

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ  
Факультет кібербезпеки, комп'ютерної та програмної інженерії  
Кафедра комп'ютерних інформаційних технологій

ДОПУСТИТИ ДО ЗАХИСТУ  
Завідувач випускової кафедри  
\_\_\_\_\_ Аліна САВЧЕНКО  
« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р

# ДИПЛОМНИЙ ПРОЕКТ

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

ВИПУСКНИКА ОСВІТНЬОГО СПУПЕНЯ «БАКАЛАВР»

**Тема:** «Інформаційна система управління обліковими записами на базі IDM»

**Виконавець:** студент УС-411 СКРИПНІК Олександр Андрійович  
(студент, група, прізвище, ім'я, по батькові)

**Керівник:** д.т.н., професор ВОРОНІН Альберт Миколайович  
(науковий ступень, вчене звання, прізвище, ім'я, по батькові)

**Нормоконтролер:** ст. викл. Олександр ШЕВЧЕНКО  
(П.І.Б.) (підпис)

КИЇВ 2022

# НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет кібербезпеки, комп'ютерної та програмної інженерії

Кафедра комп'ютерних інформаційних технологій

Освітній ступінь **Бакалавр**

Галузь знань, спеціальність, спеціалізація: 12 “Інформаційні технології”, 122 “Комп'ютерні науки”, “Інформаційні управляючі системи та технології”

**ЗАТВЕРДЖУЮ**

Завідувач кафедри

Аліна САВЧЕНКО

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

## ЗАВДАННЯ

**на виконання дипломного проекту студента**

Скрипніка Олександра Андрійовича

(прізвище, ім'я, по батькові)

1. Тема проекту: «Інформаційна система управління обліковими записами з використанням IDM» затверджена наказом ректора № 454/ст. від 29.04.2022р.
2. Термін виконання роботи: з 09.05.2022 по 13.06.2022р.
3. Вихідні дані до роботи: реалізація впровадження IDM рішення за допомогою програмного продукту One Identity Manager .
4. Зміст пояснювальної записки (перелік питань, що підлягають розробці): вступ, аналітичний огляд і постановка задачі, огляд технологій та інструментів для розробки системи, розробка функціональності системи та інтерфейсу для користувачів, висновки.
5. Перелік обов'язкового графічного матеріалу: інформаційна діаграма бази даних системи, схема рівнів архітектури системи.

## КАЛЕНДАРНИЙ ПЛАН

	Етапи виконання дипломної роботи	Термін виконання етапів	Примітка
1	Аналіз літератури та джерел за темою дипломного проекту.	09.05.2022р. – 10.05.2022р.	
2	Розробка та затвердження плану дипломного проекту.	11.05.2022р.	
3	Проведення консультації з науковим керівником щодо створення першого розділ.	12.05.2022р.	
4	Аналітичний огляд і постановка задачі.	13.05.2022р. – 18.05.2022р.	
5	Порівняльний аналіз існуючих систем управління документами.	19.05.2022р. – 22.05.2022р.	
6	Огляд технологій для розробки системи.	23.06.2022р. – 27.05.2022р.	
7	Розробка компонентів системи.	28.05.2022р. – 04.06.2022р.	
8	Висновки та оформлення пояснювальної записки дипломного проекту.	05.06.2022р. – 08.06.2022р.	
9	Підписання необхідних документів у встановленому порядку.	09.06.2022р. – 10.06.2022р.	
10	Підготовка до захисту та попередній захист дипломного проекту на випусковій кафедрі дипломного проекту	11.06.2022р. – 13.06.2022р.	

7. Дата видачі завдання: 09.05.2022 р.

Керівник дипломного проекту

\_\_\_\_\_ (підпис керівника)

Альберт ВОРОНІН

(П.І.Б.)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис випускника)

Олександр СКРИПНИК

(П.І.Б.)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Інформаційна система управління обліковими записами з використанням IDM» 50 с., 15 рис., 14 літературних джерел.

**Об'єкт дослідження:** процес керування обліковими даними для оптимізації бізнес-процесів в компанії.

**Предмет дослідження:** система керування обліковими даними One Identity Manager.

**Мета роботи:** розробити інформаційну систему адміністрування обліковими записами з базою даних та веб інтерфейсом для користувачів.

**Методи дослідження,** технічні та програмні засоби: аналітика, розробка бази даних за допомогою MySQL, розробка кастомних скриптів для написання певних правил, розробка веб-інтерфейсу в системі One Identity Manager, обробка літературних джерел.

**Отримані результати та їх новизна:** розроблена система управління обліковими даними дозволяє оптимізувати бізнес-процеси в компанії та сприяє підвищенню продуктивності роботи та скороченню документообігу в компанії.

УПРАВЛІННЯ ОБЛІКОВИМИ ЗАПИСАМИ, БАЗИ ДАНИХ, РОБОЧИЙ ПРОЦЕС, SQL, C#, VB, ONE IDENTITY MANAGER.

## ЗМІСТ

ВСТУП .....	7
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД І ПОСТАНОВКА ЗАДАЧІ .....	10
1.1. Історія розвитку системи керування обліковими записами.....	10
1.2. Вирішення питань інформаційної безпеки.....	14
1.3. Огляд системи на One Identity Manager.....	16
1.4. Архітектура One Identity Manager.....	19
1.5. Підключення цільової системи до One Identity Manager.....	21
1.6. Постановка задачі.....	21
1.7. Висновки до розділу 1.....	23
РОЗДІЛ 2. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТЕМИ КЕРУВАННЯ ОБЛІКОВИМИ ЗАПИСАМИ.....	24
2.1. Технології для розробки бази даних.....	25
2.2. Організація роботи функціональної складової системи.....	26
2.3. Розробка користувацького інтерфейсу.....	28
2.4. Середовище розробки системи.....	30
2.5. Висновки до розділу 2.....	31
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КЕРУВАННЯ ОБЛІКОВИМИ ЗАПИСАМИ.....	32
3.1. Проектування бази даних .....	32
3.2. Підключення бази даних та створення синхропроекту .....	36
3.3. Налаштування конфігурації синхропроекту.....	40
3.3.1. Налаштування підключення до цільової системи.....	41
3.3.2. Конфігурація політики паролів та ERP-ресурс.....	43
3.4. Налаштування бізнес процесів.....	44
3.5. Розширені налаштування.....	46
3.6. Демонстрація роботи Web-середовища.....	50
3.7. Висновок до розділу 3.....	54

ВИСНОВКИ.....	55
СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ	
ДЖЕРЕЛ.....	57
ДОДАТКИ.....	59

## ВСТУП

На сьогоднішній день керування обліковими записами та правами користувачів є важливою складовою для оптимізації бізнес-процесів в компанії. Такий підхід дозволяє мінімізувати документообіг між відділами і значно скоротити витрати часу та ресурсів (як людських так і технічних). Пандемія Covid-19 дала дуже важливий поштовх до того щоб компанії почали автоматизувати свої робочі та організаційні процеси.

Наприклад, у невеликій ІТ компанії, яка не автоматизувала свої внутрішні процеси, дані про працівника, який щойно влаштувався на роботу, у відділі кадрів мають занести в 1С, після чого він направляється до ІТ відділу, де буде створено обліковий запис в каталозі Active Directory. Але доступ до папок, додатків, розсилок він може отримати лише після того, як зробить запит по електронній пошті до системного адміністратора, або у службі підтримки. В залежності від компанії для того щоб новий працівник мав певні можливості використання корпоративної інформації може знадобитися додаткова згода керівника. Це якщо говорити про отримання доступу. Але може відбуватися і те що працівника звільнять і в нього залишиться доступ до ресурсів компанії. Також розповсюджений випадок, коли працівник може відправиться у відпустку і в нього все ж залишається доступ до інформаційних ресурсів компанії або у працівника може змінюватися посада та посадові обов'язки, номер телефону, місце роботи (мається на увазі філіал компанії) – всі ці дані також повинні бути внесені до робочого середовища.

У більшості компаній з'являється задача по керуванню пароллями, якщо компанія має систему документообігу, або бухгалтерську систему, зовнішній портал. Кожна система підпорядковується та керується різними особами, пароль в кожній системі створюється окремо, а саме присвоюються персональні паролі, і тут постає головна проблематика сьогоднішнього часу, яка стосується інформаційної безпеки компанії через людський фактор.

Проблеми виникають і у користувачів, оскільки при створенні паролю для кожного середовища, користувачу необхідно запам'ятати велику кількість паролів. Користувачі починають їх зберігати на папері, через що виникає проблема несанкціонованого доступу до ресурсів. Складнощі виникають також у ситуації коли потрібно змінити доступ або пароль, а адміністратор не доступний. Крім того, при багаторазовому введенні неправильно пароля до відповідного веб-ресурсу компанії, може заблокуватися весь обліковий запис користувача.

Майже всі визначені вище проблеми можуть бути вирішені за допомогою використання Identity Management (IDM) шляхом автоматизації процесів, тим самим забезпечивши користувачам та адміністраторам системи зручність та ефективність.

Після того як з'являється обліковий запис в кадровій системі UAP – рішення автоматично створюється обліковий запис в підключених системах, також буде виданий доступ згідно до атрибутів користувача (посада, відділ). За допомогою User Administration and Provisioning (UAP) – рішення можна виключити людський фактор, що допоможе підвищити рівень інформаційної безпеки компанії. Система автоматично робить перевірку атрибутів відповідно до правил, в яких прописано рівень доступу користувача до ресурсів. Дане рішення дає змогу перевірити правильність створення облікового запису. Зміни відбуваються в одній системі і відображаються у всіх системах, до яких користувач має доступ. Коли, наприклад, у користувача змінюється посада, а відповідно до посади змінюються посадові обов'язки, то даний користувач матиме не лише зміни, але може навіть розширити автоматично доступ в веб-середовищах компанії. Зміна пароля в Active Directory (AD), призводить до зміни його у всіх системах, тож користувач має один пароль на всі системи, але безпека системи компанії залишається незмінною, оскільки кожен користувач підпадає під певні правила тобто обмеження і не може використовувати систему більше ніж у нього на це є



права, до того ж можна робити аналіз того що відбувається в тій чи іншій системі.

Також рішення від IDM дає можливість вирішувати задачі запиту, підтвердження, атестації та аудиту доступу до додатків, даних та сервісів, а також надає бізнес-аналітику. Прикладом може бути ситуація, якщо користувачу потрібно мати доступ на використання на корпоративному ПК певного програмного продукту, а система яка є не автоматизованою та не має впровадженого IDM. Потрібно знову робити багато паперових запитів з проханням використання даного програмного продукту, або навіть на отримання нової техніки того ж корпоративного ПК може займати кілька днів. IDM рішення дає можливість на відповідному веб-порталі зробити запит, і запуститься процес, який при необхідності зробить запит на підтвердження у керівництва після його отримання автоматично зробить потрібні зміни.

Також дане рішення підходить для відділу безпеки, оскільки можна за аналітикою доступу користувача, бачити до яких ресурсів користувач робив запити та що саме змінював в цих середовищах, до яких звертався.

Підсумовуючи вищезазначене, можна зазначити, що на сьогодні є досить важливим процес керування обліковими даними для оптимізації бізнес-процесів в компанії. Тому актуальною є розробка системи адміністрування облікових записів з базою даних та веб інтерфейсом для користувачі

## РОЗДІЛ 1

### АНАЛІТИЧНИЙ ОГЛЯД І ПОСТАНОВКА ЗАДАЧІ

На даний час більшість підприємств намагаються автоматизувати свої процеси в середині компанії, тим самим зменшити документообіг та навантаження на деякі відділи та скоротити витрати на організацію роботи в цих самих відділах. В більшості компаній спостерігається складність з організацією процесу створення та керуванням обліковими записами не лише своїх працівників, а й партнерів. Керування обліковими даними – це процес, за допомогою якого, можна як надавати користувачу доступ до того чи іншого середовища, так і обмежити його або навіть позбавити користувача цього доступу.

#### 1.1. Історія розвитку системи керування обліковими записами

Identity and Access Management (IAM) - це широкомасштабний комплекс, який в себе включає великий набір технологій та програмних продуктів, які вирішують задачі відносно керуванню життєвим циклом облікових даних та керуванню доступом до різноманітних ресурсів компанії [1]. На сьогоднішній день можна виокремити як мінімум дві області в середині IAM:

- User Administration and Provisioning (UAP);
- Identity and Access governance (IAG).

Кафедра КІТ				НАУ 22 22 37 000 ПЗ				
Виконав	Скрипнік О.А.			Аналітичний огляд і постановка задачі	Літера		Аркуш	Арку шів
Керівник	Воронін А.М.				ДП		10	14
Консульт.					УС-411 122			
Н-контроль	Шевченко О.П.							

Згідно до сучасних вимог, що трактує ринок з надання даних рішень, то сьогоднішні IAM рішення повинні надавати функціональність в обох областях [2].

LDAP – це протокол прикладного рівня для доступу до служб каталогу X.500, розроблений IETF, як полегшений варіант розробленого ІТU-Т протокола DAP [3].

Взагалі назва класу ІdМ рішення це скорочено від «Identity Management», «керування обліковими записами». «Керування обліковими даними» - це масштабний комплекс підходів, практик, технологій та спеціальних програмних засобів, для керування обліковими даними користувачів, системами контролю та керування доступом (СКУД) з ціллю підвищення безпеки та продуктивності інформаційних систем та зниженню затрат, головною задачею це є оптимізація часу та скорочення повторюючих задач.

З кожним днем даний інфраструктурний ландшафт стає більш складним та традиційне керування в форматі ручного управління групами доступу та обліковими записами стає все більш не зручним бо кількість користувачів в компанії становиться більше, також вести рольові моделі на папері та профілі користувачів вже є не зручним та відсталим. І на сьогодні все це вже замінює автоматична система ведення обліку автентифікації користувачів.

Взагалі введення ІdМ- рішення це не лише процес розгортання вже готового рішення це певний процес створення великої налагодженої системи управління доступом з зрозумілим з усіма учасникам процесу цілями та завданнями, обліковими записами та правами користувачів, що включає перелік процесів і процедур, фізичних, технічних та адміністративних заходів, а також саме ІdМ-рішення або ІGA-платформу.

Якщо повертатися до того що під ІdМ ми повинні розуміти цілу систему, саме ІAM рішення то типові задачі які може вирішувати ІAM система то це задачі які входять до числа UAP.

Сьогоднішній ринок рішень потребує щоб система була як можна більш автоматизованою, щоб скоротити витрати на персонал який буде обслуговувати цю ж систему. Зараз більшість великих компаній включають в себе безліч відділів які повинні мати можливість між собою, якимось обмінюватися інформацією і саме за допомогою використання UAP рішень, які в ходять до числа IAM. До того ж ми не повинні забувати про безпеку, і саме рішення UAP були спочатку націлені не лише на вирішення рутинних справ IT відділів по адмініструванню користувачів та ресурсів. Такі рішення повинні були контролювати повний доступ користувачів до системи компанії. UAP система могла автоматично згідно запрограмованого алгоритму надавати користувачі доступ до тих чи інших ресурсів програмного середовища, та в той же час могла заблокувати доступ користувача до відповідних ресурсів. Також важливо було надати користувачі можливість самостійно робити запит доступу до ресурсів (додатки, данні, сервіси), а керівникам або відповідальним за той чи інший ресурс надавати доступ за певним запитом в автоматизованому режимі, тобто одним натиском клавіши, що робить систему більш автоматизованою та зменшення документообігу в компанії, а тим самим зменшення затрат коштів та часу.

IAG рішення – це повний комплекс, який ми сьогодні бачимо в сучасних IAM системах. Identity and Access governance вирішує більшість потреб користувача, а саме система вирішує такі задачі, як [4]:

- задача запиту;
- задача підтвердження;
- атестація;
- аудит доступу (додатків, даним, сервісам);
- контроль та надання бізнес-аналітики процесів створення облікових записів;
- керування обліковими записами.

Постає питання в чому різниця від того, що вже було на ринку даних рішень?

