МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ
Циклова комісія (кафедра) комп’ютерної інженерії та інформаційних технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему

СЕРВІС ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ КОНТРОЛЮ ОБЛІКУ

Виконав: студент групи 1П-21
Спеціальності
121 Інженерія програмного забезпечення
Костянтин ЯРОВЕНКО
Керівник:
Майя ЛЮТА

Черкаси 2025

АНОТАЦІЯ

Кваліфікаційна робота на тему «Сервіс для автоматизованої системи контролю обліку» складається з вступу, основної частини, що містить 3 розділи, висновку та списку використаних джерел. Загальний обсяг роботи – 52 сторінки. У роботі 28 рисунків. Перелік використаних ресурсів налічує 22 одиниці.

У роботі розглядається розробка сервісу для автоматизованої системи контролю обліку, що забезпечує централізоване збирання, обробку та зберігання облікових даних в організаціях різного типу. Основна мета полягає у створенні масштабованого, надійного та безпечного програмного рішення, яке автоматизує рутинні процеси збору та перевірки інформації, мінімізує вплив людського фактору та підвищує ефективність управлінських рішень.

Сервіс реалізовано з використанням сучасної архітектури клієнт-сервер, із RESTful API для взаємодії з іншими підсистемами. У проєкті застосовано реляційну базу даних для забезпечення структурованого зберігання облікової інформації, а також реалізовано модулі автентифікації, журналювання подій та контролю доступу.

Проведено аналіз вимог, побудовано діаграми варіантів використання та моделі даних, здійснено тестування ключових функціональних компонентів. Результати дослідження демонструють, що впровадження такого сервісу дозволяє значно зменшити витрати часу на ручний облік, підвищити точність даних і забезпечити прозорість процесів контролю.

Ключові слова: АВТОМАТИЗОВАНА СИСТЕМА, КОНТРОЛЬ ОБЛІКУ, ПРОГРАМНИЙ СЕРВІС, RESTFUL API, БАЗА ДАНИХ, ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ІНФОРМАЦІЙНА СИСТЕМА, УПРАВЛІННЯ ДАНИМИ.

ABSTRACT

The qualification work on the topic "Service for an automated accounting control system" consists of an introduction, the main part, which contains 3 sections, a conclusion and a list of sources used. The total volume of the work is 52 pages. The work contains 28 figures. The list of resources used has 22 units.

The work considers the development of a service for an automated accounting control system, which provides centralized collection, processing and storage of accounting data in organizations of various types. The main goal is to create a scalable, reliable and secure software solution that automates routine processes of information collection and verification, minimizes the impact of the human factor and increases the efficiency of management decisions.

The service is implemented using modern client-server architecture, with RESTful API for interaction with other subsystems. The project uses a relational database to ensure structured storage of accounting information, and also implements authentication, event logging and access control modules.

Requirements analysis was conducted, use case diagrams and data models were built, and key functional components were tested. The results of the study demonstrate that the implementation of such a service allows to significantly reduce the time spent on manual accounting, increase data accuracy, and ensure transparency of control processes.

Keywords: AUTOMATED SYSTEM, ACCOUNTING CONTROL, SOFTWARE SERVICE, RESTFUL API, DATABASE, SOFTWARE ENGINEERING, INFORMATION SYSTEM, DATA MANAGEMENT.

ЗМІСТ

ВСТУП 4
РОЗДІЛ 1 ОГЛЯД ТА АНАЛІЗ ЗАСОБІВ ПРОЄКТУВАННЯ ТА РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ 6
1.1 Область застосування систем інвентаризації техніки на основі хмарних технологій 6
1.2 Аналіз існуючих систем інвентаризації техніки на основі хмарних технологій 10
1.2.1 Аналіз результатів пошуку аналогів систем обліку програмних і технічних засобів фірми 12
1.3 Постановка задачі на розробку 12
Висновки до розділу 1 13
РОЗДІЛ 2 АНАЛІЗ І ПРОЄКТУВАННЯ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ 14
2.1 Виявлення та аналіз вимог до системи інвентаризації техніки на основі хмарних технологій 14
2.1.1 Моделювання предметної області системи інвентаризації техніки на основі хмарних технологій 15
2.1.2 Формування та аналіз вимог до системи інвентаризації техніки на основі хмарних технологій 16
2.1.3 Формування вимог за допомогою діаграми прецедентів системи інвентаризації техніки на основі хмарних технологій 17
2.1.4 Аналіз вимог до системи інвентаризації техніки за допомогою діаграм послідовності 19
2.2 Архітектурне проєктування системи інвентаризації техніки на основі хмарних технологій 21
2.2.1 Діаграма пакетів системи інвентаризації техніки на основі хмарних технологій 22
2.3 Детальне проєктування системи інвентаризації техніки на основі хмарних технологій 23
2.3.1 Проєктування пакету роботи з виводом інформації системи інвентаризації техніки на основі хмарних технологій 23
2.3.2 Проєктування пакету роботи з базою даних 24
2.3.3 Проєктування пакету роботи з системою інвентаризації техніки на основі хмарних технологій 25
2.3.4 Діаграма класів системи інвентаризації техніки на основі хмарних технологій 26
2.3.5 Діаграма компонентів системи інвентаризації техніки на основі

хмарних технологій 28
2.4 Розгортання системи інвентаризації техніки на основі хмарних технологій 29
2.4.1 Діаграма діяльності системи інвентаризації на основі хмарних технологій 29
2.4.2 Діаграма послідовності системи інвентаризації техніки на основі хмарних технологій 30
2.5 Об’єктно-орієнтоване проєктування системи інвентаризації техніки на основі хмарних технологій 31
Висновки до розділу2 32
РОЗДІЛ 3 РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ 33
3.1 Розробка системи інвентаризації на основі хмарних технологій 33
3.1.1 Обґрунтування вибору засобів реалізації системи інвентаризації техніки на основі хмарних технологій 33
3.1.2 Опис структурної схеми системи інвентаризації техніки на основі хмарних технологій 35
3.1.3 Розгортання програмного засобу для навчальної демонстрації різних способів знаходження визначника матриці на апаратних засобах 37
3.2 Структурна схема системи інвентаризації техніки на основі хмарних технологій 38
3.3 Розробка інтерфейсів системи інвентаризації техніки на основі хмарних технологій 39
3.3.1 Розробка головної сторінки системи інвентаризації техніки на основі хмарних технологій 40
3.3.2 Розробка сторінки серверів системи інвентаризації техніки на основі хмарних технологій 41
3.3.3 Розробка сторінки списку техніки системи інвентаризації техніки на основі хмарних технологій 43
3.3.4 Розробка сторінки додавання техніки системи інвентаризації техніки на основі хмарних технологій 44
3.3.5 Розробка сторінки додавання типу доступу користувача системи інвентаризації техніки на основі хмарних технологій 46
3.4 Тестування системи інвентаризації техніки на основі хмарних технологій 47
Висновки до розділу 3 48
ВИСНОВКИ 49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 51

ВСТУП

Важко уявити сучасне життя без комп'ютерів, а тим більше важко уявити без них роботу підприємств. Проте комп'ютер – це не тільки досягнення цивілізації, але й об'єкт обліку. У роботі розглянемо найбільш актуальні запитання про облік комп'ютера і пов'язаних із ним пристроїв.

Актуальність теми. Так чи інакше, великій кількості компаній доводиться стикатися з завданням обліку комп'ютерної техніки, програмного забезпечення та іншого обладнання. Актуальність розробки такої автоматизованої системи обумовлена тим, що в процесі функціонування фірми, вирішується безліч рутинних завдань, при виконанні яких формується документація довгострокового зберігання, робота з якими можлива тільки в рамках автоматизованої інформаційної системи.

Існує кілька способів вирішення цієї проблеми.

Варіант перший – використовувати для обліку можливості офісних додатків. Наприклад, вести таблицю в MS Excel або іншому табличному редакторі.

Переваги: тривіальне рішення, що потребує лише навичок у використанні табличного редактора. Мінуси: необхідність вводити всі дані вручну, труднощі при оновленні, відстеження змін в конфігурації обладнання.

Варіант другий – скористатися програмою для автоматичного збору інформації про апаратної конфігурації комп'ютерів з подальшим занесенням в базу даних.

Цей варіант має наступні переваги: збір конфігурації проводиться автоматично і вимагає лише невеликого редагування надалі. Зберігання інформації в форматі бази даних значно полегшує подальшу побудову звітів. Автоматичне оновлення інформації про зміни комп'ютерів дозволяє підвищити оперативність при роботі з базою даних.

Об'єктом розробки є інформаційна система інвентаризації техніки на основі хмарних технологій.

Предметом розробки є методи та засоби розробки корпоративних систем інвентаризації даних.

Метою кваліфікаційної роботи є розробка автоматизованої системи обліку і контролю обслуговування технічних і програмних засобів.

Відповідно до поставленої мети в роботі визначені наступні **завдання**:

1) Проведення аналізу предметної області.

2) Виявлення вимог користувачів до інформаційної системи.

3) Розробка структури інформаційної системи і виділення основних функціональних блоків.

4) Проєктування структури бази даних (інфологічне, даталогічне і фізичне моделювання);

5) Розробка інтерфейсу користувача;

6) Програмна реалізація автоматизованої інформаційної системи.

Практичне значення одержаних результатів. Практична значущість роботи полягає в розробці алгоритмів та їх програмної реалізації для збереження цілісності інформації з функціями підвищеної стійкості для обліку програмних і технічних засобів фірми.

РОЗДІЛ 1
ОГЛЯД ТА АНАЛІЗ ЗАСОБІВ ПРОЄКТУВАННЯ ТА РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ

1.1 Область застосування систем інвентаризації техніки на основі хмарних технологій

Інформаційна система інвентаризації (ІСІ) – це сукупність інформації, апаратно-програмних і технологічних засобів, засобів телекомунікації, баз і банків даних, методів і процедур, персоналу управління, які реалізують функції збору, оброблення, нагромадження та оброблення інформації для підготовки та прийняття ефективних управлінських рішень.

ІСО – це множина різних елементів та зв’язків між ними, що складають систему в цілому.

У сучасній концепції організації інформаційних систем у різних предметних сферах виокремлюють дві частини: забезпечувальну та функціональну. Кожна з них у свою чергу складається з підсистем.

Забезпечувальна частина ІС охоплює підсистеми, що реалізують технологію автоматизованого оброблення інформації.

Склад цих підсистем однорідний у різних інформаційних системах і відповідно до Державного стандарту охоплює: інформаційне, технічне, програмне, математичне, організаційне, правове забезпечення.

Інформаційне забезпечення (ІЗ). Інформація формується в результаті оброблення даних і є важливим елементом комп’ютерних інформаційних систем.

Організація інформаційного забезпечення в інформаційних системах має особливе значення.

Система інформаційного забезпечення передбачає створення єдиного інформаційного фонду, систематизацію та уніфікацію показників і документів, розроблення засобів формалізованого опису даних тощо.

Інформаційне забезпечення охоплює:

1) методичні та інструктивні документи.

2) єдину систему класифікації та кодування.

3) інформаційну базу, яка у свою чергу поділяється на нормативно-довідкові документи, інформаційні повідомлення, інформаційні масиви.

Технічне забезпечення. Технічні засоби служать основою побудови ІС. Потужність засобів значною мірою визначає склад задач, що розв'язуються в даній предметній сфері. До технічного забезпечення ІС належать комп'ютерна техніка, засоби комунікації та оргтехніка. Іншими словами, технічне забезпечення – це комплекс взаємопов'язаних технічних засобів, призначених для збирання, нагромадження, оброблення, передавання, обміну та відображення інформації, потрібної для управління системою.

Технічне забезпечення сучасних інформаційних систем – це комплекс різних видів техніки: обчислювальна техніка, периферійні пристрої, засоби автоматичного зчитування даних, офісне обладнання, комунікаційне обладнання, засоби передачі та обміну даними, комунікаційне обладнання, мережеве обладнання, засоби мультимедіа тощо.

Програмне забезпечення (ПЗ) – це сукупність програм, які реалізують мету й завдання інформаційної системи та забезпечують функціонування технічних засобів системи.

Програмне забезпечення охоплює сукупність програм, що реалізують функції та задачі автоматизованих інформаційних технологій і забезпечують стабільну роботу комплексів технічних засобів.

До складу ПЗ входять загальносистемні та спеціальні програми.

До загальносистемного програмного забезпечення належать програми, розраховані на широке коло користувачів і призначені для організації

облікового процесу та для розв’язання задач оброблення інформації, які часто зустрічаються.

Спеціальне програмне забезпечення – це сукупність програм, які розробляються для створення інформаційних технологій конкретного функціонального призначення.

Охоплює пакети прикладних програм (ППП), які здійснюють організацію даних і їх оброблення у процесі розв’язання функціональних задач.

Математичне забезпечення – це сукупність економіко-математичних методів, моделей та алгоритмів оброблення інформації, які є в інформаційній системі.

Організаційне забезпечення передбачає власний апарат управління, що забезпечує функціонування всіх його підсистем як єдиного цілого. Як правило, персонал ІС складають працівники відділу розроблення, відділу впровадження та супроводу нових програм, відділу експлуатації.

У теорії комп’ютерних систем оброблення інформації в інформаційному забезпеченні виокремлюють:

1) зовнішнє інформаційне забезпечення, яке охоплює систему показників даної предметної сфери, систему класифікацій, первинні документи;

2) внутрішнє забезпечення, що охоплює інформаційну базу даних на машинних носіях.

Правове забезпечення – це сукупність норм, виражених у нормативних актах, які встановлюють і закріплюють організацію цих систем, їх цілі, завдання, структуру, функції та правовий статус ІС.

