МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження моделей та методів контролю доступу до інформаційної системи

(тема)

Виконав:
Студент 2 курсу, групи ІПЗм-19-1
Михайлов А.О.
(прізвище, ініціали)

Спеціальність 121-Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми Освітньо-наукова програма
(освітньо-професійна або освітньо-наукова)

Керівник к.т.н, доц. Каук В.І
(посада, прізвище)

Допускається до захисту

Зав. кафедри _____ З.В. Дудар
(підпис) (прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук
(повна назва)

Кафедра Програмної інженерії
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)

Тип програми Освітньо-наукова програма
(освітньо-професійна або освітньо-наукова)

Освітня програма Інженерія програмного забезпечення
(повна назва)

ЗАТВЕРДЖУЮ:
Зав.кафедри _____
(підпис)
« 26 » березня 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Студента Михайлова Антона Олександровича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження моделей та методів контролю доступу до інформаційної системи

затверджена наказом університету від 26.03.2021 № 385Ст

2. Термін подання роботи до екзаменаційної комісії 18 05 2021р.

3. Вихідні дані до роботи *алгоритм вибору моделі доступу до інформаційної системи відповідно до вхідних параметрів, пояснювальна записка. Прототип браузерного додатку для визначення найбільш відповідної моделі*

4. Перелік питань, що потрібно опрацювати в роботі *мета роботи, аналіз проблемної галузі та постановка задачі, опис існуючих програмних аналогів, дослідження існуючих моделей доступу до інформаційної системи, опис*

розробленої програмної реалізації, аналіз можливих застосувань

5. Перелік графічного матеріалу із зазначенням креслеників, схем, слайдів, ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)

титульний лист, мета роботи, аналіз предметної галузі, актуальність питання, постановка задачі, досліджені моделі доступу, модель проведення дослідів, підходи до реалізації, діаграма варіантів використання, опитування експерта, інформація про системи власника, результат з рекомендаціями, апробація Висновки

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування Розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	Дата
Спецрозділ	к.т.н, доц. Каук В.І		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	01.02.2021	виконано
2	Огляд існуючих систем-аналогів	22.02.2021	виконано
3	Огляд існуючих методів контролю доступу	01.03.2021	виконано
4	Розробка бази даних та архітектури	09.03.2021	виконано
5	Програмна реалізація	27.03.2021	виконано
6	Підготовка пояснювальної записки	20.04.2021	виконано
7	Спецчастина	23.04.2021	виконано
8	Підготовка презентації та доповіді	01.05.2021	виконано
9	Попередній захист	05.05.2021	виконано
10	Нормоконтроль, рецензування	14.05.2021	виконано
11	Занесення диплома в електронний архів	15.05.2021	виконано
12	Допуск до захисту у зав. Кафедри	19.05.21	виконано

Дата видачі завдання 25 січня 2021р.

Студент _____
(підпис)

Керівник роботи _____ к.т.н, доц. Каук В.І
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота магістра містить: 83 с., 19 рис., 6 табл., 37 джер.

БЕЗПЕКА ІНФОРМАЦІЙНИХ СИСТЕМ, МОДЕЛІ ТА МЕТОДИ КОНТРОЛЮ ДОСТУПУ, КОРИСТУВАЧІ, АВТОРИЗАЦІЯ, РОЛІ, АТРИБУТИ, ACL, RBAC, ABAC, PHP, MYSQL.

Об’єктом дослідження є безпека інформаційних систем.

Метою даної роботи є аналіз основних моделей контролю доступу до інформаційних систем, їх переваги та недоліки для самостійного та комплексного використання.

Методи розробки базуються на таких технологіях, як PHP, Laravel Framework, MySQL.

В результаті роботи було досліджено методи доступу до інформаційних систем, проведено їх аналіз та розроблено алгоритм вибору найбільш оптимальної моделі в залежності від вхідних параметрів для певної системи. Розроблено схему бази даних та через роботу браузерного додатку реалізовано алгоритм вибору моделі доступу.

INFORMATION SYSTEMS SECURITY, MODELS AND METHODS OF ACCESS CONTROL, USERS, AUTHORIZATION, ROLES, ATTRIBUTES, ACL, RBAC, ABAC, PHP, MYSQL.

The object of research is the security of information systems.

The purpose of this work is to analyze the main models of access control to information systems, their advantages and disadvantages for independent and integrated use.

Development methods are based on technologies such as PHP, Laravel Framework, MySQL.

As a result, the methods of access to information systems were studied, their analysis was performed and an algorithm for selecting the most optimal model depending

on the input parameters for a particular system was developed. The scheme of the database is developed and through the work of the browser application the algorithm of the access model selection is implemented.

Я, Михайлов Антон Олександрович, студент гр. ІПЗм-19-1, здобувач вищої освіти на другому (магістерському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Дослідження моделей та методів контролю доступу до інформаційної системи», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElAr KhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ABAC	Attribute-Based Access Control
ACL	Access Control List
AIC	Confidentiality, Integrity and Availability (CIA triad)
CBAC	Context-Based Access Control
DAC	Discretional Access Control
HRBAC	Hierarchical Role-Based Access Control
IBAC	Identity-Based Access Control
MAC	Mandatory Access Control
RBAC	Role-Based Access Control
SRBAC	Spatial Role-Based Access Control
TRBAC	Temporal Role-Based Access Control
ЗЗПР	Загальна Задача Прийняття рішення
ТПР	Теорії Прийняття Рішень

ВСТУП

З початку 1970-х років було розроблено багато моделей, основними з яких є моделі дискреційного розмежування доступу DAC та мандатного розмежування доступу MAC. [1-2] З ростом мереж з'явилася необхідність в обмеженні доступу до особливо захищених об'єктів. Так з'явився механізм контролю доступу на основі ідентичності IBAC, який використовує списки контролю доступу ACL. Кожен суб'єкт має свій список. Тільки надавши доказ свого повноваження, особа отримує доступ до об'єкту. Власник об'єкта визначає доступні операції для кожного суб'єкта.[3]

Пізніше, у 1992 році Д. Феррайло і Р. Кун описали концепцію рольового управління доступом RBAC. Національним інститутом стандартів і технологій США модель RBAC була стандартизована і для обліку різних особливостей інформаційних систем була запропоновано безліч модифікацій RBAC [4-6], які націлені на роботу з невеликими або не динамічними інформаційними системами.

Але бізнес-правила неминуче ускладнювалися і ставали багатовимірними. З'явилася потреба в інших підходах, які могли надати змогу системі працювати з великою кількістю користувачів, додавати інші атрибути (місто, країну, день тижня, ліміт тощо) до процесу розмежування доступу та підтримувати динамічність у змінах цих атрибутів. Так з'явилася концепція атрибутивного управління доступом ABAC.[7]

На сьогодні актуальність проблеми захисту зростає в міру збільшення обсягів збережених даних і зростання складності програмного забезпечення для їх обробки. Переважна більшість додатків забезпечуються засобами контролю доступу в тій чи іншій формі. Системи розмежування доступу, будучи найважливішими компонентами систем захисту, найбільш схильні до ризиків через можливість помилок в конфігурації політик розмежування доступу.

Тому гостро постає питання саме вибору найбільш відповідної моделі контролю доступу в інформаційній системі для подальшого впровадження. Ці

системи відрізняються кількістю ролей, які задіяні у бізнес процесі, різним рівнем складності самих бізнес-правил, наявністю чи відсутністю динамічних атрибутів, а також необхідністю забезпечувати контроль доступу до дій, даних тощо.

Обрана модель управління доступом відіграє головну роль в системі безпеки інформаційної системи. Завдяки їй визначаються потоки інформації, які дозволені в системі, а також правила надання доступу до інформації в цілому. Модель не накладає обмежень на реалізацію тих чи інших механізмів захисту.

Дана робота певною мірою продовжує дослідження кафедри стосовно безпеки інформаційних систем.[8-13] Метою даної роботи є аналіз основних моделей контролю доступу в інформаційних системах, зокрема їх переваги та недоліки як самостійного так і комплексного використання і вибір найбільш відповідної в залежності від вхідних критеріїв, які характеризують цю інформаційну систему.

Об’єктом дослідження є безпека інформаційних систем. Предмет дослідження – моделі та методи контролю доступу та їх відповідне використання в інформаційних системах.

Програмна реалізація, яка була розроблена дозволяє спростити вибір моделі доступу до інформаційної системи на етапі проектування, або полегшити вибір нової моделі для вже існуючої системи при перепроектуванні шляхом аналізу вимог до безпеки цієї системи. Вона може використовуватися спеціалістами з технічною базою знань, але здебільшого орієнтована сама на користувачів, які не будуть мати такої бази.

За результатами проведеної роботи була написана та опублікована стаття у збірнику Monografia pokonferencyjna», випуск «Science, research, development №29», Варшава, 30.03.2021 – 31.03.2021 на тему «Дослідження моделей та методів контролю доступу до інформаційної системи», текст якої наведено у додатку Д.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз проблемної області

Сьогодні інформаційні системи здебільшого проектуються для використання більше ніж одним користувачем. Звичайно, якщо існує всього один користувач для певної системи, то йому будуть доступні всі дії та вся інформація в системі і робота в такій системі не буде мати якихось складних правил чи залежностей. Але як тільки кількість користувачів збільшується, то зазвичай разом з цим з'являються певні правила та обмеження, наприклад:

– кожен користувач має виконувати тільки свої бізнес-завдання і не мати доступу до чужих;

– кожен користувач повинен бачити тільки пов'язану зі своїми бізнес-завданнями інформацію;

– у кожного завдання повинен бути відповідальний за її виконання користувач.

Якщо не вирішити ці проблеми, фірма може зазнати фінансових втрат через:

– неефективного виконання чужих завдань в силу некомпетентності;

– навмисних або ненавмисних помилок в чужих завданнях;

– розкриття інформації стороннім особам.

За підсумками глобального дослідження, зробленого у першому півріччі 2020 року, яке проводилося аналітичним сервісом InfoWatch (https://infowatch.com/), було виявлено 695 витоків інформації (55,6% від загального числа), де основною причиною є внутрішній порушник та 555 випадків витоку інформації (44,4% від загального числа), що сталися по причині зовнішнього впливу [14]. На рисунку 1.1 надано статистику головних винуватців витоку інформації за 2019 та 2020 роки.

Як бачимо зі статистики, основна частина витоків відбувається з вини власних співробітників. Чим більш розвиненими стають інформаційні системи, тим

більш серйозними та збитковими стають загрози для організації від власних співробітників.

Рисунок 1.1 – Статистика винуватців витоку інформації

Для того щоб правильно розмежувати доступ і вирішити ці завдання, треба чітко розуміти, який саме користувач виконує дію (аутентифікація) та чи має він право на такі діє в системі (авторизація). Найбільш складним завданням є саме розв’язання проблеми авторизації.

На сьогодні існує кілька найбільш поширених методів контролю доступу, які також можна поєднувати між собою [15]:

- контроль доступу на основі атрибутів (ABAC);
- рольовий контроль доступу (RBAC);
- контроль доступу на базі списків доступу (ACL);
- контекстний контроль доступу (CBAC) ;
- контроль доступу на основі правил;
- контроль доступу на основі часу;

Головна складність полягає саме у виборі для подальшого впровадження найбільш відповідної моделі контролю доступу для різних інформаційних систем. Ці системи відрізняються кількістю ролей, які задіяні у бізнес процесі, різним рівнем складності самих бізнес-правил, наявністю чи відсутністю динамічних атрибутів, а також необхідністю забезпечувати контроль доступу до дій, даних тощо.

Модель управління доступом відіграє основну роль в системі безпеки інформаційної системи. Мета моделі – вираження суті вимог з безпеки до даної системи [16]. Вона визначає потоки інформації, дозволені в системі, і правила надання доступу до інформації. Модель не накладає обмежень на реалізацію тих чи інших механізмів захисту. Хороша модель безпеки має властивості абстрактності, простоти і адекватності системі, яка моделюється.

На рисунку 1.2 наведено тріаду AIC, компоненти якої згідно з [17] є основним принципами побудови безпечної системи.

Рисунок 1.2 – Тріада AIC

– доступність – це ознака ресурсу системи, яка означає, що користувачу (процесу), який має певні дозволи, надається можливість використовувати ресурс згідно з правилами, які встановлені політикою безпеки, при цьому не потрібно чекати довше заданого періоду часу, тобто коли ресурс присутній у вигляді, який

потрібен користувачеві, в місці, яке потрібно користувачеві, і в час, в який він йому потрібен;

– цілісність – це ознака інформації, яка означає, що інформація може модифікуватися лише авторизованим користувачем (процесом);

– конфіденційність – це ознака інформації, яка означає, що інформація може отримуватися лише авторизованим користувачем (процесом).

1.2 Огляд існуючих систем

На сьогодні існує багато програмних продуктів, які допомагають налаштувати та впровадити авторизацію в інформаційну систему, при цьому зовсім не багато з них надають консультацію саме з вибору потрібної моделі доступу: ABAC, RBAC, тощо. Але зазвичай саме проблема вибору відіграє найважливішу роль на перших етапах проектування нової або, якщо система вже існує, при суттєвих змінах бізнес процесів і підходів. Так як головна мета даної роботи проектування системи, яка б могла допомогти з вибором моделі авторизації доступу, розглянемо декілька вже існуючих систем, які також надають схожу можливість.

1.2.1 Огляд IDM365

Одним з таких продуктів є IDM365 (https://idm365.com/), який не тільки допоможе впровадити необхідну модель доступу, але й зможе надати консультацію яку саме модель краще обрати. IDM365 розроблений для середнього та великого бізнесу, а орієнтований на користувача інтерфейс дозволяє приймати важливі для

бізнесу рішення саме там, де знання та інформація знаходяться під контролем консультантів продукту, як зображено на рисунку 1.2.

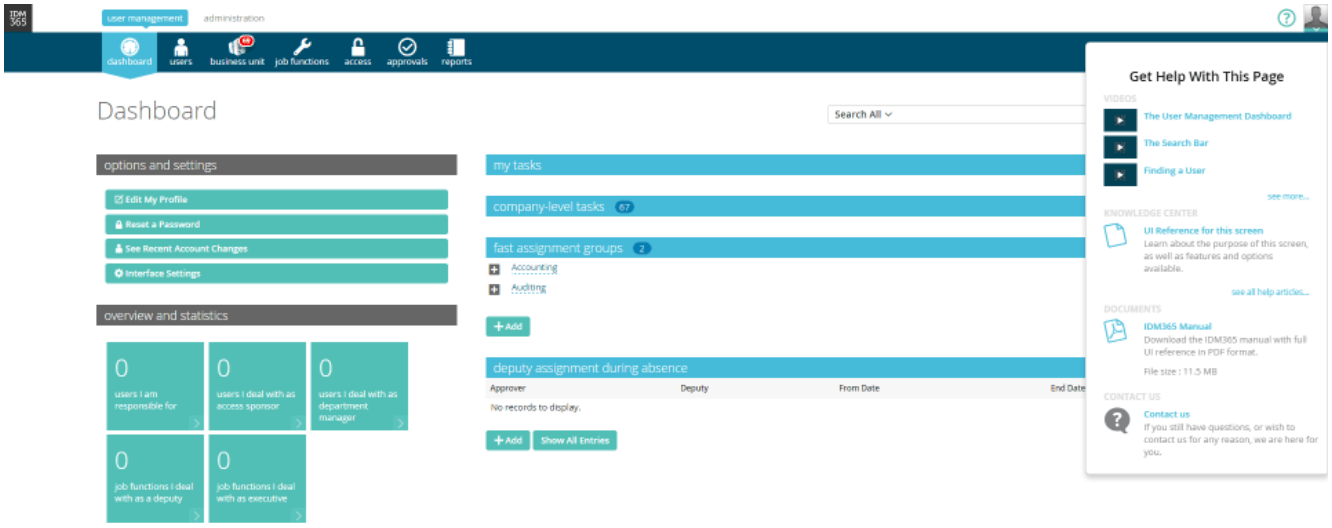

Рисунок 1.2 – Адміністративна панель IDM365

IDM365 – це набір процесів і інструментів, які забезпечують легку реалізацію і постійне обслуговування ролей ідентифікації та управління доступом. IDM365 дотримується принципів управління доступом на основі ролей (RBAC) і управління доступом на основі атрибутів (ABAC). Бекенд IDM365 може підключатись практично до будь-якої системи чи додатку, а також має інструменти, які дозволяють пришвидшити процес впровадження, забезпечуючи мінімальні витрати при збереженні максимальної точності та контролю.

