Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Кафедра інформатики факультету інформатики

**Розробка АІС «Пошук тем та запис на курсові та кваліфікаційні роботи».**

**Текстова частина до курсової роботи**

**за спеціальністю “Комп’ютерні науки та інформаційні технології” - 122**

Керівник курсової роботи

Канд. фіз.-мат. наук, доцент

Гулаєва Н. М.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(підпис)*

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2019 р.

Виконав студент КНІТ-3

Чумак В.В.

“\_\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2019 р.

Київ 2019

# Зміст

[Зміст 2](#_Toc7993629)

[Анотація 4](#_Toc7993630)

[Вступ 5](#_Toc7993631)

[1. Аналіз предметної області. Специфікація вимог. 7](#_Toc7993632)

[1.1. Аналіз предметної області 7](#_Toc7993633)

[1.1.1. Опис предметної області 7](#_Toc7993634)

[1.1.2. Збір фактів 8](#_Toc7993635)

[1.1.3. Дослідження документації 10](#_Toc7993636)

[1.1.4. Пріоритизація вимог 11](#_Toc7993637)

[1.2. Специфікація вимог 15](#_Toc7993638)

[1.2.1. Опис груп користувачів АІС 15](#_Toc7993639)

[1.2.2. Вимоги до користувачів функціональної системи 16](#_Toc7993640)

[1.2.3. Вимоги до даних 17](#_Toc7993641)

[1.3. Висновки 19](#_Toc7993642)

[2. Розробка моделі даних 20](#_Toc7993643)

[2.1. Концептуальна модель даних 20](#_Toc7993644)

[2.1.1. Опис обраної Case-системи 20](#_Toc7993645)

[2.1.2. ER-модель даних 21](#_Toc7993646)

[2.1.3. Загальні відомості 22](#_Toc7993647)

[2.2. Реляційна модель даних 23](#_Toc7993648)

[2.2.1. Опис реляційної моделі даних 23](#_Toc7993649)

[2.2.2. Перелік корпоративних обмежень цілісності 29](#_Toc7993650)

[2.3. Висновки 29](#_Toc7993651)

[3. Розробка програмного застосунку 31](#_Toc7993652)

[3.1. Опис засобів програмування 31](#_Toc7993653)

[3.1.1. Обрана СКБД 31](#_Toc7993654)

[3.1.2. Обрані технології, середовище, мови(серверна та клієнтська частини) 31](#_Toc7993655)

[3.2. Опис та реалізація алгоритмів 32](#_Toc7993656)

[3.2.1. Авторизація за допомогою Office 365 32](#_Toc7993657)

[3.2.2. Заповнення галузей за допомогою google scholar 33](#_Toc7993658)

[3.2.3. Пошук тем по імені викладача або назві теми 34](#_Toc7993659)

[3.2.4. Фільтрування тем 34](#_Toc7993660)

[3.2.5. Відправка сповіщень на пошту 35](#_Toc7993661)

[3.2.6. Збір статистики за минулі роки 35](#_Toc7993662)

[3.3. Висновки 36](#_Toc7993663)

[Висновки 37](#_Toc7993664)

[Список Використаної Літератури 38](#_Toc7993665)

[Додаток А 39](#_Toc7993666)

[Додаток Б 40](#_Toc7993667)

[Додаток В 45](#_Toc7993668)

[Додаток Г 47](#_Toc7993669)

# Анотація

В цій роботі описується розробка Автоматизованої Інформаційної Системи для запису на курсові та кваліфікаційні роботи, з використанням сучасних засобів верстки сторінок (HTML, LESS, JavaScript) та мови Python (Django framework) для налаштування та роботи сервера. Проведено дослідження предметної області, з використанням наступних методів збору фактів: мозковий штурм, інтелектуальні карти, проведення інтерв’ю та метод номінальних груп для пріоритизації вимог. На основі проведеного дослідження було розроблено специфікацію вимог та побудовано модель даних.

Результатом роботи є працездатна система, яка дозволяє автоматизувати процес запису студентів на курсові та кваліфікаційні роботи, зменшити навантаження на методиста, як учасника «ручного» процесу запису та перевести процедуру запису та затвердження в онлайн режим.

# Вступ

Кожен студент, починаючи з 2 або 3 курсу (залежно від спеціальності), має у своєму навчальному плані курсову роботу, як форму самостійної наукової роботи з елементами дослідження. Така робота є обов’язковою вимогою для отримання диплому в будь-якому ЗВО.

Для написання такої роботи студенти шукають наукового керівника, теми та наукові інтереси якого збігаються з їхніми. Процедура вибору викладача та теми роботи складається з декількох кроків: викладачі надсилають теми методистам, ті, у свою чергу, формують бланки для запису, а студенти ходять по кафедрах в пошуках потрібної їм теми, або викладача, і записуються у відповідному бланку.

Це спричиняє деякі незручності та проблеми, найголовніші з яких:

Для методиста:

1. Ручне формування бланків, втрата часу
2. Можлива втрата інформації через велику кількість листів на пошту, або потрапляння листа в спам

Для викладача:

1. Неможливість вибрати студента з декількох охочих
2. Неможливість змінити тему без прямого контакту з методистом

Для студента:

1. Необхідність відвідувати кафедри в пошуках теми
2. Складнощі під час випису з теми або її зміни
3. Обмежений вибір через те, що тему вже забрав хтось інший
4. Складнощі під час комунікації з викладачем, особливо з таким, що не часто з’являється в університеті

Електронна система запису дасть змогу запобігти подібним проблемам, адже процедура проходитиме в режимі онлайн і не вимагатиме багато часу для пошуку потрібної теми або викладача. Викладачі зможуть вибрати одного з декількох записаних студентів, легко змінити тему, якщо буде така необхідність. Крім того, план роботи з описом проміжних станів та дедлайнів буде гарним доповненням системи.

У НаУКМА вже декілька років активно використовується електронна система запису на дисципліни. Зважаючи на досвід введення такої системи, можна стверджувати, що автоматизована система запису на курсові та кваліфікаційні роботи спростить процедуру вибору викладача та теми роботи.

Отже, метою розробки є створення автоматизованої системи запису на курсові та кваліфікаційні роботи для впровадження більш зручного, модерного способу запису та заміни «ручного», полегшення роботи всіх учасників процедури, переведення процесу в режим онлайн.

# Аналіз предметної області. Специфікація вимог.

# Аналіз предметної області

# Опис предметної області

НаУКМА має 6 факультетів, кожен з яких поділяється не менше ніж на 4 кафедри. Кожен викладач працює на одній з кафедр, за кожною спеціальністю закріплена кафедра, як випускова.

Викладач, які мають у своєму навантаженні години курсової або кваліфікаційної роботи, надають список тем та набирають відповідну кількість студентів.  Якщо викладач не набрав відповідну кількість студентів, то у 2-му семестрі його години передаються іншому викладачеві (до якого здійснили запис).

На початку навчального року викладачі надсилають теми на пошту методистам факультетів, які, зібравши всі теми, формують документи та бланки для запису (Додаток А).

Після початку запису, кожен студент йде на кафедру й обирає тему роботи. Викладач, чию тему вибрав студент, закріплюється за ним як науковий керівник.

Така процедура запису має певні недоліки та проблеми, зокрема спричинені людським фактором:

* Втрата інформації, надісланої викладачем методисту;
* Некоректне формування бланків із темами;
* Складнощі з редагуванням тем;

Більшість студентів користуються можливістю попередньої домовленості з викладачем. Студенти, що не зробили цього завчасно, мають значно менший вибір, а тому, порівняно з іншими, мають менші шанси обрати потрібну тему. До того ж, студенти, які намагаються обрати тему завчасно не знають всього списку тем, а тому обирають, орієнтуючись лише на кваліфікацію викладача або область в якій він працює.

Деякі студенти не встигають вчасно записатися з різних причин, серед яких, зокрема, небажання або неможливість відвідати кафедру. Іноді запис таких студентів відбувається ближче до нового року, а це означає, що вони втрачають час і можуть не встигнути закінчити роботу.

Деякі студенти мають чіткі побажання щодо теми роботи, які відповідають їхнім науковим інтересам або навичкам. Саме тому, для них було б зручно шукати теми робіт, які безпосередньо пов'язані з їхніми інтересами, знаннями або вміннями.

