

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

Кваліфікаційна робота
на здобуття ступеня бакалавра
спеціальності 122 «Комп'ютерні науки»
на тему:

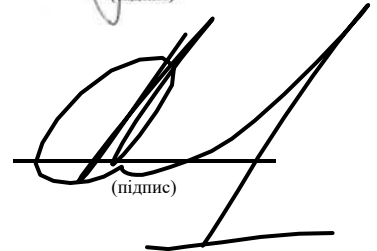
ГЕНЕРАЦІЯ АНАЛІТИЧНИХ ЗВІТІВ ДЛЯ ПІДПРИЄМСТВА

Виконала студентка 4 курсу
Ярошевська Надія Олександрівна

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Омельчук Людмила Леонідівна



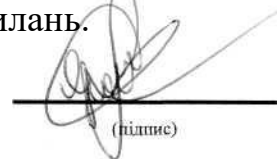
(підпис)



(підпис)

Засвідчую, що в цій роботі немає
запозичень з праць інших авторів
без відповідних посилань.

Студент



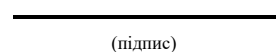
(підпис)

Роботу розглянуто й допущено до
захисту на засіданні кафедри теорії та
технології програмування

«___» _____ 2021 р.,

протокол № ____
Завідувач кафедри

Нікітченко М. С.



(підпис)

Київ – 2021

РЕФЕРАТ

Обсяг роботи 55 сторінок, 54 ілюстрації, 1 таблиця, 15 джерел посилань, 3 додатки.

ЗВІТИ, ОСНОВНІ ЗАСОБИ, C#, DEVEXPRESS, FASTREPORT, SQL, WINDOWS FORMS.

Об'єктом роботи є процес знаходження документів, їх редагування, експорт та генерація звітів за ними.

Предметом роботи є програмний засіб для оброблення інформації і подання її користувачеві.

Метою даного дослідження є проєктування та розробка додатку для користувача, який зможе використовувати його для обліку основних засобів та генерації аналітичних звітів підприємства торгово-промислової групи.

Методи розроблення: розробка власних компонентів для форми, створення бази даних, проєктування моделі даних на основі створеної БД, розробка програмного продукту на основі еволюційної моделі.

Основні інструменти розробки: безкоштовне, вільно поширюване інтегроване середовище розробки Microsoft Visual Studio 2019, програмне забезпечення DevExpress, мова програмування C# та генератор звітів FastReport.

Результати роботи: виконано загальний огляд застосунків, які дозволяють вести облік основних засобів та генерувати звіти, проаналізовано переваги і недоліки їх використання та у процесі розроблено програмний продукт.

Розроблений застосунок може використовуватися користувачами для пошуку потрібного документу, його редагування, експорту та генерації аналітичних звітів за ним.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	7
РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ НА РИНКУ ЗАСТОСУНКІВ	10
1.1 Програма «1С:Бухгалтерія»	10
1.2 Програма «TFORMer SDK»	11
1.3 Сервіс управління торгівлею «Mr.Doc».....	13
1.4 Програма «Power BI»	14
Висновки	16
РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ.....	17
2.1 Обґрунтування вибору мови і системи програмування	17
2.1.1 Платформа .NET Framework.....	17
2.1.2 Мова програмування C#.....	18
2.1.3 Інтегроване середовище розробки Visual Studio і Windows Forms	19
2.1.4 Система управління Microsoft SQL Server та мова запитів SQL ...	20
2.1.5 Генератор звітів FastReport .NET	21
2.1.6 Програмне забезпечення DevExpress.....	22
Висновки	23
РОЗДІЛ 3. ПРИЗНАЧЕННЯ, ВИМОГИ, ЦІЛІ СТВОРЕННЯ ЗАСТОСУНКУ ТА ЙОГО РЕАЛІЗАЦІЯ	24
3.1 Призначення та цілі програми	24
3.2 Вимоги до програми в цілому.....	25
3.3 Технічні вимоги.....	27
3.4 Реалізація застосунку.....	28
3.4.1 Опис шаблонів.....	28
3.4.2 Проєктування застосунку.....	31
3.4.3 Конструювання застосунку.....	33
Висновки	35
РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА	36

4.1 Вкладка «Інвентаризаційна відомість»	36
4.2 Вкладка роботи з документом	47
4.3 Формування звітів	50
Висновки	53
ВИСНОВКИ.....	54
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	56
ДОДАТКИ.....	58
ДОДАТОК А. Діаграма прецедентів.....	58
ДОДАТОК Б. Метод експортування даних в Excel-документ	59
ДОДАТОК В. Метод для друку звіту інвентаризаційної відомості	61

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ADO (ActiveX Data Objects) – прикладний програмний інтерфейс для доступу до даних;

API (Application Program Interface) – прикладний програмний інтерфейс;

ASP (Active Server Pages) – технологія, що дозволяє динамічно формувати вебсторінки з боку вебсервера;

CSV (Comma-Separated Values) – файловий формат;

CTS (Common Type System) – загальна система типів;

DLL (Dynamic Link Library) – бібліотека динамічного компонування;

EF (Entity Framework) – набір технологій;

FIFO (first in, first out) – загальний принцип накопичення та обробки завдань (об'єктів);

HTML (HyperText Markup Language) – мова розмітки гіпертекстових документів;

IDE (Integrated Development Environment) – інтегроване середовище розробки;

iOS (iPhone Operation System) – мобільна операційна система для смартфонів;

JS (JavaScript) – динамічна, об'єктно-орієнтована прототипна мова програмування;

MS SQL (Microsoft SQL Server) – СКБД;

MSIL (Microsoft Intermediate Language) – проміжна мова;

MVC (Model-view-controller) – архітектурний шаблон;

ORM (Object-relational mapping) – технологія програмування;

PDF (Portable Document Format) - міжплатформений відкритий формат електронних документів;

SDK (software Development Kit) – набір із засобів розробки, утиліт і документації;

SQL (Structured Query Language) – мова структурованих запитів;

VCL (Visual Component Library) – об'єктно-орієнтована бібліотека для розробки програмного забезпечення;

WF (Windows Forms) – інтерфейс програмування додатків;

WPF (Windows Form Application) – інтерфейс програмування додатків;

XAML (eXtensible Application Markup Language) – декларативна мова розмітки;

XML (eXtensible Markup Language) – розширювана мова розмітки;

ZPL (Zebra Programming Language) – мова опису сторінки для друку;

БД – база даних;

ЕЦП (Електронний цифровий підпис) – вид електронного підпису;

ІС – інформаційна система;

ООП – об'єктно-орієнтоване програмування;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

СКБД – система керування базами даних.

ВСТУП

Оцінка сучасного стану об'єкта розробки. Наразі завдання обробки та зберігання інформації є дуже важливим. Її втрата або пізнє одержання тягнуть за собою втрату цінних ресурсів. Також існує проблема звітності, оскільки зручніше формувати та зберігати документацію в електронному вигляді. Цим пояснюється такий бурхливий розвиток комп'ютерної техніки, електронних таблиць та систем управління базами даних. Для оперативного, гнучкого та ефективного управління підприємствами, фірмами та організаціями торгово-промислової групи широко впроваджуються системи автоматизованого управління, ядром яких є бази даних.

Актуальність роботи та підстави для її виконання. На даний момент існують застосунки подібної тематики та функціональності, але тема не втрачає своєї актуальності, адже кожен користувач шукає застосунки, на які накладає власні потреби та вимоги.

У наш час жодне з підприємств не обходиться без ведення документації, обліку основних засобів та збереження інформації про них, ось чому вибрана тема залишається актуальною.

Подивимося на ситуацію з боку користувача або ж підприємця, якому потрібно вести облік основних засобів, що його цікавить:

- можливість пошуку, фільтрації, сортування документів;
- редагування документів;
- завантаження інформації;
- генерація звітів.

Розроблюваний застосунок спрямований на те, щоб задовольнити потреби користувача в роботі з документацією.

Об'єктом дослідження є процес знаходження документів, їх редагування, експорт та генерація звітів за ними.

Предметом дослідження є програмний засіб для оброблення інформації і подання її користувачеві.

Мета й завдання роботи. Метою даного дослідження є проектування та розробка застосунку, призначеного для роботи зі звітною документацією підприємства торгово-промислової групи.

Для досягнення цієї мети поставлено такі завдання:

- дослідити існуючі на ринку застосунки для роботи зі звітною документацією;
- визначити інформаційне, організаційне, програмне та технічне забезпечення;
- обрати мову програмування та середовище створення;
- спроектувати шаблони документів для застосунку;
- дослідити різні способи подання інформації та її пошуку;
- розробити інтерфейс та дизайн програмного продукту, призначеного для роботи зі звітною документацією підприємства.

Об'єкт, методи й засоби розроблення. Під час розробки програмного продукту використана еволюційна модель, що базується на наступних принципах:

- додаток розробляється як послідовність блоків структур;
- вимоги встановлюються поступово із уточненням у кожному блоці структури.

Фактично, процес розробки додатку виконується ітераційно, шляхом еволюційного розвитку з отриманням прототипу, на якому здійснюється перевірка реалізації вимог.

Розробляється початкова версія продукту, яка передається кінцевим користувачам для оцінки, після чого продукт доробляється, враховуючи думку замовника. Аналогічно розробляються, передаються й оцінюються проміжні версії програмного продукту, поки не з'явиться повністю готовий продукт, який відповідає всім вимогам замовника. Процеси специфікації, розробки та атестації програмного продукту ведуться паралельно з постійним обміном інформацією між ними.

В якості інструменту створення програмного засобу було обрано Visual Studio 2019 – інтегроване середовище розробки (IDE), яке є вільно поширюваним, з відкритим вихідним кодом та програмне забезпечення DevExpress.

Практичне значення одержаних результатів: спроектований та розроблений застосунок спрощує пошук потрібного документу, надає можливість його відредагувати, дає досить широкий функціонал для роботи з ним та багатокористувацький доступ. Також є можливість експорту вибраного документу та генерації аналітичних звітів за ним.

Практична цінність роботи полягає в розробці ефективного програмного продукту, здатного автоматизувати роботу з аналітичними звітами конкретного замовника, групи компаній Fozzy group.

РОЗДІЛ 1. ОГЛЯД ІСНУЮЧИХ НА РИНКУ ЗАСТОСУНКІВ

На цей час існує багато можливостей для створення звітності роботи підприємства торгово-промислової групи.

У результаті аналізу наявності існуючих рішень, проведеному на ринку, було виявлено, що найпопулярнішими аналогами є програми для ПК. Розглянемо деякі з них.

1.1 Програма «1С:Бухгалтерія»

Система «1С: Бухгалтерія» забезпечує автоматизацію обліку всіх операцій, пов'язаних з надходженням і реалізацією товарів або послуг. Що стосується автоматизації обліку руху грошових коштів, то 1С дозволяє бухгалтерії фіксувати і відслідковувати рух безготівкових і готівкових коштів, обмін валюти [1].

Приклад звіту, створеного в програмі для бухгалтерів «1С:Бухгалтерія» наведено на рис. 1.1.

Выпуск продукции по плановым ценам

Параметры: Начало периода: 01.01.2016 0:00:00
 Конец периода: 02.06.2016 0:00:00
 Организация: МебельСтрой

Организация	Январь	Февраль	Март	Итого
Подразделение	Количество	Плановая стоимость	Количество	Плановая стоимость
Номенклатурная группа	Количество	Плановая стоимость	Количество	Плановая стоимость
Продукция				
Столы		50 000,00		50 000,00
Стол обеденный	10	50 000,00		50 000,00
Стулья		80 000,00		80 000,00
Парта школьная			300	750 000,00
Стул Барный			20	100 000,00
Стул кухонный			1 000	2 000 000,00
Стулья	100	80 000,00		80 000,00
Итого		130 000,00		2 100 000,00

Рисунок 1.1. – Звіт «1С:Бухгалтерія»

«1С: Бухгалтерія» дає можливість вести облік основних засобів, а також нематеріальних активів, причому програмний комплекс автоматично реалізує всі ключові операції обліку:

- надходження;
- передача;
- прийняття до рахунку;
- інвентаризація;
- нарахування амортизації.

«1С: Бухгалтерія» істотно полегшує нарахування заробітної плати працівникам підприємства, дозволяє перерахувати кошти на особовий рахунок співробітників в тому чи іншому банку.

Програма «1С: Бухгалтерія» має певні переваги, такі як:

- наявність готових налаштувань;
- за невеликий період часу, можливо, отримання необхідних документів, з використанням певних форм;
- відсутність необхідності у високій кваліфікації фахівців.

Недоліками програми «1С: Бухгалтерія» є:

- неможливість ведення обліку в динаміці;
- відносно невисока швидкість проведення обчислень;
- для певної діяльності використовують певну версію, тобто програма «1С» не є універсальною [2].

1.2 Програма «TFORMer SDK»

Мультиплатформна програма для складання звітів з підтримкою штрих-кодів: лейаути можуть бути або безпосередньо роздруковані, або збережені в форматах PDF, HTML, PostScript, графіки або ZPL-II [3].

Приклад звіту, створеного в програмі «TFORMer SDK» наведено на рис. 1.2.