Різниця в тому, що в UAP рішенні права в системі прив'язувалися напряму до користувачів, тобто до їх облікових записів, IAG рішення базуються на ролях, які закладені в систему та керування даної системи надано не IT-відділу, а бізнес-користувачам[5].

IAG рішення пропонують повністю автоматизувати основні процеси, а саме об'єднати всі відділи між собою для кращої взаємодії та полегшення роботи. IAG рішення пропонує автоматизацію процесів за допомогою веб-порталу, де можна оформити відповідний запит до ресурсу, користувачі не будемо навіть бачити як запусниться процес виконання він несе назву «невидимого», який за необхідністю зробить відповідний запит підтвердження керівника і після погодження керівництва зробить змінив системі [6]. Далі розглянемо це більш детально на прикладі. Нижче наведено, схему роботи системи IAG(рис. 1.3).

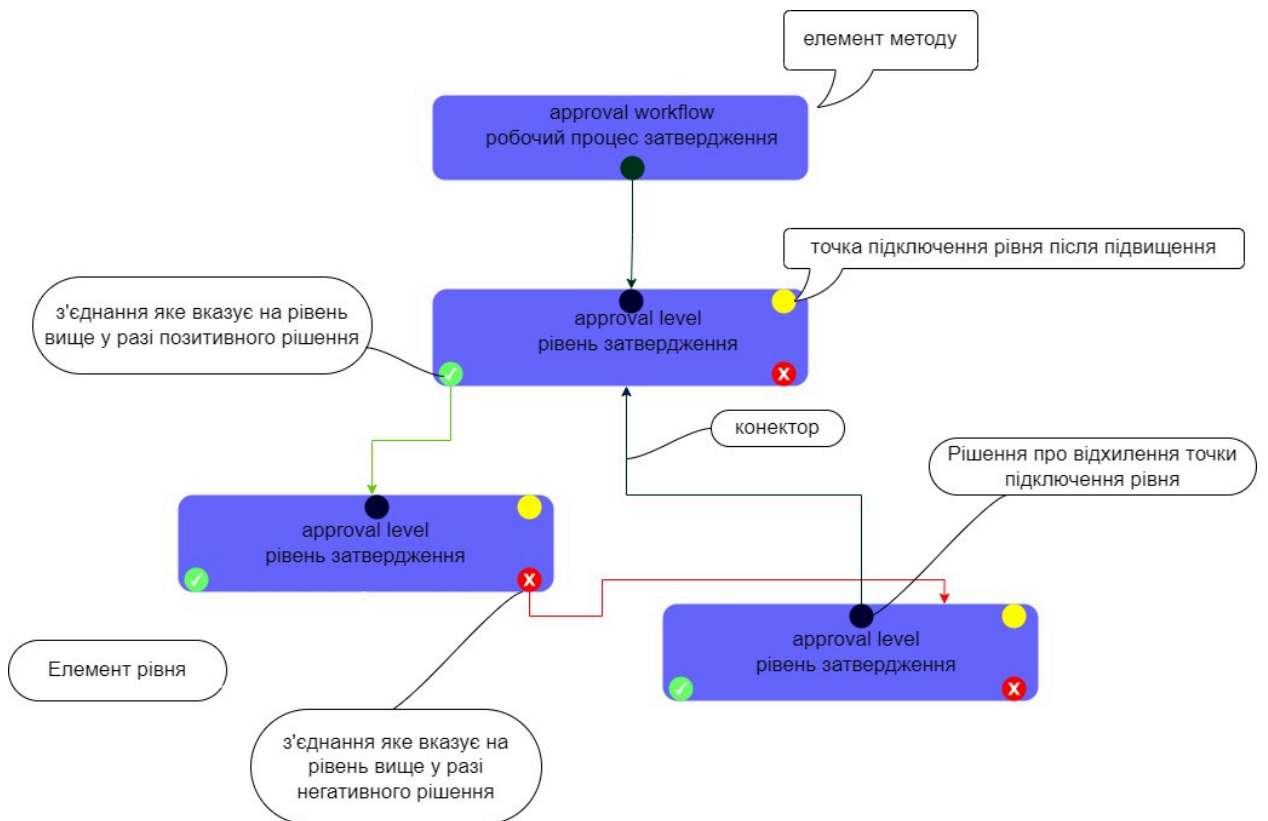


Рис. 1.1. Схематичне представлення роботи системи IAG

Кожна система має рівні, які вміщують в собі певний набір правил. Правила відповідають певним етапам розгляду системи. Маємо певний запит, який надходить від користувача, запит знаходиться на самому початку. Подальші дії базуються на політиці та правилах компанії. Далі початковий запит надходить на другий рівень, який має умови виконання. Якщо користувач не має доступу до того чи іншого ресурсу його можуть відхилити, а якщо він буде відповідати правилам, закладеним в системі, то він перейде на інший рівень і так далі до кінця. Якщо користувач зробив запит на зміни у системі, то при проходженні усього циклу до кінця (якщо буде узгоджено на різних рівнях), то зміни будуть відображені у системі.

## **1.2. Вирішення питань інформаційної безпеки**

IAG-система є досить зручною для керівників та працівників компанії, які відповідають за інформаційну безпеку. Є можливість бачити до яких систем у користувача є доступ, а також керувати даним доступом. Чому це важливо?

Питання безпеки є особливо важливим в наш час. Отримання несанкціонованого доступу до системи компанії є досить складною справою, особливо коли компанія регулярно виконує аудит інформаційної безпеки компанії. Існує багато пристроїв та програмного забезпечення, яке спроможне при правильному налаштуванні забезпечити цілісність системи. Широко застосовується соціальна інженерія, яка іноді навіть не потребує зловмисника використовувати якісь технічні засоби для отримання інформації.

Для проведення атаки за допомогою соціальної інженерії можна досить легко дізнатися про працівників компанії на яку націлена атака. Для цього лише потрібно знати особу яка працює в цій компанії, але це досить просто зробити за допомогою певного програмного забезпечення наприклад «theHarvester».

theHarvester – це інструмент, призначений для використання на ранніх етапах тесту на проникнення. Він використовується для збору інформації з



```
File Actions Edit View Help
[*] Target: microsoft
    Searching 100 results.
[*] Searching LinkedIn.

[*] LinkedIn Users found: 94
-----
Aaron Dinnage - Modern Work Specialist
Adam Harmetz
Adam Saxton - Principal Program Manager
Akarsh Shetty - Planning Leader
Akosua Boadi-Agyemang
Al Diaz
Alex Simons
Alvin Morales - Program Manager
Alysa Taylor
Amit Fulay
Andrey Lushnikov - Software Engineer
Ankita Mishra - Program Manager 2
Arun Ulag - Corporate Vice President
Ashish Jaiman - Director of Product Management
Billy Anders
Brad Smith - President
Brian Jones
Charles Lamanna - Corporate Vice President
Charlotte Yarkoni - Corporate Vice President
Chelsea Lee - Microsoft Technical Trainer
Chris L.
Chris Pratley - Corporate Vice President
Christiaan Brinkhoff - Principal Program Manager
Clay Wesener - Principal Group Program Manager
Corey Sanders
Daniel Klepner
David Carmona
```

Рис. 1.2.2 (результати виконання)

Але за допомогою IdM у зломисника можуть виникнути проблеми навіть на етапі соціальної інженерії. Тому що користувач може навіть не мати потрібного доступу до тої інформації яку прагне отримати зломисник, а також система IdM може аналізувати доступу до тих чи інших ресурсів компанії.

### 1.3. Огляд системи на One Identity Manager

Продукт One Identity Manager не є новим, а існує та стабільно розвивається на ринку вже понад 15 років. За цей час він пропонувався під різними назвами залежно від вендора-власника. Наприкінці 2016 року з'явилася нова самостійна компанія One Identity, а її основний продукт отримав теперішню назву – One Identity Manager. Саме під цим брендом та назвою європейському ринку і пропонується система управління ідентифікаційними даними та доступом.



One Identity Manager є рішенням для централізованого управління правами доступу співробітників до різних інформаційних ресурсів компанії. З його допомогою більшість завдань з управління обліковими записами та правами доступу користувачів автоматизується, що дозволяє знизити простоту у роботі співробітників та завантаження системних адміністраторів. Як і будь-яка система IdM, One Identity Manager пропонує багатий набір функцій контролю прав доступу співробітників, відповідності вимогам різних стандартів і політик безпеки, автоматизації та уніфікації самих процесів контролю[8]. Це дозволяє службам інформаційної безпеки забезпечувати більш високий рівень контролю за розмежуванням доступу до інформаційних ресурсів та, як наслідок, підвищити загальний рівень інформаційної безпеки в компанії. Продукт представлений повністю як програмне рішення та поставляється з великою кількістю наперед визначених робочих процесів. Широкий набір функцій пропонується із коробки. Портал самообслуговування в цій IdM-системі реалізований за принципом інтернет-магазину, логіка функціонування якого знайома здебільшого користувачів[9].

За допомогою One Identity Manager, можна здійснювати керування ідентифікаційними даними контролю доступу до додатків, платформ, систем даним у масштабах всього підприємства згідно моделі RBAC (Role Based Access Control).

Суттю Role Based Access Control є створення ролей, які повторюють бізнес ролі в компанії та присвоювання їх користувачам. Згідно до створених ролей буде відбуватися аналіз можливостей виконання користувачем тієї чи іншої дії.

Якщо бізнес-правила одномірні і всі дії можна розбити за ролями (бухгалтер, менеджер, адміністратор тощо), такого підходу буде достатньо. Тоді одному бізнес-правилу буде відповідати одна роль. Але в великих компаніях бізнес-правила є все більш складними і багатовимірними. Це призводить до того, що одного вже атрибуту замало, тоді додаються інші атрибути (місто, країна, філіал, власник, тощо).

Щоб впоратися з цією складністю, необхідно створювати додаткові ролі, число яких дорівнює кількості різних комбінацій всіх атрибутів.

Бізнес-правила, в яких використовуються атрибути, значення яких заздалегідь не відомі та обчислюються у процесі роботи, взагалі неможливо висловити за допомогою рольової моделі.

Існують також бізнес-правила, які обмежують доступ не до дій, а даних. Такі бізнес-правила також неможливо висловити за допомогою рольової моделі.

Щоб реалізувати підтримку таких бізнес-правил, доведеться використовувати інші інструменти, що лише здорожує та ускладнює впровадження та супровід системи контролю доступу. Виходить, що тількино бізнес-правила стають багатовимірними або вимагають контролю даних, рольова модель не тільки не вирішує поточні проблеми контролю доступу, а й створює нові, для цього і потрібне використання програмного продукту One Identity Manager.

Крім того, частково підтримується атрибутна модель доступу. У продукті реалізовано вирішення таких завдань, як управління обліковими записами, правами доступу користувачів до ресурсів інформаційних систем, управління запитом на нові облікові записи та права доступу, контроль за обліковими записами та правами доступу, контроль відповідності правилам та політикам (нормо контролю), планова атестація доступу. Сертифікація облікових записів, контроль поділу повноважень і т. д. При цьому більшість цих завдань уніфіковані (або автоматизовані), що дозволяє знизити навантаження на ІТ-персонал та скоротити операційні витрати.

Основні функціональні можливості One Identity Manager це є інтеграція з довіреними джерелами даних (інформаційні системи, з яких в IdM-систему завантажуються дані про внутрішніх та зовнішніх співробітників компанії, оргштатну структуру та ін.) та цільовими системами; при цьому як довірена система може бути використана будь-яка підключена інформаційна система. Також є можливість в автоматизації процесів, пов'язаних із створенням,

видаленням та зміною облікових записів співробітників (прийом на роботу, кадровий переклад, звільнення, довгострокову відпустку), а також наданням їм доступу до різних ресурсів відповідно до заданої рольової моделі. Якщо говорити про користувача, то є наявність зручного та сучасного порталу самообслуговування, реалізованого за принципом інтернет-магазину, права доступу можуть надаватися автоматично на основі ролей та оргштатної структури, а також через заявки на порталі самообслуговування. Безпечковий функціонал надання доступу тому чи іншому користувачу існує наявність вбудованого механізму розрахунку та оцінки ризиків, пов'язаних із наявними у співробітника правами доступу, також можливість проведення регулярної атестації прав доступу (за розкладом чи за подіями) та огляд інформації по співробітнику «на 360 градусів», що надає повне відображення всіх наявних у системі даних щодо нього та наданих йому прав доступу.

#### **1.4. Архітектура One Identity Manager**

Архітектура One Identity Manager показана на рис. 1.4.

По своїй структурі система One Identity Manager має в своєму робочому арсеналі всі стандартні компоненти, які присутні в продуктах даного класу, а саме:

- База даних;
- Опрацьовувач черги заданій та їх виконання в під'єднаних системах (через конектори та інтерфейси);
- Веб-сервер (Сервер додатків);
- Інтерфейс користувача та адміністратора.

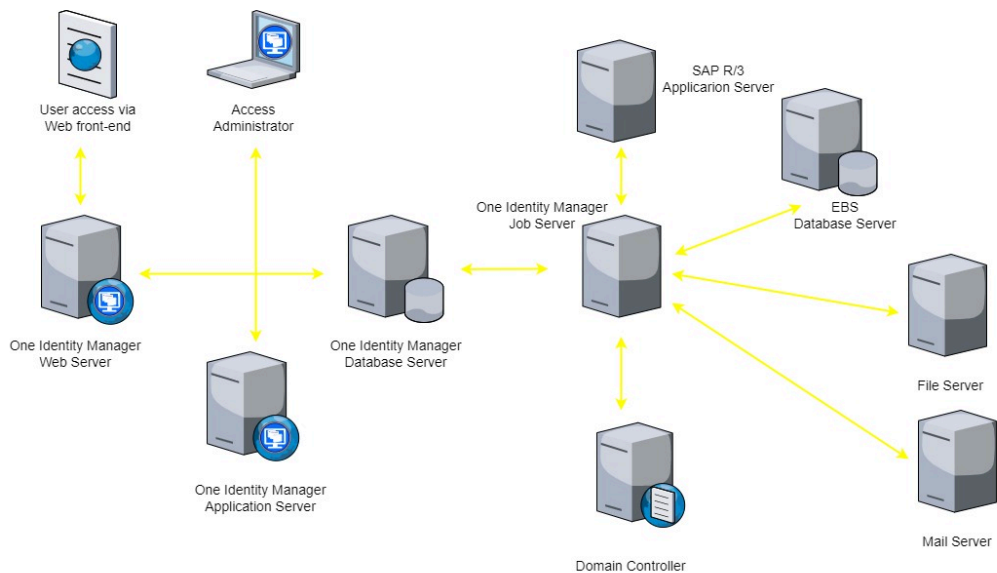


Рис. 1.4. Архітектура рішення One Identity Manager

Тобто це класична «трьох ланка». При цьому залежно від вимог, обмін даними між веб-сервером і базою даних може здійснюватися як безпосередньо, так і через сервер додатків. Нижче наведено опис компонентів One Identity Manager у термінології, яку використовує вендор:

- База даних (Database) є центральним компонентом системи. Можна виділити два типи даних, що зберігаються в базі даних рішення: дані користувача та метадані. Дані користувача містять інформацію про співробітників, облікові записи, групи, пов'язані з ними дані атестації, ресертифікації доступу і т. д. До складу метаданих входить інформація для роботи самої системи, а саме: повна конфігурація системи, параметри управління інтерфейсом, правила форматування даних, шаблони, скрипти, тощо. Буд;
- Служба сервера (Server Service) - компонент, який відповідає за синхронізацію інформації між базою даних та будь-якими підключеними цільовими системами;
- Сервер програм (Application server) - цей компонент містить бізнес-логіку системи. Клієнти підключаються безпосередньо до нього, що забезпечує безпечне з'єднання та роботу з базою даних;

- Веб-сервер (Web server) - використовується для надання користувачам web-інтерфейсу для роботи із системою (портал самообслуговування), а також для роботи сервера програм;
- Інтерфейси користувача та адміністратора (Front-Ends) - у складі системи присутня безліч інтерфейсів, кожен призначений для виконання певного кола завдань[10].

