

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

## Пояснювальна записка

до магістерської дипломної роботи

126 «Інформаційні системи та технології»  
(освітньо-кваліфікаційний рівень)

на тему «Інформаційна система обліку речового майна в  
військовій галузі»

Виконав: студент II курсу, групи ІСТ-22зм спеціальності 126

«Інформаційні системи та технології»

Пермяков Дмитро Сергійович

Керівник Захожай О. І.

Рецензент \_\_\_\_\_

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Навчально-науковий інститут (факультет) інформаційних технологій та електроніки

Кафедра інформаційних технологій та програмування

Освітньо-кваліфікаційний рівень 126

Спеціальність «Інформаційні системи та технології»

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТП

\_\_\_\_\_ д.т.н., доц. Захожай О. І.  
(підпис)

« \_\_\_\_ » \_\_\_\_\_ 2023р.

ЗАВДАННЯ

на магістерську дипломну роботу студенту

Пермякову Дмитру Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система обліку речового майна в військовій галузі

керівник роботи доцент, д.т.н. Захожай Олег Ігорович

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом університету від « 21 » 10 2023року № 614/15.15-С

2. Строк подання студентом роботи 06.12.2023

3. Вихідні дані до роботи Матеріали науково-дослідницької практики, науково-методична література; дані інтернет-мережі.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Методи, засоби, моделі та програмна реалізація інформаційної системи обліку речового майна в військовій галузі.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 20 жовтня 2023**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Розробка технічного завдання	20.10.2023-30.10.2023	
2	Аналіз літератури з досліджуваної проблеми	31.10.2023-05.11.2023	
3	Аналіз методів, засобів та моделей обліку речового майна в військовій галузі	06.11.2023-12.11.2023	
4	Реалізація інформаційної системи	13.11.2023-28.11.2023	
5	Оформлення пояснювальної записки	28.11.2023-05.12.2023	
6	Здача пояснювальної записки на кафедрі	06.12.2023	
7	Підготовка доповіді та презентації	07.12.2023-08.12.2023	

Студент \_\_\_\_\_  
(підпис)Пермяков Д. С.  
(прізвище та ініціали)Керівник роботи \_\_\_\_\_  
(підпис)Захожай О. І.  
(прізвище та ініціали)

## РЕФЕРАТ

Магістерська дипломна робота: стор.71, табл.1, рис.22, джерел 33.

Об'єкт дослідження – облік речового майна в військовій галузі. Предмет дослідження – процес розробки інформаційної системи для ефективного обліку речового майна з використанням сучасних технологій.

Мета дослідження – створення інформаційної системи, яка дозволить військовим підрозділам здійснювати облік речового майна, покращуючи ефективність та безпеку управління активами.

Дослідження спрямоване на вивчення та аналіз існуючих систем обліку речового майна в контексті військової галузі, формулювання функціональних вимог до системи, вибір технологій для розробки, створення модулів для обліку активів, реалізацію графічного інтерфейсу користувача та валідацію системи шляхом тестування.

**КЛЮЧОВІ СЛОВА: ОБЛІК, ІНФОРМАЦІЙНА СИСТЕМА, ТЕСТУВАННЯ, БЕЗПЕКА ДАНИХ, УПРАВЛІННЯ АКТИВАМИ**

## ABSTRACT

Master's dissertation: p. 71, table 1, fig. 22, references 33.

The object of research is the accounting of material property in the military sector. The subject of the study is the process of developing an information system for effective accounting of material property using modern technologies.

The purpose of the study is to create an information system that will allow military units to keep records of material property, improving the efficiency and security of asset management.

The research is aimed at studying and analyzing existing systems for accounting for tangible property in the context of the military industry, formulating functional requirements for the system, selecting technologies for development, creating modules for asset accounting, implementing a graphical user interface and validating the system through testing.

**KEYWORDS: ACCOUNTING, INFORMATION SYSTEM, TESTING, DATA SECURITY, ASSET MANAGEMENT**

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ РЕЧОВОГО МАЙНА .....	9
1.1 Аналіз літератури за темою .....	9
1.2 Особливості обліку речового майна в військовій галузі .....	14
1.3 Порівняння сучасних інформаційних систем обліку речового майна .....	18
1.4 Проблематика та постановка завдань дослідження .....	28
Висновки до розділу 1 .....	29
РОЗДІЛ 2 ФУНКЦІОНАЛЬНІСТЬ ТА ВИБІР ТЕХНОЛОГІЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ РЕЧОВОГО МАЙНА .....	31
2.1 Функціональні вимоги до системи обліку речового майна.....	31
2.2 Алгоритм обліку речового майна.....	33
2.3 Вибір технологій розробки програмної системи .....	35
Висновки до розділу 2 .....	43
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ РЕЧОВОГО МАЙНА В ВІЙСЬКОВІЙ ГАЛУЗІ.....	44
3.1 Архітектура інформаційної системи обліку речового майна в військовій галузі.....	44
3.2 Розробка модулів для обліку речового майна.....	51
3.3 Реалізація графічного інтерфейсу користувача .....	57
3.4 Тестування та валідація програмної системи .....	63
Висновки до розділу 3 .....	66
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	69

## **ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ**

ІС - інформаційна система

БД - база даних

UI - користувацький інтерфейс

API - інтерфейс програмування застосунків

IDE - середовище розробки

SQL - мова структурованих запитів

GUI - графічний інтерфейс користувача

URL - уніфікований ресурсний локатор

## ВСТУП

**Актуальність теми дослідження.** Тема дослідження є актуальною через важливість ефективного управління речовим майном в військовій галузі. Військові структури повинні точно та ефективно обліковувати та управляти різноманітними активами. Специфіка військового обліку, така як класифікація активів та вимоги до безпеки даних, вимагає розробки спеціалізованої інформаційної системи.

Використання сучасних технологій та програмних рішень допомагає покращити облік та управління речовим майном. Це спрямовано на забезпечення точності та доступності даних, а також на зменшення людського втручання та оптимізацію процесів. Актуальність полягає і в можливості економії ресурсів та зменшенні ризиків втрати даних завдяки впровадженню інформаційної системи для обліку речового майна в військовій галузі.

**Мета та завдання.** Метою даного дослідження є створення інформаційної системи, яка дозволить військовим підрозділам здійснювати облік речового майна, покращуючи ефективність та безпеку управління активами.

Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

1. Провести огляд і аналіз наявних джерел літератури, що стосуються обліку речового майна в військовій галузі. Це допоможе з'ясувати існуючі підходи, методи та проблеми, пов'язані з цією темою, і визначити напрямки подальших досліджень.
2. Провести порівняльний аналіз існуючих інформаційних систем, які використовуються для обліку речового майна в військовій галузі. Визначити переваги та недоліки кожної системи, та їхню ефективність.
3. Розробити детальний перелік функціональних вимог, які повинна виконувати інформаційна система обліку речового майна в військовій галузі. Це визначить базові вимоги до системи та її можливості.
4. Розробити алгоритм та процедури для збору, зберігання, оновлення та використання даних про речове майно. Врахувати особливості військового обліку та дотримання вимог щодо безпеки даних.

5. Визначити технології для розробки інформаційної системи обліку речового майна військової галузі. Розглянути можливості використання сучасних програмних засобів та підходів до створення програмного забезпечення.
6. Розробити інформаційну систему обліку речового майна в військовій галузі.

**Об'єкт та предмет.** Об'єкт дослідження – облік речового майна в військовій галузі. Предмет дослідження – процес розробки інформаційної системи для ефективного обліку речового майна з використанням сучасних технологій.

**Гіпотеза дослідження:** розробка та впровадження інформаційної системи для обліку речового майна в військовій галузі сприятиме покращенню ефективності та точності обліку активів, зменшить ризики втрати даних, а також спростить процеси управління речовим майном.

**Методи дослідження.** У процесі дослідження використано наступні підходи та методи: метод порівняльного аналізу для існуючих інформаційних систем обліку речового майна для військових потреб з метою визначення переваг та недоліків кожної з них, методи та технології проектування інформаційних систем.

**Новизна дослідження** є розробка та впровадження інформаційної системи обліку речового майна в військовій галузі, яка сприяє покращенню ефективності та точності управління активами та забезпечує збереження даних у захищеному вигляді.

**Практичною значущістю** дослідження є те, що розроблена інформаційна система для обліку речового майна в військовій галузі сприяє покращенню управління активами, забезпечує надійність та безпеку даних, що має велике практичне значення для військових підрозділів та організацій.

**Теоретичною основою** дослідження є наукові праці українських та зарубіжних учених, які займаються проблемами моделювання інформаційних систем обліку майна.

**Апробація результатів ...**

**Структура роботи.** Структуру роботи складають вступ, три розділи, висновки та список використаної літератури. Загальний обсяг роботи становить 71 сторінку.



## РОЗДІЛ 1

# ТЕОРЕТИЧНІ ОСНОВИ ІНФОРМАЦІЙНИХ СИСТЕМ ОБЛІКУ РЕЧОВОГО МАЙНА

### 1.1 Аналіз літератури за темою

Проведення аналізу літератури за темою є важливою частиною дослідження, оскільки це допомагає отримати розуміння сучасних підходів, понять і інструментів, що стосуються обраної теми. Важливо розробити фундамент для нашого дослідження, дізнатися, що вже відомо та зроблено в цій області і визначити прогалини або можливості для подальших досліджень.

Стаття Маркової Т.Д. та Пчелянської Г.Б. [1] досліджує інформаційні системи та технології, які використовуються для обліку і контролю. В ній акцентується увага на інтегративній ролі інформаційних систем у забезпеченні єдності процесів обліку, контролю і управління на підприємстві. Описуються основні функції інформаційних систем обліку та контролю, такі як накопичення, збереження, опрацювання та подання інформації користувачам.

У статті також виділяються типові інформаційні системи для обліку та контролю, такі як бухгалтерський, управлінський, системи оперативного контролю оборотних коштів і інші. Розглядаються питання формування інформаційних ресурсів, їх структури та зв'язки між базами даних.

Окрім цього, публікація акцентує увагу на перевагах використання інформаційних систем в контексті підвищення якості, оперативності та ефективності процесів обліку і контролю. Вона є цінним джерелом інформації для нашого дослідження і допоможе краще розуміти основні аспекти інформаційних систем обліку речового майна в підприємствах.

У статті "Методологічні аспекти побудови, функціонування і розвитку автоматизованих інформаційних систем обліку та управління підприємствами" [2], автор О.І. Волот розглядає питання, пов'язані з методологічними аспектами

автоматизованих інформаційних систем (АІС) для обліку та управління підприємствами. Основна мета дослідження полягає у вивченні теоретичних положень та методичних підходів до проектування і функціонування інформаційних систем, а також в удосконаленні системи принципів побудови та функціонування таких систем.

Стаття аналізує наукові принципи, запропоновані в літературі, щодо побудови і функціонування інформаційних систем. Також розглядаються методологічні принципи автоматизації бухгалтерського обліку. На основі проведеного аналізу, автор пропонує нові принципи побудови АІС, які включають орієнтацію на користувача, релевантність, пертинентність та креативність. Ці принципи враховують сучасні вимоги до інформаційних систем і покликані покращити їх функціонування.

Основна ідея статті полягає в тому, щоб аналізувати і вдосконалювати принципи побудови і функціонування інформаційних систем, роблячи їх більш орієнтованими на потреби користувачів і забезпечуючи відповідність сучасним вимогам. Також, автор вказує на необхідність подальших досліджень щодо алгоритмів прийняття рішень щодо впровадження інформаційних технологій на передпроектній стадії.

Стаття розглядає інноваційні аспекти побудови і функціонування інформаційних систем, надаючи принципи, які можуть покращити їх ефективність і відповідність сучасним вимогам.

У статті "Спрощена система обліку і звітності: історичний аспект" Вадима Новікова [3] досліджується історичний аспект розвитку спрощеної системи бухгалтерського обліку та звітності для малих підприємств в Україні. Автор аналізує особливості впровадження спрощеної системи обліку та звітності, яка спрямована на полегшення облікових процедур для малих підприємств в Україні та в інших країнах світу.

Стаття також аналізує нормативно-правову базу, яка регулює спрощену систему в Україні, і надає статистичні дані щодо частки малих підприємств в економіці України. Окрема увага приділяється міжнародним стандартам фінансової звітності для малих і середніх підприємств.

Загальний висновок статті полягає в тому, що спрощена система обліку і звітності допомагає полегшити облікові процедури для малих бізнесів. Однак, автор також вказує на необхідність подальшого вдосконалення цієї системи.

Ця стаття корисна для розуміння історичного розвитку систем обліку та звітності для малих підприємств та для ознайомлення з різними аспектами спрощених систем обліку в Україні та світі.

У статті "Інформаційна система планування, обліку, моніторингу та управління інноваційною діяльністю в сфері охорони здоров'я України" А. Є. Горбань та М. Л. Кони́на [4] досліджується проблематика розробки інформаційної системи для планування, обліку, моніторингу та управління інноваціями в сфері охорони здоров'я України.

Стаття розглядає законодавчі засади інноваційної діяльності в Україні та доводить необхідність розробки автоматизованої системи для обліку інноваційних проектів в галузі охорони здоров'я. Автори визначають основні принципи, яким повинна відповідати розроблювана система, зокрема системність, розвиток, взаємодія, стандартизація, ефективність, безпека даних та надійність.

Стаття також надає докладний огляд основних модулів та функціоналу інформаційної системи, яка розробляється для управління інноваціями в охороні здоров'я.

Загальний висновок статті полягає в тому, що створення інформаційної системи для управління інноваціями є актуальним завданням, і визначено її концептуальні засади.

Стаття "Удосконалення речового майна військовослужбовців" авторів Галавська, Прохоровський, Швиданенко, Котюх і Подоляка [5] має на меті розгляд проблем логістичного забезпечення військових частин Національної гвардії України в польових умовах і вдосконалення речового майна військовослужбовців. Автори розглядають декілька аспектів, включаючи:

Основна ідея полягає в застосуванні поліефірних волокон "DEO-W" для виготовлення нижньої білизни військовослужбовців з метою підвищення її

функціональних властивостей. Це може включати покращення теплоізоляції, вологостійкості та інших параметрів, що важливі в польових умовах.

Пропонується удосконалити конструкції окремих елементів білизни, щоб покращити їх зносостійкість, комфорт та відповідність вимогам польового використання.

Розглядаються питання оптимізації розміщення військових частин в польових умовах з метою забезпечення їх бойової готовності.

Автори надають рекомендації щодо покращення системи забезпечення Нацгвардії, що можуть бути важливими для поліпшення її ефективності та готовності.

Загальною метою статті є забезпечення більш ефективного та комфортного функціонування військових частин у польових умовах та поліпшення речового забезпечення військовослужбовців. Ця стаття містить практично важливі пропозиції та рекомендації для подальшого розвитку логістики і забезпечення військових частин Нацгвардії України.

Стаття [6], авторами якої є Мацнев І. П. та Даниленко В. С., присвячена розробці інформаційної автоматизованої системи обліку обладнання. Система орієнтована на сканування локальної мережі та автоматичну обробку характеристик активних комп'ютерів у цій мережі. Основні можливості розробленої програмної системи включають:

1. **Автоматичне отримання та обробка характеристик комп'ютерів.** Система здатна автоматично зчитувати характеристики активних комп'ютерів у локальній мережі.
2. **Збереження інформації у базі даних.** Отримані дані зберігаються у базі даних MySQL, що дозволяє подальшу обробку та зручний доступ до них.
3. **Редагування записів у базі даних.** Система надає можливість редагування, додавання та видалення записів у базі даних за допомогою спеціалізованого додатку.
4. **Пошук та фільтрація інформації.** Можна здійснювати пошук та фільтрацію інформації, яка зберігається в базі даних.

**5. Оновлення інформації.** Є можливість оновлення інформації щодо апаратного забезпечення конкретного комп'ютера або усієї локальної мережі через повторне сканування.

Автори статті вказують, що робота має практичну і теоретичну цінність, але не проводили розрахунок економічної ефективності. Також у статті зазначено, що реалізація додатку виконана у середовищі Visual Studio з використанням мов C# та PowerShell, СУБД MySQL та фреймворку ASP.NET.

Стаття присвячена розробці інформаційної системи, яка спрощує облік та моніторинг обладнання у локальній мережі.

Випускна кваліфікаційна робота [7] написана Д. В. Грукачем, присвячена розробці веб-системи для обліку обладнання на підприємстві. Основною метою цієї роботи є створення стабільного та зручного додатку для отримання інформації про місцезнаходження та працездатність обладнання.

Розроблено адаптивний інтерфейс, який оптимально відображається на різних пристроях, що покращує зручність користування. Ця система призначена для працівників великих підприємств, які використовують обладнання. Робота також охоплює функціонал адміністрування системи, використання реляційної бази даних для зберігання інформації, можливість користувачів переглядати, додавати, редагувати інформацію про обладнання та формувати звіти про наявність та ремонт обладнання.

Завдяки цій роботі створено функціональну веб-систему, яка допомагає підприємствам ефективно обліковувати технічне обладнання та оптимізувати процеси управління активами. Робота містить інформацію про основні можливості та принципи роботи цієї системи.

На підставі проведеного аналізу літератури, присвяченої темі інформаційних систем обліку речового майна, можна зробити наступні висновки:

1. Інформаційні системи займають ключове місце в забезпеченні інтеграції процесів обліку, контролю та управління на підприємстві. Вони сприяють єдності та координації роботи всіх відділів та підрозділів підприємства.

2. Основними функціями інформаційних систем обліку є накопичення, збереження, опрацювання та подання інформації про речове майно на підприємстві. Це допомагає забезпечити актуальну та достовірну інформацію для прийняття рішень.
3. В літературі існують дослідження, присвячені методологічним аспектам побудови та функціонування інформаційних систем. Це важливий напрямок для розвитку сучасних інформаційних технологій.
4. Існує чітка тенденція до розробки спрощених систем обліку та звітності, спеціально орієнтованих на потреби малих підприємств. Це відповідає реаліям ринкової економіки та сприяє підтримці малих бізнесів.
5. В аналізі літератури також розглянуто конкретні розробки інформаційних систем для обліку обладнання та речового майна на підприємствах, що свідчить про актуальність цієї теми та постійний розвиток сфери інформаційних технологій.

Отже, аналіз літератури надав глибокого розуміння сутності інформаційних систем обліку речового майна, їх функцій, методологічних аспектів та конкретних розробок. Цей аналіз є важливим кроком для подальшого дослідження цієї теми та сприяє розвитку сучасних підходів до управління обліком речового майна на підприємствах.

## **1.2 Особливості обліку речового майна в військовій галузі**

Облік речового майна в військовій галузі має свої особливості, які визначаються специфікою цієї діяльності та вимогами забезпечення обороноздатності країни. Основні особливості обліку речового майна в військовій галузі включають:

1. Категоризація майна. Все речове майно військового призначення підлягає категоризації в залежності від його важливості, призначення та відповідальності за нього. Одні речі можуть бути віднесені до "критичних",

інші до "загального призначення", що впливає на порядок зберігання, обліку та технічного обслуговування.