Основні особливості системи:

– зручна конфігурація необхідних атрибутів з інтерфейсу IDM365, що використовується для управління ідентифікацією та доступом;

– спеціальні атрибути для кожного клієнта, які можна синхронізувати з системами, заснованими на логіці та бізнес-правилах;

– можна налаштовувати типи даних для запитуваних атрибутів.

В цілому система має свої програмні розробки, які дозволяють аналізувати інформаційні системи та автоматично генерувати пропозиції щодо необхідних ролей та заліковування їх з існуючими бізнес процесами. Вагомим недоліком є те,

що подібний підхід може використовуватися тільки для вже існуючих інформаційних систем і націлений більше на оптимізацію моделі доступу, ніж на створення її з нуля.

1.2.2 Огляд Okera

Другим прикладом програмного продукту є платформа Okera (https://www.okera.com). Вона дозволяє створювати та керувати політиками доступу до даних за допомогою інтуїтивно зрозумілого інтерфейсу, приклад якого наведено на рисунку 1.3. Менеджери даних можуть створювати політики доступу без написання рядка коду, виключно завдяки інтерфейсу користувача.

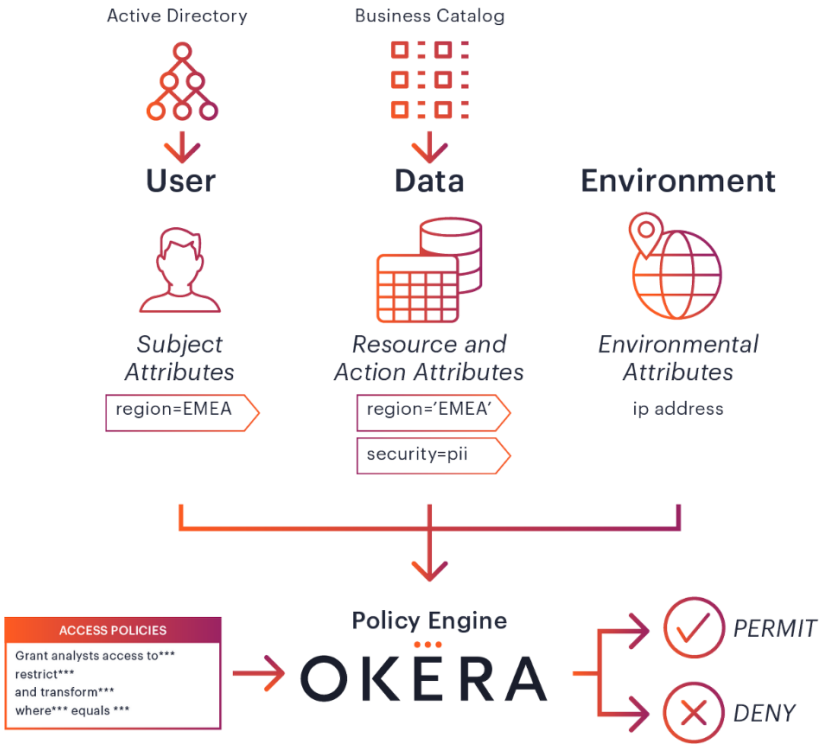

Рисунок 1.3 – Okera. Управління доступом на основі атрибутів (ABAC)

Okera позиціонує себе як платформа яка вирішує найскладніші проблеми, пов'язані з доступом до корпоративних даних, безпекою та управлінням у

розподілених середовищах та не складною інтеграцією з інформаційною системою замовника.

Зазвичай платформа дозволяє керувати доступом на основі атрибутів (ABAC), але для конкретності та виключень не забороняє використовувати підхід на основі ролей (RBAC). Привілеї та обмеження доступу можна застосовувати на багатьох рівнях деталізації файлі, стовпці, рядку та навіть на рівні комірки.

Основні особливості платформи, які зображені на рисунку 1.4.

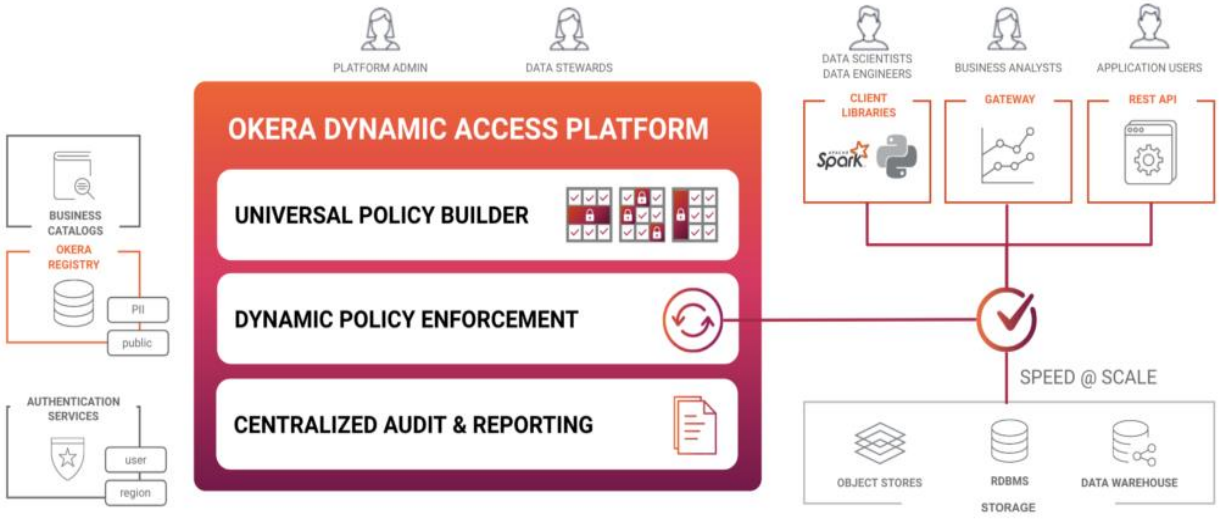

Рисунок 1.4 – Основні особливості Okera

– універсальний конструктор політик, який дозволяє розпорядникам даних створювати та керувати детальною політикою доступу до даних. Політика на 100% орієнтована на задоволення бізнес-потреб та створюється незалежно від технологій -інформаційної системи замовника;

– динамічне застосування політики. Okera стверджує, що завжди обиратиметься найкращий шаблон забезпечення, беручи до уваги контекст користувача, дані, що запитуються, та можливості контролю доступу для даного інструменту чи середовища;

– централізований аудит та звітність. Фіксується кожна дія як щодо даних, так і метаданих – від запитів до змін у політиці – для журналу аудиту, який залишається послідовним для кожного інструменту, що здійснює доступ до даних.

Ряд вже вбудованих звітів відповідає на важливі питання про те, хто використовує дані та з якими цілями, з можливістю детального вивчення точних детальних відомостей про конкретних користувачів.

Платформа Okera інтегрується з LDAP Directory та службами аутентифікації, такими як Microsoft Active Directory, тому політики контролю доступу до даних, пов'язані з певними ролями, можуть бути надані попередньо визначеним групам у межах підприємства. Це дозволяє підвищити ефективність надання доступу до даних окремим користувачам, що відображено на рисунку 1.5.

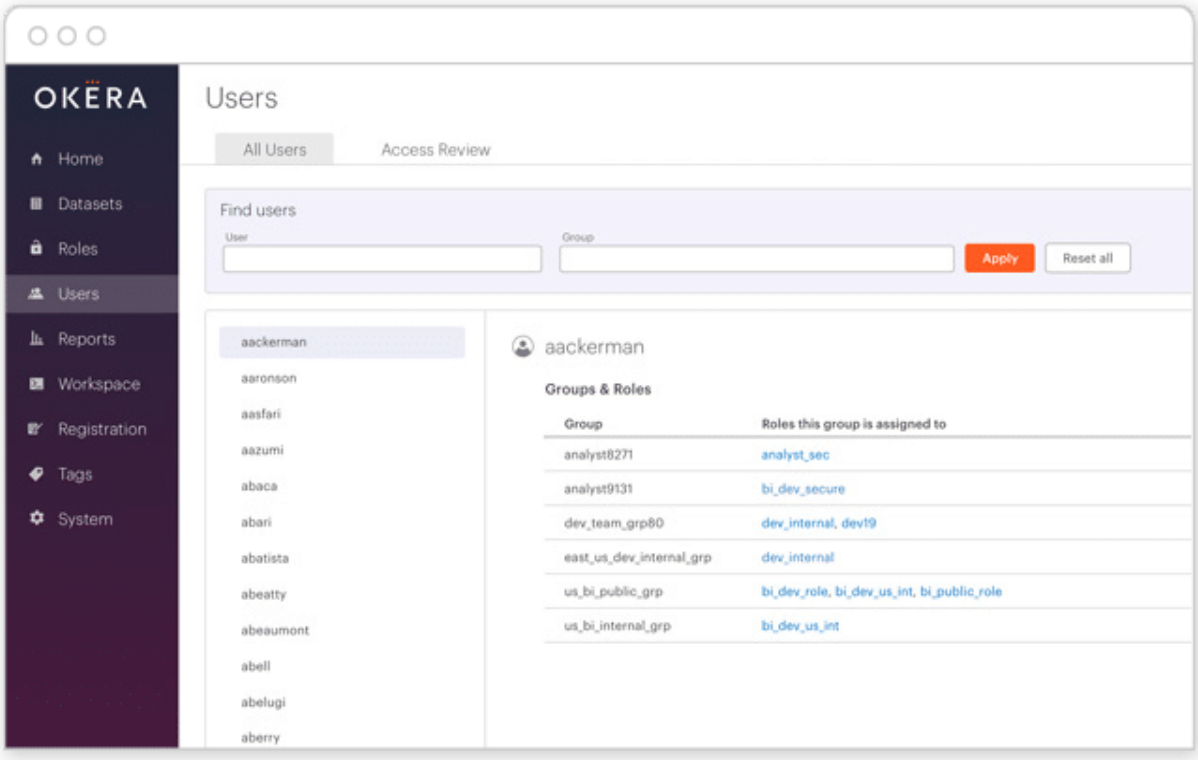

Рисунок 1.5 – Адміністративна консоль Okera. Керування користувачами

В цілому Okera дає досить гнучкий підхід, для організації доступу до даних, але вона розрахована на великі підприємства зі складною технічною та організаційною інфраструктурою і враховуючи її вартість зовсім не підходить для малих та середніх компаній.

1.2.3 Огляд Keycloak

Наступний аналог, який також можна розглянути є Keycloak (https://www.keycloak.org/). Це система управління ідентифікацією та доступом з відкритим кодом, спрямована на сучасні програми та послуги, має дуже детальну документацію. Окрім авторизації користувачів, також доступні з коробки такі функції як зберігання, аутентифікація та федерація користувачів, Single-Sign On та соціальний вхід. Keycloak має вбудовану підтримку для підключення до існуючих серверів LDAP або Active Directory. Також можна застосувати власного провайдера, якщо є користувачі в інших системах, наприклад, в реляційній базі даних. Зручна адміністративна панель дозволяє легко керувати користувачами, що видно на рисунку 1.6.

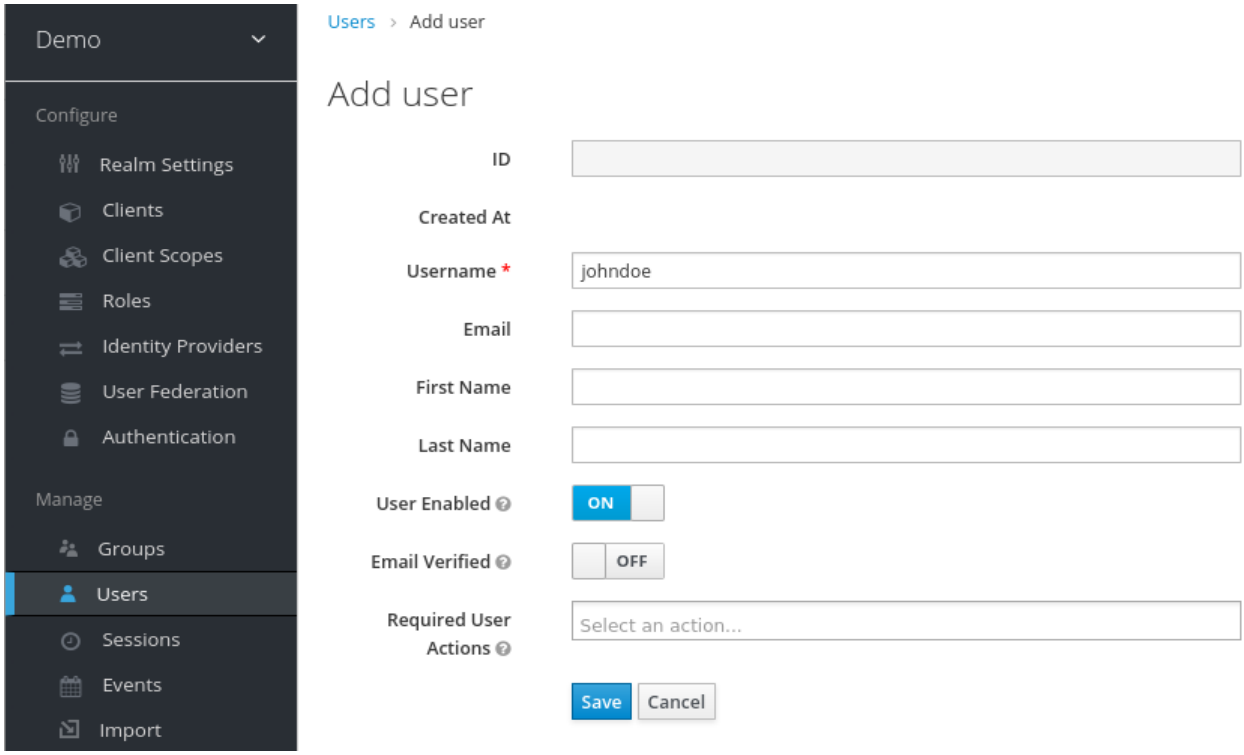

Рисунок 1.6 – Адміністративна панель Keycloak. Керування користувачами

Клієнтські адаптери Keycloak доступні для ряду платформ та мов програмування, наприклад Java, C#, Javascript, Python, Android, iOS та інших. Але

якщо для обраної платформи немає жодного, Keycloak побудований на стандартних протоколах, тому можна використовувати будь-яку бібліотеку ресурсів OpenID Connect або бібліотеку постачальника послуг SAML 2.0.