### **1.5. Підключення цільової системи до One Identity Manager**

У рішенні One Identity Manager є широкий набір готових конекторів до найбільш поширених типів цільових систем. Крім того, вендор надає графічне середовище (фреймворк) для підготовки конекторів - програма Synchronization Editor, яка дозволяє створювати конектори та підключати до IdM-рішення нові цільові системи, не вимагаючи написання коду.

Окремо варто відзначити, що система One Identity Manager має конектор для підключення системи SAP. З цією системою реалізована глибока інтеграція, у тому числі, що включає розмежування доступу (SoD) як на рівні ролей, профайлів, об'єктів авторизації, так і на рівні окремих транзакцій [11, 12].

### **1.6. Постановка задачі**

В наш час, інформаційні технології активно розвиваються та дають можливість автоматизувати будь-які процеси, для підвищення продуктивності роботи компаній. Розвиваються системи керування обліковими даними, які несуть основну функцію, а саме забезпечити контроль за створенням та подальшим керуванням цими записами.

Враховуючи наведене у розділі, існує необхідність у створенні системи керуванням облікових даних для оптимізації робочого процесу та підвищення продуктивності роботи тієї чи іншої організації. Особливо важливо, для ІТ

компанії «Intrasystem», оскільки в даній компанії працює більше як 100 осіб персоналу та ще близько 20 партнерів з якими компанія веде свою діяльність.

Отже метою дипломного проекту є розробка інформаційної системи адміністрування обліковими записами з базою даних та веб інтерфейсом для користувачів.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- провести аналітичний огляд предметної області роботи ІТ компанії «Intrasystem» та узгодити всі ключові аспекти роботи компанії, визначити основні напрямки для розробки системи;

- провести аналіз існуючих систем адміністрування обліковими записами, визначити їх слабкі сторони;

- визначити технології та засоби для розробки системи;

- визначити архітектуру web-застосунку та необхідний функціонал;

- спроектувати базу даних системи;

- розробити інтерфейс web-застосунку;

- розробити функціональну складову web-застосунку.

Інформаційна система управління обліковими даними дозволить оптимізувати бізнес-процеси в компанії, сприятиме підвищенню продуктивності роботи та скороченню документообігу в компанії, дозволить оптимізувати робочий процес з обліковими даними та забезпечить більш ефективний інформаційний захист компанії, зменшить завантаженість документообігом на кадровий та ІТ відділ.

## **1.7. Висновки до розділу 1**

Більшість сучасних підприємств намагаються автоматизувати свої процеси в середині компанії, тим самим зменшити документообіг та навантаження на деякі відділи та скоротити витрати на організацію роботи в цих самих відділах.

Система керуванням облікових даних є досить важливою складовою для впровадження даної системи на підприємстві. Особливу увагу при впровадженні даної системи на підприємстві потрібно приділити детальному вивченню роботи, а саме як влаштовані бізнес-процеси на підприємстві.

Проектована інформаційна система управління обліковими даними для компанії «Intrasystem» дозволить оптимізувати бізнес-процеси в компанії, сприятиме підвищенню продуктивності роботи та скороченню документообігу в компанії.

## РОЗДІЛ 2

### ТЕХНОЛОГІЇ ТА ІНТСРУМЕНТИ ДЛЯ СТВОРЕННЯ СИСТМЕИ КЕРУВАННЯ ОБЛІКОВИМИ ЗАПИСАМИ

Система керування обліковими даним, виконує безліч поставлених задач. По-перше, необхідно розуміти принцип роботи середовища, за допомогою якого буде впроваджено дане рішення. Основними кроками при впровадженні системи є:

- Створення бази даних, яка може бути не одна, враховуючи, під які задачі вона буде використовуватися в проекті (наприклад, це може бути як база даних працівників певного сектора компанії чи навіть якогось філіалу);
- Під налаштування «конектора» або навіть створення власного за допомогою якого, буде можливість об'єднати базу з системою;
- Створення та відлагодження роботи правил, за якими система буде працювати;
- Розробка користувацького інтерфейсу, дружнього до користувача під час використання даного рішення.

Кафедра КІТ				НАУ 22 22 37 000 ПЗ				
Виконав	Скрипник О.А.			Технології та інструменти для створення системи керування обліковими записами	Літера		Аркуш	Арку шів
Керівник	Воронін А.М.				ДП		24	8
Консульт.					УС-411 122			
Н-Контроль	Шевченко О.П.							



## 2.1. Технології для розробки бази даних

Для впровадження системи IDM, насамперед потрібно мати доступ до бази даних, в якій зберігаються та оновлюються списки користувачів. Розглянемо, які саме бази даних використовуються під час проектування.

В IDM використовується реляційна модель бази даних (далі «БД»). Реляційні БД – це один із типів, який й досі широко використовується БД загального призначення. В даному виді БД, дані та зв'язки організовані за допомогою таблиць. Кожен стовбець у таблиці має ім'я та тип. Кожен рядок представляє окремий запис або елемент в таблиці, який містить значення для кожного з стовпців.

На даний час значна кількість сховищ інформації для додатків розроблено за допомогою реляційних систем управління базами даних, є такі популярні рішення які використовуються це як: MySQL та Oracle SQL, Microsoft SQL.

Оскільки представлена компанія «Intrasystem», має відділи та підрозділи, то для проектної частини поділимо типи підключення та покажемо як це можна зробити.

В даному огляді, використовуватимемо SQL – дана мова є досить популярною серед фахівців, за допомогою її, є можливість забезпечення доступу та управління вмістом бази даних. Також не потрібно забувати і про переваги, яка має дана мова, а це є швидкість виконання запиту, простота та адаптивність.

Розглянемо процес, що стосується MySQL. По-перше MySQL побудовано за принципом клієнт-сервера. А саме клієнт-сервер це коли в зв'язці із користувацьким інтерфейсом існує сервер який має змогу обробляти ті запити, що надходять від користувача. MySQL в зв'язці клієнт-сервера виступає як сервер, який опрацьовує всі команди які надходять від користувача.

## 2.2. Організація роботи функціональної складової системи

Розробка даного програмного рішення, яке націлене на впровадження IDM. Маємо вже готове «програмне середовище», в якому під час створення рішення IDM вносимо певні налаштування та можемо інтегрувати навіть власний розроблений скрипт.

Для початку потрібно підключити базу або csv файл та створити в «Synchronization Editor», якщо коротко то «Synchronization Editor» являє собою єдину уніфіковану середовище підключення до цільових систем та можливість налаштовувати робочих процесів синхронізації даних, та створити «Synchronization project» рис.2.2.1.

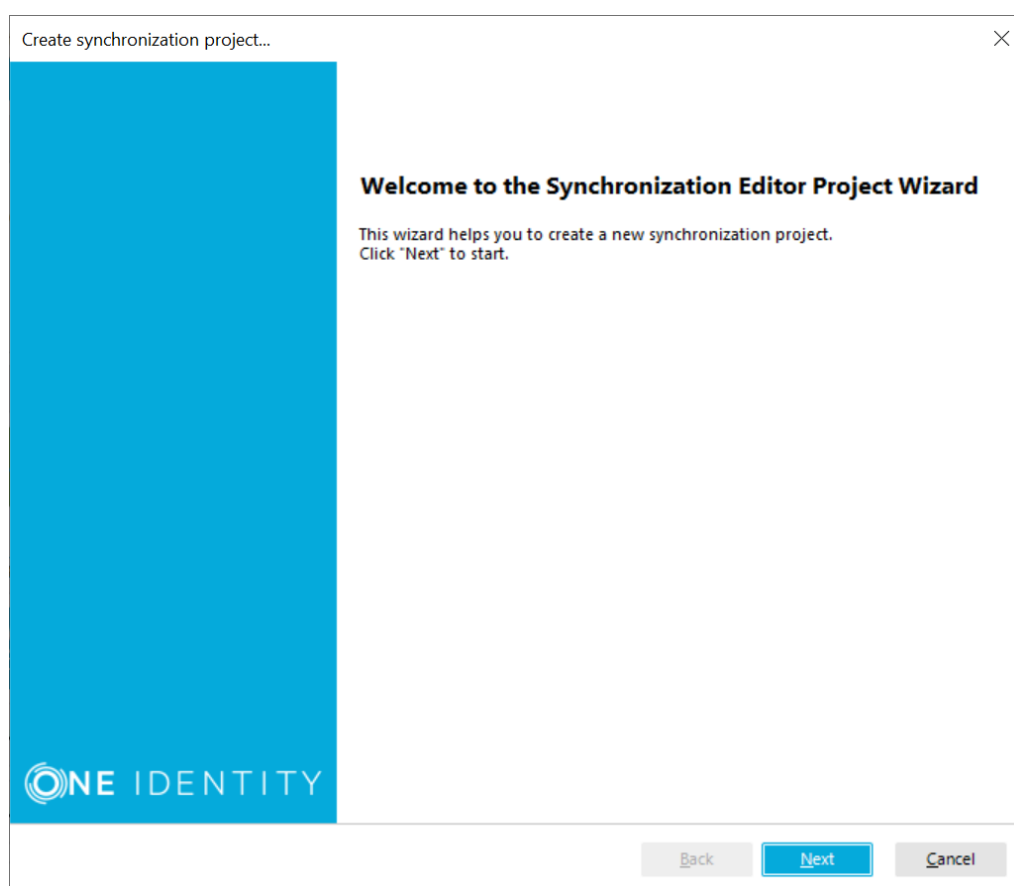


Рис.2.2.1. Старт роботи

Для старту роботи, слід використати конектор SAP. Рішення One Identity Manager, володіє конектори до системи SAP R/3. Дані конектори містять в собі декілька моделей для роботи з різними додатками та об'єктами SAP.

Розглянемо основні:

- Базовий модуль – це модуль за допомогою, якого відбувається синхронізація системи SAP EBR (інстанс, клієнти, компанії, ліцензії, принтери, облікові записи користувачів, ролі та профайли, включаючи складові, а також членство в ролях, прив'язки профайлів до ролей та ін.);
- Об'єкти авторизації – які в себе включають активності, транзакції, поля та все можливі прив'язки;
- Роботи с персоналом (SAP HCM) та структуровані профайли – дана можливість дає підключити One Identity Manager до автоматизованого джерелу даних о працівниках компанії та о оргштатної структури, що є в подальшому досить важливо для побудови рольової моделі та керування доступом в рамках концепції identity and access management;

Якщо обговорювати спосіб налаштування конектора SAP, то здійснюємо дані налаштування в графічній консолі в «Synchronization Editor». При підключення до оператора пропонується використовувати шаблон налаштування для того щоб налаштувати ту чи іншу частину конектора. Вибір шаблону рис.2.2.2.

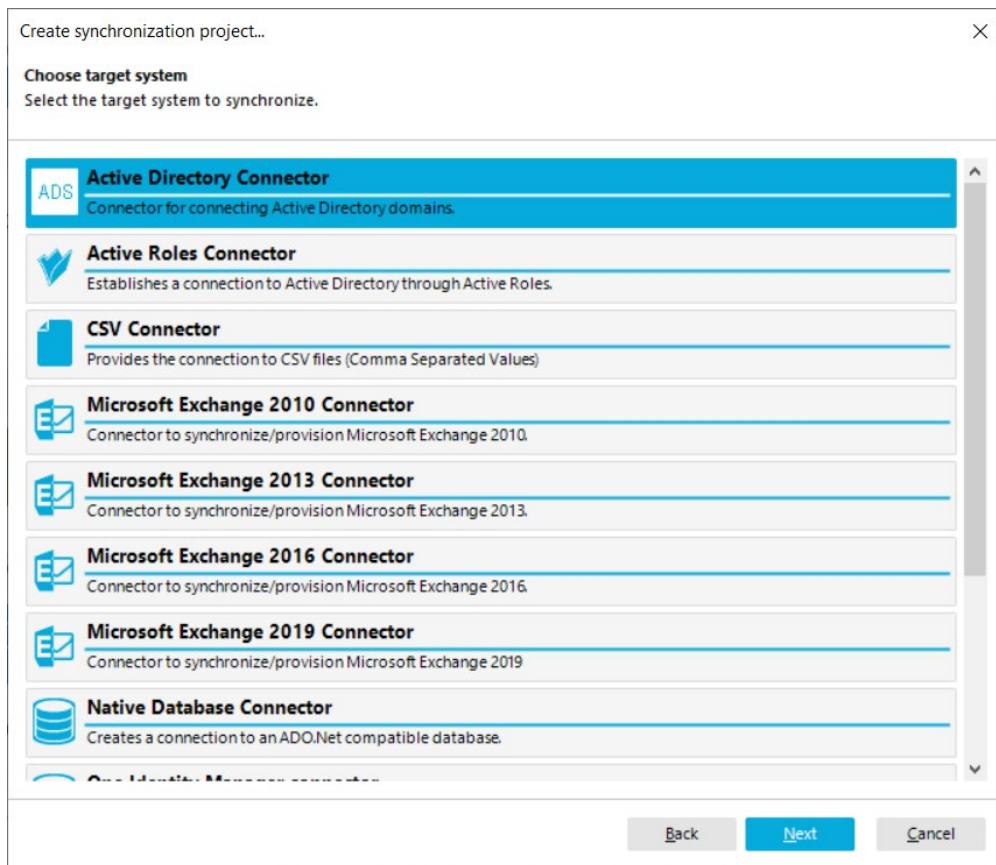


Рис.2.2.2. Вибір шаблону

Подальше налаштування відбувається при написанні коду. Ті функції, які закладені в шаблонах, включаючи схему, правила співставлення та прив'язки об'єктів та паролей, робочі процеси синхронізації для кожного класа об'єкту.

### 2.3. Розробка користувацького інтерфейсу

One Identity Manager, дає можливість досить креативно підходити до дизайну, так в даному середовищі вже є шаблон користувацького інтерфейсу, хоча можна його дещо модифікувати повністю під потреби замовника.

Якщо маємо бажання модифікувати зовнішній інтерфейс користувача, то нам знадобляться три основні мови програмування, а саме – Java Script, HTML та CSS.

Кожен із методів який використовується при проектування користувацького інтерфейсу, то HTML – це можна так сказати базовий блок будь-якого веба. Даний вид в собі має структуру веб-контенту.

Досить часто HTML називають мовою гіпертекстової розмітки, під гіпертекстом розуміється силки які об'єднують веб-сторінку друг з другом в межах одного веб-сайту або між-сайтами.

При створенні веб-реалізації та використанні HTML, створюємо об'єкти веб-дизайнера та розуміємо те що просто розміщуємо HTML сторінку. Ключові поняття HTML які ми використовуємо це:

- <div> == Контейнер;
- Style, class;
- Внутрішні віджети HTML;
- Форма, JavaScript тощо.

CSS – це мова ієрархічних правил, використовуємо як правило для того щоб передати зовнішній вид документу. Під час розробки, використовується CSS для шрифту, кольорів, макета. Ключові поняття, які можемо використати:

- Таблиці стилів
- Селектори CSS: частина селектора визначає на сторінці, до яких елементів буде застосована частина таблиці стилів. Селектори можуть ідентифікувати HTML-тег, клас і фільтрувати елементи з певними ідентифікаторами або властивостями HTML.

Java Script – це об'єктно-орієнтований язык з функціональністю першого класу.

## 2.4. Середовище розробки системи

Вибір середовища може бути, будь-яким, що дає певну гнучкість при обрані середовища проектування. One Identity Manager, має інтегровану систему для розробки та налагодження коду.

Використовуємо під час проектування такі середовища розробки, як Notepad++, Atom.

Atom – це безкоштовний текстовий редактор з відкритим вихідним кодом. Даний редактор підтримується на платформах Linux, macOS, Windows[13].

Перевагою даного редактору є те що він має підтримку плагинів які в свою чергу написані на Java Script. Більшість плагинів є відкритими для користувачів.

Notepad++ -це також безкоштовний редактор коду. Як на мій досвід даний редактор досить вдало підходить для написання скриптів саме для веб рішень тобто на мові HTML та CSS. Даний редактор не навантажує систему та є невеликим за обсягом що також вдало сприяє роботі системи[14].