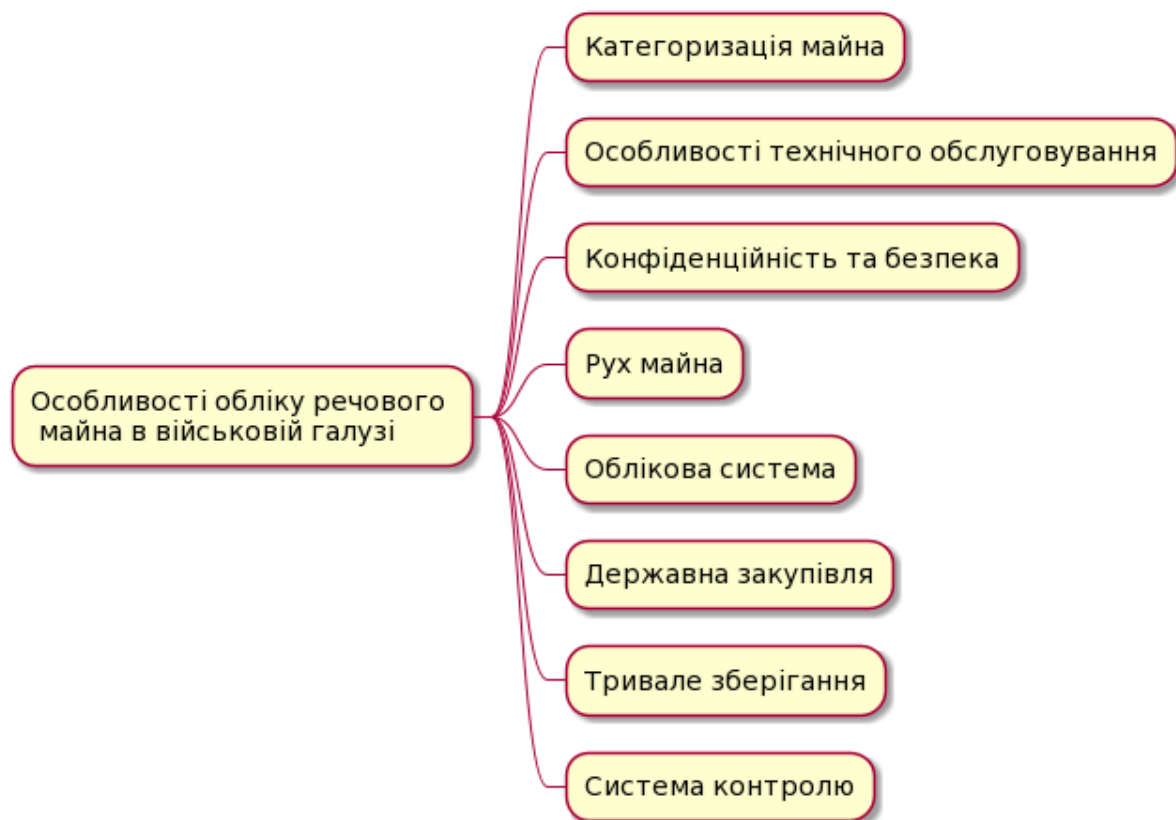


Рисунок 1.1 – Вікно програми Kickidler

2. Особливості технічного обслуговування. Військове майно часто вимагає спеціалізованого технічного обслуговування та ремонту, і ці процедури повинні відповідати вимогам військових стандартів та технічної документації.
3. Конфіденційність та безпека. Деяке військове майно може містити конфіденційну інформацію або бути важливими об'єктами для забезпечення національної безпеки. Отже, облік та збереження такого майна повинні враховувати вимоги до збереження державної таємниці та забезпечення безпеки.
4. Рух майна. Військове майно часто переміщується з одного військового об'єкта на інший. Облік руху майна вимагає точності та документування кожного переміщення.
5. Облікова система. Військова галузь використовує свою власну облікову систему, яка може відрізнятися від цивільних стандартів обліку.

6. Державна закупівля. Збирання та реєстрація інформації щодо закупівлі та вартості військового майна відбуваються відповідно до законів та договорів на виробництво військового обладнання.
7. Тривале зберігання. Деяке військове майно, таке як резерви та запаси, може перебувати у довгостроковому зберіганні та потребувати постійного обліку.
8. Система контролю. Органи військового управління мають власну систему контролю за обліком військового майна для забезпечення належного контролю та координації всіх військових ресурсів.

Ці особливості вимагають дотримання строгих вимог і процедур для забезпечення належного обліку та збереження військового майна, щоб забезпечити ефективність та надійність військових операцій.

Постанова Кабінету Міністрів України, видана 4 серпня 2000 року під номером 1225, стосується затвердження Положення про порядок обліку, зберігання, списання та використання військового майна у Збройних Силах України. Це Положення містить загальні норми та правила, які регулюють облік та використання військового майна у Збройних Силах України [8].

Згідно з цією постановою, Міністерство оборони України є центральним органом управління Збройних Сил і визначає порядок закріплення військового майна за військовими частинами на праві оперативного управління, видаючи відповідні акти. Положення також передбачає облік військового майна в кількісних, якісних, обліково-номерних та вартісних показниках за різними службами служби забезпечення органів військового управління та військових частин.

Положення регулює питання зарахування до складу Військово-Морських Сил Збройних Сил кораблів та суден, а також визначає поняття "військового майна" і "військова частина". Воно також визначає поняття "непридатний стан військового майна" та "майно для забезпечення руху опору".

Ця постанова була пізніше змінена і доповнена іншими постановами Кабінету Міністрів України у 2006, 2009, 2022 і 2023 роках, що вказує на постійний розвиток та удосконалення процесів обліку та використання військового майна у Збройних Силах України.



Ця постанова та затверджене нею Положення є важливими правовими актами, які регулюють порядок обліку та використання військового майна в Україні та забезпечують його цільове та ефективне використання у Збройних Силах.

Пункти Положення регулюють порядок обліку військового майна в Збройних Силах України. Основні положення щодо обліку військового майна можна узагальнити наступним чином:

- Облік військового майна ведеться для отримання різноманітної інформації про його наявність, рух, вартість, якісний стан, а також для організації матеріально-технічного забезпечення військових частин та контролю за умовами зберігання.

- Все військове майно, незалежно від його призначення та джерел надходження, підлягає обліку, і цей облік повинен бути своєчасним, достовірним та точним.

- Облік військового майна ведеться як в мирний час, так і в особливий період на різних об'єктах військового (корабельного) господарства, у службах забезпечення, військових частинах та інших військових формуваннях, і він регулюється даною постановою та іншими нормативно-правовими актами.

- Облік військового майна здійснюється шляхом запису даних у відповідних облікових документах, в яких фіксуються кількість, якісний стан, облікові номери, вартість та інші характеристики цього майна.

- Згідно із законодавством, облік військового майна може бути здійснений з урахуванням вимог щодо збереження державної таємниці.

- Облік військового майна включає кількісний і вартісний облік, при цьому вартість визначається за закупівельною ціною або за ціною, зазначеною у відповідних договорах на виготовлення та поставку майна. Кількісний облік проводиться в натуральних показниках, а вартісний – в національній валюті України.

Положення також передбачає категоризацію військового майна за якісним станом, ступенями придатності та іншими характеристиками згідно з державними стандартами.

Порядок організації та ведення обліку військового майна, включаючи визначення категорій та ступенів придатності, визначення вартості майна, інші аспекти обліку, а також визначення відповідальності посадових осіб і порядку

контролю за обліком визначаються нормативно-правовими актами Міністерства оборони України.

Положення також передбачає додатковий облік майна для забезпечення руху опору та облік майна Сил спеціальних операцій Збройних Сил.

Таким чином, облік речового майна в військовій галузі включає в себе категоризацію майна залежно від важливості та призначення, вимоги до спеціалізованого технічного обслуговування та безпеки, обов'язковий облік руху майна між об'єктами, використання власної системи обліку та контролю, а також дотримання специфічних законів і процедур у сфері закупівель та зберігання, що в разі недотримання може погрожувати національній безпеці та ефективності військового майна.

### **1.3 Порівняння сучасних інформаційних систем обліку речового майна**

Порівняння сучасних інформаційних систем обліку речового майна може бути корисним для вашого дослідження, оскільки це допоможе визначити, яка система найкраще відповідає конкретним потребам та вимогам військової галузі.

Сучасні інформаційні системи для обліку речового майна можуть бути представлені в різних формах, включаючи веб-системи і десктопні додатки. Порівняємо декілька таких систем:

***Oracle Asset Tracking*** - це веб-система, призначена для обліку, моніторингу та управління рухом речового майна [9]. Основні характеристики та можливості цієї системи включають:

1. ***Облік активів.*** Система дозволяє точно фіксувати всі активи та рух майна в організації. Це може бути корисним для збереження детальної інформації про кожен актив, включаючи його характеристики, стан, вартість і призначення.
2. ***Моніторинг руху майна.*** Oracle Asset Tracking дозволяє відстежувати рух активів з одного місця до іншого. Це допомагає контролювати локацію активів та зменшити втрати чи невідоме переміщення.

3. *Управління станом активів.* Система дозволяє вести облік та управляти технічним станом активів. Вона може сповіщати про необхідність технічного обслуговування та ремонту.
4. *Документація.* Oracle Asset Tracking допомагає створювати документацію, включаючи звіти та логи. Це сприяє підвищенню прозорості та може використовуватися для внутрішнього аудиту та звітності.
5. *Інтеграція з іншими системами.* Oracle Asset Tracking може бути інтегрована з іншими інформаційними системами в організації, що дозволяє забезпечити взаємодію з іншими процесами, такими як фінанси, обслуговування, інвентаризація тощо.
6. *Безпека даних.* Система надає засоби для забезпечення безпеки та конфіденційності даних, оскільки в інформації про рух майна може міститися конфіденційна і важлива інформація.

Oracle Asset Tracking є корисним інструментом для військових та інших організацій, які мають значний рух речового майна та потребують точного обліку та контролю за ним.

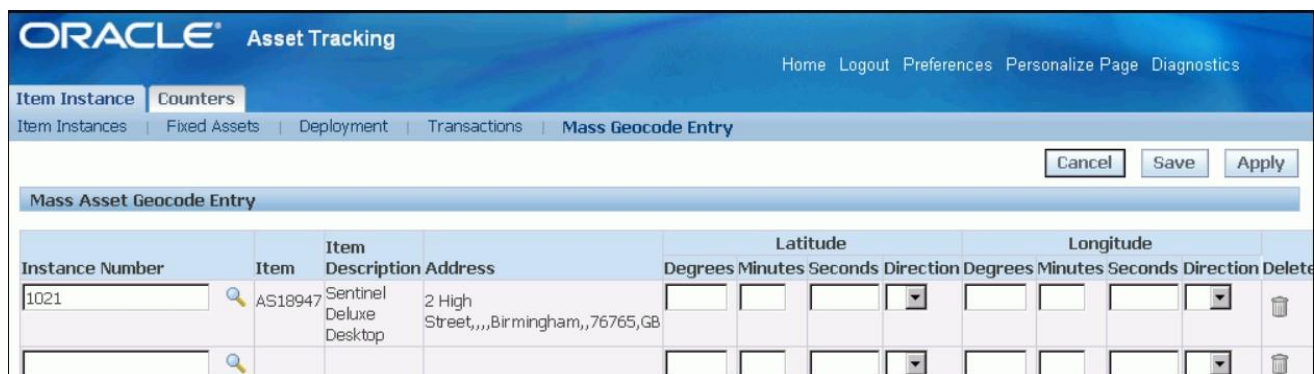


Рисунок 1.2 – Інтерфейс Oracle Asset Tracking

Хоча Oracle Asset Tracking має свої переваги, вона також має деякі недоліки, які варто враховувати:

1. *Вартість.* Ліцензія та впровадження Oracle Asset Tracking можуть бути занадто вартісними, що робить її менш доступною для менших організацій з обмеженими фінансовими ресурсами.

2. *Складність впровадження.* Імплементация системи може бути складною та вимагати спеціалізованого підходу. Вона може займати багато часу та зусиль для налаштування та інтеграції.
3. *Потребує навчання користувачів.* Використання Oracle Asset Tracking вимагає навчання користувачів і адміністраторів, оскільки вона має велику кількість функцій та опцій.
4. *Залежність від Інтернету.* Оскільки Oracle Asset Tracking є веб-системою, вона потребує доступу до Інтернету для роботи. Це може бути проблематичним у віддалених областях або у випадку відмови мережі.
5. *Специфічні вимоги до обладнання.* Використання Oracle Asset Tracking може вимагати специфічного обладнання для моніторингу руху активів, що може бути додатковою витратою.
6. *Не підходить для всіх видів організацій.* Oracle Asset Tracking в першу чергу призначена для підприємств та організацій з великим обсягом руху речового майна. Для менших організацій вона може бути надто великою та складною.

Усі ці недоліки важливо враховувати при виборі Oracle Asset Tracking або будь-якої іншої системи обліку речового майна. Вирішення цих питань може вимагати додаткових ресурсів та планування.

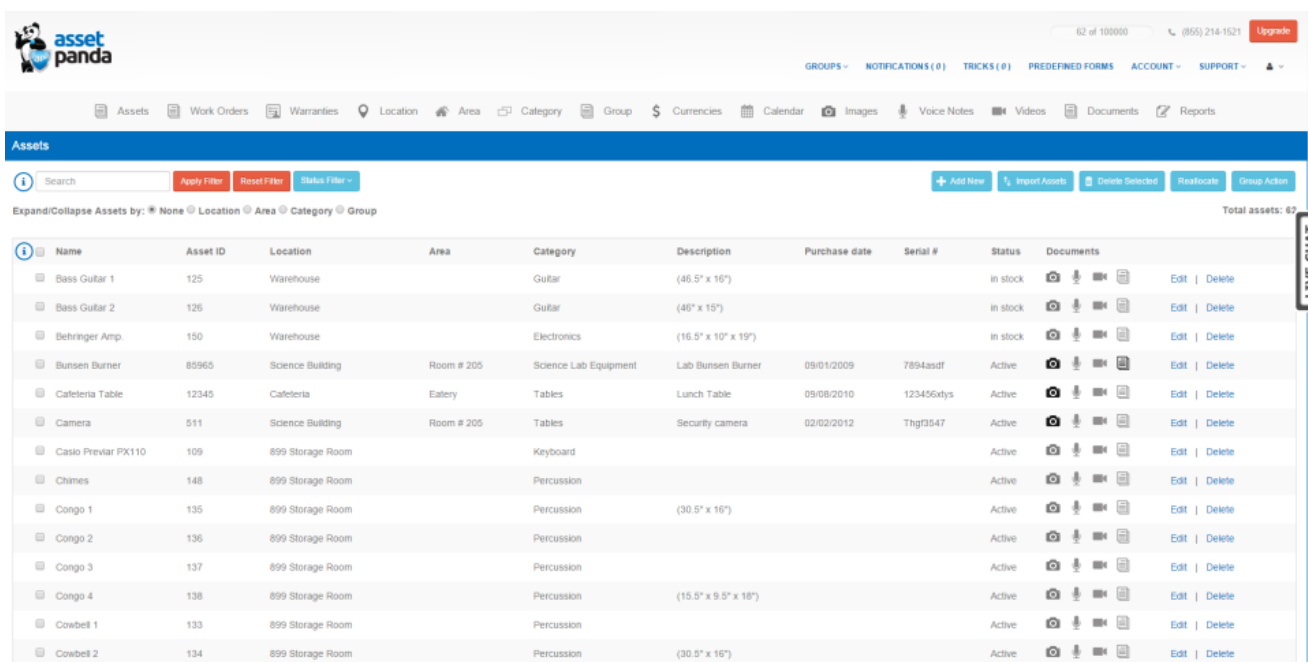
*Asset Panda* - це веб-платформа для обліку речового майна, яка надає розширені можливості для керування активами та рухом майна [10]. Основні характеристики та можливості цієї системи включають:

1. *Облік активів.* Asset Panda дозволяє створювати та оновлювати записи про всі активи організації. Це включає в себе інформацію про тип активу, його параметри, вартість, дату придбання та інші характеристики.
2. *Локація та рух активів.* Система надає можливість відстежувати локацію активів, а також рух майна в реальному часі. Це корисно для знаходження активів та відстежування їхньої історії переміщень.
3. *Технічний стан.* Asset Panda дозволяє вести облік технічного стану активів і планувати технічне обслуговування та ремонт. Відстеження стану активів може

допомогти підтримувати їх в робочому стані та запобігати непередбаченим витратам.

4. *Інтеграція та налаштування.* Asset Panda дозволяє інтегрувати систему з іншими інформаційними системами організації, такими як фінансові системи чи системи управління замовленнями. Платформа також може бути налаштована для відповідності конкретним потребам та процесам організації.
5. *Мобільні додатки:* Asset Panda надає мобільні додатки для Android та iOS, які дозволяють користувачам сканувати штрихкоди або QR-коди активів, оновлювати інформацію та додавати коментарі безпосередньо з мобільних пристроїв.
6. *Звіти та аналітика.* Платформа надає можливість генерувати звіти та аналітичні дані щодо активів та їхнього використання. Це допомагає в прийнятті управлінських рішень та оптимізації ресурсів.
7. *Безпека даних.* Asset Panda приділяє велику увагу захисту даних. Вона забезпечує засоби для збереження конфіденційності та інтегровану систему автентифікації.

Asset Panda є інструментом, який спрямований на покращення ефективності управління активами та рухом майна. Вона дозволяє організаціям забезпечити кращий контроль, ефективність та надійність управління своїм речовим майном.



Name	Asset ID	Location	Area	Category	Description	Purchase date	Serial #	Status	Documents
Bass Guitar 1	125	Warehouse		Guitar	(46.5" x 16")			In stock	Edit   Delete
Bass Guitar 2	126	Warehouse		Guitar	(40" x 15")			In stock	Edit   Delete
Behringer Amp.	150	Warehouse		Electronics	(16.5" x 10" x 19")			In stock	Edit   Delete
Bunsen Burner	85965	Science Building	Room # 205	Science Lab Equipment	Lab Bunsen Burner	09/01/2009	7894asdf	Active	Edit   Delete
Cafeteria Table	12345	Cafeteria	Eatery	Tables	Lunch Table	09/08/2010	123456tys	Active	Edit   Delete
Camera	511	Science Building	Room # 205	Tables	Security camera	02/02/2012	Thg93547	Active	Edit   Delete
Casio Previar PX110	109	899 Storage Room		Keyboard				Active	Edit   Delete
Chimes	148	899 Storage Room		Percussion				Active	Edit   Delete
Congo 1	135	899 Storage Room		Percussion	(30.5" x 16")			Active	Edit   Delete
Congo 2	136	899 Storage Room		Percussion				Active	Edit   Delete
Congo 3	137	899 Storage Room		Percussion				Active	Edit   Delete
Congo 4	138	899 Storage Room		Percussion	(15.5" x 9.5" x 18")			Active	Edit   Delete
Cowbell 1	133	899 Storage Room		Percussion				Active	Edit   Delete
Cowbell 2	134	899 Storage Room		Percussion	(30.5" x 16")			Active	Edit   Delete

Рисунок 1.3 – Інтерфейс Asset Panda

Asset Panda, як і будь-яка інша система, має свої недоліки, які важливо враховувати при розгляді її використання:

1. *Вартість.* Вартість впровадження та підтримки Asset Panda може бути значною, особливо для менших організацій або неприбуткових організацій.
2. *Складність налаштування.* Платформа може вимагати складних налаштувань для відповідності конкретним потребам організації. Це може призвести до затримок у впровадженні.
3. *Навчання користувачів.* Користувачам та адміністраторам може знадобитися час для навчання та адаптації до нової системи.
4. *Залежність від Інтернету.* Asset Panda, як веб-платформа, вимагає доступу до Інтернету. Це може створити проблеми у випадку відмови мережі або в областях з обмеженим доступом до Інтернету.
5. *Не підходить для всіх видів організацій.* Платформа може бути більш вигідною для організацій з великим обсягом активів та рухом майна. Для менших організацій вона може бути занадто великою та складною.
6. *Інтеграція з іншими системами.* Деякі користувачі можуть стикатися з труднощами при інтеграції Asset Panda з іншими інформаційними системами, особливо якщо вони мають вже існуючі програми для обліку активів.
7. *Можливість помилок при скануванні.* Хоча мобільні додатки для сканування штрихкодів допомагають полегшити процес обліку активів, вони також піддаються помилкам при неправильному скануванні або помилковому введенні даних.
8. *Обмежена функціональність у базових планах.* Базові плани Asset Panda можуть обмежувати доступ до деяких розширених функцій, що можуть бути доступні тільки у вищих тарифах.

