IOException {
    HttpURLConnection connection = (HttpURLConnection) new URL(url).openConnection();
    connection.setRequestMethod("HEAD");
    connection.connect();
    String contentType = connection.getContentType();
    connection.disconnect();
    return contentType;
}
```

Рисунок 3.6 – Метод для отримання заголовків файлу

Після того як файл скачаний ми маємо тримати з нього текстові дані, щоб відобразити їх на нашій JSP сторінці результатів, для цього ми використовуємо бібліотеку Apache Tika і створюємо парсер, який працює з документами різних типів, у нашому випадку найпоширенішими є PDF, HTML та DOCX.

```

public static String getContent(File file) throws TikaException, IOException, SAXException {

    Parser parser = new AutoDetectParser();
    BodyContentHandler handler = new BodyContentHandler( writeLimit: -1);
    Metadata metadata = new Metadata();
    ParseContext context = new ParseContext();
    InputStream stream = new FileInputStream(file);

    metadata.set(Metadata.CONTENT_ENCODING, StandardCharsets.UTF_8.name());

    parser.parse(stream, handler, metadata, context);

    String content = handler.toString();

    return content;
}

```

Рисунок 3.7 – Отримання контенту з файлу

3.2.6 Агент СумДУ

Всі агенти реалізують інтерфейс SyllabusAgent, який має один метод (Рис. 3.8)

```

package com.serhiikuzmenko.spring.mas.agents;

2 usages 2 implementations
public interface SyllabusAgent {
    4 usages 2 implementations
    public String linkFinder(String subject, String uni);
}

```

Рисунок 3.8 – Інтерфейс Агенту.

Для того щоб знайти потрібний елемент, агент СумДУ використовує інтернет сторінку з розкладом занять де знаходяться всі силабуси, через те що вони підвантажуються динамічно ми використовуємо JavascriptExecutor, який дозволяє прокручувати та вантажити сторінку до низу.

Ми використовуємо обмежений час, щоб цикл не працював нескінченно, максимально дається 30 секунд на пошук, реалізацію циклу можна побачити на рис. 3.9.

```

public String LinkFinder(String desiredSubject, String uni) {
    String link;
    String []words = desiredSubject.split(regex: "[\\p{Punct}\\s]");
    String xpathExpression = "//a";
    for (int i=0;i< words.length;++i) {
        xpathExpression += "[contains(text(), " + words[i] + ")]";
    }
    System.setProperty("webdriver.chrome.driver", "D:\\USER\\JAVA\\sources\\ua\\edu\\sumdu\\j2se\\Multi-agent-system-data-collection\\chromedriver.exe");
    ChromeOptions options = new ChromeOptions();
    options.addArguments("--headless");
    WebDriver driver = new ChromeDriver();
    try {
        String url = "https://pg.cabinet.sumdu.edu.ua/catalog";
        driver.get(url);
        JavascriptExecutor js = (JavascriptExecutor) driver;
        WebDriverWait wait = new WebDriverWait(driver, timeOutInSeconds: 10);
        WebElement element = null;

        long startTime = System.currentTimeMillis();
        long duration = 30000;
        Thread.sleep( millis: 500);
        while (System.currentTimeMillis() - startTime < duration) {
            Thread.sleep( millis: 100);
            js.executeScript( "window.scrollTo(0, document.body.scrollHeight)");

            try{
                element = driver.findElement(By.xpath(xpathExpression));
            }catch (NoSuchElementException e){
                element=null;
            }finally {
                if(element!=null){
                    break;
                }
            }
        }
    }
}

```

Рисунок 3.9 – Агент СумДУ

3.2.7 Агент Google

Для пошуку силабусів з інших університетів, ми будемо використовувати Google.

Цей агент за замовчуванням реалізує пошук в англomовному середовищі, але якщо це ЗВО України, то мова пошуку змінюється (Рис. 3.10)

```

public class GoogleAgent implements SyllabusAgent {
    2 usages
    private String lang = "+lang:en";
    2 usages
    public void setLang(String lang) {
        this.lang = lang;
    }
}

```

Рисунок 3.10 – Встановлення мови

Основна логіка будується так що в браузер вводиться пошуковий запит, який складається з назви університету та назви дисципліни, потім проводиться пошук елементів на сторінці які мають певний зміст, та дістаються з цих блоків посилання на силабуси. Реалізацію можна побачити на рис. 3.11.