## 2.5. Висновки до розділу 2

Керування обліковими даними дає можливість не лише створювати та керувати даними, але і робити повний контроль над ними та оптимізувати певні процеси в компаніях.

На початку роботи відбувається підключення бази даних користувачів або її створення. Також не достатньо її підключити до системи потрібно ще створити певний конектор, за допомогою якого можна повноцінно поєднати базу з системою розробки. Для цього доцільно використовувати Oracle SQL.

При реалізації веб-рішення для користувача, доцільно є можливість використання вже готового шаблону, який надається розробником програмного продукту, а саме One Identity Manager. Даний шаблон дає змогу на початкових етапах розробки перевірити роботоздатність системи, далі можна вносити певні зміни у відображення того який матиме вигляд робоча система. Це відбувається за допомогою таких «інструментів», як HTML або CSS. За їх допомогою, є можливість змінювати фон сторінки додавати певні зображення та змінювати шрифти.

## РОЗДІЛ 3

### РОЗРОБКА СИСТЕМИ КЕРУВАННЯ ОБЛІКОВИМИ ЗАПИСАМИ

При виконанні, даної роботи буде впроваджено систему керування обліковими записами в компанію «Intrasystem». Компанія Intrasystem надає широкий спектр комплексних рішень для ІТ-проектів, які відповідають світовим стандартам, а саме (консалтинг, аналіз та проектування, ІТ-аудит, модернізація та розвиток ІТ-інфраструктури, рішення в сфері безпеки, автоматизації та управління інформацією)[15]. Компанія нараховує близько 100 працівників та приблизно 20 партнерів, з якими співпрацює компанія.

#### 3.1. Проектування бази даних

Для початку роботи, слід використовувати Oracle Database, яку використовуватимемо для керування базами даних компанії Oracle.

Та після встановлення інструменту Oracle Database обирає, утиліту Database Configuration Assistant рис.3.1.1, за допомогою якої створимо базу в якій будемо надалі працювати.

Кафедра КІТ				НАУ 22 22 37 000 ПЗ				
Виконав	Скрипнік О.А.			Розробка системи керування обліковими записами	Літера		Аркуш	Арку шів
Керівник	Воронін А.М.				ДП		32	23
Консульт.					УС-411 122			
Н-контроль	Шевченко О.П.							



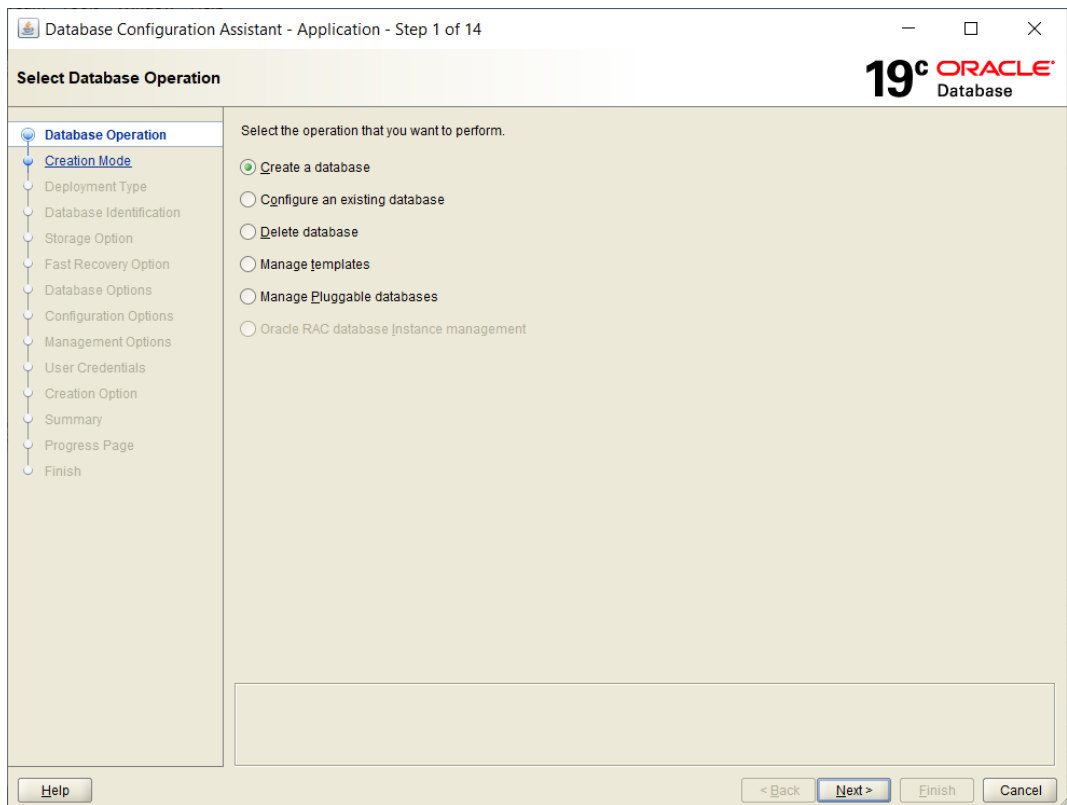


Рис.3.1.1. Database Configuration Assistant.

Налаштовуємо базу при створені а саме надаємо назву пароль, рис.3.1.2.

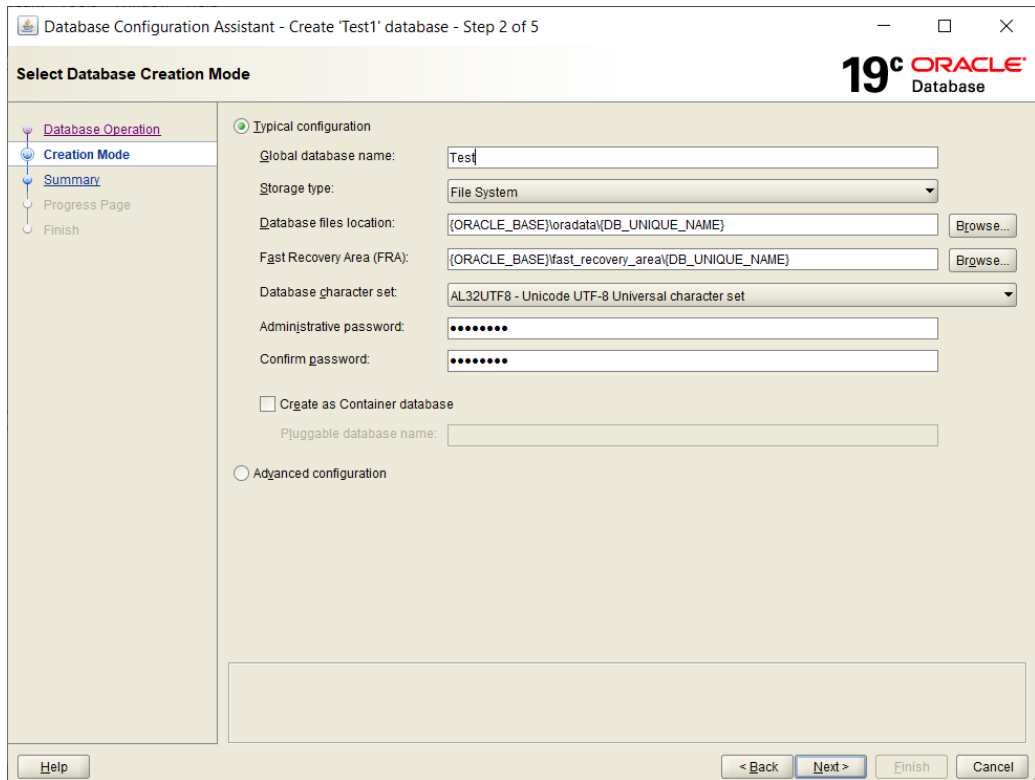


Рис.3.1.2. Налаштування бази.

Після того як база буде створена, запускаємо sql developer, де слід її наповнити користувацькими даними, але перед цим продемонструю структуру БД, рис 3.1.3 .

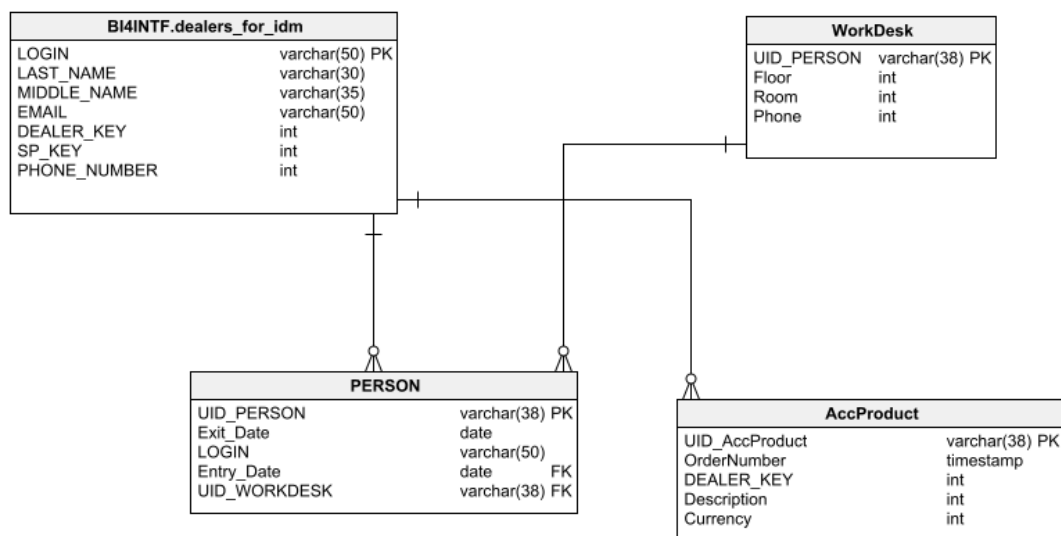


Рис.3.1.3. Структура БД.

Створена база в sql developer має такий вигляд рис.3.1.4.

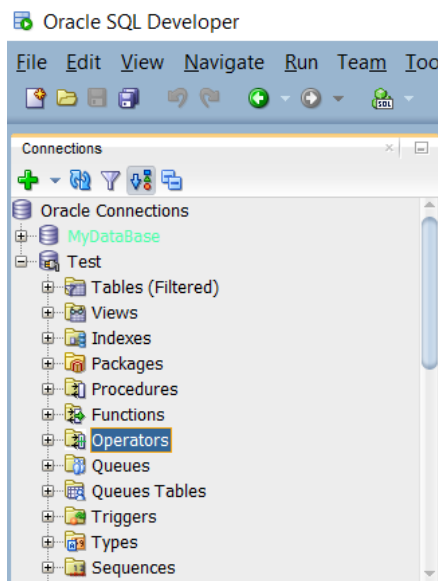


Рис.3.1.4. База в в sql developer.

Для наповнення, слід створити таблицю яку заповнимо даними, напишемо невеликий скрипт на мові sql, дамо назву таблиці та назві стовпців. Наведу фрагмент коду для створення таблиці та наповнення цієї таблиці даними рис3.1.5., весь код буде доданий в додатку В.

```

-- CREATE TABLE
CREATE TABLE system.DEALERS_FOR_IDM
(
  "LOGIN" VARCHAR2(255 BYTE),
  "FIRST_NAME" VARCHAR2(100 BYTE) NOT NULL ENABLE,
  "LAST_NAME" VARCHAR2(100 BYTE),
  "MIDDLE_NAME" VARCHAR2(40 BYTE),
  "EMAIL" VARCHAR2(80 BYTE) NOT NULL ENABLE,
  "DEALER_KEY" VARCHAR2(40 BYTE),
  "SP_KEY" VARCHAR2(200 BYTE),
  "PHONE_NUMBER" VARCHAR2(20 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

INSERT ALL
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('DR_GO_E_LOSHMANOV', 'Евген',
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_M_PURAVETS', 'Михайло'
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_O_MIRKOVSKIY', 'Олександр
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_E_SUPRUNENKO', 'Олена'
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('BR_GO_E_OBOZNA', 'Катерина',
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_A_SEMERENKO', 'Олексій
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_G_SLUZNUCHENKO', 'Ганн
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_R_BORODAY', 'Руслан',
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('VP_GO_N_MELNIK', 'Микола', 'М
INTO system.DEALERS_FOR_IDM (LOGIN, FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER) VALUES ('BR_GO_V_KOVALENKO', 'Ковален
SELECT 1 FROM DUAL;
INSERT ALL

```

Рис.3.1.5. Фрагмент коду створення бази.

### 3.2. Підключення бази даних та створення синхропроекту

Після того, як була створена база цільової системи, її потрібно підключити до One Identity Manager. Для цього створимо новий синхропроект.

Для цього потрібно на стартовій сторінці в Synchronization Editor, обрати Start new synchronization project та пройти етапи налаштування рис3.2.1-2.

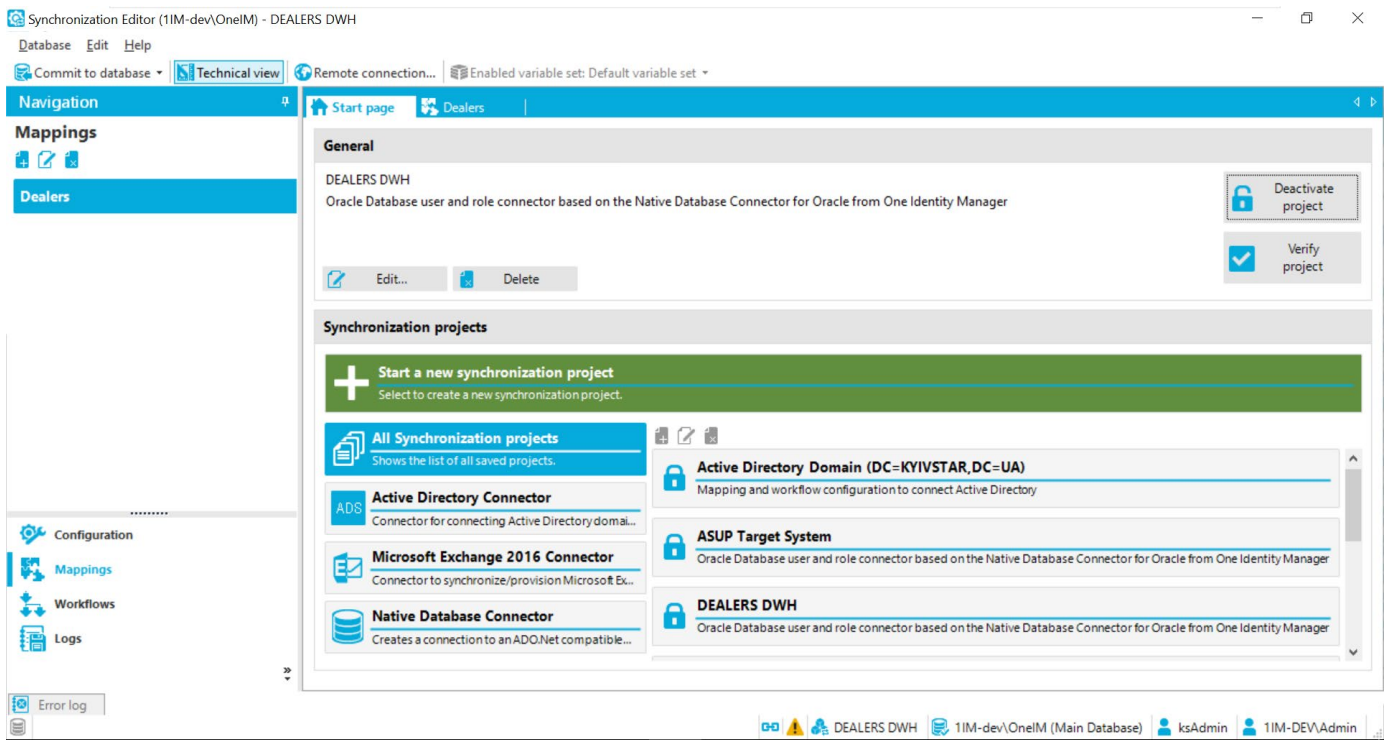


Рис.3.2.1. Стартова сторінка для створення синхропроекту.