Всі ці недоліки можуть впливати на вибір Asset Panda та вимагати додаткових ресурсів та уважної підготовки для успішного впровадження та використання системи.

*Microsoft Excel* - це десктопний додаток, який належить до пакету програм Microsoft Office і використовується для створення, редагування та аналізу аркушів



5. *Графічне представлення даних.* Крім таблиць, Excel дозволяє створювати графіки та діаграми для візуалізації даних. Це може бути корисним для аналізу та презентації результатів.
6. *Звіти та аналітика.* Excel дозволяє генерувати звіти та аналітичні дані на основі введених даних. Користувачі можуть створювати різноманітні звіти для внутрішнього аудиту та звітності.
7. *Працює офлайн.* Один з переваг Excel - це можливість працювати з даними офлайн, без потреби в постійному з'єднанні з Інтернетом.

Однак, Excel має деякі обмеження та недоліки:

- *Вартість.* Одним з очевидних недоліків Microsoft Excel є те, що це платний продукт. Користувачам потрібно придбати ліцензію або платити за підписку на Microsoft Office, щоб мати доступ до Excel. Це може стати фінансовим обмеженням для деяких користувачів, особливо для невеликих підприємств або осіб з обмеженими фінансовими ресурсами.
- *Обмежена спільна робота.* У випадку спільної роботи над таблицею Excel, можуть виникати проблеми зі синхронізацією та контролем версій даних.
- *Обмежена масштабованість.* Для великих обсягів даних Excel може стати повільним та непрактичним.
- *Складність.* Ще одним недоліком Microsoft Excel є його складність, особливо при роботі з формулами та функціями. Оскільки Excel має велику кількість можливостей для обчислень та аналізу даних, новачкам може бути важко орієнтуватися та налагодити необхідні формули. Відсутність досвіду або незнання синтаксису може призвести до помилок у розрахунках, що може міцно вплинути на точність обліку речового майна та призвести до невірних результатів.
- *Відсутність автоматизації.* Excel не завжди надає вбудованих можливостей для автоматизації обліку активів та процесів підтримки.

Excel є потужним інструментом для обліку речового майна, але його використання може бути обмеженим для складних завдань та великих організацій, які вимагають більш розширених інструментів управління активами.



**QuickBooks Desktop** - це десктопний пакет програм, розроблений компанією Intuit, який призначений для обліку фінансів та розрахунків у підприємствах [12]. Основні характеристики та можливості QuickBooks Desktop, пов'язані з обліком речового майна, включають:

1. **Облік активів.** QuickBooks Desktop дозволяє вести облік речового майна, включаючи його придбання, вартість та локацію. Користувачі можуть створювати записи про кожен актив та оновлювати інформацію за потреби.
2. **Створення фінансових звітів.** Пакет програм надає засоби для створення різних фінансових звітів, які включають інформацію про речове майно та активи підприємства. Це може бути корисним для внутрішнього аудиту та фінансової звітності перед сторонніми органами.
3. **Синхронізація з банківськими рахунками.** QuickBooks Desktop дозволяє підключити банківські рахунки, щоб автоматично відстежувати транзакції та переміщення коштів, пов'язані з речовим майном.
4. **Керування графіком амортизації.** Пакет програм надає можливість створювати та відстежувати амортизаційні графіки для активів, що дозволяє користувачам розподіляти вартість активів протягом їхнього придатного до використання строку.

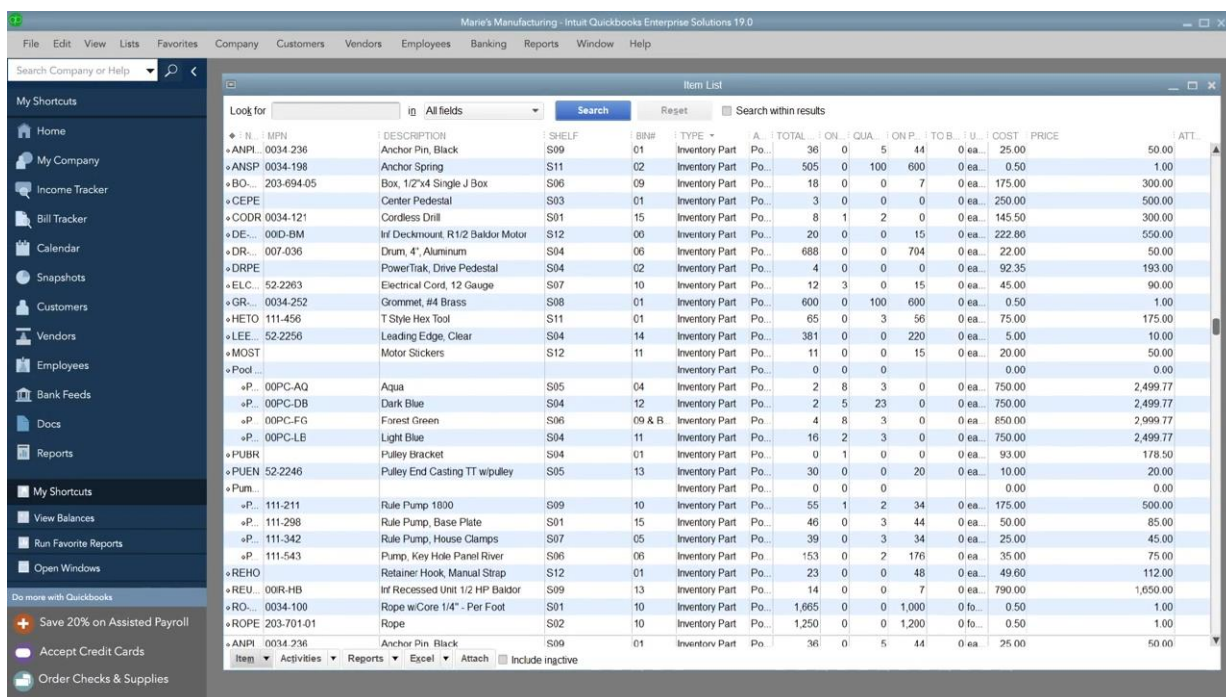


Рисунок 1.5 – Інтерфейс QuickBooks Desktop

5. *Інтеграція з іншими програмами.* QuickBooks Desktop може бути інтегрований з іншими програмами та сервісами, що спрощує обмін даними та забезпечує сумісність з іншими системами.
6. *Доступ для багатьох користувачів.* Пакет програм дозволяє створювати облікові записи для різних користувачів та надавати обмежений доступ до даних відповідно до ролей та обов'язків.

QuickBooks Desktop використовується великою кількістю бізнесів для обліку фінансів та активів, включаючи речове майно. Він дозволяє забезпечити високу точність та ефективність обліку активів підприємства.

QuickBooks Desktop, незважаючи на свої переваги, має деякі недоліки:

1. *Вартість та оновлення.* QuickBooks Desktop - це платна програма, і вартість ліцензій та оновлень може бути високою. Потрібно розглядати витрати на програмне забезпечення та регулярні оновлення.
2. *Потребує технічних знань.* Використання QuickBooks Desktop вимагає певних технічних навичок та знань бухгалтерських процесів. Це може бути важким для користувачів без відповідного досвіду.
3. *Відсутність реального часу.* QuickBooks Desktop оновлює дані не в реальному часі, що може призвести до затримок у відстеженні останніх транзакцій та руху активів.
4. *Залежність від спеціалізованих додатків.* Деякі функції для обліку речового майна можуть вимагати використання спеціалізованих додатків або доповнень до QuickBooks Desktop, що може збільшити витрати та складність системи.

Загальна відсутність деяких сучасних можливостей та обмеженість доступу може зробити QuickBooks Desktop не ідеальним вибором для деяких організацій, особливо для тих, хто потребує більшої гнучкості та мобільності у сфері обліку речового майна.

Розглянувши основні характеристики кожної системи складемо порівняльну таблицю з їх перевагами та недоліками (таблиця 1.1).

Таблиця 1.1. Порівняльна таблиця систем обліку речового майна

Система обліку речового майна	Переваги	Недоліки
Oracle Asset Tracking	Високий рівень функціональності	Висока вартість
	Висока інтеграція з іншими системами	Складне впровадження
	Висока безпека даних	Потреба в постійному підключенні до мережі
	Підтримка великих організацій	
Asset Panda	Гнучкість в налаштуванні	Вартість
	Можливість використання мобільних додатків	Складність налаштування
	Зручність у створенні звітів	Залежність від інтернет-з'єднання
Microsoft Excel	Гнучкість та можливість роботи офлайн	Обмежена спільна робота
	Зручність для створення звітів	Обмежена обробка великих обсягів даних
QuickBooks Desktop	Інтеграція з фінансовими системами	Вартість
	Зручність у створенні фінансових звітів	Вимагає технічних знань
	Доступність для користувачів	Відсутність оновлень даних в реальному часі

Ця таблиця надає загальний огляд переваг і недоліків кожної системи обліку речового майна на основі розглянутих характеристик.

За результатами розгляду різних систем обліку речового майна, головним загальним недоліком, що виділяється, є висока вартість їхнього впровадження та підтримки. Це може бути значним обмеженням для багатьох організацій, особливо для менших підприємств і неприбуткових організацій, які мають обмежений бюджет.

Як реакція на цей недолік, одним із наших завдань буде розробка системи обліку речового майна, яка буде доступною безкоштовно. Така система може стати важливим ресурсом для неприбуткових організацій та інших установ, які працюють на загальне благо.

## 1.4 Проблематика та постановка завдань дослідження

Проблема забезпечення безпеки країни та ефективного використання військових ресурсів є надзвичайно актуальною з точки зору національної безпеки та обороноздатності. Для розв'язання цих проблем важливо мати належний контроль за військовим майном, оскільки воно є важливою складовою військового потенціалу країни.

В контексті дослідження з обліку речового майна в військовій галузі виникають кілька ключових проблем:

1. Забезпечення національної безпеки та обороноздатності вимагає збереження, контролю та готовності до використання військового майна. Належний облік цього майна є важливим для уникнення втрат та його зловживання.
2. Військові ресурси, включаючи військове майно, представляють собою значну частину бюджетних видатків. Оптимальне використання цих ресурсів є важливим для збереження фінансових та матеріальних ресурсів країни. Недоцільне використання військового майна може призвести до зайвих витрат.
3. Військове майно має свої особливості та вимоги, які відрізняються від цивільного сектору. Це можуть бути вимоги до збереження, транспортування, експлуатації та обліку військового майна. Спеціалізований облік необхідний для забезпечення відповідності цим вимогам.

На основі аналізу проблематики дослідження, визначимо основні завдання, які необхідно вирішити під час проведення дослідження. Завдання дослідження слугують основою для структурування та організації дослідницької роботи.

Основні завдання нашого дослідження:

7. Провести огляд і аналіз наявних джерел літератури, що стосуються обліку речового майна в військовій галузі. Це допоможе з'ясувати існуючі підходи, методи та проблеми, пов'язані з цією темою, і визначити напрямки подальших досліджень.

8. Провести порівняльний аналіз існуючих інформаційних систем, які використовуються для обліку речового майна в військовій галузі. Визначити переваги та недоліки кожної системи, та їхню ефективність.
9. Розробити детальний перелік функціональних вимог, які повинна виконувати інформаційна система обліку речового майна в військовій галузі. Це визначить базові вимоги до системи та її можливості.
10. Розробити алгоритм та процедури для збору, зберігання, оновлення та використання даних про речове майно. Врахувати особливості військового обліку та дотримання вимог щодо безпеки даних.
11. Визначити технології для розробки інформаційної системи обліку речового майна військової галузі. Розглянути можливості використання сучасних програмних засобів та підходів до створення програмного забезпечення.
12. Розробити інформаційну систему обліку речового майна в військовій галузі.

Ці завдання стануть основою для подальших досліджень та розробки програмної системи обліку речового майна в військовій галузі.

## **Висновки до розділу 1**

У цьому розділі проведено аналіз літератури за темою обліку речового майна в військовій галузі. Також розглянуто особливості обліку військового майна, порівняно існуючі сучасні інформаційні системи обліку речового майна та визначено проблематику, яку передбачено досліджувати.

За результатами аналізу літератури стало очевидним, що облік речового майна в військовій галузі включає в себе багато складних аспектів, включаючи забезпечення безпеки країни, ефективне використання ресурсів та врахування особливостей військового майна.

Порівняльний аналіз існуючих інформаційних систем обліку речового майна вказує на те, що багато з них не враховують усіх специфічних потреб військової галузі, існують проблеми з вартістю та ефективністю обліку.

Проблематика дослідження визначена наступним чином: розробити інформаційну систему обліку речового майна в військовій галузі, яка враховуватиме особливості цього процесу та забезпечуватиме безпеку та ефективність використання ресурсів. Дослідження також передбачає вивчення функціональних вимог до системи та вибір оптимальних технологій розробки.

Отже, розділ 1 надає теоретичну базу для подальшого дослідження і розробки інформаційної системи обліку речового майна в військовій галузі, яка буде відповідати висунутим вимогам та вирішувати актуальні проблеми в цій сфері.

## РОЗДІЛ 2

### ФУНКЦІОНАЛЬНІСТЬ ТА ВИБІР ТЕХНОЛОГІЙ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ РЕЧОВОГО МАЙНА

#### 2.1 Функціональні вимоги до системи обліку речового майна

Функціональні вимоги - це частина специфікації або документації, яка визначає функції, операції, функціонал або послуги, які повинні бути надані програмному продукту або інформаційній системі. Ці вимоги описують, як система має взаємодіяти з користувачем, іншими системами або обладнанням, і які завдання вона повинна виконувати для досягнення своїх цілей [13].

Функціональні вимоги зазвичай включають в себе перелік конкретних можливостей, які повинні бути реалізовані в системі, і описують їхню поведінку та обмеження. Вони визначають, які операції можуть бути виконані користувачем або іншими системами, які дані можуть бути оброблені, і як система має реагувати на різні вхідні сценарії.

Функціональні вимоги є важливою частиною процесу розробки програмного продукту або інформаційної системи, оскільки вони визначають базовий функціонал, який має бути реалізований і випробуваний для задоволення потреб користувачів та досягнення мети проекту.

Функціональні вимоги до системи обліку речового майна в військовій галузі включають наступні основні елементи та функції:

##### 1. Реєстрація військового майна:

- Можливість додавання нового майна до системи.
- Збереження основних характеристик майна, таких як id, назва виробу, розмір, примітки тощо.
- Визначення категорій, кількості і термін експлуатації майна.

##### 2. Редагування майна:

- Можливість редагування і зміни інформації про майно за його унікальним ідентифікатором (ID).

### 3. Видалення майна:

- Можливість виділення та видалення військового майна із системи.

### 4. Аналіз і звітність:

- Формування графіка, що відображає інформацію про найбільше внесених в систему видів майна.
- Можливість сортування та фільтрації інформації за різними параметрами, включаючи назву та розмір майна.

### 5. Забезпечення безпеки та захисту інформації:

- Захист від несанкціонованого доступу до системи.
- Резервне копіювання даних для запобігання втраті інформації.

### 6. Доступність і підтримка:

- Підтримка надійності та високої доступності системи для безперервного обліку військового майна.
- Можливість технічної підтримки і оновлень системи.

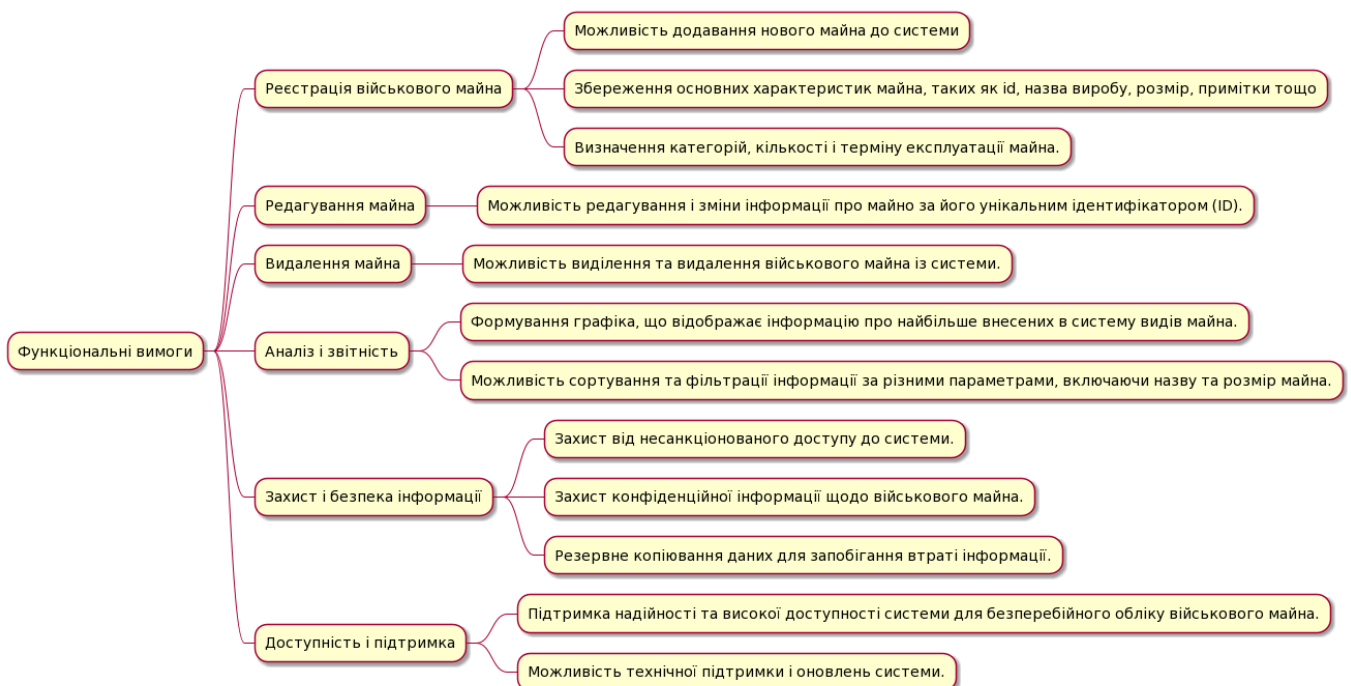


Рисунок 2.1 – Функціональні вимоги до системи обліку речового майна

Ці функціональні вимоги дозволяють системі обліку речового майна відповідати вимогам військової галузі та забезпечувати належний контроль та ефективне використання військових ресурсів.



## 2.2 Алгоритм обліку речового майна

Алгоритм обліку речового майна в військовій галузі - це послідовність кроків і процедур, які визначають, як інформаційна система взаємодіє з військовим майном для ефективного обліку та управління ним.

Алгоритм обліку має наступні елементи:

1. *Реєстрація нового майна.* Цей етап включає в себе процедури додавання нового військового майна до системи. Це може бути здійснено шляхом введення основних характеристик майна, таких як ідентифікатор (ID), назва виробу, розмір, примітки та інші параметри. Визначення категорій, кількості та терміну експлуатації майна також включено до процедури реєстрації.
2. *Редагування і оновлення інформації.* Алгоритм обліку передбачає можливість редагування та оновлення інформації про військове майно за його унікальним ідентифікатором (ID). Це дозволяє користувачам вносити зміни в дані про майно, які можуть змінюватися з часом.
3. *Видалення майна.* Алгоритм обліку містить процедури для виділення та видалення військового майна з системи. Це дозволяє відокремлювати майно, яке більше не використовується або було вилучено зі складу.
4. *Аналіз і звітність.* Для виконання аналізу та створення звітів алгоритм обліку включає в себе формування графіка, який відображає інформацію про найбільш важливі види майна. Також передбачається можливість сортування та фільтрації інформації за різними параметрами, включаючи назву та розмір майна.
5. *Захист і безпека інформації.* В алгоритмі обліку обов'язково враховується захист конфіденційної інформації щодо військового майна від несанкціонованого доступу. Також включає в себе можливість резервного копіювання даних для запобігання втраті інформації.
6. *Доступність і підтримка.* Алгоритм обліку передбачає підтримку надійності та високої доступності системи для безперебійного обліку військового майна. Також включає в себе можливість технічної підтримки і оновлень системи.