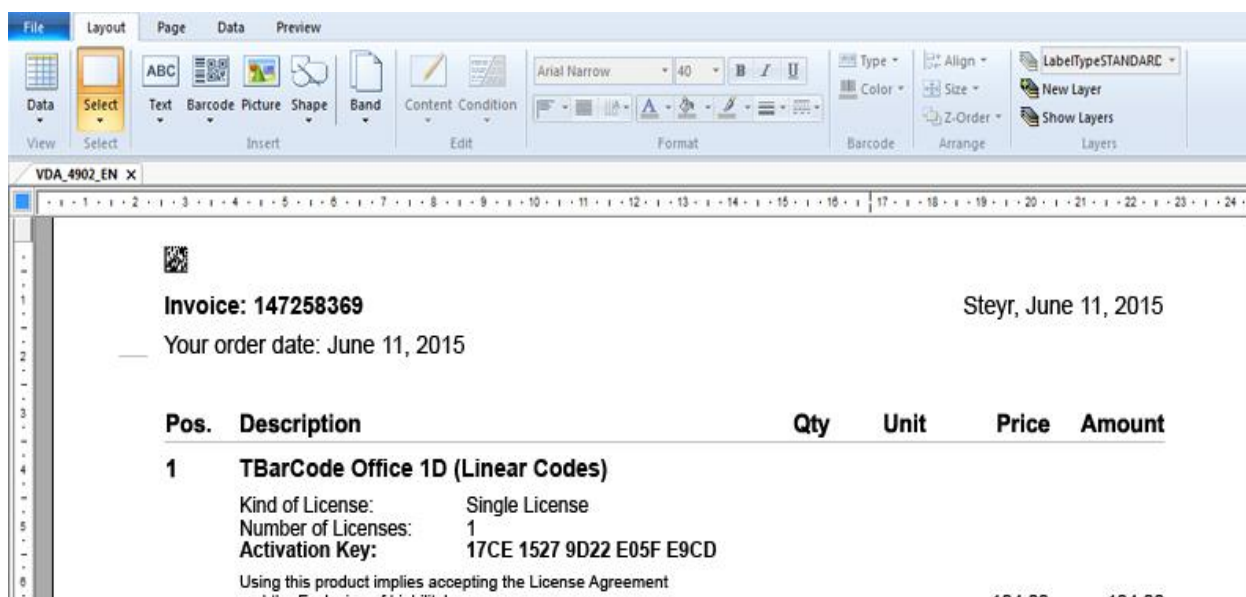


Рисунок 1.2. – Звіт «TFORMer SDK»

Шаблони документів, створені за допомогою TFORMer Designer, повністю включені в процес виведення даних. У графічному шаблоні вказується не тільки логіка лейаута, а й логіка виведення даних (наприклад, обчислення, умови для друку, лотки для паперу).

TFORMer SDK генератора звітів може бути використаний:

- як компонент для складання звітів в звичайних додатках;
- як генератор PDF файлів для клієнта і сервера (складання звітності на серверної і клієнтської сторонах);
- як вбудований компонент прямого друку;
- як онлайн рішення для генерації PDF в вебдодатках.

За допомогою даної програми складання звітів є можливість за мить створити звіти, листи, таблиці, серійні листи, форми і етикеток зі штрих-кодом. Результати генеруються безпосередньо або за допомогою TFORMer Designer, або за допомогою TFORMer SDK складання звітів. Ніякі додаткові витрати з програмування не потрібні. Застосунок є професійним інструментом, який відповідає промисловим стандартам, для створення форм, звітів, таблиць, етикеток і серійних листів [3].

Дизайнер може бути використаний як користувачами, так і розробниками програмного забезпечення. Він є ідеальним рішенням для виведення даних.

1.3 Сервіс управління торгівлею «Mr.Doc»

Mr.Doc – це хмарна система для зручного ведення управлінського обліку, що підходить як для інтернет-магазинів, так і для точок роздрібних продажів.

Онлайн система автоматизації і управління торгівлею. Управління продажами/закупівлями, облік доходів / витрат, складський облік, оформлення всіх необхідних документів. Інтеграція з клієнт-банком і 1С [4].

Приклад звіту, створеного в програмі «Mr.Doc» наведено на рис. 1.3.

Дата продажи	Организация	Тип товара	Товар	Кол-во	Сумма продаж	Себестоимость	Маржа
10.09.2014	Ромашка, ООО	Грабли	Грабли стальные	2,00	3 120,00	26,66	3 093,34
10.09.2014	Ромашка, ООО	Лопаты	Лопата совковая	14,00	22 120,00	16 800,00	5 320,00
10.09.2014	Ромашка, ООО	Ножницы	Ножницы садовые	1,00	845,00	576,00	269,00

Рисунок 1.3. – Звіт «Mr.Doc»

Сервіс управління торгівлею Mr.Doc реалізував механізм синхронізації даних з інтернет-магазинами на базі PrestaShop. Даний механізм дозволяє синхронізувати товари, замовлення, ціни і залишки в двосторонньому порядку. Крім цього в черговому пакеті оновлень доданий облік собівартості. Тепер для кожного, або оприбуткованого товару фіксується ціна і документ

надходження. При продажу товару ці параметри відображаються в окремих колонках табличній частині документа. Списання собівартості відбувається за принципом FIFO. Тобто, чим раніше товар надійшов, тим швидше спишеться його собівартість при продажу. Також в сервісі з'явилася можливість масового зміни довідників і документів, експорт даних в CSV, а також звіти «Продажі» і «Собівартість складських залишків» актуальності. Він же виконує і функції поштового клієнта, який теж не вимагає налаштування [4].

1.4 Програма «Power BI»

Microsoft Power BI – це набір аналітичних інструментів для аналізу бізнес-даних і обміну знаннями за допомогою візуалізацій. Попередньо створені панелі моніторингу об'єднують важливі показники в єдині уявлення, показуючи оновлення в реальному часі для кожного пристрою.

Централізована вебслужба Power BI взаємодіє зі шлюзами Power BI для підключення та уніфікації даних організації, в той час як Power BI Desktop для Windows надає функції локального об'єднання даних і створення звітів. Живі інформаційні панелі надають миттєвий огляд стану бізнесу, використовуючи 59 готових підключень для додатків або сервісів, таких як Salesforce, Google Analytics, Azure, Facebook, Github, QuickBooks Online, Twilio і Zendesk.

Завдяки Power BI Desktop користувачі можуть створювати звіти за допомогою інтуїтивно зрозумілих інструментів перетягування і обмінюватися ними з колегами через онлайн-сервіс Power BI [5].

Приклад звіту, створеного в програмі «Power BI» наведено на рис. 1.4.

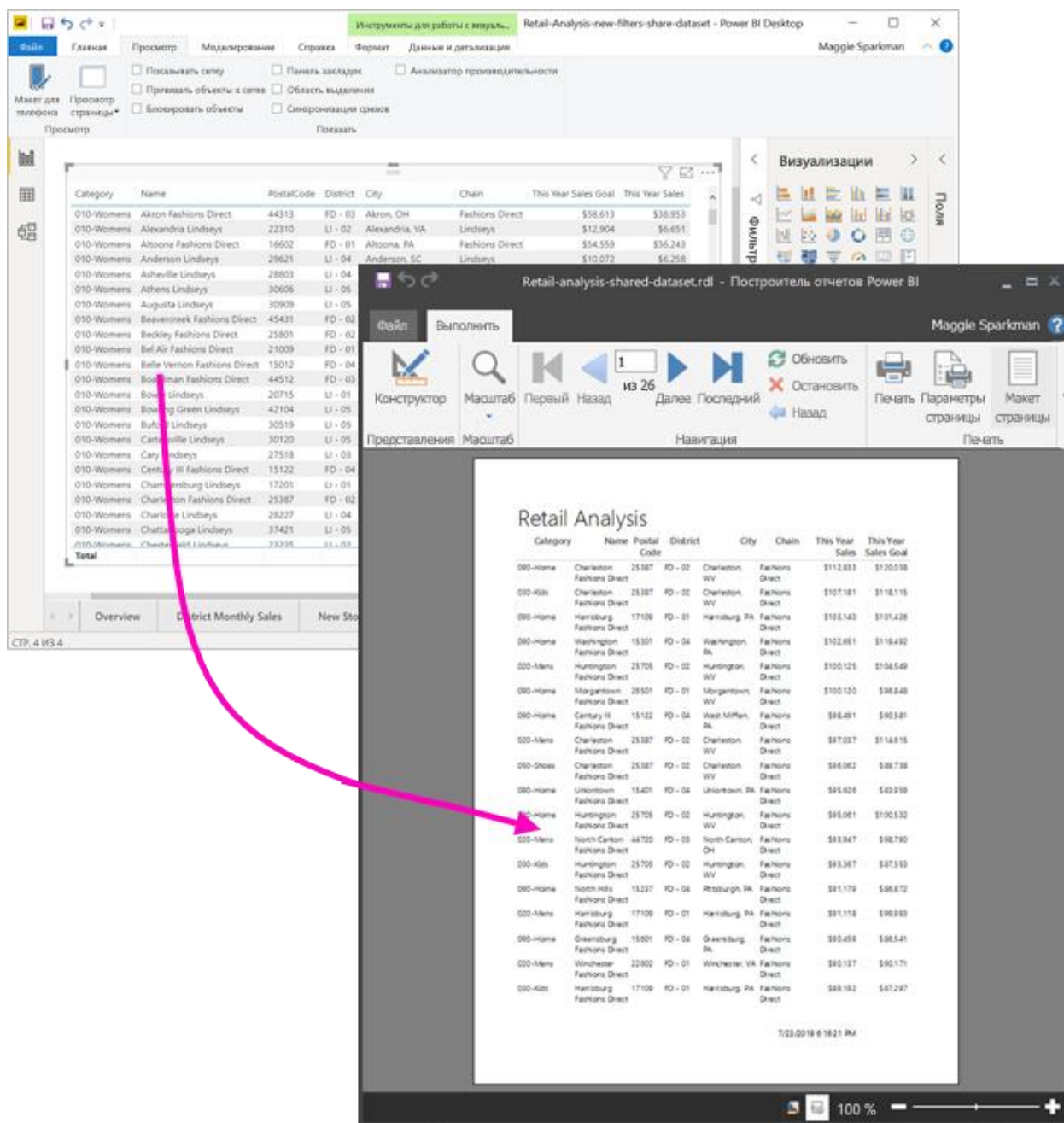


Рисунок 1.4. – Звіт «Power BI»

Інтегруючись з встановленим Microsoft механізмом запитів, технологіями моделювання та візуалізації даних, Power BI Desktop об'єднує і формує звіти, щоб централізувати, як правило, окремий процес створення звітів. Виходячи в формат .pbix, ці файли звітів можуть бути передані вручну або завантажені в загальну службу. Крім того, ця можливість постійно підтримувати всіх користувачів на одній і тій же сторінці поширюється на набір оптимізованої мобільного підтримки. Power BI Mobile підтримує

узгоджене віддалене підключення до власних додатків для пристроїв Windows Mobile, iOS і Android, додаючи інтуїтивно зрозумілі операції на основі жестів. Живі інформаційні панелі можна переглядати на льоту, поряд з можливостями повного обміну звітами і оповіщенням на основі даних [6].

Висновки

В якості висновку можна сказати, що користувачам потрібне рішення з базовою функціональністю, простим інтерфейсом та забезпеченням інструкцій для роботи на ПК. В результаті дослідження існуючих рішень було виявлено, що на ринку наявні широкофункціональні рішення. Вимогою замовника була розробка спеціалізованого рішення, що направлене на вирішення саме його задач, пов'язаних з формуванням та аналізом аналітичних звітів. Проте, при розробці цього рішення був проведений аналіз універсальних програм, призначених в тому числі і для вирішення задач аналізу документації.

РОЗДІЛ 2. ОГЛЯД ВИКОРИСТАНИХ ТЕХНОЛОГІЙ

2.1 Обґрунтування вибору мови і системи програмування

Для розробки застосунку обрано мову програмування C#, платформу .NET Framework, технології WPF, використовуючи програмне забезпечення DevExpress в інтегрованому середовищі розробки Microsoft Visual Studio 2019. Для створення запитів до бази даних використовувалась мова SQL у вільній системі керування реляційними базами даних MS SQL Server. Створення шаблонів відбувалося в генераторі звітів FastReport .NET.

Були використані компоненти, бібліотеки та класи, створені програмістами компанії Fozzy group.

2.1.1 Платформа .NET Framework

Microsoft .NET Framework – це одна з найпоширеніших платформ для написання програм і інсталяції їх на ПК. Вона здатна працювати з різними мовами програмування і функціонує практично на всіх операційних системах.

Ця платформа створена для утиліт, в основі яких лежить архітектура .NET, вперше вона вийшла в світ у 2002 році і вважалася аналогом поширеної в той час технології Java. Ця розробка корпорації Microsoft здатна працювати з різними мовами програмування, тому її дуже часто використовують при написанні додатків і різних утиліт [7].

Платформа Microsoft .NET Framework здатна працювати з різними типами програм, дозволяє використовувати практично всі мови програмування, її можна встановлювати на різні операційні системи.

Варто відзначити простий і зрозумілий інтерфейс платформи. Перевагою .NET Framework можна назвати безкоштовне поширення, що дозволяє бажаючим вільно оновити встановлену на комп'ютер платформу [8].

Спеціально для платформи .NET Microsoft було розроблено нову мову програмування C# [9, 10].

2.1.2 Мова програмування C#

C# (C-Sharp) – об'єктно-орієнтована мова програмування, розроблена Microsoft, яка працює на .NET Framework.

У назві «Сі шарп» (англ. Sharp – діз) є свій «сакральний» сенс. Знак «#» означає підвищувати звук на півтону. З іншої сторони, назва «C#» отримується шляхом «еволюційного ланцюжка»: $C \rightarrow C++ \rightarrow C++++ (C \#)$, так як символ «#» можна скласти з 4-х знаків «+».

Будучи об'єктно-орієнтованою мовою, вона багато перейняла у Java і C++. Як і Java, C# спочатку призначалася для веброзробки, і приблизно 75% його синтаксичних можливостей такі ж, як у Java. C# також називають «очищеною версією Java». Ще 10% з C++ і 5% – з Visual Basic. Решта 10% C# – це реалізація власних ідей розробників. Об'єктно-орієнтований підхід дозволяє будувати за допомогою C# великі, але в той же час гнучкі, масштабовані додатки [10].

C# вже давно підтримує багато корисних функцій:

- інкапсуляція;
- наслідування;
- поліморфізм;
- перевантаження операторів;
- статична типізація.

При цьому вона все ще активно розвивається, з'являється все більше цікавого, наприклад: лямбда вирази, динамічне зв'язування, асинхронні методи [10].

Уперше мова C# вийшла разом з Microsoft Visual Studio .NET в лютому 2002 року.

Переваги і недоліки мови C#:

1. Підтримка продуктів Microsoft.
2. Безкоштовність ряду інструментів для індивідуальних розробників – Visual Studio, хмара Azure, Windows Server, Parallels Desktop для Mac Pro і ін.

3. Типи даних мають визначений розмір, що підвищує «мобільність» мови і спрощує програмування.
4. Автоматичний прибиральник сміття. Тобто, в більшості випадків не потрібно піклуватися про звільнення пам'яті.
5. Велика кількість спеціальних конструкцій, розроблених для розуміння і спрощеного написання коду.
6. Низький поріг входження. Синтаксис C# має багато спільного з іншими мовами програмування, завдяки чому полегшується перехід для програмістів.