Ще однією проблемою, з якою часто стикаються студенти, є комунікація з науковим керівником. Частина викладачів дуже рідко з’являється в університеті, і тому існує проблема з проведенням консультацій стосовно роботи.

Дослідивши предметну область, можна виділити чіткий список недоліків та проблем, які з легкістю можна було б обійти, впровадивши електронну систему запису, що ще раз підкреслює актуальність такої роботи.

# Збір фактів

Для збору фактів використовувались наступні методики: проведення інтерв’ю(Додаток Б), вивчення документації, створення інтелектуальної карти (mind map), метод номінальних груп для пріоритизації вимог.

Після проведення інтерв’ю був складений список основних ідей і побажань щодо системи, які згодом були внесені до інтелектуальної карти, а деякі з них – реалізовані. Прямий контакт з потенційними користувачами системи дозволяє оцінити їх вимоги та побажання і врахувати це під час розробки.

Mind map (інтелектуальна карта) - один зі способів збору та представлення думок, ідей, завдань або інших елементів у вигляді діаграми. Зазвичай реалізована як дерево думок, вітки якого є зв’язками. Ця діаграма допомагає впорядкувати та структурувати великий набір інформації для полегшення подальшої її обробки та вивчення.

Для створення інтелектуальної карти було використано сервіс Bubble us.

Bubble us – безкоштовний кросплатформенний інструмент для побудови інтелектуальних карт(mind maps). Дозволяє графічно представити свої ідеї, впорядкувати та структуризовати інформацію для кращого її сприйняття.

В результаті було отримано карту з усіма, зібраними ідеями та думками.

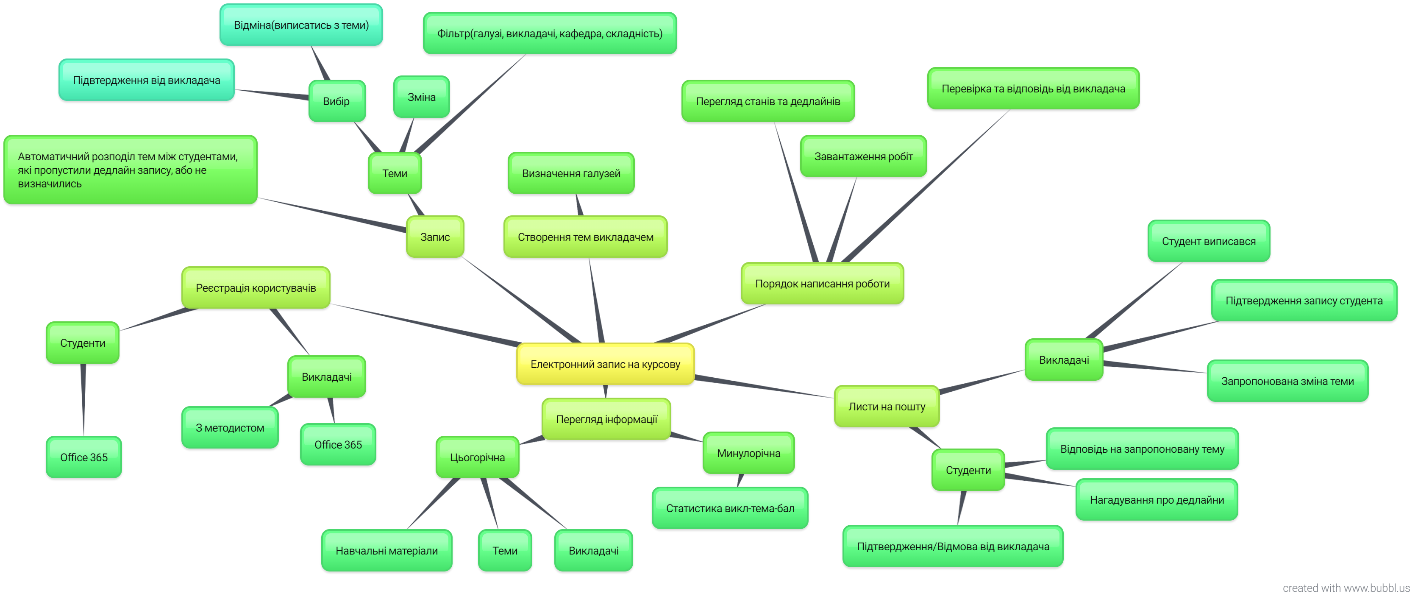


Рисунок 1.1

1. Реєстрація студентів за допомогою Office 365
2. Реєстрація викладачів за допомогою Office 365
3. Перегляд навчальних матеріалів
4. Перегляд тем та викладачів
5. Перегляд минулорічної статистики
6. Вибір теми студентом
7. Пропозиція зміни теми
8. Відповідь викладача по зміні
9. Фільтр тем по галузях
10. Фільтр тем по викладачах
11. Фільтр тем по кафедрах
12. Запис на курсову студентом
13. Підтвердження або відхилення запису викладачем
14. Створення тем викладачем
15. Можливість додавати наукові  галузі до теми
16. Автоматизований запис студентів, які залишились без тем
17. Сповіщення викладачу про випис студента
18. Сповіщення викладачу про запропоновану зміну теми
19. Сповіщення викладачу про запис студента
20. Сповіщення студенту про підтвердження або відмову від викладача
21. Сповіщення студенту про зміну теми
22. Нагадування студенту про дедлайни
23. Сповіщення студенту про відповідь на проміжну роботу
24. Сповіщення викладачу про завантаження проміжного результату
25. Можливість планувати порядок роботи (після вибору та підтвердження теми)
26. Завантаження проміжних результатів студентом
27. Перевірка та відповідь викладача

Побудова інтелектуальної карти допомагає візуально представити майбутню систему, впорядкувати та систематизувати всі ідеї та вимоги до неї.

Звісно, не все, що там описано буде створено, адже MindMap - це лише набір існуючих ідей, який згодом буде змінюватись (після вибору найголовніших, проведення додаткових опитувань та анкетувань).

# Дослідження документації

Вивчення документації – це спосіб дізнатися область роботи та зрозуміти потреби користувачів, що в майбутньому допоможе в коректній побудові архітектури програми.

Мною було проаналізовано структуру бланків запису, сформованих методистами (Додаток А), а також положення про організацію освітнього процесу в НаУКМА[1]. Це дало можливість краще зрозуміти процес запису та зібрати необхідні, для створення системи, дані.

# Пріоритизація вимог

Пріоритизація вимог – метод визначення основних вимог для реалізації певної системи. Проводиться шляхом оцінювання кожної вимоги (1-5 балів). Вимоги, які набирають найбільше балів - обов’язкові для реалізації.

Після проведення інтерв’ю зі студентами та аналізу інтелектуальної карти була проведена пріоритизація вимог. Ось список цих вимог:

1. Реєстрація з використанням існуючих ресурсів, таких як Office 365 або my.ukma
2. Реєстрація за студентським квитком
3. Статистика минулих років у вигляді “викладач-тема”
4. Статистика минулих років у вигляді “викладач-тема-студент (можливо з контактами)”, для можливості доопрацювання теми
5. Можливість ознайомитися зі списком тем до початку запису
6. Можливість фільтру тем (по напрямку діяльності, викладачах, складності)
7. Фільтр викладачів за їх науковими інтересами
8. Можливість запису декількох студентів на одну тему і надання викладачу права обирати студента
9. Можливість виписатися з теми
10. Можливість автоматичного запису на теми, що залишились, для студентів, що пропустили дедлайн або не визначились
11. Можливість отримувати навчальні матеріали до відповідної теми
12. Комунікація з викладачем безпосередньо на сайті
13. Можливість завантажувати роботи на сайт, для їх перевірки викладачем
14. Можливість отримувати сповіщення на пошту про зміну стану запису(підтвердження/відмова) та стану роботи

Серед всіх вимог  до системи, кожен з опитуваних обрав 5 найважливіших(на його думку) та присвоїв їм бали від 1 до 5.