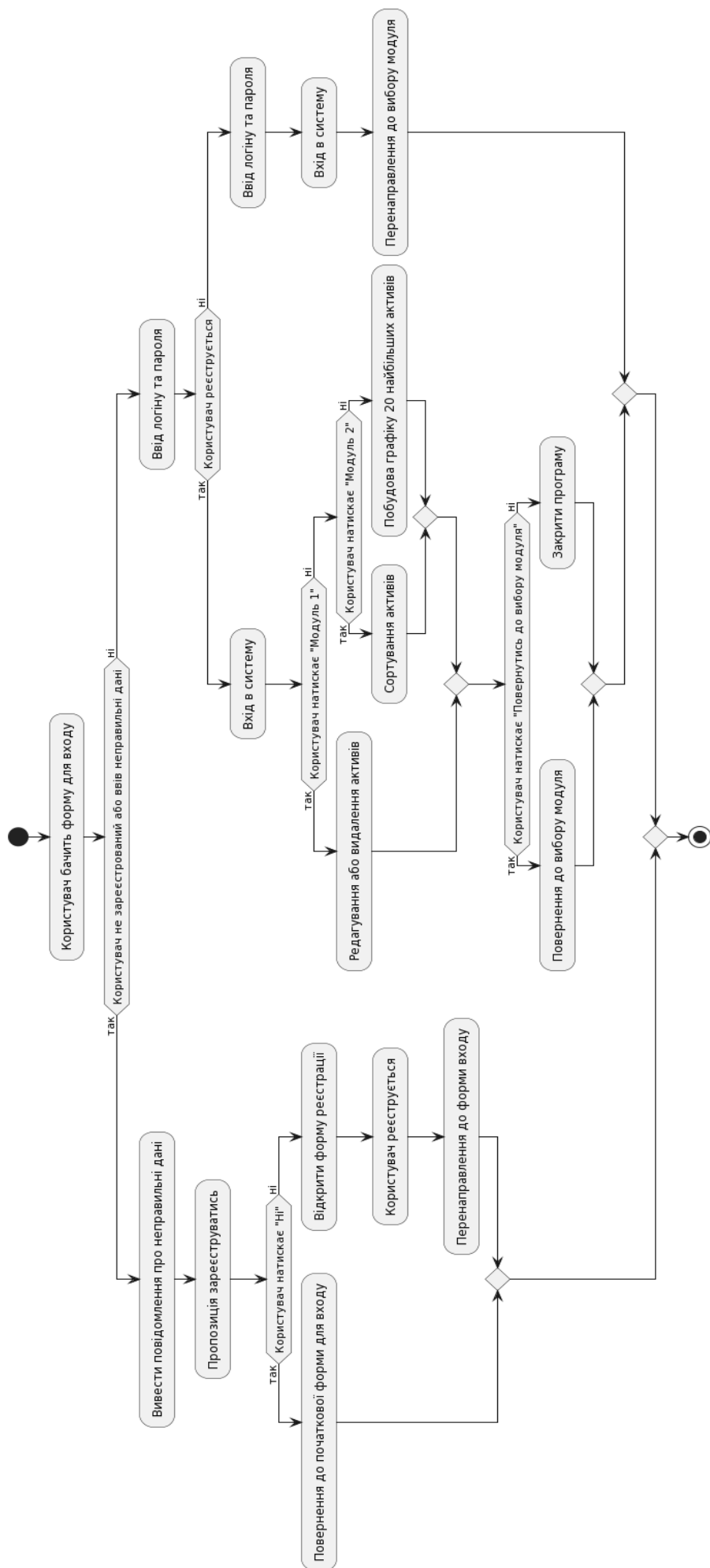


Рисунок 2.2 – Алгоритм обліку речового майна системи

На рисунку 2.2 зображено алгоритм обліку речового майна в програмі яку ми будемо розробляти. В цьому алгоритмі:

1. Користувач відкриває систему і бачить форму для введення логіну та пароля.
2. Якщо користувач не зареєстрований або ввів неправильні дані, система виводить повідомлення про неправильні дані та пропонує зареєструватись.
3. Якщо користувач натискає "Ні", то він повертається до початкової форми для входу.
4. Якщо користувач натискає "Так", то відкривається форма для реєстрації, після чого користувач реєструється і перенаправляється до форми входу.
5. Після реєстрації або входу в систему користувач бачить три модулі:
  - a. Перший модуль дозволяє користувачеві вводити, редагувати або видаляти активи.
  - b. Другий модуль дозволяє користувачеві сортувати активи за зростанням або спаданням будь-якого поля.
  - c. Третій модуль будує графік, який відображає 20 найбільших активів за назвою та розміром.
6. З будь-якого модуля користувач може повернутись до вибору модуля або закрити програму.

Цей код послідовно демонструє взаємодію користувача з системою і дозволяє легко розуміти послідовність дій у нашій системі.

### **2.3 Вибір технологій розробки програмної системи**

Вибір технологій розробки програмної системи є важливим етапом при створенні інформаційної системи обліку речового майна в військовій галузі. Для вдалого вибору технологій розробки слід враховувати ряд аспектів:

- Потрібно обрати тип програмного додатку, це можуть бути веб-системи, десктопні додатки та мобільні додатки.
- Потрібно обрати мови програмування, які найкраще відповідають потребам системи. Деякі мови можуть бути підходящі для веб-розробки (наприклад, JavaScript,

Python, Ruby), тоді як інші можуть бути оптимальними для розробки багатофункціональних десктопних додатків (наприклад, Java, C++, C#).

- Важливо визначити, яка система керування базами даних (СКБД) буде використовуватися для зберігання і керування інформацією про військове майно. Наприклад, можливі варіанти включають MySQL, PostgreSQL, або SQLite.

- Використання фреймворків та бібліотек спрощує розробку, забезпечуючи готові рішення для багатьох завдань. Наприклад, для веб-розробки можна використовувати фреймворки, такі як Django, .Net або React.

- Вибір середовища розробки, інтегрованих середовищ розробки (IDE) та інших засобів для командної роботи, тестування і керування версіями є ключовим.

Почнемо із вибору *типу програмного додатку*.

*Веб-системи.* Веб-системи, або веб-додатки, це програми, які доступні через веб-браузер. Вони призначені для виконання різноманітних завдань, від роботи з електронною поштою до відстеження фінансів.

Веб-системи зазвичай не вимагають установки на локальному комп'ютері, а доступ до них можливий з будь-якого пристрою з доступом до Інтернету. Вони можуть бути створені з використанням різних мов програмування та фреймворків.

*Десктопні додатки.* Десктопні додатки — це програми, які встановлюються на локальному комп'ютері і використовуються безпосередньо на ньому. Вони призначені для виконання різних завдань, включаючи обробку даних, графічний дизайн, офісну роботу тощо.

Десктопні додатки зазвичай мають більшу функціональність і доступ до ресурсів локального комп'ютера. Вони можуть бути розроблені для різних операційних систем, таких як Windows, macOS або Linux.

*Мобільні додатки.* Мобільні додатки, які запускаються на смартфонах і планшетах, розробляються для виконання завдань, що специфічні для мобільних пристроїв, таких як комунікація, навігація, розваги, організація робочих завдань тощо.

Мобільні додатки мають інтерфейс, оптимізований для дотикових екранів та малих екранів смартфонів. Вони можуть бути розроблені для різних операційних систем, таких як iOS та Android, і зазвичай доступні через магазини додатків.

Для нашої системи вибір десктопного додатку є доцільним з ряду причин:

– Наша система вимагає обробки та зберігання великих обсягів даних, що часто пов'язані з військовим майном та безпекою. Десктопний додаток надає більше можливостей для локальної обробки та зберігання даних, що забезпечує більшу ефективність та безпеку.

– Військові об'єкти можуть знаходитися в місцях з обмеженим чи відсутнім доступом до Інтернету. Десктопна програма дозволить користувачам зручно використовувати систему навіть без постійного з'єднання з мережею.

– Десктопний додаток може надавати більше контролю над доступом до даних та забезпечувати вищий рівень безпеки, що важливо для військового майна та інших конфіденційних інформаційних ресурсів.

– Десктопні системи зазвичай мають більшу потужність та можливості для обробки та аналізу даних, що важливо для великих обсягів інформації, характерних для військової галузі.

– Десктопна програма може працювати в офлайн-режимі, що важливо для військових операцій у віддалених або незручно доступних місцях.

З урахуванням цих факторів десктопний додаток відповідає вимогам нашої системи найкраще.

Далі оберемо *мову програмування*.

*Java* є однією з найпопулярніших мов програмування, відомою своєю кросплатформенністю. Вона використовується для розробки різноманітних програм, включаючи веб-додатки, мобільні додатки та десктопні додатки. *Java* базується на віртуальній машині *Java (JVM)*, що дозволяє запускати код на різних операційних системах без змін [14].

*Java* має велику кількість бібліотек, підтримку об'єктно-орієнтованого програмування, велике співтовариство розробників.

Java часто використовується для веб-розробки (зокрема за допомогою фреймворку Spring), створення Android-додатків, розробки серверних додатків та корпоративних систем.

C++ є мовою програмування загального призначення, відомою своєю швидкістю та можливістю низькорівневого програмування. Вона використовується для розробки десктопних додатків, системного програмування та ігор [15].

C++ має високу продуктивність, близькість до мови машинного коду, можливість робити оптимізації для вимогливих за ресурсами задач, об'єктно-орієнтований підхід.

C++ використовується в ігровій індустрії, вбудованому програмуванні, розробці операційних систем, драйверах, графічних програмах та програмуванні для мікроконтролерів.

C# (*C Sharp*) розроблена компанією Microsoft. Вона поєднує в собі можливості C++ та зручність Java, має багато схожостей з Java щодо синтаксису та об'єктно-орієнтованого програмування [16].

C# має високу безпеку завдяки системі управління пам'яттю, підтримку великої кількості бібліотек та інструментів, простота в розробці додатків для Windows.

C# часто використовується для розробки Windows-додатків, включаючи десктопні, мобільні та веб-додатки. Вона також популярна в галузі створення ігор для платформи Xbox.

Для нашої системи ми обираємо мову програмування C# з наступних причин:

- C# розроблена компанією Microsoft та оптимізована для роботи на платформі Windows. Це дозволяє нам створити десктопний додаток, який інтегрується з операційною системою Windows та іншими продуктами Microsoft, що може бути корисним для системи обліку речового майна в військовій галузі.

- C# має багато інструментів та бібліотек, які полегшують розробку додатків для Windows, включаючи інтерфейс користувача, роботу з базами даних та інші аспекти, специфічні для десктопних додатків.

– C# використовує систему управління пам'яттю та має вбудовані механізми безпеки, що робить її відмінним вибором для розробки надійних та стабільних додатків.

– Для C# існує велика кількість інструментів та бібліотек для розробки додатків різного типу, що допоможе нам вибудувати систему обліку речового майна з урахуванням усіх необхідних функціональних вимог.

З огляду на ці переваги та враховуючи специфіку нашого проекту, ми вважаємо C# оптимальним вибором для розробки десктопного додатку обліку речового майна в військовій галузі.

Наступним кроком оберемо **систему керування базами даних**.

*MySQL* - це відкрита реляційна система керування базами даних (RDBMS) з великою ком'юніті та безкоштовною версією. Вона має хорошу продуктивність для середнього обсягу даних та різноманітних завдань [17].

*PostgreSQL* - інша відкрита реляційна СКБД, відома своєю потужністю та розширеннями. PostgreSQL підтримує геопросторові дані та розширення JSON. Вона рекомендується для проектів, де потрібна складна обробка даних [18].

*SQLite* - це легка вбудовувана СКБД з простотою використання та невеликими вимогами до ресурсів. Вона ідеально підходить для десктопних та мобільних додатків, оскільки може бути вбудована в додатки та використовуватися з обмеженими ресурсами. SQLite володіє високим рівнем надійності та можливостями забезпечити безпеку даних [19].

Для нашої системи обліку речового майна в військовій галузі ми обираємо SQLite. Наведемо декілька причин, чому саме SQLite є відмінним вибором для нашого проекту:

– SQLite - це легка СКБД, яка може бути вбудована безпосередньо в наше десктопне додаток. Це дозволяє нам включити базу даних без потреби в окремому сервері чи налаштуваннях.

– SQLite відомий своєю простотою використання та конфігурації. Ця СКБД не вимагає складних налаштувань і легко інтегрується з десктопними додатками.

– SQLite добре працює з невеликими до середніми обсягами даних, що відповідає вимогам нашого проекту. Вона володіє високою швидкістю завдяки вбудованим механізмам оптимізації та кешування.

– SQLite володіє високим рівнем надійності та можливостями забезпечити безпеку даних. Для нашої системи, яка відповідає за облік важливих ресурсів у військовій галузі, це дуже важливо.

– SQLite підтримує мову структурованих запитів SQL, що робить його ідеальним для роботи з реляційними даними.

З урахуванням цих переваг та специфіки нашого проекту, обрання SQLite як СКБД виглядає належним та оптимальним рішенням.

Для розробки десктопного додатку на мові програмування C# проаналізуємо наступні *фреймворки та бібліотеки*:

*Windows Presentation Foundation (WPF)* - це фреймворк для розробки десктопних додатків, спеціально призначених для операційних систем Windows. Він дозволяє створювати багатофункціональні графічні інтерфейси для програм, надаючи розробникам широкі можливості для реалізації креативних дизайнів [20].

*Windows Forms* - це бібліотека для розробки класичних десктопних додатків на платформі Windows [21]. Вона надає стандартні елементи інтерфейсу, такі як вікна, кнопки та тексти, що дозволяє розробникам швидко створювати традиційні програми.

*ASP.NET* - це фреймворк для створення веб-додатків з використанням мови програмування C#. Включає в себе ASP.NET Core, який дозволяє розробляти як веб-додатки, так і десктопні додатки. Це робить ASP.NET потужним інструментом для створення різних типів програм [22].

*Xamarin* - це фреймворк для розробки мобільних додатків для платформ iOS та Android за допомогою мови програмування C# [23]. Він дозволяє створювати кросплатформенні додатки, які можуть працювати на різних мобільних пристроях.

*Unity* - це платформа для розробки ігор та інтерактивних додатків на C#. Вона дозволяє створювати багатофункціональні ігри та симуляції для різних платформ і пристроїв. Unity також підходить для розробки додатків віртуальної реальності та доповненої реальності [24].



Для нашої системи ми обираємо Windows Presentation Foundation (WPF) через наступні переваги:

- WPF надає можливість створення досить ефективного та естетичного графічного інтерфейсу для десктопних додатків. Це особливо важливо в контексті системи обліку речового майна, де користувачі будуть взаємодіяти з великою кількістю даних.

- WPF дозволяє розробникам створювати індивідуальні та унікальні графічні інтерфейси з високим рівнем кастомізації. Це важливо, оскільки система обліку речового майна може вимагати специфічного дизайну.

- WPF і C# добре поєднуються між собою, що робить розробку більш зручною та ефективною. Оскільки ми вже обрали C# як мову програмування для нашого проекту, використання WPF буде логічним вибором.

- WPF дозволяє створювати динамічні графічні інтерфейси з анімацією та можливістю відслідковувати події. Це може бути корисним для реалізації функцій обліку та взаємодії з речовим майном в системі.

З цими перевагами WPF відповідає вимогам нашого проекту, і ми можемо досягти високої якості та зручності інтерфейсу для користувачів системи обліку речового майна в військовій галузі.

Останнім кроком оберемо *середовище розробки*.

*Visual Studio* є однією з найпоширеніших та потужних інтегрованих середовищ розробки (IDE) для C# та WPF. Воно надає широкий спектр інструментів та підтримку для створення десктопних додатків на основі технології WPF. *Visual Studio* дозволяє легко створювати, налагоджувати та відлагоджувати програми, надає доступ до різних ресурсів та бібліотек, а також підтримує контроль версій [25].

*Visual Studio Code (VS Code)* - це безкоштовне та легке середовище розробки, яке підтримує роботу з мовою програмування C# та технологією WPF. Хоча воно має менше функцій, ніж *Visual Studio*, *VS Code* відзначається простотою використання та незалежністю від платформи, що робить його популярним серед розробників [26].

*JetBrains Rider* - це інша популярна IDE для розробки на мові програмування C#. Вона надає широкі можливості для розробки десктопних додатків, включаючи підтримку технології WPF. Rider відомий своєю продуктивністю та інтегрованими інструментами для налагодження коду [27].

Ми обираємо Visual Studio для нашої системи обліку речового майна в військовій галузі з декількох ключових причин:

- Visual Studio надає відмінну підтримку мови програмування C# та технології Windows Presentation Foundation (WPF), які ми обрали для розробки нашої системи. Це робить його ідеальним інструментом для створення десктопних додатків з графічним інтерфейсом.

- Visual Studio - це потужна інтегрована середовище розробки, яке надає широкий спектр інструментів і ресурсів для розробки програм. Ми зможемо легко створювати та налагоджувати нашу систему, використовуючи різні функції та розширення.

- Visual Studio дозволяє легко інтегрувати різні плагіни та розширення, що сприяє розширенню функціональності і пристосуванню середовища до наших потреб.

- Visual Studio має велику та активну спільноту розробників, що означає, що ми зможемо легко знайти підтримку та відповіді на наші запитання.

З урахуванням цих переваг, Visual Studio є ідеальним вибором для розробки нашої системи обліку речового майна в військовій галузі.

Підсумовуючи вибір технологій та інструментів для нашої системи обліку речового майна в військовій галузі, основні рішення включають:

1. **Тип програмного додатку:** Десктопний додаток.
2. **Мова програмування:** C#.
3. **Операційна система:** Windows.
4. **Система керування базами даних:** SQLite.
5. **Фреймворк графічного інтерфейсу:** Windows Presentation Foundation (WPF).
6. **Середовище розробки:** Visual Studio.

Ці вибори здійснюються на підставі потреб та характеристик нашого проекту.

## Висновки до розділу 2

У цьому розділі були визначені основні функціональні вимоги до нашої інформаційної системи обліку речового майна в військовій галузі. Вимоги включають додавання, редагування та видалення майна, а також функції аналізу та звітності, забезпечення безпеки і доступності системи, підтримку користувачів. Ці функції є важливими для ефективного управління військовим майном.

Також визначили послідовність операцій, які користувач виконує в системі. Алгоритм включає кроки, які описують процес входу в систему, вибір дії, виконання цієї дії, і можливість повернутися до вибору дії або завершення роботи.

Проаналізували різні аспекти вибору технологій для розробки нашої інформаційної системи. Визначивши, що наш додаток буде десктопним на платформі Windows, ми обрали C# як мову програмування. Для управління базою даних було обрано SQLite, оскільки вона найкраще відповідає нашим потребам. Фреймворком для графічного інтерфейсу стала Windows Presentation Foundation (WPF). Для розробки вибрано середовище Visual Studio, яке надає всі необхідні інструменти для створення додатку.

