НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри інформаційних
систем і технологій
Швиденко М.З.
_____2025р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему

Організація роботи з базою даних користувачів в гетерогенній інформаційній системі НУБіП

Спеціальність 126 “Інформаційні системи та технології”

Гарант освітньої програми _____ Мокрієв М.В.

Керівник кваліфікаційної роботи _____ Мокрієв М.В.

Виконав _____ Шарандак Андрій Володимирович

Київ 2025

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Завідувач кафедри інформаційних систем і технологій

_____ _Швиденко М.З.___
підпис ініціали та прізвище

__ _____ 2025 р.

ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи

студенту(ці) _____Шарандака А.В._____

Спеціальності 126 “Інформаційні системи та технології”

1. Тема роботи: **«Організація роботи з базою даних користувачів в гетерогенній інформаційній системі НУБіП»**

Затверджена наказом ректора від 16.12.2024р. №2245С

2. Термін подання завершеної роботи на кафедру – ___06.2025р.

3. Вихідні дані Розробити підходи до управління користувачами в інформаційній системі з гетерогенним середовищем.

4. Перелік питань, що розглядаються:

1. Теоретико-методологічні засади дослідження систем управління користувачами в різних інформаційних системах

2. Архітектурно-технологічні аспекти проєктування системи управління користувачами

3. Реалізація тестових програмних компонентів управління користувачами

5. Календарний план

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Вибір теми, планування та підготовка	25.02.2025	Завдання виконано
2	Дослідження та збір матеріалів	30.03.2025	Завдання виконано
3	Аналіз та написання	20.05.2025	Завдання виконано
4	Попередній перегляд та оцінка	27.05.2025	Завдання виконано
5	Захист бакалаврської роботи	06.2025	Завдання виконано

Керівник кваліфікаційної роботи _____ / __Мокрієв М.В., к.е.н._____
підпис ПІБ, вчене звання та ступінь

Завдання прийняв до виконання _____ / _____Шарандак А.В._____
підпис

Дата отримання завдання 15.02.2025

ЗМІСТ

ВСТУП

В сучасному цифровому середовищі вищої освіти особливого значення набуває ефективне управління інформаційними ресурсами, зокрема користувацькими даними. Інформаційні системи університетів поступово перетворюються на складні багаторівневі платформи, які об'єднують велику кількість сервісів – електронні щоденники, освітні платформи, системи документообігу, бази даних студентів і викладачів, внутрішні CRM-системи. Така архітектура створює гетерогенне середовище, в якому питання централізованого обліку користувачів, синхронізації інформації та інтеграції систем стають критично важливими для стабільної роботи та інформаційної безпеки.

В цій синергії надзвичайно важливою є організація роботи з базою даних користувачів, що дозволяє оптимізовувати доступ до послуг і забезпечувати узгоджене функціонування різнорідних систем. Зокрема, інформаційна система НУБіП є прикладом гетерогенного середовища, що має низку особливостей і викликів, пов’язаних із інтеграцією даних між підсистемами, керуванням ідентифікацією користувачів, забезпеченням цілісності, безпеки та актуальності інформації.

У дослідженні буде використано такі методи, як аналіз архітектури наявних інформаційних систем, порівняння підходів до реалізації баз даних користувачів, моделювання структури даних, проектування і розробка прототипу бази, а також тестування ефективності взаємодії окремих підсистем. Це дозволить представити цілісне бачення щодо організації роботи з користувацькими даними в умовах гетерогенної ІТ-інфраструктури.

Вибір ІС НУБіП як об'єкту дослідження обумовлено її багатокомпонентною природою, широким охопленням функцій та наявністю практичних прогалин, які потребують корекції. Отримані результати можуть бути застосовані перш за все для оптимізації роботи інформаційної системи університету, а також стати у пригоді деяким ІТ-адміністраторам, аналітикам та

розробникам які займаються впровадженням і підтримкою інформаційних рішень у закладах вищої освіти.

Актуальність теми

У добу активної цифровізації освіти всі розвинуті освітні установи переходять до хмарних технологій і дистанційного навчання з мультисервісною взаємодією, вкрай важливим є ефективне управління користувачами та їхніми даними. Розподіленість інформації, відсутність єдиного джерела правди, складність синхронізації між системами – все це створює ризики для безперебійної роботи університетської IT-інфраструктури. Тому організація роботи централізованої бази даних користувачів, що адаптована до гетерогенного середовища, є завданням з технічною і управлінською цінністю.

Мета дослідження

Розробити архітектурний та програмний підхід до організації централізованої бази даних користувачів у гетерогенній інформаційній системі НУБіП, який забезпечує інтеграцію з різнорідними підсистемами, зручне управління ідентифікацією, актуальністю даних і захистом інформації.

Завдання дослідження

- Проаналізувати архітектуру гетерогенної ІС НУБіП.
- Дослідити теоретичні підходи до організації роботи з БД користувачів.
- Визначити проблеми, пов’язані з інтеграцією облікових систем у гетерогенному середовищі.
- Сформулювати вимоги до єдиної БД користувачів.
- Розробити структуру БД та спроектувати механізми взаємодії з іншими компонентами ІС.
- Реалізувати фрагменти прототипу на технології.
- Провести тестування функціональності та ефективності рішення.

Об'єкт дослідження

Інформаційна система Національного університету біоресурсів і природокористування України як приклад гетерогенного середовища.

Предмет дослідження

Процеси організації, проєктування та реалізації бази даних користувачів в умовах гетерогенної архітектури інформаційної системи, а також механізми їхньої взаємодії з іншими модулями системи.

Структура роботи

Дипломна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Таким чином, результати роботи сприятимуть підвищенню ефективності управління даними в університеті, зменшенню дублювання інформації, спрощенню адміністрування та забезпеченню стабільної роботи ІС НУБіП.

РОЗДІЛ 1. АНАЛІЗ ГЕТЕРОГЕННОЇ ІС НУБІП ТА ПРИНЦИПІВ ОРГАНІЗАЦІЇ РОБОТИ З БАЗОЮ ДАНИХ КОРИСТУВАЧІВ

1.1. Гетерогенні інформаційні системи: поняття, структура та особливості

Сучасні умови цифрової трансформації освітніх установ передбачають ефективну організацію інформаційного середовища яку забезпечує навчальний, науковий та адміністративний процеси. Одним із ключових понять, що характеризують складність і багатофункціональність таких середовищ, є гетерогенна інформаційна система, або ж ГІС.

ГІС – це сукупність взаємопов’язаних, але технологічно та функціонально різнорідних компонентів, таких як апаратне і програмне забезпечення, бази даних, які забезпечують спільну обробку, зберігання, передавання та використання інформації. Така система об’єднує в собі елементи, створені на різних платформах, і використання різних мов програмування, баз даних, протоколів зв’язку та архітектурних рішень. В порівнянні з гомогенними системами, де всі компоненти мають однакову архітектуру та технології, ГІС характеризуються різноманітністю.

У контексті вищої освіти, ГІС часто включає електронні освітні платформи, наприклад Moodle; системи управління навчальним процесом, такі як електронні журнали, модулі реєстрації, відомості; бази даних студентів та викладачів; внутрішні портали, поштові служби, системи документообігу; аналітичні сервіси і сервіси тестування доступності.

Ключова особливість – неоднорідність, проявляється в різнорідності, різних структурах і форматах даних, відсутності єдиного джерела автентифікації користувачів і наявності декількох точок входу для взаємодії з системою.

Інформаційна система НУБіП – яскравий приклад гетерогенного середовища. Вона охоплює низку окремих платформ, які постійно взаємодіють

у процесі навчання та управління університетом. Першорядним елементом є електронна навчальна платформа elearn.nubip.edu.ua, побудована на базі Moodle, яка забезпечує онлайн-доступ до курсів з завданнями і тестами, журналів успішності та комунікації між студентами і викладачами.

Ця платформа не може працювати без інших підсистем, що в свою чергу ускладнює централізоване управління:

- Модулі електронного документообігу;
- Внутрішні акаунти Microsoft 365;
- Облікові записи для локального доступу до мережі;
- Адміністративні бази даних для деканатів та кафедр.

ГІС мають багато переваг в застосуванні, включаючи додавання нових підсистем без повного перепроєктування, їх оптимізацію для конкретних завдань, гнучку інтеграцію з іншими підсистемами. Навпроти – є і низка викликів, таких як ризик дублювання даних (студент що поновився на той самий курс), ризик безпеки (застарілі і неактивні акаунти в системі), невідповідність форматів і протоколів даних (ускладнення синхронізації), завантаженість системи в конкретні дні (велика кількість запитів між підсистемами може знизити швидкість роботи).

Для подолання цих проблем застосовуються різні підходи, зокрема введення єдиних служб автентифікації (SSO, LDAP, Active Directory), уніфіковані бази даних користувачів, API-синхронізація між підсистемами.

Потрібно враховувати, що окрім технічної складової, організація роботи з базами користувачів у навчальних закладах повинна підпорядковуватися також нормативним актам:

- Закон України “Про захист персональних даних” вимагає забезпечення конфіденційності та цілісності зібраної інформації;
- ISO/IEC 27001 – міжнародний стандарт з управління інформаційною безпекою, який рекомендується при розробці політик доступу;

- Внутрішні регламенти ВНЗ, які визначають політику створення, супроводу, деактивації акаунтів.

Таким чином, будь-яка система, яка працює з обліковими даними, має бути не лише технологічно надійною, а й юридично захищеною та документально оформленою.

1.2. Огляд архітектури інформаційної системи НУБіП та її підсистем

Інформаційна система Національного університету біоресурсів і природокористування України (НУБіП) формувалася поступово, відповідно до зростаючих потреб цифровізації навчального процесу, адміністрування та внутрішньої комунікації. Особливо стрімкий прогрес випав на спалах вірусу COVID-19 в Україні, коли всі заклади освіти, підприємства і більшість державних установ переходили на дистанційний формат. Вона не є єдиним монолітним рішенням, а представляє собою розподілену, гетерогенну багаторівневу систему, яка представляє собою ряд самостійних підсистем, що взаємодіють між собою опосередковано або частково.