Keycloak пропонує авторизацію на основі ролей, а також надає детальні послуги авторизації на основі атрибутів. Керувати дозволами для всіх своїх служб можна за допомогою консолі адміністратора, загальний вигляд якої зазначено на рисунку 1.7, де надається можливість визначати потрібні політики доступу.

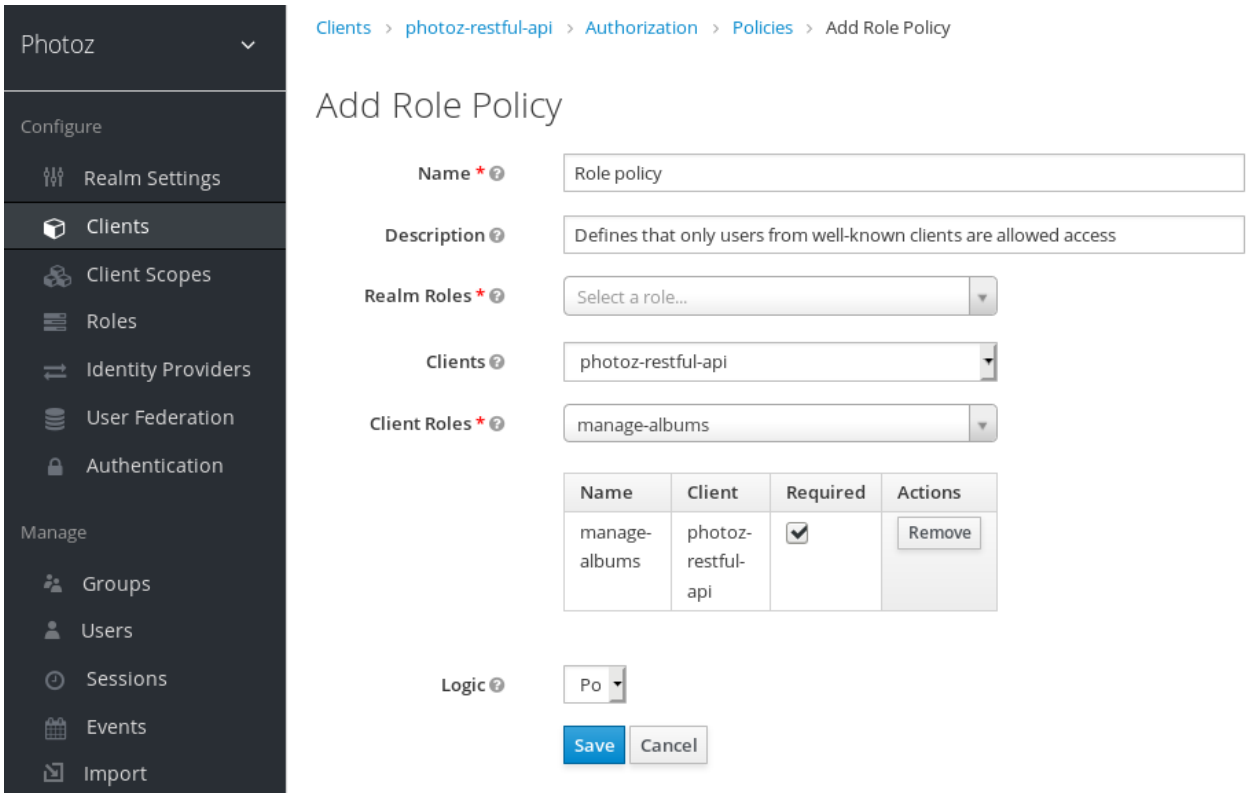

Рисунок 1.7 – Приклад додавання ролевої політики в Keycloak

Keycloak підтримує чіткі політики авторизації та може поєднувати різні механізми контролю доступу, такі як:

- контроль доступу на основі атрибутів (ABAC);
- рольовий контроль доступу (RBAC);
- контроль доступу на основі списків доступу (ACL);
- контроль доступу на основі контексту (CBAC);
- контроль доступу на основі правил;

- використання JavaScript;
- контроль доступу за часом.

Перевага Keycloak в тому, що це опенсорс проект, має сумісність з більшістю мов програмування, але ця система розрахована здебільшого на досвідчених користувачів, тому що поріг входу досить високий.

1.3 Постановка завдання на дослідження

Проблема задачі, яка досліджується, полягає саме у виборі найбільш відповідної моделі авторизації при будь-якому структурованому наборі вхідних параметрів, які визначатимуть ключові задачі та пріоритети бізнесу. Іншими словами, це дослідження дасть відповідь на питання, як вибрати необхідний підхід, щоб він не був достатньо складним і дорогим для певної інформаційної системи, але при цьому, щоб ця модель авторизації повністю покривала всі необхідні вимоги і її не довелося повністю змінювати через рік-два.

Для цього будуть вивчатися найбільш розповсюджені на сьогодні моделі доступу, їх переваги та недоліки, а також буде сформовано набір критеріїв, за яким та чи інша модель буде обиратися, як оптимальний вибір для певного набору вимог.

Критерії вибору певної моделі доступу будуть базуватися на створенні «паспорта системи». По суті, це опитування по конкретній інформаційній системі, в якому детально зафіксовані всі параметри та атрибути управління доступом до неї.

Результат, який очікується після виконання роботи, це створення алгоритму вибору та на його базі прототипу програми, завдяки якій, знаючи лише певний набір вхідних даних (приблизна кількість користувачів, ієрархічна складність бізнес-процесів, наявність тимчасового доступу, перспективи розширення філій тощо) розраховуючи кількість необхідних ролей, привілеїв та атрибутів система буде надавати рекомендацію та обґрунтування, яка сама модель або поєднання моделей

найбільше підходить для даного випадку. Ця інформація значно поліпшить процес створення інформаційної системи, допоможе зробити вірний вибір моделі авторизації, який надалі буде безпосередньо впливати на безпеку даних та вартість операційного часу підтримки та розробки нових політик безпеки.

2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

2.1 Модель доступу ACL

Модель ACL – це перелік прав доступу до об’єкта, в якому визначається чи може суб’єкт отримувати доступ до нього, а також які саме операції суб’єкту дозволено чи заборонено проводити над об’єктом.

Списки контролю доступу це основа систем де управлінням доступом є вибірковим. Рішення про авторизацію приймається до будь-якого конкретного запиту на доступ. Якщо суб’єкт робить запит на виконання операції над об’єктом, він спочатку надає облікові дані, які система перевіряє у списку операцій, які дозволені для цього суб’єкта, і тільки потім дозволяє або не дозволяє доступ до ресурсу. Індивідуальні привілеї. що дозволяють виконання операцій (читання, запис, редагування, видалення тощо) керуються індивідуально власником об’єкта. Кожен об’єкт потребує власного ACL та набору привілеїв, призначених кожному суб’єкту.

Основні правила моделі ACL наведено на рисунку 2.1.

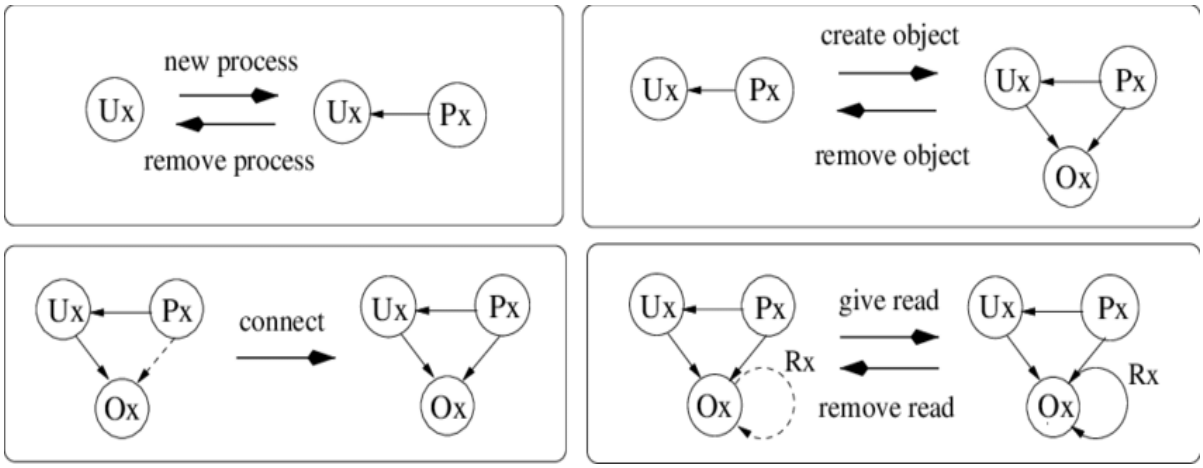

Рисунок 2.1 – Модель ACL

Існує поняття матриці доступу[18], де списки контролю доступу зберігаються централізовано, при цьому по осях розміщуються об’єкти та суб’єкти, а відповідні

права в клітинках. Але велика кількість систем зберігає ці списки окремо для кожного об’єкта, і зазвичай це робиться поруч з самим об’єктом.

Системи, які використовують ACL умовно поділяють на дві категорії:

– дискреційні – власник об’єкта може повністю контролювати доступ до цього об’єкта, в тому числі тих, кому дозволено змінювати права доступу до об’єкта;

– мандатні – якщо задані користувачем ACL перекриваються системними обмеженнями.

Досить часто ACL використовується для різноманітних файлових систем, при цьому головну роль відіграє ідентифікатор користувача процесу.

В традиційних ACL системах права надають індивідуально користувачам. З часом кількість користувачів в системі зростає і списки доступу стають досить об’ємними. Окрім того помилки під час видалення або скасування доступу з часом призводить до накопичення недійсних привілеїв користувачів, що в цілому впливає на безпеку системи не найкращим чином.

Цю проблему можна частково вирішувати через надання прав не персонально, а групам. Інший підхід для подолання цієї проблеми – управління доступом на основі ролей, тобто використання RBAC, де функціональні підмножини прав до ряду об’єктів об’єднуються в «ролі», а потім користувачам призначаються ці ролі.

2.2 Модель доступу RBAC

Модель RBAC [19] контролює доступ користувачів на основі виконуваних ними завдань (ролей). Роль є семантична конструкція, що лежить в основі політики обмеження доступу. Під роллю розуміється сукупність дій і обов’язків, пов’язаних з певним видом діяльності. Ролі дозволяють отримати конкретним особам доступ

до ресурсів в тій мірі, в якій це необхідно їм щоб виконувати свої обов'язки. У моделі RBAC використовуються такі терміни:

– користувач – авторизований користувач системи;

– об'єкт – ресурс системи, до якого регулюється доступ;

– привілей – мінімально можлива атомарна дія користувача, яка підпадає під дію механізму розмежування доступу;

– роль – це набір прав, що визначають, якими привілеями і над якими об'єктами буде спроможнім виконувати дії користувач, якому присвоюється дана роль;

– операція – складова частина ролі, яка визначає привілей, підмножина об'єктів, що володіють даним привілеєм, і щоб дозволити чи заборонити виконання даної дії;

– сесія – безліч ролей даного користувача в певний проміжок часу (одночасно може виконуватися декілька сесій одного і того ж користувача).

Основні елементи класичної моделі RBAC наведено на рисунку 2.2.

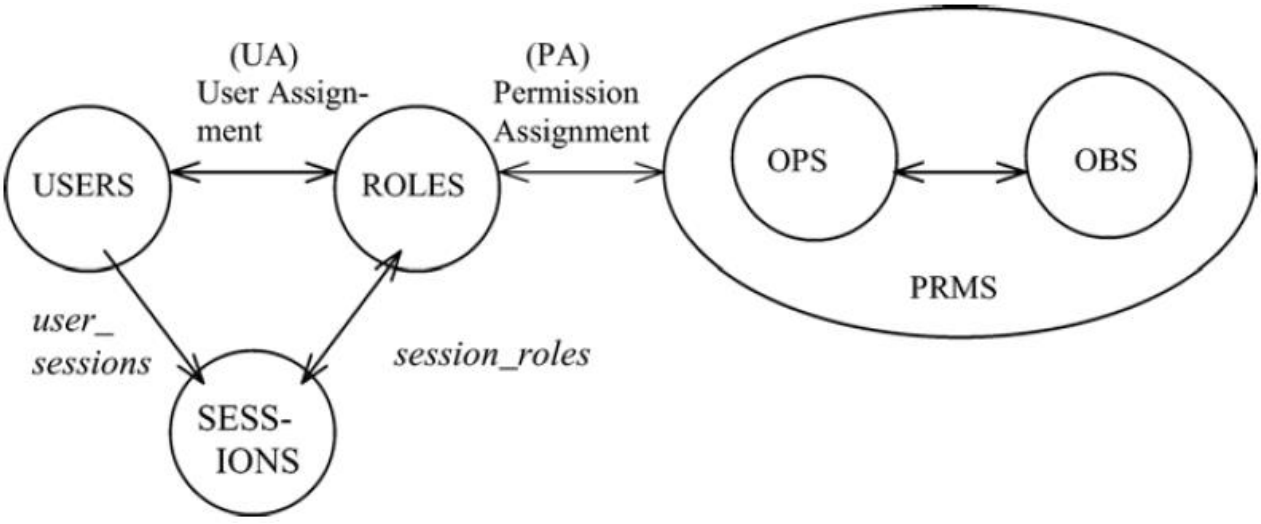

Рисунок 2.2 – Модель RBAC

Процес доступу складається з наступних кроків:

– користувач робить запит про доступ до ресурсу за допомогою людино-машинного інтерфейсу (призначеного для користувача інтерфейсу, файлу даних,

апаратного пристрою);

– служба аутентифікації перевіряє його облікові дані. У разі відсутності прав користувачеві відправляється повідомлення і доступ не надається;

– якщо користувач пройшов аутентифікацію, далі відбувається перевірка прав запитаного доступу до об’єкта на підставі присвоєної користувачеві ролі.

– у разі успіху відбувається надання права на доступ до ресурсу, в протилежному випадку – відмова.

Суть підходу RBAC полягає в створенні набору ролей, які повторюють бізнес ролі. Засновуючись на цих ролях система перевіряє можливість виконання користувачем тих або інших дій. Якщо всі (або принаймні більша частина) бізнес-правила одномірні і всі дії можна розбити по ролям (бухгалтер, менеджер, адміністратор і т. ін.), то такого підходу буде досить. В такому випадку одному бізнес-правилу буде відповідати одна роль.