Всі ці вибори зроблені з урахуванням особливостей та потреб нашого проекту, що дозволить створити ефективну та зручну інформаційну систему обліку речового майна в військовій галузі.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ РЕЧОВОГО МАЙНА В ВІЙСЬКОВІЙ ГАЛУЗІ

#### 3.1 Архітектура інформаційної системи обліку речового майна в військовій галузі

Архітектура інформаційної системи є важливим етапом при розробці будь-якої програмної системи або додатку. Вона визначає структуру, компоненти, модулі та взаємозв'язки системи, а також розподіл функціональності.

Архітектура визначає загальну структуру системи, компоненти, підсистеми та їх взаємозв'язки. Це допомагає розробникам зрозуміти, як система буде побудована та організована [28].

Вона допомагає визначити, які компоненти відповідають за певні функції системи. Це дозволяє розподілити роботу між різними командами та розробниками, сприяє паралельній роботі та покращує продуктивність.

Архітектура визначає структуру системи, яка сприяє високій якості програми. Це включає в себе питання про масштабованість, надійність, продуктивність, безпеку та інші аспекти якості.

Чітка архітектура допомагає всім учасникам проекту, включаючи розробників, тестувальників, менеджерів та клієнтів, зрозуміти, як система буде працювати та які функції вона надає. Це полегшує спілкування та спільне розуміння.

Гарна архітектура дозволяє додавати нові функції та модифікувати існуючі без необхідності переписування всієї системи. Це полегшує підтримку та розвиток програми.

Визначення архітектури допомагає ідентифікувати можливі ризики та недоліки на ранніх стадіях проекту. Це дозволяє приймати рішення щодо виправлення проблем та забезпечення успішного завершення проекту.

Архітектура інформаційної системи обліку речового майна в військовій галузі включає в себе ряд компонентів та модулів, які взаємодіють для забезпечення функціональності системи. Вона побудована з урахуванням потреб ведення обліку та керування речовим майном, а також забезпечує зручний інтерфейс для користувачів.

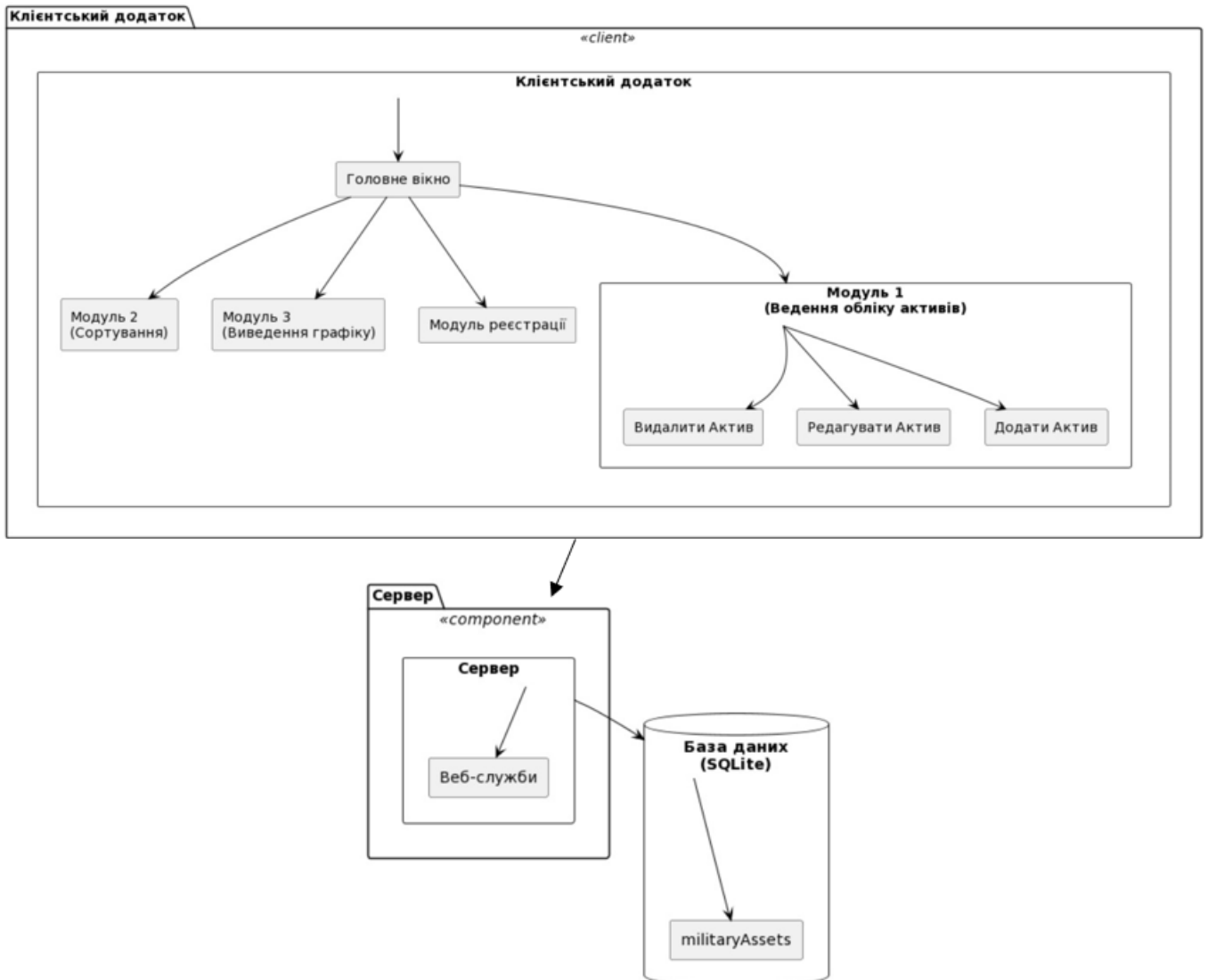


Рисунок 3.1 – Архітектура інформаційної системи обліку речового майна в військовій галузі

Основні компоненти системи включають:

### 1. Клієнтський додаток (Client Application):

- Це десктопний додаток, розроблений за допомогою Windows Presentation Foundation (WPF), який виступає інтерфейсом для користувачів.
- Клієнтський додаток взаємодіє з іншими частинами системи через веб-служби.

## 2. Головне вікно (Main Window):

- Головне вікно є вихідною точкою системи і відповідає за авторизацію користувачів, які мають доступ до системи.
- Після входу користувача в систему, він має можливість обирати модулі системи для виконання певних дій.

## 3. Модуль 1 (Ведення обліку активів):

- В цьому модулі користувач має можливість вести облік активів.
- Модуль містить наступні підмодулі:
  - **Додати Актив (Add Asset):** Користувач може додати новий актив до бази даних.
  - **Редагувати Актив (Edit Asset):** Надає можливість редагувати існуючий актив в системі.
  - **Видалити Актив (Delete Asset):** Користувач може видаляти активи зі списку без необхідності вводити ID.

## 4. Модуль 2 (Сортування):

- В цьому модулі користувач може сортувати активи за різними критеріями, щоб отримати відсортований список активів.

## 5. Модуль 3 (Виведення графіку):

- Модуль, який дозволяє користувачам переглядати графік, що показує кількість активів за іменем виробу та розміром.
- Цей графік відображається на інтерфейсі користувача.

## 6. Модуль реєстрації (Registration Module):

- Це компонент, який відповідає за реєстрацію нових користувачів в системі.
- Нові користувачі можуть отримати доступ до системи після реєстрації.

## 7. Сервер (Server):

- Серверна частина системи включає в себе веб-служби, які реалізовані за допомогою C# і ASP.NET.
- Веб-служби відповідають за зв'язок між клієнтським додатком і базою даних.

- Вони обробляють запити користувачів, виконують операції з базою даних та передають результати на клієнтський додаток.

#### 8. База даних (Database):

- Система використовує базу даних SQLite для зберігання і обробки інформації про речове майно.
- База даних містить таблицю "militaryAssets," в якій зберігаються дані про активи, включаючи ім'я, технічні характеристики, розмір і кількість.

Ця архітектура розроблена так, щоб забезпечити взаємодію між різними компонентами системи та забезпечити користувачам можливість ефективно керувати та відслідковувати речове майно військової галузі через зручний інтерфейс користувача.

Окрім загальної діаграми архітектури інформаційної системи ми ще створимо різні UML-діаграми.

Unified Modeling Language (UML) - це стандартний мовний інструментарій, який використовується для моделювання програмних систем. UML надає набір графічних символів та правил для створення діаграм, які допомагають розробити та аналізувати структуру та поведінку програмних систем [29].

*Use Case Diagram (діаграма використання).* Ця діаграма використовується для моделювання функціональних можливостей системи з точки зору користувачів або зовнішніх сутностей. Вона допомагає ідентифікувати ролі користувачів та їх взаємодію з системою [30].

Діаграма використання містить такі елементи:

- Актори: "Користувач" і "Гість".
- Use Case (випадки використання) групуються в рамці "Use-Case."

Зв'язки:

- "Користувач" має можливість виконати наступні дії:
  - Авторизуватися ("Авторизація").
  - Вибрати модуль для введення активу для:
    - Додавання активу.
    - Редагування активу.

- Видалення активу.
- Сортувати активи.
- Вивести графік.
- "Гість" має можливість виконати реєстрацію ("Реєстрація").

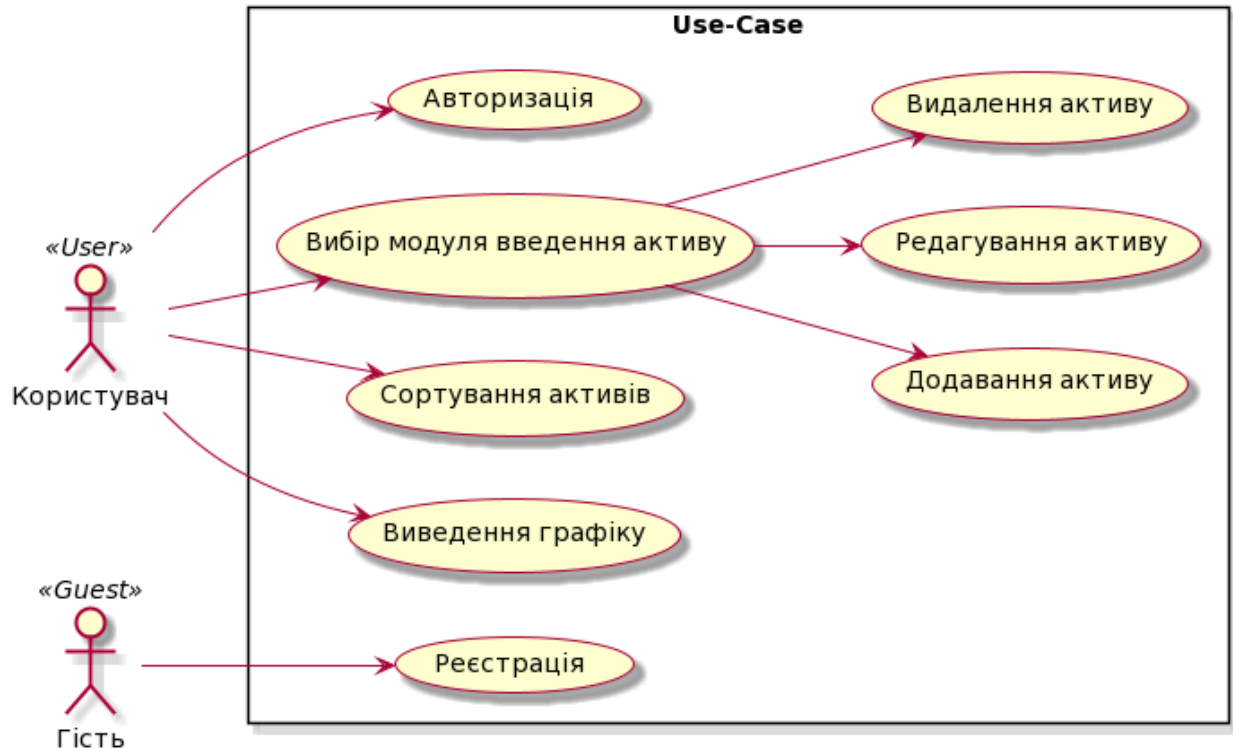


Рисунок 3.2 – Use Case Diagram (діаграма використання) системи

Ця діаграма ілюструє взаємодію користувачів ("Користувач" та "Гість") з системою та вказує на доступні для них функціональності.

**Activity Diagram (діаграма активності).** Ця діаграма моделює діаграму станів або послідовності дій, які виконуються в рамках операцій або процесів системи. Вона допомагає відобразити потік керування та потік даних в процесі виконання [31].

У нашій діаграмі активності показана послідовність дій, які виконує користувач під час взаємодії з модулем "Module1" системи обліку речового майна. Кроки в цій діаграмі включають таку послідовність:

1. Користувач починає взаємодію з модулем "Module1", запускаючи його (позначено ключовим станом **start**).
2. Після ініціалізації модуля, система звертається до бази даних і отримує дані про активи (позначено оператором : **Отримати дані з бази даних;**).



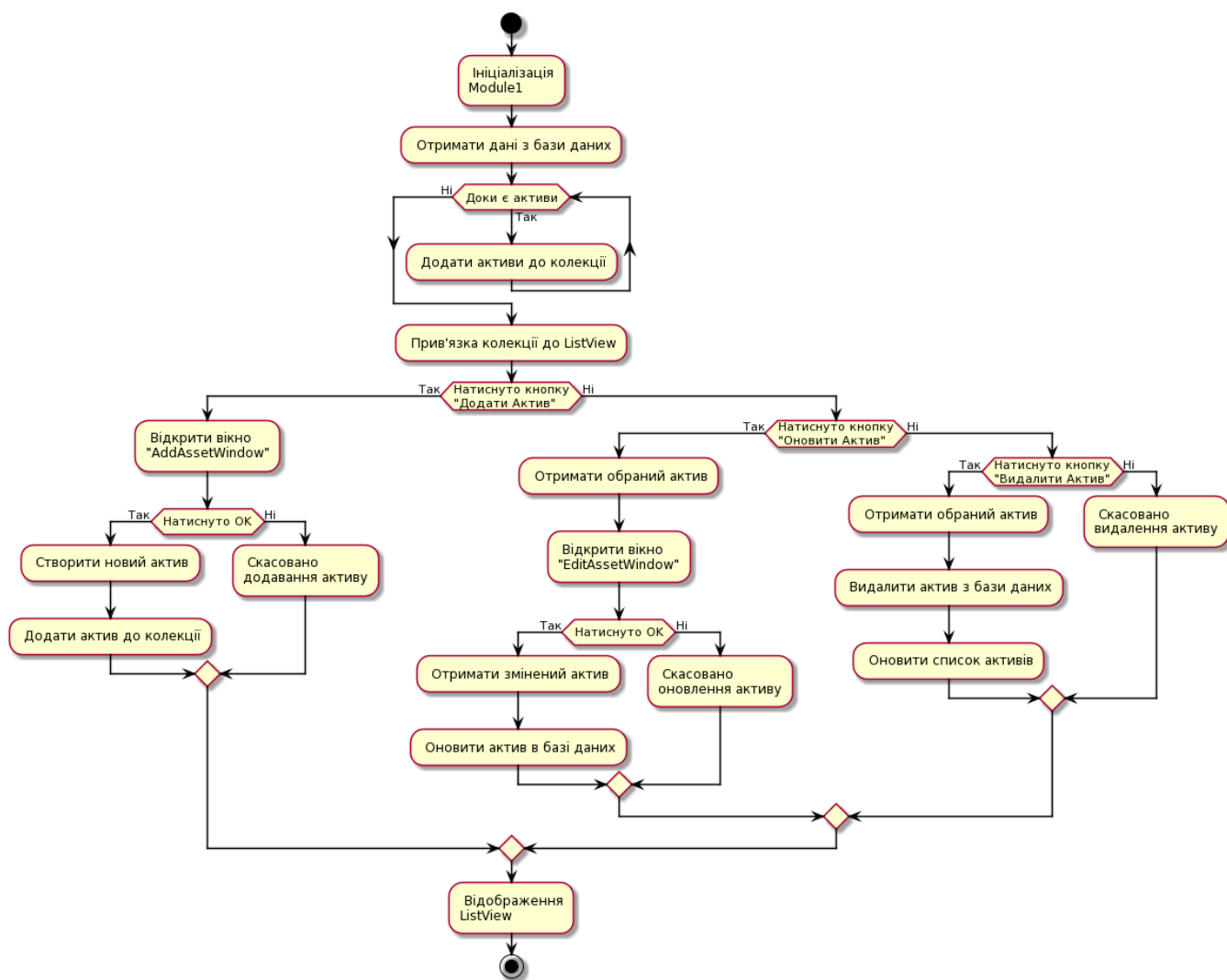


Рисунок 3.3 – Activity Diagram (діаграма активності) системи

3. Далі виконується цикл, де система перевіряє, чи є активи для обробки. Якщо активи є, то вони додаються до колекції (позначено циклом **while (Доки є активи)**).
4. Користувач може натиснути кнопку "Додати Актив," що викликає відкриття вікна для додавання нового активу. Якщо користувач підтверджує додавання (кнопка "ОК"), то новий актив додається до колекції активів (позначено ліворуч відповідними операціями).
5. Користувач також може натиснути кнопку "Оновити Актив," яка відкриває вікно редагування активу. Якщо користувач внесе зміни і підтвердить їх, то актив оновлюється в базі даних (позначено оператором : **Оновити актив в базі даних;**).

6. Якщо користувач натисне кнопку "Видалити Актив," то вибраний актив видаляється з бази даних (позначено оператором : **Видалити актив з бази даних;**).
7. Усі зміни в колекції активів відображаються у ListView, який відображає активи, і користувач бачить оновлені дані (позначено оператором : **Відображення\nListView;**).
8. Взаємодія користувача з модулем "Module1" завершується (позначено ключовим станом **stop**).

Ця діаграма активності ілюструє основні операції, які може виконати користувач в контексті модуля введення активів "Module1" системи обліку речового майна, включаючи додавання, редагування та видалення активів, а також оновлення відображення на інтерфейсі.

*Діаграма класів* є ще одним із типів діаграм в мові моделювання UML, вона використовується для візуалізації структури об'єктно-орієнтованої програми або системи. Діаграма класів допомагає представити класи, їх атрибути, методи та взаємозв'язки між класами в системі [32].

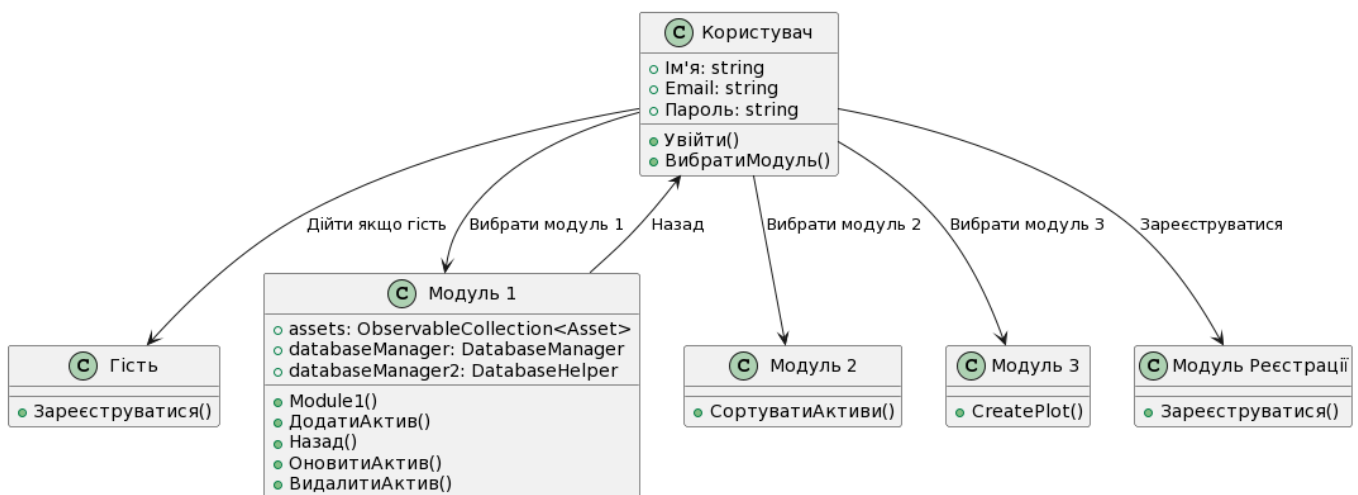


Рисунок 3.4 – Діаграма класів системи

Цей код описує основні класи, методи та взаємозв'язки між ними у нашій інформаційній системі. Кожен клас має свої властивості та методи, а також показано, як користувач взаємодіє з різними модулями. Діаграма включає класи "Користувач," "Гість," "Модуль 1," "Модуль 2," "Модуль 3" і "Модуль Реєстрації".