Структуру ІС НУБіП можна розділити на групи підсистем:

a) навчальні або ж академічні:
   1) elearn.nubip.edu.ua (LMS Moodle) – управління курсами, завданнями, тестуванням, журналами;
   2) календарно-тематичні плани, графіки і розклади;
   3) реєстрація на вибіркові дисципліни (можливо окремо через портал);
b) адміністративні:
   1) внутрішні модулі обліку студентів (накази, академічна мобільність);
   2) система реєстрації заяв, довідок, замовлень документів (електронний документообіг);

3) викладацькі або привілейовані акаунти (керівники, деканати, навчально-методичний відділ);

c) комунікаційні:

1) сервіси Microsoft 365: Outlook, Google Meet, 365 – для корпоративної пошти, відеозв’язку, спільної роботи;

2) веб-портал nubip.edu.ua — офіційний сайт із персональними сторінками кафедр, новинами, доступом до нормативної документації;

d) інфраструктурні:

1) локальні сервери (автентифікація, резервне копіювання, хостинг сайтів);

2) служба автентифікації користувачів (можливо, OpenLDAP або власна авторизаційна система).

Підсистеми ІС НУБіП не завжди інтегровані повністю, що є характерною ознакою для гетерогенних систем. Ключові аспекти взаємодії можуть відбуваютися через ручний або частково автоматизований обмін даними через проміжні формати CSV, XML, у випадках із обмеженою технологічною сумісністю; пряме API-з’єднання REST, SOAP, web-сервіси для реального часу; використання middleware-рішень, для прикладу Enterprise Service Bus, задля трансформації даних між різними стандартами; через дублювання функціональностей. Це створює додаткове навантаження на адміністраторів, особливо під час зміни статусу студентів (відрахування, переведення, поновлення), з’являється необхідність вручну оновлювати дані в кількох підсистемах одночасно.

Система elearn.nubip.edu.ua, побудована на платформі Moodle, функціонує як автономний навчальний простір із власною базою даних. Вона зберігає повний спектр навчальних даних – від профілів користувачів до курсів, оцінювання, матеріалів та журналів присутності.

Хоча Moodle підтримує інтеграцію зі зовнішніми сервісами Microsoft і Google, в НУБіП ці можливості використовуються частково. Наразі система не

синхронізована повністю із централізованим обліком університету, тому деякі акаунти створюються вручну або через проміжні механізми імпорту.

Для подальшого розвитку потрібна поглиблена інтеграція з внутрішніми API університету, це дозволить автоматизувати більше процесів, підвищити точність даних та зменшити навантаження на адміністраторів.

1.3. Роль бази даних користувачів в інформаційній системі НУБіП та проблеми її функціонування в гетерогенному середовищі

Ключовим компонентом будь-якої сучасної інформаційної системи вважається ефективне управління користувацькими даними. В контексті гетерогенного освітнього середовища НУБіП, де співіснують десятки підсистем з різною архітектурою та функціональністю, база даних користувачів виконує роль центрального інтеграційного елемента, що забезпечує злагоджену роботу всієї цифрової інфраструктури.

В такій архітектурі база користувачів відіграє роль єдиної точки істини – тобто саме на цій основі повинні формуватись і актуалізуватись всі дані в усіх підсистемах, що стосуються кожної особи. Це дозволяє уникнути дублювання, деяких розбіжностей даних, складності в обслуговуванні та забезпечує цілісність інформаційної структури. В ідеалі, кожен підрозділ ІС НУБіП повинен взаємодіяти з БД як з основним джерелом облікових даних, використовуючи інтеграційні шари, такі як API, шини даних або ETL-процеси. Також можливо застосовувати прямі запити, як базову функцію звернення.

Таблиця 1.1 - Функції та значення бази даних користувачів

Функція	**Значення**	**Додаткові аспекти**
Ідентифікація та автентифікація	Забезпечення можливості входу до системи, розмежування доступу на рівні ролей	Підтримка мультифакторної автентифікації, інтеграція з LDAP

Управління персональними даними	Зберігання та оновлення ПІБ, академічного статусу, електронної пошти та логінів	Відповідність вимогам GDPR, можливість самоконтролю для користувачів

Продовження таблиці 1.1

Рольова авторизація	Надання доступу до різних модулів залежно від повноважень викладача і студента	Гнучкі налаштування ролей, підтримка тимчасових доступів
Синхронізація даних між підсистемами	Передача або обмін обліковою інформацією між внутрішніми реєстраційними сервісами, Microsoft 365 та elearn	Автоматизовані сценарії синхронізації, обробка конфліктів даних, API
Ведення журналів активності	Фіксація подій входу, взаємодії та зміни даних	Зберігання логів для аналізу безпеки і можливість аудиту, унеможливлення незворотніх змін
Інтеграція з зовнішніми сервісами	Забезпечення взаємодії з платформами електронного навчання, хмарними сервісами	Підтримка стандартів OAuth, SAML, SCIM для безпечного обміну даними
Масштабованість та резервування	Підтримка зростаючої кількості користувачів і сервісів	Реалізація кластерних рішень, регулярне резервне копіювання, планування навантаження

Ідеальна модель БД дає можливість витягувати однин запис із всєю потрібною інформацією для всіх підсистем, вона функціонує як єдине джерело істини.

Сьогоднішні реалії роботи бази користувачів в інформаційній системі НУБіП – функціонування з можливим фрагментуванням цифрового середовища.

Навчальна платформа Moodle (elearn.nubip.edu.ua) має повноцінну власну базу користувачів з усіма необхідними атрибутами для навчального процесу. Однак ця база існує ізольовано, адже додавання нових студентів чи викладачів потребує ручного введення даних, навіть якщо ця особа вже буда зареєстрована в інших системах університету.

Система вибору дисциплін працює як окремий модуль з власною логікою авторизації. Наприклад, в багатьох студентів виникає проблема нерозуміння приналежності до якихось курсів, і якщо це не було вчасно помічено, виникають додаткові труднощі в кінці кожного семестру. Доводиться проходити повторну реєстрацію бентежити викладача.

Microsoft 365 (офісні та поштові сервіси) – ця платформа хоча і використовується як єдина поштова система для навчання, її інтеграція з Moodle та іншими сервісами обмежена. Зміна пароля в Office 365 не впливає на доступ до elearn.

Основними проблемами функціонування БД користувачів у гетерогенному середовищі можна вважати вразливість до помилок через людський фактор, можуть бути дублікати або відсутні ролі при ручному додаванні користувачів; також порушення принципів інформаційної безпеки через застарілий, неоновлений акаунт – це створює загрозу несанкціонованого доступу; вірогідне зростання кількості користувачів та сервісів може спричинювати проблеми підтримки системи без єдиного центру обліку.

Не менш серйозними є приховані ризики, створені через неповну або несвоєчасну синхронізацію даних. В основному ці ризики є для:

- **академічної доброчесності** – студенти можуть мати кілька акаунтів для тестування;
- **аналітики** – збої під час формування єдиної статистики по відвідуванню, успішності;

- **звітності перед МОН** – при порушенні відповідності між різними джерелами даних.

На сучасному етапі багато організацій, як от провідні університети Європи та США, стикаються з проблемою фрагментації облікових записів користувачів у різних підсистемах. Це ускладнює адміністрування, разом із зниженням безпеки відбувається збільшення навантаження на відділ IT. Проте всі проблеми призводять до ефективних рішень, що дозволяють оптимізовувати процеси без шалених витрат.

Одним з цих рішень є інтеграція із системами автентифікації, такими як Azure AD або Google Workspace. Користувачі отримають змогу входити в усі сервіси за допомогою одного логіна та пароля. Це перш за все підвищує безпеку, унеможливлює використання застарілих і слабких паролів. Зручність також є вагомим фактором.

Наступним рішенням є створення централізованої адмінпанелі для управління користувачами. Воно значно спрощує роботу адміністраторів через можливість створювати облікові записи, блокувати і видаляти доступи, налаштовувати різні права для різних сервісів. Швидке оновлення інформації вкрай важливе для довготривалого навчання онлайн.

В рішенні підключення до зовнішніх платформ, наукових база даних чи міжуніверситетських систем можна використовувати стандарти Shibboleth, OAuth 2 або OpenID Connect. Це дозволяє без збереження паролів та інших даних безпечно делегувати авторизацію.

Узагальнюючи все вище написане, можна ствердити, що архітектура інформаційної системи може характеризуватися фрагментованістю, відсутністю централізованого механізму обліку та численними дублюваннями акаунтів у різних підсистемах. З великою вірогідністю це буде створювати значні труднощі в адмініструванні, супроводі даних, забезпеченні безпеки та якості цифрових сервісів. Все більш актуальним стає запровадження єдиної централізованої БДК з доступом через API, яка буде єдиним джерелом істини для всієї інформаційної інфраструктури. Умовна архітектурна модель системи представлена в додатку А

– вона ілюструє взаємодію ключових підсистем через єдиний шлюз, який забезпечує синхронізацію, актуальність і узгодженість всіх даних користувача. Така модель є бажаним і технологічно обґрунтованим напрямом розвитку інформаційних систем університетів.

РОЗДІЛ 2. МОДЕЛЮВАННЯ ВДОСКОНАЛЕНОЇ ОРГАНІЗАЦІЇ РОБОТИ З БАЗОЮ ДАНИХ КОРИСТУВАЧІВ

2.1. Формування функціональних вимог і проєктування інтелектуальної взаємодії з базою даних користувачів

В умовах зростаючої складності цифрової інфраструктури вищої освіти і стрімкого розвитку електронного адміністрування, звичайні моделі роботи з базами даних користувачів втрачають ефективність. Вони не здатні гнучко реагувати на багаторівневі потреби сучасних освітніх закладів, де кожен користувач може взаємодіяти з десятками різних систем, кожна з яких містить власну логіку обліку, політику доступу, і власні технологічні обмеження. Через це проєктування вдосконаленої системи управління обліковими записами має брати початок не з технічної реалізації а з розуміння того, що саме потрібно адміністраторам, керівникам підрозділів, інтеграторам і звичайно ж користувачам.