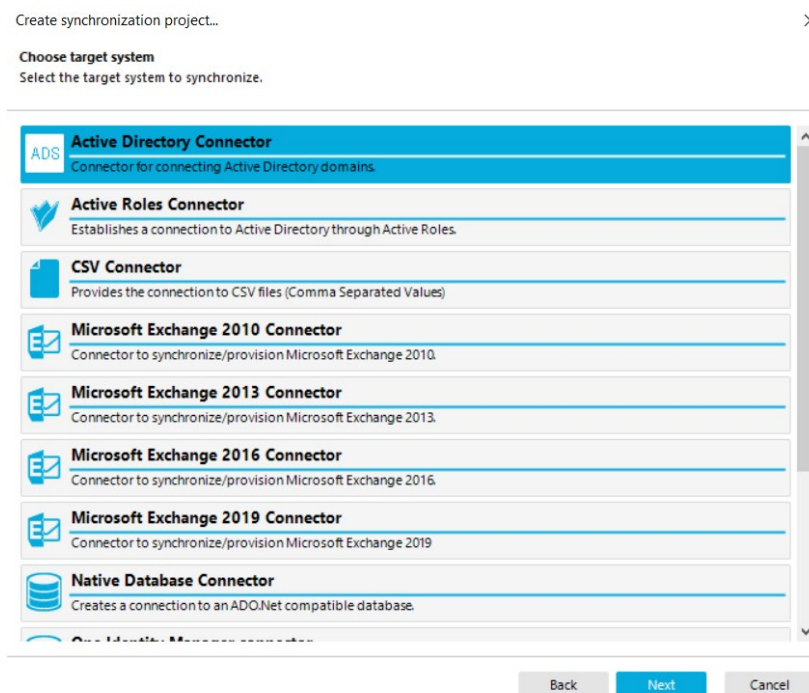


Рис.3.2.2. Етапи створення синхропроекту.

Після того як створений новий синхροпроект потрібно підключити цільову систему, для цього перейдемо в Synchronization Editor в Target System, рис.3.2.3.

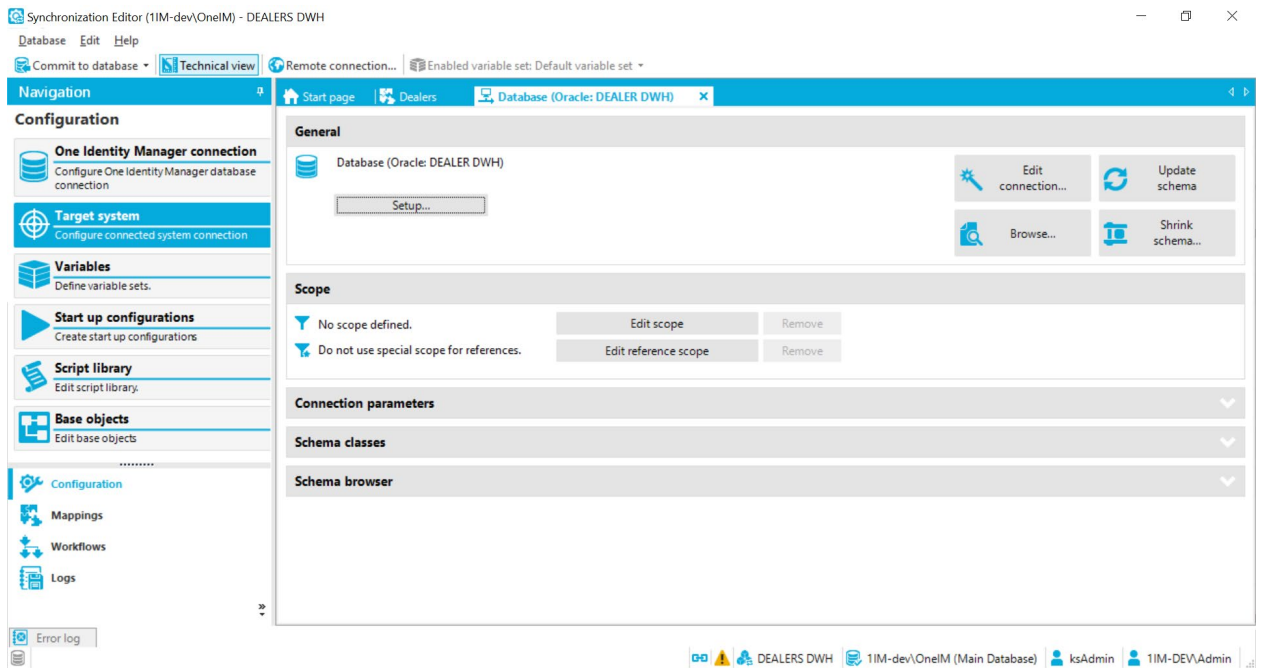


Рис.3.2.3. Target System.

Далі перейдемо до меню Edit connection, та налаштуємо підключення, рис.3.2.4., після цих налаштувань потрібно оновити схему та активувати синхροпроект.

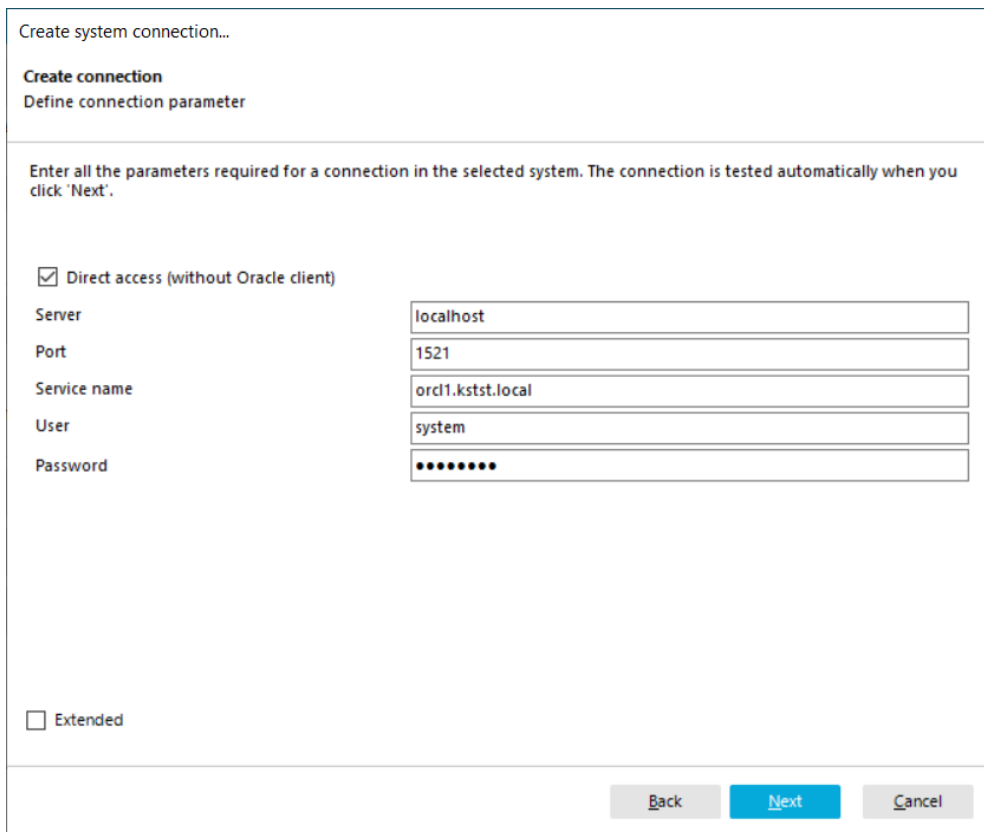


Рис.3.2.4 Налаштування підключення

Після всіх налаштування перевіriamo, як відбулося підключення та як відображається база рис.3.2.5.

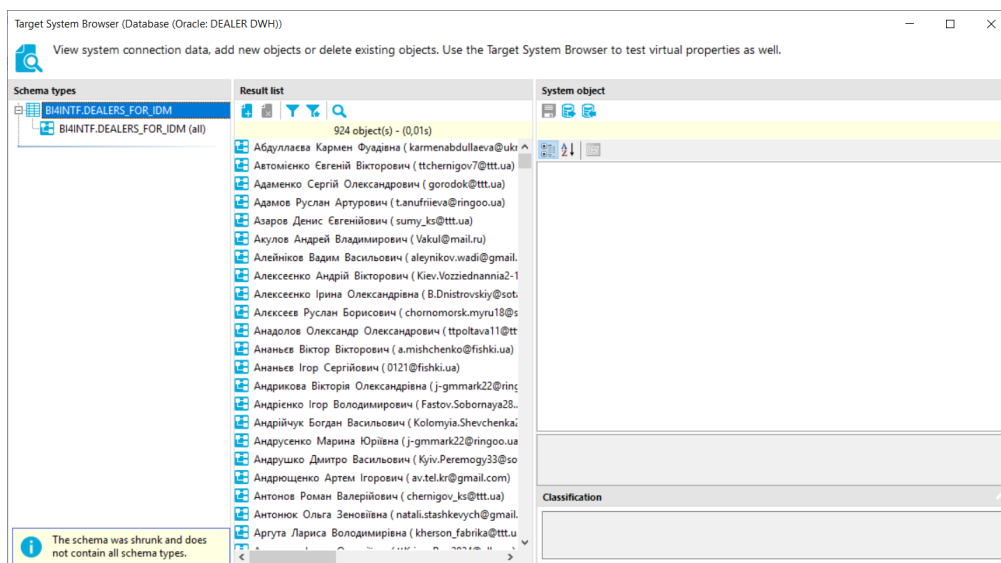


Рис.3.2.5. Відображення підключеної цільової системи.

### 3.3. Налаштування конфігурації синхропроект

Після того, як створили новий синхропроект та підключили до нього наші дані потрібно виконати певні конфігурації, для того щоб синхропроект коректно опрацьовував запити.

Для подальших дій, щоб була можливість акаунтам керувати своїми запитами в системі потрібно додати дозволи «Permission» та ролі додатків «Application Role», в One Identity Manager є можливість обрати вже існуючі розширення або навіть створити кастомні.

Реалізуємо це за допомогою використання додаткового програмного рішення, яке входить пакет програмних рішень One Identity Manager «Designer». Перейдемо на вкладку permission де створимо нові дозволи за допомогою task manager, результати дозволів на рис.3.3.1.

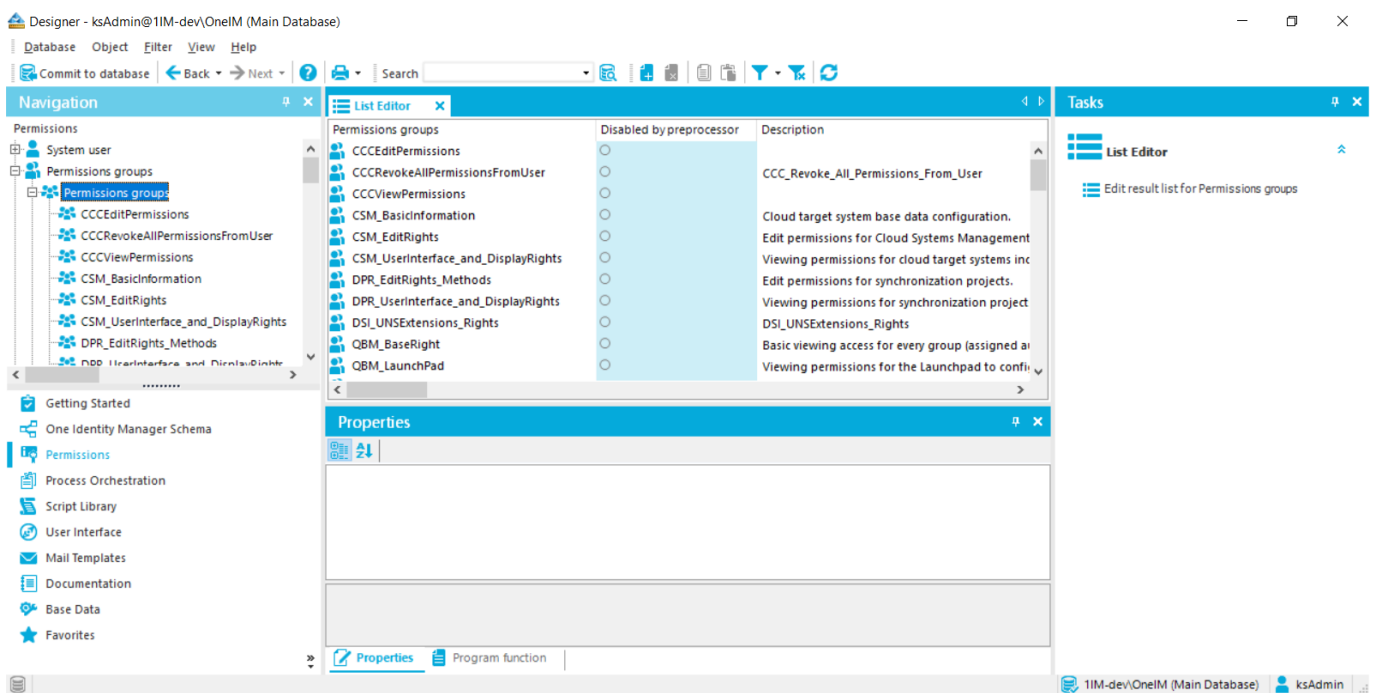


Рис.3.3.1 Дозволи «Permission»

Щоб дані зміни були додані до бази нам слід зробити «Commit to database».



### 3.3.1. Налаштування підключення до цільової системи

Конектор, і інтерфейс виконують однакову функцію, підключають цільову систему до One Identity Manager. Як правило, конектор використовується для добре відомої цільової системи, такої як Microsoft Active Directory. Сюди також входять загальні джерела даних, такі як LDAP, CSV, ADO.NET або загальне підключення API, як-от PowerShell Connect. Інтерфейси використовуються для рідкісних систем. Роз'єм, як правило, є найшвидшим способом підключення цільової системи, якщо він підтримується. У готовому варіанті One Identity Manager налаштовано на початок із підключенням до системи. Тим не менш, деякі передумови повинні бути перевірені, щоб Identity Manager дізнався про конфігурацію.

Далі проводимо налаштування які потрібні для правильної конфігурації  
рис.3.3.1.1.