## 3.2 Розробка модулів для обліку речового майна

Розробка модулів для обліку речового майна - це процес створення окремих компонентів та модулів програмної системи, які відповідають за збереження та обробку інформації про речове майно в рамках військової галузі. Ці модулі спрямовані на ефективний облік, аналіз, редагування та виведення інформації про активи.

Спочатку створимо модуль для авторизації *MainWindow.xaml*. Модуль авторизації - це важлива частина інформаційної системи, оскільки він дозволяє контролювати доступ користувачів до системи та її функціональності.

```

13
14 namespace militarySystem
15 {
16     /// <summary>
17     /// Interaction logic for MainWindow.xaml
18     /// </summary>
19     public partial class MainWindow : Window
20     {
21         public MainWindow()
22         {
23             InitializeComponent();
24         }
25
26         private void TextBox_TextChanged(object sender, TextChangedEventArgs e)
27         {
28         }
29
30
31         private void Button_Click(object sender, RoutedEventArgs e)
32         {
33             string enteredUsername = txtLogin.Text;
34             string enteredPassword = txtPassword.Text;
35
36             // Перевірка користувача в базі даних або іншому джерелі даних.
37             if (IsValidUser(enteredUsername, enteredPassword))
38             {
39                 // Успішний вхід - відкрийте основну форму програми.
40                 SystemWindow mainForm = new SystemWindow();
41                 mainForm.Show();
42                 this.Hide(); // Приховує форму авторизації.
43             }
44             else
45             {
46                 MessageBoxResult result = MessageBox.Show("Невірний логін або пароль. Спробуйте ще раз.\n\nБажаєте заре
47                 if (result == MessageBoxResult.Yes)
48                 {
49                     // Відкриття нового вікна для реєстрації
50                     Register Register = new Register();
51                     Register.Show();
52                     // Закриття поточного вікна

```

Рисунок 3.5 – Фрагмент коду модуля авторизації *MainWindow.xaml*

Цей модуль представляє головне вікно програми, яке відповідає за авторизацію користувача і відображення основної форми після успішної авторизації. Ось опис цього модуля:

1. **using Statements:** ця частина включає в себе всі необхідні імпорти для модуля. Наприклад, **using System.Data.SQLite;** імпортує простір імен, пов'язаний з роботою з базою даних SQLite.

2. **MainWindow Class**: цей клас представляє головне вікно програми. Цей клас є похідним від **Window**, що дозволяє створювати вікна WPF додатків.
  3. **Конструктор MainWindow**: конструктор цього класу встановлює інтерфейс вікна, який визначається відповідним файлом XAML.
  4. **Button\_Click**: цей метод викликається, коли користувач натискає кнопку авторизації. Він зчитує інформацію, введену користувачем (логін і пароль), та перевіряє цю інформацію, використовуючи метод **IsValidUser**. Якщо користувач успішно авторизується, він отримує доступ до головного вікна **SystemWindow**. В іншому випадку виводиться повідомлення про помилку.
  5. **GetConnection**: цей метод створює та повертає об'єкт **SQLiteConnection**, який використовується для з'єднання з базою даних.
  6. **IsValidUser**: цей метод перевіряє користувача в базі даних. Він виконує запит до бази даних, щоб отримати збережений хеш паролю для вказаного користувача та порівнює його з хешем введеного паролю, використовуючи метод **PasswordHasher.VerifyPassword**. Якщо паролі співпадають, метод повертає **true**, в іншому випадку - **false**.
  7. **TextBox\_GotFocus**: цей метод обробляє подію "отримання фокусу" для текстових полів логіну і пароля. Якщо користувач клікає на поле введення, текст у полі очищається для спрощення введення даних.
  8. **txtLogin\_TextChanged**: цей метод викликається при зміні тексту в полі логіну.
- Цей модуль відповідає за авторизацію користувача і взаємодію з базою даних SQLite. Після авторизації користувача він отримує доступ до основної форми програми. Якщо авторизація не вдається, користувач має можливість перейти до форми реєстрації.

Далі створимо модуль *Register.xaml* який відповідає за реєстрацію користувачів у програмі та використання бази даних SQLite. Опис цього модуля:

1. **using Statements**: Ця частина включає в себе імпорти для необхідних просторів імен та класів, такі як **System.Text**, **System.Windows**, **System.Data.SQLite**, **System.Text.RegularExpressions** та **System.Security.Cryptography**. Вони

дозволяють використовувати різноманітні функції та класи для опрацювання тексту, відображення вікон та обробки даних.

## 2. **DatabaseHelper Class:** Цей клас відповідає за взаємодію з базою даних SQLite.

- **DatabaseFileName** - константа, що містить ім'я файлу бази даних.
- **GetConnection** - метод, який повертає об'єкт **SQLiteConnection**, необхідний для підключення до бази даних.
- **InitializeDatabase** - метод для створення бази даних та таблиці користувачів, якщо вони не існують.
- **IsValidEmail** та **IsValidPassword** - методи для перевірки правильності введеного електронного листа та паролю відповідно.
- **RegisterUser** - оновлений метод реєстрації користувача. Він перевіряє валідність введених даних, хешує пароль та зберігає користувача у базі даних.

## 3. **Register Class:** Цей клас представляє вікно реєстрації користувача.

- Конструктор **Register** - ініціалізує вікно реєстрації.
- **TextBox\_GotFocus** - обробляє подію "отримання фокусу" для текстових полів введення. Якщо користувач клікає на поле введення, текст у полі очищається для спрощення введення даних.
- **PasswordHasher Class:** Цей вкладений клас відповідає за хешування паролю користувача. Він містить методи **HashPassword** для створення хешу паролю та **VerifyPassword** для перевірки паролю при авторизації.

```
public static bool VerifyPassword(string password, string hashedPassword)
{
    // Конвертуємо введений пароль у байтовий масив
    byte[] passwordBytes = Encoding.UTF8.GetBytes(password);

    // Хешуємо введений пароль
    using (var sha256 = new SHA256Managed())
    {
        byte[] hash = sha256.ComputeHash(passwordBytes);
        string hashedInputPassword = Convert.ToBase64String(hash);

        // Порівнюємо хеш введеного паролю з збереженим хешем
        return hashedInputPassword == hashedPassword;
    }
}
```

Рисунок 3.6 – Код методу VerifyPassword

- **Button\_Click** - метод, який викликається, коли користувач натискає кнопку реєстрації. Він спершу ініціалізує базу даних за допомогою методу **InitializeDatabase**, а потім перевіряє валідність введених даних і викликає метод **RegisterUser** для реєстрації нового користувача.
- **usernameTextBox\_TextChanged** - обробляє подію зміни тексту у полі введення логіну.

Цей модуль дозволяє користувачам реєструватися у програмі, перевіряти правильність введених даних та взаємодіяти з базою даних SQLite для зберігання інформації про користувачів.

Наступним створимо модуль *Module1.xaml*, який представляє вікно для роботи з активами системи. Він взаємодіє з базою даних SQLite для відображення, редагування та видалення активів. Опишемо цей модуль:

**Module1 Class:** Цей клас представляє вікно для роботи з активами системи.

Властивості:

- **assets** - колекція даних про активи, представлена як **ObservableCollection<Asset>**.
- **databaseManager** - екземпляр класу **DatabaseManager**, який використовується для взаємодії з базою даних.
- **databaseManager2** - об'єкт класу **DatabaseHelper**, який може бути використаний для інших функцій бази даних.
- Конструктор **Module1** - ініціалізує вікно та налаштовує його. Він також ініціалізує базу даних, отримує дані про активи з бази даних та відображає їх у **ListView**.

Методи:

- **Button\_Click** - викликається, коли користувач натискає кнопку для додавання нового активу. Він відкриває нове вікно для введення даних про новий актив та додає його до колекції **assets**.
- **BackButton\_Click** - викликається, коли користувач натискає кнопку "Назад". Він відкриває головне вікно програми і закриває поточне вікно.

- **UpdateAssetButton\_Click** - викликається, коли користувач натискає кнопку для редагування вибраного активу. Він відкриває вікно для редагування та оновлює дані про актив після редагування.
- **DeleteAsset\_Click** - викликається, коли користувач натискає кнопку для видалення вибраного активу. Він видаляє вибраний актив з бази даних та оновлює список активів у `ListView`.

Цей модуль дозволяє користувачеві керувати активами системи, додавати нові, редагувати та видаляти існуючі. Він взаємодіє з базою даних для зберігання та оновлення інформації про активи.

Далі розглянемо *Module2.xaml*, який представляє вікно для відображення та сортування даних про військові активи з бази даних. Опис цього модуля:

**Module2 Class:** цей клас представляє вікно для відображення та оновлення даних про військові активи з бази даних.

Властивості:

- **connectionString** - рядок підключення до бази даних SQLite.
- Конструктор **Module2** - ініціалізує вікно та викликає метод **UpdateDataGrid** для відображення початкових даних з бази даних.
- Методи:

**UpdateDataGrid** - викликається для оновлення даних у **DataGrid**. Метод відкриває підключення до бази даних, виконує запит SQL для вибору всіх записів з таблиці **militaryAssets**, отримує дані, та відображає їх у **DataGrid**.

- **UpdateButton\_Click** - викликається, коли користувач натискає кнопку "Оновити". Він просто викликає метод **UpdateDataGrid** для оновлення даних у **DataGrid**.
- **BackButton\_Click** - викликається, коли користувач натискає кнопку "Назад". Він відкриває головне вікно програми (**SystemWindow**) і закриває поточне вікно.

Цей модуль дозволяє користувачеві переглядати та оновлювати дані про військові активи, які зберігаються в базі даних SQLite. Він відображає дані у **DataGrid** та дозволяє користувачеві оновлювати їх, натискаючи кнопку "Оновити".

Також, користувач може повернутися до головного вікна програми, натиснувши кнопку "Назад".

Наостанок проаналізуємо *Module3.xaml*, який представляє вікно для створення графіка, який відображає дані про кількість військових активів за іменем виробу та розміром. Ось опис цього модуля:

**Module3 Class:** цей клас представляє вікно для створення та відображення графіка з даними про кількість військових активів.

Властивості:

- Немає явно визначених властивостей.
- Конструктор **Module3** - ініціалізує вікно та викликає метод **CreatePlot** для створення графіка з даними.

Методи:

- **CreatePlot** - створює графік для відображення даних про кількість військових активів за іменем виробу та розміром. Метод отримує дані викликом методу **GetData**, створює стовпчастий графік (бар-графік) та додає дані до нього. Налаштовує вигляд графіка та відображає його на вікні.
- **GetData** - виконує запит до бази даних SQLite для отримання даних про кількість активів для кожного імені виробу та розміру. Метод повертає список кортежів, де кожен кортеж містить ім'я виробу, розмір і кількість активів.
- **Button\_Click** - викликається, коли користувач натискає кнопку "Назад". Він відкриває головне вікно програми (**SystemWindow**) і закриває поточне вікно.

Цей модуль дозволяє користувачеві відобразити графік з даними про кількість військових активів для кожного імені виробу та розміру. Він використовує бібліотеку TeeChart для створення та відображення графіка на вікні програми.

У ході розробки системи для обліку речового майна було створено кілька модулів для ефективного управління військовими активами. Кожен модуль має свою конкретну функціональність та допомагає вирішувати різні завдання.

Модулі для обліку речового майна дозволяють автоматизувати процеси збору, збереження та аналізу даних. Це значно спрощує завдання обліку та моніторингу великої кількості активів, що призводить до підвищення продуктивності.



### 3.3 Реалізація графічного інтерфейсу користувача

Реалізація графічного інтерфейсу користувача (GUI) є однією з найважливіших частин процесу розробки програмного забезпечення. Графічний інтерфейс визначає, як користувач взаємодіє з програмою, і повинен бути зручним та інтуїтивно зрозумілим.

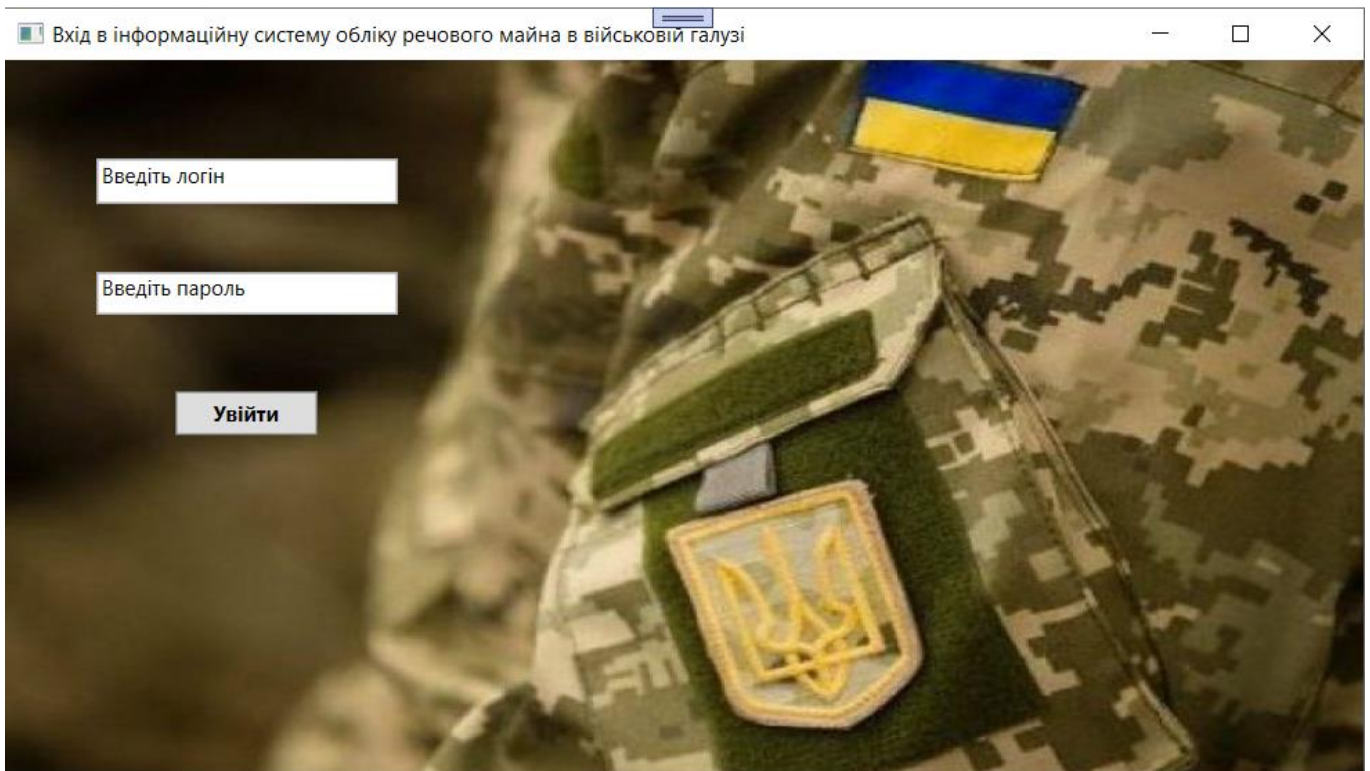


Рисунок 3.7 – Інтерфейс вікна авторизації в додатку

1. **Window**: вікно авторизації додатку є точкою входу в систему, яке має атрибути, такі як заголовок (**Title**), розміри (**Height** і **Width**), жирний шрифт (**FontWeight**), положення при запуску (**WindowStartupLocation**), та показ у панелі завдань (**ShowInTaskbar**).
2. **Grid**: головний контейнер для розміщення всіх інших елементів. У цьому випадку, Grid поділений на шість колонок (**Grid.ColumnDefinitions**), кожна з яких має відмічену ширину. Також Grid має задній фон, встановлений за допомогою **Grid.Background**, де фоном є зображення ("img/1-1-8.jpg").
3. **TextBox**: два текстові поля для введення логіну і паролю. Кожен TextBox має унікальне ім'я (**x:Name**), визначене за допомогою атрибуту **x:Name**. Текстові

поля мають попередні значення, які зникають при отриманні фокусу (**GotFocus**) і змінюють текст після введення (**TextChanged**). Вони розміщені у відповідних колонках Grid.

4. **Button**: кнопка для входу в систему. Має текст "Увійти" та обробник події **Click**, який спрацьовує при натисканні кнопки.

Цей інтерфейс представляє форму для введення логіну та паролю для авторизації користувача в системі. Фонове зображення додає атмосферу до інтерфейсу, і два текстові поля та кнопка надають можливість введення облікових даних та запуску авторизації.

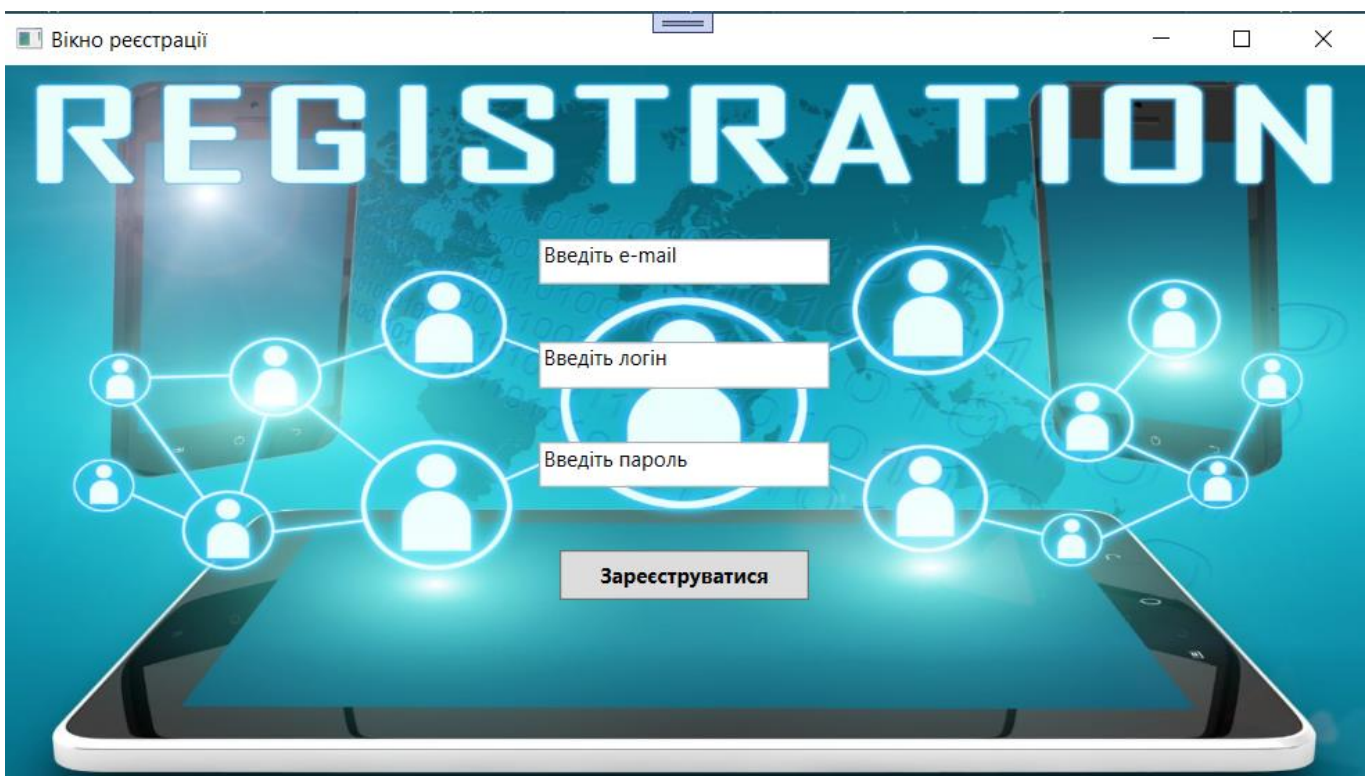


Рисунок 3.8 – Інтерфейс вікна реєстрації в додатку

XAML-код описує інтерфейс вікна реєстрації додатку. Опис елементів інтерфейсу вікна реєстрації:

1. **Window**: вікно реєстрації з атрибутами, такими як заголовок (**Title**), розміри (**Height** і **Width**), положення при запуску (**WindowStartupLocation**). Вікно розміщене по центру екрана.