В результаті було отримано:

|  |  |
| --- | --- |
| Реєстрація з використанням існуючих ресурсів, таких як Office 365 або my.ukma.edu.ua | 61 |
| Реєстрація за студентським квитком | 26 |
| Статистика минулих років у вигляді “викладач-тема” | 44 |
| Статистика минулих років у вигляді “викладач-тема-студент (можливо з контактами)”, для можливості доопрацювання теми | 28 |
| Можливість ознайомитися зі списком тем до початку запису | 28 |
| Можливість сортування тем (по напрямку діяльності, викладачах, складності) | 39 |
| Фільтрування викладачів за їх науковими інтересами | 24 |
| Можливість запису декількох студентів на одну тему і надання викладачу права обирати студента | 20 |
| Можливість виписатися з теми до підтвердження запису викладачем | 33 |
| Можливість автоматичного запису на теми, що залишились, для студентів, що пропустили дедлайн або не визначились | 30 |
| Можливість отримувати навчальні матеріали до відповідної теми | 23 |
| Можливість створення та редагування плану виконання роботи | 30 |
| Можливість отримувати сповіщення на пошту про зміну стану запису (підтвердження / відмова) та стану роботи | 50 |

Таблиця 1.1

В голосуванні взяло участь 20 студентів, згідно присвоєним балам отримуємо впорядкований список вимог:

1. Реєстрація з використанням існуючих ресурсів, таких як Office 365 або my.ukma.edu.ua
2. Можливість отримувати сповіщення на пошту про зміну стану запису (підтвердження / відмова) та стану роботи
3. Статистика минулих років у вигляді “викладач-тема”
4. Можливість фільтрації тем (по напрямку діяльності, викладачах, складності)
5. Можливість виписатися з теми до підтвердження запису викладачем
6. Можливість автоматичного запису на теми, що залишились, для студентів, що пропустили дедлайн або не визначились
7. Можливість створення та редагування плану виконання роботи
8. Статистика минулих років у вигляді “викладач-тема-студент (можливо з контактами)”, для можливості доопрацювання теми
9. Можливість ознайомитися зі списком тем до початку запису
10. Реєстрація за студентським квитком
11. Фільтрування викладачів за їх науковими інтересами
12. Можливість отримувати навчальні матеріали до відповідної теми
13. Можливість запису декількох студентів на одну тему і надання викладачу права обирати студента

Під час опитування, була поставлена умова, щоб одним із вибраних пунктів був спосіб реєстрації, адже важливо знати, який з них був би зручніший. Реєстрація за допомогою сервісів Office 365 або  my.ukma.edu.ua виявилась зручнішою для більшості опитуваних. Важливим пунктом стала можливість отримувати сповіщення про дедлайни та зміну станів запису.

Отже, після проведення даного опитування та співбесіди з викладачем(Додаток Б.6) з приводу визначення основних вимог, можна відібрати найбільш важливі функції системи, які будуть реалізовані, а також один із способів реєстрації, який виявився більш зручним для респондентів:

* Реєстрація з використанням існуючих ресурсів, таких як Office 365
* Можливість отримувати сповіщення на пошту про зміну стану запису (підтвердження / відмова) та стану роботи, про дедлайни
* Статистика минулих років у вигляді “викладач-тема”
* Можливість фільтрації тем (по напрямку діяльності, викладачах, складності)
* Можливість виписатися з теми до підтвердження запису викладачем
* Можливість створення та редагування плану виконання роботи
* Статистика минулих років у вигляді “викладач-тема-студент (можливо з контактами)”, для можливості доопрацювання теми
* Можливість ознайомитися зі списком тем до початку запису, а також з темами минулих років, доступних для продовження
* Можливість отримувати навчальні матеріали до відповідної теми
* Можливість запису декількох студентів на одну тему і надання викладачу права обирати студента.

# Специфікація вимог

# Опис груп користувачів АІС

Учасниками запису на курсові та кваліфікаційні роботи є методисти, викладачі та студенти.

Методисти відповідають за всю документацію, формування списків та бланків і фактично є зв’язуючим компонентом між викладачами і студентами, вирішуючи будь-які проблеми, що стосуються запису.

Викладачі надсилають теми робіт методистам і, в подальшому, закріплюються, як наукові керівники студентів, що обрали їхні теми. Вони консультують студентів, надають потрібну літературу та допомогу в написанні роботи, перевіряють результати, пишуть рецензію.

Студент обирає одну з запропонованих тем, записуючись у бланку.

У системі методист відповідатиме за внесення інформації про години навантаження для кожного викладача, про дати захисту і передзахисту робіт, які пишуться у викладачів його кафедри.

Викладачі можуть додавати та редагувати теми своїх робіт, затверджувати одного із студентів, записаних на тему, переглядати всю відкриту інформацію на сайті (теми робіт, інформація про студентів, викладачів та запис). Викладач має змогу отримувати на пошту повідомлення про те, що студент записався, або виписався з теми. Для отримання галузей роботи конкретного викладача, використовується посилання на його профіль в Google Scholar.

Студенти можуть переглядати інформацію про теми, фільтрувати за кафедрами свого факультету, статусом запису, науковими інтересами керівників та галузями робіт, шукати тему за ім'ям викладача або назвою теми, записуватись на тему та виписуватися з неї до затвердження викладачем, переглядати відкриту інформацію на сайті. Після запису на тему, студент може переглядати дедлайни передзахисту та захисту, прописані на сайті. Студент отримує повідомлення на пошту про рішення викладача, стосовно затвердження на тему.

Для управління сайтом створена роль модератора, який може додавати, редагувати або видаляти всі дані за необхідністю. Оскільки метадані про користувачів системи відсутні, модератор сам призначає ролі кожного з них.

# Вимоги до користувачів функціональної системи

**Методист кафедри**

* реєстрація в системі, використовуюючи засоби Office 365 API
* виставлення навантаження викладачам своєї кафедри
* перегляд відкритої інформації на сайті;
* додавати інформацію про дати захистів та попередніх захистів робіт для спеціальностей своєї випускової кафедри;

**Викладач**

* реєстрація в системі, використовуюючи засоби Office 365 API
* додавати напрями своїх інтересів з інших ресурсів (зокрема google scholar)
* формулювати теми робіт в кількості, зазначеній в навантаженні
* переглядати список студентів, записаних на його на тему
* підтверджувати запис студента
* формувати план виконання роботи, додаючи дедлайни та коментарі
* переглядати відкриту інформацію на сайті;
* отримувати листи на пошту з повідомленням про те, що студент записався на його тему або виписався з неї

**Студент**

* реєстрація в системі, використовуюючи засоби Office 365 API
* переглядати список тем
* здійснювати пошук по імені викладача, назві теми
* фільтрація по кафедрі, галузях роботи, наукових інтересах викладача, статусу запису
* записуватись на тему
* виписуватись з теми
* переглядати план роботи, дедлайни, дати захисту
* отримувати листи на пошту про підтвердження запису

**Модератор**

* класифікувати потрібних користувачів, як методистів, додавати необхідну інформацію про них (факультет, кафедра)
* додавати, редагувати, видаляти інформацію про факультети
* додавати, редагувати, видаляти інформацію про спеціальності
* додавати, редагувати, видаляти інформацію про кафедри
* додавати, редагувати, видаляти інформацію про користувачів
* додавати, редагувати, видаляти інформацію про теми робіт
* додавати, редагувати, видаляти інформацію про галузі знань
* редагувати будь-яку іншу інформацію за необхідності, проводити запис, реєстрацію, виписування студентів тощо

Передбачений автоматичний функціонал системи:

* розсилка листів на пошту(сповіщення про запис або випис, тощо)
* заповнення галузей викладача на основі профілю в google scholar
* заповнення інформації про користувача на основі профілю в office 365

# Вимоги до даних

**Викладачі**

Про викладача відомо його повне ім’я, кафедра, на якій працює, кількість місць (для курсових та кваліфікаційних робіт, з поділом по роках та спеціальностях на кожен рік, в який викладач має години навантаження курсової), логін Office 365 для входу в систему, список наукових інтересів, науковий ступінь, вчене звання, посада, додаткова пошта, телефон та посилання на профіль в google scholar за бажанням.

**Студент**

Із системою запису співпрацюють студенти, які мають курсову роботу у своєму індивідуальному плані на поточний рік. Про них має бути відомо: повне ім’я, логін Office 365, курс, спеціальність, середній бал, рік вступу, додатковий email.

**Методист**

Про методиста має бути відомо повне ім’я, кафедра, на якій працює, логін Office 365 для входу в систему, телефон, кабінет, додатковий email за бажанням.