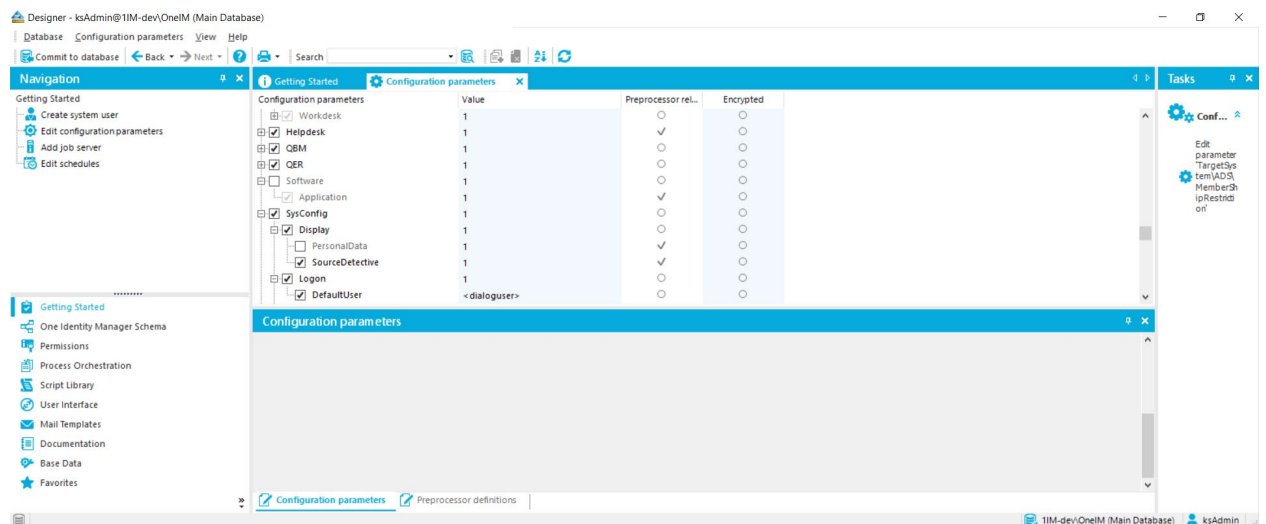


Рис.3.3.1.1 Налаштування

До того ж потрібно розуміти при налаштуванні щоб не було в подальшому проблем налаштувати основні параметри, такі як, PersonAutoFullSync описує поведінку того, як система повинна обробляти

неіснуючий об'єкт ідентифікації (Person), якщо під час звірки створюється обліковий запис у базі даних OneIM. У нашій конфігурації призначається існуючий об'єкт ідентифікації або, якщо такого об'єкта немає, він створюється. Зрозумійте, що в типовому проекті управління ідентифікацією дані HR (ідентифікації) імпортуються з авторитетних джерел (SAP HR, Peoplesoft тощо). Під час первинної синхронізації користувачі AD призначаються до існуючих об'єктів Person (пошук). У навчальному сценарії використовуємо можливість створювати об'єкти Person на основі користувачів AD (створювати). Це друга частина SearchAndCreate. PersonAutoDefault Цей параметр описує поведінку системи, якщо в OneIM створюється обліковий запис (параметр вище робив це для звірки). Конфігурація призводить до «нічого не робити», що означає, що жодні ідентифікатори не створюються або не призначаються новому обліковому запису. Два параметри конфігурації PersonAutoFullSync і PersonAutoDefault існують для багатьох цільових систем і мають бути налаштовані окремо. Exchange2000 — це параметр, який вказує, що MS Exchange можна підключити. PersonUpdate керує потоком даних між ідентифікаторами (об'єктами особи) та об'єктами облікового запису Active Directory. Зазвичай Identity Manager контролює Active Directory. Це означає, що зміни властивостей об'єкта особи також змінюють властивості облікового запису AD. У сценаріях, де не підключено авторитетну систему управління персоналом, цю поведінку можна змінити, перевібивши цей параметр конфігурації. Використовуйте опис параметра конфігурації або посібник OneIM, щоб дізнатися про його дію. Ця конфігурація існує лише для цільової системи Active Directory. DefaultMailDomain дозволяє нам створити адресу електронної пошти за замовчуванням для будь-якого об'єкта person. Це не залежить від того, скільки систем має поштові скриньки. Усі можна призначити одній людині, але тільки одну адресу може бути адресою за замовчуванням. AutoUpdate вмикає функцію, коли служба OneIM і двійкові файли інтерфейсу автоматично оновлюються з бази даних.

### 3.3.2. Конфігурація політики паролів та ERP-ресурс

Головним чинником є налаштування безпеки паролів. Політики паролів будемо налаштувати в Designer і Manager. На відміну від Менеджера, який завжди показує політику паролів як частину розділу адміністрування цільової системи, Designer показує всі політики паролів в одному місці (незалежно від цільової системи). З точки зору конфігурації політики паролів, немає різниці між обома інструментами.

Деякі поля захищені від запису. Це вказує на те, що цей об'єкт із коробки не можна змінити, як завжди. Тим не менш, деякі поля НЕ захищені від запису. У випадку політик паролів можемо змінити конфігурацію стандартної політики, але не можете змінити імена, опис чи власника. Якщо хочете це зробити, ви повинні створити спеціальну політику паролів і призначити її об'єкту замість стандартної політики на рис.3.3.2.1 видно картку користувача.



Employee	
<b>Балясний Андрій Володимирович (ANDREY.BALYASNIY)</b>	
Type	внутренний сотрудник
Full name	Балясний Андрій Володимирович
Default email address	andrey.balyasniy@kyivstar.net
Mobile phone	
Primary location	
Primary department	Сектор розвитку мереж
Position	Інженер з будівництва об'єктів мережі 1 категорії
Is manager?	–
Manager	Кондратенко Юрій Миколайович (YURIY.KONDRATENKO)
Employee status	Штатный
Disabled permanently	–
Temporarily disabled	–
Central user account	ANDREY.BALYASNIY
Identity	Primary identity

Рис.3.3.2.1 картка користувача

Для цього відкриємо Meneger, та увійдемо у систему, використовуючи обліковий запис користувача. Оберемо цільову систему Active Directory та

розгорнемо фільтр основні дані конфігурації. Створимо політику паролів вузла фільтра. Після того як створимо виберемо одну політику паролів, щоб відобразити огляд рис.3.3.2.2.

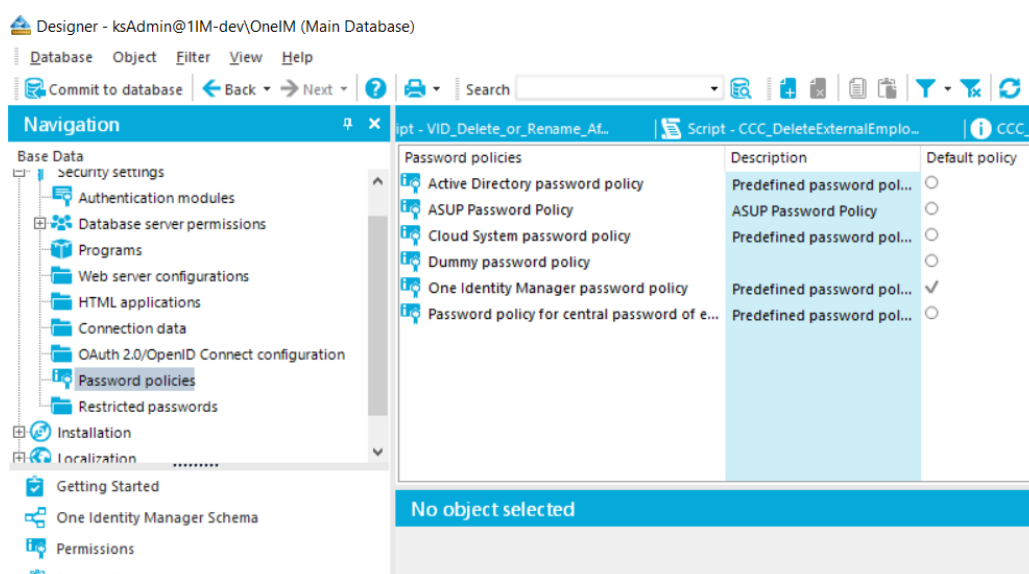


Рис.3.3.2.2 Політика паролів

Далі зробимо певні налаштування та присвоюємо всій групі користувачів, що маємо.

### 3.4. Налаштування бізнес процесів

Під час впровадження даного програмного рішення, потрібно досить чітко налаштувати бізнес процеси які і є основою.

У One Identity Manager можемо створити скільки завгодно дерев ролей. Кожному дереву ролей потрібен кореневий елемент, іменованій класом ролей. Цей клас ролей використовується для визначення ресурсів, які можна призначати, і успадкування ресурсів у цьому дереві ролей. Якщо буде порушене успадкування, це неможливо змінити після першого збереження

класу ролі. Щоб змінити це, повинні видалити клас ролі та створити його заново. Через це має сенс правильно спланувати класи ролей, перш ніж почати створювати об'єкт.

Дані налаштування виконуємо в «Manager» в графі «Business Role», після всіх налаштувань слід пам'ятати про те, що автоматичне оновлення знижує продуктивність бази даних, оскільки ця функція створює великий обсяг трафіку даних.

Для того щоб призначити структурі ролей, налаштуємо цю функцію для кожного класу об'єктів, яку будемо обробляти. Це потрібно для обмеження кількості обчислень, які DBQueueProcessor має виконати для перевірки призначень за спадковістю. Починаємо з порожнього вибору нового класу ролей, що означає, що ролям нічого не можна призначати та в результаті налаштувань матимемо ось такий результат рис.3.4.1-2.

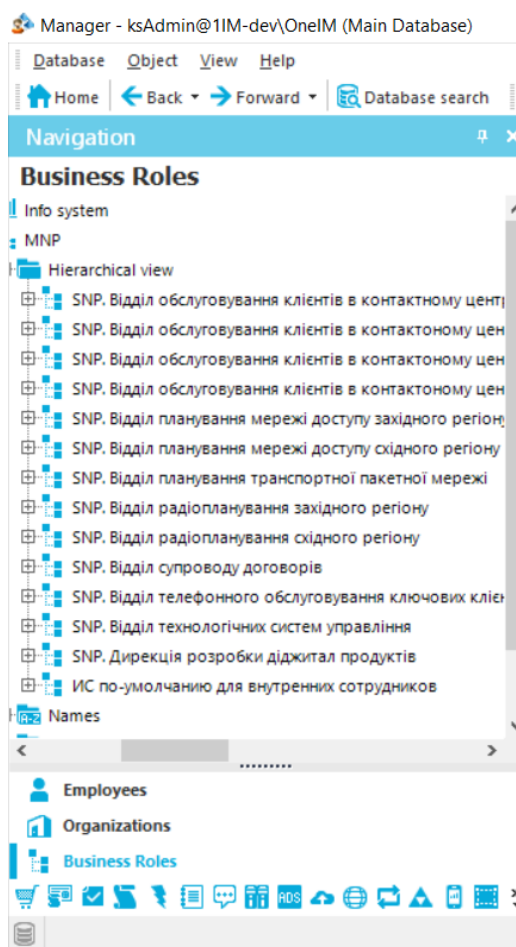


Рис.3.4.1. Результати налаштувань

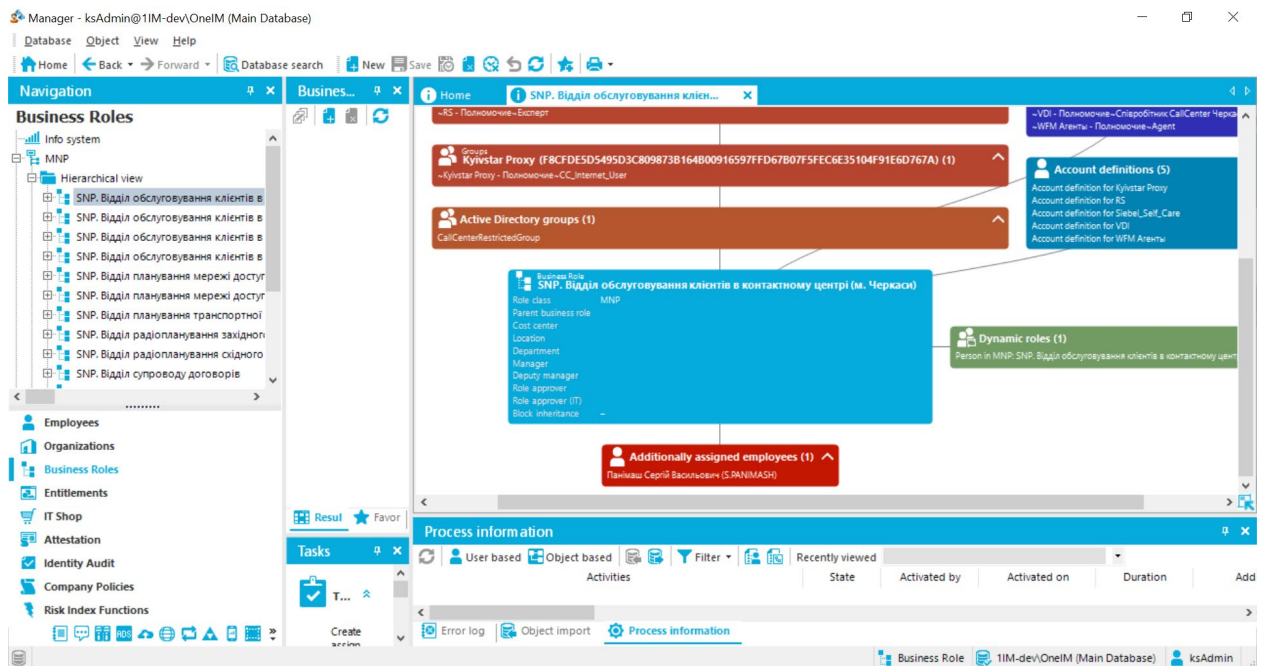


Рис.3.4.2. Вигляд створеного бізнес процесу

### 3.5. Розширені налаштування

Для реалізації розширених налаштувань потрібна можливість розширити функціональність програмного продукту One Identity Manager.

Для цього в «Designer» є можливість використати вже готові скрипти або є можливість додати власний скрипт який під час синхронізації буде виконувати ту чи іншу функціональність. Якщо говорити про відладку то є можливість зробити відладку вшитим інтерпретатором рис.3.5.1 приклад інтерпретатора.

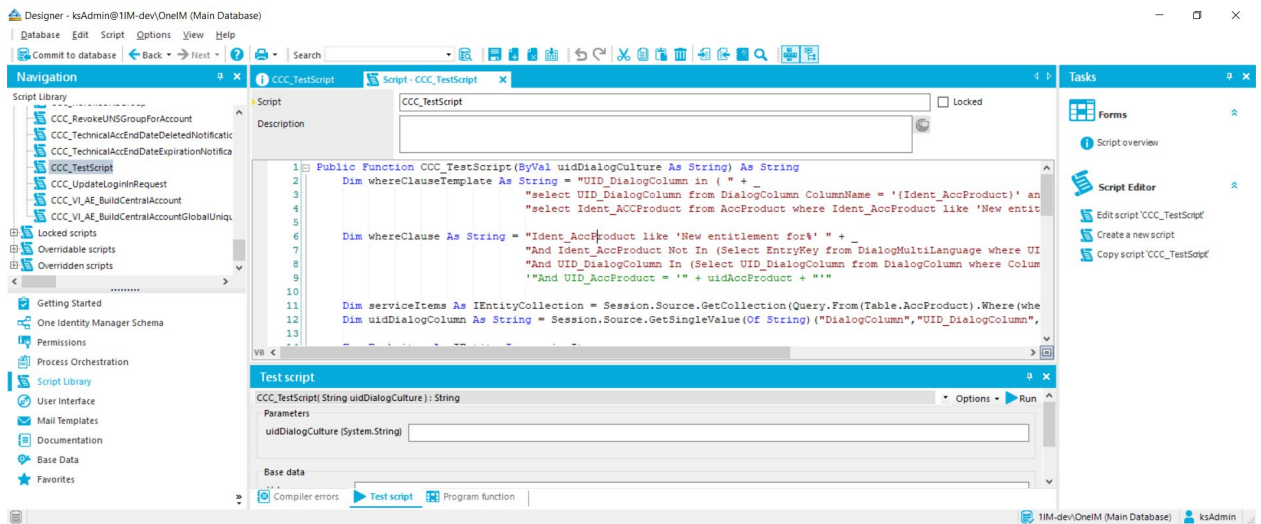


Рис.3.5.1. Інтерпретатор

В даному проєкті будемо додавати задачу яка при синхронізації проєкту буде виконуватися автоматично. Реалізуємо скрипт за допомогою, якого буде відбуватися оновлення даних облікового запису користувача, писатиму на Visual Basic тому що One Identity Manager підтримує Visual Basic, для таких цілей. Фрагмент коду, а повний формат коду представлений в додатку А.

```
Public Overridable Function ADS_PersonUpdate_ADSAccount(ByVal UID_Account As String, ByVal OldAccountDN As String, ByVal ProcID As String) As Boolean
```

Даний фрагмент коду потрібен для оновлення запису облікового ADSA, та оновлення пов'язаних атрибутів запису особи. Також важливим є те що виконується завантаження об'єктів ADSA об'єктів.

```
Dim Account As ISingleDbObject =
Connection.CreateSingle("ADSAccount", UID_Account)

Dim fkPerson As IForeignKey = Account.GetFK("UID_Person")

If Not fkPerson.IsEmpty Then

Dim Person As ISingleDbObject = fkPerson.Create()
```

```

        If Person.GetValue("ImportSource").String = "" AndAlso
Person.GetValue("DistinguishedName").String = "" Then

```

Вище зазначений фрагмент коду потрібен для перевірки зв'язків ADSA, та для виключення списків які не слід призначати користувачам.

```

        Dim strCentralAccount As String =
Account.GetValue("SAMAccountName").String

        Dim strAccountPattern As String = "^(" &
Connection.GetConfigParm("TargetSystem\ADS\PersonExcludeList").T
oString & ")$"

        If Regex.IsMatch(strCentralAccount,
strAccountPattern, RegexOptions.IgnoreCase Or
RegexOptions.CultureInvariant) Then
            Return True                                End
If

        VID_PutValueSafe(Person, "ImportSource",
"ADS")

        Else
            If Person.GetValue("ImportSource").String <>
"ADS" Then
                Return                                End If

            If
StrComp(Person.GetValue("DistinguishedName").String,
OldAccountDN, CompareMethod.Text) <> 0 Then
                Return                                End If
            End If

            VID_PutValueSafe(Person, "Canonicalname",
Account.GetValue("Canonicalname").String)

```



```

        VID_PutValueSafe(Person, "DistinguishedName",
Account.GetValue("DistinguishedName").String)
        Dim f As ISqlFormatter = Connection.SqlFormatter

```

Вище зазначений код опрацьовує та робить зіставлення для облікових записів для оновлення даних.