2.3 Модель доступу ABAC

Як було зазначено вище для одномірних бізнес-правил цілком підходить модель RBAC, але бізнес-правила з часом стають багатовимірними та ускладнюються. Як наслідок одного атрибута (ролі) для вираження всіх бізнес-правил стає недостатньо і починають додаватися інші атрибути (місто, країна, філія, день тижня, власник, ліміт і т. ін.). Щоб впоратися з цією складністю, виникає необхідність у створенні додаткових ролей, кількість яких буде дорівнювати числу комбінацій всіх атрибутів. На кожне додавання нового значення атрибута доведеться додатково додавати нові ролі. З цього моменту починається стрімке зростання числа ролей, що значно ускладнює підтримку актуальності таких наборів та управління ними.

Крім цього існують бізнес-правила, в яких використовуються динамічні атрибути, значення яких заздалегідь невідомі і обчислюються вже в процесі роботи системи. Такі задачі взагалі неможливо виразити за допомогою рольової моделі.

Для можливості роботи саме з такими підходами та для подолання обмежень які накладає RBAC підхід, було створено інший підхід, який зображено на рисунку 2.3 ґрунтується він на атрибутах – ABAC.[20]

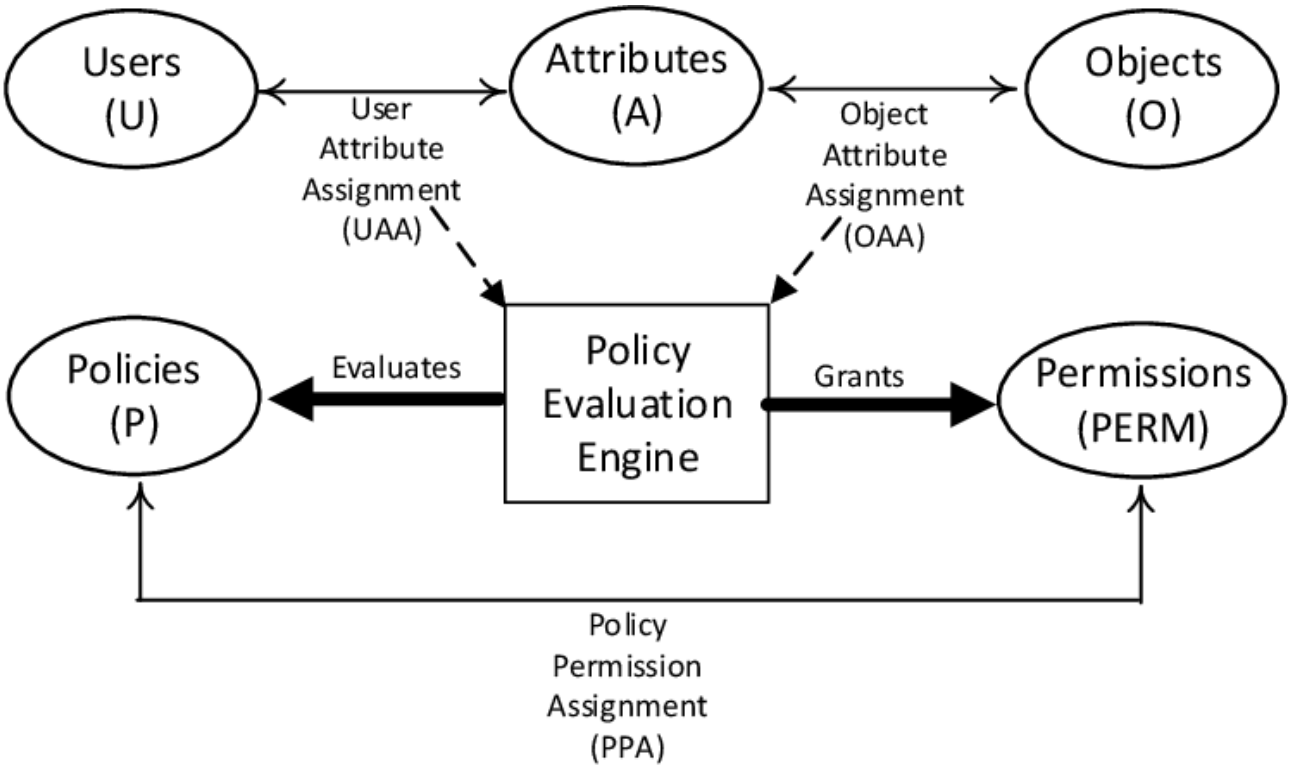

Рисунок 2.3 – Модель ABAC

Основна відмінність цього підходу полягає в тому, що будь-яка ситуація чи дія в системі оцінюється не з точки зору ролі користувача та дії, яку він планує зробити, а з точки зору атрибутів, які до них належать. Бізнес-правило – це, по суті, набір умов, в яких різні атрибути повинні задовольняти запропонованим до них вимогам.

Останнім часом було розроблено безліч ABAC-моделей, як основних (базових), так і спеціалізованих.[21-24] Їх всіх об’єднує те, що вони можуть бути

розглянуті в якості основних моделей нового напрямку захисту, які здатні вирішувати завдання розмежування доступу, з якими не справляється RBAC.

Типова модель ABAC містить наступні компоненти [25]:

– атрибути користувачів;

– атрибути об’єктів;

– атрибути контексту;

– політики авторизації, засновані на цих атрибутах.

Загальну схему моделі ABAC, яку наведено на рисунку 2.3, можна описати наступним чином:

– у моделі є безлічі сутностей E, суб’єктів $S \subset E$, прав доступу R і об’єктів-параметрів $P = \{p_1, \dots, p_m\} \subset E$. Кожній сутності поставлено у відповідність деяка множина атрибутів – змінних з кінцевими множинами значень, і набір значень атрибутів сутності e позначений як $A(e)$. Кожній трійці $(s, e, r) \in S \times E \times R$ поставлені у відповідність деякі параметри $q_1, \dots, q_n \in P$ і предикат, що залежить від $A(s), A(e), r, A(p_1), \dots, A(p_k)$ так, що суб’єкт $s \in S$ отримує право доступу $r \in R$ до сутності $e \in E$, коли правдивий цей предикат.

– в момент часу t стан моделі G_t визначається за формулою 2.1.

$$G_t = \left(E_t, V_t, \left(p_1, A(p_1)\right), \dots, \left(p_m, A(p_m)\right)\right) \tag{2.1}$$

де E_t – набір сутностей системи в момент часу t,

V_t – набір всіх реалізацій прав доступу суб’єктів до сутностей, які мають місце в момент часу t.

Сама множина V_t складається з елементів v_t, які обчислюються за формулою 2.2.

$$v_t = \left(\left(s, A(s)\right), \left(e, A(e)\right), r\right), s \in S, e \in E, r \in R \tag{2.2}$$

– траєкторією функціонування моделі називається кінцева послідовність

станів $G_{0,} \dots, G_t$, де G_0 – початковий стан, G_i виходить з $G_{(i-1)}$ або при появі нової сутності системи, або при зміні значення атрибута деякого об'єкта – параметра, або при отриманні суб'єктом деякого права доступу до сутності. Множина всіх траєкторій функціонування інформаційної системи з початковим станом G_0 позначається $P(G_0)$;

– відповідно до політики безпеки $P(G_0)$ розбивається на дві непересічні підмножини: $LP(G_0)$ – дозволених і $NP(G_0)$ – недозволених траєкторій, і визначаються множини $L_a, N_a, L_r, N_r, L_f, N_f$, дозволених і заборонених доступів, прав доступу та інформаційних потоків відповідно. Порушення безпеки інформаційної системи визначається як перехід в стан, в якому є заборонений доступ з N_aабо на траєкторії до якого відбулося отримання забороненого права доступу з N_r, або реалізований заборонений інформаційний потік з безлічі N_f.

ABAC не обмежує складність бізнес-правил. Завдяки більш зрозумілому бізнесу і компактному вислову цей підхід дозволяє не збільшувати вартість підтримки при реалізації досить складних правил, а також дає можливість забезпечувати контроль доступу не тільки до дій, але і до даних.

2.4 Змішані моделі доступу на базі RBAC та ABAC

В частині випадків проектування доступу до інформаційної системи виникає необхідність поєднати модель RBAC і модель ABAC, додавши правила, які, використовуючи різні атрибути об'єктів, можуть виконувати функцію розподілу відповідальності. Тоді набір привілеїв для користувача, буде визначатися за формулою 2.3.

$$\text{Privileges} = \text{UserRole} \times \text{Operation} + \text{UserRules} \qquad (2.3)$$

де Privileges – привілеї, необхідні у системі,

UserRole – ролі користувача у системі,

Operation – операції, які може виконувати користувач у системі,

UserRules – правила, за якими діє користувач у системі.

Тепер для розгортання рольової моделі достатньо закріпити за користувачем одну або кілька ролей і визначити правила, щоб визначити всі привілеї, якими він повинен володіти.

Для недопущення конфліктів ролей необхідно використовувати правила лише одного знаку – наприклад, тільки того, що дозволяє. Тоді у множині ролей, яке складене лише з дозвільних правил, привласнення користувачеві ще однієї ролі призводить тільки до розширення його прав доступу, без внесення обмежень на вже існуючі.

Деякі системи можуть мати ієрархічну структуру, тому для них доцільно застосовувати модель рольового управління доступом з ієрархією сутностей – HRBAC. Тоді співробітники підрозділів, що знаходяться вище за ієрархією зможуть отримувати права на доступ до даних підрозділів, які нижче. У той же час співробітники підрозділів, що знаходяться нижче за ієрархією не зможуть без спеціального дозволу отримати доступ до даних підрозділів, які знаходяться вище. При використанні ієрархічного рольового управління доступом набір привілеїв, привласнених користувачеві, буде визначатися формулою 2.4

$$\text{Privileges} = \text{Level}(\text{UserRole} \times \text{Operation} + \text{UserRules}) \quad (2.4)$$

де Privileges – привілеї, необхідні у системі,

Level – кількість рівнів в ієрархії,

UserRole – ролі користувача у системі,

Operation – операції, які може виконувати користувач у системі,

UserRules – правила, за якими діє користувач у системі.

Окрім рівнів ієрархії певні системи також можуть вимагати врахування часу та/або місця з якого відбувається доступ до інформаційної системи. У таких випадках роль повинна бути доступна тільки протягом певного часового інтервалу. Розширення моделі RBAC, що підтримує дані вимоги, отримало назву TRBAC. Ще

одним корисним розширенням моделі RBAC в контексті інформаційних систем, які мають враховувати місцеположення, є модель SRBAC, яка дозволяє обмежувати / дозволяти доступ в залежності від місця знаходження члена ролі. Спільне використання TRBAC і SRBAC дозволяє будувати правила, що дозволяють, наприклад, доступ до певних ресурсів з 10:00 до 18:00 в робочі дні тільки з офісу. З урахуванням часу і місця розташування набір привілеїв користувача визначається формулою 2.5.

$$\text{Privileges} = Location\left(Time\left(Level(\text{UserRole} \times \text{Operation} + \text{UserRules})\right)\right) \quad (2.5)$$

де Privileges – привілеї, необхідні у системі,

Location – кількість місць, з яких має бути організований доступ,

Time – кількість часових проміжків,

Level – кількість рівнів в ієрархії,

UserRole – ролі користувача у системі,

Operation – операції, які може виконувати користувач у системі,

UserRules – правила, за якими діє користувач у системі.

Такий підхід більш гнучкий, ніж чиста модель RBAC, так як дозволяє будувати моделі доступу на основі атрибутів суб'єкта ABAC. Відпадає необхідність в створенні окремих ролей, можна просто змінювати атрибути правил. Платою за гнучкість є ускладнення моделі. У разі n атрибутів ми маємо справу з 2n можливими комбінаціями значень. Використовуючи чисту модель RBAC легко адмініструвати права, але потрібно багато часу для розробки самої моделі розмежування доступу. І, навпаки, модель ABAC проста в налаштуванні, проте, аналіз і зміна прав користувачів може виявитися проблематичним. Саме тому для багатьох інформаційних систем компромісом є об'єднання моделей RBAC і ABAC для того, щоб скористатися їх сильними сторонами.

В цілому, основні передумови вибору відповідної моделі контролю доступу до інформаційних систем можна описати наступним чином:

– кількість ролей невелика – найкраще вибрати модель авторизації RBAC;

– одному бізнес-правилу буде відповідати лише одна роль – також достатньо буде RBAC;

– окремої ролі для вираження бізнес-правил вже недостатньо і починають додаватися інші атрибути (місто, країна, день тижня, власник, ліміт і т.п.) – більш ефективніше буде застосувати модель контролю доступу ABAC;

– існує ієрархії ролей – слід використовувати HRBAC.

В таблиці 2.1 наведено загальний аналіз RBAC та ABAC моделей.

Таблиця 2.1 – Порівняння підходів RBAC та ABAC

	RBAC	ABAC
Підтримка простих бізнес-правил	+	+
Підтримка складних бізнес-правил	+	+
Підтримка правил з динамічними параметрами	-	+
Без зайвої ручної праці	-	+
Опис, близький до бізнес-термінів	-	+
Фільтрація даних (тобто контроль не тільки дій)	-	+

Як видно з порівняння двох підходів, RBAC найбільше підходить для реалізації простих бізнес-правил. Зі збільшенням складності правил доцільність використання RBAC зменшується через зростання вартості підтримки системи контролю доступу, а починаючи з певного рівня складності правил цей підхід взагалі не дає результату [26].

ABAC, в свою чергу, не обмежує складність бізнес-правил. Завдяки більш зрозумілому бізнесу і компактному вислову цей підхід дозволяє не збільшувати вартість підтримки при реалізації більш складних правил, а також дає можливість забезпечувати контроль доступу не тільки до дій, але і до даних.

Обґрунтований правильний вибір моделі авторизації може значно заощадити час розробки та підтримки інформаційної системи.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Модель проведення дослідів

Метою дослідження є вибір найбільш оптимального метода авторизації до інформаційної системи на основі критеріїв, отриманих шляхом опитування певної кількості респондентів, які приймають безпосередню участь в розробці системи. Вирішення подібних задач здебільшого виконуються за допомогою ТПР.

Теорія прийняття рішень – це область дослідження, де використовуються різні поняття та методи статистики, математики, економіки, психології і менеджменту. Вона вивчає різні закономірності вибору людьми варіантів вирішення різних завдань, крім цього досліджує шляхи пошуку самих вигідних з усіх можливих рішень.[27]

Сьогодні ТПР використовується здебільшого для аналізу проблем, які можна досить просто та чітко формалізувати, а результати досліджень, які будуть отримано інтерпретувати однозначно. Методи ТПР застосовують в багатьох сферах та галузях.

Необхідність застосування засобів і методів ТПР в першу чергу продиктована швидким розвитком і ускладненням економічних зв’язків, виявленням залежностей між окремими складними явищами та процесами, які до цього здавалися відокремленими одне від одного. Все це призводить до різкого зростання складності прийняття обґрунтованого рішення. Витрати на такі прийняття рішень збільшуються, наслідки помилок стають більш серйозними, а звернення до інтуїції та досвіду не завжди гарантує вибір найбільш відповідної стратегії. Використання методів ТПР дає можливість вирішити це питання, більш того, зробити це точно, швидко та ефективно.