**Факультети**

Кожен студент та викладач відносяться до певного факультету. Щоб була можливість розділити користувачів системи по факультетам, важливо знати назву факультету, декана факультету, методиста, та їх контакти (адреса електронної скриньки, номер телефону), корпус та номери кімнат (декана і деканату).

**Спеціальності**

Кожен студент навчається на певній спеціальності, за якою готують на факультеті. Про спеціальність мають бути відомі: шифр та назва галузі, до якої дана спеціальність належить(наприклад 12 – Інформаційні технології), а також шифр та назва самої спеціальності(122 – Комп'ютерні науки)[2], факультет до якого вона належить. Кожна тема передбачається для конкретних спеціальностей та курсів.

**Кафедри**

Кожен викладач працює на певній кафедрі, яка відноситься до одного з факультетів. Про кафедру нам важливо знати її назву та факультет, якому вона належить, методистів та контакти (адреса електронної пошти, номер телефону), корпус, кімната.

Кафедра може бути випусковою для якоїсь спеціальності.

**Галузі**

Існує список галузей знань, що присвоюються кожній темі та кожному викладачу для фільтрації за ними. В галузі записується лише її назва. Галузь може бути підгалуззю більш узагальненої.

**Теми курсових та кваліфікаційних робіт**

Кожен викладач подає теми, у кількості що відповідають його навантаженню, та прописує наукові галузі, до яких належить тема. Для кваліфікаційної роботи тема прописується українською та англійською мовами.

Згодом студент обирає одну з тем (відображаються лише теми, що належать спеціальності студента).

Про тему відома її назва, спеціальність, на якій її пишуть, галузі, викладач, потім – студенти, що подали заявку, на обрання теми, та відповідь викладача на неї.

Деякі теми можна продовжити з попередніх років, в тому числі від інших студентів.

**План виконання роботи**

Після того, як студент записався на тему, викладач може формувати план роботи, для зручності організації процесу написання та комунікації. Кожен пункт плану містить опис та може містити термін виконання.

# Висновки

Дослідивши предметну область, визначивши вимоги та побажання до майбутньої системи, зібравши всю потрібну документацію та інформацію, можна переходити до визначення основних сутностей та побудови моделі даних. Проведене дослідження показує наскільки необхідною є подібна система, а тому й доводить актуальність її створення.

# Розробка моделі даних

# Концептуальна модель даних

# Опис обраної Case-системи

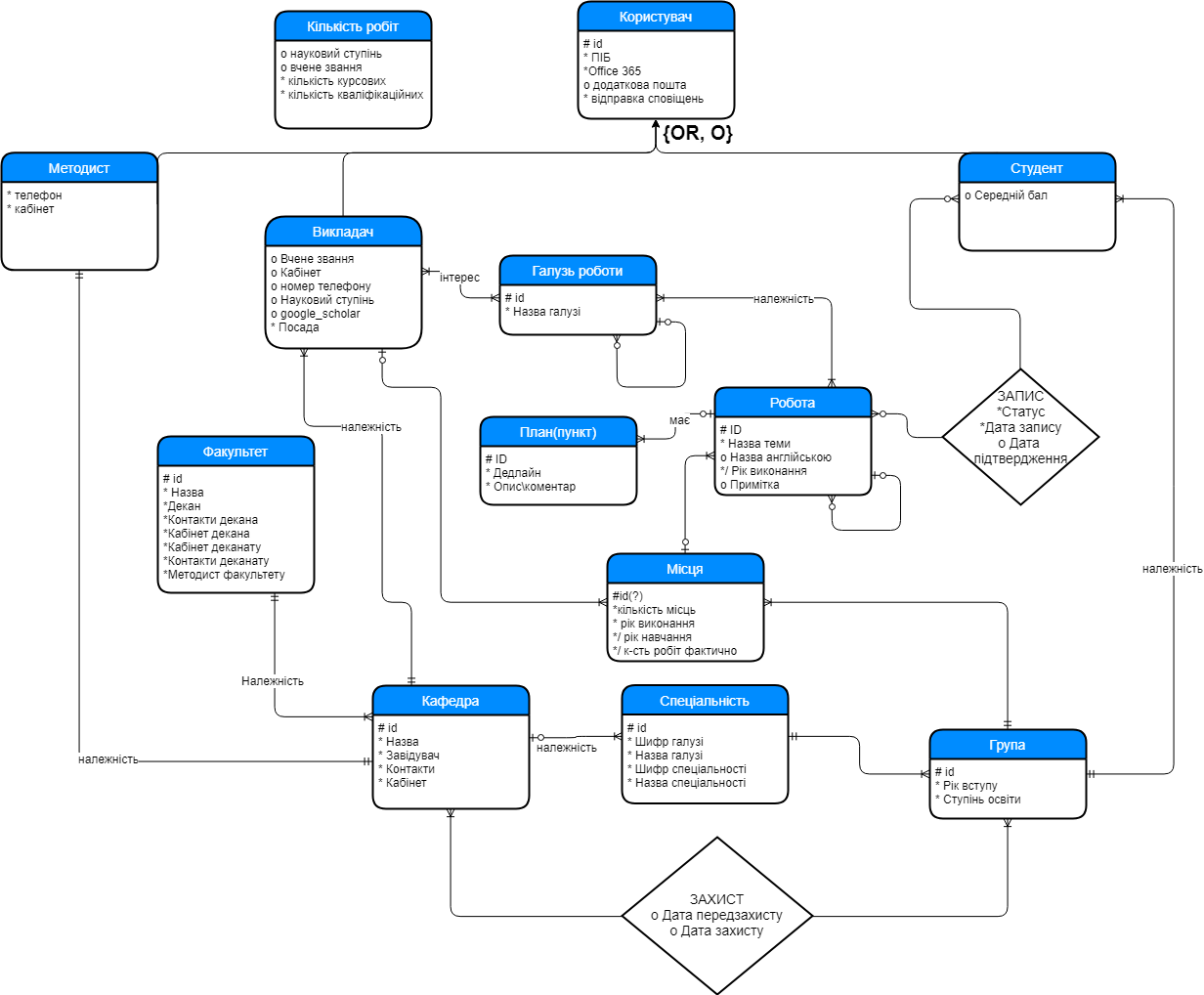
Для побудови ER-моделей існує багато інструментів, case- систем. Мною була використана одна з найпопулярніших – [draw.io](https://www.draw.io). Дана система дозволяє будувати будь-які діаграми класів, ER-моделі, різноманітні блок-схеми. Систему зручна та зрозуміла у використанні, дозволяє тримати дані у хмарі і швидко дістає їх звідти за потреби. Варто зауважити також синхронізацію з Google та Github аккаунтами користувача. Система використовує досить популярну систему позначень в побудові ER-моделей.

|  |  |
| --- | --- |
| Позначення | Пояснення |
|  | Зображення сутності та перелік атрибутів, де:  # – первинний ключ  \* – обов’язковий атрибут  о – необов’язковий атрибут  \*/ – похідний атрибут |
|  | Позначення зв’язку «багато до багатьох» з обов’язковістю (|) та необов’язковістю (о) входження учасників зв’язку |
|  | Позначення зв’язку «один до багатьох» з необов’язковістю (о) та обов’язковістю (|) входження учасників зв’язку |
|  | Позначення складеного зв’язку його назви та атрибутів |

Таблиця 2.1

Дана case-система досить зручна у використанні, хоча і має певні недоліки, наприклад, відсутність форматування атрибутів, які доводиться вручну виставляти в стовпці; відсутність функції автоматичної побудови реляційної моделі по створеній ER. Загалом, система повністю підходить для побудови моделі даних.

# ER-модель даних



# Загальні відомості

* 1. **Користувач** – відомості про користувачів системи, серед яких студенти, викладачі та методисти. Специфікація вимог – студенти, викладачі, методисти.

*Потенційні ключі*: id, office365.

*Первинний ключ*: id.

* 1. **Факультет** – відомості про факультети університету. Специфікація вимог – факультет.

*Потенційні ключі:* id, назва.

*Первинний ключ:* id.

* 1. **Кафедра** – відомості про кафедри факультетів. Специфікація вимог – кафедра.