Правове забезпечення на етапі розроблення автоматизованих інформаційних систем та інформаційних технологій охоплює нормативні акти, пов’язані з договірними взаєминами розробника й замовника у процесі створення ІС та ІТ, із правовим регулюванням різних відхилень у ході цього процесу, а також зумовлені потребою забезпечення процесу розроблення ІС і ІТ різними видами ресурсів.

Економічне забезпечення – сукупність методів і засобів, призначених для створення оптимальних умов ефективної та безпомилкової діяльності людини в інформаційній системі та її освоєння.

Функціональна частина інформаційної системи реалізує розв’язання задач предметної сфери.

Підсистема – це сукупність компонентів системи, виокремлених за певною ознакою. Кожна функціональна підсистема має свій склад комплексів задач, призначений для реалізації функцій управління.

Основні принципи виокремлення самостійних функціональних підсистем (комплексів задач):

1) відносна самостійність кожної з них;

2) наявність відповідного набору функцій і функціональних задач із чітко виявленою локальною ціллю функціонування;

3) мінімізація складу елементів, що входять у підсистему.

В основі інформаційної системи інвентаризації лежать розв’язання облікових задач, об’єднаних у комплекси, які виконують окремі ділянки обліку. Комплекс задач характеризується визначеним економічним змістом, веденням затверджених синтетичних рахунків, первинними та зведеними документами, взаємопов’язаними алгоритмами розрахунків, а також методичними матеріалами й нормативними документами конкретної ділянки обліку.

Аналіз предметної області – це початковий крок етапу системного аналізу, з якого починається розробка програмного продукту.

Предметна область надзвичайно сильно впливає на всі аспекти проєкту: взаємодія з користувачем, вимоги до системи, використана для зберігання даних модель, реалізація і багато іншого.

Аналіз предметної області проводиться для виділення її сутностей, визначення первинних вимог до функціональності і визначення меж проєкту.

1.2 Аналіз існуючих систем інвентаризації техніки на основі хмарних технологій

Щоб створити будь-яку складну систему, розробники повинні звернути увагу на уже існуючі рішення у цій галузі, аби зрозуміти, які рішення є вдалими, а які – ні. Тому для створення власної інформаційної системи інвентаризації (ІСІ) програмних і технічних засобів проаналізуємо існуючі системи.

Програма для інвентаризації комп'ютерів та мережевого устаткування – Total Network Inventory 3.

Система Total Network Inventory від Softinventive дозволяє вести повний облік програмного забезпечення, встановленого на комп'ютері, а також здійснювати контроль апаратної частини комп'ютерів офісів, малих і великих корпоративних локальних мереж. Адміністратор може просканувати мережу і отримати вичерпну інформацію про кожному комп'ютеру. Програма дозволяє переглядати мережу як в реальному часі (негайно), так і, використовуючи скрипт, в момент підключення клієнта до домену.

У багатьох організаціях, які постійно розширюють парк робочих ПК, гостро постає питання про необхідність ведення моніторингу систем, а також інвентаризації техніки. Фахівці повинні постійно контролювати список встановлених програм, зміна основних конфігурацій, а ручний опис стану кожного комп'ютера іноді буває просто неможлива. Для оптимізації процесу ведення обліку була розроблена спеціальна програма інвентаризації комп'ютерів в мережі – Total network inventory 3. Немає необхідності проводити підготовку перед початком роботи і стартом процесу сканування, а набір заданих параметрів може бути мінімальним.

Для запуску майстра сканування потрібна установка програмного забезпечення тільки на один комп'ютер. Перед стартом збору інформації необхідно лише вказати список мережевих вузлів, ввести логін з паролем - і можна починати. За кілька хвилин ви зможете провести віддалене сканування

по мережі і зібрати дані про всі пристрої, які можуть перебувати під управлінням різних операційних систем. Вся інформація про кожному комп'ютері з усієї мережі, яким відведені в пам'яті окремі файли, буде зберігатися на жорсткому диску. Вам надається можливість не тільки спостерігати за зібраними даними, але і самостійно організовувати систему ведення обліку: рухайте створені файли, об'єднуйте їх в групи і присвоюйте інвентарні номери. Крім того, програма обліку комп'ютерів Total network inventory дозволяє формувати стандартні мережеві звіти і самостійно конструювати форми, які будуть містити тільки найнеобхідніші відомості.

Завдяки зведеннями з вузькими параметрами ви без проблем можете відстежувати встановлене програмне забезпечення, а також ліцензії на їх використання. Крім того, програми для моніторингу системи комп'ютера допомагають виявити нерозумне використання технічних ресурсів і визначити обладнання, на якому встановлені застарілі компоненти.

Сповіщення повідомлять про вразливі місця мережі, а також про комп'ютери, на яких антивіруси були встановлені або застаріли. З вищеописаного матеріалу випливає, що інвентаризація комп'ютерів - досить просте завдання, яка виконується в кілька кліків миші, потрібно лише купити описану програму. Зовнішній вигляд продукту представлений на рисунку 1.1.

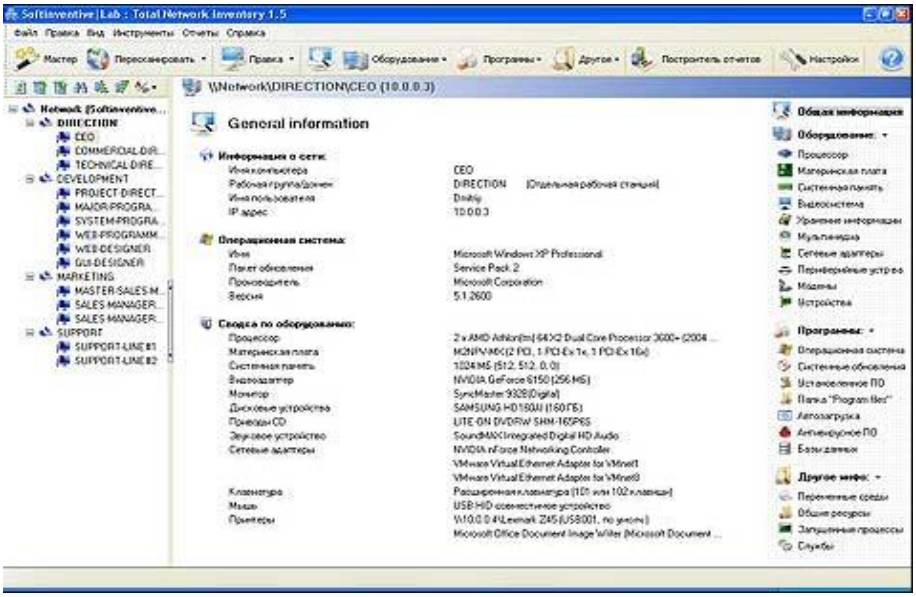

Рисунок 1.1 – Зовнішній вигляд системи «Total Network Inventory 3»

1.2.1 Аналіз результатів пошуку аналогів систем обліку програмних і технічних засобів фірми

На сьогодні не існує аналогів як web-додатків так і desktop-application для обліку програмних і технічних засобів фірми. Був знайдений подібний продукт для обліку встановлених програмних засобів. Для реалізації функціональної і якісної системи, необхідно розглянути та проаналізувати цей програмний аналог.

1.3 Постановка задачі на розробку

Система інвентаризації і обслуговування технічних і програмних засобів виконує наступні функції:

1) облік технічних і програмних засобів, які закріплені за працівниками;

2) проведення інвентаризації технічних і програмних засобів;

3) постановкою на облік нового обладнання;

4) переміщення техніки по офісам.

Виходячи з функцій, які виконує система, доцільним є створення автоматизованої інформаційної системи, що виконує наступні завдання:

1) автоматизація обліку технічних і програмних засобів, які закріплені за працівниками;

2) автоматизація інвентаризації;

3) автоматизація постановки на облік нового обладнання;

4) автоматизація переміщення техніки по офісам;

5) вхідні/вихідні данні;

6) формати даних;

7) регламентація роботи;

8) кросплатформеність.

Висновки до розділу 1

У першому розділі був проведений опис та аналіз предметної області – процес і, сформовано постановку завдання на розробку. Провівши аналіз предметної області було виділено мету та задачі системи. Реалізована система має відповідати вимогам, що описані в першому розділі.

Було проведено пошук і аналіз аналогів. Огляд програмних аналогів дав можливість спланувати коректне і вдале проєктування та реалізацію системи інвентаризації техніки на основі хмарних технологій.

РОЗДІЛ 2
АНАЛІЗ І ПРОЄКТУВАННЯ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ

2.1 Виявлення та аналіз вимог до системи інвентаризації техніки на основі хмарних технологій

Системний підхід став домінуючим при проєктуванні сучасних ІС.

Проєктування ІС – процес, спрямований на вдосконалення економічної ІС об’єкта управління, що передбачає створення та впровадження комплексного розв’язання задач із застосуванням сучасних електронних обчислювальних машин і технічних засобів управління об’єктом. Системний підхід до проєктування ІС передбачає вивчення, розгляд, опис певного об’єкта у повному обсязі з урахуванням його істотних властивостей. Головна мета підходу полягає в дотриманні таких вихідних засад:

– всебічне та цілісне оцінювання динамічних характеристик об’єкта;

– його взаємозв’язки із зовнішнім середовищем;

– урахування можливих зовнішніх і внутрішніх несприятливих умов, що можуть вивести об’єкт зі стану рівноваги.

Загальнотеоретичні засади проєктування автоматизованих ІС. Загальнотеоретичні засади проєктування автоматизованих інформаційних систем. Таким об’єктом може бути будь-яка модель, що описує систему, процес чи певну сукупність об’єктів. Система як об’єкт має складну внутрішню структуру. Наприклад це може бути підприємство, науково-дослідні та проєктні організації, виробничі процеси тощо.

Для загального представлення потрібно сформувати функції які потрібні користувачеві у вигляді користувацьких вимог до програмного засобу для демонстрації різних способів знаходження визначника матриці.

2.1.1 Моделювання предметної області системи інвентаризації техніки на основі хмарних технологій

Створення будь-якої складної системи неможливо уявити без попереднього створення її моделі. Підбір проєктних рішень, визначення функціональних обмежень та можливостей реальної системи краще спочатку підібрати на моделі, так як це дозволяє створити чітку структуру продукту, не зосереджуючись на особливостях реалізації того чи іншого алгоритму.

Систем моделювання різноманітних процесів безліч: для технологічних процесів використовують спеціальні інтегровані системи з 3D моделюванням, для математичних моделей спеціалізовані математичні пакети ПЗ і так далі. Моделювання як таке застосовують скрізь і часто у неявному вигляді.

Для моделювання предметної області використовують різні технології візуалізації інформаційних потоків та їх обробників. Однією з таких технологій є UML діаграми. UML-діаграма класів дозволить приблизно зрозуміти, яким чином повинна працювати майбутня система. Саме тому була сформована початкова діаграма класів, яка зображена на рисунку 2.1.

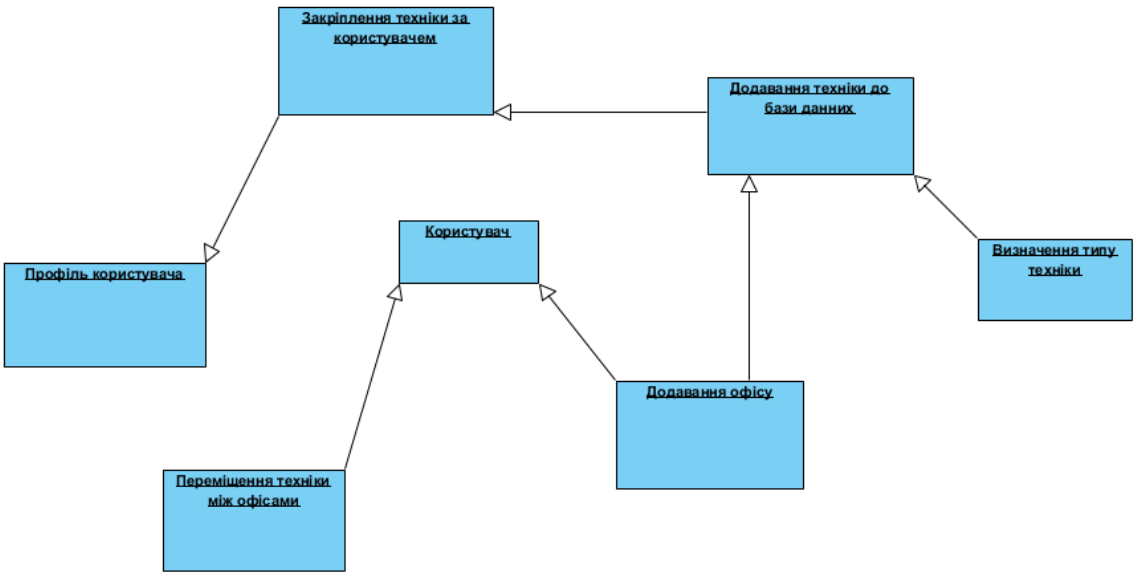

Рисунок 2.1 – Модель предметної області системи інвентаризації техніки на основі хмарних технологій

2.1.2 Формування та аналіз вимог до системи інвентаризації техніки на основі хмарних технологій

Під час формування системи, потрібно розуміти якою вона має бути система, та які функції система виконує. Тому необхідно детально сформувати та задокументувати вимоги.

Після чого необхідно реалізувати ряд діаграм, які будуть візуально демонструвати систему з різних поглядів.