В контексті ТПР рішення – це результат дослідження, що підводить до конкретного висновку або до дій, які необхідно виконати. Тут здійснюється вибір між альтернативними можливостями, які, як правило, є конкуруючими.[28–29]

Ще одним терміном, яким оперує ТПР є альтернатива. Альтернативи – це варіанти рішень, які приймаються. Якщо множина альтернатив порожня, то немає бути і самого вибору, отже необхідно мати хоча б дві альтернативи для проведення будь-якого дослідження з ціллю прийняття рішення. Будь-які задачі прийняття рішень можливо розділити на задачі наступних класів:

– задача розподілу альтернатив на упорядковані за якістю групи;

– задача впорядкування альтернатив (визначення порядку у множині альтернатив);

– задача вибору кращої альтернативи.

Альтернативи можуть бути попередньо заданими, конструйованими в процесі прийняття рішень, отриманими після вироблення правила прийняття рішень.[30]

Процес прийняття рішення – це послідовність операцій, які призводять до знаходження рішення. Він складається наступної послідовності дій:

– визначення проблеми та постановка задач;

– описання властивостей зовнішнього середовища для визначення можливих альтернатив вибору;

– визначення характеристики рішення та формалізація характеристики до рівня критеріїв;

– вибору способу оцінку;

– оцінювання якості альтернатив, їх порівняння, а також вибір однієї чи декількох альтернатив, що найбільш повно відповідають поставленій меті.[31]

У рамках цієї роботи, головною задачею дослідження є пошук найкращого рішення (найбільш оптимальної моделі) при наявності різноманітних комбінації критеріїв для визначення оптимальності. У випадку, коли задача може вирішуватися засобами математичного програмування, така задача належить до класу задач багатокритеріальної оптимізації. Задачі такого типу можуть носити нелінійний та лінійний характер та вирішуватися за допомогою багатокритеріальної задачі оптимізації. При цьому задачу, яка має чітко описаний набір альтернатив та чітко

визначений принцип оптимальності прийнято називати задачею вибору. Загальну схему прийняття рішень описано на рисунку 3.1.

Рисунок 3.1 – Схема прийняття рішень

У різноманітних прикладних задачах при формалізації етапів процесу прийняття рішень виникають деякі, іноді досить складні проблеми. Першою з них є проблема коректної постановки мети та визначення інструментів для досягнення цієї мети. Досить легко припуститися помилки та встановити недосяжну ціль. Формалізація ЗЗПР та її описання за допомогою мови математики, щоб змоделювати прикладні ситуації прийняття рішень, має винятковий інтерес. Саме під час описання математичної моделі, може виявитися що задача не має розв’язку та поставлена мета насправді недосяжна.

Наступною проблемою, яку необхідно вирішити є знаходження множини альтернатив, тобто варіантів вибору, кожен з яких спрямовано на досягнення мети.

Перше, що необхідно зробити на другому етапі, це побудувати якомога більш повний список альтернатив. Іноді, пропустивши лише одну з них, можна отримати

цілком невірний висновок, або навіть не отримати розв’язок задачі. Друге – зробити оцінку альтернатив, тобто визначити можливі результати обраної дії. Найчастіше, визначені для альтернатив оцінку мають досить особистий характер, тому що базуються на досвіді та знаннях певного експерта. Навіть при підході, коли альтернативи оцінюються за допомогою об’єктивних процедур, то проблема відбору найбільш важливих критеріїв оцінки кожної з альтернатив нікуди не зникає, а як було зазначено вище, втрата бодай одного з таких критеріїв може вести до цілком неочікуваного результату.

Ще однією проблемою ЗЗПР є визначення принципу, за яким будуть порівнюватися альтернативи, та як наслідок визначення самого принципу оптимальності. На попередньому етапі кожен критерій був визначений числовою оцінкою. На цьому етапі вже принцип оптимальності полягає у виборі саме тих критеріїв оптимізації, які найбільше відповідають меті загальної задачі прийняття рішень.

Коли мета ЗЗПР має декілька числових критеріїв, то сама задача буде зводитися до задачі багатокритеріальної оптимізації. При такому підході визначається головний принцип, на базі якого буде проводитися порівняння, для знаходження найкращої альтернативи.

Самі критерії можуть бути несуперечливими один до одного, такими що доповнюють один одного або взагалі не мати жодного зв’язку. Вирішення багатокритеріальної задачі завжди пов’язане з експертними оцінками критеріїв та їх відношення один до одного. Ці задачі мають наступні методи:

– оптимізація одного, визначеного як найбільш важливий, критерія;

– упорядкування всієї множини критеріїв та покрокова оптимізація кожного з них;

– зведення декількох критеріїв до одного шляхом використання вагових коефіцієнтів для критеріїв. Чим більш важливий критерій тим більшу вагу він отримує.

Всі методи вирішення задач прийняття рішень, які засновані на приведенні багатокритеріальної задачі оптимізації до скалярної за допомогою введення деякого узагальненого критерію проходять за наступною схемою:

– абсолютно всі обрані критерії нормують, тобто вони приводяться до порівняльного безрозмірного вигляду;

– після нормування виконується їх «згортання» в єдину цільову функцію. Ця функція називається «узагальнений критерій». Під час згортання враховується відносна важливість критерія з огляду на його вагових коефіцієнт.

Результатом зазначених вище перетворень багатокритеріальна задача оптимізації зводиться до звичайної задачі оптимізації за одним критерієм.

Найбільш розповсюдженими на сьогодні згортками є:

– мультиплікативна згортка критеріїв. Така згортка заснована на принципі справедливої сатисфакції відносних змін приватних критеріїв. Основною перевагою такої згортки є спрощення задачі аналізу завдяки зведенню її до однокритеріальної задачі. Недоліком є невиправдана компенсація одних ефектів іншими та неможливість розділити та порівняти такі ефекти;

– узагальнення визначених критеріїв, за допомогою середньозваженої функції. Серед таких згорток особливо можна виділити лінійну згортку критеріїв. Вона є зручною у застосуванні, а також дозволяє зберігати лінійність результуючих функцій. Це означає, що у випадку коли вихідні критерії лінійні, то отримуваний результуючий критерій також буде лінійним [32].

Вирішення задачі даного дослідження буде проводитися за допомогою лінійної адитивної згортки з ваговими коефіцієнтами.

Вагові коефіцієнти – це параметри, які показують важливість даного критерія у порівнянні з іншими критеріями, які розглядаються в рамках вирішення дослідження. Частіше за все вага обирається засновуючись на інтуїтивній ідеї відносної важливості критеріїв.[33] В загальному вигляді вага кожного з критеріїв (або групи критеріїв) розраховується за формулою 3.1.

$$\sum_{i=1}^{n} \beta_i = 1, \beta_i \geq 0 \quad (3.1)$$

де β_i – ваговий коефіцієнт критерія або групи критеріїв, при цьому $\beta_{i+1} \geq \beta_i$, якщо критерій f_{i+1} є більш переважним ніж f_i.

Адитивної згортку з ваговими коефіцієнтами визначає формула 3.2.

$$Z = max \sum_{i=1}^{n} \alpha_i \beta_i a_{ij} \quad (3.2)$$

де Z – лінійна адитивна згортка,

α_i – нормуючи множники, які розраховуються за формулою 3.3,

β_i – вагові коефіцієнти, що відображають відносний внесок окремих критеріїв до загального результату,

a_{ij} – коефіцієнт j критерія для i стратегії.

$$\alpha_i = \frac{1}{\sum_{j=1}^{m} a_{ij}} \quad (3.3)$$

де α_i – нормуючий множник,

a_{ij} – коефіцієнт j критерія для i стратегії.

Врахувавши всі вищенаведені дані в таблиці 3.1 можемо надати матрицю прийняття рішень для вирішення поставленої задачі у загальному вигляді.

Таблиця 3.1 – Матриця прийняття рішень

	Альтернатива 1	…	Альтернатива j
Критерій 1	a_{11}	…	a_{1j}
…	…	…	…
Критерій i	a_{i1}	…	a_{ij}

Основною перевагою адитивної згортки є те, що з нею пов’язані «класичні», достатні і необхідні, умови оптимальності за Парето. Адитивна згортка особливо корисна, коли зменшення оцінки за якимось одним критерієм компенсується збільшенням оцінки за якимось іншим критерієм (або по декільком критеріям). Нескладно помітити, що навіть якщо за якимось критерієм оцінка нульова, загальна оцінка може бути цілком непоганою, якщо з іншими критеріями справи кращі.[34]

3.2 Проведення порівнянь альтернатив

Для впровадження програми було проведено інформаційну підготовку прийняття рішення з вибору моделі управління доступом – ACL, RBAC, ABAC та їх похідні. В таблиці 3.2 наведено комбінації моделей та опис формування дозволів для кожної з цих моделей в залежності від наявності користувача/ID суб’єкта (К), ролі (Р) або атрибута (А).

Таблиця 3.2 – Комбінації моделей ACL, RBAC та ABAC

К	Р	А	Модель	Відображення прав доступу
0	0	0	Не визначена	–
0	0	1	ABAC-основна	$A_1, A_2 \dots A_n \rightarrow permissions$
0	1	0	Не визначена	–
0	1	1	Гібридна ABAC-RBAC	$R, A_1, A_2 \dots A_n \rightarrow permissions$
1	0	0	Списки доступу ACL	$U \rightarrow permission$
1	0	1	ABAC-ID	$U, A_1, A_2 \dots A_n \rightarrow permissions$
1	1	0	RBAC-основна	$U \rightarrow R \rightarrow permission$
1	1	1	RBAC-A, динамічні ролі	$U, A_1, A_2 \dots A_n \rightarrow R \rightarrow permissions$
1	1	1	RBAC-A, на основі атрибутів	$U, R, A_1, A_2 \dots A_n \rightarrow permissions$
1	1	1	RBAC-A, на основі ролей	$U \rightarrow R \rightarrow A_1, A_2 \dots A_n \rightarrow permissions$

Задача знаходження оптимального рішення на основі даних, які будуть отримані від користувачів, можна віднести до класу задач, які можуть бути вирішені методами прийняття рішень в умовах визначеності. Отже є можливість навести векторний опис цієї задачі та знайти її найкраще рішення на базі метода теорії корисності.

Отже у якості множини альтернатив було обрано наступний набір моделей доступу до інформаційної системи:

- ABAC-основна;
- гібридна ABAC-RBAC;
- списки доступу ACL;
- ABAC-ID;
- RBAC-основна;
- RBAC-A, динамічні ролі;
- RBAC-A на основі атрибутів;
- RBAC-A на основі ролей;
- не визначена.

А також критерії вибору за якими буде проводитися оцінка певної моделі для впровадження у певну інформаційну систему:

- наявність користувачів/ID суб’єктів;
- наявність ролей;
- наявність атрибутів;
- необхідність динамічно змінювати атрибути для ролі;
- роль повинна відігравати у якості атрибута;
- необхідність додавати атрибут до ролі суб’єкта.

При цьому шкала оцінок мала би лише дві опції – критерій існує і впливає на вибір системи доступу та критерій відсутній і не впливає на вибір системи доступу. Результат такої вибірки можна побачити в таблиці 3.3.

Таблиця 3.3 – Шкала оцінок метода від наявності/відсутності критерія

	Існують користувачі	Існують ролі	Існують атрибути	Треба динамічно змінювати атрибути для ролі	Роль має бути у якості атрибута	Треба додавати атрибут до ролі суб'єкта
ABAC-Осн.	0	0	1	0	0	0
ABAC-RBAC	0	1	1	0	0	0
ACL	1	0	0	0	0	0
ABAC-ID	1	0	1	0	0	0
RBAC-Осн.	1	1	0	0	0	0
RBAC-А динамічні ролі	1	1	1	1	0	0
RBAC-А на основі атрибутів	1	1	1	0	1	0
RBAC-А на основі ролей	1	1	1	0	0	1
Не визначена	0	0/1	0	0	0	0

В такому вигляді задача не потребує використання теорії корисності для розрахунків оптимальної стратегії, так як рішення про вибір тієї чи іншої стратегії залежить виключно від наявності чи відсутності критеріїв, які всі відносяться до одного класу. Для розрахунку задачі з використанням теорії корисності та побудови саме прогнозу вибору моделі, яка буде максимально відповідати набору критеріїв отриманих під час проведення опиту припустимо що у нас є система, яку треба буде розробити або вдосконалити і в якій можуть бути присутні всі критерії з попередньої таблиці. Тоді для вибору оптимальної стратегії треба врахувати

додаткові критерії такі як кількість можливих користувачів, складність ієрархічної структури, плинність кадрів тощо. Список критеріїв буде відповідати переліку питань, визначених для проведення опитування і зазначених у додатку В.

Крім цього для даного дослідження буде трохи змінена множина альтернатив. Будуть виключені з розгляду моделі RBAC-A в силу специфічності даних моделей і досить маленького поширення. Також буде прибрано варіант, коли система не може бути визначена (так як для системи в якій немає ані користувачів, ані ролей, ані атрибутів неможливо визначити наведені вище критерії). Гібридна ABAC-RBAC буде розширена на розгляд з атрибутом ієрархії сутностей – HRBAC, атрибутом часу – TRBAC, та атрибутом місцезнаходження – SRBAC.

Таким чином множина альтернатив буде мати наступний вигляд:

– ACL;
– RBAC;
– ABAC;
– HRBAC;
– TRBAC;
– SRBAC.

Критерії вибору будуть співпадати з відповідями на запитання з опитування, які зазначені у додатку В. Крім цього у кожного запитання, а отже і у критерія вибору є ваговий коефіцієнт, який буде визначатися за пропорційним методом. Всі критерії можна розділити на три групи – вагомі для визначення моделі доступу, які будуть мати коефіцієнт три, звичайні – матимуть коефіцієнт два та досить мало вагомі з коефіцієнтом один. Сума всіх коефіцієнтів має дорівнювати одиниці.

На основі проаналізованих джерел інформації і спеціалізованої літератури в таблиці 3.4 наведено матрицю прийняття рішень з коефіцієнтами впливу кожного з критеріїв для вибору тієї чи іншої альтернативи. При цьому назви можливих моделей доступу – наші альтернативи розміщені у стовпчиках, а в строках ознаки, які впливають на вибір моделі доступу, які виступають у якості критеріїв. Також в

цю таблицю додано нормуючи множниками, розраховані згідно з формулою 3.3 та ваговими коефіцієнти, розраховані згідно з формулою 3.1 для кожного з критеріїв.