*Потенційні ключі:* id, назва.*.*

*Первинний ключ:* id*.*

* 1. **Спеціальність** – відомості про спеціальності факультетів. Сутність розбита на дві частини для розбиття спеціальностей по рокам. Специфікація вимог – спеціальність.

*Потенційні ключі:* id, шифр галузі + шифр спеціальності, назва галузі + назва спеціальності*.*

*Первинний ключ:* id*.*

* 1. **Робота** – відомості про теми курсових робіт, наданих викладачами. Специфікація вимог – теми курсових та кваліфікаційних робіт.

*Потенційні ключі:* id, назва теми + рік виконання*.*

*Первинний ключ:* id*.*

*Похідні атрибути:* рік виконання – відповідне значення із сутності Місця.

* 1. **Місця** – відомості про кількість місць для тем виділених для викладача цього року. Специфікація вимог – місця.

*Потенційні ключі:* id*.*

*Первинний ключ:* id*.*

* 1. **Галузь роботи –** інформація про галузі роботи викладачів. Специфікація вимог – галузі.

*Потенційні ключі:* id, назва*.*

*Первинний ключ:* id*.*

* 1. **Кількість робіт** – інформація про максимальну кількість робіт для викладачів.

*Потенційні ключі:* id*.*

*Первинний ключ:* id*.*

* 1. **План роботи –** інформація про пункти плану, створені для конкретної роботи.

*Потенційні ключі:* id*.*

*Первинний ключ:* id*.*

* 1. Реляційна модель даних

# Опис реляційної моделі даних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FACULTY  Відповідає типу сутності «Факультет» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
| AK | name | Char(40) | NOT NULL |  |
|  | dean | Char(40) | NOT NULL |  |
|  | dean\_contact | Char(200) | NOT NULL |  |
|  | dean\_office | Char(40) | NOT NULL |  |
|  | deanery\_contact | Char(200) | NOT NULL |  |
|  | deanery\_office | Char(40) | NOT NULL |  |
|  | faculty\_methodist | Char(40) | NOT NULL |  |

Таблиця 2.2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Department  Відповідає типу сутності «Кафедра» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
| AK | department\_name | Char(250) | NOT NULL |  |
|  | head\_of\_department | Char(250) | NOT NULL |  |
|  | office | Char(250) | NOT NULL |  |
|  | phone | Char(250) | NOT NULL |  |
| FK | faculty\_id | Integer | NOT NULL | **ON DELETE**  NO ACTION  **ON UPDATE**  NO ACTION |

Таблиця 2.3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SPECIALTY  Відповідає типу сутності «Спеціальність» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
| PAK | specialty\_code | SmallInteger | NOT NULL |  |
|  | specialty\_name | Char(250) | NOT NULL |  |
| PAK | branch\_code | SmallInteger | NOT NULL |  |
|  | branch\_name | Char(250) | NOT NULL |  |
| FK | сathedra\_id | Integer | NOT NULL | **ON DELETE**  NO ACTION  **ON UPDATE**  NO ACTION |

Таблиця 2.4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StudentGroup  Відповідає типу сутності «Група» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | year\_of\_entry | SmallInteger | NOT NULL |  |
| FK | specialty\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Protection  Відповідає зв’язуючому відношенню для двостороннього зв’язку «багато-до-багатьох» між відношеннями StudentGroup i Department | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | Id | Integer | NOT NULL |  |
| FK | spec\_group | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
| FK | teacher\_department | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
|  | date\_of\_pre\_protection | Date | NULL |  |
|  | date\_of\_protection | Date | NULL |  |

Таблиця 2.6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TopicOffer  Відповідає типу сутності «Місця» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | count\_of\_themes | SmallInteger | NOT NULL |  |
|  | fact\_count\_of\_themes | SmallInteger | NOT NULL |  |
|  | year\_of\_study | SmallInteger | NOT NULL |  |
|  | year\_of\_work | SmallInteger | NOT NULL |  |
| FK | teacher\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
| FK | specialtyGroup\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.7

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CountOfWork  Відповідає типу сутності «Кількість робіт» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | degree | Char(100) | NULL |  |
|  | academic\_status | Char(100) | NULL |  |
|  | course\_work\_count | SmallInteger | NOT NULL |  |
|  | qualification\_work\_count | SmallInteger | NOT NULL |  |

Таблиця 2.8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| WriteWork  Відповідає типу сутності «Робота» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | work\_name | Char(500) | NOT NULL |  |
|  | english\_work\_name | Char(500) | NULL |  |
|  | note | Char(500) | NULL |  |
|  | year\_of\_work | SmallInteger | NOT NULL |  |
| FK | teacher\_offer | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
| FK | prev\_version | Integer | NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.9

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Record  Відповідає зв’язуючому відношенню для двостороннього зв’язку «багато-до-багатьох» між відношеннями Student i WriteWork | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
| FK | student\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
| FK | work\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
|  | status | Char(50) | NULL |  |
|  | date\_of\_record | Date | NOT NULL |  |
|  | date\_of\_confirmation | Date | NULL |  |

Таблиця 2.10

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BRANCH  Відповідає типу сутності «Галузі» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
| AK | name | Char(100) | NOT NULL |  |
| FK | wider\_branch | Integer | NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.11

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LECTOR-BRANCH  Відповідає зв’язуючому відношенню для двостороннього зв’язку «багато-до-багатьох» між відношеннями Lecturer і Branch | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PPK,FK | BranchID | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
| PPK,FK | LecturerID | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| BRANCH\_WORK  Відповідає зв’язуючому відношенню для двостороннього зв’язку «багато-до-багатьох» між відношеннями Branches i WriteWork | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PPK,FK | branch\_id | Integer | NOT NULL | **ON DELETE**  NO ACTION  **ON UPDATE**  CASCADE |
| PPK,FK | coursework\_id | Integer | NOT NULL | **ON DELETE**  NO ACTION  **ON UPDATE**  CASCADE |

Таблиця 2.13

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| USER  Відповідає типу сутності «Користувач» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | first\_name | Char(50) | NOT NULL |  |
|  | second\_name | Char(50) | NOT NULL |  |
|  | third\_name | Char(50) | NOT NULL |  |
|  | add\_email | Char(50) | NULL |  |
|  | office\_365 | Char(50) | NOT NULL |  |
|  | send\_email | Boolean | NOT NULL |  |

Таблиця 2.14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student  Відповідає типу сутності «Студент» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK, FK | user\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
|  | average\_mark | SmallInteger | NULL |  |

Таблиця 2.15

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Teacher  Відповідає типу сутності «Викладач» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK, FK | user\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
|  | degree | Char(250) | NULL |  |
|  | academic\_status | Char(250) | NULL |  |
|  | cabinet | Char(10) | NULL |  |
|  | phone | Char(20) | NULL |  |
|  | google\_scholar | Char(250) | NULL |  |
|  | position | Char(250) | NOT NULL |  |
| FK | depatment | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |

Таблиця 2.16

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Methodist  Відповідає типу сутності «Методист» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK, FK | user\_id | Integer | NOT NULL | **ON DELETE**  CASCADE  **ON UPDATE**  CASCADE |
|  | phone | Char(20) | NOT NULL |  |
|  | cabinet | Char(10) | NOT NULL |  |

Таблиця 2.17

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PLAN  Відповідає типу сутності «План(пункт)» | | | | |
| Ключ | Ім’я атрибуту | Тип атрибуту | NULL  /NOT NULL | FK: ON DELETE  ON UPDATE |
| PK | id | Integer | NOT NULL |  |
|  | deadline | Date | NULL |  |
|  | comment | Char(255) | NOT NULL |  |

Таблиця 2.18

# Перелік корпоративних обмежень цілісності

Додаткові правила підтримки цілісності даних, які відображують обмеження предметної області:

* Кількість місць для робіт у конкретного викладача на конкретний рік не може бути більшою ніж кількість виділена методистом
* Кількість місць, виділених методистом для деякого викладача не може бути більше 8
* Кількість кафедр на факультеті не може бути менше 4
* Роки вступу студента до університету та написання ним роботи не можуть бути більшими, ніж поточний рік
* Рік написання роботи не може бути меншим за рік вступу
* Середній бал студента не може бути менше 0 або більше 100
* Дата запису на роботу не може бути більшою від дати підтвердження
* Фактична кількість робіт викладача не може перевищувати кількість виділених для нього місць

Для підтримки даних обмежень створено файл constants.py (Додаток Г), який зберігає константи і їх значення.