Користувацький обліковий запис більше не є простою одиницею в таблиці бази даних – це ключова теза сучасного підходу. Це динамічне цифрове представленням особи в інформаційному просторі університету повинне підтримувати зміну ролей, адаптацію до академічного статусу, доступ обмежений або тимчасовий, історію взаємодії і здатність до міжсистемної автентифікації. Університет як живий організм, у якому щодня змінюються статуси студентів, з’являються нові викладачі, відбувається оновлення нормативних регламентів і ввід нових підсистем. Якщо база користувачів не відповідає цим динамічним умовам, вона стає гальмом у цифровому розвитку, а не високотехнологічним інструментом керування.

Для того щоб система взаємодії з обліковими записами була справді ефективною, вона має враховувати три головні виміри: індивідуальну гнчкість, організаційну ієрархію та архітектурну відкритість. Індивідуальна гнучкість передбачає можливість багатофакторного опису користувача через призму атрибутів: факультет, курс, тип навчання, додаткові повноваження, участь у

проєктах, тимчасова мобільність. Це дозволяє формувати не рольову, а атрибутивну авторизацію – підхід, впроваджений в цифрових кампусах передових університетів Європи. Для прикладу, студент третього курсу спеціальності “Інформаційні системи і технології” може мати базову роль студента, але одночасно виконувати функції лаборанта кафедри і бути учасником програми подвійного навчання в іноземних ВНЗ. Усе це має бути відображено в його обліковому записі без дублювання.

Організаційна ієрархія передбачає певний набір делегованих повноважень у керуванні акаунтами, що повинні бути в кожному рівні академічної та адміністрації університету. Завідувач кафедри має бачити структуру доступів своїх викладачів і студентів, мати право ініціювати зміну ролі або створення тимчасового акаунту. Куратор студентської групи повинен мати змогу подати запит на оновлення даних про студента, який змінив паспорт або мобільний телефон. Адміністратор курсу в LMS повинен мати доступ до логів активності і дій учасників. Для цього є необхідність впроваджувати систему гнучкого делегування: не на рівні “дозволено / заборонено”, а на рівні дозволених дій в чітко визначеному контексті.

Третій вимір – архітектурна відкритість, стосується побудови бази даних користувачів як платформи, що виступає єдиним джерелом правди для всієї екосистеми цифрових сервісів. Вона має підтримувати обмін даними через стандартні протоколи, як от REST API, GraphQL, OAuth2, бути придатною для взаємодії з постачальниками різних освітніх сервісів, державними реєстрами і академічними хмарами. Ключовим є принцип: будь-яка система, яка потребує інформацію про користувача, має отримувати її не з локальної копії, а з API централізованої БДК – синхронно або асинхронно.

Побудова такої моделі потребує відмови від статичних схем користувача і реалізації динамічної моделі циклу облікового запису, яка описує не лише момент створення акаунта, а весь його шлях в системі: від генерації після зарахування до архівації або передачі у випускні реєстри після завершення навчання. Детальне представлення моделі знаходиться в Додатку Б. У рамках

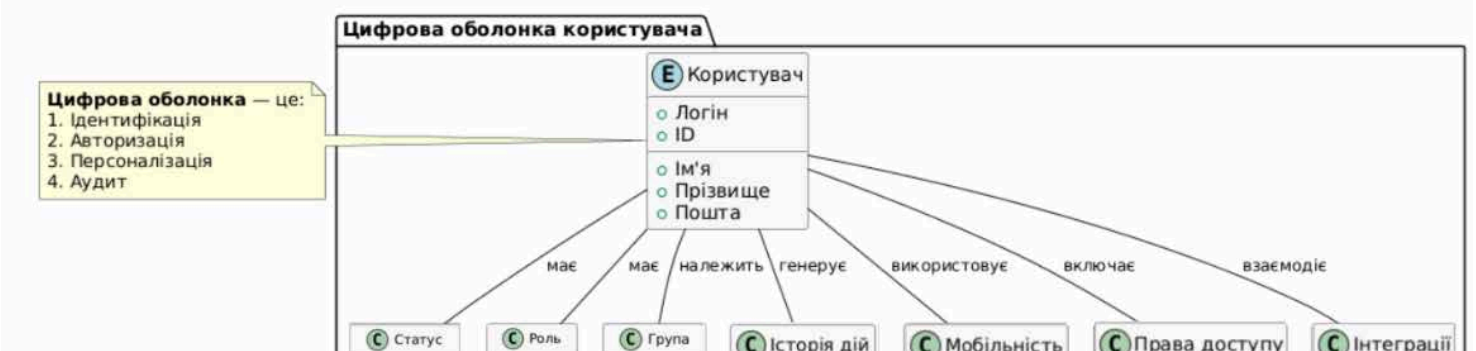
цієї моделі кожен акаунт повинен містити набір обов'язкових атрибутів, які динамічно змінюються відповідно до подій — переведення на інший курс, зміна форми навчання, академічна відпустка, стажування за кордоном. Така динамічна структура даних дозволяє точно керувати доступом, формувати автоматизовані сценарії адміністрування, забезпечувати відповідність внутрішнім регламентам та зовнішнім вимогам щодо обліку й звітності.

У цій концепції важливу роль відіграє сценарна логіка взаємодії з обліковими записами. Кожна дія, що стосується акаунта, повинна мати уніфікований сценарій із визначеними кроками, повноваженнями та точками логування. Як зазначено в Додатку Б, при зарахуванні нового студента система повинна автоматично активувати послідовність: створення акаунта в БДК, генерація email-адреси, додавання до групи, ініціалізація профілю в LMS, активація доступу до Microsoft 365, логування подій. Так сценарії мають бути конфігурованими через шаблони, адаптовані до типів користувачів, підрозділу чи факультету. Такий підхід спрощує і автоматизовує рутинні дії, адаптує логіку до організаційної специфіки без втручання в код базової системи, що є важливим.

Окремо слід наголосити на потребі у реєстрі змін – механізмі, який дозволяє фіксувати кожну зміну облікового запису: від умов і наслідків змін до IP-адреси "редактора". Це критично важливий момент для відповідності політикам інформаційної безпеки. В сучасному уявленні БДК без централізованого аудиту – це система з втраченим контролем.

В результаті всіх описаних підходів формується інтелектуальна цифрова оболонка користувача, зорбажена на рисунку 2.1 .

Рисунок 2.1 – Схема цифрової оболонки користувача



Вона здатна адаптуватися до змін контексту, інтегруватися в мультиплатформове середовище, забезпечувати якісну і видиму взаємодію з підсистемами і бути контрольованою в усіх точках життєвого циклу.

Впровадження подібного концептуального підходу в умовах НУБіП відкриває нові горизонти в адмініструванні облікових записів і в побудові єдиного цифрового середовища, здатного масштабуватись, бути інтероперабельним з різними платформами, відповідати вимогам кібербезпеки і гарантувати якість під час користування. Ця модель створює основу для впровадження автоматизованих процесів управління доступом, полегшує аналітичну обробку інформації, сприяє зменшенню кількості технічних збоїв.

Підсумовуючи, моделювання вдосконаленої організації бази даних користувачів у гетерогенній інформаційній системі університету – це не просто технічне завдання студента, а стратегічний напрямок цифрового розвитку ЗВО.

2.2. Формування архітектури інтеграції нових функцій у наявну ІС

Реалізація концепції інтелектуального облікового запису, що описана в попередньому підрозділі, вимагає чітко спроєктованої архітектури взаємодії між підсистемами, яка забезпечить як технічну сумісність так і логічну цілісність усієї інформаційної системи університету. У випадку НУБіП, де співіснують різнопланові рішення такі як Moodle (elearn.nubip.edu.ua), Microsoft 365, системи документообігу, сервіси розкладу та зовнішні освітні портали, питання інтеграції не може бути розв’язане засобами одноразової синхронізації або дублювання даних. Потрібна системна платформа – архітектура інтеграції, яка дозволить централізовано керувати обліковими записами та розгортати нові функції без переривання всієї роботи системи.

У центрі такої архітектури має знаходитися централізована БДК, яка виконуватиме роль не тільки сховища, а й координаційного ядра цифрової ідентичності. Усі підсистеми повинні взаємодіяти з нею через спеціально спроєктований API-шлюз, який виступає посередником між БДК та

сервісами-споживачами даних. Це дозволить забезпечити контрольований доступ, стандартизовану форму запитів, уніфікацію структури відповідей і – що особливо важливо – централізовану систему логування, моніторингу та обмеження навантаження.

З технічної точки зору, така архітектура має відповідати принципам модульності, розподіленості, гнучкості та масштабованості. Вона має дозволяти поступове підключення нових систем, як от наприклад системи реєстрації на події, наукові портали, ресурси бібліотеки, без необхідності повної перебудови всієї інфраструктури. Кожна підсистема виконує свою локальну функцію, але всі вони звертаються до єдиного джерела даних про користувача через уніфікований протокол.

Особливу роль у цій моделі відіграє сервіс авторизації (SSO), який забезпечує єдину точку входу до всіх систем. Цей компонент заснований на протоколах OAuth2, SAML 2.0 і виконує і функцію автентифікації і централізованої авторизації – тобто, визначає, які права доступу має користувач у конкретній підсистемі спираючись на його атрибути, що зберігаються у БДК. Це дозволяє реалізувати атрибутивну модель доступу, скорочено ABAC, у якій доступ визначається як роллю, так і статусом, курсом навчання, організаційною приналежністю, результатами верифікації.

Крім основних компонентів – БДК, API-шлюзу та SSO – ефективна інтеграційна архітектура передбачає створення середовища подій і обробників змін, відомих як event-driven logic. У великих університетських інформаційних системах, умови яких передбачають велику кількість акаунтів, приблизно 25-30 тисяч, надзвичайно важливою є можливість реагування системи на зміни у статусі користувача в режимі реального часу. Наприклад, переведення студента з денної на заочну форму навчання не повинно обмежуватись лише зміною запису в базі – це має автоматично надати йому доступ до розкладу і прибрати колишній, перевести курси в LMS у режим "самостійної навчання", призупинити доступ до фізичних ресурсів, запустити генерацію нових звітів для деканату. Саме тому в рамках інтеграції є необхідність передбачити підтримку

механізмів publish–subscribe або webhook-обробників, які дозволять ініціювати каскад змін через одне центральне оновлення.