2.1.3 Інтегроване середовище розробки Visual Studio і Windows Forms

Інтегроване середовище розробки Microsoft Visual Studio – це комплексне програмне рішення для написання, налагодження і збірки коду, а також публікування застосунків. Інтегроване середовище розробки (IDE) – багатофункціональна програма, яку використовують для розробки ПЗ. Окрім редактора і відладчика, які включають в себе більшість середовищ IDE, Visual Studio містить компілятори, засоби автозавершення коду, графічні конструктори та інші функції.

Windows Forms (WF) – це інтелектуальна клієнтська технологія для .NET Framework, набір керованих бібліотек, що спрощують типові завдання програми.

У WF форма – це візуальна поверхня, на яку виводиться інформація для користувачів. Такий додаток будується шляхом переміщення елементів управління на форму і написання коду для реагування на дії користувачів, такі як натискання на клавіші миші та клавіатури [7].

При виконанні користувачами певних дій з формою або з її елементами управління створюється подія. Додаток реагує на ці події за допомогою коду і обробляє їх.

Windows Form має набір елементів управління, які можна додати на форму: текстові поля, кнопки, списки, перемикачі, вебсторінки. Якщо можливостей існуючих елементів управління не достатньо, то є можливість створити власні елементи за допомогою класу UserControl [11].

2.1.4 Система управління Microsoft SQL Server та мова запитів SQL

SQL Server також відомий, як Microsoft SQL Server, був розроблений компанією Microsoft в 80-х.

Плюси SQL Server:

1. СКБД масштабується, тому працювати з нею можна на портативних ПК або потужній мультипроцесорній техніці. Процесор може одночасно обробляти запити великого обсягу.
2. Розмір сторінок – до 8 кб, тому дані отримуються швидко, детальну і складну інформацію зберігати зручніше. Система дозволяє обробляти транзакції в інтерактивному режимі, є динамічне блокування.
3. Рутинні адміністративні завдання автоматизовані: це управління блокуванням, пам'яттю, редагування розмірів файлів. У системи продумані налаштування, можна створити профілі користувачів.
4. Реалізовано пошук за фразами, текстом, словами також є можливість створювати ключові індекси.
5. У SQL Server є реплікації через інтернет, передбачено синхронізацію.
6. У систему інтегровано сервер інтерактивного аналізу для прийняття рішень та створення корпоративних звітів, який містить служби перетворення інформації.
7. СКБД підтримує роботу з іншими продуктами Microsoft [12].

Structured Query Language (SQL) є стандартним засобом маніпулювання і формування запитів до даних в реляційних БД, хоча з певними запатентованими розширеннями серед комерційних продуктів. Стандартизація є одним з головних переваг мови. Другий плюс – це простота

і поширеність SQL, що навіть змусило творців багатьох «NoSQL» або нереляційних сховищ даних, таких як Hadoop, прийняти підмножини SQL або створити власні SQL-подібні мови запитів.

Переваги:

1. Незалежність від конкретних реляційних СКБД.
2. Повноцінність як мови, призначеної для роботи з базами даних.
3. Можливість динамічної зміни даних і структури таблиць.

За допомогою SQL можна динамічно змінювати і розширювати структуру БД навіть у той час, коли користувачі звертаються до її вмісту.

4. Підтримка архітектури клієнт/сервер.

SQL служить сполучною ланкою між клієнтською системою, яка взаємодіє з користувачем, і серверної системою, яка керує БД [12].

2.1.5 Генератор звітів FastReport .NET

Бібліотека генерації звітів та створення документів для .NET 5, .NET Core, ASP.NET, MVC і WinForms. Можливе використання в середовищі Microsoft Visual Studio.

Генератор звітів – це програма, що дозволяє на основі набору даних створювати готові до демонстрації документи.

FastReport .NET не використовує інтегроване середовище розробки Microsoft Visual Studio для створення та зміни звітів, як інші генератори звітів. Замість цього, у ньому міститься візуальний дизайнер, що підтримує масштабованість, буфер undo/redo, виносні лінії, лінійки. Він повністю налаштовується і імітує середовище Visual Studio. Дизайнер використовується в run-time, отже є змога власноруч змінювати існуючі звіти і створювати нові [13].

Використовуючи FastReport .NET, є можливість створювати звіти, що працюють незалежно, має:

- можливість підключитися до будь-якої БД, використовувати її таблиці або створювати власні запити на мові SQL;

- можливість додавати діалогові форми в звіт для запиту параметрів перед запуском;
- можливість використовувати вбудований скрипт, управляти взаємодією між елементами управління діалогових форм і здійснювати складну обробку даних;
- можливість переглядати результат у вікні попереднього перегляду, вивести звіт на друк або зберегти його.

Переваги:

1. FastReport .NET написаний на C # і містить в собі керований код. Він сумісний з .NET Framework 4.x, .NET Core, .NET 5.
2. FastReport.NET містить вихідний код (редакції Professional і Enterprise). Його можна адаптувати під власні потреби.
3. FastReport .NET дозволяє підключати до звіту об'єкти, фільтри експорту, функції та БД [13].

2.1.6 Програмне забезпечення DevExpress

Компанія DevExpress займається розробкою компонентів і доповнень для WinForms, ASP.NET, WPF, Silverlight, Windows 8 XAML, HTML JS, VCL. Десятки тисяч програмістів з усього світу обирають продукти DevExpress для створення своїх додатків. Мета компанії – зберегти користувачам їх час, тому DevExpress створює продукти, які володіють великою функціональністю, зручністю і простотою використання. Плюс до цього, компанія надає високоякісну професійну технічну підтримку клієнтів.

Всі програмні рішення компанії Developer Express діляться на 2 групи: програмні продукти і бібліотеки.

Програмні продукти дозволяють створити додатковий інтерфейс або прискорити компіляцію.

DevExpress WinForms Subscription – це комплект з понад 110 засобів управління і бібліотек WinForms для середовища Visual Studio. Оптимізовані вебкомпоненти дозволяють розробникам створювати гнучкі програми, що

відповідають конкретним завданням бізнесу і надають кінцевим користувачам всі необхідні інструменти в найкоротші терміни. Також пропонує клікабельні керуючі елементи, більш 40 тем для додатків, а також відрізняється підвищеною продуктивністю. Пакет включає бібліотеки, що відображають застосування C# і оптимізуються під .NET [14].

Висновки

У цьому розділі описані технології, які були використані в ході розробки застосунку для формування та аналізу звітної документації підприємства торгово-промислової групи.

РОЗДІЛ 3. ПРИЗНАЧЕННЯ, ВИМОГИ, ЦІЛІ СТВОРЕННЯ ЗАСТОСУНКУ ТА ЙОГО РЕАЛІЗАЦІЯ

3.1 Призначення та цілі програми

Програма призначена для підвищення надійності зберігання та обробки інформації.

Основним призначенням є зберігання, відображення даних та формування їх у звіт.

В рамках проєкту автоматизується процес відображення та зміни даних зручними та ефективними способами:

- зберігання і обробка інформації про документ;
- пошук потрібної інформації та її відображення;
- забезпечення інформаційних потреб користувача.

Програма створюється з такими цілями:

- забезпечення збору і первинної обробки вихідної інформації;
- пошук необхідної для користувача інформації;
- створення системи звітності;
- підвищення якості (повноти, точності, достовірності, своєчасності, узгодженості) інформації;
- забезпечення безпеки даних;
- полегшення роботи з інформацією.

В результаті створення сховища даних повинні бути поліпшені значення наступних показників:

- час збору і первинної обробки вихідної інформації;
- час пошуку інформації;
- час, що витрачається на інформаційно-аналітичну діяльність.

Проект передбачає для досягнення поставленої мети вирішувати наступні завдання:

- аналіз існуючих методик, моделей і алгоритмів оцінки надійності програмних засобів;
- аналіз методик, моделей і алгоритмів оцінки надійності ІС;
- моделювання предметної області;
- проектування та програмну реалізацію розроблюваного застосунку;
- формування підмоделей оцінки надійності, шляхом модифікації базової моделі;
- тестування та супровід системи.

3.2 Вимоги до програми в цілому

На основі дослідження існуючих рішень можна сформулювати наступні вимоги:

1. Простий інтерфейс.
2. Вибірка даних за певними критеріями:
 - 2.1. Можливість введення періоду у вигляді дати.
 - 2.2. Можливість введення табельного номеру.
3. Робота з отриманими даними:
 - 3.1. Сортування.
 - 3.2. Фільтрація даних за текстом, датою та значеннями.
 - 3.3. Експортування.
 - 3.4. Деталізація вибраного документу.
4. Робота з вибраним документом:
 - 4.1. Редагування.
 - 4.2. Зберігання змін.
 - 4.3. Експортування.
 - 4.4. Сортування даних документу.
 - 4.5. Фільтрація даних документу.

5. Формування звітів у вибраному документі:

5.1. Інвентаризаційної відомості.

5.2. Порівняльної відомості.

5.3. Штрих-кодів вибраних товарів.

Програма повинна:

- працювати максимально швидко;
- бути стійкою до різного роду помилок і нештатних ситуацій;
- мати зрозуміло сконструйований та логічно зв'язаний користувацький інтерфейс;
- мати масштабований додаток, що дозволяє швидко вносити зміни у вихідний код.

У програмі передбачається функціональна підсистема користувача.

Сценарій взаємодії користувача з застосунком зображений на use-case діаграмі наведених в додатку А.

Сформуємо основні функції користувача (див. табл. 3.1).

Таблиця 3.1. – Перелік функцій, задач що підлягають автоматизації

Функція	Задача
Робота з реєстром	Вибірка даних за певними критеріями: датою та табельним номером
	Експорт отриманих даних
	Сортування
	Фільтрація за текстом, датою та значеннями

Продовження таблиці 3.1

Функція	Задача
Робота з вибраним документом	Експорт отриманих даних
	Редагування
	Зберігання внесених змін
	Друк відомостей та штрих-кодів
	Фільтрація за текстом, датою та значеннями
	Сортування

3.3 Технічні вимоги

Так як програма повинна працювати на широкому загалі пристроїв, вимоги до неї повинні бути мінімальними.

Вимоги до апаратного забезпечення:

- 2х ядерний процесор з частотою 1 ГГц;
- 512 Мб оперативної пам'яті;
- 50 Мб пам'яті на диску.

Вимоги до програмного забезпечення:

- операційна система Windows 7/8/8.1/10;
- встановлена версія .NET Framework 4.5.2.

На довільні некоректні дії користувача, пов'язані з введенням невірних даних, незаповненням обов'язкових полів введення в формах та інші, які можуть бути оброблені системою, генеруються відповідні повідомлення про помилки англійською мовою, в межах загального дизайну застосунку.

3.4 Реалізація застосунку

3.4.1 Опис шаблонів

Для розробленого застосунку потрібні шаблони інвентаризаційної та порівняльної відомостей і штрих-коду.

Все майно і всі види фінансових зобов'язань організації підлягають інвентаризації. Для оформлення підсумків ревізії застосовуються форми первинних документів з обліку результатів інвентаризації. Одним з таких первинних облікових документів є інвентаризаційна відомість. Відомість інвентаризації складається за підсумками проведення ревізії майна компанії і призначена для відображення фактичної і облікової кількості майна по кожному найменуванню із зазначенням його вартісного вираження на певну дату [15].

Було створено шаблони в FastReport для відомостей та штрих-коду наведені на рис. 3.1 – 3.5:

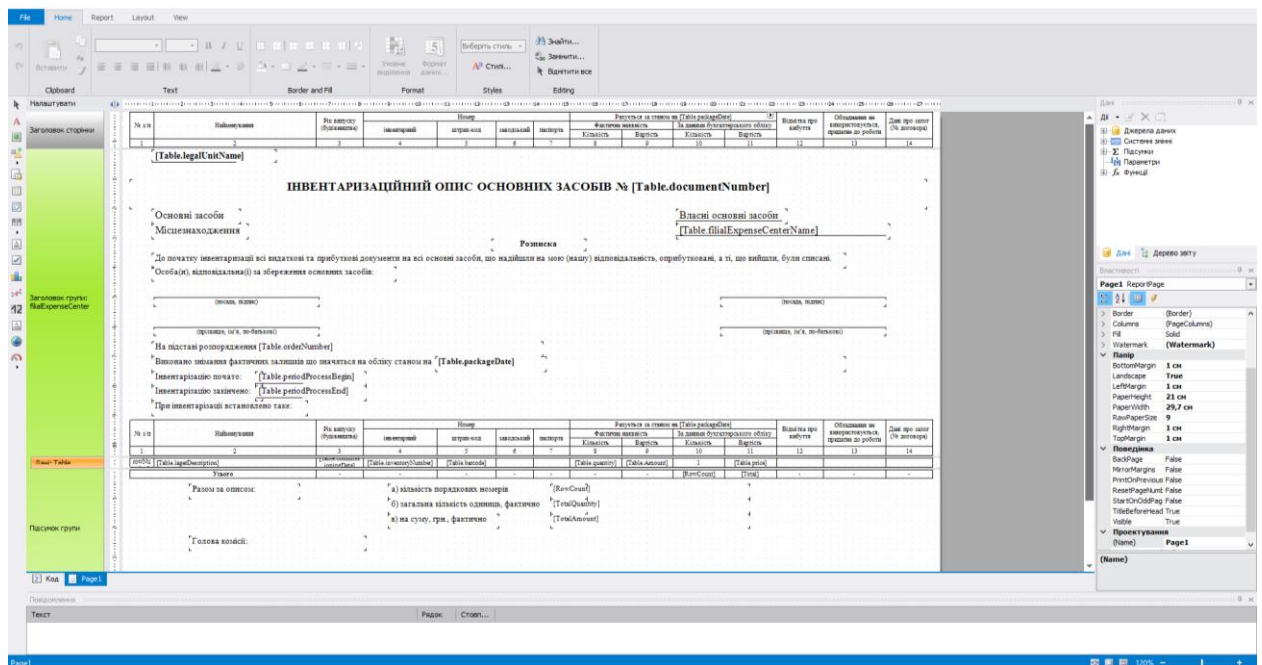


Рисунок 3.1. – Шаблон інвентаризаційної відомості

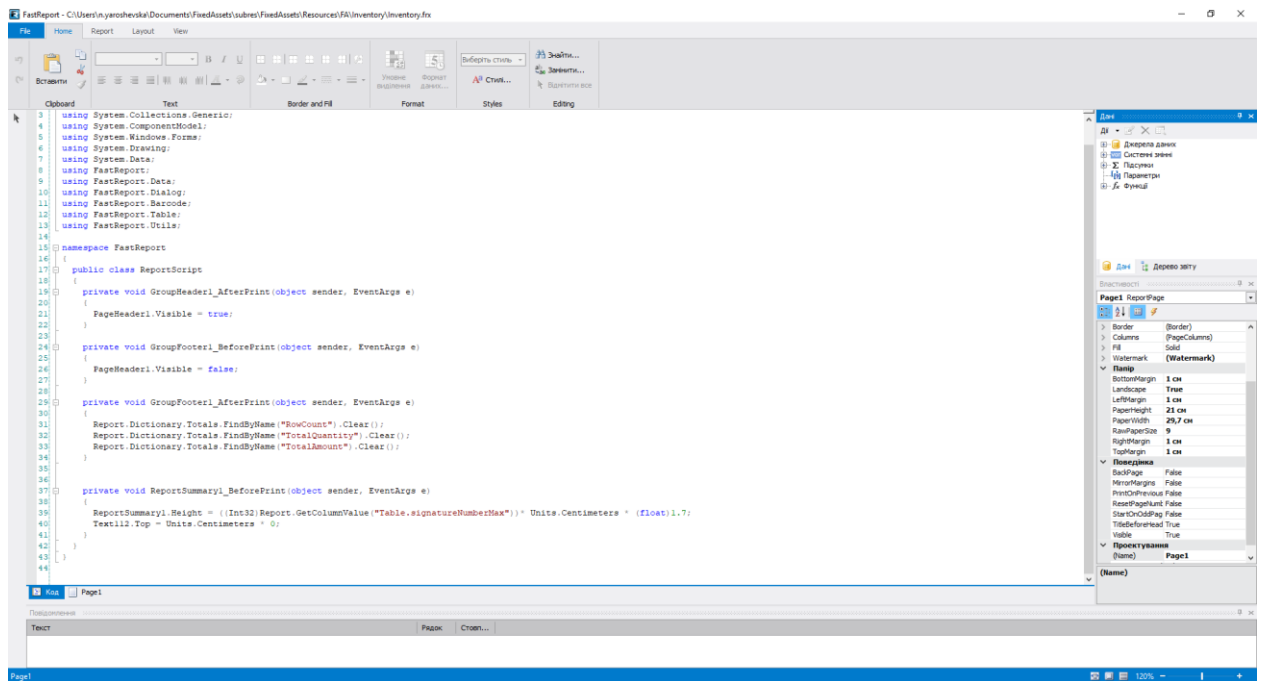


Рисунок 3.2. – Код шаблону інвентаризаційної відомості

Що стосується розбіжності фактичної кількості – перевіряється майно з даними бухгалтерського обліку, після цього оформляється порівняльна відомість, в якій вказуються лишки або недоліки даного майна [15].

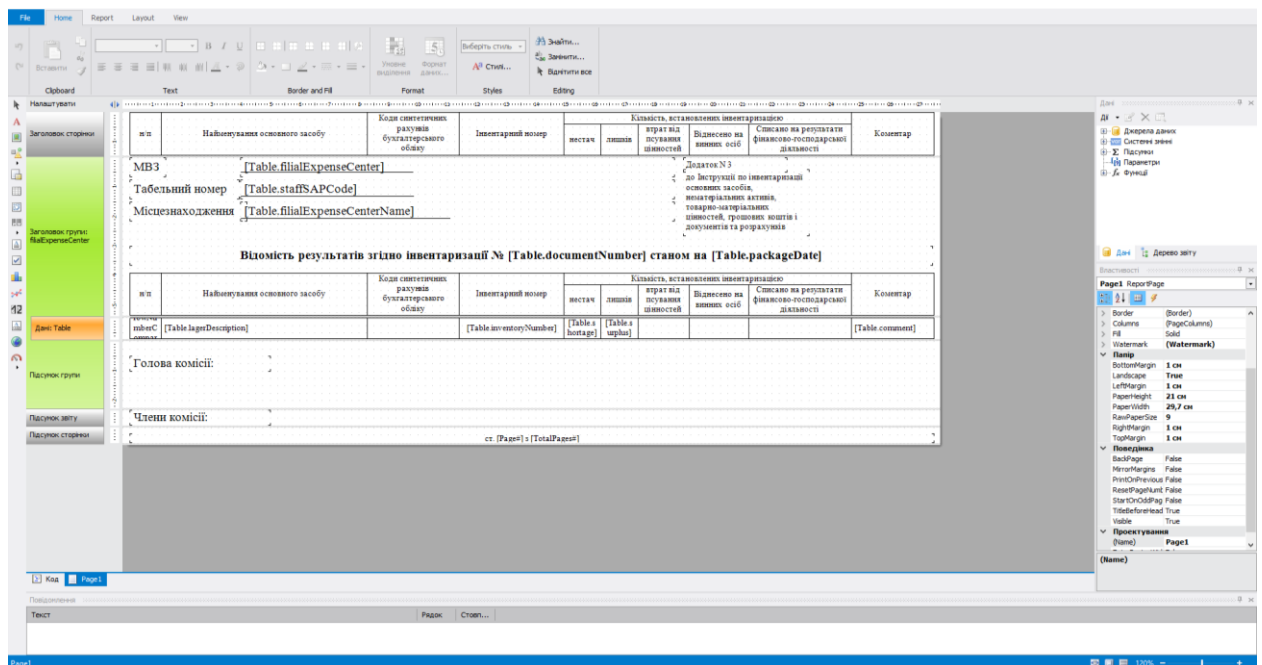


Рисунок 3.3. – Шаблон порівняльної відомості

Штрих-код наноситься на упаковку у вигляді штрихів і являє собою закодовану інформацію, сканується спеціальними пристроями. За допомогою штрих-коду кодують інформацію про найбільш істотні параметри продукції.

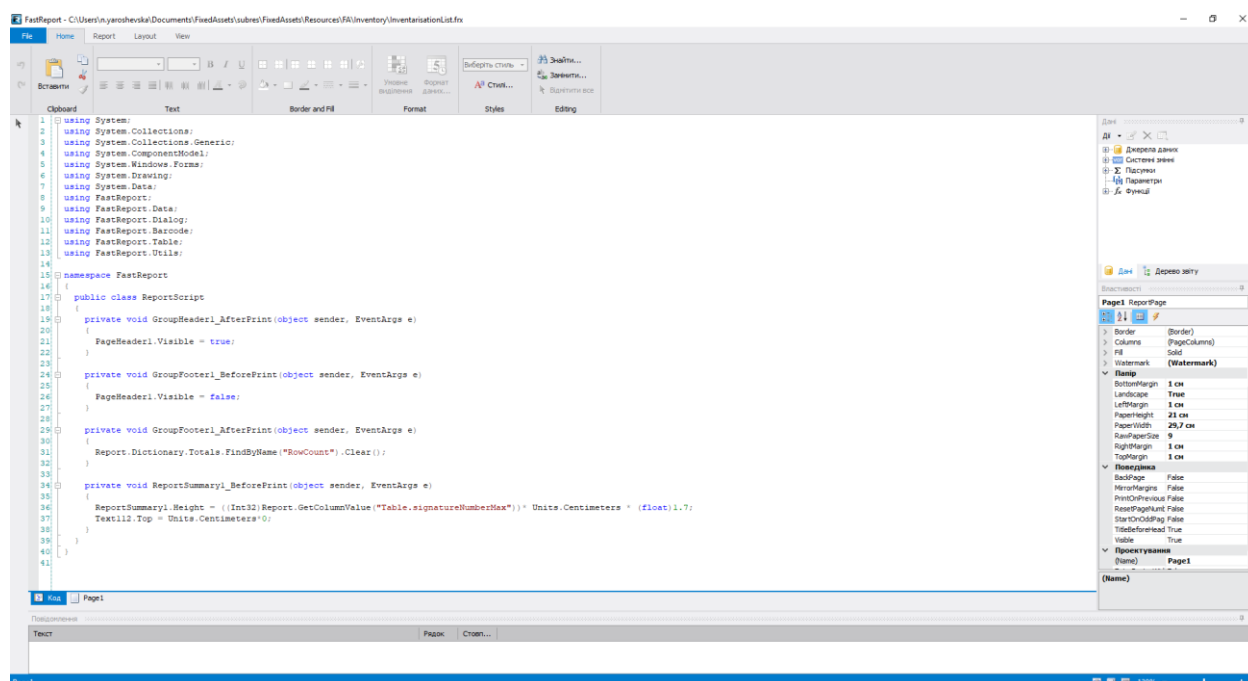


Рисунок 3.4. – Код шаблону порівняльної відомості

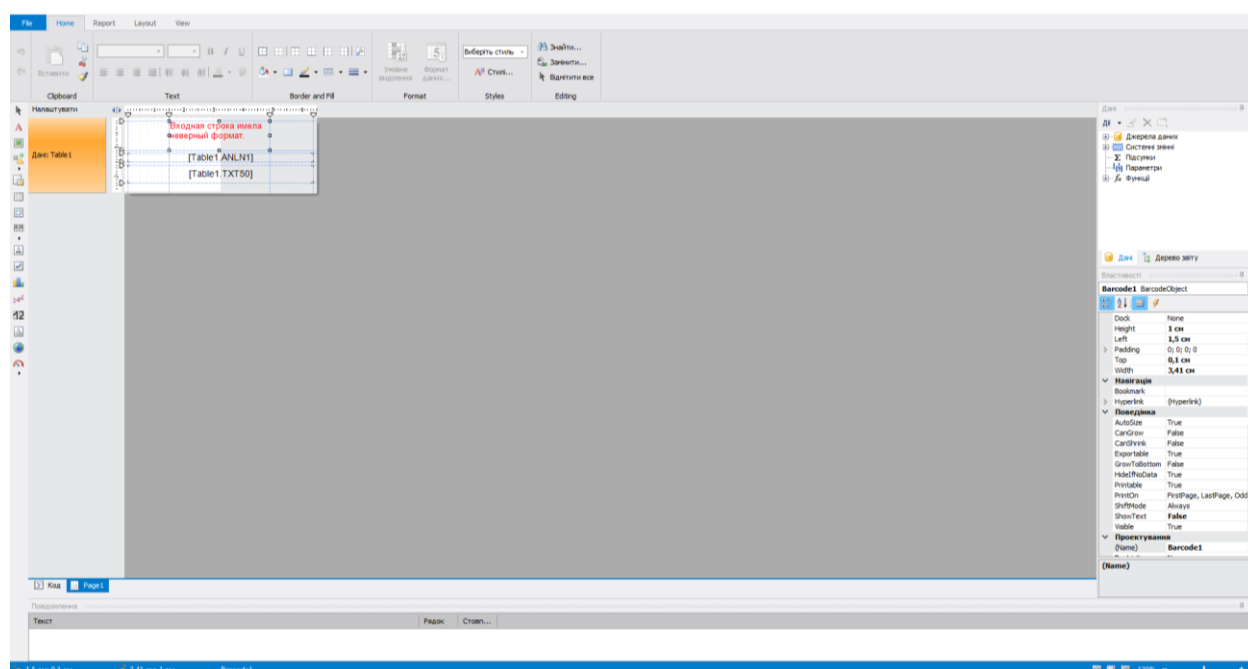


Рисунок 3.5. – Шаблон штрих-коду

3.4.2 Прокєтування застосунку

У середовищі розробки було створено проєкт, в якому: «InventoryList» – форма-реєстр, в якій міститься панель (dockPanel) з фільтром та таблиця для відображення даних (gridControl); «InventoryListEntry» – форма, в якій міститься таблиця для відображення даних про документ. Обидві форми являють собою XtraForm.

Форма «InventoryList» містить:

1. Панель (dockPanel) з фільтром, на якій розміщено:
 - компонент для введення періоду на календарі (див. рис. 3.6);
 - поле (popupContainerEdit) для введення табельного номеру, являє собою випадаючий список (див. рис. 3.7);
 - дві кнопки (simpleButton) для застосування та очищення фільтру (див. рис. 3.8).
2. Панель (bar), на якій розміщені:
 - дві кнопки (simpleButton) для оновлення та експорту таблиці (див. рис. 3.9).
3. Таблиця (gridControl) для відображення даних отриманих за умовами фільтру.

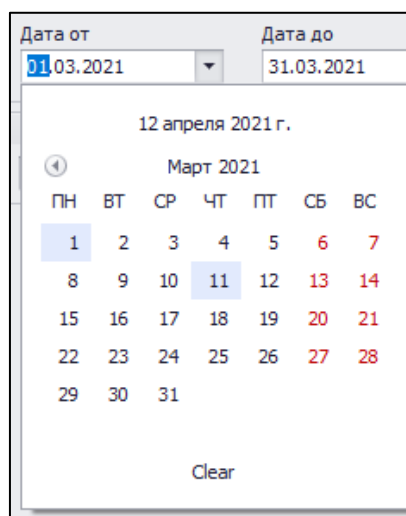


Рисунок 3.6. – Компонент для введення дати

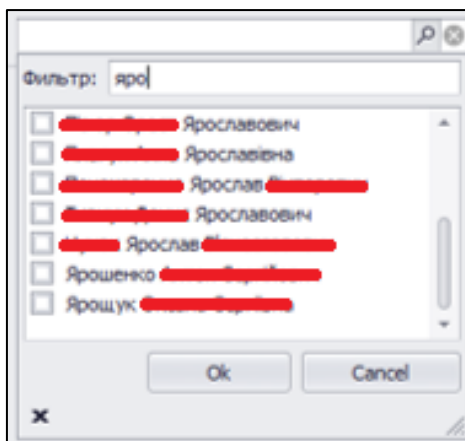


Рисунок 3.7. – Поле для введення табельного номеру



Рисунок 3.8. – Кнопки фільтру



Рисунок 3.9. – Кнопки верхньої панелі

Форма «InventoryListEntry» містить:

1. Панель (bar), на якій розміщені:
 - три кнопки (simpleButton) для оновлення та експорту таблиці і збереження внесених змін (див. рис. 3.10);
 - випадаюче меню (barSubItem), яке містить три кнопки (simpleButton) для друку відомостей та штрих-кодів (див. рис. 3.11);
 - перемикач (barEditItem) для відображення розходжень (див. рис. 3.12).
2. Таблиця (gridControl) для відображення даних вибраного документу.