# Висновки

Розроблена модель, враховує всі вимоги зазначені в розділі 1, дозволяє реалізувати весь запланований функціонал та створити працездатну систему. Модель буде перенесено в певну СКБД для подальшого її використання у розробці та впровадженні системи. Під час створення моделі визначився великий список корпоративних обмежень цілісності, які також будуть враховані під час розробки застосунку.

# Розробка програмного застосунку

# Опис засобів програмування

# Обрана СКБД

Для розробки застосунку була обрана СКБД – MySQL версії 8.0.15. Дана СКБД є однією з найпоширеніших та найпопулярніших у використанні. Однією з найбільших переваг даної СКБД є наявність добре структурованої та зрозумілої документації, де можна дізнатись будь-які нюанси її функціонування.

Варто зауважити про зручний графічний інтерфейс, представлений додатком Workbench, який дозволяє будувати реляційну модель, заповнювати базу інформацією, тестувати запити, тощо.

MySQL володіє великою клієнтською базою, а тому розв’язання майже будь якої проблеми, пов’язаної з роботою даної СКБД можна знайти в інтернеті, якщо такого немає, то точно знайдеться причина і способи вирішення.

# Обрані технології, середовище, мови(серверна та клієнтська частини)

Для розробки бекенд частини прототипу застосунку був обраний фреймворк Django[3] на мові Python, який використовує шаблон проектування MVC. Даний фреймворк є легким для розуміння та дозволяє швидко і ефективно організувати роботу застосунку, вважається фреймворком з низьким порогом входження та має чітку і структуровану документацію.

Для розробки фронтенд частини активно використовувалась зв’язка HTML+LESS [4]. Для забезпечення адаптивної верстки та плавного переходу між елементами використовувалась технологія Bootstrap [5].

Для забезпечення авторизації на сайті використовувалось API Office 365 [6], яке дозволяє контролювати вхід користувачів лише по їх пошті. Даний метод авторизації був обраний респондентами, серед яких проводилось опитування. Він активно використовується в інших системах НаУКМА, а тому система автоматизованого запису на курсові та кваліфікаційні роботи може бути легко синхронізована з ними.

Робота над реалізацією системи була розділена між мною та Олександрою Радзієвською. Колега відповідала за деяку клієнтську частину системи та написання функціоналу для викладача і методиста. Моєю задачею було забезпечити авторизацію за допомогою Office 365 та ідентифікувати користувачів, спираючись на дані, отримані зі сховища Microsoft.

Основною частиною роботи було створення зручних фільтрів та можливостей пошуку тем, адже це є чи не найбільшою перевагою онлайн системи над «ручним» способом запису. Іншою, не менш важливою, задачею було навчити систему автоматично відправляти сповіщення на пошти користувачів про зміни, які відбувалися в процесі запису.

Відповідно до вимог, була реалізована статистика запису минулих років, для того, щоб користувачі могли переглядати викладачів, теми та студентів, які писали по ним роботи.

# Опис та реалізація алгоритмів

# Авторизація за допомогою Office 365

Для реєстрації та авторизації користувачів у системі було використане Office 365 Management API. Даний набір інструментів дозволяє реєструвати та класифікувати користувачів системи засобами Office 365, надає доступ до інформації про користувача, яка зазначена в акаунті (повне ім’я, громадянство, електронна адреса, тощо).

Такий спосіб авторизації дозволяє швидко дістати інформацію про користувача та наповнити нею базу даних для подальшого використання у своїй системі.

Для того, щоб налаштувати авторизацію таким способом, слід виконати наступні кроки:

1. Реєстрація застосунку в Azure AD, отримання його ключа та ідентифікатора
2. Налаштування посилань для повернення після успішної авторизації
3. Надсилання запиту на отримання токену для користувача
4. Звернення до API-сервісів з передаванням токену для отримання інформації

Робота ресурсу налаштована по принципу токенізації, який широко застосовується в інформаційній безпеці. За умови успішної авторизації, користувач отримує токен – унікальний шифр, який дійсний лише під час даної сесії та дозволяє звертатись до серверу Office 365 для отримання необхідних даних.

Після отримання токену для доступу, застосунок може звертатись до API-сервісів та отримувати потрібну інформацію у вигляді JSON-формату.

Прикладом такого звернення є функція get\_me() (Додаток В.1), яка використовується для отримання основної інформації про користувача, який щойно авторизувався.

Після успішної авторизації користувач потрапляє на головну сторінку застосунку(сторінку з темами), а його ідентифікатор зберігається у веб сховищі (Session Storage) до закінчення сесії поточного користувача. За допомогою цього ідентифікатора, легко розпізнати авторизованого користувача і налаштувати його роботою з базою даних.

# Заповнення галузей за допомогою google scholar

Ще одним зовнішнім ресурсом, який був використаний, у створенні системи є google scholar (Академія Google) [7] – безкоштовна система для пошуку наукових публікацій та видань будь-яких дисциплін. Оскільки, на цьому ресурсі присутня більшість викладачів НаУКМА, сторінки їх акаунтів можна використовувати для збору даних по науковим інтересам та галузям роботи. Кожен профіль містить в собі перелік наукових інтересів, якими було заповнено базу.

Якщо викладач залишив посилання на свій профіль в google scholar, то при кожній його авторизації відбувається перевірка профілю на наявність нових галузей і додавання їх в базу, якщо такі є.

За парсинг профілю викладача в google scholar відповідає функція find\_branches() (Додаток В.2), яка по класу html-тегу, знаходить інформацію про галузі науки і додає їх в базу.

Такий спосіб пошуку, дозволяє заощадити час на створювання галузей, які вже додавались в іншому застосунку та автоматизує заповнення бази даних інформацією.

# Пошук тем по імені викладача або назві теми

Для зручного вибору теми, користувачам пропонується можливість пошуку роботи по її назві або імені викладача, якому ця тема належить. Оскільки багато студентів мають попередні домовленості з викладачами, їх не цікавлять інші теми, тому, щоб швидше знайти потрібну, вони можуть скористатись такою функцією.

Пошук здійснюється за допомогою запиту до бази даних і перевірки відповідних полів(Додаток В.3). Роботи, які відповідають критеріям пошуку з’являються після автоматичного оновлення сторінки.

# Фільтрування тем

Для зручного користування системою, була впроваджена система фільтрів. Користувачі мають змогу фільтрувати роботи по декількох властивостях: кафедра(лише з факультету користувача), яка пропонує роботу, галузі теми(можливий вибір декількох), наукові інтереси викладача(можливий вибір декількох), статус запису(вже зайняті, або навпаки – вільні теми).

Фільтрування відбувається шляхом утворення множин результатів по кожному з фільтрів, а потім їх перетину одне з одним для пошуку спільних результатів(Додаток В.4).

# Відправка сповіщень на пошту

Користувачі мають змогу отримувати сповіщення на пошту при зміні статусу запису(запис або випис нового студента, підтвердження запису викладачем, тощо). Сповіщення приходять на основну пошту користувача, одразу після зміни статусу запису. Для відправки сповіщень було створено спеціальну пошту: [naukma.recording@gmail.com](mailto:naukma.recording@gmail.com), саме звідси користувачам приходять сповіщення.

Для відправки повідомлень на пошту була використана бібліотека smtplib, яка вважається основним модулем для роботи з поштою. Для відправки повідомлення методу потрібно надати доступ до своєї пошти, повідомлення та пошту отримувача(Додаток В.5).

# Збір статистики за минулі роки

Система передбачає перегляд статистики минулих років у форматі «викладач-тема-студент-рік виконання». Дана статистика знаходиться на сторінці «Інформація» і оформлена у вигляді таблиці, яка розгортається при натисканні.

Статистика збирається по сутностям бази даних у такому порядку:

1. Пошук записів на теми, що мають статус «Підтверджено викладачем»
2. Пошук викладачів, яким належать дані теми
3. Пошук студентів, за якими вони закріплені
4. Побудова таблиці

# Висновки

Описані вище алгоритми та методи програмування впроваджені та активно використовуються в робочому прототипі системи. Система володіє всім функціоналом, визначеним у розділі 1 та безпосередньо у вимогах до її створення, дозволяє здійснювати запис та випис, переглядати відкриту інформацію та здійснювати інші операції відповідно до ролі користувача.