Функціональні і не функціональні вимоги. Функціональні вимоги – це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: калькулювання даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система.

Так як точність і цілісність даних, що зберігатимуться у системі, важливі, необхідно передбачити усі варіанти обробки вхідних даних за умови впливу на систему зовнішніх факторів, введенні некоректних даних користувачем.

1) перевірки валідності введених даних;

2) перевірка заповнення користувачем всіх обов’язкових полів введення;

3) перевірка збереження даних при збоях системи;

4) перевірка правильності заповнення полів, в яких використовується маска для введення даних.

Нефункціональні вимоги – це програмного забезпечення, які задають критерії для оцінки якості його роботи. На відміну від функціональних вимог, які визначають що система повинна робити, нефункціональні вимоги визначають якою система повинна бути.

1) перевірка безпеки зв’язків між серверною частиною системи і частиною користувача;

2) перевірка зберігання даних;

3) перевірка проведення валідації;

4) обмеження доступу користувачів залежно від статусу.

Описані вимогу конкретизують дії системи, а отже спрощують подальшу розробку.

2.1.3 Формування вимог за допомогою діаграми прецедентів системи інвентаризації техніки на основі хмарних технологій

Діаграма прецедентів – в UML, діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Для демонстрації реалізації описаних вимог створено діаграму прецедентів.

Суть даної діаграми полягає в наступному: проєктована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Отже для ІСК мережею пошукових агентів. Отриману діаграму прецедентів можна розглянути на рисунку 2.2.

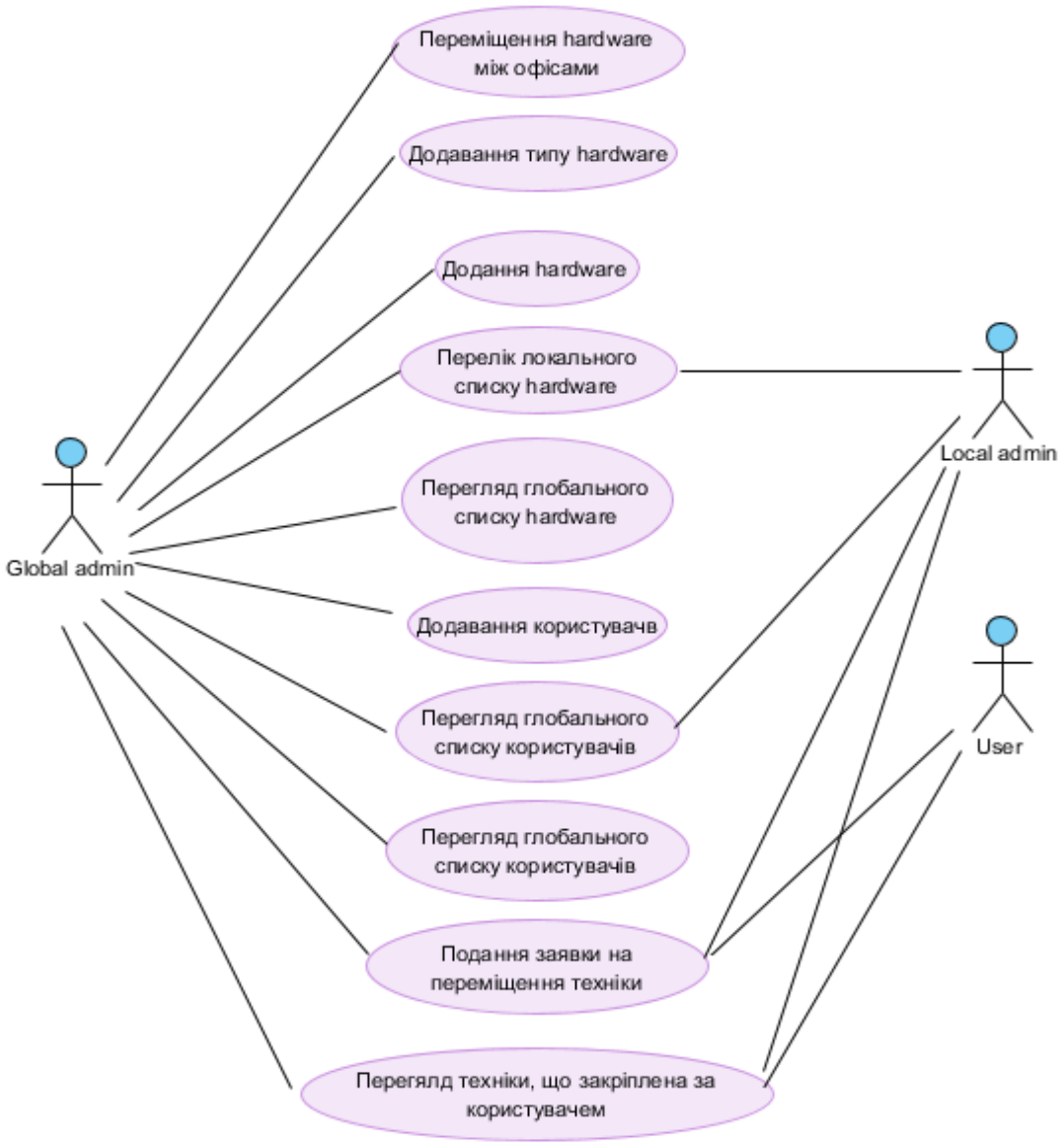

Рисунок 2.2 – Діаграма прецедентів системи автоматизації системи

На діаграмі розкриті наступні ролі:

1) Global Admin – головний адміністратор системи. Роль, якій доступні всі можливі дії в системі.

2) Local Admin – адміністратор окремого офісу. Роль, якій доступні лише локальні дані та дії окремого офісу, а також подача заявок на переміщення техніки між офісами.

3) User – працівник окремого офісу. Роль, якій доступний лише перегляд техніки, що закріплена за ним та подача заявок на переміщення техніки між офісами.

2.1.4 Аналіз вимог до системи інвентаризації техніки за допомогою діаграм послідовності

Діаграма додавання користувачем нового hardware. Звичайні користувачі не мають доступу до додавання нового hardware, тому першим кроком даної діяльності є перевірка рівня доступу користувача. Після перевірки можливості доступу користувач заповнює необхідні поля для створення нового hardware якщо йому був наданий доступ, або система сигналізує про проблеми з доступом до даної діяльності.

Діаграму діяльності описаного вище процесу зображено на рисунку 2.3.

Рисунок 2.3 – Діаграма додавання нової техніки до бази даних

Діаграма додавання користувачем нового типу hardware. Звичайні користувачі не мають доступу до додавання нового типу hardware, тому першим кроком даної діяльності є перевірка рівня доступу користувача. Після перевірки можливості доступу користувач заповнює необхідні поля для

створення нового типу hardware якщо йому був наданий доступ, або система сигналізує про проблеми з доступом до даної діяльності.

Діаграму діяльності описаного вище процесу зображено на рисунку 2.4.

Рисунок 2.4 – Діаграма додавання нового типу техніки

Діаграми перегляду користувачем локального та глобального списку користувачів та hardware схожі між собою, тому для прикладу додано діаграму перегляду користувачем локального списку hardware – рисунок 2.5. Доступ до перегляду даної інформації має локальний та глобальний адміністратор, тому першим кроком даної діяльності є перевірка рівня доступу користувача. Після перевірки можливості доступу відкривається вікно з списком локального hardware одного офісу, або система сигналізує про проблеми з доступом до даної діяльності.

Рисунок 2.5 – Діаграма локального списку техніки офісу

2.2 Архітектурне проєктування системи інвентаризації техніки на основі хмарних технологій

Для здійснення переходу від логічного представлення до реальної моделі, необхідно перетворити усі логічні сутності в матеріальні. Для детального опису архітектурного проєктування розглянуто компоненти з яких складається система, та яким чином вона працює.

2.2.1 Діаграма пакетів системи інвентаризації техніки на основі хмарних технологій

Для подальшого розуміння логіки проєкту необхідною складовою являється архітектурне проєктування. Таким чином буде сформовано представлення про структуру проєкту. Для цього було обрану діаграму пакетів.

В свою чергу діаграма пакетів уніфікованої мови моделювання відображає залежності між пакетами.

Невід'ємною частиною діаграми пакетів є пакет. Пакет – це елемент моделі, який в першу чергу використовується для групування інших елементів даної моделі. Елементи моделі, які входять у склад певного пакету, називаються членами пакету, а пакет вже володіє всіма своїми членами. Члени пакетів являються власністю пакетів, що означає що вони повністю належать їм, що дає змогу згрупувати функціонал проєкту, для уявлення структури.

Діаграма пакетів для системи інвентаризації техніки на основі хмарних технологій зображено на рисунку 2.6.

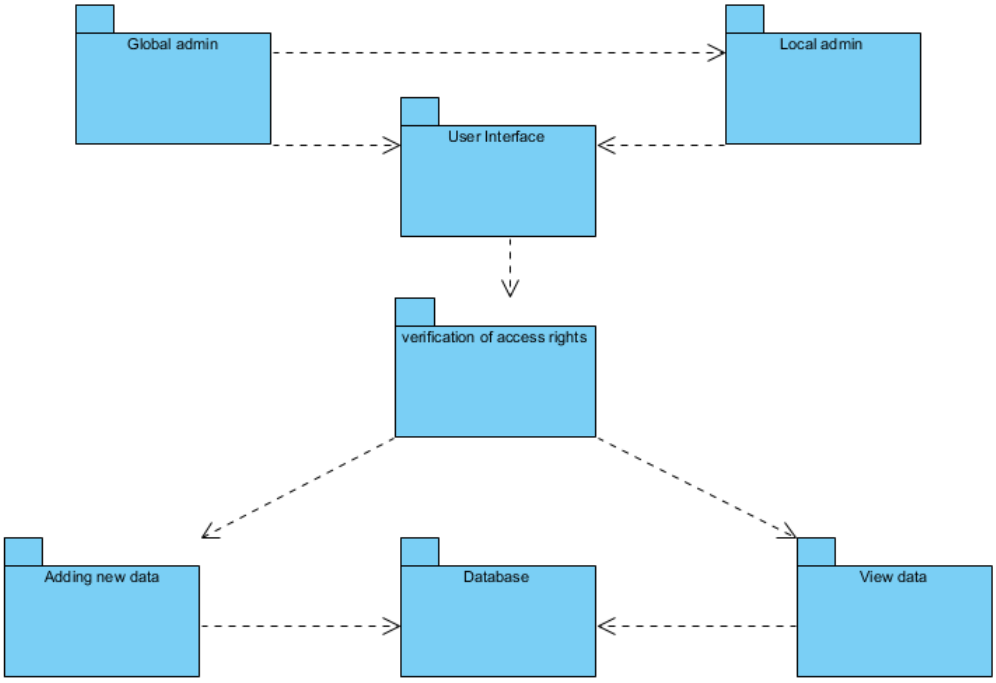

Рисунок 2.6. – Діаграма пакетів системи інвентаризації техніки на основі хмарних технологій

Дана діаграма демонструє залежність можливих дій користувача від рівня прав доступу. Є три основні пакети – інтерфейс користувача, перевірка рівня доступу і база даних. В свою чергу в системі описані три ролі користувачів, з своїм рівнем рівнем доступу і з переліком можливих дій з системою.

Наступним кроком архітектурного проєктування є опис кожного із пакетів. Для візуального представлення використаємо діаграми послідовності.

2.3 Детальне проєктування системи інвентаризації техніки на основі хмарних технологій

При детальному проєктування системи виникає необхідність деталізувати особливості алгоритмічної і логічної реалізації особливостей системи. Необхідно також детально описати кожен із пакетів програми, які були наведені вище. Для такого моделювання в мові UML використовуються діаграми діяльності, діаграми послідовності, діаграми комунікацій та діаграми кінцевого автомату.

2.3.1 Проєктування пакету роботи з виводом інформації системи інвентаризації техніки на основі хмарних технологій

Пакет роботи з виводом інформації є першим, з чим зустрічається користувач. Цей пакет передбачає вивід форм з інформацією, а також за відображення проміжних результатів у процесі знаходження визначника матриці. Діаграма послідовності пакету для роботи з виводом інформації зображена на рисунку 2.7. Як видно з рисунку, спочатку виконується перевірка прав доступу користувача, а потім, згідно результатів перевірки, користувачу виводиться головна сторінка системи з обмеженнями згідно його статусу. Потім користувач може виконати наступний крок роботи з системою, а пакет для роботи з виводом інформації виведе результат дії.