Ще один важливий аспект – захист даних і контроль доступу на рівні міжсистемної взаємодії. У більшості традиційних систем виклики API реалізовані на рівні перевірки токена доступу, однак у запропонованій архітектурі доцільніше буде впроваджувати контекстну авторизацію, яка враховує не лише ідентифікацію клієнта, а й мету запиту, історію доступу та статус користувача. Якщо модуль LMS запитує атрибут «email» студента, система повинна дозволити це лише коли курс, до якого звертається викладач, дійсно містить цього студента як учасника. Такий рівень контролю реалізується через access control middleware, вбудований в API-шлюз.

Зважаючи на потенційні ризики дублювання акаунтів, логічно впровадити унікальний глобальний ідентифікатор користувача, який буде незмінним упродовж усього життєвого циклу – навіть у разі зміни ПІБ, email чи академічного статусу. Цей ідентифікатор використовується для прив’язки до акаунтів у всіх підсистемах і також для журналізації подій, включно з даними після архівації запису. НУБіП на даний час використовує такі ідентифікатори у всій системі, що робить його досить технологічно прогресивним університетом. Додатково до цього, доцільно впровадити внутрішню карту зв’язків: таблицю фіксації взаємозв’язків між акаунтом та об’єктами ІС (групи, кафедри, курси, модулі, події, навчальні матеріали), що дозволяє з високою точністю відстежувати вплив змін і оцінювати наслідки навіть у разі деактивації акаунта або зміни його ролі.

Не менш важливою частиною інтеграційної архітектури є підсистема оновлення та міграції даних. У практиці НУБіП, як і в інших великих університетах, часто виникають ситуації, коли необхідно оновити великі масиви облікових записів – при зміні форматів логінів, оновленні політик безпеки, введенні нового набору атрибутів. Для цього необхідно передбачити окремий механізм етапного розгортання змін (staged deployment), з яким зміни тестуються на контрольній вибірці акаунтів, після чого поступово

застосовуються до всіх інших із можливістю відкату (rollback). Це дозволяє уникнути глобальних збоїв і гарантувати стабільність роботи ІС.

Щоб повною мірою забезпечити інтеграцію, архітектура повинна передбачати вбудовані механізми зворотного зв'язку з підсистемами. Наприклад, LMS після виконання певної дії користувача повинна мати можливість надіслати сигнал до БДК або адміністративного модуля про зміну статусу і ця подія має запускати відповідну логіку (перерахунок балів, активація доступу до сертифікатів, зміна значень профілю). Такий підхід створює інтеграційну двосторонність – дані не лише читаються з БДК, а й формуються та збагачуються зовнішніми подіями.

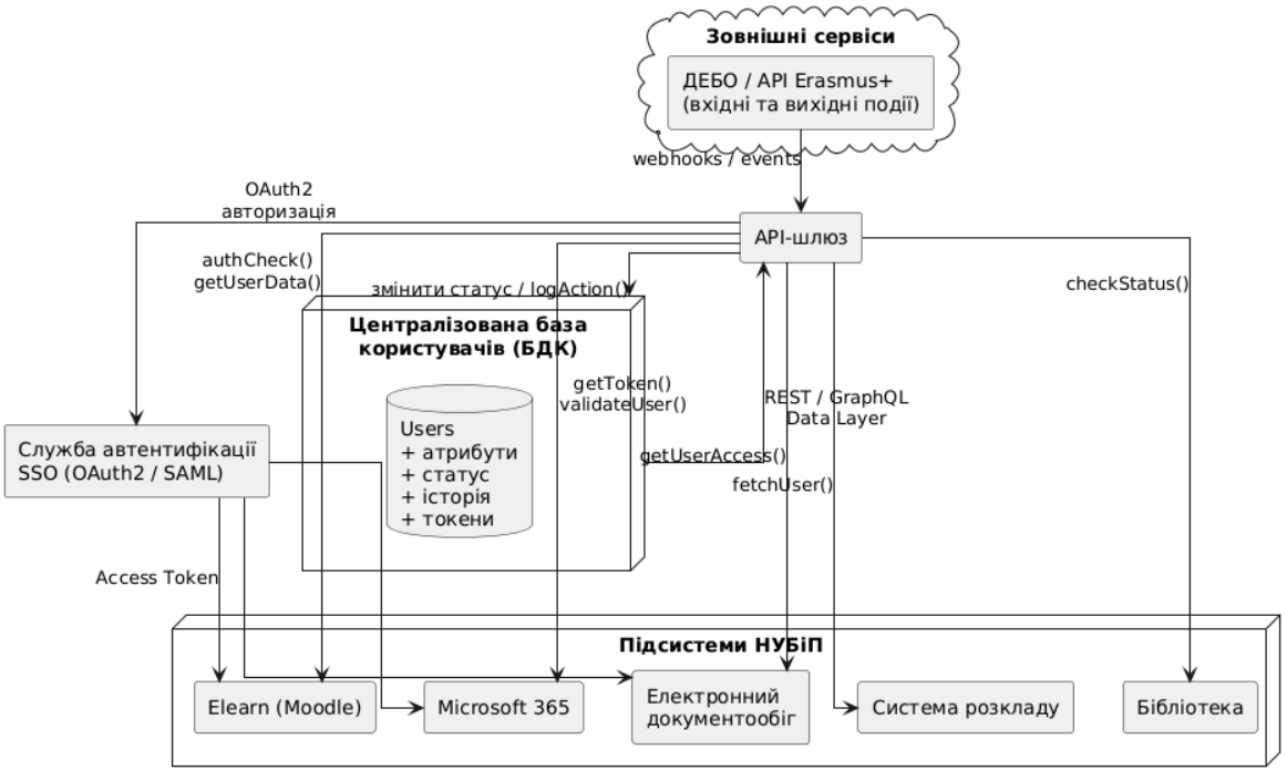
Важливим моментом є те, що ця модель повинна бути технологічно агностичною, тобто не залежати від конкретної реалізації платформи. Вона має реалізовуватись як RESTful API із захищеним доступом, що здатен обробляти як синхронні запити, так і асинхронні події (запуск сценарію переведення на інший курс, після чого відбувається зміна прав доступу у LMS, оновлення у календарях, відправка email-повідомлень).

Впровадження нових функцій у межах цієї архітектури повинно здійснюватись через розширення сценарного API. Додавання модулю для подання запитів на зміну статусу користувача не вимагає зміни всієї логіки, а лише створення нового ендпоінту в API, який взаємодіє з базою, запускає відповідні тригери, фіксує зміни і відображає їх у профілі користувача та в пов'язаних з ним підсистемах.

Таким чином, архітектура інтеграції має бути побудована за принципом "єдине ядро – множинні точки доступу". Централізована база забезпечує консистентність даних, API-шлюз – контроль доступу, SSO – безпечну авторизацію, а підсистеми – спеціалізовану реалізацію окремих сервісів. Така архітектура дозволяє забезпечити повну прозорість дій, централізований аудит, гнучкість в реалізації політик доступу, зменшення дублювання акаунтів та скорочення витрат на адміністрування. Вона здатна не лише відповідати сучасним вимогам цифрової трансформації освіти а й формувати конкурентну

перевагу університету у сфері цифрової культури. Детальний вигляд моделі інтеграції відображений на рисунку 2.2 , що підсумовує весь підрозділ.

Рисунок 2.2 – Архітектурна модель інтеграції



2.3. Створення моделей даних, інтерфейсів і сценаріїв використання

Процес створення інформаційної системи, що орієнтується на інтегроване управління обліковими записами користувачів, передбачає не лише визначення архітектури чи логіки взаємодії, а й чітке формалізоване описання внутрішньої структури даних, зовнішньої взаємодії через інтерфейсні компоненти та типових сценаріїв використання, що охоплюють повний життєвий цикл акаунта в системі. Саме така триєдність – дані, інтерфейси й сценарії – визначає

життєздатність інформаційної системи та її здатність бути зрозумілою, масштабованою, ефективною та зручною для користувача та адміністратора.

В межах ініційованої моделі управління цифровими ідентичностями користувачів, моделювання даних виконує роль структурної основи, виступає засобом забезпечення цілісності, достовірності та відповідності даних між підсистемами. Побудова моделі даних передбачає формування логічної схеми сутностей, які описують обліковий запис, його властивості, історію змін, зв'язки з іншими об'єктами, такими як академічна група, викладач, навчальний курс, а також можливість уніфікованого опису прав доступу та дій. Ця модель повинна підтримувати історію змін, багатозначність значень, а також бути орієнтованою на атрибутивну авторизацію, де кожен запис містить деякий набір властивостей, що визначають його поведінку в різних контекстах.

Ключовим підходом у проєктуванні такої моделі виступає концепція гнучких метаданих. Замість жорстко структурованої фіксації полів користувача краще реалізувати гібридну структуру, в якій основні дані зберігаються у фіксованих атрибутах – ідентифікатор, email, ПІБ, а контекстні властивості у вигляді розширюваного набору ключ-значення. Приклад створення простої SQL таблиці атрибутів в лістингу 3.1:

Лістинг 3.1

```
CREATE TABLE user_attributes (
attr_id SERIAL PRIMARY KEY,
user_id INTEGER REFERENCES users(user_id),
attribute_name VARCHAR(100),
attribute_value TEXT,
valid_from DATE,
valid_to DATE
```

Це забезпечить можливість адаптації до змін без необхідності модифікації всієї структури БД, що критично важливо для систем з високим еволюційним рівнем. Наприклад, введення нового поля – тип стажування,

напрям мобільності чи мови навчання – не вимагає структурної міграції а реалізується додаванням нового атрибуту в окрему таблицю розширень.

Щодо проєктування інтерфейсів, то на цьому етапі йдеться не лише про зовнішній вигляд системи а про логіку і послідовність дій, які користувач виконує під час взаємодії з обліковим записом. Це включає як інтерфейси кінцевого користувача, так і інтерфейси адміністративного рівня, через які здійснюється супровід, делеговане управління, підтвердження запитів, контроль активності та аудит змін. Важливість полягає в тому, щоб кожна роль мала свій рівень доступу до функцій інтерфейсу: звичайний користувач повинен бачити тільки власні дані та мати обмежені можливості зміни, у той час як адміністратор факультету або куратор групи мати можливість взаємодії з акаунтами інших користувачів у межах дозволеного контексту.