# Висновки

Після проведеної роботи стає зрозумілим, що процедура запису на курсові та кваліфікаційні роботи не така проста, як здається на перший погляд, адже має купу нюансів та можливих проблем, пов’язаних, наприклад, з навантаженням викладача, переслуховуванням предмету, тощо.

Дослідивши предметну область, провівши опитування та анкетування, я дійшов до висновку, що дана АІС матиме попит. Ідея створення такої системи є досить цікавою, адже вона допомагає вирішити актуальну проблему – «ручний» запис студентів на курсові та кваліфікаційні роботи, полегшити роботу методистів та викладачів, автоматизувати процедуру запису та випису.

Прототип системи розроблений за архітектурою MVC, яка передбачає поділ роботи системи на три логічні рівні: Model, View, Controller. Дана архітектура є найбільш зручною в реалізації та розумінні для подібних систем, адже дозволяє легко змінювати бізнес-логіку застосунку, або окремих його частин.

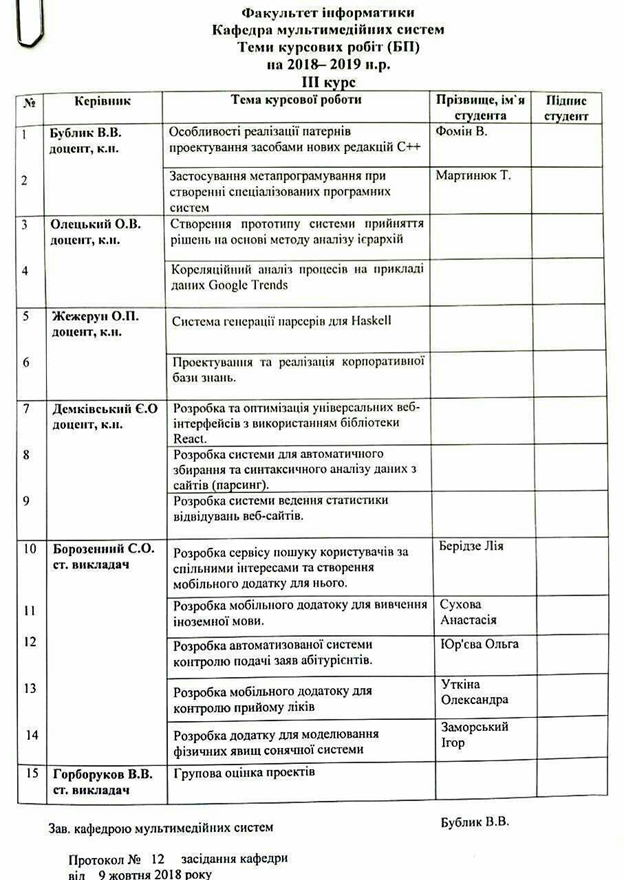
Створення такої системи дозволяє пройти весь процес становлення застосунку: від визначення вимог до запуску в експлуатацію. Виконана мною робота, допомогла зрозуміти деталі цього процесу, надала досвід не лише у програмуванні подібних систем, але й у дослідженні предметної області та створенні концептуальної моделі.

# Список Використаної Літератури

1. Положення про організацію освітнього процесу в НаУКМА[Електронний ресурс] – Режим доступу до ресурсу: <https://www.ukma.edu.ua/index.php/about-us/sogodennya/dokumenty-naukma/cat_view/1-dokumenty-naukma/12-normatyvna-baza-naukma/6-systema-zabezpechennia-iakosti-osvitnoi-diialnosti-ta-iakosti-vyshchoi-osvity/8-normatyvni-dokumenty-naukma/9-polozhennia-naukma?start=10>
2. Постанова Кабінету Міністрів України про перелік галузей знань і спеціальностей, за якими здійснюється підготовка здобувачів вищої освіти[Електронний ресурс] – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/266-2015-%D0%BF#n11>
3. Документація для Django framework[Електронний ресурс] – Режим доступу до ресурсу: <https://docs.djangoproject.com/en/2.2/>
4. Документація для LESS[Електронний ресурс] – Режим доступу до ресурсу: <http://lesscss.org/features/>
5. Документація для Bootstrap[Електронний ресурс] – Режим доступу до ресурсу: <https://getbootstrap.com/docs/4.1/getting-started/introduction/>
6. Документація для авторизації за допомогою Office 365[Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/office/office-365-management-api/get-started-with-office-365-management-apis>
7. Google-Академія[Електронний ресурс] – Режим доступу до ресурсу: <https://scholar.google.com.ua/intl/en/scholar/about.html>

# Додаток А

Бланк запису на курсову роботу



# Додаток Б

Співбесіди з потенційними користувачами системи

**Інтерв’юер(І)**

**Студент(С)**

№1

**І: Зараз для того щоб записатися на дисципліну необхідно розписатися у бланку на кафедрі. Багато хто завчасно домовляється з викладачами, займаючи собі місце завчасно та обговорюючи тему. На скільки це зручно для тебе, та скільки ти часу витратив на цю процедуру?**

С: Насправді я досі так і не записався, не кажучи про те, щоб домовитися з викладачем завчасно. Я постійно забуваю зайти на кафедру, а коли згадую, в мене немає часу на це.

**І: Як ти дивишся на ідею створення системи електронного запису на курсову роботу, як на альтернативу сьогоднішньому способу.**

С: Думаю це було б дуже зручно, бо я міг би записатися в будь-який момент з будь-якого місця, не підлаштовуючись під години роботу деканату, свої пари і таке інше.

**І: Добре, Уяви, що така система існує. На твою думку, які функції та можливості вона повинна була б мати?**

С:Хотілось би мати можливість перегляду минулорічних тем та оцінок, відфільтрувати і оцінити в кого які теми, бали. Думаю, це б спонукало викладачів виконувати свою роботу більш якісно, створило конкуренцію між ними.

**І:Як було б зручно авторизуватись в системі?**

С:Найзручніший, як на мене, це варіант зі студентським квитком, як це було в записі на предмети. Ідеально, щоб реєстрація на всі ці системи проходила по одному принципу, адже не зручно мати купу логінів/паролей для кожної могилянської системи.

№2

**І: Зараз для того щоб записатися на дисципліну необхідно розписатися у бланку на кафедрі. Багато хто завчасно домовляється з викладачами, займаючи собі місце завчасно та обговорюючи тему. На скільки це зручно для тебе, та скільки ти часу витратив на цю процедуру?**

С: Не скажу, що багато часу, але все ж довелося цим завчасно зайнятися, домовитись з викладачем, тому що під час запису вже б не лишилося місця у тих, з ким я хотіла працювати.

**І: Як ти дивишся на ідею створення системи електронного запису на курсову роботу, як на альтернативу сьогоднішньому способу.**

С: Не знаю, як би це вирішило проблему завчасної домовленості, але всеодно не потрібно було б йти зайвий раз в деканат.

**І: Один з твоїх колег сказав, що хотів би бачити реєстрацію за допомогою студентського квитка для такої системи. Що про це думаєш ти?**

С: Як на мене це не зовсім гарна ідея, бо є така обставина як “втрата квитка”, і що тоді робити. Новий квиток робиться досить довго, а коли ти його отримуєш, то розробникам потрібно буде оновити дані в системі про тебе.

**І: Окей, який варіант ти вважаєш найкращим?**

С: Як на мене, було б непогано використовувати дані з my.ukma.edu.ua, адже там зареєстрований кожен студент і тому не має виникнути жодних проблем .

№3

**І: Зараз для того щоб записатися на дисципліну необхідно розписатися у бланку на кафедрі. Багато хто завчасно домовляється з викладачами, займаючи собі місце завчасно та обговорюючи тему. На скільки це зручно для тебе, та скільки ти часу витратив на цю процедуру?**

С: Хвилин 5 на перерві, щоб сходити. Я не домовлявся з викладачем завчасно, просто обрав собі тему на кафедрі з тих, що лишилось.

**І: Як ти дивишся на ідею створення системи електронного запису на курсову роботу, як на альтернативу сьогоднішньому способу.**

С: Чому ні.