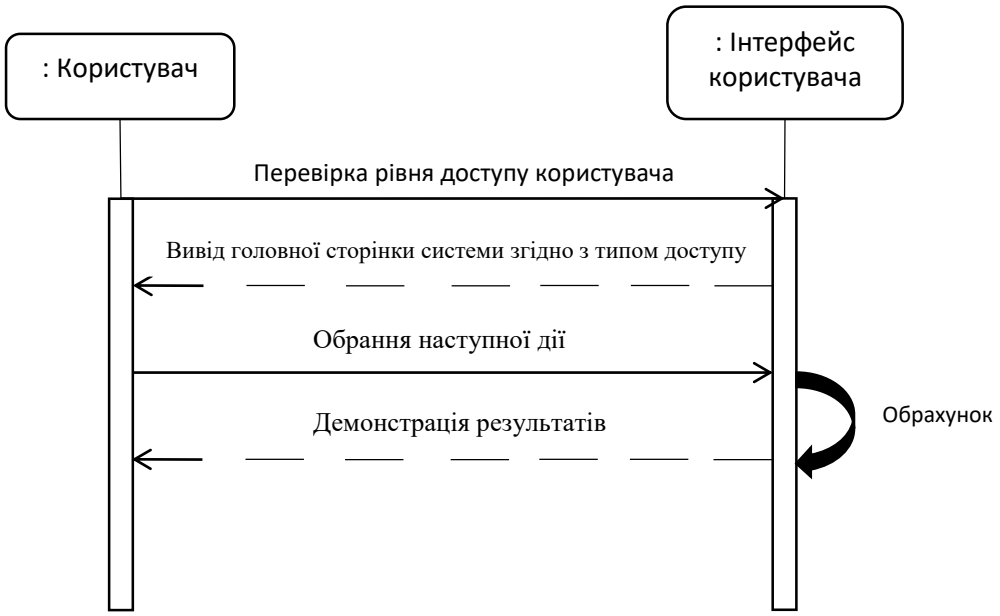

Рисунок 2.7 – Діаграма послідовності пакету для роботи з виводом інформації системи інвентаризації техніки на основі хмарних технологій

2.3.2 Проєктування пакету роботи з базою даних

Пакет роботи з базою даних відповідає за додавання та редагування техніки. Користувач з відповідними правами визначає яка дія має виконатись, натискаючи відповідну кнопку, після чого система відображає відповідне вікно додавання або редагування техніки. Діаграма послідовності пакету для роботи з принтером зображена на рисунку 2.8.

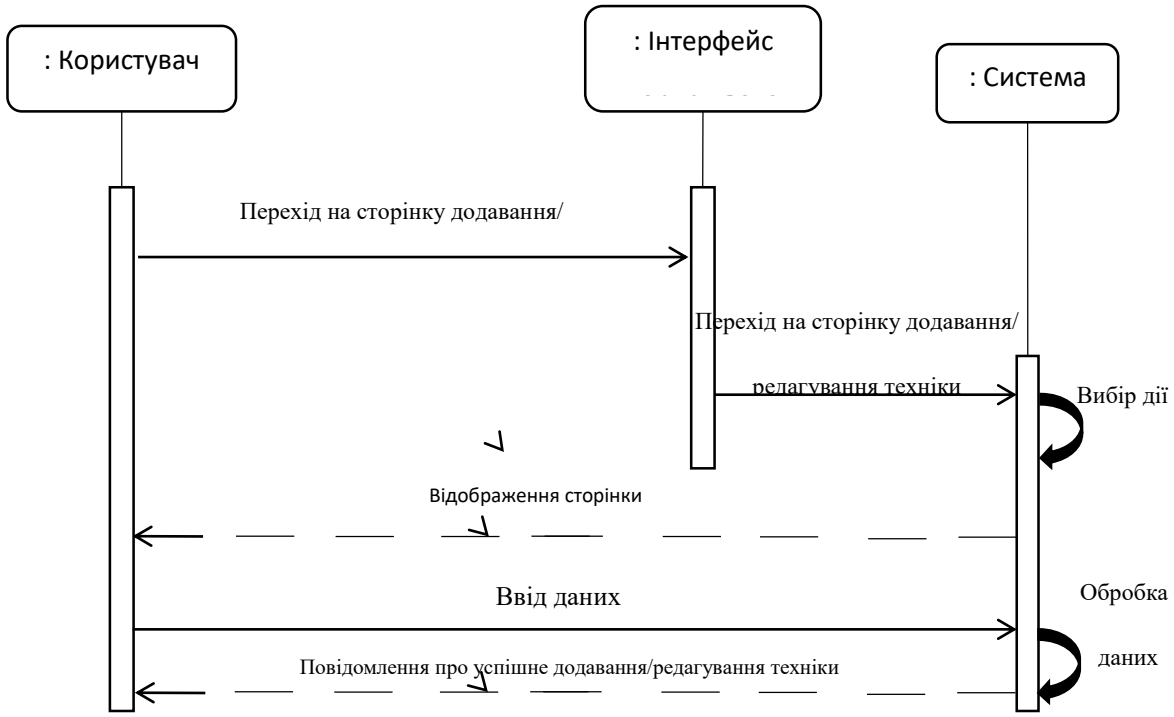

Рисунок 2.8 – Діаграма послідовності пакету для роботи з базою даних системи інвентаризації техніки на основі хмарних технологій

2.3.3 Проєктування пакету роботи з системою інвентаризації техніки на основі хмарних технологій

Пакет роботи з системою є основним пакетом програми. Він пов’язує у собі усі інші пакети. Можна сказати, що даний пакет повністю описує усю систему в цілому. Спочатку перевіряється рівень доступу користувача, потім відображення сторінок і в залежності від ролі, додавання або редагування списку техніки. На рисунку 2.9 зображено діаграму послідовності даного пакету.

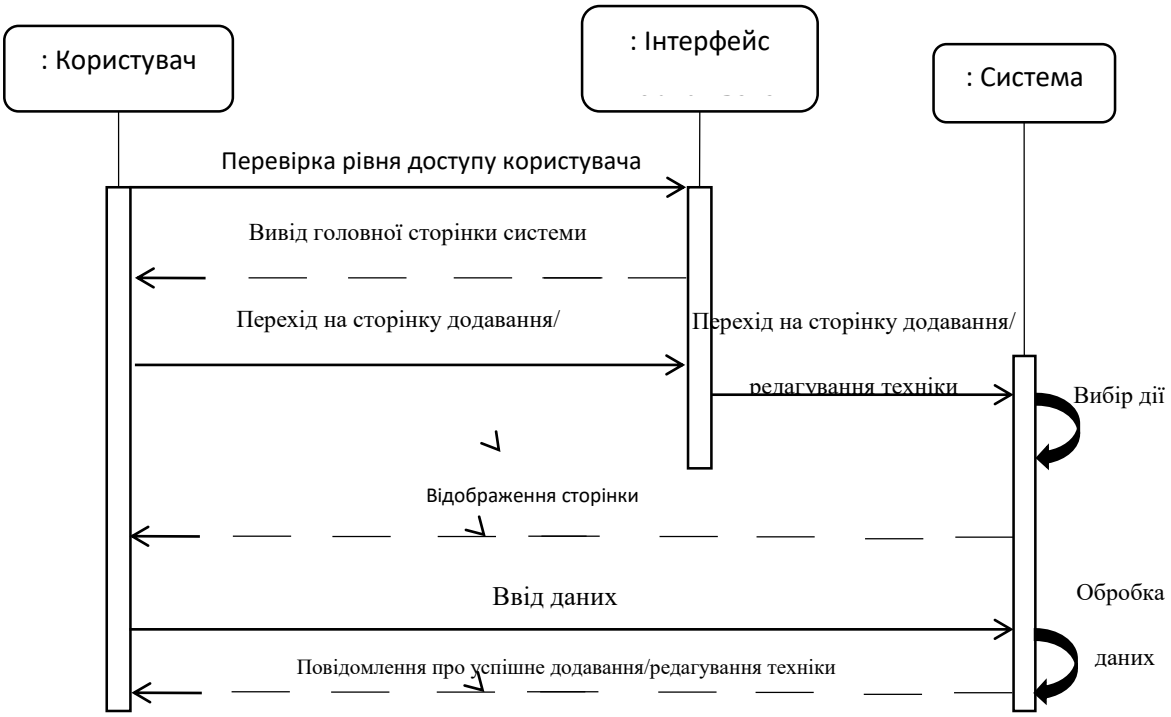

Рисунок 2.9 – Діаграма послідовності пакету роботи з системою інвентаризації техніки на основі хмарних технологій

2.3.4 Діаграма класів системи інвентаризації техніки на основі хмарних технологій

Діаграма класів спрямована на деталізацію для подальшої реалізації програми для навчальної демонстрації знаходження визначника матриці. Для створення даної діаграми потрібно виділити наступні основні сутності, які в свою чергу повністю дають змогу описати систему в цілому:

– користувач;
– сервер;
– техніка;
– офіс;
– права користувача.

Далі після виділення класів, детально опишемо кожен з них, для того щоб всі класи були максимально конкретним та інформаційним.

Поля класів програми, для демонстрації знаходження визначника матриці різними способами:

1. Користувач:
– повне ім’я;
– місто;
– тип користувача.
2. Офіс:
– місто.
3. Сервер:
– IP;
– назва сервера.
4. Техніка:
– тип техніки;
– назва.
5. Права користувача:
– місто;
– користувач;
– права.

Діаграма класів програмного продукту для навчальної демонстрації знаходження визначника матриці різними способами подана на рисунку3.10.

Клас «Hardware» відповідає за додавання нової техніки та інформації про неї.

Клас «Account» відповідає про інформацію користувача.

Клас «Server» відповідає за додавання, редагування і видалення серверів.

Клас «UserPermission» відповідає за зберігання інформації про роль користувача.

Клас «Location» відповідає за зберігання списку міст.

Така структура даних дає змогу просто і головне швидко маніпулювати з даними.

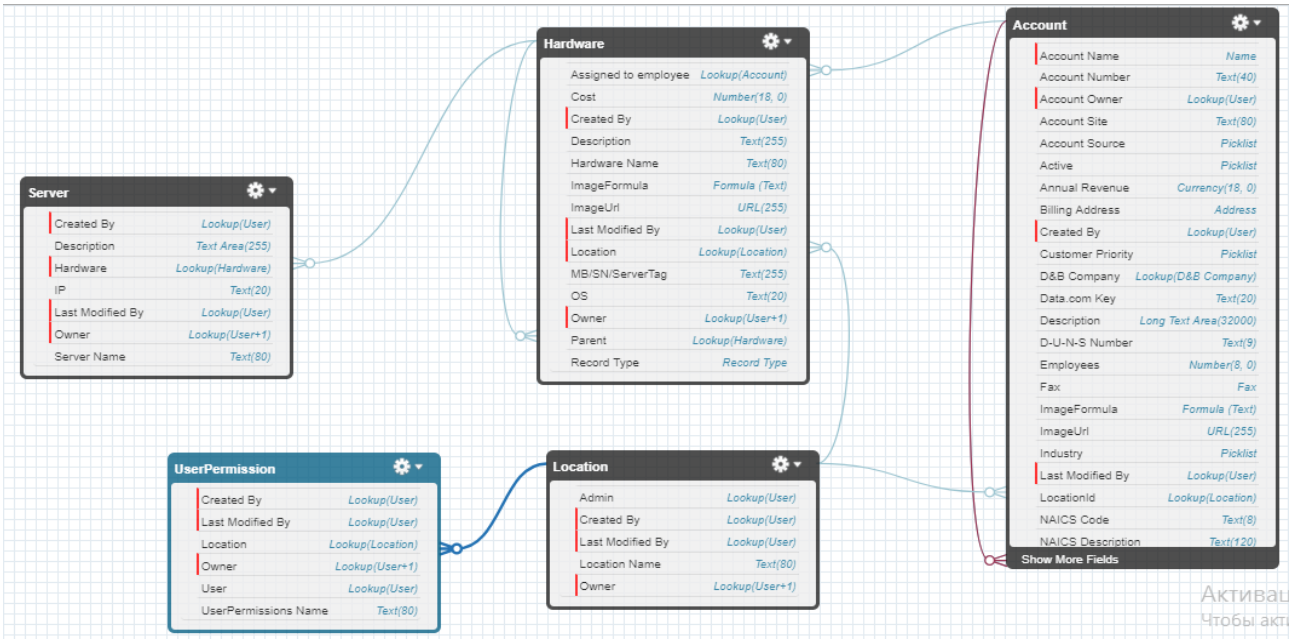

Рисунок 2.10 – Діаграма класів системи інвентаризації техніки на основі хмарних технологій

2.3.5 Діаграма компонентів системи інвентаризації техніки на основі хмарних технологій

Діаграма компонентів описує внутрішню структуру системи, яка складається з модулів, класів, та файлів.

Основні компоненти системи:

– сервер;
– компонент бази даних;
– компонент додавання користувачів;
– компонент додавання техніки;
– компонент виведення інформації на сторінки системи.

Сервер в свою чергу складається з процесів визначення рівня доступу та транзакцій.

Майже всі системи з’єднані з сервером, за допомогою процесу транзакцій. Це зумовлено тим, що всі запити проходять через сервер, який має доступ до бази даних. Після отримання необхідних даних сервер проводить необхідну обробку та повертає результат в необхідну систему.

Більша частина системи зв’язана з системою взаємодії з користувачем. Це зумовлено тим, що система взаємодії з користувачем є основною системою програми. Діаграма компонентів системи інвентаризації техніки на основі хмарних технологій – рисунок 2.11.