Таблиця 3.4 – Матриця рішень для конкретних методів

	Альтернативи							
Критерії	A_1	A_2	A_3	A_4	A_5	A_6	α	β
1	2	3	4	5	6	7	8	9
Користувачів менше 50	4	3	0	3	3	3	0.0625	0.5
Користувачів від 50 до 100	2	3	1	3	3	3	0.0666	0.5
Користувачів від 100 до 500	0	3	2	3	3	3	0.0714	0.5
Користувачів від 500 до 1000	0	2	3	2	2	2	0.0909	0.5
Користувачів більше 1000	0	1	4	1	1	1	0.125	0.5
Чітко структуровані робочі групи	1	2	0	2	2	2	0.1111	0.33
Немає чітко структурованих робочих груп	1	0	2	0	0	0	0.3333	0.33
Зміна обов’язків 1 раз на тиждень	0	0	2	0	0	0	0.5	0.33
Зміна обов’язків 1 раз на місяць	0	0	1	0	0	0	1	0.33
Зміна обов’язків 1 раз на рік	1	1	1	1	1	1	0.1666	0.33
Зміна обов’язків ніколи	2	1	0	1	1	1	0.1666	0.33
Сутностей баз даних десятки	2	1	0	1	1	1	0.1666	0.5
Сутностей баз даних сотні	0	2	1	2	2	2	0.1111	0.5
Сутностей баз даних тисячі	0	0	2	0	0	0	0.5	0.5
Існує необхідність налаштовування доступу до ресурсів не однотипно	1	1	3	1	1	1	0.125	0.33
Не існує необхідності налаштовування доступу до ресурсів не однотипно	1	1	0	1	1	1	0.2	0.33
Існує висока плинність кадрів	0	1	1	1	1	1	0.2	0.16
Не існує високої плинності кадрів	2	1	1	1	1	1	0.1428	0.16

Кінець таблиці 3.4

1	2	3	4	5	6	7	8	9
Має бути розподіл доступ за певним часом	0	1	2	1	3	1	0.125	0.33
Не має бути розподілу доступу за певним часом	1	1	0	1	0	1	0.25	0.33
Має бути розподіл по місцезнаходженню	0	1	2	1	1	3	0.125	0.33
Не має бути розподілу по місцезнаходженню	1	1	0	1	1	0	0.25	0.33
Має бути розподіл по ієрархічним зв’язкам	0	1	2	3	1	1	0.125	0.33
Не має бути розподілу по ієрархічним зв’язкам	1	1	0	0	1	1	0.25	0.33
Має бути поділ видимості різних даних однієї сутності	0	0	2	0	0	0	0.5	0.16
Не має бути поділу видимості різних даних однієї сутності	0	1	1	1	1	1	0.2	0.16

В таблиці наведено наступні скорочення:

- A_1 – альтернатива ACL;
- A_2 – альтернатива RBAC;
- A_3 – альтернатива ABAC;
- A_4 – альтернатива HRBAC;
- A_5 – альтернатива TRBAC;
- A_6 – альтернатива SRBAC;
- α – нормуючий множник;
- β – ваговий коефіцієнт.

Результат обчислення лінійної адитивної згортки з ваговими коефіцієнтами буде залежати від відповідей, які користувачі будуть надавати під час проходження опитування. Для більш об’єктивного оцінювання даних отриманих від користувачів, має існувати мінімальна кількість пройдених опитувань стосовно однієї й тієї самої системи.

4 ОПИС РОЗРОБЛЕНОЇ ПРОГРАМНОЇ СИСТЕМИ

4.1 Обґрунтування вибору типу програмного забезпечення

Для створення автоматизованої системи вибору оптимальної моделі авторизації відповідно до вимог, що випливають з постановки задачі, необхідні:

- серверна частина;
- клієнтська частина;
- доступ до бази даних для збереження результатів аналізу.

На сьогодні існує багато способів розробки програмного забезпечення. Відповідно до поставленої задачі було розглянуто декілька варіантів розроблення програмного продукту: вебдодаток, мобільний або десктопний додаток. Враховуючи такі критерії як апаратні вимоги для користувача, складність встановлення та обслуговування, розділення прав доступу до системи, збір даних для аналізу та вартість розробки[35], була проведена порівняльна характеристика, яка наведена у таблиці 4.1.

Таблиця 4.1 – Аналіз варіантів розробки програмного продукту

Критерій	Вебдодаток	Мобільний додаток	Десктопний додаток
1	2	3	4
Апаратні вимоги для користувача	Відсутні, крім сучасного веббраузера, оскільки обчислення виконуються сервером	Розробка під конкретну мобільну платформу; залежність від апаратного забезпечення пристрою	Розробка під конкретну ОС; обчислення потребуватимуть відповідної потужності апаратної частини користувача
Встановлення та обслуговування	Не вимагає	Вимагає	Вимагає; може бути певна складність
Розділення прав доступу до системи	Легке і звичне	Можливе	Складне

Кінець таблиці 4.1

1	2	3	4
Збір даних для аналізу	Не потребує завантаження додатку; дані завантажуються в будь-якому разі	Збереження даних користувача можливе після завантаження додатку	Збереження даних користувача можливе після завантаження додатку
Вартість розробки	Розробка дешевша, ніж розробка десктопного додатку, а також підходить для будь-якої цільової платформи	Потребує додаткових витрат для кожної мобільної платформи	Необхідні витрати на розробку під кожну цільову платформу

В результаті порівняльного аналізу було обрано розробку системи у вигляді вебдодатку. Вирішальними факторами вибору варіанту розробки додатку стали незалежність від платформи, відсутність встановлення для користувача, а також можливість доступу з будь-якого пристрою з підключенням до мережі Інтернет. Розробка вебдодатку також надає більш широкі можливості в забезпеченні спільного доступу для групи користувачів, що є важливим фактором для даної системи, так як аналіз результатів базується на групових відповідях.

4.2 Вибір технологій та мови програмування для розробки вебдодатку

Для реалізації прототипу системи було вирішено взяти за основу PHP-фреймворк Laravel 8. Це популярний інструмент для розробки вебдодатків. Фреймворк є легким у використанні, включає в себе готовий функціонал для стандартних завдань, зокрема аутентифікації та авторизації, захисту від несанкціонованого входу, тощо. Механізм захисту на доступ до ресурсів від неавторизованого користувача виступає технологія Middleware – «посередник»,

який при кожному запиті до сервера автоматично перевіряє всі необхідні дані користувача на доступ.

Основні переваги обраного фреймворку:

– оптимізований для створення професійних вебдодатків і готовий справлятися з корпоративними робочими навантаженнями;

– легке горизонтальне масштабування;

– потужний інструмент з відкритим кодом;

– Eloquent ORM обгортка для SQL запитів.

Для роботи з БД було обрано MySQL. Перевагами даної СУБД перед іншими є її популярність, простота та безкоштовність. Для реалізації інтерфейсу користувача в браузері використовуються стандартні інструменти HTML5, CSS3 та JavaScript.

Уся програмна логіка знаходиться на вебсервері, який забезпечує видачу запитів до бази даних, які передає на виконання SQL-серверу. Для того щоб програмний продукт, написаний на мові програмування PHP, працював у будь-якій операційній системі, потрібен віртуальний вебсервер Apache або Nginx, який отримує HTTP-запити від клієнтів, здебільшого це веббраузери, та видає їм відповіді у вигляді HTML-сторінки, зображення, файла, медіа-потока або інші дані. Вебсервером називається програмне забезпечення, яке виконує функції вебсервера, або це може бути комп’ютер, на якому встановлено це програмне забезпечення.

4.3 Структура бази даних

База даних складається з 8 таблиць: 6 таблиць для запису інформації та 2 таблиці сформовані фреймворком додатку. На рисунку 4.1 зображено ER-діаграму архітектури бази даних. Модель сутності-зв’язку (ER-модель) – модель даних, що дозволяє описувати концептуальні схеми предметної області. ER-модель

використовується при проектуванні бази даних. За допомогою можна виділити ключові сутності та визначити зв’язки, які можуть встановлюватися між ними.[36]

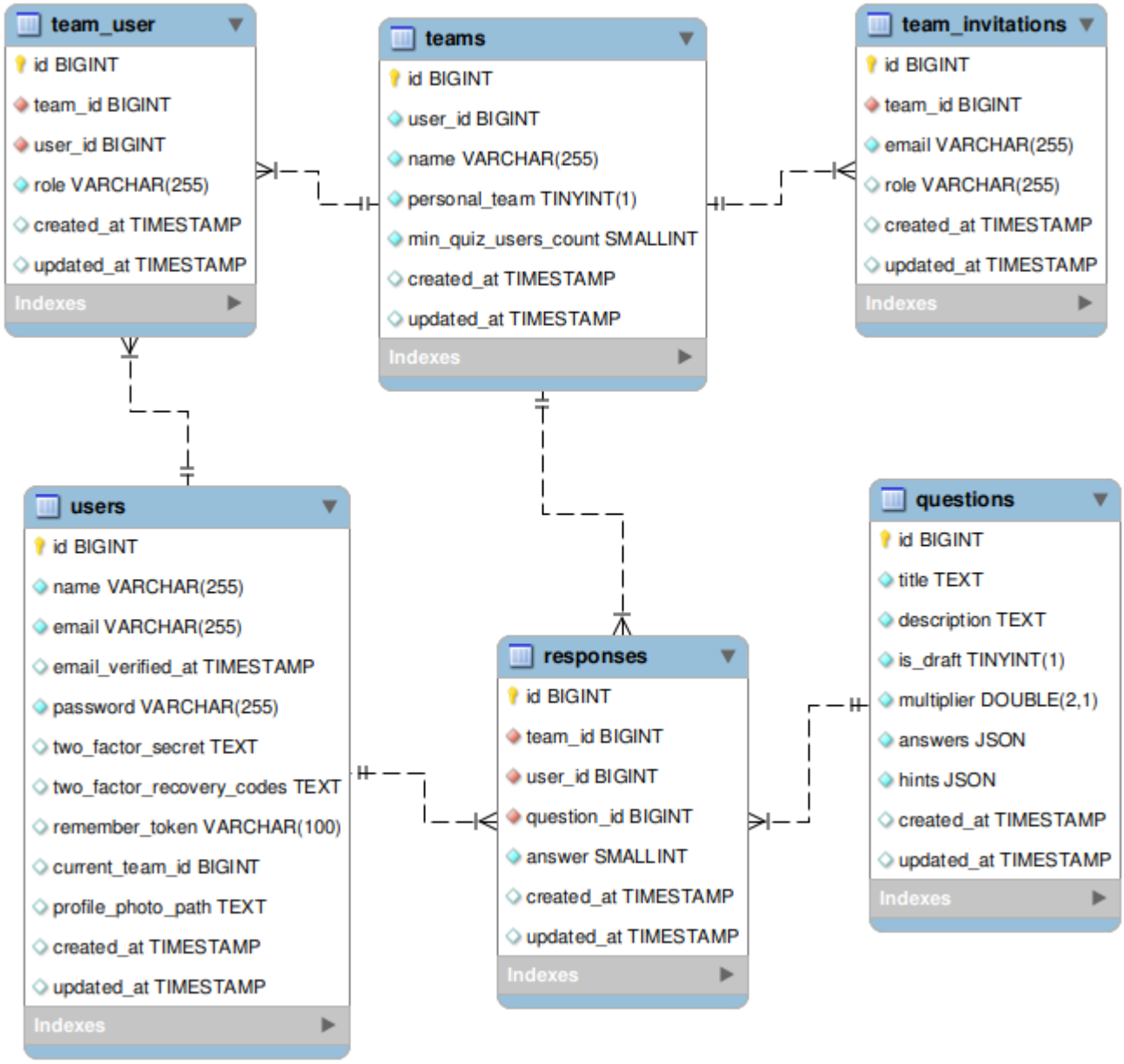

Рисунок 4.1 – Схема бази даних

Розглянемо детальніше кожну таблицю бази даних:

– в таблиці users зберігаються данні про користувачів системи: id, ім’я, електронна адреса, пароль тощо; ця інформація потрібна для аутентифікації та авторизації користувачів в системі AuthzChoice;

– таблиця teams призначена для зберігання інформації про системи для яких

проводиться опитування, щоб визначити яка модель авторизація краще підходить; основні поля цієї таблиці це id, назва системи, id власника системи (зовнішній ключ на id користувача з таблиці users) та мінімальна кількість експертів потрібних для підрахунку результатів;

– team_user – це таблиця для зберігання зв’язків багато до багатьох між користувачами експертами та системами для опитування;

– таблиця team_invitations призначена для збереження id системи (зовнішній ключ на id системи з таблиці teams) та електронної адреси, на яку було вислано запрошення до проходження опитування по цій системі;

– в таблиці questions зберігається інформація про питання, які використовуються для опитування експертів: id, назва, опис (не обов’язкове поле), ваговий коефіцієнт, варіанти відповідей в форматі json;

– таблиця responses призначена щоб зберігати відповіді користувачів за питаннями з таблиці questions; за цими відповідями потім проводиться аналіз яку саме систему авторізації обрати для системи.

4.4 UML проектування системи

Діаграма варіантів використання – в UML це діаграма, на яка відображає відношення в системі між акторами та прецедентами.[37]

Суть даної діаграми представити спроектовану систему як безліч сутностей або акторів, які за допомогою варіантів використання взаємодіють з системою. Будь-який варіант використання описує певний набір дій, який при діалозі з актором виконується системою.

Спроектована діаграма варіантів використання системи зображена на рисунку 4.2 і дозволяє чітко визначити, який саме функціонал та варіанти дій будуть надані користувачам, а також формалізувати функціональні вимоги до системи і

надає можливість узгодити з замовником отриману модель на перших стадіях проектування.

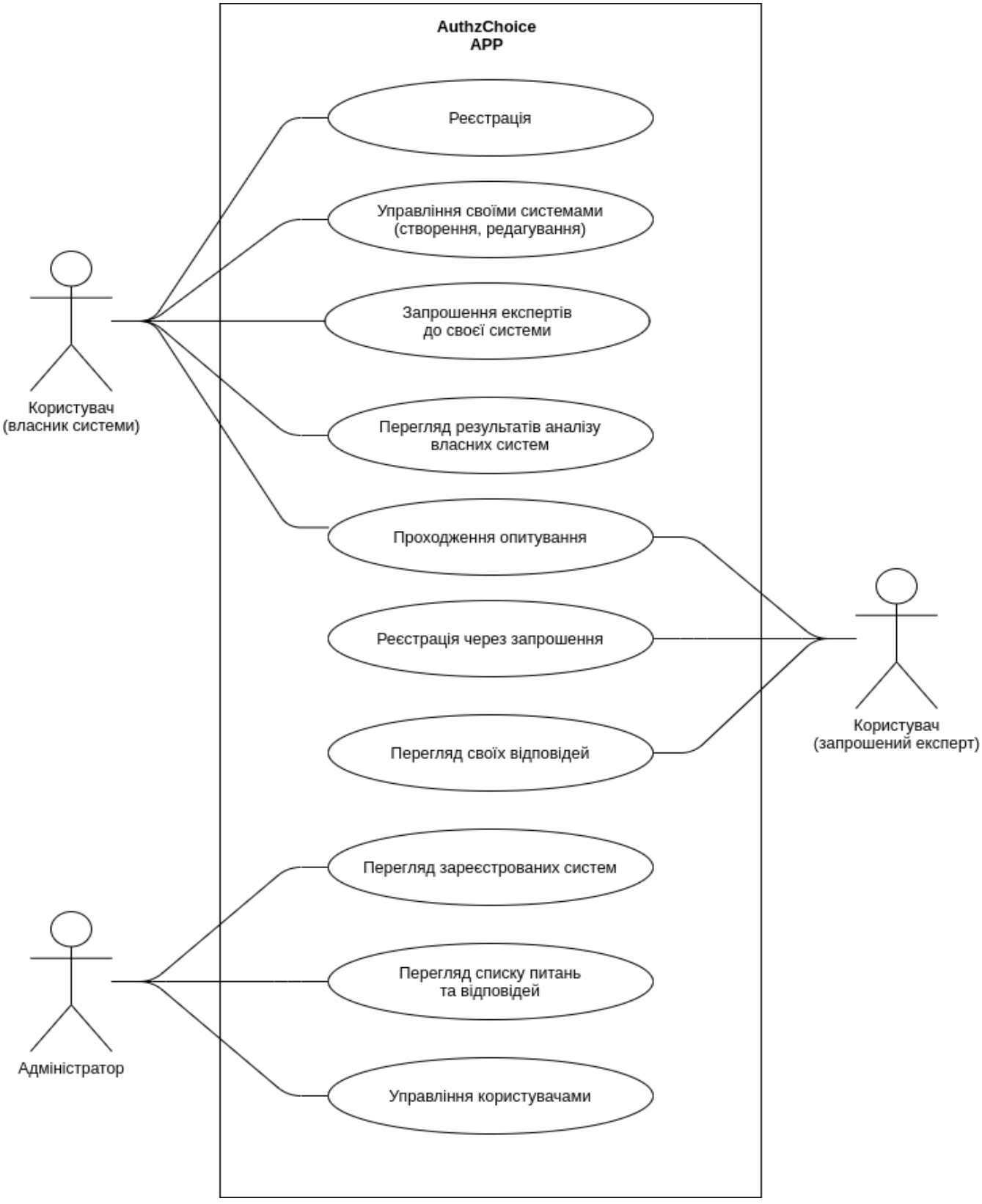

Рисунок 4.1 – UML діаграма варіантів використання

Система «AuthzChoice» має трьох акторів, які є користувачами цієї системи:

- користувач (власник системи);
- користувач (експерт);
- адміністратор.