Рисунок 3.10. – Кнопки верхньої панелі

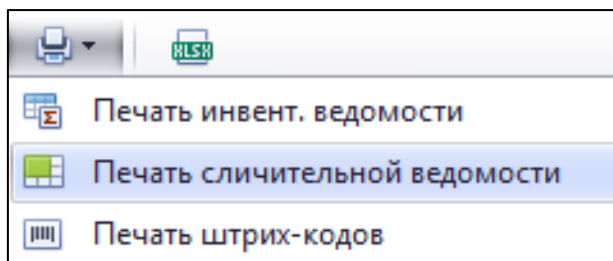


Рисунок 3.11. – Кнопки для друку



Рисунок 3.12. – Перемикач

3.4.3 Конструювання застосунку

Розглянемо реалізацію кнопки «Експорт» на формі «InventoryListEntry»:

- при натисканні на дану кнопку викликається метод

ExportToFile(mainGridView, Text, _exportSheetName),

де mainGridView – це компонент, що містить дані таблиці; Text – властивість, яка містить назву форми; _exportSheetName – поле, яке містить назву для листів в Excel.

Реалізацію методу для експортування даних в Excel-документ наведено в додатку Б.

Метод формує назву файлу, SaveFileDialog – діалогове вікно, для визначення шляху зберігання файлу, Filter – задає фільтр, завдяки чому в діалоговому вікні можна відфільтрувати файли. Сам метод записує дані з gridViewControl в файл з розширенням .xlsx, форматуючи деякі стовпці в числовий формат.

Розглянемо реалізацію кнопки «Друк інвент. відомості» на формі «InventoryListEntry»:

- при натисканні на дану кнопку викликається метод

PrintFastReport(DocumentType.Inventory, fastReportTemplateInventory),


```

{
    DataSet dsFromSQL =
Helpers.ParseExecuteReaderContract(oFromSQL);
    if (dsFromSQL == null)
        return;
    mainGridControl.DataSource = dsFromSQL.Tables[0];
    },
    loadOperation,
    Serialization.Serialize(ds),
    null, 3600);
}
}

```

У методі формується таблиця, в яку записуються дані з фільтру: дата та табельний номер. Ця таблиця передається в процедуру loadOperation, що містить запит SQL, який повертає певний набір даних. Цей набір серіалізується в послідовність бітів, після цього ним заповнюється GridControl.

Висновки

У цьому розділі описані призначення, вимоги та цілі створення застосунку для формування та аналізу звітної документації підприємства торгово-промислової групи. Також були наведені шаблони звітів, етапи проєктування та кодування.

РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА

4.1 Вкладка «Інвентаризаційна відомість»

Користувач відкриваючи вкладку «Інвентаризаційна відомість» бачить на екрані вікно наведене на рис. 4.1.

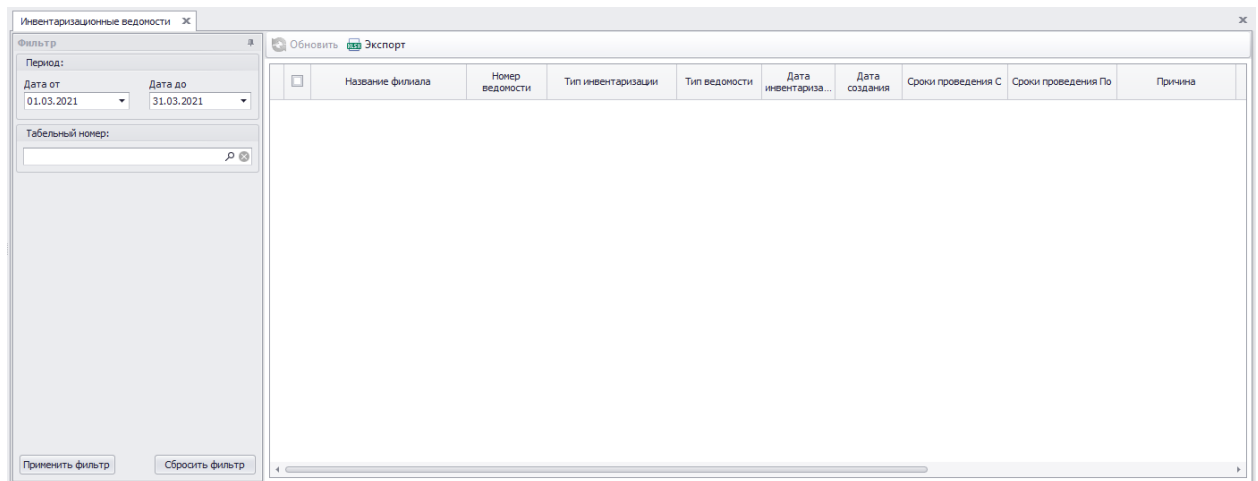


Рисунок 4.1. – Вкладка «Інвентаризаційна відомість»

На даній вкладці потрібно задати період в календарі (дату «з» та «по») за який потрібно відобразити документи (див. рис. 4.2).

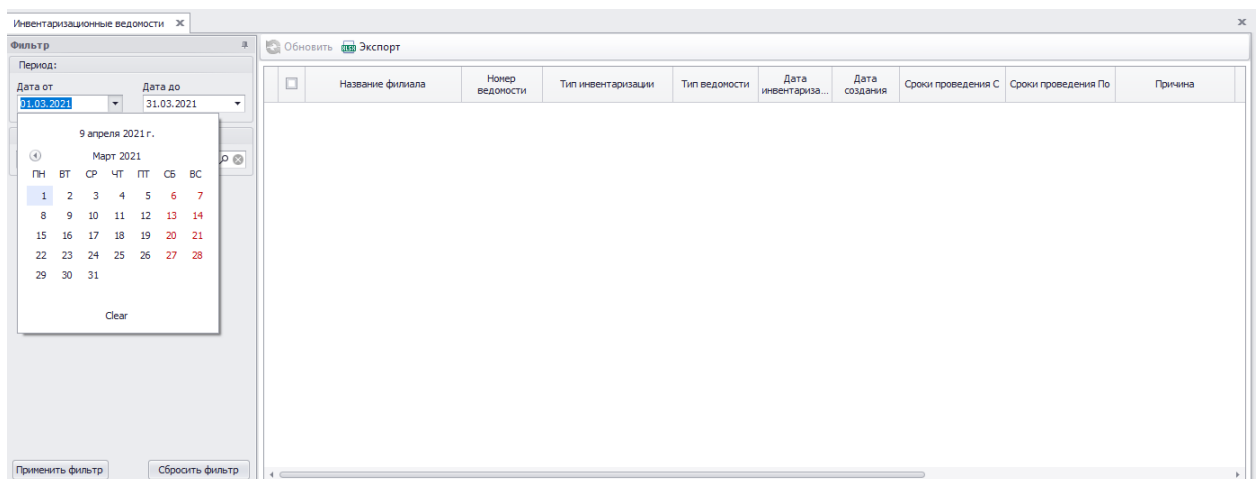


Рисунок 4.2. – Вибір дати

Далі користувач має можливість ввести табельний номер, ввівши певний рядок, після цього вибрати потрібні варіанти з випадаючого списку нижче (див. рис. 4.3)

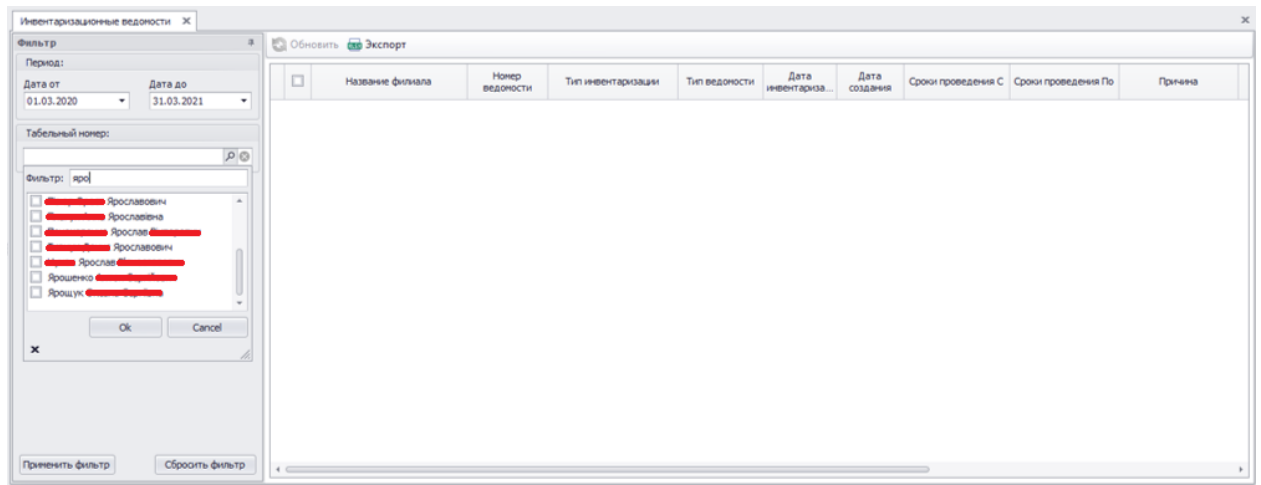


Рисунок 4.3. – Вибір табельного номеру

Після вибору періоду та табельного номеру, користувач повинен натиснути на кнопку «Застосувати фільтр», далі на екрані з'явиться набір даних, що задовольняють введеним умовам вибору (див. рис. 4.4).

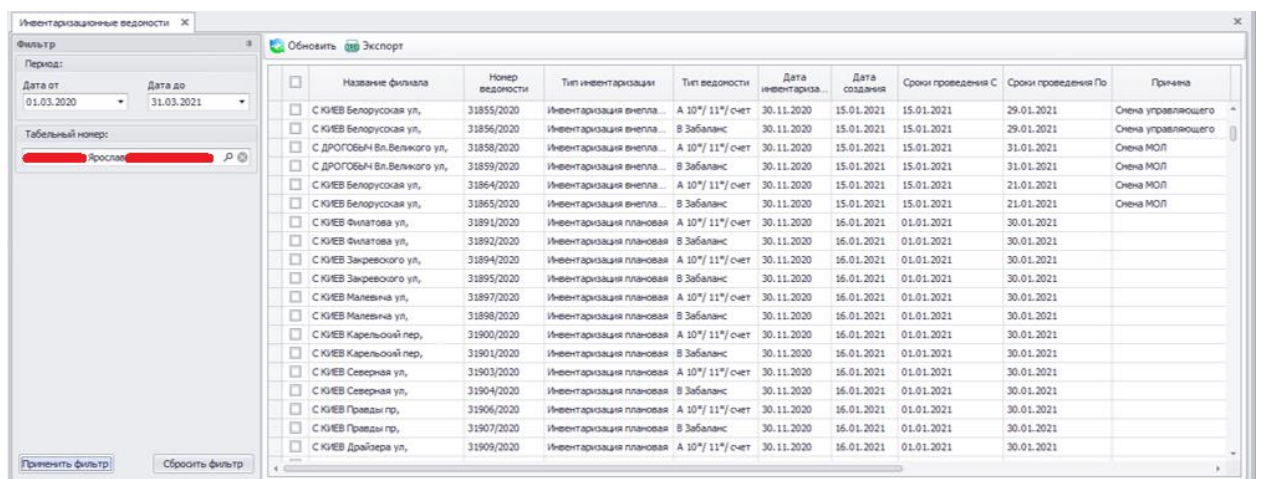


Рисунок 4.4. – Застосування умов фільтру

Далі розглянемо сортування та фільтрування отриманих даних:

1. На стовпчику «Назва філіалу» натискаємо на значок фільтру, з'являється вікно, в якому ми можемо вибрати вид вибірки по стовпцю з філіалами. На вкладці «values» можна вибрати філіали, які цікавлять користувача (див. рис. 4.5).

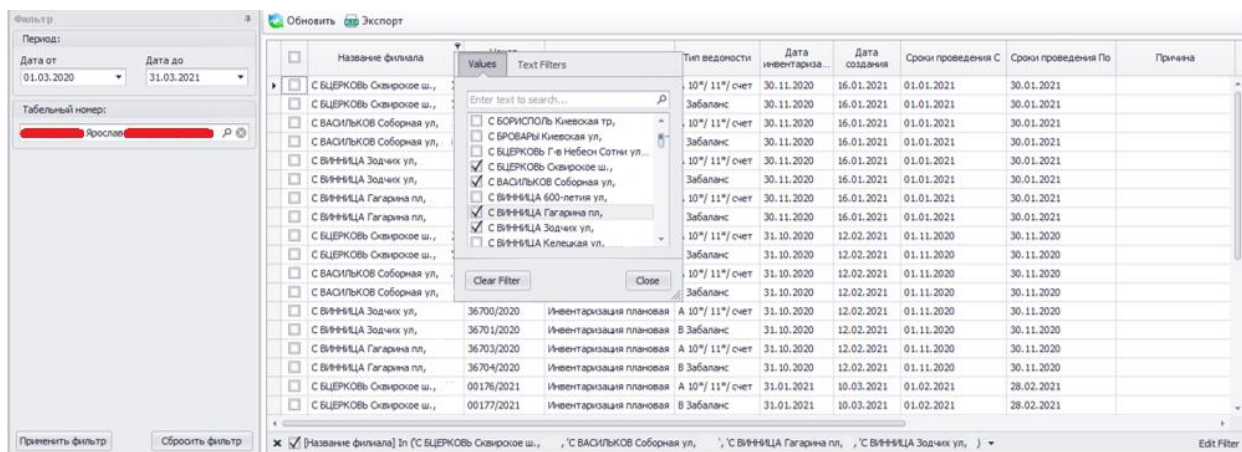


Рисунок 4.5. – Вибір філіалів

На вкладці «textFilters» можна вибрати вид текстового фільтру, який цікавить користувача (див. рис. 4.6)