```

try {
    String url = "https://www.google.com/search?q=";

    if(this.lang.equals("+lang:ua")){
        driver.get(url+uni.replace(target: " ", replacement: "+")+синабус+searchedSubject.replace(target: " ", replacement: "+")+lang);
    }else{
        driver.get(url+uni.replace(target: " ", replacement: "+")+syllabus+searchedSubject.replace(target: " ", replacement: "+")+lang);
    }
}

JavascriptExecutor js = (JavascriptExecutor) driver;
Thread.sleep( millis, 2000);
WebElement element = null;
//FIX ME!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
try{
    if(this.lang.equals("+lang:ua")){
        element = driver.findElement(By.xpath( xpathExpression: "//a[./h3[contains(translate(text(), 'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЪЫЭЮЯ', 'абвгдежзийклмнопрстуфхцчщьяэя'),
    }else{
        element = driver.findElement(By.xpath( xpathExpression: "//a[contains(translate(., 'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'), 'syllabus') and not(ance
    }
} catch (NoSuchElementException e){
}finally {
    if(element==null){
        return "Not found";
    }else{
        link = element.getAttribute( s: "href");
        return link;
    }
}
} catch (InterruptedException e) {
    throw new RuntimeException(e);
} finally {
    driver.quit();
}

```

Рисунок 3.11 – Реалізація Google агенту

Таким чином, якщо є знайдене посилання на сторінці, цей метод, який перевизначений з інтерфейсу SyllabusAgent повертає посилання.

3.3 Аналіз результатів

На головній сторінці сайту ми бачимо форму(Рис 3.10) з двома випадючими списками, це допомагає нам відфільтрувати університети різних країн, якщо пропустити цей етап, то на порівнянні університетів ми будемо обирати зі списку усього світу, що не є зручним, тому фільтрація по містам є досить влучним рішенням.

Multi Agent System Data Collection

System of academic mobility

Country 1

Ukraine ▼

Country 2

Ukraine ▼

Select counties

Rights reserved © 2024. The application was developed by Serhii Kuzmenko

Рисунок 3.10 – Фільтрація за містом

Після обраних міст, нас контроллер перенаправляє на сторінку вибору університетів та дисципліни рис. 3.11.

Change countries
Multi Agent System Data Collection

System of academic mobility

University 1

University 2

Subject Field

Rights reserved © 2024. The application was developed by Serhiy Kuzmenko

Рисунок 3.11 – Етап вибору університетів та дисципліни

Після пошуку ми отримуємо результуючий view у вигляді JSP сторінки рис. 3.12 на якій поділено навпіл сторінку web-додатку.

Multi Agent System Data Collection

<p>First Uni: Sumy State University (Ukraine)</p> <p>Link to original</p> <p>СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ 1. Загальна інформація про навчальну дисципліну Повна назва навчальної дисципліни Бази даних та інформаційні системи Повна офіційна назва закладу вищої освіти Суумський державний університет Повна назва структурного підрозділу Факультет електроніки та інформаційних технологій Кафедра комп'ютерних наук Розробник(и) Кузюков Борис Олександр Рівень вищої освіти Перший рівень вищої освіти, НРК – 6 рівень, QF-LLL – 6 рівень, FQ-EHEA – перший цикл Семестр вивчення навчальної дисципліни 16 тижнів протягом 5-го семестру, 16 тижнів протягом 6-го семестру Обсяг навчальної дисципліни Обсяг становить 10 кредитів ЕКТС: 300 год., з яких 1 кред. ЕКТС: 30 год. становить курсова робота Для денної форми навчання 112 год. становить контактна робота з викладачем (40 год. лекцій, 72 год. лабораторних занять), 188 год. становить самостійна робота Мова викладання Українська 2. Місце навчальної дисципліни в освітній програмі Статус дисципліни Обов'язкова навчальна дисципліна для освітньої програми "Інформатика" Передумови для вивчення дисципліни Програмування під платформу. Net Framework, Алгоритми і структури даних Додаткові умови Додаткові умови відсутні Обмеження відсутні 3. Мета навчальної дисципліни Метою навчальної дисципліни є формування у майбутніх фахівців здатностей проєктувати бази даних і розробляти прикладне програмне забезпечення на базі систем керування базами даних, використовувати бази даних на основі конкретних бізнес-вимог до інформаційної системи, поширюючи розгляд теми на питання безпеки, визначати межі застосування технологій, використовувати нереляційні бази даних. 4. Зміст навчальної дисципліни Модуль 1. Моделювання БД та ІС. Тема 1 Моделювання інформаційної системи. Визначення вимог до ІС. Концептуальна модель ІС. Моделювання потоків даних з застосуванням DFD (0,1-рівень). Моделювання структури бази даних. Використання ер-діаграм для моделювання структури даних. Функциональні залежності. Нормальні форми. Нормалізація структури бази даних. Перетворення ер-діаграм в відношення. Функціональні залежності. Алгоритми редукції Кодда. Таблична модель даних. Відображення ер-діаграм у табличну модель даних. Методи забезпечення цілісності даних.