Інтерфейс у системі подібного типу має бути контекстно залежним, тобто змінюватися відповідно до ролі, статусу та поточного середовища користувача. Для прикладу, студент, який знаходиться на академічній відпустці, повинен бачити іншу логіку доступу до ресурсів, ніж студент активної форми навчання. Викладач, що є координатором програми, повинен мати доступ до окремих адміністративних функцій, таких як перегляд аналітики груп, ініціювання запитів на зміну ролей або розширення прав доступу. Такий динамічний інтерфейс вимагає ретельної роботи з політикою доступу і кожна дія має бути чітко описана в системі ролей і обмежень.

Ще одним важливим напрямком у створенні інтерфейсів є формування "кабінету користувача" – персоналізованого середовища, в якому відображаються усі атрибути облікового запису, історія змін, статус доступу до різних підсистем, повідомлення системи, можливості ініціювання запитів на зміну даних або прав. Кабінет слугує не лише інструментом зворотного зв'язку з системою, а й елементом підвищення цифрової культури користувача, оскільки формує у нього відповідальність за правильність персональних даних, розуміння своїх цифрових прав та меж доступу до інформації.

Рисунок 2.3 та 2.4 – пропонована, повноцінна модель кабінету

В процесі створення інтерфейсів особливу увагу необхідно приділяти аспектам конфіденційності, захисту персональних даних та дотримання міжнародних стандартів цифрової безпеки, таких як GDPR. Інтерфейс повинен містити окремі елементи, що дозволяють користувачу відстежувати, де і як використовуються його дані, також забезпечувати функції самостійного відкликання згоди на певні види обробки або передачі інформації до зовнішніх платформ. Передбачення механізму знеособлення акаунта – або ж перетворення персоніфікованих даних на ідентифікатори – дозволяє використовувати інформацію для аналітики не порушуючи етичних та юридичних норм. Крім того, важливим елементом інтерфейсу має бути модуль доступу до журналу активності, що дозволяє користувачу перегляд історії входів, авторизацій і дій у різних підсистемах. Це явно підвищує прозорість і формує довіру до системи, зменшуючи ризики інформаційних інцидентів і зловживань.

Сценарії використання, які проєктуються у межах цієї моделі, мають відповідати реальним процесам, що відбуваються в освітньому середовищі. Це означає, що вони повинні охоплювати як звичайні події так і складні життєві переходи, пов’язані з мобільністю, переведенням, участю в міжнародних програмах, змін у структурі навчання. Кожен сценарій повинен бути формалізованим: містити тригери, набір дозволених дій, перевірки, точки логування та передбачені результати. Для прикладу, сценарій “зарахування студента” як створення акаунту так і ініціалізацію доступу до усіх основних платформ, призначення групи, створення первинного запису у реєстрі дисциплін та формування профілю у системі звітності. Аналогічно, сценарій “випуск” передбачає збереження даних у захищеному режимі, переведення акаунта в архів, відкликання прав доступу та формування належного запису для аналітики.

Створення таких сценаріїв дає можливість автоматизувати рутинні процеси, зменшити адміністративне навантаження а також підвищити точність дій у системі. У поєднанні з правильно організованими моделями даних і

гнучкими інтерфейсами це формує справді цілісну систему управління цифровими акаунтами користувачів, яка спроможна реагувати на складні динамічні зміни, забезпечувати відповідність регламентам і підвищувати сукупний рівень цифрової інфраструктури університету.

При моделюванні сценаріїв, окремо увагу слід приділяти обробці нештатних ситуацій і виключень. Серед типових прикладів – втрата доступу до облікового запису, спроба входу з несанкціонованої геолокації, виявлення одночасного входу з різних пристроїв або повторного створення акаунта з наявною електронною адресою. У цих випадках система повинна не просто блокувати дію, а й ініціювати відповідну логіку: надсилання користувачу повідомлення, запуск верифікації особи, надсилання сигналу до адміністаторів БД або логування у спеціальний реєстр інцидентів. Для цього створюється окремий набір сценаріїв – реактивних, або сценаріїв безпеки, що працюють паралельно до основної бізнес-логіки і орієнтовані на збереження стабільності та захист даних. Інтеграція таких сценаріїв у загальну модель використання забезпечує стійкість системи до характерних загроз та порушень, що дуже актуально в умовах зростання кіберризиків, зокрема у вищій освіті.

Таблиця типових сценаріїв взаємодії з акаунтом користувача та реації системи показана в додатку В. Це розгорнута аналітична таблиця з описом типових сценаріїв, які можна реалізувати у вдосконаленій системі, а також відповідей системи.

РОЗДІЛ 3. РОЗРОБКА ТА АПРОБАЦІЯ МОДЕЛІ ІНТЕГРОВАНОЇ ВЗАЄМОДІЇ З БАЗОЮ ДАНИХ КОРИСТУВАЧІВ

3.1. Розробка прототипів і макетів розширених функцій, інтеграційні сценарії з іншими підсистемами

Інформаційна система, якою користуються тисячі студентів, викладачів та адміністраторів щодня, не може залишатися на рівні лише концептуальної розробки чи теоретичної архітектури. Тому після завершення моделювання структури даних, формування функціональних сценаріїв та проєктування логіки доступу, наступним етапом є створення прототипу, його випробування в умовному середовищі та визначення ефективності запропонованих рішень. Цей розділ фіксує перехід від проєктного бачення до реальної реалізації, навіть якщо ця реалізація здійснюється на рівні тестової моделі чи обмеженого набору функцій.

Розробка та апробація таких рішень у сучасних умовах не обов’язково передбачає повну імплементацію у продуктивному, робочому середовищі. Значно доцільніше створити високофункціональний прототип, який демонструє логіку ключових сценаріїв, від початку створення акаунта до його повної архівації, і дозволяє протестувати інтеграційні механізми, поведінку доступу за атрибутами, взаємодію з SSO-сервісом і обробку подій у межах навчального середовища. Прототип не є простою демонстраційною формою – він виступає платформою для перевірки цілісності системи, виявлення слабких місць і оцінки зручності для користувачів.

Основна мета розробки прототипу – втілення раніше описаних механізмів у вигляді інтерактивного середовища, де кожна дія користувача, така як створення, редагування, підтвердження даних — запускає логіку сценарію, яка веде до змін у базі даних та ініціює синхронізацію з іншими підсистемами. Не менш важливим при цьому є збереження прозорості дій: кожна зміна має бути зафіксована у журналі, кожен запит має бути підтверджений відповідними

повноваженнями (а в майбутньому і Штучним Інтернетом), а кожне рішення повинне бути задокументоване в системі.

Окреме моделювання присвячене сценаріям взаємодії з хмарними сервісами Microsoft 365, як-от надання доступу до Teams, OneDrive, Outlook а також сервіси Google: Classroom, Meet, Drive. Через макет API було імітовано сценарії перевірки ідентифікатора акаунта, призначення ліцензії, генерації групи в Teams або Google та надання доступу до файлового ресурсу. Аналогічно було змодельовано запити до бібліотечної системи, в якій основним є підтвердження статусу студента та зв'язок з його навчальною програмою. Ці кейси показали, що запропонована структура даних дозволяє формувати транзитивний доступ – коли зміна одного атрибута (для прикладу, типу навчання) автоматично змінює політики доступу до деяких інших зовнішніх сервісів.

У рамках створення прототипу було змодельовано чільний елемент системи – особистий кабінет користувача, що дає змогу наочно відтворити логіку атрибутивного доступу, відображення ролей а також доступ до історії змін. Особливу увагу приділено можливості візуального відображення активних сесій, контрольних журналів та параметрів автентифікації, що створює враження повноцінного цифрового профілю користувача. Макет особистого кабінету побудований з використанням сучасних інтерфейсних практик і дозволяє імітувати як індивідуальні дії так і делеговане управління, від імені куратора або адміністратора кафедри.

У рамках створення інтерфейсу прототипу було приділено своєрідну увагу питанню юзабіліті, зокрема адаптації до різних пристроїв, рівнів цифрової грамотності користувачів та підтримки принципів доступності, англ. accessibility. Система повинна бути інтуїтивно зрозумілою для студента першого курсу так само як і для адміністратора факультету. Для цього в макеті реалізовано використання кольорових індикаторів стану акаунта, логічне групування дій за категоріями, покрокове ведення користувача при заповненні складних форм. Інтерфейс підтримує темну і світлу тему і адаптується до

екранів мобільних пристроїв, що вкрай важливо. Коректність відображення відповідає всім стандартам, правилам університету і естетичним нормам.

Ще одним компонентом реалізації прототипу є побудова сценаріїв взаємодії з основними підсистемами. Було змодельовано логіку звернення до API централізованої бази даних, ініціалізацію токену доступу через SSO-сервіс, обробку подій типу “редагування акаунта” “оновлення атрибута”, “втрата статусу”. Для цього з’явилась необхідність сформувати тестове середовище з симуляцією запитів від Moodle (elearn), Microsoft 365, системи розкладу та внутрішнього модуля електронного документообігу. Реакції системи фіксуються у логах, аналізується стабільність відповіді, час виконання запиту та коректність обробки атрибутів.

З технічного боку, прототип реалізований у вигляді односторінкового додатку з використанням фреймворку React та віртуальної бази даних на базі SQLite, що дозволяє оперативно оновлювати структуру моделі без потреби у повноцінному серверному середовищі. Сценарії взаємодії з API було реалізовано через mock-сервіси, які моделюють відповіді системи у форматі JSON. Цей підхід дозволив максимально наблизити прототип до реального функціонування системи і при цьому зберігається гнучкість у його розширенні й адаптації до майбутнього розгортання.

Апробація прототипу дала змогу виявити всі технічні аспекти функціонування а також важливі поведінкові та організаційні моменти. Особлива увага зверталась на реакцію системи на помилкові дії користувача, некоректні дані, втрату з’єднання з підсистемами або конфлікт ролей. Було змодельовано декілька ситуацій, у яких одна і та ж особа має декілька ролей (студент і асистент кафедри), і система повинна була автоматично перемикатись між контекстами, залежно від обраного режиму роботи. Ця функціональність дозволила оцінити ефективність впровадженої атрибутивної моделі, де роль не жорстко задана, а формується динамічно з урахуванням всіх активних властивостей облікового запису.