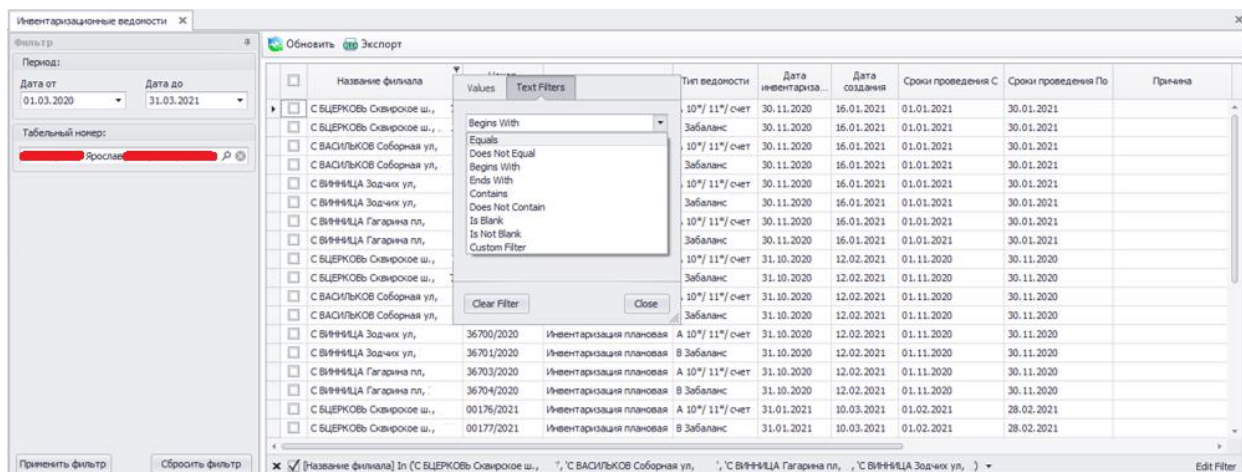


Рисунок 4.6. – Вибір текстового фільтру

Наприклад:

1) виберемо текстовий фільтр «does not contain» та введемо значення «64»: отримаємо набір даних, в якому назва філіалу не матиме тексту «64» (див. рис. 4.7);

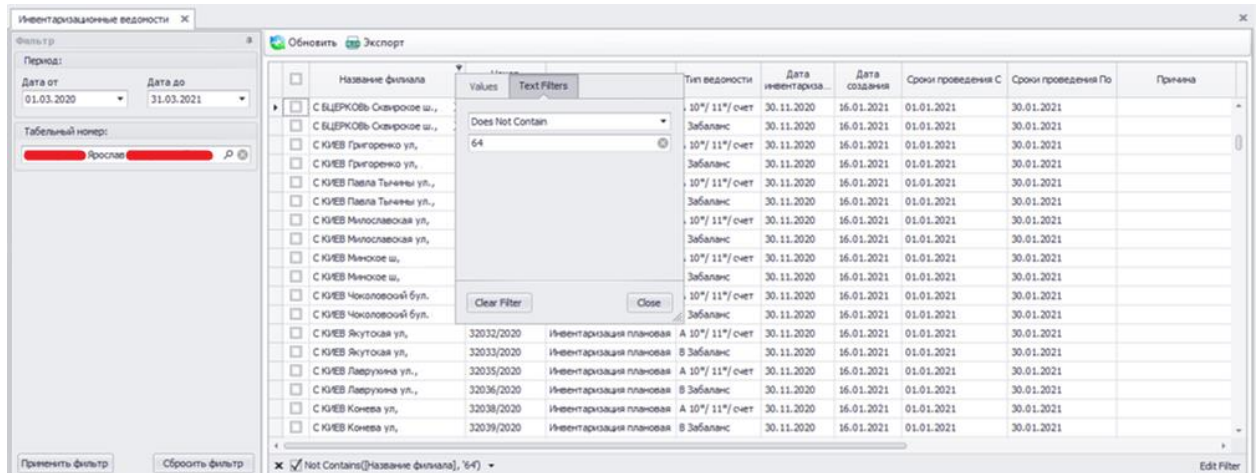


Рисунок 4.7. – Застосування текстового фільтру

2) виберемо текстовий фільтр «equals» та введемо певний текст: отримаємо набір даних, в якому назва філіалу співпадатиме з введеним значенням (див. рис. 4.8);

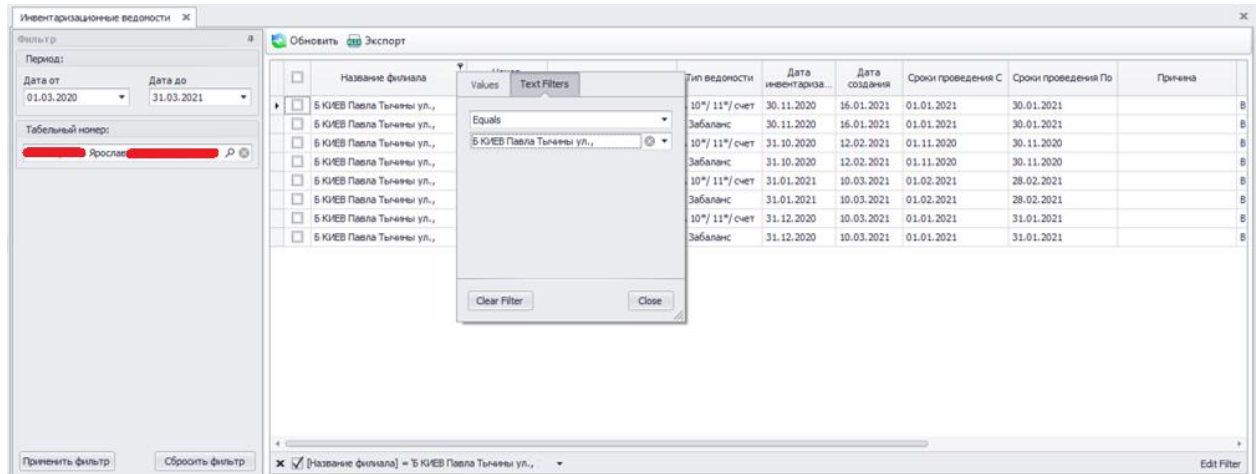


Рисунок 4.8. – Застосування текстового фільтру

3) виберемо текстовий фільтр «ends with» та введемо значення «1»:
отримаємо набір даних, в якому назва філіалу закінчується на введений нами текст (див. рис. 4.9).

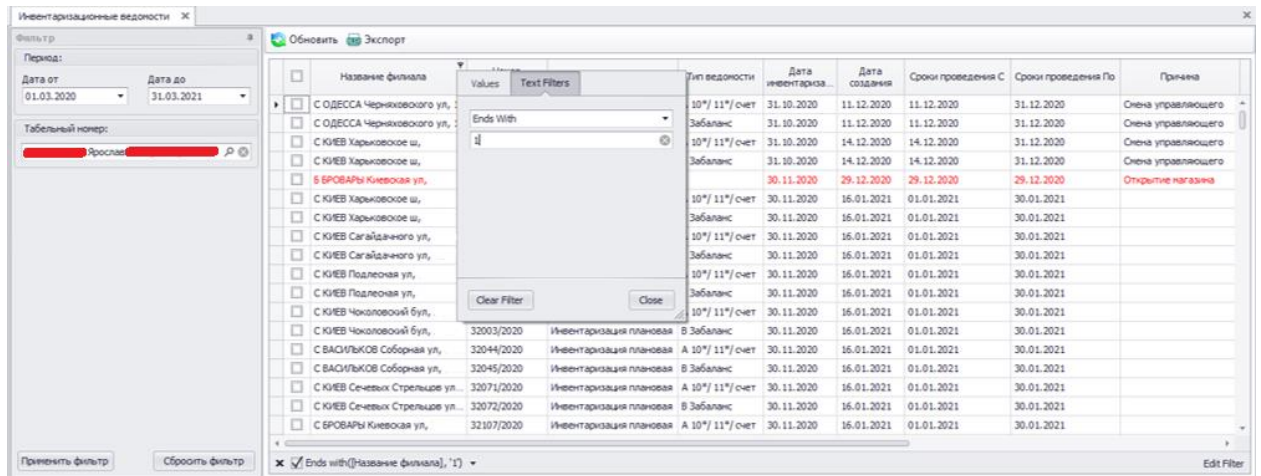


Рисунок 4.9. – Застосування текстового фільтру

2. На стовпчику «Тип відомості» натискаємо на значок фільтру, з'являється вікно, в якому ми можемо вибрати вид вибірки по стовпцю з типом відомостей. На вкладці «values» можна вибрати типи, які цікавлять користувача (див. рис. 4.10).

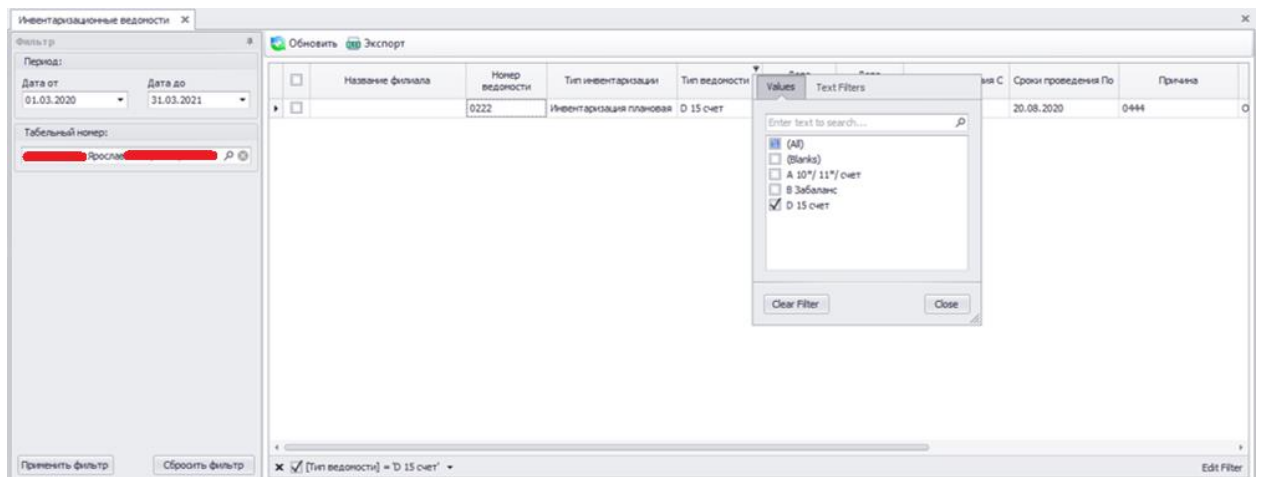


Рисунок 4.10. – Вибір типів відомостей

3. На стовпчику «Тип інвентаризації» натискаємо на шапку стовпця – автоматично відбувається сортування значень. В залежності від того, як направлена стрілочка, можна зрозуміти, в якому порядку відсортовані значення. Наприклад, за спаданням (див. рис. 4.11).

Название филиала	Номер ведомости	Тип инвентаризации	Тип ведомости	Дата инвентаризации	Дата создания	Сроки проведения С	Сроки проведения По	Причина
С ДРУЖКОВКА Соборная ул,	0222	Инвентаризация плановая	О 15 счет	01.07.2020	17.08.2020	10.08.2020	20.08.2020	0444
С ДРУЖКОВКА Соборная ул,	00007/2020	Инвентаризация плановая	А 10*/11*/счет	30.06.2020	25.08.2020	05.06.2020	15.06.2020	TEST RQA-400
С ДРУЖКОВКА Соборная ул,	00008/2020	Инвентаризация плановая	В Забаланс	30.06.2020	25.08.2020	05.06.2020	15.06.2020	TEST RQA-400
С ДРУЖКОВКА Соборная ул,	35584/2020	Инвентаризация плановая	А 10*/11*/счет	30.06.2020	25.08.2020	05.06.2020	15.06.2020	TEST RQA-400
С ДРУЖКОВКА Соборная ул,	35585/2020	Инвентаризация плановая	В Забаланс	30.06.2020	25.08.2020	05.06.2020	15.06.2020	TEST RQA-400
С КИЕВ Филатова ул,	31891/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Филатова ул,	31892/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Закревского ул,	31894/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Закревского ул,	31895/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Малевина ул,	31897/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Малевина ул,	31898/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Карельский пер,	31900/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Карельский пер,	31901/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Северная ул,	31903/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Северная ул,	31904/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Правды пр,	31906/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Правды пр,	31907/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Драйзера ул,	31909/2020	Инвентаризация плановая	А 10*/11*/счет	30.11.2020	16.01.2021	01.01.2021	30.01.2021	
С КИЕВ Драйзера ул,	31910/2020	Инвентаризация плановая	В Забаланс	30.11.2020	16.01.2021	01.01.2021	30.01.2021	

Рисунок 4.11. – Сортування значень за спаданням

4. На стовпчику «Дата інвентаризації» натискаємо на значок фільтру, з'являється вікно, в якому ми можемо вибрати вид вибірки по стовпцю з датою. На вкладці «date filters» можна вибрати один з запропонованих періодів (див. рис. 4.12).