Кожний з акторів має декілька прецедентів використання системи. Користувач (власник системи) може використовувати програму для таких цілей:

- реєстрація;
- управління своїми системами (створення, редагування);
- запрошення експертів пройти опитування;
- перегляд результатів аналізу власних систем;
- проходження опитування.

Користувач (експерт) може використовувати програму для таких цілей:

- реєстрація через запрошення;
- проходження опитування;
- перегляд своїх відповідей.

Адміністратор може використовувати систему для:

- перегляд зареєстрованих систем;
- перегляд списку питань та відповідей;
- управління користувачами.

4.5 Опис інтерфейсу системи

Створена система має інтуїтивно зрозумілий та простий вебінтерфейс. Вебінтерфейс – це сукупність вебсторінок, яка надає можливість користувачу взаємодіяти з вебсайтом або вебзастосунком через браузер. Коли користувач ще не є аутентифікованим, він потрапляє на сторінку логіну, яка зображена на рисунку 4.3. де повинен ввести свій логін (електронну адресу) та пароль доступу до системи.

Якщо користувач ще не має акаунту в системі AuthzChoice, то він має змогу зареєструватися, а на наступному кроці створити запис про свою інформаційну систему, яка далі має бути проаналізована.

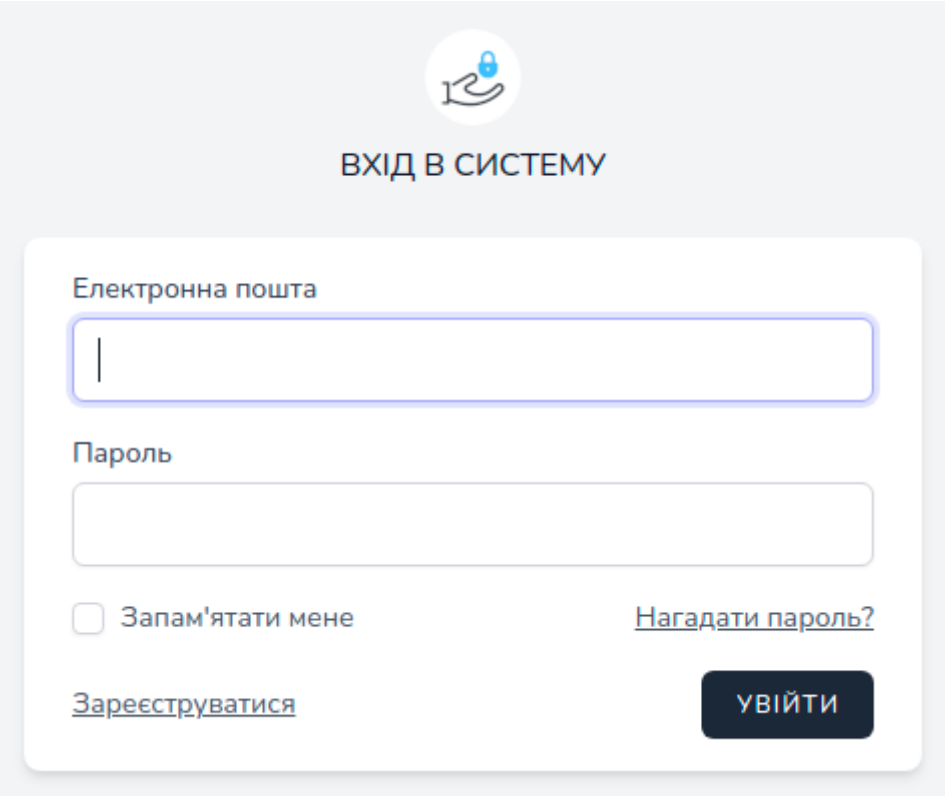

Рисунок 4.3 – Графічний інтерфейс системи. Авторизаційна форма входу

Якщо логін та пароль були введені правильно, користувач отримує доступ до системи. Завдяки меню, яке знаходиться зверху на кожній сторінці, можна перейти на інші сторінки вебдодатку, вийти з системи, а також змінити поточну систему, з якою відбувається робота.

Окрім цього на головній сторінці, яку зображено на рисунку 4.4, описані основні функціональні кроки, які треба зробити, щоб отримати результат за своєю системою. Першим кроком є створення та налаштування системи, де власник додає її з унікальною назвою та вказує кількість людей для опитування. На другому кроці відбувається процес запрошення експертів для опитування, а саме, власник системи додає електронні адреси і відправляються листи з реферальними посиланнями на участь в опитуванні. Після переходу за посиланням, користувач-експерт потрапляє

на сторінку реєстрації, а потім на сторінку опитування. Під час опитування кожен з експертів має відповісти на десять запитань стосовно майбутньої інформаційної системи. Цей третій крок має бути пройдено мінімальною встановленою кількістю опитуваних. На четвертому кроці власнику надаються висновки та рекомендація яку модель авторизації краще обрати для системи.

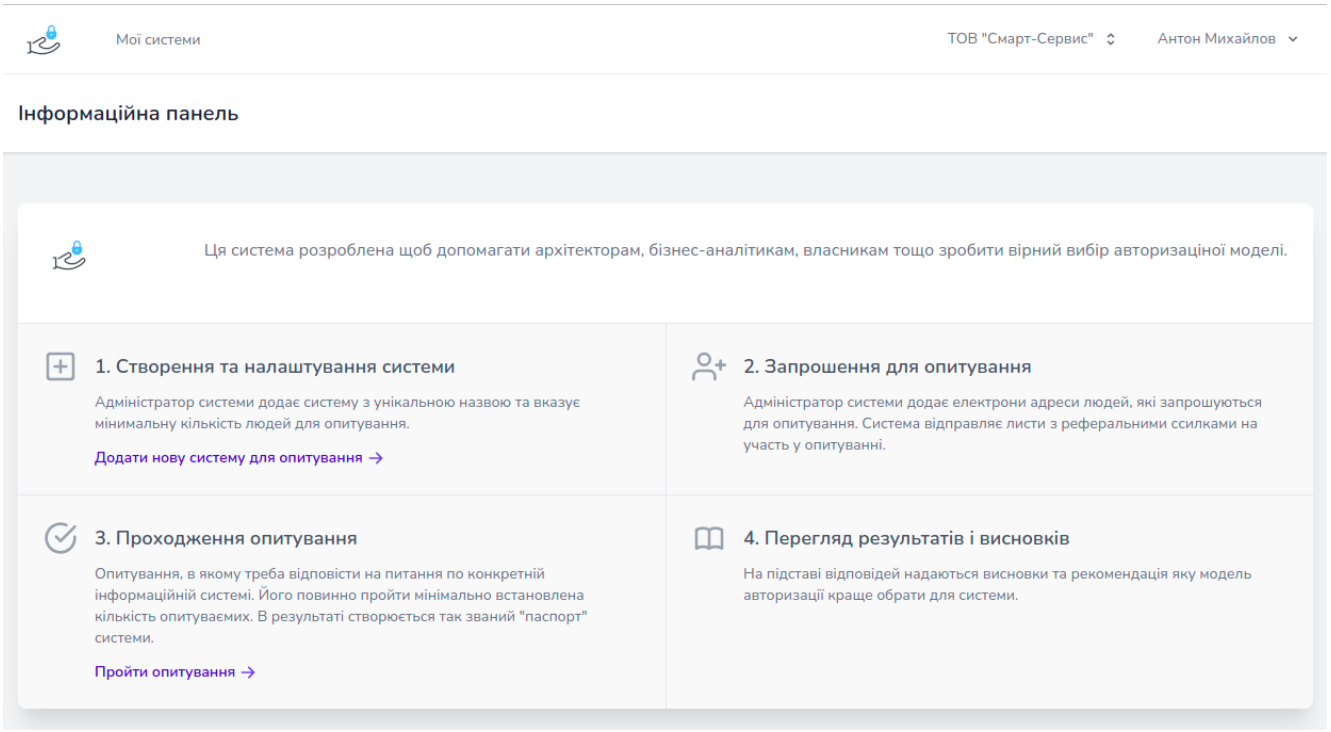

Рисунок 4.4 – Графічний інтерфейс системи. Головна сторінка користувача

Для перегляду повного списку систем, в яких користувач є власником або експертом, треба перейти на сторінку «Мої системи», яка зображена на рисунку 4.5. Ця сторінка відображає статус по кожній з систем, та надає можливість перейти до проходження опитування, налаштування параметрів опитування, перегляду результатів тощо, в залежності від того які права має певний користувач. Отже, якщо у користувача роль «експерт», то йому не доступна інформація про загальну кількість запрошених людей для проходження опитування, а також кількість людей, яка вже його пройшла для певної інформаційної системи. Проте якщо у користувача роль «власник», то для нього буде відображатися вся інформація по системі. Також «власник» системи має можливість переглянути поточний результат аналізу

системи або закрити систему для опитування, і отримати фінальний варіант рекомендацій.

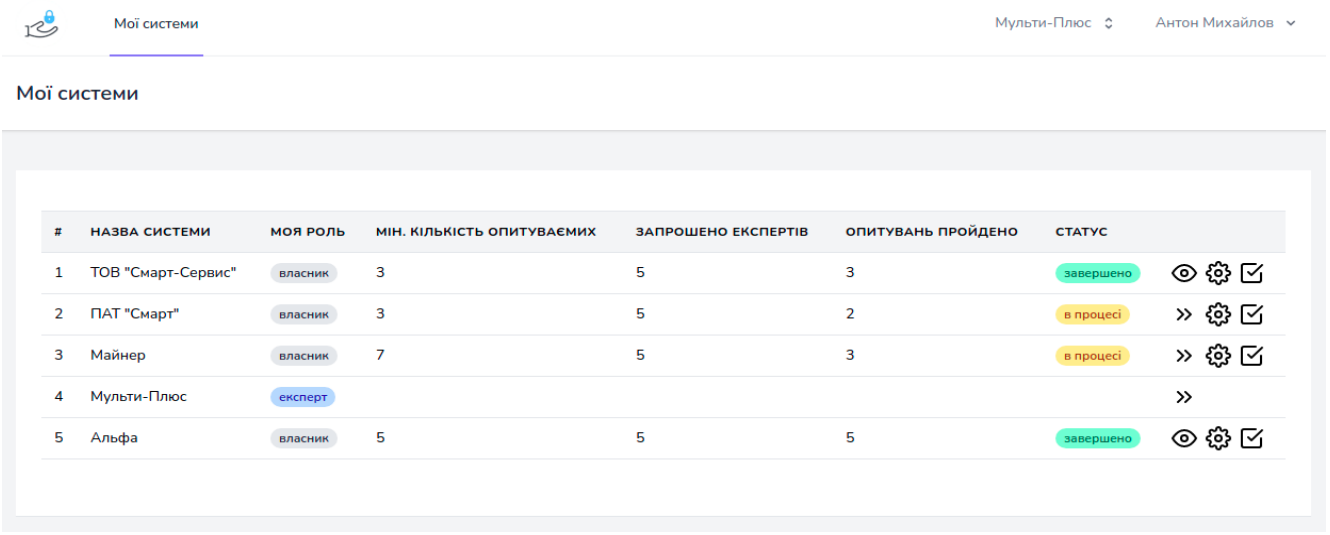

Рисунок 4.5 – Графічний інтерфейс системи. Сторінка «Мої системи»

На рисунку 4.6 зображено вигляд окремого запитання з опціями для відповіді з опитувальника. Деякі питання мають додатковий опис, з прикладом для більш чіткого розуміння користувачем формулювання.

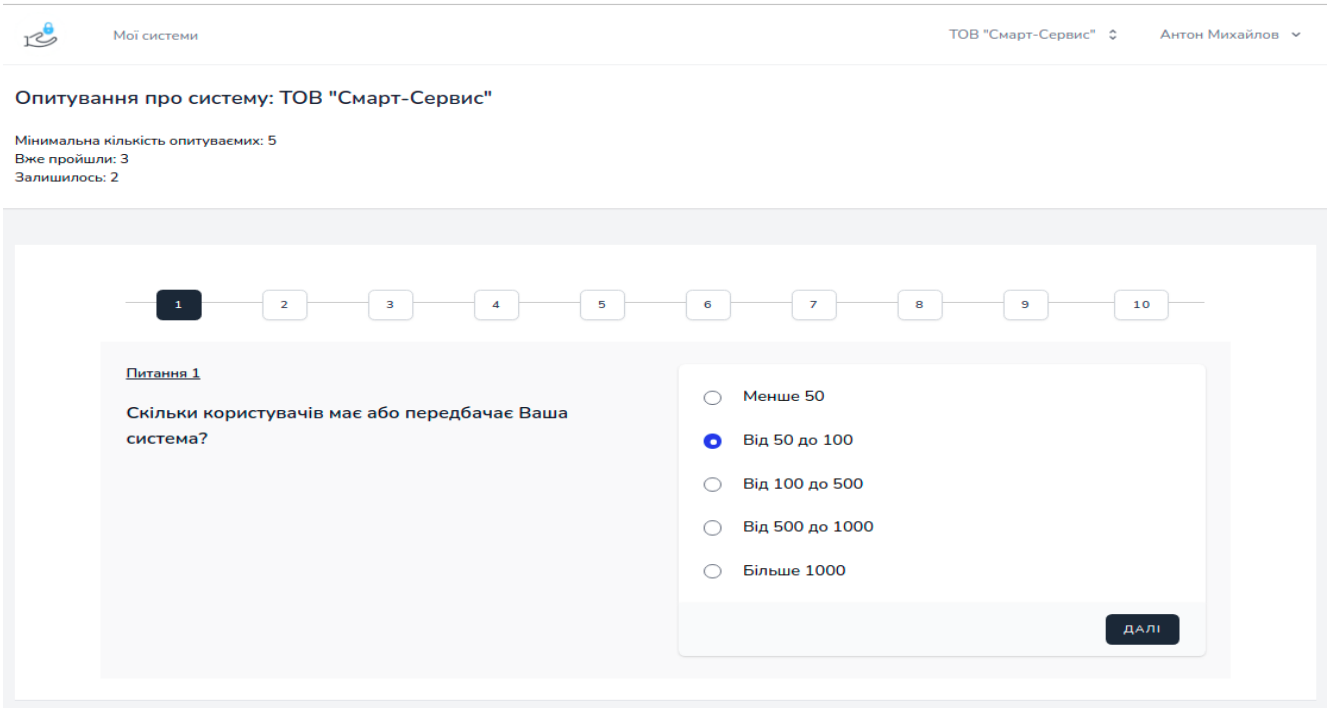

Рисунок 4.6 – Графічний інтерфейс системи. Сторінка опитування

Після проходження опитування та досягнення необхідної кількість відповідей експертів, яка визначена у налаштуваннях для кожної окремої інформаційної системи, власник цієї системи на сторінці «Результати», вигляд якої наведено на рисунку 4.7, може переглянути результати та дізнатися яка сама модель авторизації рекомендована для даної інформаційної системи.