**І: Уяви, що така система існує. На твою думку, які функції та можливості вона повинна була б мати?**

С:Як на мене, можна додати можливість автоматичного(рандомного) запису студентів, можливо залишились місця у викладачів, яких я не знаю, або на теми, які мені не дуже цікаві і тому, щоб не обирати самому, хай система запише мене на будь-який можливий варіант.

№ 4

**І: Зараз для того щоб записатися на дисципліну необхідно розписатися у бланку на кафедрі. Багато хто завчасно домовляється з викладачами, займаючи собі місце завчасно та обговорюючи тему. На скільки це зручно для тебе, та скільки ти часу витратив на цю процедуру?**

С: Просто прийшов і записався, нічого складного.

**І: Як ти дивишся на ідею створення системи електронного запису на курсову роботу, як на альтернативу сьогоднішньому способу.**

С: Мені здається, що ця система не потрібна, тому що це не так важко сходити на кафедру і заповнити бланк.

**І: Але ж ми йдемо в світ цифрових технологій, навіщо заповнювати бланки і кудись йти, якщо все можна зробити онлайн?**

С: Ну, якщо ця система працюватиме стабільно і надаватиме якусь корисну інформацію і додаткові можливості, то напевно буде якийсь сенс її створення.

№5

**І: Зараз для того щоб записатися на дисципліну необхідно розписатися у бланку на кафедрі. Багато хто завчасно домовляється з викладачами, займаючи собі місце завчасно та обговорюючи тему. На скільки це зручно для тебе, та скільки ти часу витратив на цю процедуру?**

С: Ну я довго не могла визначитись з ким писати, потім ще довго шукала викладачів в університеті, в яких всі місця виявлялись вже зайнятими і я шукала інших. Тому так, довелось побігати. До того ж мені не сподобались теми, які пропонував викладач, і я потім ще відправляла на пошту йому свої варіанти, і ми узгоджували додатково.

**І: Як ти дивишся на ідею створення системи електронного запису на курсову роботу, як на альтернативу сьогоднішньому способу.**

С: Наскільки я знаю, зараз проводиться програма електронного університету, це цілком потрапляє в її рамки.

**І: Який функціонал ти хотіла би бачити в АІС ?**

С: Я б хотіла щоб прямо на сайті можна було б домовитись про зміну теми. А ідея з минулорічними темами і оцінками мені не подобається. Це, як на мене, зайве - результат твоєї роботи залежить лише від того, як ти сам на неї працював, а не який в тебе був викладач чи тема.

№6 Інтерв’ю з викладачем

**І: Провівши пріоритизацію вимог серед студентів, ми отрималии список вимог, які, на думку студентів, є найважливішими для реалізації. На вашу думку, чи є якісь з них зайвими?**

**В:** Як на мене пункт про автоматичний запис реалізовувати не варто, адже це неправильно з методичного боку і порушує права викладачів та студентів. Також зайвою вважаю реєстрацію через студентський квиток, коли є можливість зареєструватись через Office 365. Не зовсім зрозумілим є пункт про навчальні матеріали, адже не зрозуміло хто їх має надавати: викладач, чи можливо методист.

**І: Мається на увазі можливість викладача поширювати студенту покликання на різноманітні ресурси, які безпосередньо стосуються теми роботи.**

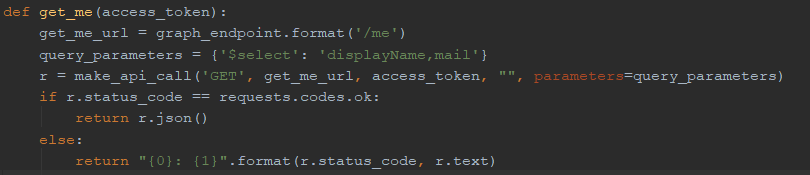
**В:** Можливо, це буде корисно для поширення вимог до роботи, її оформлення.

**І: Можливо у вас є свої ідеї, які можуть бути корисними для такої системи?**

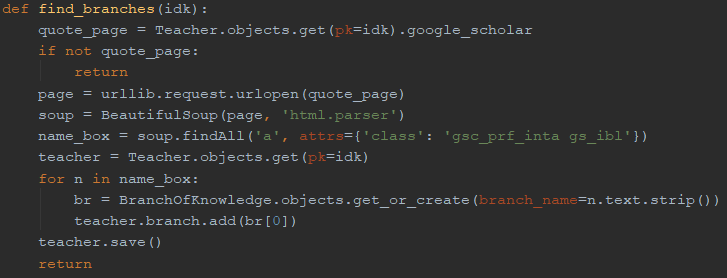
**В:** Хотілося б отримувати сповіщення про дедлайни, а також мати можливість переглянути теми попередніх років, які можна продовжити, до початку запису.

# Додаток В

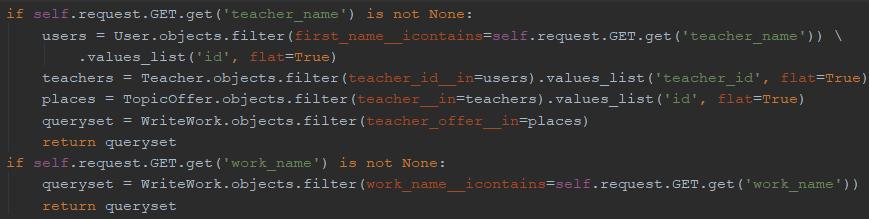
1. Функція доступу до Office 365 для отримання інформації про користувача:



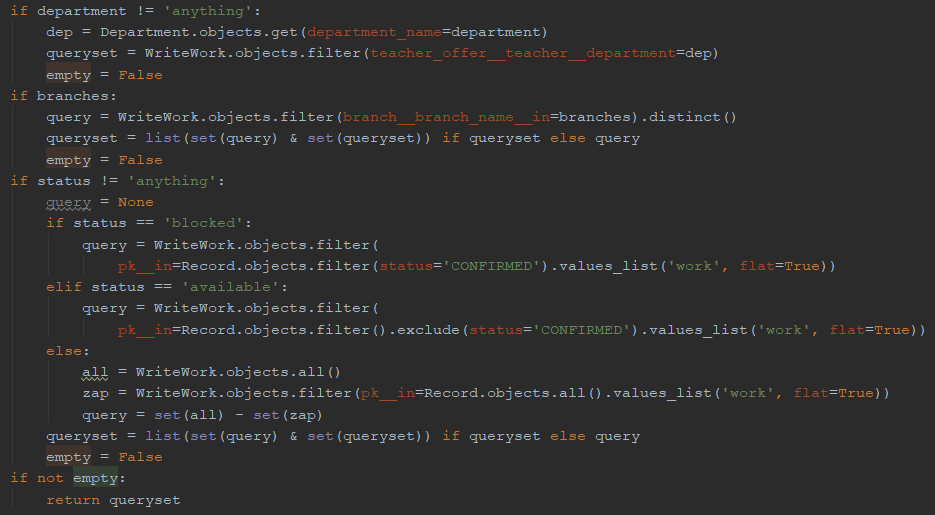
1. Функція парсингу сторінки google scholar та додавання знайдених галузей до бази:



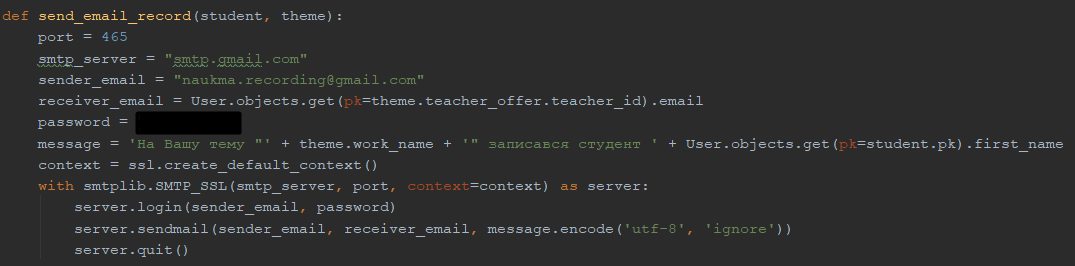
1. Функція пошуку тем по назві або імені викладача, що її пропонує:



1. Фільтр тем по різним властивостям:



1. Відправка повідомлення викладача про запис певного студента на його тему:



# Додаток Г

Файл з константами