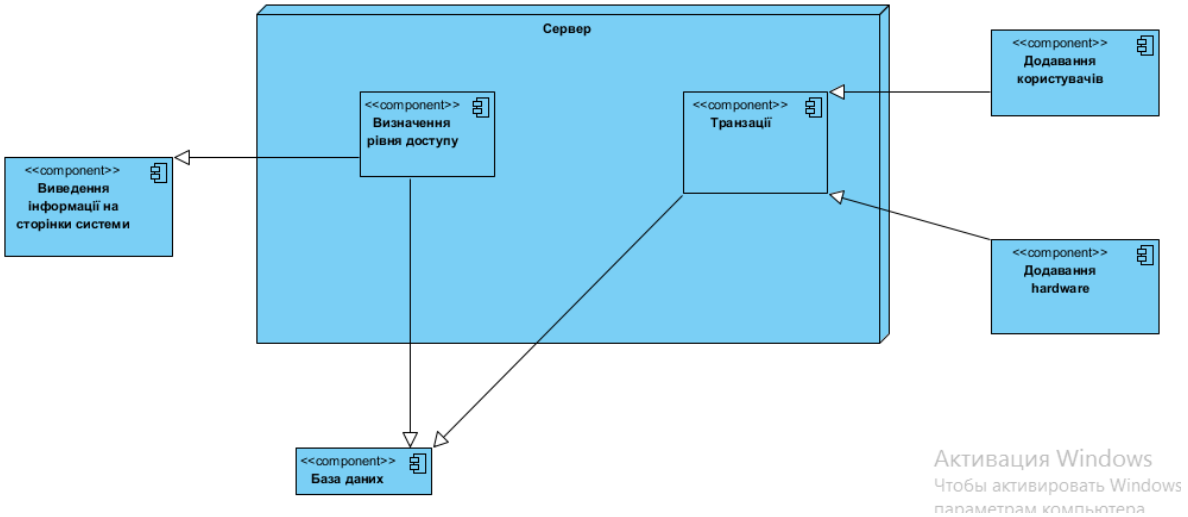

Рисунок 2.11 – Діаграма компонентів системи інвентаризації техніки на основі хмарних технологій

2.4 Розгортання системи інвентаризації техніки на основі хмарних технологій

2.4.1 Діаграма діяльності системи інвентаризації на основі хмарних технологій

Діаграма діяльності задає поведінку системи у вигляді певних робіт, які може виконувати система або актор і ці роботи можуть залежати або від заданих умов, або від обмежень. Ця діаграма відображає функціональну структуру системи і принципи поведінки її окремих елементів під час виконання відповідної діяльності.

Діаграма діяльності, що відображає поведінку проєктованої системи подана на рис. 2.12.

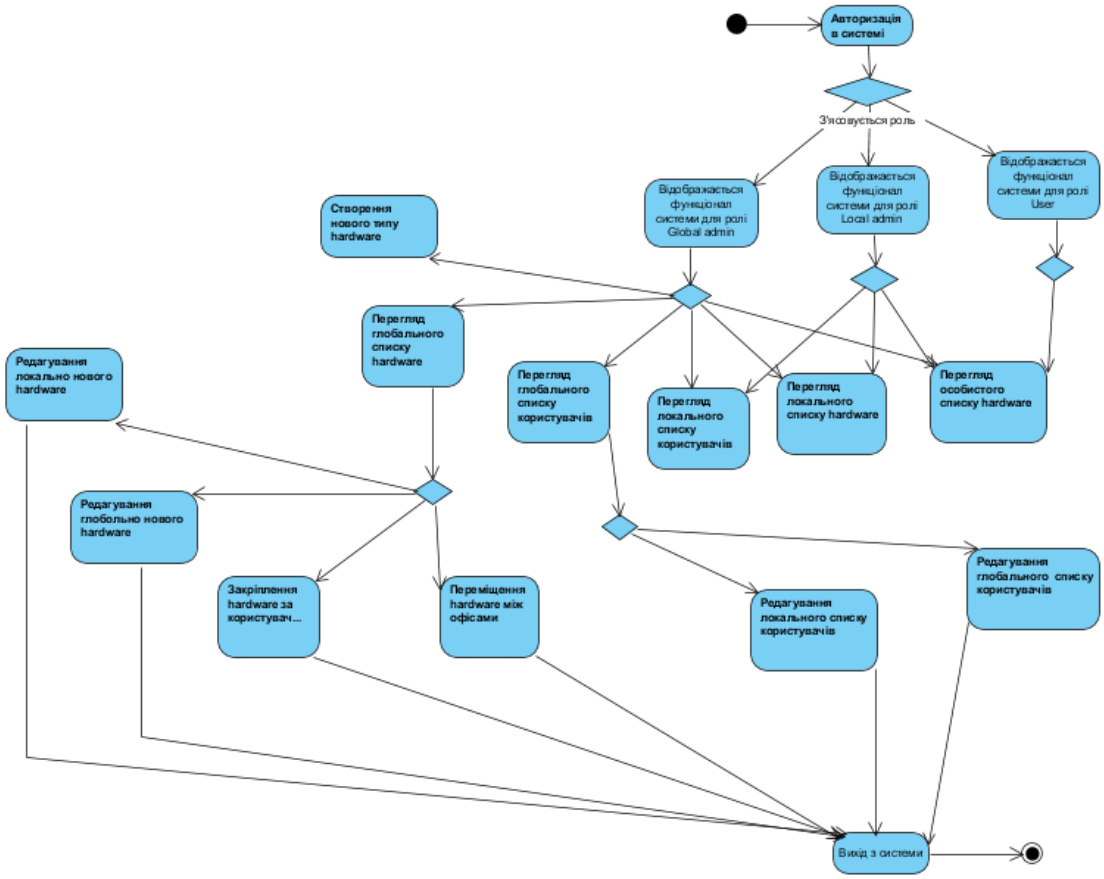

Рисунок 2.12 – Діаграма діяльності системи інвентаризації техніки на основі хмарних технологій

Переглянувши дану діаграму, користувач може дізнатися, які кроки необхідно здійснити щоб комфортно працювати з системою.

2.4.2 Діаграма послідовності системи інвентаризації техніки на основі хмарних технологій

Діаграма послідовності була розроблена для опису взаємодії об’єктів за допомогою сценаріїв, що відображають події, пов’язані з їх створенням і видаленням. Взаємодія об’єктів контролюється подіями, які відбуваються в сценарії і ініціюються повідомленнями до інших об’єктів.

Всі дійові особи показані у верхній частині діаграми. Стрілки відповідають повідомленням, переданим між дійовою особою і об’єктом або між об’єктами для виконання необхідних функцій.

Рисунок 2.13 – Діаграма послідовності системи інвентаризації техніки на основі хмарних технологій

Як видно з діаграми послідовності, спочатку потрібно визначити рівень доступу користувача. Наступний крок, є вибір дії користувача. Після вибору дії, користувач перенаправляється до потрібної сторінки. При додаванні або редагуванні даних користувач перенаправляється на сторінку додавання (редагування) даних, працює з даними. Після чого система оброблює нові дані і, залежно від дії, показує успішність її виконання.

2.5 Об’єктно-орієнтоване проєктування системи інвентаризації техніки на основі хмарних технологій

Аналіз об’єкта проеєктування – це розробка відповідної логічної моделі, то проєктування – це розробка рішення згідно моделі аналізу.

Об’єктно-орієнтоване проєктування використовує різні прийоми представлення моделі що в свою чергу відбивають класи, об’єкти, процеси та

модулі структури системи, також воно ґрунтується на об’єктно-орієнтованій декомпозиції.

Висновки до розділу2

Протягом другого розділу було проаналізовано проєктування системи інвентаризації техніки на основі хмарних технологій, було проведено:

1) аналіз вимог до об’єкту проєктування;
2) формування вимог за допомогою діаграми прецедентів системи;
3) побудовано діаграму діяльності системи;
4) побудовано діаграми послідовності;
5) архітектурне проєктування;
6) розгортання системи на апаратних засобах;
7) моделювання поведінки системи.

Отримані данні дозволять надалі конкретно сформувати об’єкт проєктування, однозначно сформувати те, як має виглядати система, що і як необхідно реалізувати. Був обраний метод проєктування – вебдодаток для розширення можливостей кросплатформленості, конкретизовано поведінку системи в залежності від дій користувача.

РОЗДІЛ 3
РОЗРОБКА ТА ТЕСТУВАННЯ СИСТЕМИ ІНВЕНТАРИЗАЦІЇ ТЕХНІКИ НА ОСНОВІ ХМАРНИХ ТЕХНОЛОГІЙ

У даному розділі передбачається вибір програмного середовища, розробка алгоритмів, та кодування.

3.1 Розробка системи інвентаризації на основі хмарних технологій

Розробка є одним з фундаментальних етапів створення програмного забезпечення. Цей етап дозволяє здійснити перехід від моделі системи до її реального втілення. При виконанні розробки відбувається перетворення всіх теоретико-концептуальних рішень у прикладні рішення для системи. Звісно, під час розробки деякі з отриманих на попередніх етапах рішень можуть зазнати деяких метаморфоз внаслідок чого зміниться їх сутність.

Проте сильний відхід від початкової концепції неможна вважати нормальним, так як він може зруйнувати ту архітектуру яку було вже осмисллено. Тому цей етап необхідно виконувати дуже ретельно, так як саме він визначає якість отриманого продукту.

Етап розробки включає в себе вибір програмного середовища, розробку бази даних, усіх необхідних частин з яких складається програма та безпосередній процес написання програми.

3.1.1 Обґрунтування вибору засобів реалізації системи інвентаризації техніки на основі хмарних технологій

Вибір програмних та апаратних засобів для розв’язання поставленої задачі проводиться на підставі постановки задачі та розробки структури та алгоритму функціонування розроблюваного об’єкту.

Основним засобом реалізації даної системи – є система хмарних обчислень SalesForce.

Saleforce.com – американська компанія, розробник однойменної CRM-системи, що надається замовникам виключно за моделлю SaaS. Під найменування Force.com компанія представляє PaaS-платформу для самостійної розробки додатків, а під брендом Database.com – хмарну систему управління базами даних.

Для розробки користувачами використовується власна Java-подібна мова Apex і власне засіб проєктування Visualforce з вихідним форматом на основі XML, що забезпечує генерацію користувальницьких HTML / AJAX- і Flex-інтерфейсів. Платформа надається виключно за передплатою, в рамках концепції PaaS. Залежно від рівня підписки доступні різні технічні можливості.

Apex – мова програмування, яка використовується розробниками, створюючи додаток на серверіFlow.com. Розробники можуть виконувати операції з управління потоком та транзакціями, а також здійснювати виклики API до API Flow.com за допомогою Apex. Apex революціонує спосіб розробників створювати додатки на замовлення. Це дозволяє клієнтам додати ділову логіку до свого застосування.

Як систему управління базами даних платформа використовує три рипліцированих кластера Oracle RAC з восьми вузлів кожен, кластери розташовані в трьох віддалених один від одного центрах обробки даних. В одній схемі Oracle Database обробляються дані відразу декількох компаній-передплатників, використовується стабільна схема даних, передпідготовлена до розширення додатковими об’єктами таким чином, що в одних і тих же таблицях зберігаються дані різних передплатників, незалежно від відмінності в специфічних атрибутах об’єктів для різних передплатників. Широко використовується секціонування таблиць бази даних.

IntelliJ IDEA – інтегроване середовище розробки ПЗ на багатьох мовах програмування, зокрема Java, JavaScript, Python, розроблена компанією JetBrains.

IntelliJ IDEA добре сумісна з багатьма популярними вільними інструментами розробників, такими як CVS, Subversion, Apache Ant, Maven і JUnit.

Після огляду усіх недоліків та переваг було обрано мову програмування Apex. Вона має більш зручне середовище розробки, не прив’язане до операційної системи, так як розробники системи працюють на різних операційних системах, то і система керування повинна працювати і розроблятися однаково на різних ОС. Також IntelliJ IDEA має більш гнучкий інтерфейс, саме середовище є інтелектуальнішим за Visual Studio. Для створення інтерфейсу використовуватиметься JavaFX, яка має зручний редактор інтерфейсу програми, що теж є перевагою. Тому для створення системи керування буде відбуватися за допомогою мови програмування Apex та середовищем розробки IntelliJ IDEA.

3.1.2 Опис структурної схеми системи інвентаризації техніки на основі хмарних технологій

Одним з найважливіших етапів архітектурного проєктування є побудова ER-діаграми.

Модель «сутність-зв’язок» – модель даних, яка дозволяє описувати концептуальні схеми за допомогою узагальнених конструкцій блоків. ER-модель – це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення що його реалізує, є модель «сутність-зв’язок». Важливим є той факт, що з моделі «сутність-зв’язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об’єктна), тому вона є найзагальнішою.

Модель сутність-зв’язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які

пов’язані між собою певними зв’язками, які виражають залежності і вимоги між ними, такі як: одна будівля може бути розділена на нуль або більше квартир, але одна квартира може бути розташована лише в одній будівлі. Сутності можуть мати різні властивості (атрибути), які характеризують їх. Діаграма, створена для представлення цих сутностей, атрибутів і зв’язків графічно, називають сутність-зв’язок діаграмами.

ER-модель була реалізована в вигляді бази даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є покажчиками фізичної реалізації зв’язків між сутностями.