</p>	<p>Second Uni: National Technical University of Ukraine Kyiv Polytechnic Institute (Ukraine)</p> <p>Link to original</p> <p>Вхід як в Силабусі (Діаграма дисциплін Викладачі Національний технічний університет (україні) "Київський політехнічний інститут імені Ігоря Сікорського" Інформаційні системи та технології БАЗИ ДАНИХ - Робоча програма навчальної дисципліни (Силабус) Реферати навчальної дисципліни Рівень вищої освіти Перший (бакалаврський) Галузь знань 12 Інформаційні технології Спеціальність 121 Інженерія програмного забезпечення Освітня програма Інженерія програмного забезпечення (суми) Іютерних систем Статус дисципліни Нормативна форма навчання очна/денна/зочна/дистанційна Рік підготовки семестр 2 курс, осінній семестр Обсяг дисципліни 120 годин (36 годин – Лекції, 18 годин – Лабораторні, 66 годин – СРС) Семестровий контроль/ контрольні заходи Записувалькова робота Розклад занять http://rozklad.kpi.ua/Schedules/ScheduleGroupSelecton.aspx Мова викладання Українська Інформація про керівника курсу / викладача Лектор: к.т.н., Ульяновська Ксенія Олександрівна k.ulianovska@kpi.ua моб. +38(050)355-62-15 Лабораторні: к.т.н., доцент, Бойко Олександра Володимирівна alexbb1412016@gmail.com моб. +38(097)294-40-83 Розміщення курсу https://sapiros.kpi.ua/ Програма навчальної дисципліни Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання Опис дисципліни При проходженні даної дисципліни, студенти розійомляться з поняттям «бази даних», видами баз даних, та детальноше вивчатимуть реляційну модель даних. На лабораторних заняттях опанують етапи проєктування реляційних баз даних. На риві користувач з досвідом опанують систему управління базами даних Oracle Database 11g/18c. з курсу передачі/введення контрольні якості отриманих знань у вигляді експрес-контрольних та модульних контрольних робіт. На лекціях проводитимуться практичні групові заняття по опануванню принципів побудови реляційної моделі даних. Предмет навчальної дисципліни Основні поняття реляційної теорії, введення в реляційні бази даних, інфологічна модель, датовлогічна модель, реляційні системи управління базами даних SQL на основі Oracle Databases. 4.Дисциплінарні завдання Дисципліна Бази даних базуються на дисциплінах: Операційні системи Програмування – 1. Основні програмування Програмування – 2. Структури даних та алгоритми Спеціальні розділи математики-1. Дискретна математика Мета навчальної дисципліни Підготувати висококваліфікованих фахівців, які володіють основними поняттями реляційної теорії, термінами реляційних баз даних, моделюванням предметної області для побудови реляційних баз даних, нормалізацією, алгоритмом побудови реляційних баз даних, фізичною реалізацією спроектованих схем баз даних на реляційних системах управління базами даних, адмініструванням реляційних систем управління базами даних на рівні користувача з досвідом а саме Oracle Databases. Основні завдання навчальної дисципліни Знання ролі та місця реляційних баз даних в інформаційно-технологічному середовищі їх використання, реляційної моделі побудови реляційних баз даних, етапів проєктування реляційних баз даних, нормальних форм, мови SQL (Structured Query Language) для створення схеми баз даних та роботи з даними (запити на вибірку, модифікацію, оновлення, видалення). Уміння аналізувати предметну область, визначену рамками певного проєкту, будувати концептуальну модель даних по результатам аналізу предметної області, перетворювати концептуальну модель даних в датовлогічну модель фізичної структури бази даних, на основі датовлогічної моделі здійснювати фізичну реалізацію бази даних – розробляти скрипти та адресувати їх на вибраній системі управління базами даних (СУБД); володіти мовою SQL в її роздаках DDL (Data Definition Language) та однократичні запити конструкції SELECT, адмініструвати на рівні користувача з досвідом реляційну СУБД Oracle Database 11g/18c.</p>
---	--

Рисунок 3.12 – Результуюча сторінка

У разі не знаходження одного або двох силабусів буде виведена відповідна сторінка (Рис. 3.13)

Multi Agent System Data Collection

<p>First Uni: American Samoa Community College (American Samoa)</p> <p>Link to original</p> <p>No information</p>	<p>Second Uni: Ama International University of Bahrain (American Samoa)</p> <p>Link to original</p> <p>No information</p>
--	--