2. **Grid**: головний контейнер для розміщення всіх інших елементів. У цьому випадку, Grid має задній фон, встановлений за допомогою **Grid.Background**, де фоном є зображення ("img/registration.jpg").
3. **TextBox**: три текстові поля для введення e-mail, логіну та паролю для реєстрації. Кожен TextBox має унікальне ім'я (**x:Name**) і текстове вміст, який зникає при отриманні фокусу (**GotFocus**). Зміни в тексті обробляються за допомогою події **TextChanged** і **TextBox\_TextChanged**.
4. **Button**: кнопка для реєстрації. Має текст "Зареєструватися" і обробник події **Click**, який спрацьовує при натисканні кнопки.

Цей інтерфейс дозволяє користувачеві ввести інформацію для реєстрації в системі. Фонове зображення прикрашає інтерфейс реєстрації, а текстові поля для введення e-mail, логіну та паролю надають можливість ввести облікові дані для створення нового облікового запису. По завершенні користувач може натиснути кнопку "Зареєструватися", щоб завершити процес реєстрації.

Інформаційна система обліку речового майна в військовій галузі — □ ×

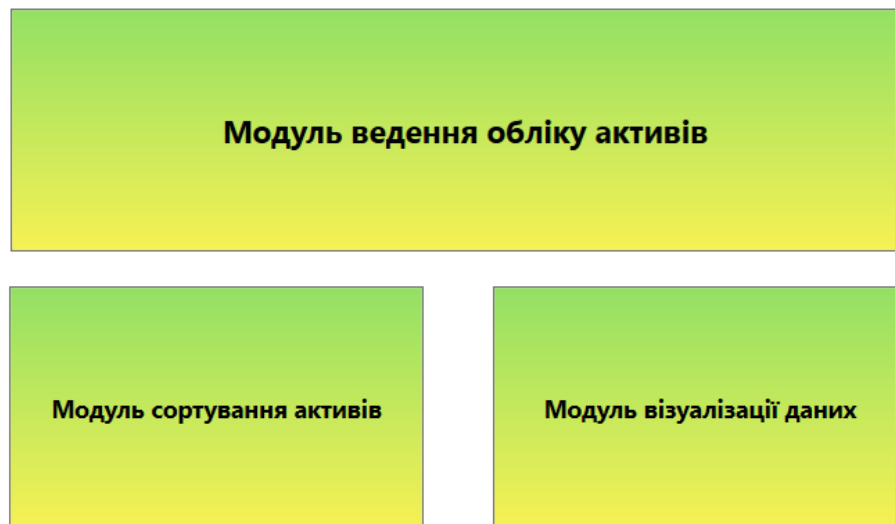


Рисунок 3.9 – Інтерфейс вікна реєстрації в додатку

Опишемо інтерфейс вікна вибору модулів в програмі:

1. **StackPanel**: вертикальний контейнер для розміщення елементів інтерфейсу в стековому порядку. Вміст цього контейнера розміщений по центру вікна.
2. **WrapPanel**: горизонтальний контейнер для розміщення кнопок модулів.

3. **Button**: кнопки для вибору різних модулів системи. Кожна кнопка має власний текст, великий шрифт (**FontSize**), і колірний фон в градієнтному стилі. Кнопки мають обробники подій для взаємодії з відповідними модулями.

- Перша кнопка має текст "Модуль ведення обліку активів" і відповідний обробник події **Button\_Click**.
- Друга та третя кнопки відповідають модулям "Модуль сортування активів" і "Модуль візуалізації даних". Мають свої власні обробники подій (**Module2Button\_Click** та **Module3Button\_Click**, відповідно).

Цей інтерфейс надає користувачу можливість вибору між різними модулями системи. Кожна кнопка відображає модуль та забезпечує зручний доступ до нього.

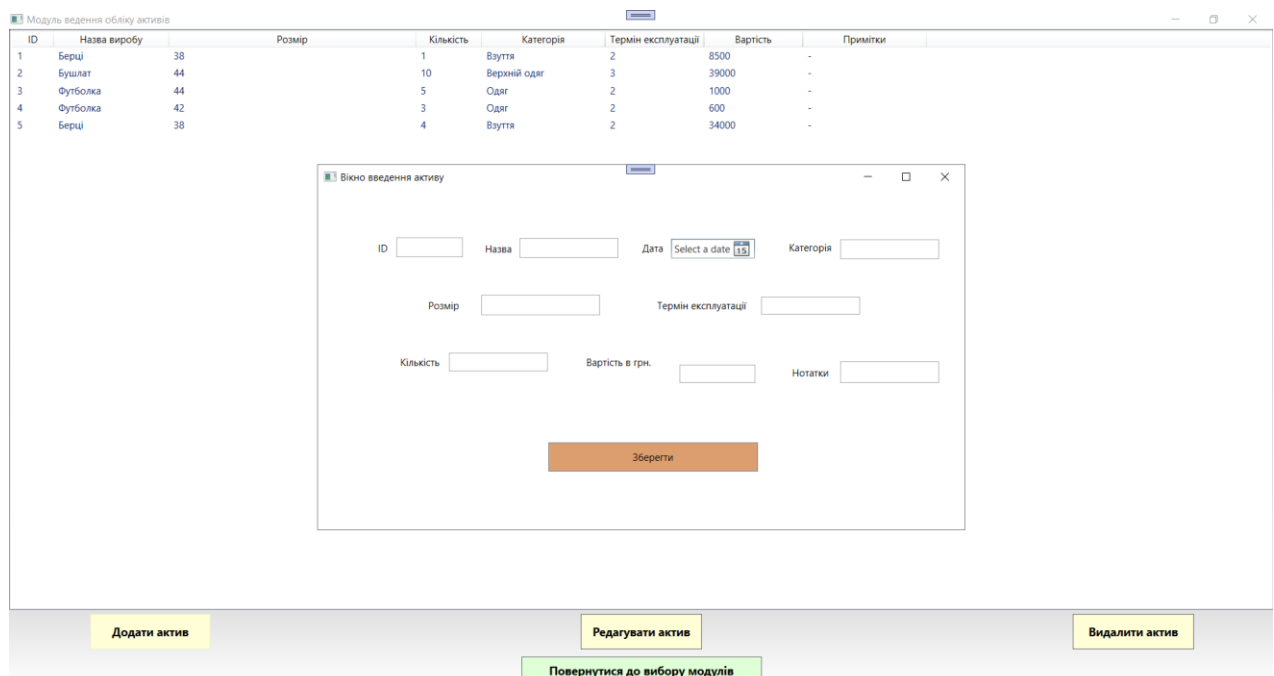


Рисунок 3.10 – Інтерфейс вікна модуля ведення обліку активів

Цей інтерфейс для "Модуль ведення обліку активів" має такі елементи:

1. **Window**: вигляд вікна встановлено на максимізований (Maximized).
2. **LinearGradientBrush**: фон вікна, який є градієнтним від чорного до білого.
3. **ListView**: елемент для відображення списку активів. ListView містить GridView для налаштування стовпців.
4. **GridView**: головний контейнер для створення стовпців у ListView.
  - **GridViewColumn**: кожен стовпець відповідає певному полю активу, такому як ID, Назва виробу, Розмір, Кількість, Категорія, Термін

експлуатації, Вартість та Примітки. Кожен стовпець має власний заголовок і ширину.

5. **Button**: кнопки для взаємодії з активами:

- "Додати актив": Додає новий актив до списку.
- "Редагувати актив": Дозволяє редагувати вибраний актив.
- "Видалити актив": Видаляє вибраний актив зі списку.

6. **BackButton**: кнопка для повернення до вибору модулів. Повертає користувача до головного вікна програми.

Цей інтерфейс дозволяє користувачеві вести облік активів, відображає список активів у вигляді таблиці, і надає зручний спосіб додавання, редагування та видалення активів з системи.

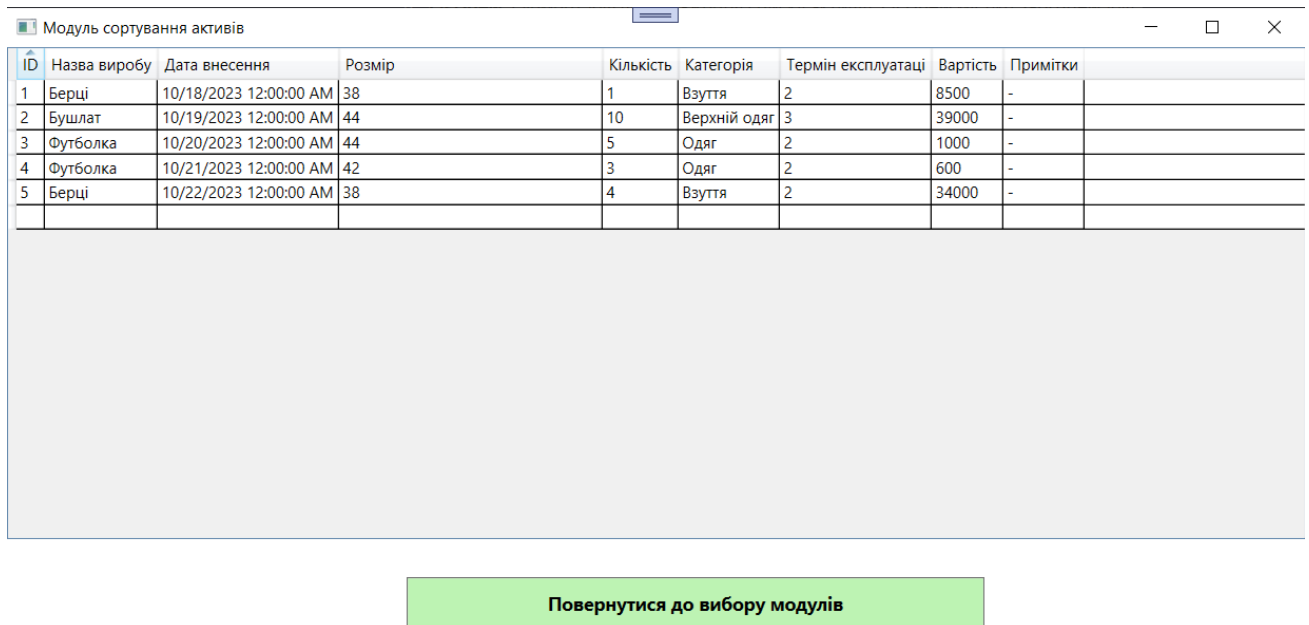


Рисунок 3.11 – Інтерфейс вікна модуля сортування активів

Інтерфейс для "Модуль сортування активів" має такі елементи:

1. **Window**: вікно модуля з атрибутами, такими як заголовок (**Title**), розміри (**Height**, **Width**) та місце відкриття (**WindowStartupLocation**).
2. **Grid**: головний контейнер для розміщення всіх інших елементів.
  - **ColumnDefinition**: визначає дві колонки з різною шириною відсотками екрану.

3. **DataGrid**: елемент для відображення табличних даних. Атрибути включають **AutoGenerateColumns="False"**, що вимикає автоматичне створення стовпців.
  - **DataGrid.Columns**: визначає стовпці в DataGrid. Кожен стовпець відображає певне поле активу.
4. **Button**: кнопка "Повернутися до вибору модулів", яка дозволяє користувачеві повернутися до головного вікна вибору модулів.

Цей інтерфейс дозволяє користувачеві переглядати активи у вигляді табличних даних і використовувати модуль для сортування активів.

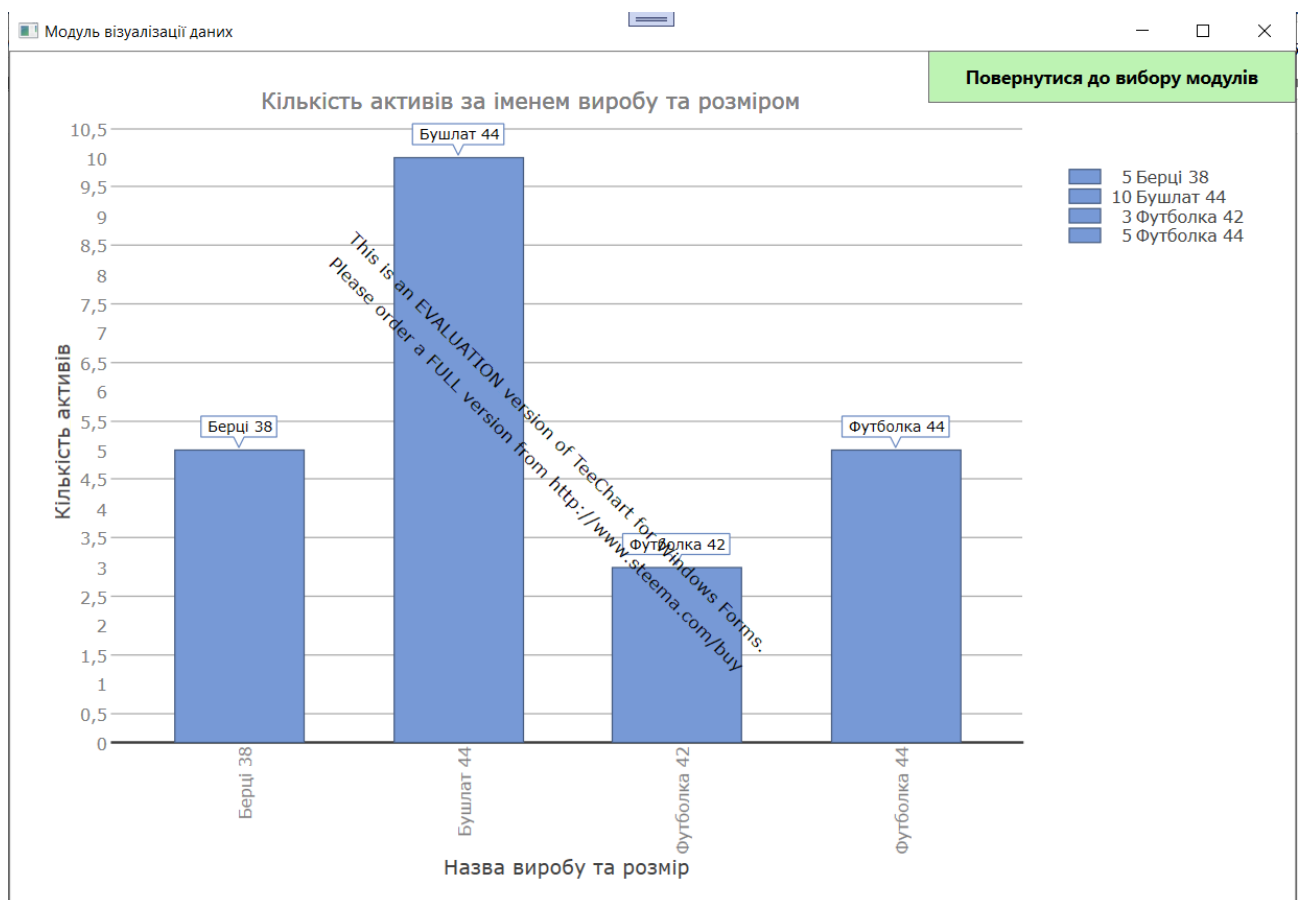


Рисунок 3.12 – Інтерфейс вікна модуля візуалізації даних

Інтерфейс для "Модуля візуалізації даних" системи має такі елементи:

1. **Window.Style**: встановлення стилів вікна, яке використовує **ControlTemplate** для оформлення вікна з границею та кнопкою "Повернутися до вибору модулів".
  - **Grid**: головний контейнер для розміщення всіх інших елементів, включаючи границю вікна і додаткові елементи.



- **Border:** встановлює границю вікна з певними характеристиками, такими як товщина границі та колір.
- **AdornerDecorator:** використовується для оформлення вмісту вікна.

2. **StackPanel:** горизонтальна панель для розміщення додаткових кнопок.

- **Button:** кнопка "Повернутися до вибору модулів", яка дозволяє користувачеві повернутися до головного вікна вибору модулів.

Цей інтерфейс дозволяє користувачеві візуалізувати дані та повертатися до головного вікна вибору модулів за допомогою кнопки "Повернутися до вибору модулів".

Інтерфейс має чітку структуру, добре розміщення елементів і кольорову палітру, що роблять його зручним для користувачів. Ми врахували деталі та ергономіку інтерфейсу, забезпечивши можливість взаємодії з різними модулями та повернення до головного меню. Це допомагає полегшити користувачам навігацію та використання нашої інформаційної системи. У загальному, ми створили хороший та привабливий інтерфейс, який забезпечує комфортне користування системою.

### 3.4 Тестування та валідація програмної системи

Тестування та валідація програмної системи - це два важливих етапи в розробці програмного продукту, спрямовані на перевірку і підтвердження якості та правильності роботи програми. Ці процеси відіграють ключову роль в забезпеченні того, що програмний продукт відповідає вимогам та очікуванням користувачів [33].

**Тестування** - це процес виявлення помилок, дефектів в програмному продукті шляхом активного виконання різних тестів і тестових сценаріїв.

Його мета полягає в перевірці функціональності, якості та безпеки програми, а також виявленні помилок, які можуть призвести до некоректної роботи програми або навіть її аварії.

**Валідація** - це процес перевірки, що програмний продукт відповідає потребам та очікуванням користувача і вирішує їхні завдання та проблеми.

Його мета - це забезпечити, що програмний продукт правильно задовольняє вимогам бізнесу і користувачів, тобто він "робить те, що треба".

### Відмінності між тестуванням та валідацією:

- Тестування зазвичай зосереджене на технічних аспектах, які можна оцінити конкретними тестами та аналізом коду.
- Валідація спрямована на перевірку того, чи задовольняє програма реальні потреби користувачів і бізнес-цілі.
- Тестування виявляє дефекти і проблеми в програмному коді.
- Валідація переконується, що продукт робить те, що потрібно бізнесу та користувачам.

Загалом, тестування та валідація є важливими етапами в процесі розробки програмних систем і допомагають забезпечити якість та відповідність продукту вимогам і очікуванням користувачів.

Під час розробки, наше програмне забезпечення Visual Studio, використовувало автоматичне тестування синтаксису програми. Цей процес відбувався під час редагування або компіляції програми. Якщо система виявляла неправильно написаний код або синтаксичну помилку, вона відразу ж надавала повідомлення про цю помилку в консолі чи в спеціальних вікнах і підсвічувала місце, де виникла помилка в інтегрованому середовищі розробки.