Название филиала	Номер ведомости	Тип инвентаризации	Тип ведомости	Дата инвентаризации	Дата создания	Сроки проведения С	Сроки проведения По	Причина
С ОСТЕР Б. Мельничинского ул,		Инвентаризация внеплановая		31.01.2021				Смена МОП
С КИЕВ Драгочирова ул,		Инвентаризация внеплановая		31.01.2021				Открытие магазина
С КИЕВ Харьковское ш,		Инвентаризация внеплановая		31.01.2021				Закрытие магазина
С БОРИСПОЛЬ Киевская тр,	00008/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Смена управляющего
С БОРИСПОЛЬ Киевская тр,	00009/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Смена управляющего
С КИЕВ Милославская ул,	00011/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Смена управляющего
С КИЕВ Милославская ул,	00012/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Смена управляющего
С КИЕВ Милославская ул,	00014/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Смена управляющего
С КИЕВ Милославская ул,	00015/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Смена управляющего
С КИЕВ Драйзера ул,	00017/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Смена управляющего
С КИЕВ Драйзера ул,	00018/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Смена управляющего
С БАХМУТ Торговая ул,	00023/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Закрытие магазина
С БАХМУТ Торговая ул,	00024/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Закрытие магазина
С КИЕВ Щербакоского Дачнп...	00029/2021	Инвентаризация внеплановая	А 10*/11*/счет	31.01.2021				Закрытие магазина
С КИЕВ Щербакоского Дачнп...	00030/2021	Инвентаризация внеплановая	В Забаланс	31.01.2021				Закрытие магазина
С КИЕВ Филатова ул,	00056/2021	Инвентаризация плановая	А 10*/11*/счет	31.01.2021				
С КИЕВ Филатова ул,	00057/2021	Инвентаризация плановая	В Забаланс	31.01.2021				
С КИЕВ Закревского ул,	00059/2021	Инвентаризация плановая	А 10*/11*/счет	31.01.2021				

Рисунок 4.12. – Застосування фільтру по даті

Розглянемо приклад фільтрації даних та який вигляд має вікно редагування фільтра даних при цьому:

1) По стовпцю «Назва філіалу» – текстовий фільтр: назва філіалу закінчується на введене значення «7» (див. рис. 4.13);

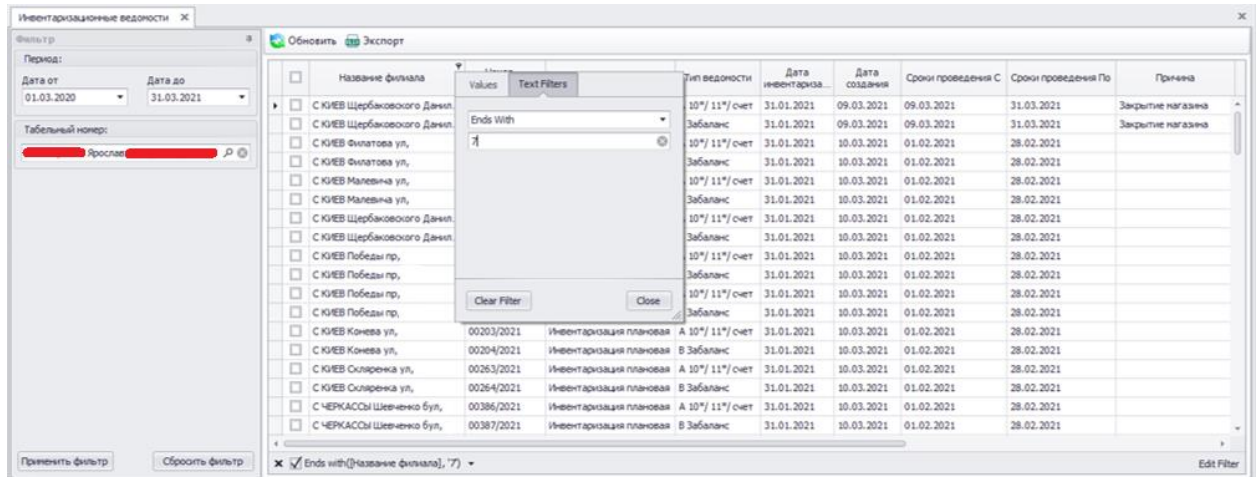


Рисунок 4.13. – Застосування текстового фільтра

2) По стовпцю «Тип інвентаризації» – фільтр по значенням: вибраний тип «Планова інвентаризація» (див. рис. 4.14);

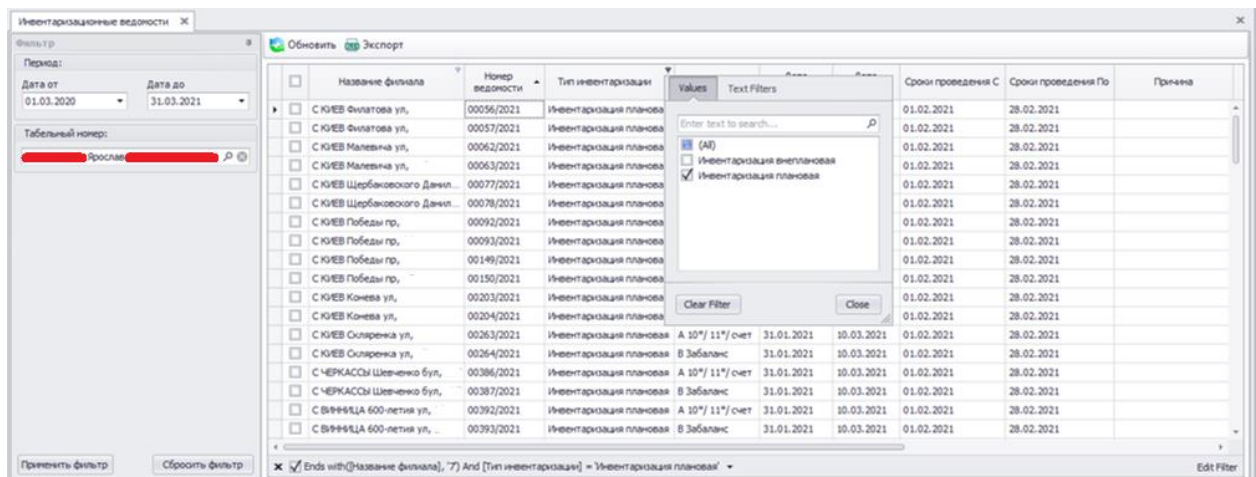


Рисунок 4.14. – Застосування фільтра по значенням

3) По стовпцю «Дата інвентаризації» – фільтр по періоду: вибрано дати цього року (див. рис. 4.15);

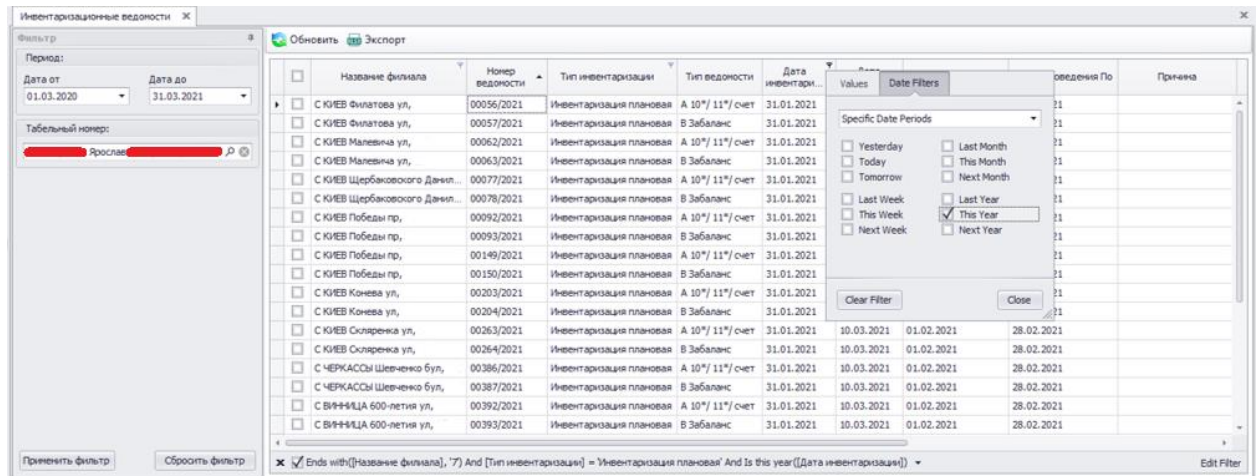


Рисунок 4.15. – Застосування фільтру по даті

4) По стовпцю «Статус» – фільтр по значенням: вибраний статус «В роботі» (див. рис. 4.16).

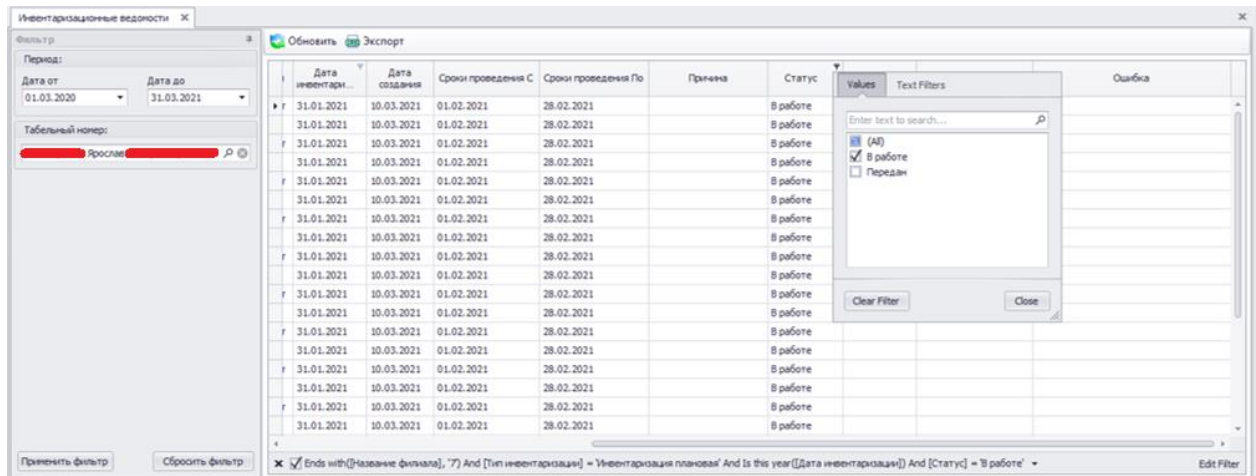


Рисунок 4.16. – Застосування фільтру по значенням

Внизу таблиці з даними можна побачити всі вибрані умови (див. рис. 4.17).

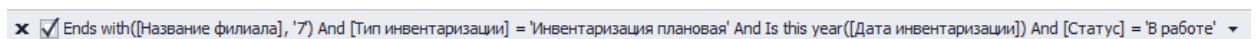


Рисунок 4.17. – Вибрані умови

При натисканні на кнопку «Edit Filter» можна відредагувати вибрані умови фільтрації даних (див. рис. 4.18).

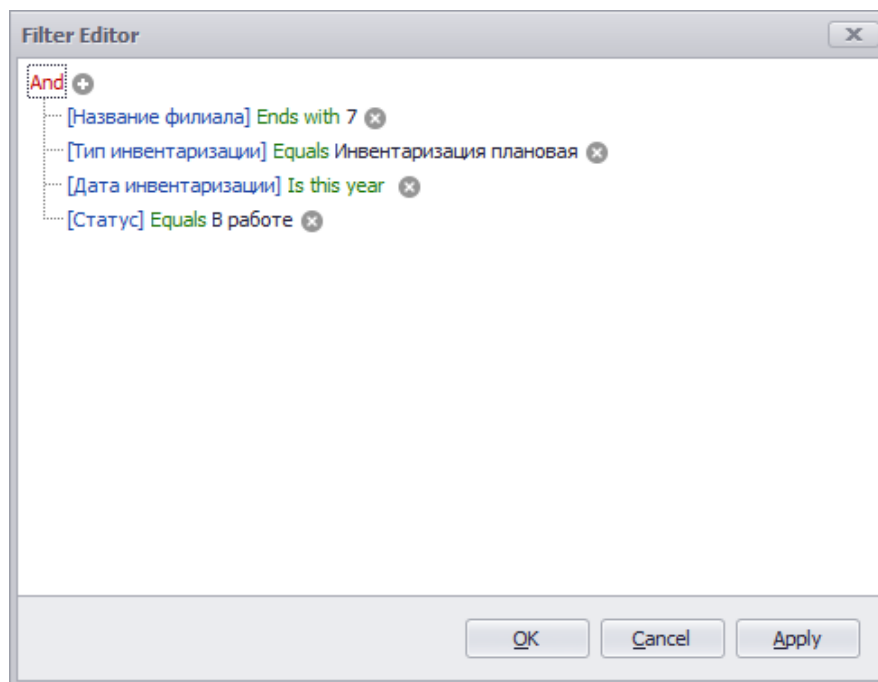


Рисунок 4.18. – Редагування умов фільтрації

В даній вкладці доступна функція експорту вибраних даних в форматі .xlsx, які відображаються в таблиці, для цього потрібно натиснути кнопку «Експорт» (див рис. 4.19).

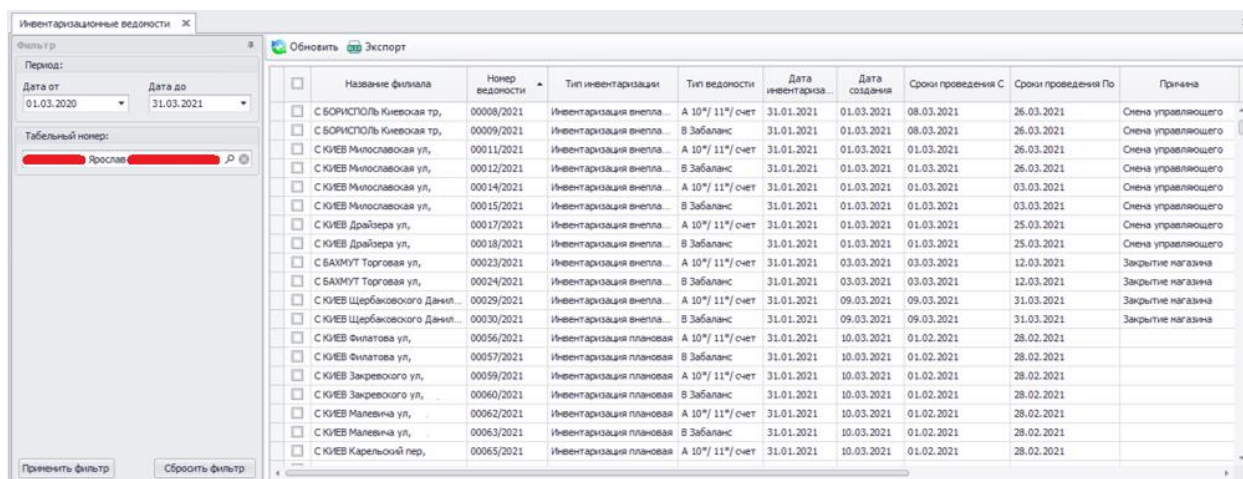


Рисунок 4.19. – Экспорт данных

Щоб скинути умови фільтру, потрібно натиснути на кнопку «Скинути фільтр», після цього поле з табельним номером стає порожнім, а період змінюється на перше число минулого місяця до останнього числа минулого місяця, при цьому дані в таблиці залишаються (див. рис. 4.22)

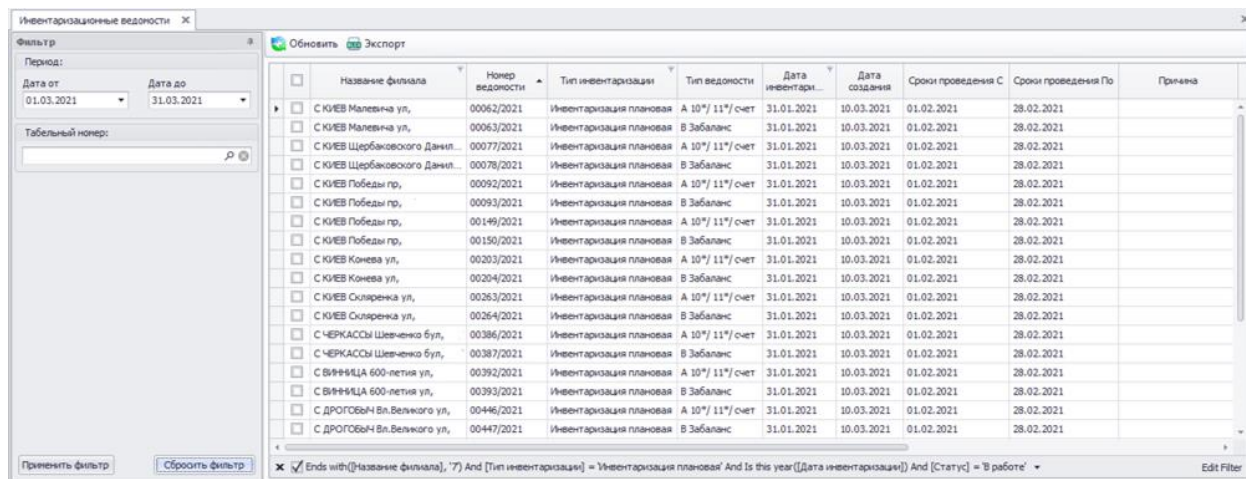


Рисунок 4.22. – Очищення фільтру

Перейдемо до деталізації відомостей на прикладі вибраних документів (див. рис. 4.23).