Також реалізуємо програмно реалізацію запису користувачем, за допомогою якого користувач матиме можливість здійснювати запит в веб-середовищі One Identity Manager. Фрагмент коду, а повний формат коду представлений в додатку Б.

```

Public Sub CCC_Create_New_Enable_Accounts_Request(ByVal UID_Org
As String, ByVal OrderReason As String, ByVal OrderDetail1 As
String)
    Dim colPersons As IEntityCollection
    Dim strUIDPerson As String
    Dim strUIDPersonHead As String
    Dim strCentralAccount As String
    Dim strUID_SCO As String
    Dim f As ISqlFormatter = Connection.SqlFormatter
    Dim strUIDPersonIDM =
Session.Config.GetConfigParm("Custom\Request\IDMUser")

    Dim qPerson = Query.From("Person") _
.Where(f.AndRelation(
    f.Comparison("IsInactive", True, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("CCC_IsManualDeactivated", False, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("EntryDate", Date.UtcNow, ValType.Date,
CompareOperator.LowerOrEqual, FormatterOptions.None),
    f.Comparison("EntryDate", DbVal.MinDate, ValType.Date,
CompareOperator.GreaterThan, FormatterOptions.None),

```

```

        f.OrRelation(f.Comparison("ExitDate", "", ValType.Date,
CompareOperator.Equal, FormatterOptions.ConvertNull),
                    f.Comparison("ExitDate", Date.UtcNow,
ValType.Date, CompareOperator.GreaterThan,
FormatterOptions.ConvertNull)),
        f.Comparison("CustomProperty10", "1", ValType.String,
CompareOperator.NotEqual, FormatterOptions.None),
        f.OrRelation(f.Comparison("CCC_EmpStatus", "EMP",
ValType.String, CompareOperator.Equal, FormatterOptions.None),
                    f.Comparison("CCC_EmpStatus", "DFT",
ValType.String, CompareOperator.Equal, FormatterOptions.None),
                    f.Comparison("CCC_EmpStatus", "TEM",
ValType.String, CompareOperator.Equal, FormatterOptions.None)),
        f.Comparison("CCC_Deleted", False, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None))) _
.SelectAll()

```

### 3.6. Демонстрація роботи Web-середовища

One Identity Manager, має свій портал на якому користувачі можуть виконувати свої записи. Виберемо створену персону та продемонструємо, як відбувається робота рис.3.6.1-3.6.2.

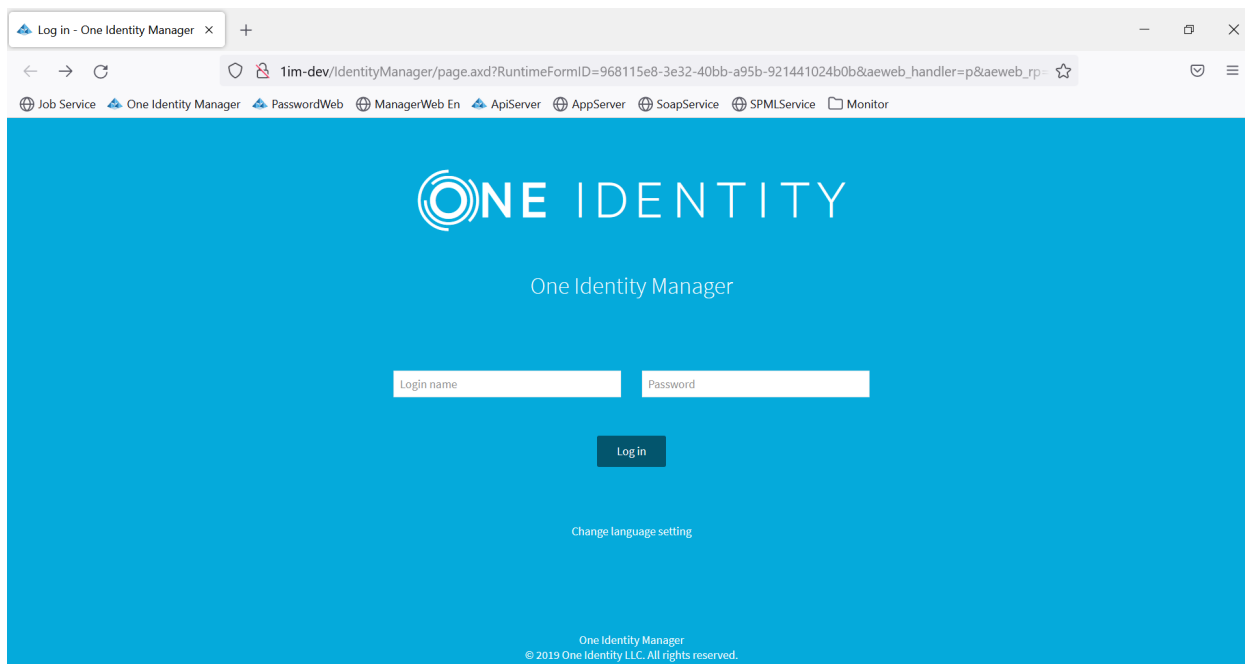


Рис.3.6.1. Стартова сторінка

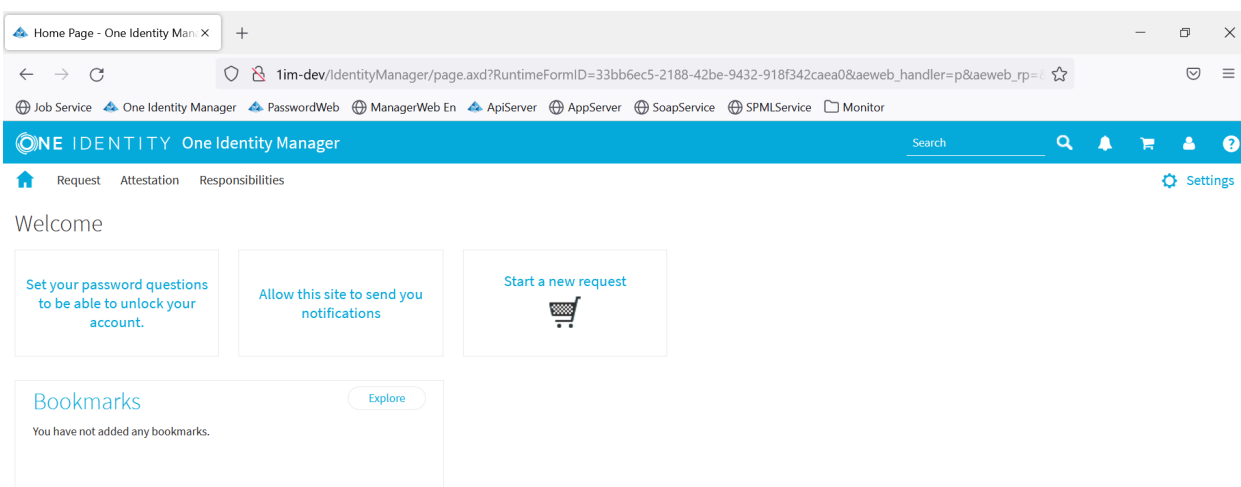


Рис.3.6.2. Користувачький інтерфейс

Користувач в веб-середовищі має можливість замовити ту чи іншу послугу до якої має доступ все це відбувається в Start new request, що і зробимо.

При переході до запиту та виберемо той запит, який потрібен, рис.3.6.3.

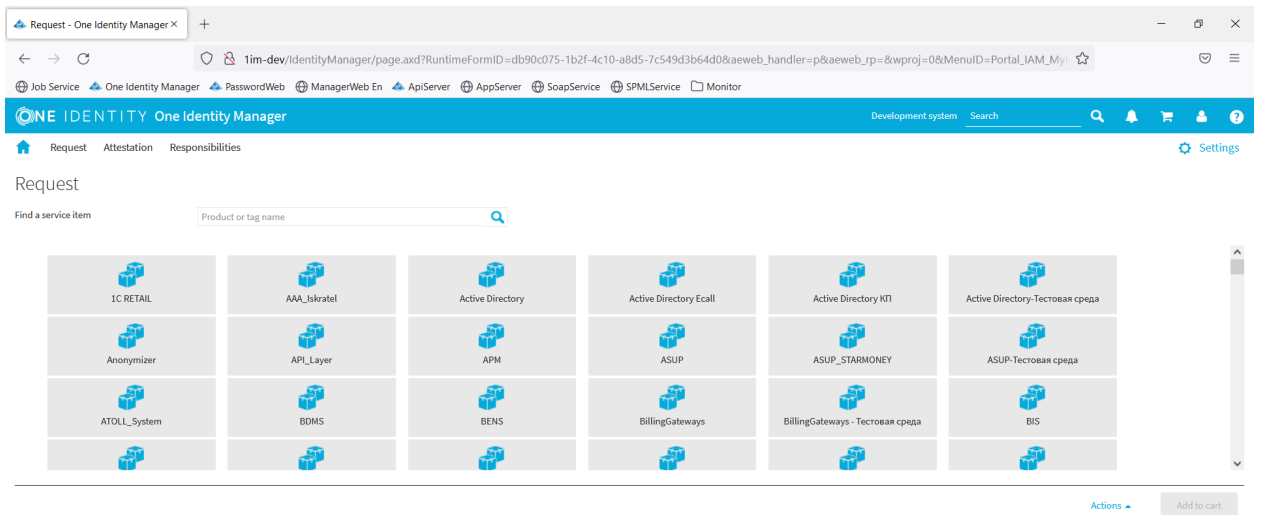


Рис.3.6.3. Доступні запити користувачу

Виберемо запит та продемонструємо як виглядає погодження, рис.3.6.4.

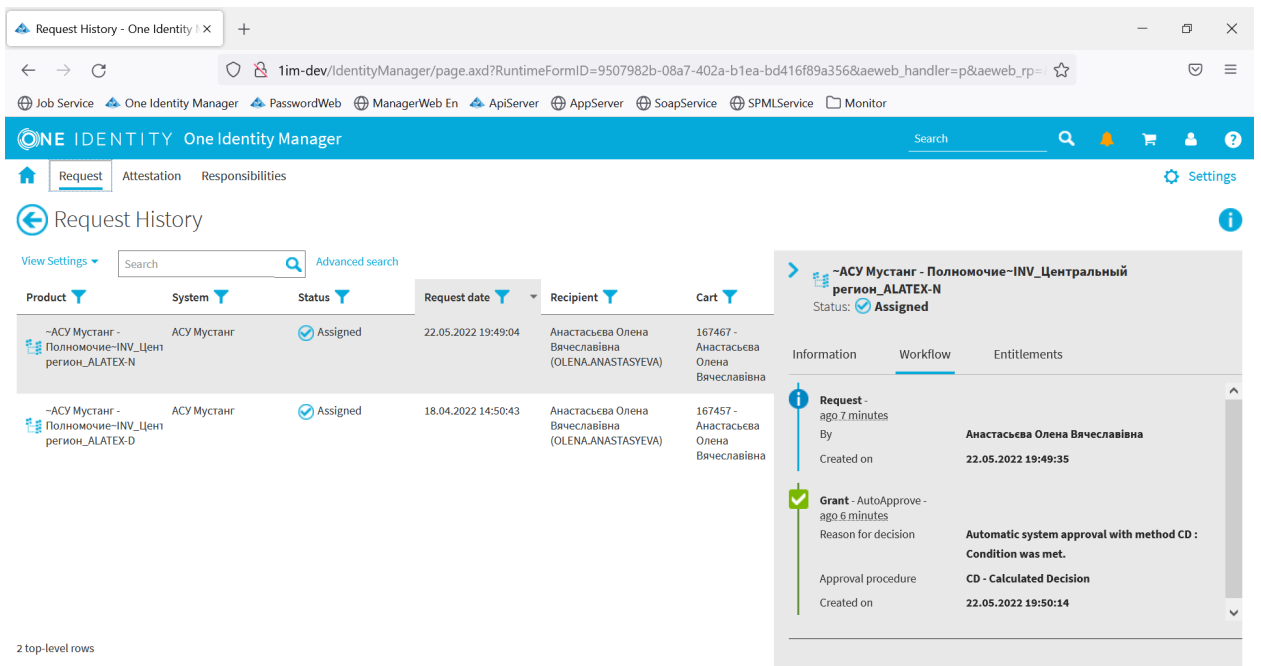


Рис.3.6.4. Процесс опрацювання заявки.

Як видно опрацювання даної заявки пройшло успішно та доступ даному користувачу був наданий, зі сторони Idm погодження заявки виглядає наступним чином, рис.3.6.5.

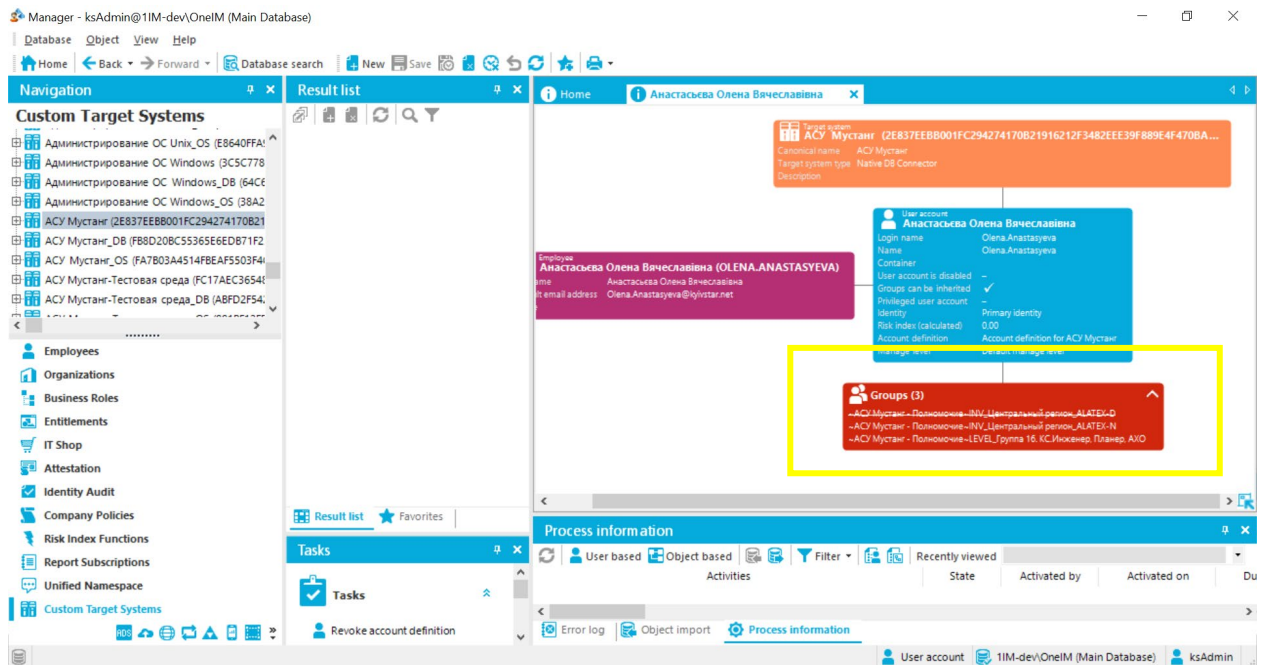


Рис.3.6.5. Доданий запит на сторони Idm.