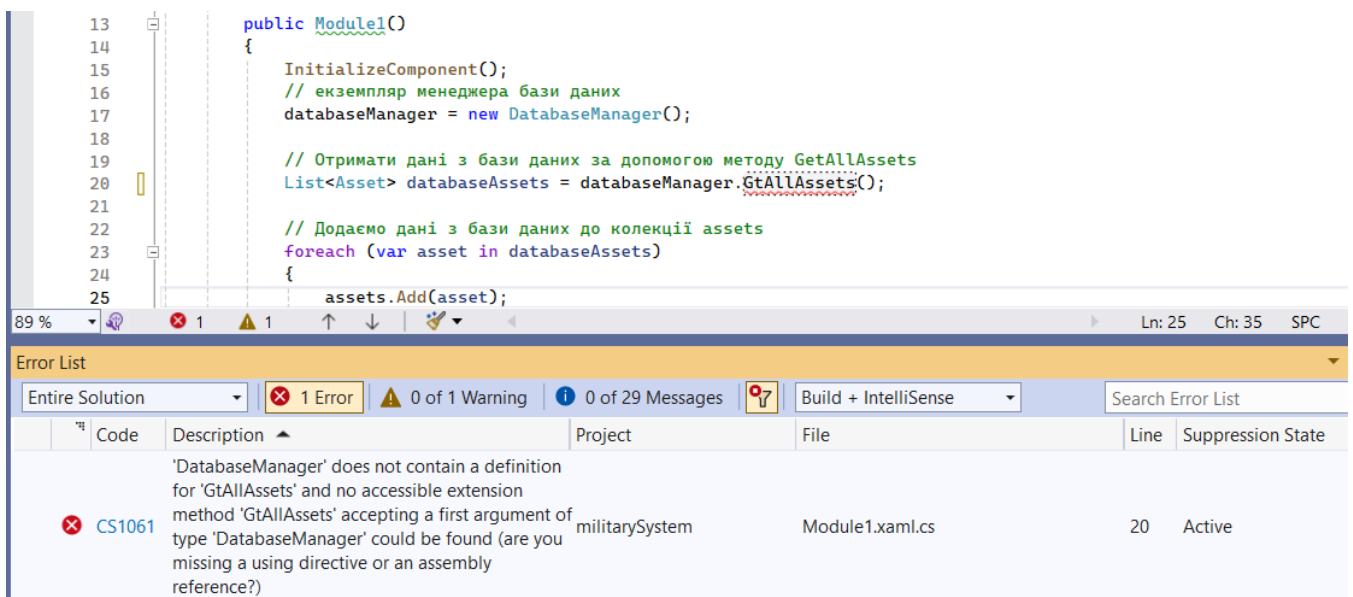
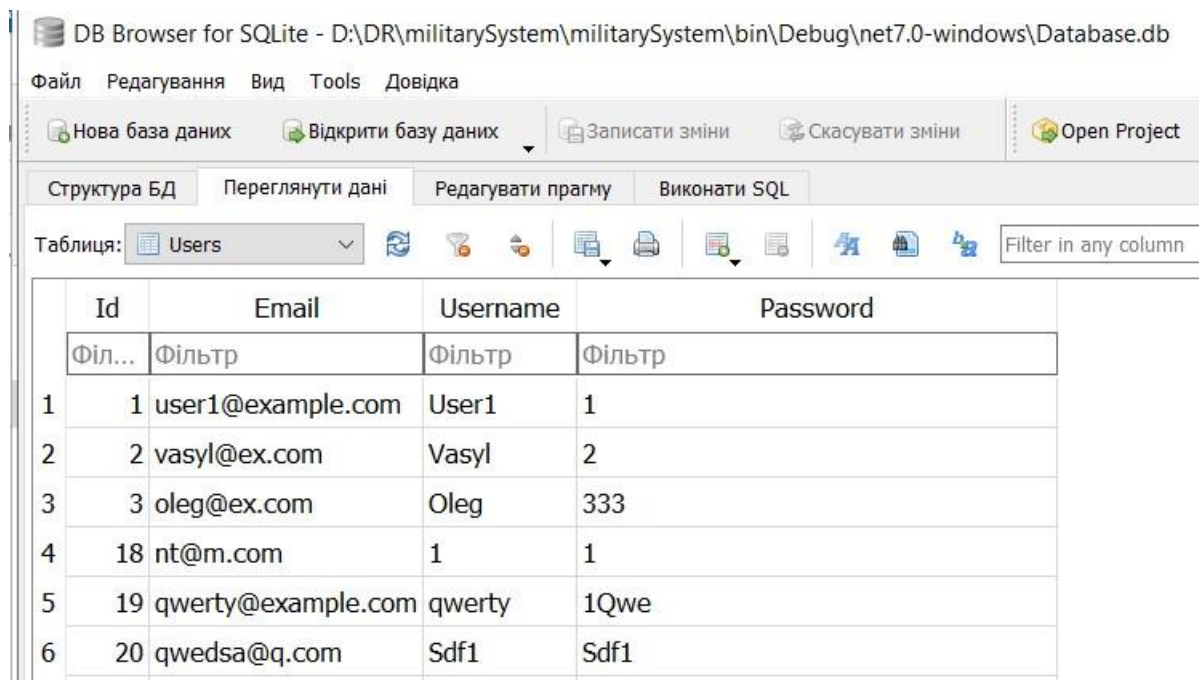


Рисунок 3.13 – Автоматичне тестування синтаксису програми



Цей підхід значно полегшує виявлення та виправлення помилок під час розробки, допомагає підтримувати високий рівень якості програмного забезпечення та відокремлює розробника від необхідності ручного пошуку синтаксичних помилок.

Ручне тестування - це процес перевірки програмної системи або програмного продукту шляхом ручного виконання тестових сценаріїв та аналізу результатів. Під час ручного тестування тести виконуються вручну, без використання автоматизованих тестових інструментів. Тестувальник взаємодіє з програмою, перевіряючи її функціональність та відповідність вимогам.



Id	Email	Username	Password
1	user1@example.com	User1	1
2	vasyl@ex.com	Vasyl	2
3	oleg@ex.com	Oleg	333
4	nt@m.com	1	1
5	qwerty@example.com	qwerty	1Qwe
6	qwedsa@q.com	Sdf1	Sdf1

Рисунок 3.14 – Виявлення дефекту під час ручного тестування

У нашому випадку, під час ручного тестування було виявлено, що паролі користувачів відображаються в незахищеному вигляді, що є серйозною безпековою проблемою. Це означає, що паролі зберігалися або передавалися без жодного шифрування або хешування, і ця інформація могла б бути доступною зловмисникам, якщо б вони мали доступ до бази даних.

Для вирішення цієї проблеми були внесені зміни в код програмної системи, які тепер забезпечують захищене зберігання паролів. Це включає хешування паролів перед зберіганням у базі даних, що робить їх незрозумілими для читання навіть адміністраторами баз даних. Такий підхід підвищує безпеку системи і захищає конфіденційні дані користувачів від несанкціонованого доступу.

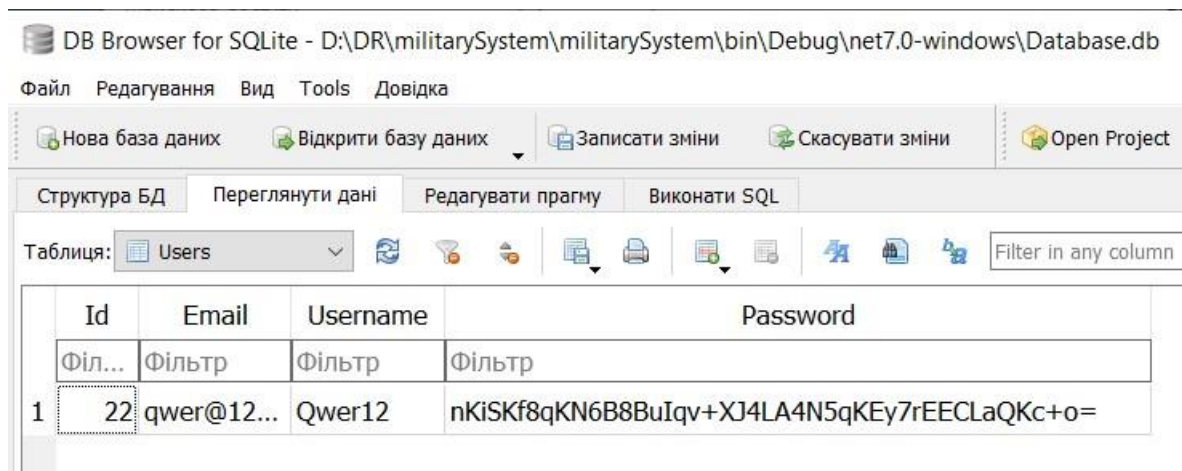


Рисунок 3.15 – Виправлення виявленого дефекту

Тобто, хоч пароль у нас зараз відповідає логіну «Qwer12», але зараз він збережений у захищеному вигляді, як хеш, а не у відкритому тексті.

Ці зміни у коді підвищили безпеку нашої програмної системи, і тепер паролі користувачів залишаються конфіденційними та незрозумілими для всіх, окрім самих користувачів, що зміцнює захист важливих даних і робить систему менш вразливою до атак.

Валідація системи підтвердила, що вона відповідає всім вимогам користувачів, це означає, що програмна система пройшла всі тести і перевірки, і її функціональність, безпека і надійність відповідають очікуванням. Це важливий крок у розробці програмного забезпечення, оскільки він гарантує, що система буде виконувати свої завдання ефективно та безпечно для користувачів.

### Висновки до розділу 3

У розділі 3 "Програмна реалізація інформаційної системи обліку речового майна в військовій галузі" була розглянута інформаційна система, призначена для обліку та управління речовим майном у військовій галузі.

Архітектура інформаційної системи обліку речового майна в військовій галузі дозволила нам розглянути загальну архітектуру інформаційної системи та визначити основні компоненти, які входять до її складу. Ми визначили, що система буде

складатися з трьох основних модулів, призначених для різних завдань, а також бази даних для зберігання інформації про активи.

Розробка модулів для обліку речового майна детально показала розробку кожного з модулів і надала опис їх функціональності. Ми розробили модуль для ведення обліку активів, модуль сортування активів і модуль візуалізації даних. Кожен модуль має власний інтерфейс користувача та здатний взаємодіяти з базою даних.

Реалізація графічного інтерфейсу користувача показала, як був створений графічний інтерфейс для кожного з модулів. Ми використовували мову розмітки XAML для розробки інтерфейсів, і вони були створені з урахуванням зручностей користувача та відповідно до вимог проекту.

Тестування та валідація програмної системи показало, що система була протестована і валідована перед впровадженням. Під час тестування були виявлені певні помилки, зокрема, що паролі користувачів відображалися в незахищеному вигляді. Однак, завдяки внесеним змінам в код, ця проблема була виправлена, що підвищило безпеку системи. Валідація також показала, що система відповідає всім вимогам користувачів і працює стабільно.

У цілому, розділ 3 підкреслює важливість програмної реалізації та тестування в розробці інформаційних систем для обліку речового майна у військовій галузі. Розроблені модулі та інтерфейси користувача відповідають вимогам проекту, а тестування та валідація гарантують їх надійність та безпеку в роботі.

## ВИСНОВКИ

У даній роботі була розроблена інформаційна система для обліку речового майна в військовій галузі. Розглянуті та досліджені теоретичні аспекти, а також функціональні вимоги до системи обліку речового майна. Розглянута можливість використання сучасних інформаційних систем для обліку речового майна та сформульовано постановку завдань для створення інформаційної системи.

У розділі 1 був проведений аналіз літератури та визначені особливості обліку речового майна в військовій галузі. Було зроблено порівняння існуючих інформаційних систем обліку речового майна та поставлені завдання для дослідження.

Розділ 2 включав у себе функціональні вимоги до системи обліку речового майна, розгляд алгоритму обліку речового майна та вибір технологій для розробки програмної системи.

У розділі 3 була розглянута архітектура інформаційної системи та проведена розробка модулів для обліку речового майна. У процесі розробки було розроблено модулі для ведення обліку активів, сортування активів і візуалізації даних. Кожен модуль має власний інтерфейс користувача та взаємодіє з базою даних.

Також, була реалізована графічна частина користувацького інтерфейсу для кожного модуля. Графічний інтерфейс користувача було реалізовано з використанням мови розмітки XAML, з урахуванням зручностей користувача та вимог проекту. Завершальним етапом було тестування та валідація програмної системи, під час яких були виявлені і виправлені деякі помилки, зокрема, викрита проблема відображення паролів користувачів у незахищеному вигляді, яка була успішно виправлена.

Розробка системи була здійснена з урахуванням сучасних технологій розробки програмного забезпечення та функціональних вимог користувачів.

Результатом даної роботи є розроблена інформаційна система, яка дозволить ефективно вести облік речового майна в військовій галузі. Система відповідає вимогам користувачів та забезпечує надійність та безпеку даних.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Маркова, Т. Д., & Пчелянська, Г. Б. (2016). ОСОБЛИВОСТІ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ В СИСТЕМІ ОБЛІКУ ТА КОНТРОЛЮ. *Food Industry Economics*, 8(3). <https://doi.org/10.15673/fie.v8i3.207>
2. О.І.Волот. (2018). Методологічні аспекти побудови, функціонування і розвитку автоматизованих інформаційних систем обліку та управління підприємствами. Головна сторінка IRChNUT. [http://ir.stu.cn.ua/bitstream/handle/123456789/16560/Волот\\_%20Метод\\_аспекти.pdf?sequence=1&isAllowed=y](http://ir.stu.cn.ua/bitstream/handle/123456789/16560/Волот_%20Метод_аспекти.pdf?sequence=1&isAllowed=y)
3. Новіков, В. (2023). СПРОЩЕНА СИСТЕМА ОБЛІКУ І ЗВІТНОСТІ: ІСТОРИЧНИЙ АСПЕКТ. *Grail of Science*, (24), 107–109. <https://doi.org/10.36074/grail-of-science.17.02.2023.013>
4. Gorban, A. E., & Kochyna, M. L. (2016). ІНФОРМАЦІЙНА СИСТЕМА ПЛАНУВАННЯ, ОБЛІКУ, МОНИТОРИНГУ ТА УПРАВЛІННЯ ІННОВАЦІЙНОЮ ДІЯЛЬНІСТЮ В СФЕРІ ОХОРОНИ ЗДОРОВ'Я УКРАЇНИ. *Medical Informatics and Engineering*, (1). <https://doi.org/10.11603/mie.1996-1960.2016.1.5905>
5. Галавська, Л. Є., Прохоровський, А. С., Швиданенко, О. А., Котюх, М. В., & Подоляка, С. А. (2021). Удосконалення речового майна військовослужбовців [Thesis, Національна академія Національної гвардії України]. erKNUTD – Електронний архів Київського національного університету технологій та дизайну. <https://er.knutd.edu.ua/handle/123456789/19036>
6. Мацнєв, І. П., & Даниленко, В. С. (2021). Інформаційна автоматизована система обліку обладнання [Thesis, Чернігів]. Електронний архів Чернігівського національного технологічного університету (IRChNUT). <http://ir.stu.cn.ua/123456789/24982>
7. Грукач, Д. В. (2021). Веб-система обліку обладнання на підприємстві. Електронний архів Чернігівського національного технологічного університету (IRChNUT). <http://ir.stu.cn.ua/123456789/23017>

8. КАБІНЕТ МІНІСТРІВ УКРАЇНИ ПОСТАНОВА N 1225. (2000). Про затвердження Положення про порядок обліку, зберігання, списання та використання військового майна у Збройних Силах. <https://ips.ligazakon.net/document/KP001225?an=29>
9. Understanding Oracle Asset Tracking. (б. д.). Moved. [https://docs.oracle.com/cd/E18727\\_01/doc.121/e13672/T470709T470712.htm](https://docs.oracle.com/cd/E18727_01/doc.121/e13672/T470709T470712.htm)
10. Asset Management Software & Asset Tracking - Asset Panda. (б. д.). Asset Panda. <https://www.assetpanda.com/>
11. Microsoft. (б. д.). Excel. <https://www.microsoft.com/uk-ua/microsoft-365/excel>
12. QuickBooks Desktop, Compare QuickBooks Premier and Enterprise | QuickBooks. (б. д.). QuickBooks®: Official Site | Smart Tools. Better Business. <https://quickbooks.intuit.com/desktop/>
13. Що таке функціональні вимоги: приклади, визначення, повний посібник - Рішення Visure. (б. д.). Visure Solutions. <https://visuresolutions.com/uk/blog/functional-requirements/>
14. Java. (б. д.). Get Java for desktop applications. <https://www.java.com/en/>
15. C++ Language Reference. (б. д.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/uk-ua/cpp/cpp/cpp-language-reference?view=msvc-170>
16. C# docs - get started, tutorials, reference. (б. д.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/uk-ua/dotnet/csharp/>
17. The world's most popular open source database. (б. д.). mysql. <https://www.mysql.com/>
18. PostgreSQL: The World's Most Advanced Open Source Relational Database. (б. д.). PostgreSQL. <https://www.postgresql.org/>
19. SQLite Home Page. (б. д.). SQLite. <https://www.sqlite.org/index.html>
20. What is Windows Presentation Foundation - WPF .NET. (б. д.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/overview/?view=netdesktop-7.0>
21. What is Windows Forms - Windows Forms .NET. (б. д.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/uk-ua/dotnet/desktop/winforms/overview/?view=netdesktop-7.0>

22. ASP.NET | Open-source web framework for .NET. (б. д.). Microsoft. <https://dotnet.microsoft.com/en-us/apps/aspnet>
23. What is Xamarin? - Xamarin. (б. д.). Microsoft Learn: Build skills that open doors in your career. <https://learn.microsoft.com/uk-ua/xamarin/get-started/what-is-xamarin>
24. Unity Real-Time Development Platform. (б. д.). 3D, 2D, VR & AR Engine. <https://unity.com/>
25. *Visual Studio: IDE and Code Editor for Software Developers and Teams.* (б. д.). Visual Studio. <https://visualstudio.microsoft.com/>
26. Microsoft. (2021, 3 листопада). *Visual Studio Code - Code Editing. Redefined.* Visual Studio Code. <https://code.visualstudio.com/>
27. *Rider: The Professional IDE for Xamarin Developers.* (б. д.). JetBrains: Developer Tools for Professionals and Teams. <https://www.jetbrains.com/idea/rider-xamarin/>
28. *Архітектура інформаційних систем.* (б. д.). Pidru4niki. [https://pidru4niki.com/10561127/bankivska\\_sprava/arhitektura\\_informatsiynih\\_sistem](https://pidru4niki.com/10561127/bankivska_sprava/arhitektura_informatsiynih_sistem)
29. *Діаграми UML для моделювання процесів і архітектури проекту.* (б. д.). Evergreen - web розробка і діджиталізація бізнесу за допомогою AI продуктів. <https://evergreens.com.ua/ua/articles/uml-diagrams.html>
30. *Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти.* (б. д.). DOU. <https://dou.ua/forums/topic/40575/>
31. Учасники проєктів Вікімедіа. (2007, 6 грудня). *Діаграма діяльності — Вікіпедія.* Вікіпедія. [https://uk.wikipedia.org/wiki/Діаграма\\_діяльності](https://uk.wikipedia.org/wiki/Діаграма_діяльності)
32. *Діаграми класів – UA5.org.* (б. д.). UA5.org – Матеріали з інформаційних технологій. <https://ua5.org/ooop/392-diagrami-klasiv.html>
33. Авраменко А.С Авраменко В.С Косенюк Г.В. (б. д.). *ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.* Навчальний посібник. <https://eprints.cdu.edu.ua/1482/1/testyvan.pdf>.