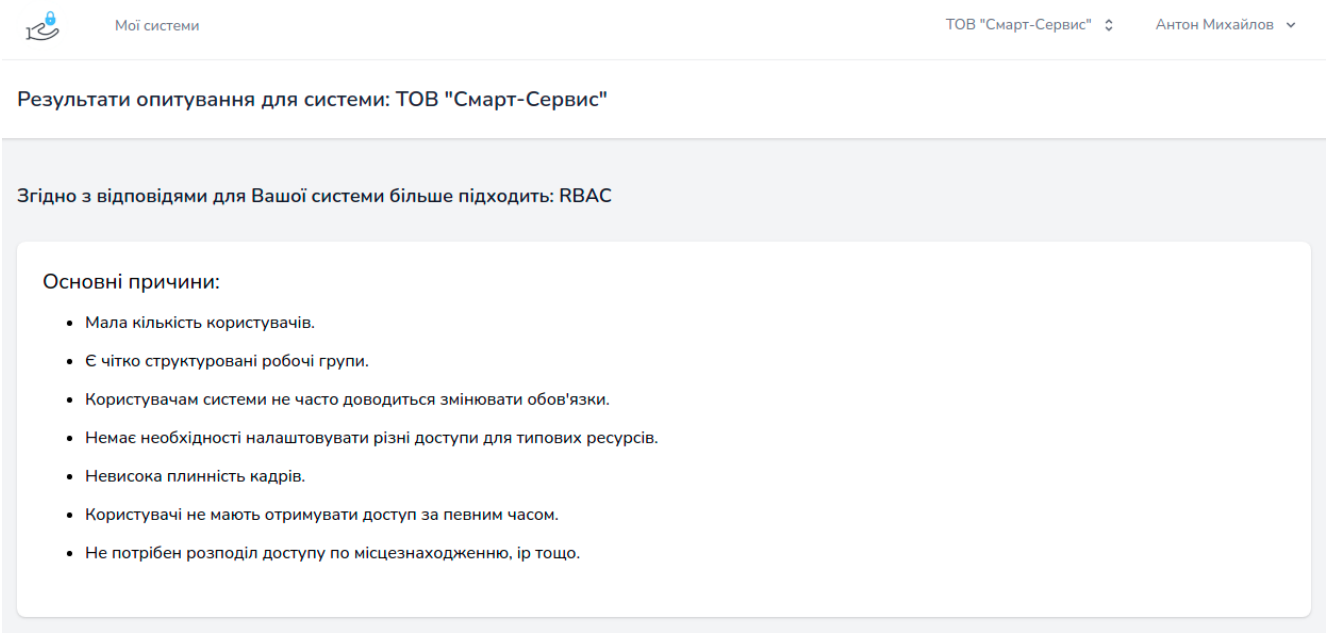

Рисунок 4.7 – Графічний інтерфейс системи. Сторінка результатів опитування

Після цього власник системи може закрити опитування, якщо вважає що кінцевий результат вже отримано і інформації, яка надана, цілком достатньо для початку проектування модуля безпеки майбутньої інформаційної системи. В іншому випадку, власник може додати нових експертів, збільшити мінімально-потрібну кількість опитувань і через якийсь час отримати більш точний результат.

4.5 Опис можливості використання отриманих результатів

Результат даної роботи може використовуватися як сервіс, який допомагає з вибором оптимальної моделі авторизації. Розроблений програмний продукт

автоматизує цей вибір шляхом збору даних про систему за допомогою опитування певної кількості експертів та аналізу цих даних, чим полегшує роботу бізнес-аналітиків, архітекторів тощо в період побудови архітектури безпеки для майбутньої інформаційної системи. Також програма стане в нагоді при глобальних змінах авторизаційного підходу для вже існуючих інформаційних систем.

Система досить добре масштабується. Алгоритм розрахунку оптимальної моделі авторизації легко розширювати новими моделями. Крім цього існуючий підхід до розрахунку моделі можна розширювати та уточнювати новими питаннями, що дасть ще більш точний аналіз та чіткий висновок для власників систем, які потребують інформацію щодо можливою оптимальної моделі авторизації.

ВИСНОВКИ

В ході кваліфікаційної роботи були проаналізовані основні моделі контролю доступу, зокрема ACL, RBAC, ABAC, HRBAC, TRBAC та SRBAC, а також виявлені їх переваги та недоліки.

Були розглянуті аналоги серед існуючих систем, які допомагають обрати та впровадити потрібну модель авторизації. Ці системи хоча і мають якісь функції та підходи, які допомагають визначитися з необхідною моделлю доступу, але все ж таки в основному орієнтовані на побудову самого контролю доступу до інформаційної системи, коли вибір вже зроблений, та є чіткий набір ролей, ресурсів, політик та дозволів. Саме тому є необхідність створення власної системи, яка саме буде допомагати архітекторам або бізнес-аналітикам зробити вірний вибір моделі авторизації, маючи певні вхідні дані, щоб мати як найменше помилок при налаштуванні авторизації для системи.

Для розробки браузерного додатку було обрано мову PHP 8 у поєднанні з Laravel framework та реляційну базу даних MySQL. Це дозволить використовувати програму як браузерний додаток досить легко та на великій кількості платформ. Сама програма буде цікава маленьким та середнім компаніям які починають проектувати нову інформаційну систему або значно розширювати функціонал існуючої. Безперечною перевагою системи є швидке розширення як на покриття нових моделей, так і на кількість питань-параметрів, які впливають на підсумковий результат з вибору певної моделі. Система може використовуватися без будь-якої попередньо технічної підготовки опитуваних.

Таким чином структура даного дослідження відповідає поставленим завданням. Перша частина присвячена аналізу предметної галузі та розгляду існуючих аналогів. У другій частині наведено теоретичний опис найбільш поширених моделей доступу. Третя частина присвячена розгляду алгоритмів, на базі яких побудовано дослідження, а саме розглянуто алгоритм лінійної адитивної згортки з ваговими коефіцієнтами. Четверта частина – це опис розробленої

програми, з обґрунтуванням вибору саме такого підходу, а також з наведенням скринів інтерфейсу програми.

За результатами кваліфікаційної роботи було створено презентацію (див. додаток Г) та було опубліковано тези дослідження у збірника Monografia pokonferencyjna», випуск «Science, research, development №29», Варшава, 30.03.2021 – 31.03.2021 на тему «Дослідження моделей та методів контролю доступу до інформаційної системи». Тези доповіді наведено в додатку Д.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Sandhu R. S., Samarati P. Access control: principle and practice // IEEE Commun. Mag. − 1994. No. 32(9). − P. 40-48.

2. Девянин П. Н. Модели безопасности компьютерных систем: учеб. пособие для вузов. − Москва: Академия, 2005. − 144 с.

3. Vincent C. Hu, etc. Guide to Attribute Based Access Control (ABAC) // Definition and Considerations. – National Institute of Standards and Technology, 2014.

4. Kuhn D. R., Coyne E. J., Weil T. R. Adding attributes to role-based access control // IEEE Computer. − 2010. − No. 43 (6). − P. 79-81.

5. Joshi J. A., Bertino E. A., Latif U., Ghafoor A. Generalized Temporal Role-Based Access Control Model // IEEE Trans. Knowledge and Data Engineering. – 2005. − No 17 (1). − P. 4-23.

6. Bertion E., Catania B., Damiani M.L. GEO-RBAC: A Spatially Aware RBAC // Proc. 10th ACM Symposium on Access Control Models and Technologies (SACMAT'05), Stockholm, Sweden, 2005. − P. 29-37.

7. William Fisher. Attribute Based Access Control. – National Institute of Standards and Technology. – 2015. – P. 22

8. Ilona Revenchuk, Sergiy Zagorodnyuk, Bohdan Sus, Oleksandr Bauzha. Information Security of Users Rights Assignment via the Software Solutions Based on LDAP // Problem of Infocommunications. Science and Technology – PIC S&T'2020, Kharkiv, Ukraine. – 6-9 October, 2020. – P. 5.

9. Кирій В. В., Мірошниченко Д. Д. Передумови організації інноваційної безпеки підприємства за умов розвитку креативної економіки // Матеріали І Міжнародної науково-практичної конференції «Сучасні стратегії економічного розвитку: наука, інновації та бізнес-освіта. (м. Харків, 3 листопада 2020 р.)» // За заг. ред. Т. В. Полозової [та ін.], Харків: ХНУРЕ. – 2020. – С. 144-147.

10. Дмитро Матвєєв, Дар'я Федоренко. Проблема захисту персональних даних в інтернеті // ΛΌΓΟΣ.ONLINE: International scientific e-journal 2019. № 4.

URL: https://ojs.ukrlogos.in.ua/index.php/2663-4139/article/view/530/545 (дата звернення: 24.04.2021).

11. Дмитро Матвєєв, Наталія Гордієнко. Захист великих даних та мінімізація ризиків втрати // ΛΌГОΣ.ONLINE: International scientific e-journal 2019. № 4. URL: https://ojs.ukrlogos.in.ua/index.php/2663-4139/article/view/522/531 (дата звернення: 24.04.2021).

12. Трощило О. С., Климентов В. В. Новый подход к защите данных в информационных системах // Сборник научных трудов МВД Украины. Киев, 2010. № 5. 7 с.

13. Качко О. Г., Балагура Д. С., Волощук О. Г., Головашич С. О. Створення та використання систем захисту інформації для банківських технологій // 8-а науково-практична конференція «Безпека інформації в інформаційно-телекомунікаційних системах»: Тези доповідей. Київ, 2005.

14. Аналіз статистики несанкціонованого доступу та джерел даних для систем аналітики поведінки користувачів та сутностей − URL: https://ojs.ukrlogos.in.ua/index.php/logos/article/view/9384/9081 (дата звернення: 23.03.2021).

15. Амелин Р. В. Информационная безопасность. URL: http://nto.immpu.sgu.ru/sites/default/files/3/__77037.pdf (дата звернення:15.04.2021).

16. Соколов А. В., Шаньгин В. Ф. Защита информации в распределенных корпоративных сетях и системах – М.:ДМК Пресс, 2002. – 187 с.

17. Information Security Basics – CIA/AIC Triad. URL: http://www.cybersalts.com/fundamental-principles-of-security-cia-aic/ (дата звернення: 28.03.2021).

18. RFC 4949 – Internet Security Glossary, Version 2. URL: https://tools.ietf.org/html/rfc4949/ (дата звернення: 15.04.2021).

19. Ferraiolo D., Kuhn D.R., Chandramouli R. Role-Based Access Control. // Artech House Inc. – 2007.

20. Daniel Servos, Sylvia L. Osborn. Current Research and Open Problems in Attribute-Based Access Control. − 2017.

21. Jin X., Krishnan R., Sandhu R. S. A unified attribute-based access control model covering DAC, MAC and RBAC // LNCS. − 2012. V. 7371. − P. 41-55.

22. Servos D., Osborn S. HGABAC: Towards a formal model of hierarchical attribute-based access control // Foundations and Practice of Security. Springer. − 2014. − P. 187-204.

23. Karp A., Haury H., and Davis M. From ABAC to ZBAC: The evolution of access control models // ISSA J. − 2010.

24. Yuan E., Tong J. Attributed based access control (ABAC) for web services // Proc. ICWS'2005. Washington, 2005. − P. 561-569.

25. Калимолдаев М. Н., Бияшев Р. Г., Рог О. А. Анализ методов атрибутивного разграничения доступа // Прикладная дискретная математика. – 2019.

26. How Privacy Killed RBAC – URL: https://medium.com/immuta-engineering/how-privacy-killed-rbac-328edf6e4be7 (дата звернення: 25 .03. 2021).

27. Кушлик-Дивульська О. І., Кушлик Б. Р. Основи теорії прийняття рішень. – К., 2014. – 94 с.

28. Шарапов О. Д., Дербенцев В. Д., Семьонов Д. Є. Економічна кібернетика: Навч. посібник. – К.: КНЕУ, 2004. – 231 с.

29. Ситник В. Ф. Системи підтримки прийняття рішень: Навч. посіб. – К.: КНЕУ, 2004. – 614 с.

30. Мазурова О. О. Методичні вказівки до лабораторних робіт та практичних занять з дисципліни «Теорія ігор та прийняття рішень» для студентів денної форми навчання спеціальностей 8.05010301 – «Програмне забезпечення систем», 8.05010302 – «Інженерія програмного забезпечення». Методичні вказівки – Харків: ХНУРЕ, 2016. – 32 с.

31. Орлов А. И. Теория принятия решений. – М.: Март, 2004. – 656 с.

32. Горюнов Ю. Ю., Горюнова Т. Ю., Дружинин Д. В. Теория и методы принятия решений: Учебное пособие. – Пенза: РГУИТП, 2010. – 50 с.

33. Кондрук Н. Е., Маляр М. М. Багатокритеріальна оптимізація лінійних систем: навч. посібник– Ужгород: РА "АУТДОР-ШАРК", 2019. – 76 с

34. Мазурова О. О. Разработка классификационной модели описания альтернатив для слабо структуризованных задач принятия решений. Вісник Національного Технічного університету «ХПІ». Тематичний збірник наукових праць «Нові рішення у сучасних технологіях» – Харків: НТУ «ХПІ». – 2001. – № 14. – С. 103-110.

35. Ревенчук І. А., Арсенюк Д. С. Подходы и механизмы реализации рекомендательных систем. Матер. 19-го Междунар. Молодежного форума «Радиоэлектроника и молодежь в XXI веке». – Харьков: ХНУРЭ. – 2015. – С. 171-172.

36. Peter P. Chen. Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned. URL: http://bit.csc.lsu.edu/~chen/pdf/Chen_Pioneers.pdf (дата звернення: 20.04.2021).

37. James Rumbaugh, Ivar Jacobson, Grady Booch. The unified modeling language reference manual. Addison Wesley Longman Inc. ISBN 0-201-30998-X. – 1999.