Rights reserved © 2024. The application was developed by Serhiy Kuzmenko

Рисунок 3.13 – Приклад коли силабусів не знайдено

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було виконано такі завдання:

Результатом аналізу предметної області є такі сервіси як: Times Higher Education, QS World University Rankings та Academic Ranking of World Universities. На основі аналізу було сформовано вимоги до додатку, що розробляється.

На основі огляду технології серед наявних варіантів обрані Selenium, Apache Tika, Spring MVC, Hibernate, MySQL, Jsoup. Ці технології допомагають гнучко працювати з контентом у будь-якому представленні. При проектуванні та розробці використано патерн MVC в імплементації фреймворку Spring.

В ході виконання кваліфікаційної роботи було розроблено web-додаток для багатоагентного збору даних що до змісту навчальних дисциплін, головна мета якого за вхідними даними знаходити у відкритому доступі силабуси дисциплін і представляти їх зміст для порівняння.

По завершенню розробки додаток було протестовано, були надані результати у вигляді знімків екрану.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Times Higher Education home | Times Higher Education (THE). [Електронний ресурс]. Режим доступу: <https://www.timeshighereducation.com/> [Дата звернення: 14.11.2023].
2. QS World University Rankings: Top global universities | Top Universities. [Електронний ресурс]. Режим доступу: <https://www.topuniversities.com/university-rankings/world-university-rankings> [Дата звернення: 14.11.2023].
3. ShanghaiRanking's Academic Ranking of World Universities. [Електронний ресурс]. Режим доступу: <http://www.shanghairanking.com/> [Дата звернення: 14.11.2023].
4. jsoup: Java HTML parser, built for HTML. editing, cleaning, scraping, and XSS safety. [Електронний ресурс]. Режим доступу: <https://jsoup.org/> [Дата звернення: 14.11.2023].
5. Apache HttpComponents-HttpClient Overview. [Електронний ресурс]. Режим доступу: <https://hc.apache.org/httpcomponents-client-4.5.x/> [Дата звернення: 14.11.2023].
6. JSON API користувачького пошуку | Programmable Search Engine | Google for Developers. [Електронний ресурс]. Режим доступу: <https://developers.google.com/custom-search/v1/overview> [Дата звернення: 14.11.2023].
7. Apache Tika Apache Tika. [Електронний ресурс]. Режим доступу: <https://tika.apache.org/> [Дата звернення 14.11.2023]
8. MVC Framework Introduction Режим доступу: <https://www.geeksforgeeks.org/mvc-framework-introduction/> [Дата звернення 01.11.2024]
9. Spring Framework. [Електронний ресурс]. Режим доступу: <https://spring.io/projects/spring-framework> [Дата звернення 11.04.2024]
10. Selenium automates browsers. [Електронний ресурс]. Режим доступу: <https://www.selenium.dev/> [Дата звернення 01.05.2024]

11. Hibernate Everything data. [Електронний ресурс]. Режим доступу: <https://hibernate.org/> [Дата звернення 05.04.2024]
12. Apache Tomcat. [Електронний ресурс]. Режим доступу: <https://tomcat.apache.org/> [Дата звернення 05.04.2024]
13. Мультиагентні інтелектуальні інформаційні системи. [Електронний ресурс]. Режим доступу: <http://surl.li/togls> [Дата звернення 04.05.2024]
14. Конспект лекцій з дисципліни «Мультиагентні інтелектуальні інформаційні системи, ч.1» для усіх форм навчання спеціальності 122 Комп'ютерні науки [Електронний ресурс] / укл. Н.О. Маслова. – Луцьк : ДЗВО ДонНТУ, 2023. – 60 с. [Дата звернення 01.05.2024]
15. DIGIUNI [Електронний ресурс]. Режим доступу: <https://mu.edu.ua/digiuni> [Дата звернення 04.05.2024]
16. Матвеев О. М. Багатоагентні моделі та методи збору та видобування бізнес-інформації у веб-просторі. Видавець: Національний технічний інститут "Харківський політехнічний інститут" 2021. [Дата звернення 07.05.2024]
17. Кузьмініх В. О. Збір та первинна обробка даних наукової діяльності з різномірних джерел. Видавець: КПІ ім. Ігоря Сікорського 2024. [Дата звернення 07.05.2024]
18. Лаушкін В. О. Багатоагентна система дослідження простору на основі технології інтернету речей. Видавець: КПІ ім. Ігоря Сікорського 2019. [Дата звернення 06.05.2024]
19. Зюзько А. О. Модель системи електронного навчання на основі багатоагентного підходу. [Електронний ресурс]. Режим доступу: <https://openarchive.nure.ua/server/api/core/bitstreams/e313a60b-cc28-48fe-8e64-387fb133ffa2/content> [Дата звернення 06.05.2024]
20. Янголенко О.В., Матвеев О., Острогляд В.В., Корсун І.А. ЗБІР ТА ВИДОБУВАННЯ ІНФОРМАЦІЇ ШЛЯХОМ ПОБУДОВИ МУЛЬТИАГЕНТНОЇ СИСТЕМИ. [Електронний ресурс]. Режим доступу: <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/3e7c414a-9bd2-4239-aca2-1516b17b73ca/content> [Дата звернення 07.05.2024]

ДОДАТОК

MasController.java:

```
package com.serhiikuzmenko.spring.mas.controller;

import com.serhiikuzmenko.spring.mas.agents.GoogleAgent;
import com.serhiikuzmenko.spring.mas.agents.SumyStateUniversityAgent;
import com.serhiikuzmenko.spring.mas.entity.Country;
import com.serhiikuzmenko.spring.mas.entity.University;
import com.serhiikuzmenko.spring.mas.service.CountryService;
import com.serhiikuzmenko.spring.mas.service.UniversityService;

import org.apache.tika.exception.TikaException;
import org.apache.tika.metadata.Metadata;
import org.apache.tika.parser.AutoDetectParser;
import org.apache.tika.parser.ParseContext;
import org.apache.tika.parser.Parser;
import org.apache.tika.sax.BodyContentHandler;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.xml.sax.SAXException;

import javax.net.ssl.*;
import javax.servlet.http.HttpServletRequest;
```

```
import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.security.SecureRandom;
import java.security.cert.X509Certificate;
import java.util.List;
```

```
@Controller
```

```
public class MasController {
```

```
    @Autowired
```

```
    private UniversityService universityService;
```

```
    @Autowired
```

```
    private CountryService countryService;
```

```
    @GetMapping("/")
```

```
    public String showAllUniversities(Model model){
```

```
        List<Country> allCountries = countryService.getAllCountries();
```

```
        model.addAttribute("allCountries", allCountries);
```

```
        return "select-countries";
```

```
    }
```

```
    @RequestMapping("/selectCountries")
```

```
    public String selectCountries(HttpServletRequest request, Model model){
```

```
        String county1Param = request.getParameter("country1");
```

```
        String county2Param = request.getParameter("country2");
```

```

List<University> localUniversities1=null;
List<University> localUniversities2=null;

if(county1Param!=null){
    int c1 = Integer.parseInt(request.getParameter("country1"));
    localUniversities1 = countryService.getCountry(c1).getUniversities();
}else{
    localUniversities1 = universityService.getAllUniversity();
}

if(county2Param!=null){
    int c2 = Integer.parseInt(request.getParameter("country2"));
    localUniversities2 = countryService.getCountry(c2).getUniversities();
}else{
    localUniversities2 = universityService.getAllUniversity();
}

model.addAttribute("localUniversities1", localUniversities1);
model.addAttribute("localUniversities2", localUniversities2);

return "universities";
}

@RequestMapping("/handleSyllabus")
public String handleSyllabus(HttpServletRequest request, Model model){
    String firstUni = request.getParameter("university1");
    String secondUni = request.getParameter("university2");
    String desiredSubject = request.getParameter("subjectField");

    String firstURL, secondURL;

```

```
model.addAttribute("firstUniName", firstUni);
model.addAttribute("secondUniName", secondUni);
model.addAttribute("firstCounty",
universityService.getCountry(firstUni).getName());
model.addAttribute("secondCountry",
universityService.getCountry(secondUni).getName());
```

```
if(firstUni.equals("Sumy State University")){
    SumyStateUniversityAgent sumyStateUniversityAgent = new
SumyStateUniversityAgent();
    firstURL = sumyStateUniversityAgent.linkFounder(desiredSubject,
firstUni);
}else{
    GoogleAgent googleAgent = new GoogleAgent();
    if(universityService.getCountry(firstUni).getName().equals("Ukraine")){
        googleAgent.setLang("+lang:ua");
    }
    firstURL = googleAgent.linkFounder(desiredSubject, firstUni);
}
```