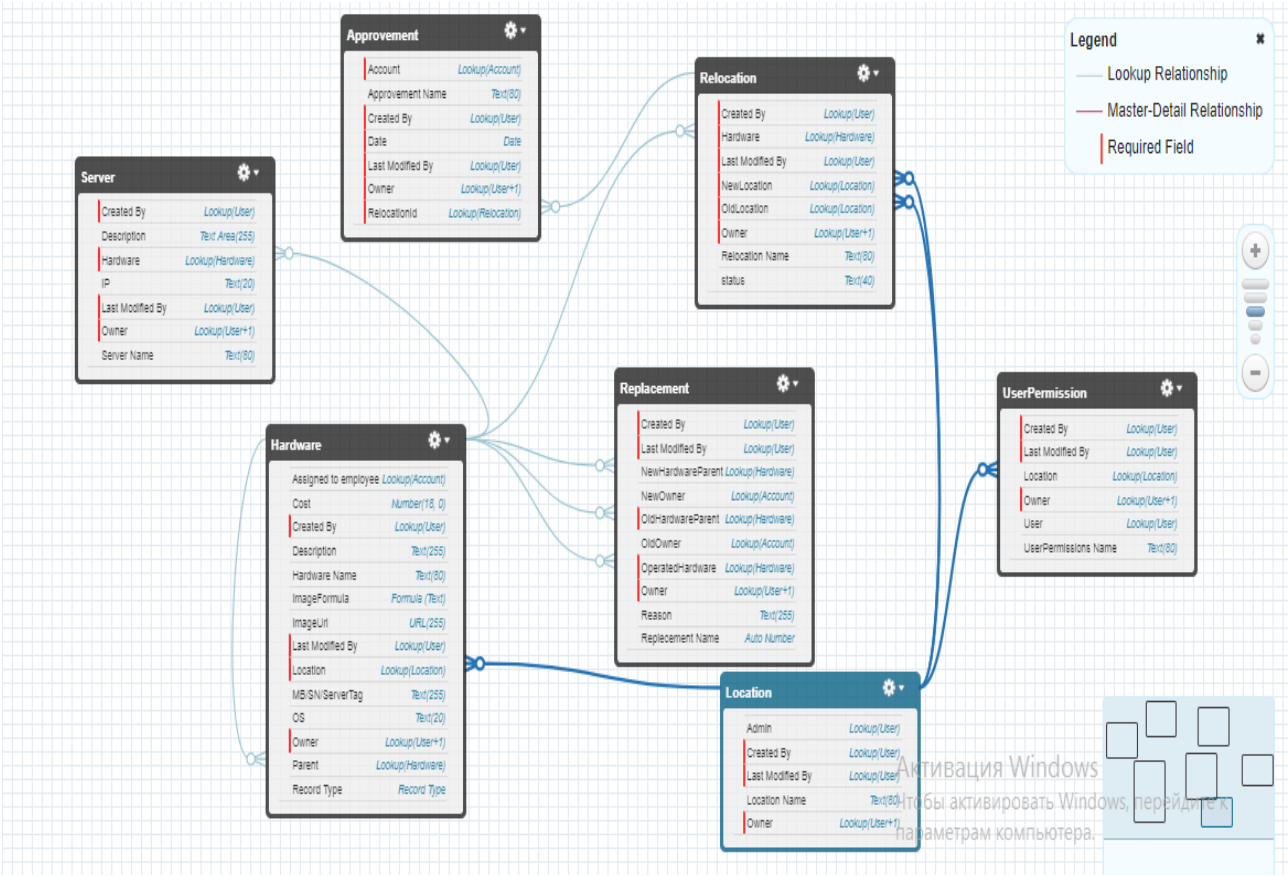

Рисунок 3.1 – Діаграма класів системи інвентаризації техніки на основі хмарних технологій

3.1.3 Розгортання програмного засобу для навчальної демонстрації різних способів знаходження визначника матриці на апаратних засобах

Для опису розгортання системи на апаратних засобах було створено діаграму розгортання, яка демонструє принцип роботи системи загалом.

Було вирішено реалізовувати систему, як вебдодаток для кросплатформеності. Це зменшить кількість обмежень як програмних так і апаратних. Таким чином доступ до системи буде можливий за наявності мережі INTERNET і встановленого будь-якого браузера.

Діаграма розгортання відображає обчислювальні вузли під час роботи програми, компоненти, та об’єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. На діаграмі розгортання (рис. 3.2) представлено принцип роботи системи.

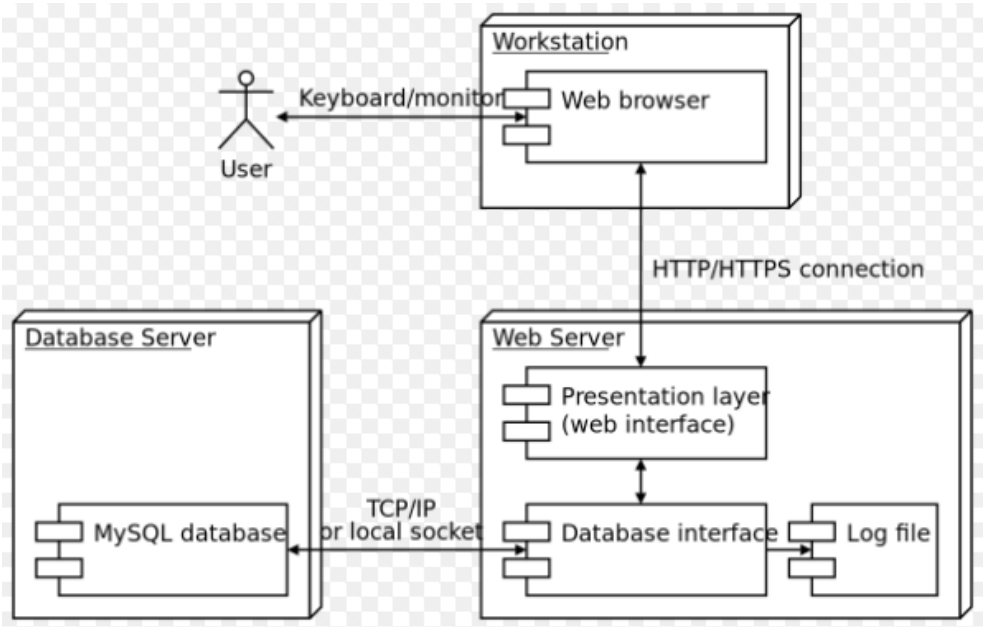

Рисунок 3.2 – Діаграма розгортання вебсистеми інвентаризації техніки на основі хмарних технологій

3.2 Структурна схема системи інвентаризації техніки на основі хмарних технологій

Структурна схема системи інвентаризації техніки на основі хмарних технологій складається з наступних етапів:

1) першим етапом роботи з програмою є авторизація в системі.

2) програма отримує введені дані від користувача та перевіряє рівень доступу даного користувача.

3) наступним кроком відкривається головна сторінка програми з інформацією про користувачів та технікою, яка за ним закріплена.

4) також на екрані є кнопки сторінок, на які користувач може перейти і ознайомитись з відповідною інформацією.

5) при переході на активні сторінки додавання техніки або користувачів, система відкриває форми вводу для додавання нових даних.

6) роль користувача «Global admin» підтримує функціонал переміщення техніки між офісами. При переході на сторінку техніки, активною є кнопка переходу до моду переміщення. Натискаючи на неї таблиця списку техніки доповнюється колонкою Місто, в якій можна змінити офіс, в якому перебуває техніка.

Структурна схема мобільного програмного додатку «Асистент викладача», зображена на рисунку 3.3.

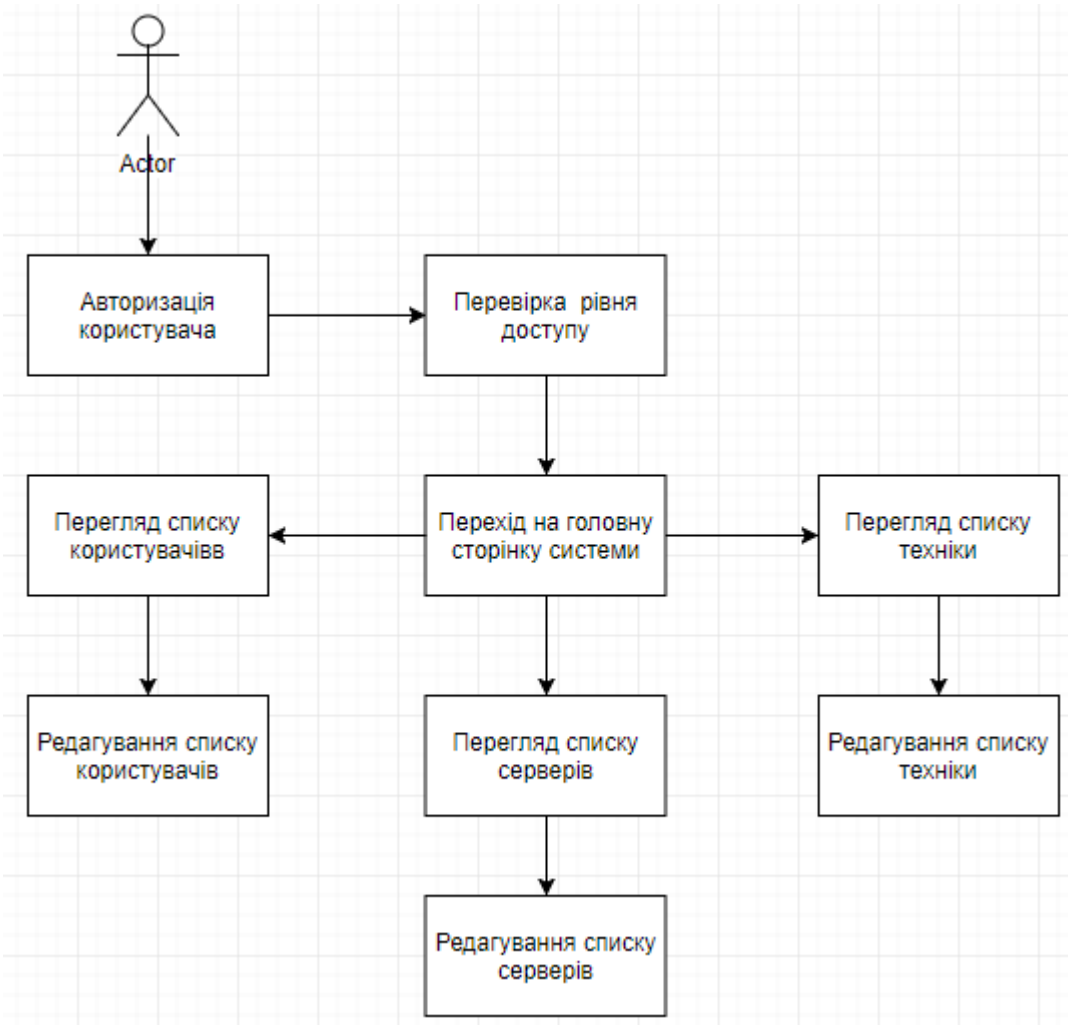

Рисунок 3.3 – Структурна схема вебсистеми інвентаризації техніки на основі хмарних технологій

3.3 Розробка інтерфейсів системи інвентаризації техніки на основі хмарних технологій

Інтерфейс – сукупність засобів і методів взаємодії між елементами системи. Залежно від контексту, поняття застосовне як до окремого елементу (інтерфейс елемента), так і до зв’язків елементів (інтерфейс сполучення елементів, зв’язуюча ланка). Наприклад, клавіатура і миша є інтерфейсом комп’ютера в контексті «користувач – ЕОМ», а адреса електронної пошти – комунікаційним інтерфейсом користувача інтернету. У широкому сенсі інтерфейс – це визначена стандартами межа між взаємодіючими незалежними

об’єктами. Інтерфейс задає параметри, процедури і характеристики взаємодії об’єктів.

Фізичний інтерфейс – пристрій, що перетворює сигнали і передає їх від одного компоненту устаткування до іншого. Фізичний інтерфейс визначається набором електричних зв’язків і характеристиками сигналів. Коли кажуть про фізичний інтерфейс, найчастіше мова йде про комп’ютерні порти.

Інтерфейс користувача – елементи і компоненти програми, які здатні впливати на взаємодію користувача з програмним забезпеченням.

В процесі розробки програмному засобі для навчальної демонстрації знаходження визначника матриці було визначено необхідність розробки зовнішнього інтерфейсу форм системи. Для цього були спроєктовані макети таких форм: головна форма програми, форма яка відповідає за вибір розмірності матриці, форма підказки.

3.3.1 Розробка головної сторінки системи інвентаризації техніки на основі хмарних технологій

Макет зовнішнього вигляду головної сторінки системи зображено на рисунку 3.4.

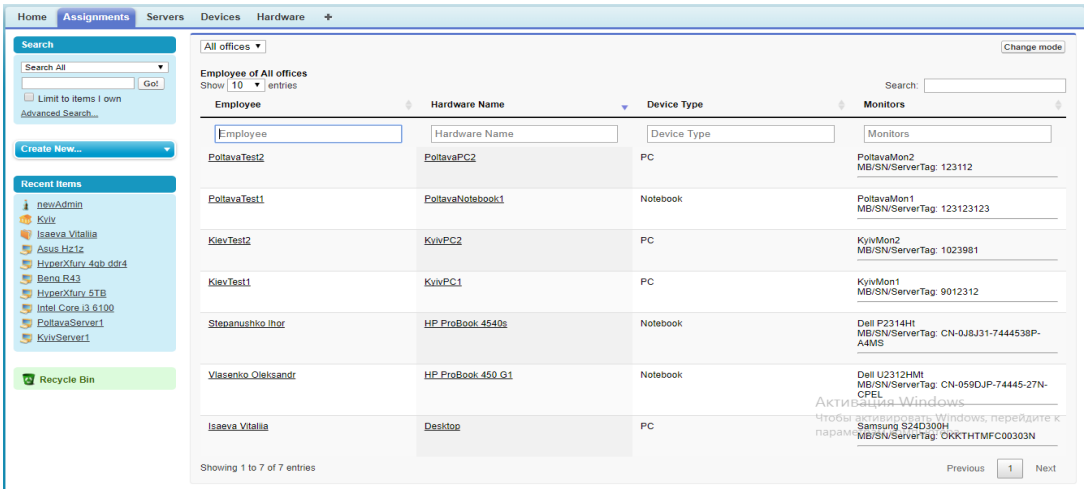

Рисунок 3.4 – Головна сторінка системи інвентаризації техніки на основі хмарних технологій

Як видно з рисунка 3.4, основне вікно програми складається з сторінок «Assignments», «Servers», «Devices», «Hardware». Також тут присутня таблиця для виводу інформації про користувачів всіх офісів, та девайсах, що за ними закріплені.