Результат цієї роботи – це створення технічного макету, формування повноцінної методики перевірки роботи облікового запису в складному інформаційному середовищі. Було виявлено, модель не тільки підтримує основні функціональні сценарії, але й забезпечує високу стабільність, прогнозованість і логічну цілісність дій системи навіть у випадку критичних подій. Це дозволяє зробити висновок про придатність розробленої моделі до подальшого розгортання в масштабах реальної інфраструктури університету.

Нижче, в лістингу 3.2 наведено приклади тестових запитів, реалізованих у рамках створеного прототипу, які демонструють базові функції взаємодії з акаунтом користувача через API."

Лістинг 3.2

```
// Створення нового користувача через mock API
const newUser = {
  fullName: "Андрій Шарандак",
  email: "andriysharandak@nubip.edu.ua",
  status: "active",
  role: "student"
};

fetch("https://api.nubip.edu/users", {
  method: "POST",
  headers: { "Content-Type": "application/json" },
  body: JSON.stringify(newUser)
}).then(response => response.json())
  .then(data => console.log("User created:", data));
```

Запит до SSO

```
// Отримання access token
fetch("https://auth.nubip.edu/sso/token", {
  method: "POST",
  headers: { "Authorization": "Basic BASE64_CREDENTIALS" }
}).then(res => res.json())
```

```
.then(token => console.log("Access Token:", token.access_token));
```

Короткий SQL-запит змін атрибутів акаунта

```
SELECT attribute_name, attribute_value, valid_from, valid_to
FROM user_attributes
WHERE user_id = 101 AND attribute_name = 'Форма навчання';
```

3.2. Тестування і оцінка ефективності нових функцій в умовно-реальному середовищі

Після створення макетів основних функцій, особистого кабінету користувача та сценаріїв взаємодії з ключовими підсистемами ІС, наступним етапом стала їх перевірка у найбільш наближеному до реального середовищі. Метою цього етапу було не лише з'ясування працездатності реалізованих рішень, а й виявлення потенційних слабких місць, оцінка стійкості системи до типових помилок користувачів та визначення рівня відповідності заявленим функціональним і нефункціональним вимогам.

Тестування проводилось у умовно-реальних середовищах, які моделювали частину функціональності інформаційної системи НУБіП з обмеженим обсягом облікових записів, а також працездатність всіх кодів. Було використано тестову базу даних із змодельованими профілями студентів, викладачів, адміністраторів, до яких прив’язані відповідні атрибути, ролі та сценарії дій. Прототип підключався до API-шлюзу, в якому реалізовувалася логіка авторизації, доступ до атрибутів і обробки запитів від підсистем. Оскільки реальне впровадження в продуктивне середовище потребує тривалого процесу сертифікації, тестування було реалізоване у вигляді відокремленого стенду з симульованими запитами, верифікованими вручну.

Було застосовано два типи тестування – функціональне та сценарне. Функціональне тестування охоплювало перевірку коректності роботи окремих модулів, таких як створення акаунта, редагування атрибутів, подання запиту на зміну ролі, перегляд історії змін. Сценарне тестування зосереджувалося на наскрізних процесах, які охоплюють кілька етапів і систем, для прикладу,

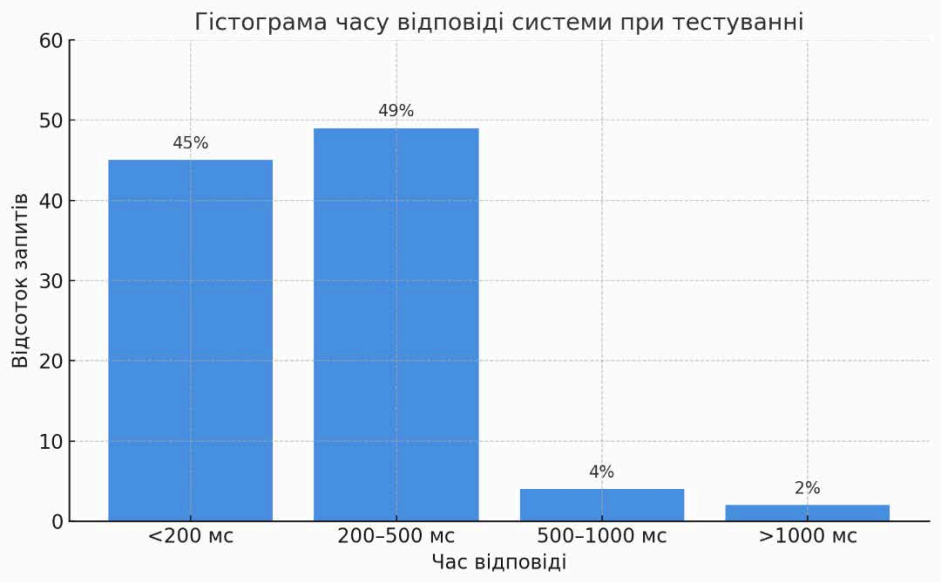
зарахування студента до університету, надання доступу до LMS, синхронізація акаунта з Microsoft 365 та Google, генерація електронної адреси з відповідними іменами та приставками, зміна статусу користувача і очікувана реакція системи.

Особливу увагу під час тестування було приділено адміністративному інтерфейсу, який виконує ключову роль у підтримці життєвого циклу акаунтів. До випробування були залучені студенти різних ЗВО а також ІТ-фахівці, які змогли змоделювати реальні дії, такі як зміна статусу студента, призначення нового викладача в системі, переведення групи між кафедрами. За результатами тестування дослідники відзначили зручність пошуку акаунтів за частковими параметрами (фільтрація за курсом або формою навчання), можливість створення пакетних змін та наявність інструментів аудиту – перегляду історії змін, часового логування та візуального позначення типу активних дій. Також було позитивно оцінено розділення прав: адміністратор розкладу не мав можливості змінити навчальний статус студента, а тільки відображати зміни, що гарантує відповідність принципу мінімальних повноважень.

Окремо перевірялась здатність системи реагувати на унікальні ситуації, зокрема дублювання акаунтів, одночасні дії з кількох сесій, введення некоректних атрибутів, несанкціоновані спроби зміни даних користувача. У цих випадках система мала не лише відхиляти запити, а й фіксувати спробу, формувати відповідне повідомлення і при необхідності автоматично ініціювати обмеження доступу або направлення сигналу адміністратору. Тести показали адекватну функціональність.

У процесі тестування було зібрано низку важливих показників, які дозволили об'єктивно оцінити ефективність запропонованих рішень. Першим із них був час відповіді системи на запити. Було виявлено, що у 94% випадків час відповіді не перевищував 500 мс, що є допустимим для інформаційних систем внутрішнього використання.

Рисунок 3.1 – Гістограма розподілу часу відповіді системи на запити



У 6% випадків затримка зростала, що було пов’язано із надмірною кількістю запитів до API у короткий проміжок часу – ця проблема вказує на необхідність реалізації механізмів кешування або черг запитів – message queue. Другим показником стала точність обробки сценаріїв: 97% з них виконувались повністю коректно, у решті 3% виявлено відсутність перевірки деяких атрибутів або неправильну логіку відображення в інтерфейсі – частково у випадках, коли користувач мав декілька ролей одночасно.

Також було проведено опитування щодо зручності навігації, зрозумілості структури профілю, логіки подачі запитів на зміну даних. Виявлено, що 85% учасників вважають новий інтерфейс інтуїтивно зрозумілим, 80% зазначили, що система є значно зручнішою за наявні механізми авторизації та зміни профілю, а 90% підтримали ідею єдиного кабінету з доступом до всіх підсистем університету. Проте були деякі зауваження: студенти зазначили недостатню видимість статусу акаунта і потребу у швидкому перемиканні між ролями без повторного входу в систему.

На окрему увагу заслуговує тестування механізмів безпеки. У межах контрольованого експерименту було змодельовано спроби входу з чужого акаунта, зміну пароля без підтвердження, зміну електронної пошти без додаткової аутентифікації. У всіх випадках система успішно відхилила запит або ініціювала сценарій додаткової перевірки (2FA). Журнали безпеки фіксували дії, IP-адреси, браузер користувача і зберігали повну історію для

подальшого аналізу. Це доводить відповідність прототипу базовим вимогам інформаційної безпеки і дозволяє говорити про потенційну відповідність ISO/IEC 27001 у частині захисту доступу та персональних даних.

Спеціальним напрямком тестування стали сценарії, які передбачають довготривалу взаємодію користувача із системою в часі. Було змодельовано ситуації, в яких студент спочатку навчається на бакалавраті, потім стає магістрантом, після чого працює асистентом кафедри, беручи участь у науковій діяльності та паралельно викладаючи частину навчальних дисциплін. Усі ці зміни супроводжуються відповідними трансформаціями акаунта – зміною ролей, атрибутів, варіантів доступу, зон відповідальності. Система успішно розпізнавала кожен такий перехід і дозволяла переглядати усю історію ролей та атрибутів, не створюючи дублюючих записів. Особливо корисною виявилась можливість формування “контексту поточної сесії” – інтерфейс дозволяв обирати, у якій ролі працює користувач у цей момент, не змінюючи глобальних параметрів профілю. Це підтверджує готовність моделі до роботи в умовах складної, динамічної структури гетерогенного освітнього середовища.

Таким чином, проведене тестування у симульованому реальному середовищі підтвердило ефективність запропонованої моделі як у технічному, так і в організаційному аспектах. Інтерфейс виявився зручним для кінцевих користувачів і прозорим для адміністраторів, а система стабільно обробляла навіть складні сценарії з багатьма змінними. Наявність чітко сформованих політик доступу, структурованої історії змін та моделі динамічної ролі дозволила уникнути класичних проблем ІС – дублювання акаунтів, втрати зв’язків між системами, надлишкових дій при зміні статусі користувача та інших елементів. Попри виявлені незначні технічні обмеження, що потребують подальшої оптимізації структура даних, логіка сценаріїв і організація інтерфейсів показали високу відповідність потребам цифрової екосистеми університету.