```
if(secondUni.equals("Sumy State University")){
    SumyStateUniversityAgent sumyStateUniversityAgent = new
SumyStateUniversityAgent();
    secondURL = sumyStateUniversityAgent.linkFounder(desiredSubject,
secondUni);
}else{
```



```

    GoogleAgent googleAgent = new GoogleAgent();

    if(universityService.getCountry(secondUni).getName().equals("Ukraine")){
        googleAgent.setLang("+lang:ua");
        System.out.println(googleAgent.getLang());
    }
    secondURL = googleAgent.linkFounder(desiredSubject, secondUni);
}

System.out.println(firstURL);
System.out.println(secondURL);
final String saveUrl =
"D:\\USER\\JAVA\\sources\\ua\\edu\\sumdu\\j2se\\Multi-agent-system-data-
collection\\src\\main\\webapp\\syllabuses";

File syl1 = null;
File syl2 = null;
String content1 = "";
String content2 = "";
try {

    if(!firstURL.equals("Not found")){

if(getContentType(firstURL).equals("text/html")||getContentType(firstURL).equals
("text/html; charset=utf-8")){
        System.out.println("QWERQWERQWER");
        downloadHtml(firstURL, saveUrl, 1);
        syl1 = new File(saveUrl+"\\syllabus1.html");
    }else if(getContentType(firstURL).equals("application/pdf")){
        downloadFile(firstURL, saveUrl, 1, ".pdf");
    }
}
}

```

```

        syl1 = new File(saveUrl+"\\syllabus1.pdf");
    }else
if(getContentType(firstURL).equals("application/vnd.openxmlformats-
officedocument.wordprocessingml.document")){
        downloadFile(firstURL, saveUrl, 1, ".docx");
        syl1 = new File(saveUrl+"\\syllabus1.docx");
    }else if(getContentType(firstURL).equals("application/msword")){
        downloadFile(firstURL, saveUrl, 1, ".doc");
        syl1 = new File(saveUrl+"\\syllabus1.doc");
    }else{
        content1="No information";
    }

    if(syl1!=null){
        try{
            content1 = getContent(syl1).trim().replaceAll("\\s+", "
").replaceAll("\\. ", "<br>");
        }catch (TikaException | IOException | SAXException e){
            e.printStackTrace();
        }
    }

    model.addAttribute("firstUni", content1);
    if(!content1.equals("No information")){
        model.addAttribute("firstOrigURL", firstURL);
    }
}else{
    model.addAttribute("firstUni", "No information");
}

```

```

if(!secondURL.equals("Not found")){
    System.out.println(getContentType(secondURL));
    if
(getContentType(secondURL).equals("text/html")||getContentType(secondURL).eq
uals("text/html; charset=utf-8")){
        downloadHtml(secondURL, saveUrl, 2);
        syl2 = new File(saveUrl+"\\syllabus2.html");
    }else if(getContentType(secondURL).equals("application/pdf")){
        downloadFile(secondURL, saveUrl, 2, ".pdf");
        syl2 = new File(saveUrl+"\\syllabus2.pdf");
    }else
if(getContentType(firstURL).equals("application/vnd.openxmlformats-
officedocument.wordprocessingml.document")){
        downloadFile(secondURL, saveUrl, 2, ".docx");
        syl2 = new File(saveUrl+"\\syllabus2.pdf");
    }else if(getContentType(firstURL).equals("application/msword")){
        downloadFile(secondURL, saveUrl, 2, ".doc");
        syl2 = new File(saveUrl+"\\syllabus2.doc");
    }else {
        content2 = "No information";
    }

if(syl2!=null){
    try{
        content2 = getContent(syl2).trim().replaceAll("\\s+", "
").replaceAll("\\. ", ".<br>");
    }catch (TikaException | IOException | SAXException e){
        e.printStackTrace();
    }
}

```

```

    }

    model.addAttribute("secondUni", content2);
    if(!content2.equals("No information")){
        model.addAttribute("secondOrigURL", secondURL);
    }
}else{
    model.addAttribute("secondUni", "No information");
}