При розробці функціоналу головної сторінки були використані дефолтні стилі таблиць SalesForce і можливості мови запитів SOQL, за допомогою якої було виконано вибірку даних до таблиці.

Для зручності перегляду та пошуку інформації було додано поля фільтрації даних за іменем полів і по містам (для ролі Гобалььного адміністратора). Також був реалізований розширений вигляд таблиці для перегляду додаткової інформації про техніку.

Клас MainInventoryController – відповідає за контент головної сторінки системи зображено на рисунку 3.5.

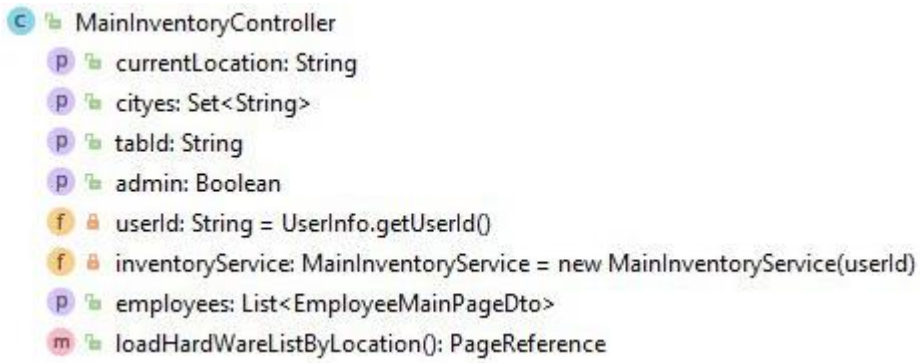

Рисунок 3.5 – Клас MainInventoryController

3.3.2 Розробка сторінки серверів системи інвентаризації техніки на основі хмарних технологій

Макет зовнішнього вигляду сторінки серверів системи зображено на рисунку 3.5. На даному рисунку зображено зовнішній вигляд сторінки списку серверів фірми. Були використані описані вище дефолтні стилі середовища, для формування таблиці і мова запитів SOQL. До кожного серверу, для зручності редагування інформації, було додано кнопку Редагувати.

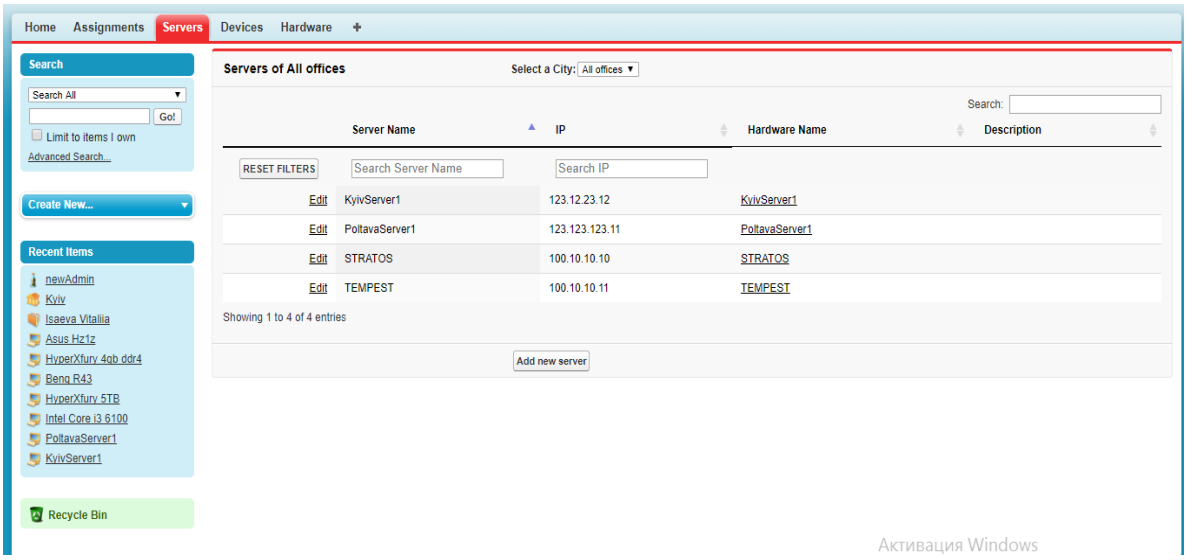

Рисунок 3.6 – Сторінка серверів системи інвентаризації техніки на основі хмарних технологій

Для зручності перегляду та пошуку інформації було додано поля фільтрації даних за іменем полів і по містам (для ролі Глобального адміністратора). Також був реалізований розширений вигляд таблиці для перегляду додаткової інформації про техніку.

КласServersService – відповідає за оновлення і редагування серверів, що знаходяться на обліку фірми показано на рисунку 3.7.

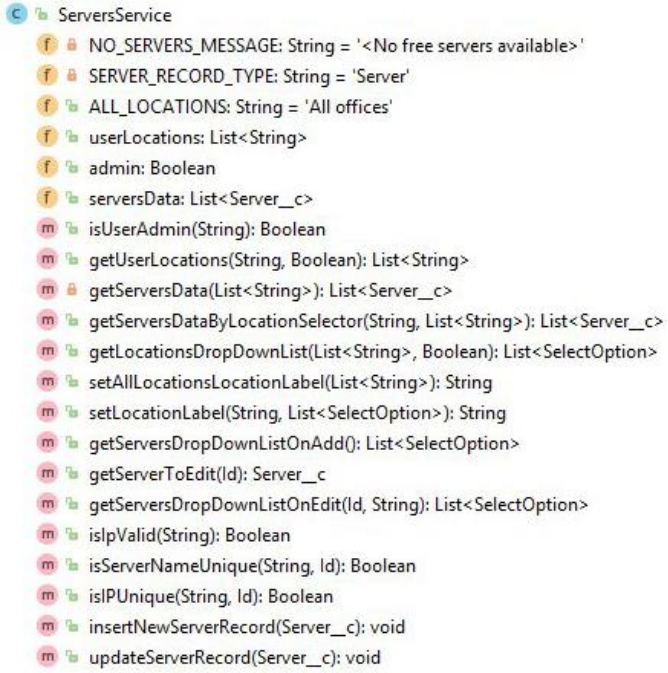

Рисунок 3.7 – Клас ServersService

Клас EditServerontroller – відповідає за логіку оновлення та редагування інформації про сервери фірми зображено на рисунку 3.8.

Рисунок 3.8 – Клас EditServerontroller

3.3.3 Розробка сторінки списку техніки системи інвентаризації техніки на основі хмарних технологій

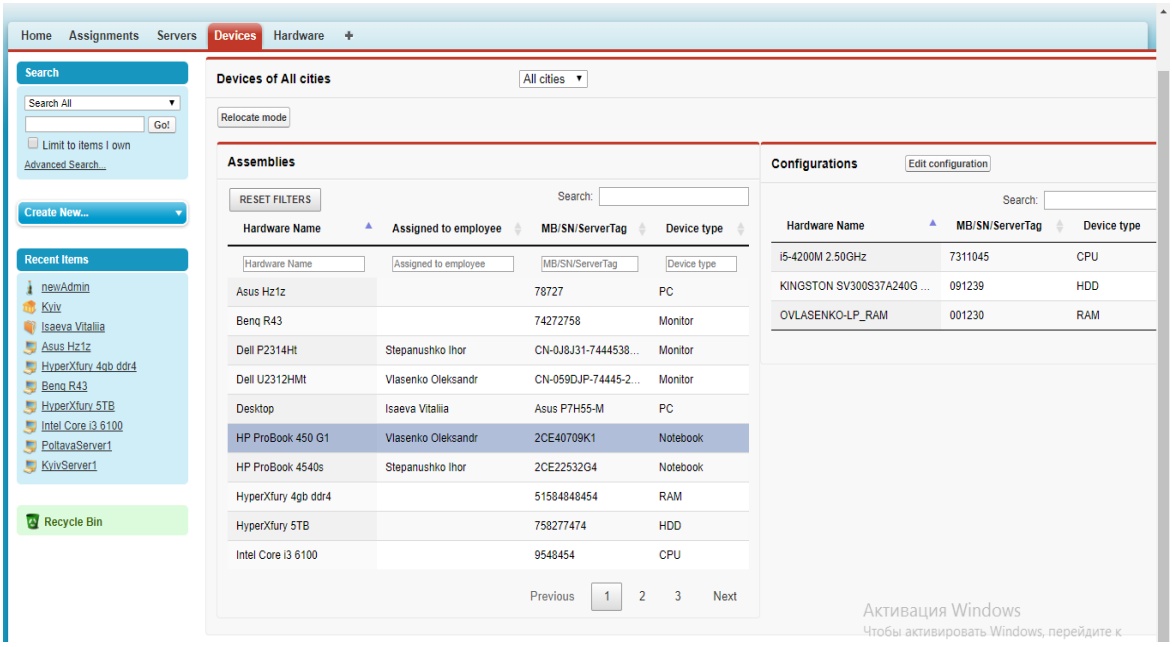

Рисунок 3.9 – Сторінка техніки системи інвентаризації техніки на основі хмарних технологій

Як видно на рисунку 3.9, сторінка списку техніки складається з двох таблиць. Перша таблиця зберігає інформацію про всі технічні пристрої, що зберігаються на фірмі. Друга таблиця створена для виводу комплектуючих та серійного номеру окремого пристрою.

При розробці функціоналу головної сторінки були використані дефолтні стилі таблиць SalesForce і можливості мови запитів SOQL, за допомогою якої було виконано вибірку даних до таблиці.

Для зручності перегляду та пошуку інформації було додано поля фільтрації даних за іменем полів і по містам (для ролі Глобального адміністратора). Також був реалізований розширений вигляд таблиці для перегляду додаткової інформації про техніку.

Клас DevicePageController – відповідає за опрацювання та вивід даних на сторінку девайсів.

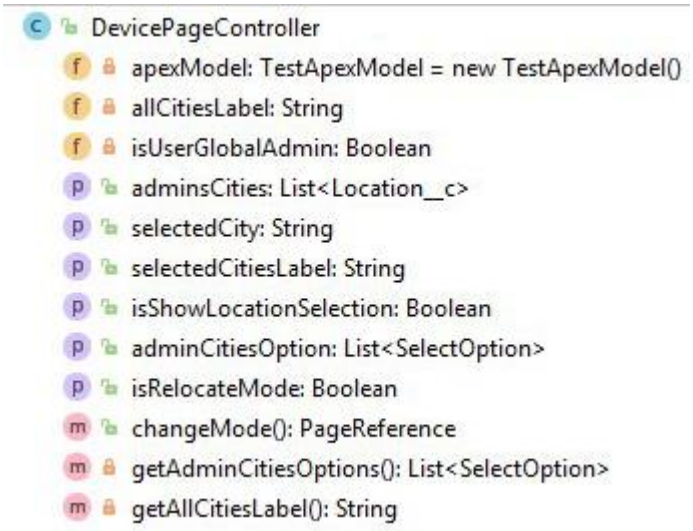

Рисунок 3.10 – Клас DevicePageController

3.3.4 Розробка сторінки додавання техніки системи інвентаризації техніки на основі хмарних технологій

Для зручності додавання нових пристроїв на головну сторінку системи було додано вкладку додавання техніки. Макет сторінки зображений на рисунку 3.11.

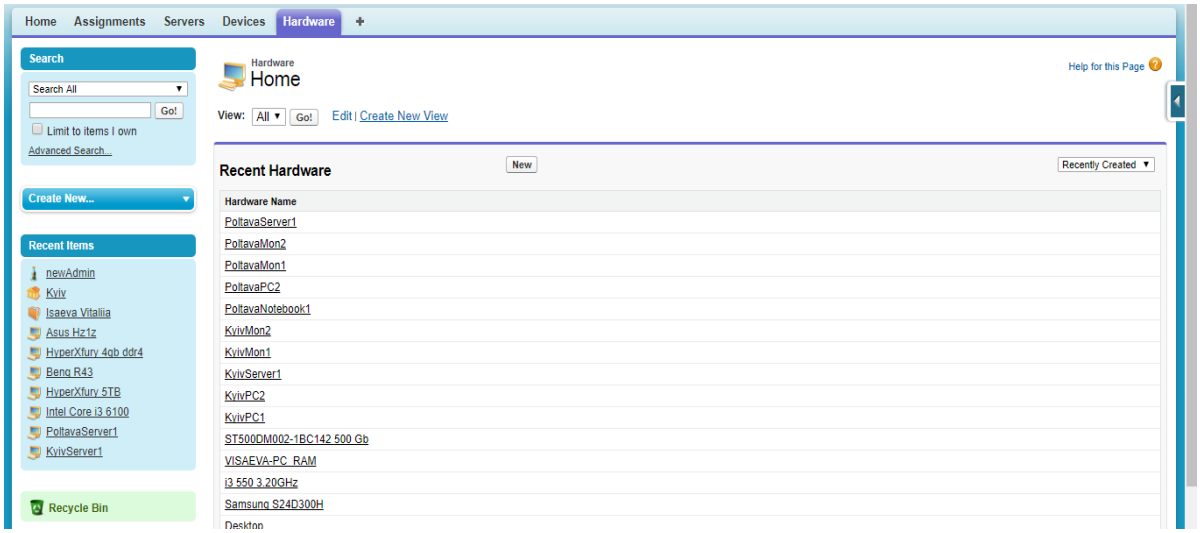

Рисунок 3.11 – Сторінка додавання техніки системи інвентаризації техніки на основі хмарних технологій

При натисканні на кнопку додавання нового пристрою, відкривається сторінка вибору типу пристрою. Після обрання типу, відкривається наступна форма вводу інформації про новий пристрій. Обидва кроки продемонстровані на рисунку 3.12.