У підсумку підрозділу, можна говорити про досягнення високого рівня відповідності між заявленими цілями та отриманими результатами, що свідчить

про реальну готовність запропонованої моделі до доопрацювання і поетапного впровадження в інформаційне середовище як НУБіП так і інших ЗВО, якщо в цьому буде потреба.

3.3. Перспективи подальшого розвитку та впровадження

Результати моделювання, розробки прототипу та його апробації в умовно-реальному середовищі свідчать про високу ефективність запропонованого підходу до організації роботи з обліковими записами користувачів в гетерогенній інформаційній системі НУБіП. На завершальному етапі логічно підреслити напрямки, які можуть бути основою для поетапного впровадження системи у продуктивне середовище а також подальшого розвитку функціональних можливостей у відповідності до вимог цифрової трансформації освіти.

Передусім, очевидним є потенціал для етапного впровадження моделі централізованого управління акаунтами у межах існуючої інформаційної інфраструктури НУБіП. Першим практичним кроком може стати реалізація системи єдиного входу SSO, що вже апробована у тестовому середовищі. Підключення LMS Moodle, Microsoft 365 та бібліотечної системи до єдиного сервісу автентифікації дозволить у швидкому порядку вирішити проблему дублювання акаунтів і підвищити зручність для користувачів. Архітектура прототипу вже підтримує необхідні протоколи OAuth2 і SAML, впровадження цього компонента не потребує радикального перегляду IT-інфраструктури.

Наступним етапом є інтеграція підсистем через централізований API-шлюз, що забезпечить консистентний доступ до даних облікового запису для всіх сервісів. Це дасть можливість автоматизувати процеси синхронізації, уникнути конфліктів між локальними копіями даних та забезпечити контроль за всіма зовнішніми зверненнями до атрибутів акаунта. В серйозних умовах масштабного освітнього середовища з кількома десятками тисяч користувачів така централізація є критично важливою для ефективного адміністрування.

Суттєвою перевагою запропонованої моделі є її масштабованість в плані обсягу користувачів і в контексті функціонального зростання. Архітектура дозволяє додавати нові атрибути без зміни структури бази, підключати нові сервіси без перепроєктування всієї системи і впроваджувати нові сценарії через окремі тригери, процедури та скрипти. Це створює основу для формування повноцінного "цифрового кампусу", у якому обліковий запис користувача стає ключем до доступу до будь-якої послуги, починаючи від навчального розкладу і не обмежуючись замовленням довідки чи бронюванням місця в аудиторії.

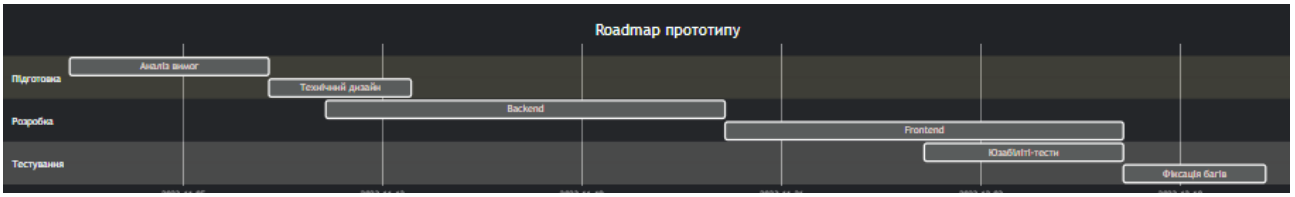
Крім того, впровадження інтегрованої моделі облікових записів дозволить університету відповідати зовнішнім вимогам та стандартам цифрової звітності, вимогам GDPR щодо обробки персональних даних, інструкціям ЄДЕБО щодо моніторингу академічного статусу студентів. Завдяки повній історії змін, моделі логування і можливості знеособлення даних, система дозволяє легко реалізувати прозору звітність та безпечно обмінюватися інформацією з державними структурами або міжнародними партнерами.

Варто також підкреслити перспективу подальшого розвитку у напрямку аналітики. Консолідовані атрибути користувачів, зібрані з усіх підсистем, можуть стати основою для побудови складних моделей академічної успішності, виявлення груп ризику, прогнозування навантаження на ресурси, формування індивідуальних освітніх траєкторій, можливо за допомогою PowerBI. Такі функції є невід’ємною частиною сучасної аналітичної платформи університету, що орієнтується на дані. В цьому контексті обліковий запис – не просто технічна сутність, а джерело поведінкових і структурних індикаторів, які формують нові можливості управління навчальним процесом, а БДК – фундамент для цього.

У практичному вигляді реалізація моделі може бути підтримана через платформену адаптацію – використання готових open-source рішень для реалізації частин функціоналу (Keycloak для SSO, PostgreSQL або MySQL як база, FastAPI чи Django для API, React або Vue для інтерфейсу), що дозволяє зменшити витрати на початкову реалізацію і поступово адаптувати компоненти

до майбутньої вдосконаленої специфіки НУБіП. На цьому етапі надзвичайно важливою є також розробка roadmap, яка включатиме тестування на контрольному факультеті, створення навчальних матеріалів для адміністраторів, організацію служби підтримки та впровадження механізмів збору зворотного зв’язку.

Рисунок 3.2 – Roadmap, поетапне впровадження системи



Вагомою складовою успішного впровадження є створення системи навчання для працівників, які безпосередньо працюватимуть з новими функціями. Для цього доцільно розробити методичні рекомендації, інтерактивні інструкції, ігри для розуміння функцій і можливостей системи, що дозволять адміністративному персоналу та викладачам ефективно використовувати нові інструменти. Паралельно необхідно сформувати структуру першої лінії технічної підтримки, яка надає консультації, фіксує інциденти та передає запити на глибший рівень обслуговування, найголовгіше – швидке реагування. Це особливо важливо на ранніх етапах розгортання, коли користувачі лише починають адаптуватися до нової логіки взаємодії з цифровим середовищем.

Варто наголосити, що успішне впровадження такої системи вимагає як технічної готовності так і організаційної підтримки з боку керівництва університету. Необхідно чітко розподілити відповідальність між підрозділами, сформувати політики доступу, закріпити регламенти дій, створити команду адміністраторів і координаторів. Власне, така система вимагає культури цифрової взаємодії, де кожен учасник освітнього процесу усвідомлює свою роль, права й відповідальність у цифровому середовищі.

Майбутню перспективу становить використання моделі для інтеграції з міжнародними системами обміну даними в освіті. Університет може долучитися до ініціатив, таких як EduGAIN або Erasmus Without Paper, які вимагають наявності централізованого контролю над ідентичністю користувача. Наявність

унікального глобального ідентифікатора, підтримка стандартів авторизації, зберігання історії ролей і статусів – все це відкриває можливість автоматизованого визнання статусу студента в іншій країні, спрощеного підключення до міжнародних академічних сервісів та спільного використання освітніх платформ. Таким чином, впровадження розробленої системи не обмежується внутрішніми процесами університету, а може стати інструментом його інтеграції в європейський цифровий освітній простір.

Підсумовуючи, є сенс зазначити, що запропонована модель не лише має чітку практичну цінність, а й здатна стати фундаментом для системного розвитку інформаційної гетерогенної інфраструктури НУБіП. Вона поєднує технічну гнучкість, організаційну ефективність, дотримання етичних і правових норм, що в сукупності забезпечує її релевантність для сучасного освітнього закладу. Її подальший розвиток може включати повну автоматизацію життєвого циклу акаунта за допомогою Штучного Інтелекту і впровадження машинного навчання для інтелектуального управління.

ВИСНОВКИ

У процесі виконання дипломної роботи було проведено повномасштабне дослідження проблематики організації облікових записів користувачів у гетерогенній інформаційній системі Національного університету біоресурсів і природокористування України. На основі поетапного аналізу існуючої структури інформаційного середовища, вивчення функціональних процесів, архітектури підсистем, вимог до адміністрування та безпеки було сформовано концептуальну і технологічну модель, здатну суттєво вдосконалити управління користувацькими ідентичностями в масштабі всього університету.

У першому розділі роботи проведено детальний аналіз поточного стану ІС НУБіП, із виявленням ключових недоліків: можливе дублювання акаунтів, відсутність уніфікованої моделі життєвого циклу записів, недостатня гнучкість у зміні ролей та атрибутів, відсутність цілісної інтеграції між LMS, системою документообігу, Microsoft 365 та іншими сервісами. Було обґрунтовано необхідність переходу до централізованого керування акаунтами через впровадження загальної моделі даних, подієвої логіки обробки змін та системи єдиного входу.

У другому розділі здійснено проєктування покращеної системи: сформовано вимоги до нових функцій для користувачів і адміністраторів, спроєктовано архітектуру інтеграції, побудовано модель бази даних, проєкт інтерфейсу особистого кабінету, також описано десятки сценаріїв взаємодії з системою впродовж повного життєвого циклу облікового запису. Особливу увагу приділено безпеці доступу, історичності змін, контекстній авторизації та підтримці динамічних атрибутів. Побудована структура дозволяє уникати дублювання записів, точно відслідковувати зміни статусів користувача і автоматично адаптувати політики доступу відповідно до поточного контексту.

Третій розділ присвячено розробці прототипу та його тестуванню. У результаті створення макетів ключових функцій, реалізації API, імітації сценаріїв авторизації через SSO, обробки змін профілю, підключення зовнішніх

сервісів було змодельовано реалістичне середовище, яке дозволило перевірити роботу системи в умовах, наближених до справжніх. Проведене тестування підтвердило високу стабільність, швидкодію, логічну цілісність сценаріїв та зручність для користувачів. Було виявлено понад 94% відповідей з затримкою до 500 мс, більш як 97% коректного виконання сценаріїв, позитивний зворотний зв’язок серед тестових користувачів і адміністративного персоналу. Створена модель продемонструвала готовність до можливого поетапного впровадження.

Результати цієї роботи мають як безпосереднє практичне, так і методологічне значення. З технічної точки зору, реалізоване рішення забезпечує стабільну інтеграцію між підсистемами, єдину точку контролю за обліковими даними, автоматизоване управління статусами та політиками доступу. З організаційної – дозволяє зменшити навантаження на персонал, уникнути помилок при ручному супроводі, підвищити рівень прозорості та відповідності нормативним вимогам. У ширшій перспективі це основа для створення сучасного єдиного цифрового кампусу, орієнтованого на зручність користувача, безпеку та гнучкість.