} catch (IOException e) {
    e.printStackTrace();
}

System.out.println(content1);
System.out.println("-----");
-----");
System.out.println(content2);

return "result-view";
}

public static void downloadHtml(String url, String saveFilePath, int num)
throws IOException {
    Document doc = Jsoup.connect(url).get();
    try (BufferedWriter writer = new BufferedWriter(new
FileWriter(saveFilePath+"\\syllabus"+num+".html"))) {
        writer.write(doc.html());
    }
    System.out.println("HTML-page successfully downloaded and saved:

```

```

syllabus"+num+".html");
    }
    public static String getContentType(String url) throws IOException {
        disableSSLVerification(); // Вызов метода для отключения проверки SSL-
сертификатов

        HttpURLConnection connection;
        if (url.startsWith("https")) {
            connection = (HttpsURLConnection) new URL(url).openConnection();
        } else {
            connection = (HttpURLConnection) new URL(url).openConnection();
        }
        connection.setRequestMethod("HEAD");
        connection.connect();
        String contentType = connection.getContentType();
        connection.disconnect();
        return contentType;
    }
    public static void downloadFile(String fileURL, String saveDir, int num, String
ext) throws IOException {
        URL url = new URL(fileURL);
        HttpURLConnection httpConn = (HttpURLConnection)
url.openConnection();
        int responseCode = httpConn.getResponseCode();

        if (responseCode == HttpURLConnection.HTTP_OK) {

            InputStream inputStream = httpConn.getInputStream();
            String fileName = "syllabus"+num+ext;

```

```

String saveFilePath = saveDir + File.separator + fileName;
FileOutputStream outputStream = new FileOutputStream(saveFilePath);
int bytesRead;
byte[] buffer = new byte[4096];
while ((bytesRead = inputStream.read(buffer)) != -1) {
    outputStream.write(buffer, 0, bytesRead);
}
outputStream.close();
inputStream.close();
System.out.println("File successfully downloaded and saved: " +
fileName);
} else {
    System.out.println("Error. Code response: " + responseCode);
}
httpConn.disconnect();
}

public static String getContent(File file) throws TikaException, IOException,
SAXException {

    Parser parser = new AutoDetectParser();
    BodyContentHandler handler = new BodyContentHandler(-1);
    Metadata metadata = new Metadata();
    ParseContext context = new ParseContext();
    InputStream stream = new FileInputStream(file);

    metadata.set(Metadata.CONTENT_ENCODING,
StandardCharsets.UTF_8.name());

```

```
parser.parse(stream, handler, metadata, context);
```

```
String content = handler.toString();
```

```
return content;
```

```
}
```

```
public static void disableSSLVerification() {
```

```
try {
```

```
    TrustManager[] trustAllCerts = new TrustManager[]{
```

```
        new X509TrustManager() {
```

```
            public X509Certificate[] getAcceptedIssuers() {
```

```
                return null;
```

```
            }
```

```
            public void checkClientTrusted(X509Certificate[] certs, String
```

```
authType) {
```

```
            }
```

```
            public void checkServerTrusted(X509Certificate[] certs, String
```

```
authType) {
```

```
            }
```

```
        }
```

```
    };
```

```
SSLContext sc = SSLContext.getInstance("TLS");
```

```
sc.init(null, trustAllCerts, new SecureRandom());
```

```
HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
```

```
HostnameVerifier allHostsValid = (hostname, session) -> true;
```

```
HttpsURLConnection.setDefaultHostnameVerifier(allHostsValid);
```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

SumyStateUniversityAgent.java:

```
package com.serhiikuzmenko.spring.mas.agents;
```

```

import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.openqa.selenium.support.ui.WebDriverWait;

```

```
import java.util.NoSuchElementException;
```

```
public class SumyStateUniversityAgent implements SyllabusAgent{
```

```
    @Override
```

```

    public String linkFinder(String desiredSubject, String uni) {
        String link;
        String []words = desiredSubject.split("[\\p{Punct}\\s]");
        String xpathExpression = "//a";
        for (int i=0;i< words.length;++i) {
            xpathExpression += "[contains(text(),'" + words[i] + "')]";
        }
    }

```



```
System.setProperty("webdriver.chrome.driver",
"D:\\USER\\JAVA\\sources\\ua\\edu\\sumdu\\j2se\\Multi-agent-system-data-
collection\\chromedriver.exe");