### 3.7. Висновки до розділу 3

Під час розгляду даного розділу, було розглянуто досить багато чинників які повинні бути досить правильно налаштовані в програмному середовищі. Тому що в даному програмному середовищі слід пам'ятати про те що є деякі чинники, які при збереженні, не матимемо можливість змінити налаштування.

До того ж слід розуміти те що є можливість створити кастомні скрипти які розширюють можливості даного програмного продукту. Під час виконання даної роботи розроблено скрипт, який буде блокувати користувачів, у яких закінчився термін дії (наприклад, коли з працівником закінчується дія контракту на роботу). Під час створення даного виду скрипта необхідно врахувати вибірку із бази даних для, того щоб була можливість пошуку особи за певним ідентифікатором, тобто ключем, наприклад це може бути як `userkey` так `enddata`.

Також розглянули можливість кастомізації веб-середовища. Веб-середовище присутнє, але даний програмний продукт дає змогу налаштувати відображення веб.

## ВИСНОВКИ

В наш час питання керування обліковими записами досить часто постає перед різноманітною кількістю компаній для того щоб скоротити документообіг та фінансові витрати.

Так як на сьогоднішній день керування обліковими записами це не лише присвоєння якихось ролей чи просто навіть надання доступу. Впровадження даної системи в організацію дає можливість не лише оптимізувати деякі процеси, а й вкладе вагомий вклад в інформаційну безпеку компанії. Сьогодні інформаційна безпека будь-якої компанії це не лише гроші а й її репутація. Як говориться заробити репутацію досить складно, а ось втрати можна за одну секунду, а в інформаційному просторі взагалі можна втрати при натисненні однієї клавіши.

Дана робота демонструвала як саме впроваджується система IDM на прикладі компанії «Intrasystem», в якій є працівники, з певним рівнем доступу до тих чи інших ресурсів. Так як під час виконання було зрозуміло, що певні процеси компанії в бізнес-моделі є не оптимізовані та була необхідність зменшити документообіг та покращити інформаційну безпеку компанії.

Після аналізу було з'ясовано, що при запитах та видачі облікових записів користувачам витрачається досить багато часу та витрат на норму години працівників, які займаються даним питанням. До того ж деякі облікові записи після того, як користувача або звільнили або перевели в інший офіс чи філіал не змінюється інформація або не блокується даний обліковий запис.

В процесі виконання проекту для розробки було застосовано такі технології та інструменти, як:

- MySQL – система управління реляційними базами даних.
- Програмний продукт One Identity Manager Manager;
- Програмний продукт One Identity Manager Designer;

- Програмний продукт One Identity Manager Synchronization Editor;
- HTML, CSS – для створення користувацького інтерфейсу;
- Visual Basic – для створення кастомних скриптів.



## СПИСОК БІБЛІОГРАФІЧНИХ ПОСИЛАНЬ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Які задачі вирішує IAM системи? [Електронний ресурс] Режим доступу: <https://habr.com/ru/post/221159/> (дата звернення 06.01.22) – Назва екрана.
2. Огляд One Identity Manager Частина 1. [Електронний ресурс] Режим доступу: <https://www.anti-malware.ru> (дата звернення 10.01.22) - Назва екрана.
3. LDAP [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/> (дата звернення 10.01.22) - Назва екрана.
4. Як з'являвся One Identity Manager [Електронний ресурс] Режим доступу: <https://www.anti-malware.ru> (дата звернення 25.02.22) - Назва екрана.
5. Впровадження IdM [Електронний ресурс] <https://habr.com/ru/company/solarsecurity/blog/340960/> Режим доступу: (дата звернення 01.03.22) - Назва екрана.
6. Впровадження IdM. Процедури та технічні середовища від базових до IdM [Електронний ресурс] Режим доступу: <https://habr.com/ru/company/solarsecurity/blog/412687/> (дата звернення 25.03.22) - Назва екрана.
7. Опис TheHarvest [Електронний ресурс] Режим доступу: <https://kali.tools/?p=2286#:~:text=theHarvester%20%E2%80%94%20%D1%8D%D1%82%D0%BE%20%D0%B8%D0%BD%D1%81%D1%82%D1%80%D1%83> (дата звернення 27.03.22) - Назва екрана.
8. Поява One Identity Manager [Електронний ресурс] Режим доступу: <https://www.anti-malware.ru> (дата звернення 01.04.22) - Назва екрана.

9. Архітектура One Identity Manager [Електронний ресурс] Режим доступу: <https://www.anti-malware.ru/reviews/one-identity-manager-80-part-2#part2> (дата звернення 07.04.22) – Назва екрана.

10. Структура One Identity Manager [Електронний ресурс] Режим доступу: <https://www.anti-malware.ru/reviews/one-identity-manager-80-part-2#part2> (дата звернення 15.04.22) – Назва екрана.

11. Підключення цільової системи [Електронний ресурс] Режим доступу: <https://support.oneidentity.com/identity-manager/8.1.5> (дата звернення 25.04.22) - Назва екрана.

12. Підключення конекторів [Електронний ресурс] Режим доступу: <https://support.oneidentity.com/identity-manager/8.1.5> (дата звернення 27.04.22) - Назва екрана.

13. Atom [Електронний ресурс] Режим доступу: <https://hyperhost.ua/info/uk/22-populyarnikh-redaktoriv-kodu-v-2022> (дата звернення 30.04.22) - Назва екрана.

14. Notepad ++ [Електронний ресурс] Режим доступу: <https://ru.wikipedia.org/wiki/Notepad%2B%2B> (дата звернення 30.04.22) - Назва екрана.

15. Інформація про компанію [Електронний ресурс] Режим доступу: <https://www.intrasystems.ua/pro-kompaniyu/> (дата звернення 01.05.22) – Назва екрана.

## ДОДАТКИ

Додаток А

### Програмна реалізація оновлення даних облікового запису користувача

```
Public Overridable Function ADS_PersonUpdate_ADSSAccount (ByVal
UID_Account As String, ByVal OldAccountDN As String, ByVal
ProcID As String) As Boolean
    Dim Account As ISingleDBObject =
Connection.CreateSingle("ADSSAccount", UID_Account)
    Dim fkPerson As IForeignKey =
Account.GetFK("UID_Person")
    If Not fkPerson.IsEmpty Then
        Dim Person As ISingleDBObject =
fkPerson.Create()
        If Person.GetValue("ImportSource").String = ""
AndAlso Person.GetValue("DistinguishedName").String = "" Then
            Dim strCentralAccount As String =
Account.GetValue("SAMAccountName").String
            Dim strAccountPattern As String = "^(" &
Connection.GetConfigParm("TargetSystem\ADS\PersonExcludeList").T
oString & ")$"
            If Regex.IsMatch(strCentralAccount,
strAccountPattern, RegexOptions.IgnoreCase Or
RegexOptions.CultureInvariant) Then
                Return True
            End If
        End If
        VID_PutValueSafe(Person, "ImportSource", "ADS")
    Else
        If Person.GetValue("ImportSource").String <>
"ADS" Then
            Return True
        End If
    End If
    If
StrComp(Person.GetValue("DistinguishedName").String,
OldAccountDN, CompareMethod.Text) <> 0 Then
        Return True
    End If
End Function
```

```

End If

        VID_PutValueSafe(Person, "Canonicalname",
Account.GetValue("Canonicalname").String)
        VID_PutValueSafe(Person, "DistinguishedName",
Account.GetValue("DistinguishedName").String)
        Dim f As ISqlFormatter = Connection.SqlFormatter
        VID_PutValueSafe(Person, "City",
Account.GetValue("Locality").String)

VID_PutValueSafe(Person, "CompanyMember",
Account.GetValue("Company").String)
        VID_PutValueSafe(Person, "DefaultEmailAddress",
Account.GetValue("Mail").String)
        VID_PutValueSafe(Person, "Description",
Account.GetValue("Description").String)
        VID_PutValueSafe(Person, "ExitDate",
Account.GetValue("AccountExpires").Date)
        VID_PutValueSafe(Person, "Fax",
Account.GetValue("Fax").String)
        VID_PutValueSafe(Person, "FirstName",
Account.GetValue("GivenName").String)
        VID_PutValueSafe(Person,
"IsTerminalServerAllowed",
Account.GetValue("AllowLogonTerminalServer").Bool)
        VID_PutValueSafe(Person, "LastName",
Account.GetValue("Surname").String)
        VID_PutValueSafe(Person, "MiddleName",
Account.GetValue("Initials").String)
        VID_PutValueSafe(Person, "PersonalTitle",
Account.GetValue("Title").String)

VID_PutValueSafe(Person, "Phone",
Account.GetValue("TelephoneNumber").String)
        VID_PutValueSafe(Person, "PhoneMobile",
Account.GetValue("Mobile").String)
        VID_PutValueSafe(Person, "PostalOfficeBox",
Account.GetValue("PostOfficeBox").String)
#If ANYCOUNTRY Then
        VID_PutValueSafe(Person, "UID_DialogState",
Connection.GetSingleProperty("DialogState", "UID_DialogState",

```

```
                f.Comparison("NationalStateName",
Account.GetValue("State").String, ValType.String)).String)
#Else
VID_PutValueSafe(Person, "UID_DialogCountry",
Account.GetValue("UID_ADSCountryCode").String)
                VID_PutValueSafe(Person, "UID_DialogState",
Connection.GetSingleProperty("DialogState", "UID_DialogState",
f.Comparison("NationalStateName",
Account.GetValue("State").String, ValType.String)).String)
#End If

                VID_PutValueSafe(Person, "Street",
Account.GetValue("StreetAddress").String)
                VID_PutValueSafe(Person, "Title",
Account.GetValue("PersonalTitle").String)
                VID_PutValueSafe(Person, "UID_Locality",
Connection.GetSingleProperty("Locality", "UID_Locality",
f.Comparison("Ident_Locality",
Account.GetValue("Locality").String, ValType.String)).String)
                VID_PutValueSafe(Person, "UID_Department",
Connection.GetSingleProperty("Department", "UID_Department",
```

## Програмна реалізація запису користувачем

```

Public Sub CCC_Create_New_Enable_Accounts_Request(ByVal UID_Org
As String, ByVal OrderReason As String, ByVal OrderDetail1 As
String)
    Dim colPersons As IEntityCollection
    Dim strUIDPerson As String
    Dim strUIDPersonHead As String
    Dim strCentralAccount As String
    Dim strUID_SCO As String
    Dim f As ISqlFormatter = Connection.SqlFormatter ' For
WHERE clause generation

    Dim strUIDPersonIDM =
Session.Config.GetConfigParm("Custom\Request\IDMUser")

    Dim qPerson = Query.From("Person") _
.Where(f.AndRelation(
    f.Comparison("IsActive", True, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("CCC_IsManualDeactivated", False, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("EntryDate", Date.UtcNow, ValType.Date,
CompareOperator.LowerOrEqual, FormatterOptions.None),
    f.Comparison("EntryDate", DbVal.MinDate, ValType.Date,
CompareOperator.GreaterThan, FormatterOptions.None),
    f.OrRelation(f.Comparison("ExitDate", "", ValType.Date,
CompareOperator.Equal, FormatterOptions.ConvertNull),
    f.Comparison("ExitDate", Date.UtcNow,
ValType.Date, CompareOperator.GreaterThan,
FormatterOptions.ConvertNull)),
    f.Comparison("CustomProperty10", "1", ValType.String,
CompareOperator.NotEqual, FormatterOptions.None),
    f.OrRelation(f.Comparison("CCC_EmpStatus", "EMP",
ValType.String, CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("CCC_EmpStatus", "DFT",
ValType.String, CompareOperator.Equal, FormatterOptions.None),
    f.Comparison("CCC_EmpStatus", "TEM",
ValType.String, CompareOperator.Equal, FormatterOptions.None)),
    f.Comparison("CCC_Deleted", False, ValType.Bool,
CompareOperator.Equal, FormatterOptions.None))) _

```

```
.SelectAll()  
  
    ' Create a collection of Person entries  
    colPersons = Session.Source.GetCollection(qPerson,  
EntityCollectionLoadType.Bulk)  
  
    Using uow = Session.StartUnitOfWork()  
  
        Dim Bag As New PropertyBag()  
        ' Run through the list of Person entries  
        For Each colElement As IEntity In colPersons  
  
            ' Get the values directly from the collection  
            element without creating a full object  
            strUIDPerson =  
colElement.GetValue("UID_Person").String  
            strUIDPersonHead =  
colElement.GetValue("UID_PersonHead").String  
            strCentralAccount =  
colElement.GetValue("CentralAccount").String  
            If String.IsNullOrEmpty(strUIDPersonHead) Or  
String.IsNullOrEmpty(strCentralAccount) Then
```

## Програмна створення таблиці та заповнення бази

```

-- USER SQL
ALTER USER "BI4INTF"
DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP"
ACCOUNT UNLOCK ;
-- CREATE TABLE
CREATE TABLE BI4INTF.ZAQ1WSX
(
  "LOGIN" VARCHAR2(255 BYTE),
  "FIRST_NAME" VARCHAR2(100 BYTE) NOT NULL ENABLE,
  "LAST_NAME" VARCHAR2(100 BYTE),
  "MIDDLE_NAME" VARCHAR2(40 BYTE),
  "EMAIL" VARCHAR2(80 BYTE) NOT NULL ENABLE,
  "DEALER_KEY" VARCHAR2(40 BYTE),
  "SP_KEY" VARCHAR2(200 BYTE),
  "PHONE_NUMBER" VARCHAR2(20 BYTE)
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255
NOCOMPRESS LOGGING
STORAGE(INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1
BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE DEFAULT)
TABLESPACE "USERS" ;

INSERT ALL
      INTO BI4INTF.DEALERS_FOR_IDM
(LOGIN,FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER)
VALUES ('DR_GO_E_LOSHMANOV', 'Євген', 'Лошманов', 'Владиславович', 'na@na.ua', 'DR-GO', 'Дейра Україна
- Dealer', '380963201712')
      INTO BI4INTF.DEALERS_FOR_IDM
(LOGIN,FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER)
VALUES ('VP_GO_M_PURAVETS', 'Михайло', 'Пуравець', 'Васильович', 'ocean_plaza@ttt.ua', 'VP-GO', 'СВП
Плюс - Dealer', NULL)
      INTO BI4INTF.DEALERS_FOR_IDM
(LOGIN,FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER)
VALUES ('VP_GO_O_MIRKOVSKY', 'Олександр', 'Мірко́вський', 'Миколайович', 'ttkiev43@ttt.ua', 'VP-GO', 'СВП
Плюс - Dealer', '380681165454')
      INTO BI4INTF.DEALERS_FOR_IDM
(LOGIN,FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER)
VALUES ('VP_GO_E_SUPRUNENKO', 'Олена', 'Супруненко', 'Олександрівна', 'romni@ttt.ua', '1-68', '54-
71', '380961947999')
      INTO BI4INTF.DEALERS_FOR_IDM
(LOGIN,FIRST_NAME, LAST_NAME, MIDDLE_NAME, EMAIL, DEALER_KEY, SP_KEY, PHONE_NUMBER)
VALUES ('BR_GO_E_OBOZNA', 'Катерина', 'Обозна', 'Сергіївна', 'BAL.OKT@ukr.net', '1-70', '57-
192', '380688576404')

```