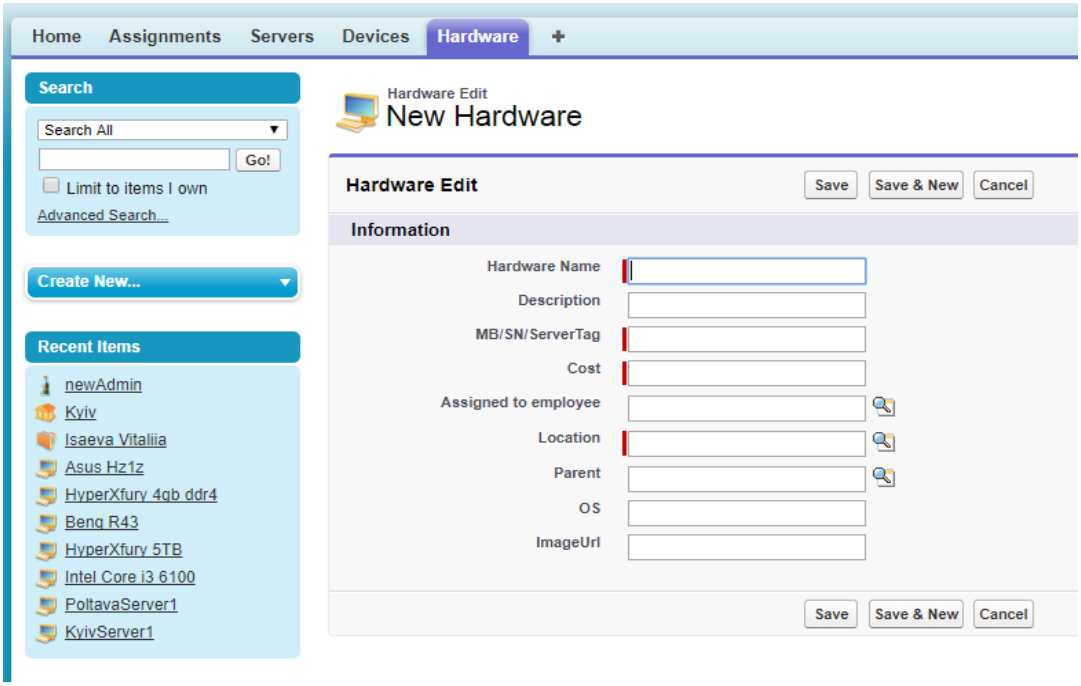

Рисунок 3.12 – Форма введення інформації, про пристрій, що додається системи інвентаризації техніки на основі хмарних технологій

Обов’язкові для заповнення поля виділені червоним кольором. Для точного введення інформації, що пов’язує даний девайс з користувачем або містом, було додано пошук по даній, вже існуючій, інформації.

Клас HardwareController – відповідає за додавання інформації на сторінку Hardware показаний на рисунку .

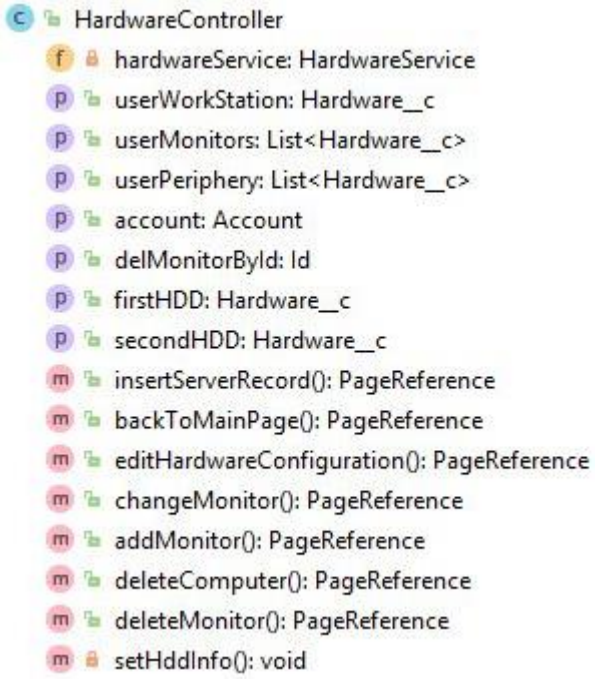

Рисунок 3.13 – Клас HardwareController

3.3.5 Розробка сторінки додавання типу доступу користувача системи інвентаризації техніки на основі хмарних технологій

Сторінка додавання типу доступу користувача скрита на основному сайті, для збільшення впевненості в захисті інформації.

Структура сторінки має три поля для введення інформації, в які адміністратору необхідно ввести назву типу доступу, ім’я користувача і місто, адміністратором якого буде новий користувач. У разі не заповнення поля міста, користувач наділяється правами глобального адміністратора системи. Макет сторінки зображений на рисунку 3.14.

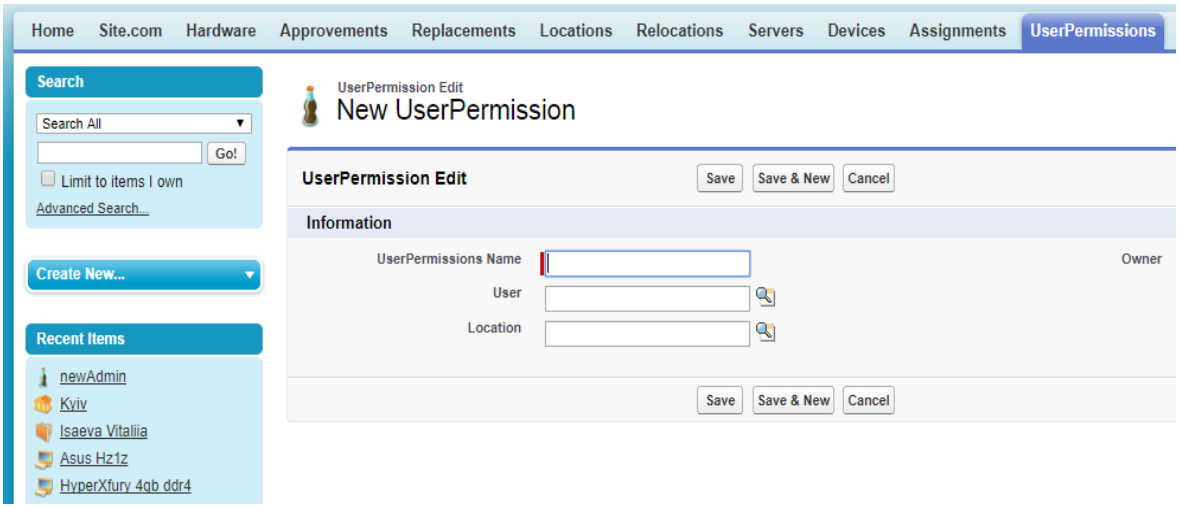

Рисунок 3.14 – Форма введення інформації, про тип доступу, що надається користувачу системи інвентаризації техніки на основі хмарних технологій

3.4 Тестування системи інвентаризації техніки на основі хмарних технологій

Для впевненості в якості написаного продукту було проаналізовано методи для тестування, обрано методи функціонального тестування, випадкове тестування, структурне тестування. У традиційній класифікації їх відносять до методів «чорної скриньки». Вони застосовуються для тестування зовнішніх та внутрішніх функцій системи і припускають наявність специфікації, використовуваної як еталон. Методи відрізняються між собою підходами до вибору даних з множини входів функцій.

Процес тестування почався до того, як було розроблено код. Для якісного написання системи було написано приймальні критерії. Після створення макетів системи, було написано ряд тестових випадків і випадків використання, які в повній мірі описували бізнес логіку системи. Перелік тестових випадків, за якими проводилося тестування подано в додатку А. Перелік випадків використання, за якими проводилося тестування подано в додатку Б.

Тестування випадків використання і тестових випадків проводилось, як тільки було створено виконуваний код.

Завданням функціонального тестування є перевірка відповідності програми своїм специфікаціям. При даному підході текст програми не доступний, і програма розглядається як «чорна скринька». Найпоширенішими видами функціонального тестування є методи випадкового тестування, еквівалентної розбивки й аналізу граничних умов.

За результатом тестування було зазначено, що система пройшла валідацію та верифікацію, тобто відповідає описаним вимогам і всі задачі бізнес логіки були реалізовані.

Висновки до розділу 3

У цьому розділі було наголошено на особливостях реалізації тих проєктних рішень та вимог, що було сформовано у другому розділі. Для системи керування мережею пошукових агентів було обрано мову програмування Apex та середовище розробки IntelijIDEA. Вони були обрані за універсальність, можливість отриманого продукту працювати на більшості сучасних операційних системах та сучасний редактор інтерфейсу.

Після цих етапів сформовано готову систему інвентаризації техніки на основі хмарних технологій , і її лишається лише протестувати. Тестування теж важливий етап, так як без нього неможливо знайти ті помилки, які міг допустити розробник у процесі розробки. Звісно, часткове тестування функцій відбувалося і під час розробки системи, але фінальне тестування має більше значення, бо воно демонструє те, наскільки життєздатна отримана система.

Після розробки програми було проведено ручне та модульне тестування, яке допомогло переконатись в правильності роботи системи.

ВИСНОВКИ

Система інвентаризації техніки на основі хмарних технологій – важливий етап у функціонуванні будь-якого виробництва. Зручність використання і захист даних – це основні аспекти, які не були виявленні в жодній із розглянутих систем аналогів.

Був проведений опис та аналіз предметної області – процес і, сформовано постановку завдання на розробку. Провівши аналіз предметної області було виділено мету та задачі системи.

Після розгляду теоретичного матеріалу було сформовано вимоги до системи інвентаризації техніки. Визначено первинні вимоги, на основі яких за допомогою діаграм було виведено детальні вимоги.

Після визначення вимог, які необхідні для створення системи було проведено архітектурне проєктування та сформовано діаграму пакетів. З її допомогою було з’ясовано початкову структуру реального проєкту, на основі пакетів сформовано класи. При детальному проєктуванні розроблено діаграму класів.

У роботі було наголошено на особливостях реалізації тих проєктних рішень та вимог, що було сформовано до етапу проєктуванні Apex.

Після етапу проєктування було створено діаграму розгортання на апаратних засобах.

Для реалізації системи інвентаризації техніки на основі хмарних технологій було обрано мову програмування Apex та середовище розробки IntelijIDEA. Ця мова і середовище були обрані за їх універсальність, легкість та можливість створювати систему на основі хмарної технології SalesForce.

На наступному етапі розробки системи було сформовано логічну структуру системи. Таким чином розроблено зручний і зрозумілий інтерфейс системи.

Після виконання цих етапів сформовано готову систему інвентаризації техніки на основі хмарних технологій, і її залишилося лише протестувати.

Тестування системи інвентаризації техніки на основі хмарних технологій пройшло вдало, очікувані результати під час тестування аналогічні тим діям, що виконала система.

Результатом даної кваліфікаційної роботи є система інвентаризації техніки на основі хмарних технологій. Отриманий програмний продукт у майбутньому буде застосовано на виробництвах, що пов’язані з використанням великої кількості технічних засобів. Він отримає додаткові функції, та розвиватиметься.

Поставлені до системи завдання було виконано, а вимоги, сформовані у другому розділі досягнуто.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley.

2. Brown, S. (2020). *Software Architecture for Developers*. Leanpub.

3. Coronel, C., & Morris, S. (2020). *Database Systems: Design, Implementation, and Management* (13th ed.). Cengage Learning.

4. Elmasri, R., & Navathe, S. (2019). *Fundamentals of Database Systems* (7th ed.). Pearson.

5. Fowler, M. (2019). *Patterns of Enterprise Application Architecture*. Addison-Wesley.

6. Howard, M., & LeBlanc, D. (2019). *Writing Secure Code* (2nd ed.). Microsoft Press.

7. Humble, J., & Farley, D. (2019). *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley.

8. Kaner, C., Falk, J., & Nguyen, H. Q. (2019). *Testing Computer Software* (2nd ed.). Wiley.

9. Kim, G., Humble, J., Debois, P., & Willis, J. (2020). *The DevOps Handbook*. IT Revolution Press.

10. Kleppmann, M. (2019). *Designing Data-Intensive Applications*. O'Reilly Media.

11. Loukides, M. (2019). *Infrastructure as Code: Managing Servers in the Cloud*. O'Reilly Media.

12. McGraw, G. (2018). *Software Security: Building Security In*. Addison-Wesley.

13. Myers, G. J., Sandler, C., & Badgett, T. (2018). *The Art of Software Testing* (3rd ed.). Wiley.

14. Newman, S. (2021). *Building Microservices: Designing Fine-Grained Systems* (2nd ed.). O'Reilly Media.

15. Pressman, R. S., & Maxim, B. R. (2020). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.

16. Richards, M. (2019). *Software Architecture Patterns*. O'Reilly Media.

17. Richardson, C. (2020). *Microservices Patterns: With Examples in Java*. Manning Publications.

18. Sommerville, I. (2019). *Software Engineering* (10th ed.). Pearson.

19. Бабич, А. В., & Жарко, Д. М. (2018). *Ефективна розробка програмного забезпечення*. BHV.

20. Гриценко, В. І., & Павлов, О. А. (2020). Методи тестування програмних систем. *Кібернетика та системний аналіз*, (4), 67–75.

21. Лавріщева, К. М. (2019). *Програмна інженерія: підручник*. Академперіодика.

22. Федорус, А. П., & Триус, Ю. В. (2020). *Методи та засоби інженерії програмного забезпечення*. ЧДТУ.