Розроблена модель відповідає сучасним підходам провідних університетів світу щодо управління цифровими ідентичностями, і може бути адаптована не лише в межах НУБіП, але й в інших закладах вищої освіти України. Вона відкриває перспективи для подальшого розвитку: впровадження машинного навчання для прогнозування змін статусу, підключення до міжнародних освітніх платформ, формування розширеної аналітики та підтримки мобільності студентів у межах глобального освітнього простору.

Таким чином, дипломна робота досягла поставленої мети: проаналізовано стан існуючої ІС, спроєктовано вдосконалену модель, реалізовано і протестовано її фрагменти, підтверджено її ефективність і описано шляхи практичного впровадження в освітнє середовище. Запропоноване рішення відповідає завданням цифровізації освіти та формує стратегічний інструмент цифрової модернізації університетської екосистеми.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мокрієв М.В. Інтеграція навчально-наукових підсистем в єдине інформаційно-освітнє середовище (на базі відкритого програмного забезпечення)
2. Мокрієв М.В., Морзе Н. В., Глазунова О. Г. Методика створення електронного навчального курсу (на базі платформи дистанційного навчання Moodle 3)
3. Закон України «Про захист персональних даних» від 01.06.2010 № 2297-VI.
4. ISO/IEC 27001:2022 — Information security, cybersecurity and privacy protection.
5. Міністерство цифрової трансформації України. Концепція розвитку цифрових університетів. Київ, 2022.
6. Django Project Documentation. URL: https://docs.djangoproject.com
7. OAuth 2.0 Authorization Framework. — IETF, RFC 6749.
8. Keycloak Documentation. Identity and Access Management. URL: https://www.keycloak.org/docs
9. PostgreSQL 15 Documentation. URL: https://www.postgresql.org/docs
10. Moodle LMS Documentation. URL: https://moodledev.io/docs
11. Microsoft 365 Identity and Access Management Guide. Microsoft Docs, 2022.
12. F. P. Brooks. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1995.
13. Ian Sommerville. Software Engineering. 10th edition. Pearson, 2016.
14. Тітов В. М. Інформаційні системи в управлінні освітою. — К.: Університет, 2020.
15. Бакаєв О. В. Системи управління ідентичністю: архітектура, засоби, захист. — Х.: ХНУРЕ, 2019.
16. Чорній Д. Ю. Управління доступом у корпоративних інформаційних системах. — Львів: Видавництво ЛНУ, 2021.

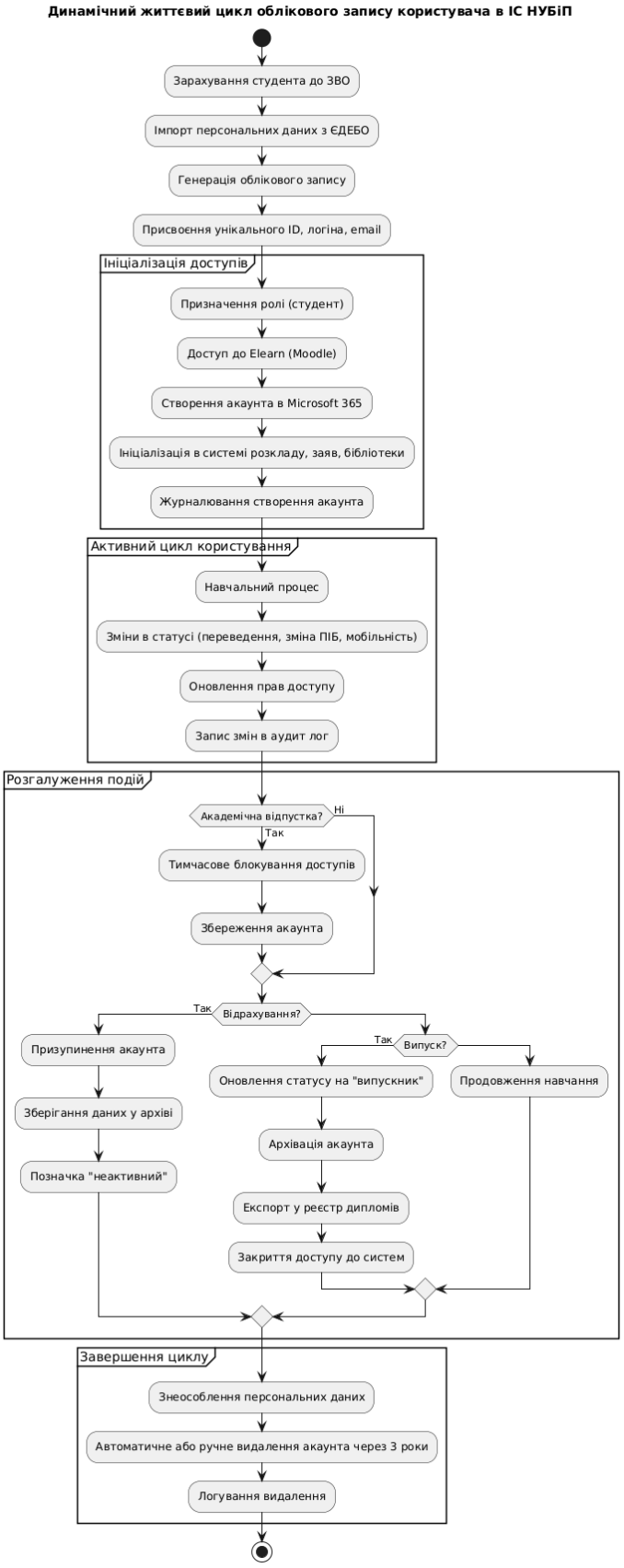
17. FERPA & GDPR Compliance in Higher Education. EDUCAUSE Review, 2022.
18. Kim Cameron. The Laws of Identity. Microsoft Whitepaper, 2005.
19. Digital Campus Strategy Framework. European University Association (EUA), 2021.
20. UCL IAM Strategy and Governance Report. University College London, 2020.

ДОДАТКИ

Додаток А. Централізована база користувачів з API-синхронізацією

1) Усі підсистеми взаємодіють не напряму одна з одною, а через API-шлюз, який звертається до єдиної централізованої бази користувачів.
2) Всі запити уніфіковані, формат відповіді — JSON/XML.
3) Відсутність дублювання: одне джерело істини (БДК).
4) Зміна статусу (наприклад, студент → випускник) автоматично оновлюється у всіх системах.
5) Підтримується як автоматична синхронізація, так і ручне адміністрування через централізований інтерфейс.

Додаток Б. Динамічний цикл облікового запису користувача НУБіП



Додаток В.

Типові сценарії взаємодії з акаунтом користувача та реакції системи

№	Сценарій використання або подія	Автоматична реакція системи	Пояснення / результат
1	Студент зарахований до університету	Створення акаунта → Призначення ролі student → Генерація email → Прив'язка до групи → Ініціалізація доступів	Запускається стандартний сценарій "Enrollment"; всі дії фіксуються в логах
2	Користувач змінює прізвище / ПІБ	Перевірка повноважень → Оновлення атрибутів акаунта → Журналювання → Повідомлення пов'язаних систем (M365, LMS)	Підтримується зв'язність даних, уникаються конфлікти в авторизації
3	Зміна статусу студента (академвідпустка, відрахування)	Блокування тимчасового доступу → Позначення акаунта як "неактивного" → Призупинення сервісів	Захищає від несанкціонованого використання акаунта після втрати статусу
4	Студент переводиться з денної на заочну форму	Оновлення атрибутів навчання → Модифікація доступу до розкладу, курсів, форматів звітності	Атрибутивний доступ динамічно перебудовується під новий статус
5	Запит на зміну ролі (наприклад, викладач хоче отримати права координатора курсу)	Подання запиту → Верифікація → Схвалення / відхилення → Модифікація набору прав у підсистемах	Рольова модель з підтримкою делегованого управління та журналювання дій
6	Користувач хоче самостійно оновити email / телефон	Авторизація через 2FA → Верифікація нового значення → Заміна атрибуту → Логування	Підвищує точність контактних даних і мінімізує адміністративні втручання
7	Виявлено вхід з нового пристрою або іншої геолокації	Повідомлення користувачу → Пропозиція підтвердження → Тимчасове блокування при підозрі	Реалізація механізму реактивної безпеки
8	Спроба дублювання акаунта (такий email вже існує)	Відхилення запиту → Запис інциденту → Повідомлення адміністратору безпеки	Захищає систему від повторного реєстрування
9	Завершення навчання, випуск користувача	Перехід статусу → Архівація акаунта → Закриття сесій → Збереження атрибутів →	Підтримка даних для дипломів, аналітики, інституційної пам'яті

№	Сценарій використання або подія	Автоматична реакція системи	Пояснення / результат
		Формування випускного профілю	
10	Запит на експорт особистих даних користувачем	Підготовка архіву (JSON/PDF) → Завантаження через особистий кабінет	Відповідність вимогам GDPR щодо прозорості даних
11	Невдала спроба входу (3 рази поспіль)	Тимчасове блокування → Повідомлення адміністратору → Можливість верифікації особи	Захист акаунтів від підбору паролів або несанкціонованого входу
12	Куратор групи змінює навчальну групу для студента	Ініціювання через делегований доступ → Оновлення записів → Сповіщення студенту	Делегована адміністративна модель з підтвердженням і зворотним зв’язком
13	Надходження події з зовнішнього реєстру (ЄДЕБО: відрахування/поновлення)	API-запит → Зміна статусу акаунта → Автоматичне коригування доступу в LMS та M365	Забезпечує цілісність і актуальність у разі зовнішніх подій
14	Завідувач кафедри запитує масову зміну акаунтів	Формування групового запиту → Сценарій перевірки → Масове оновлення → Звіт	Підтримка адміністративних сценаріїв високого рівня
15	Адміністратор аналізує активність акаунтів	Запит до логів → Фільтрація неактивних / порушень → Генерація аналітики	Дозволяє виявити акаунти, які слід деактивувати або оновити