ChromeOptions options = new ChromeOptions();
options.addArguments("--headless");
WebDriver driver = new ChromeDriver();
try {
    String url = "https://pg.cabinet.sumdu.edu.ua/catalog";
    driver.get(url);
    JavascriptExecutor js = (JavascriptExecutor) driver;
    WebDriverWait wait = new WebDriverWait(driver, 10);
    WebElement element = null;

    long startTime = System.currentTimeMillis();
    long duration = 30000;
    Thread.sleep(500);
    while (System.currentTimeMillis() - startTime < duration) {
        Thread.sleep(100);
        js.executeScript("window.scrollTo(0, document.body.scrollHeight)");

        try{
            element = driver.findElement(By.xpath(xpathExpression));
        }catch (NoSuchElementException e){
            element=null;
        }finally {
            if(element!=null){
                break;
            }else{
                continue;
            }
        }
    }
}
```

```
        }  
    }  
    if(element==null){  
        return "Not found";  
    }  
    link = element.getAttribute("href");  
} catch (InterruptedException e) {  
    throw new RuntimeException(e);  
} finally {  
    driver.quit();  
}  
return link;  
}  
}  
  
GoogleAgent.java:  
package com.serhiikuzmenko.spring.mas.agents;  
  
import com.serhiikuzmenko.spring.mas.service.UniversityService;  
import org.openqa.selenium.By;  
import org.openqa.selenium.JavascriptExecutor;  
import org.openqa.selenium.WebDriver;  
import org.openqa.selenium.WebElement;  
import org.openqa.selenium.chrome.ChromeDriver;  
import org.openqa.selenium.chrome.ChromeOptions;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Component;  
  
import java.util.NoSuchElementException;  
import java.util.regex.Matcher;  
import java.util.regex.Pattern;
```

@Component

```
public class GoogleAgent implements SyllabusAgent{
```

```
    private String lang = "+lang:en";
```

```
    public void setLang(String lang){
```

```
        this.lang = lang;
```

```
    }
```

```
    public String getLang(){
```

```
        return this.lang;
```

```
    }
```

@Override

```
public String linkFounder(String desiredSubject, String uni) {
```

```
    String link;
```

```
    Pattern pattern = Pattern.compile("\\b[a-zA-Z]+\\b");
```

```
    Matcher matcher = pattern.matcher(desiredSubject);
```

```
    StringBuilder newDesiredSubject = new StringBuilder();
```

```
    boolean found = false;
```

```
    while(matcher.find()){
```

```
        found=true;
```

```
        matcher.appendReplacement(newDesiredSubject,
```

```
        "\"" + matcher.group() + "\"");
```

```
        matcher.appendTail(newDesiredSubject);
```

```
    }
```

```
    String searchedSubject = found?newDesiredSubject.toString():desiredSubject;
```

```
    System.out.println(desiredSubject);
```

```
    System.out.println(searchedSubject);
```

```

    System.setProperty("webdriver.chrome.driver",
"D:\\USER\\JAVA\\sources\\ua\\edu\\sumdu\\j2se\\Multi-agent-system-data-
collection\\chromedriver.exe");

    ChromeOptions options = new ChromeOptions();
    options.addArguments("--headless");

    WebDriver driver = new ChromeDriver();

    try {
        String url = "https://www.google.com/search?q=";

        if(this.lang.equals("+lang:ua")){
            driver.get(url+uni.replace(" ",
"+"+"силабус"+searchedSubject.replace(" ", "+")+lang);
        }else{
            driver.get(url+uni.replace(" ",
"+"+"syllabus"+searchedSubject.replace(" ", "+")+lang);
        }
        JavascriptExecutor js = (JavascriptExecutor) driver;
        Thread.sleep(2000);
        WebElement element = null;
        //FIX ME!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
        try{
            if(this.lang.equals("+lang:ua")){
                element =
driver.findElement(By.xpath("//a[//h3[contains(translate(text(),
'АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ',
'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'),
'силабус'])]"));
            }
        }
    }

```

```

'АБВГГДЕЄЖЗИІЙКЛМНОПРСТУФХЦЧШЩЬЮЯ',
'абвггдеежзиийкклмнопрстуфхцчшщьюя'), 'силабус') and
not(ancestor::div[@class='cUnQKe']]);
    }else{
        element = driver.findElement(By.xpath("//a[contains(translate(.,
'ABCDEFGHIJKLMNOPQRSTUVWXYZ', 'abcdefghijklmnopqrstuvwxyz'),
'syllabus') and not(ancestor::div[@class='cUnQKe'])]"));
    }
} catch (NoSuchElementException e){
} finally {
    if(element==null){
        return "Not found";
    }else{
        link = element.getAttribute("href");

        return link;
    }
}
} catch (InterruptedException e) {
    throw new RuntimeException(e);
} finally {

    driver.quit();
}
}
}
}

```