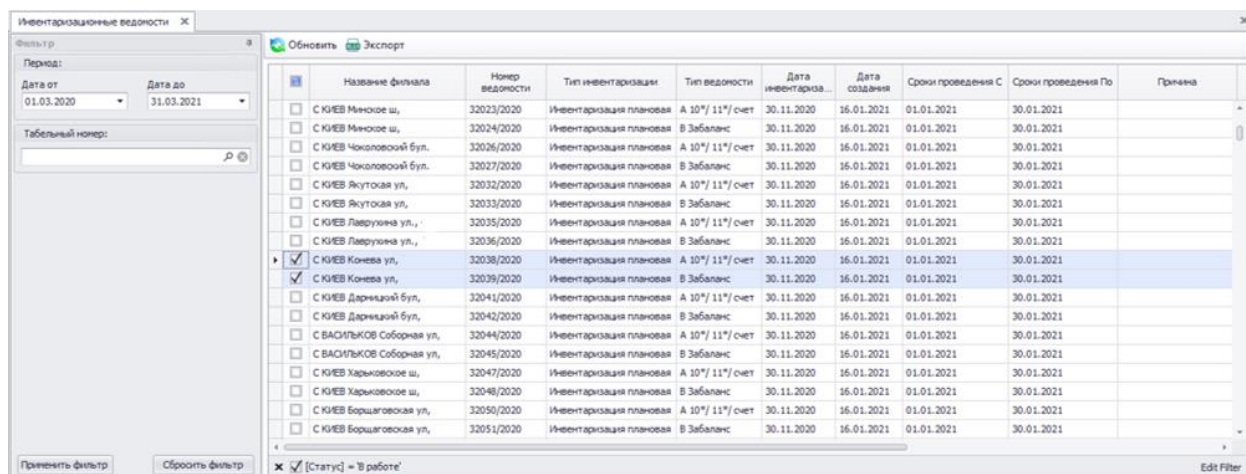
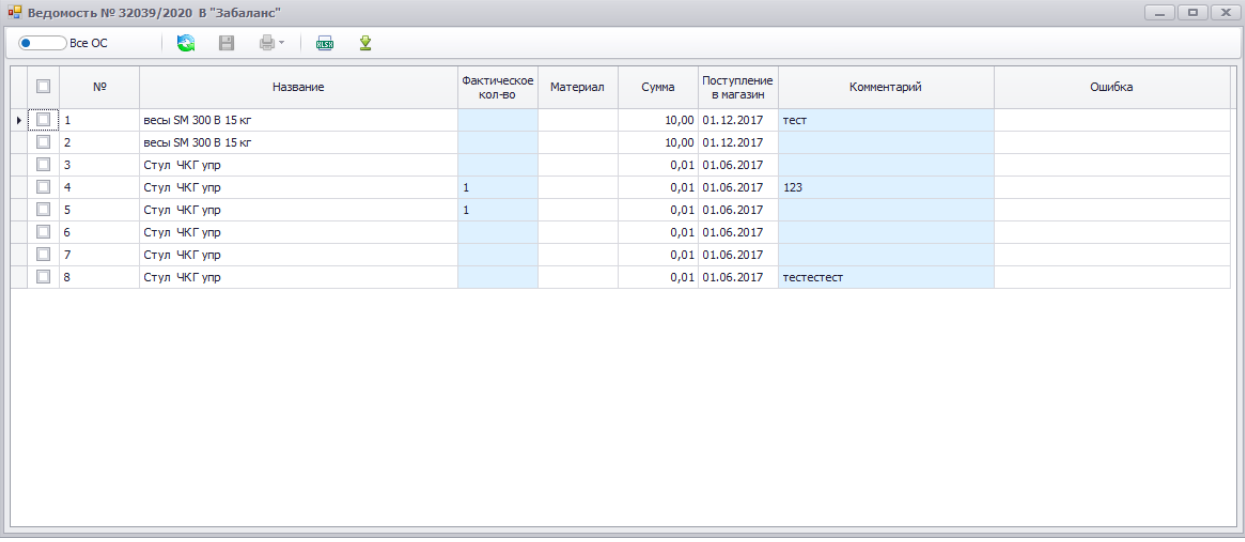


Рисунок 4.23. – Вибір документів

4.2 Вкладка роботи з документом

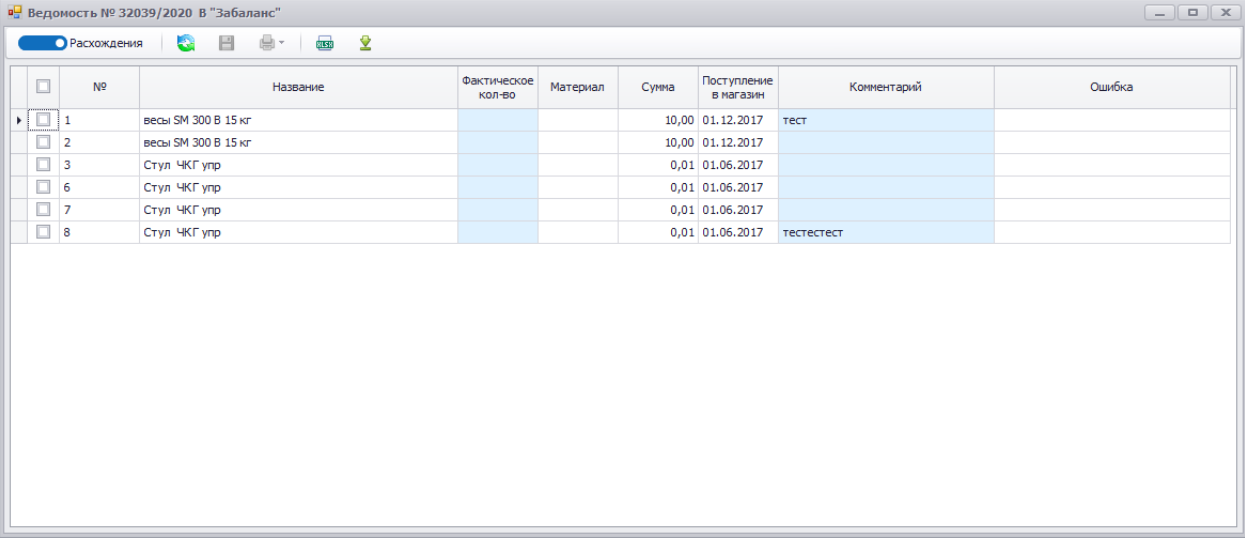
При подвійному кліку на рядку таблиці відкривається нове вікно, в якому відображаються дані обраного документу. Якщо відомість доступна до редагування, стовпці «Фактична кількість» та «Коментар» будуть зафарбовані синім та доступні для введення (див. рис. 4.24).



№	Название	Фактическое кол-во	Материал	Сумма	Поступление в магазин	Комментарий	Ошибка
1	весы SM 300 B 15 кг			10,00	01.12.2017	тест	
2	весы SM 300 B 15 кг			10,00	01.12.2017		
3	Стул ЧКГ упр			0,01	01.06.2017		
4	Стул ЧКГ упр	1		0,01	01.06.2017	123	
5	Стул ЧКГ упр	1		0,01	01.06.2017		
6	Стул ЧКГ упр			0,01	01.06.2017		
7	Стул ЧКГ упр			0,01	01.06.2017		
8	Стул ЧКГ упр			0,01	01.06.2017	тесттест	

Рисунок 4.24. – Вікно деталізації відомості

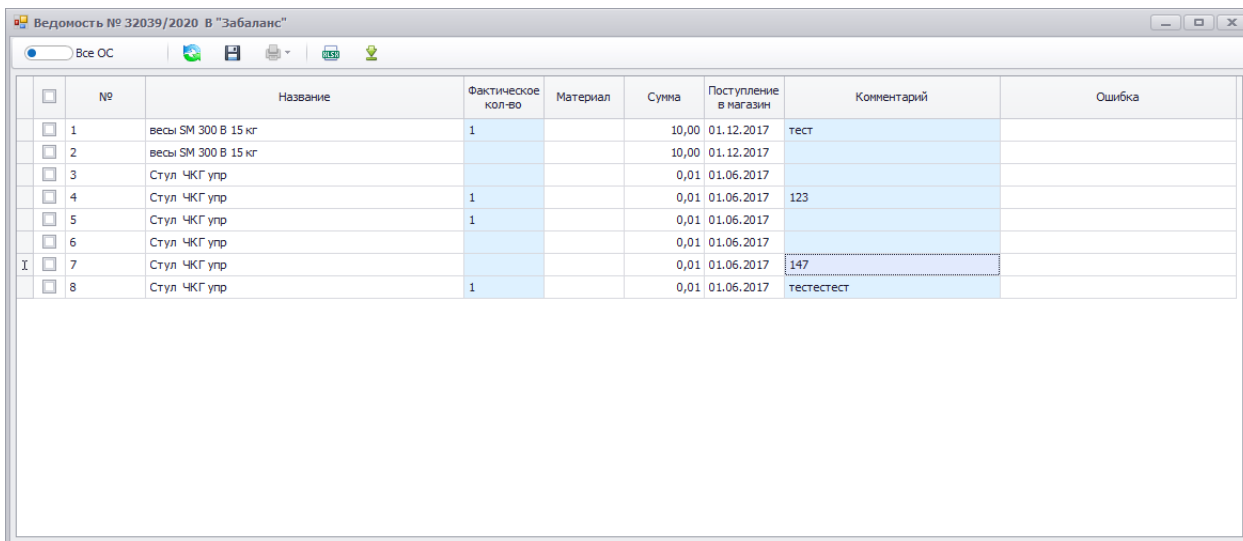
У верхньому лівому кутку знаходиться перемикач, якщо переключити його, можна побачити рядки, в яких фактична кількість не задана, в початковому положенні – будуть відображатися всі дані (див. рис. 4.25)



№	Название	Фактическое кол-во	Материал	Сумма	Поступление в магазин	Комментарий	Ошибка
1	весы SM 300 B 15 кг			10,00	01.12.2017	тест	
2	весы SM 300 B 15 кг			10,00	01.12.2017		
3	Стул ЧКГ упр			0,01	01.06.2017		
6	Стул ЧКГ упр			0,01	01.06.2017		
7	Стул ЧКГ упр			0,01	01.06.2017		
8	Стул ЧКГ упр			0,01	01.06.2017	тесттест	

Рисунок 4.25. – Використання перемикача

Введемо нові дані в стопці «Фактична кількість» та «Коментар» (див. рис. 4.26).



	№	Название	Фактическое кол-во	Материал	Сумма	Поступление в магазин	Комментарий	Ошибка
☐	1	весы SM 300 B 15 кг	1		10,00	01.12.2017	тест	
☐	2	весы SM 300 B 15 кг			10,00	01.12.2017		
☐	3	Стул ЧКГ упр			0,01	01.06.2017		
☐	4	Стул ЧКГ упр	1		0,01	01.06.2017	123	
☐	5	Стул ЧКГ упр	1		0,01	01.06.2017		
☐	6	Стул ЧКГ упр			0,01	01.06.2017		
☒	7	Стул ЧКГ упр			0,01	01.06.2017	147	
☐	8	Стул ЧКГ упр	1		0,01	01.06.2017	тесттест	

Рисунок 4.26. – Введення даних

Для зберігання змін, потрібно натиснути на кнопку «Зберегти», яка зображена дискетою. При закриванні вікна, в якому присутні незбережені зміни з’являється вікно-попередження (див. рис. 4.27).

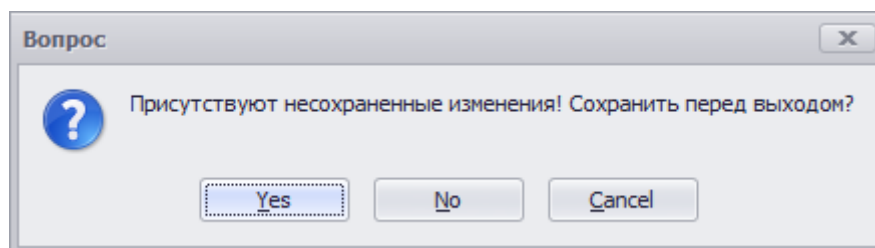


Рисунок 4.27. – Вікно-попередження

Після успішного зберігання змін, внесених в відомість, з’являється вікно, яке інформує користувача про це (див. рис. 4.28).

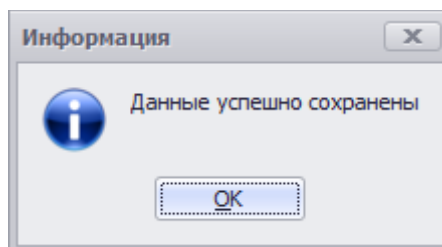


Рисунок 4.28. – Вікно-повідомлення

В даній вкладці доступна функція експорту вибраних даних в форматі .xlsx, які відображаються в таблиці, для цього потрібно натиснути кнопку «Експорт». Після натискання, з'являється вікно, в якому потрібно вибрати шлях зберігання файлу та задати його ім'я (див. рис. 4.29).

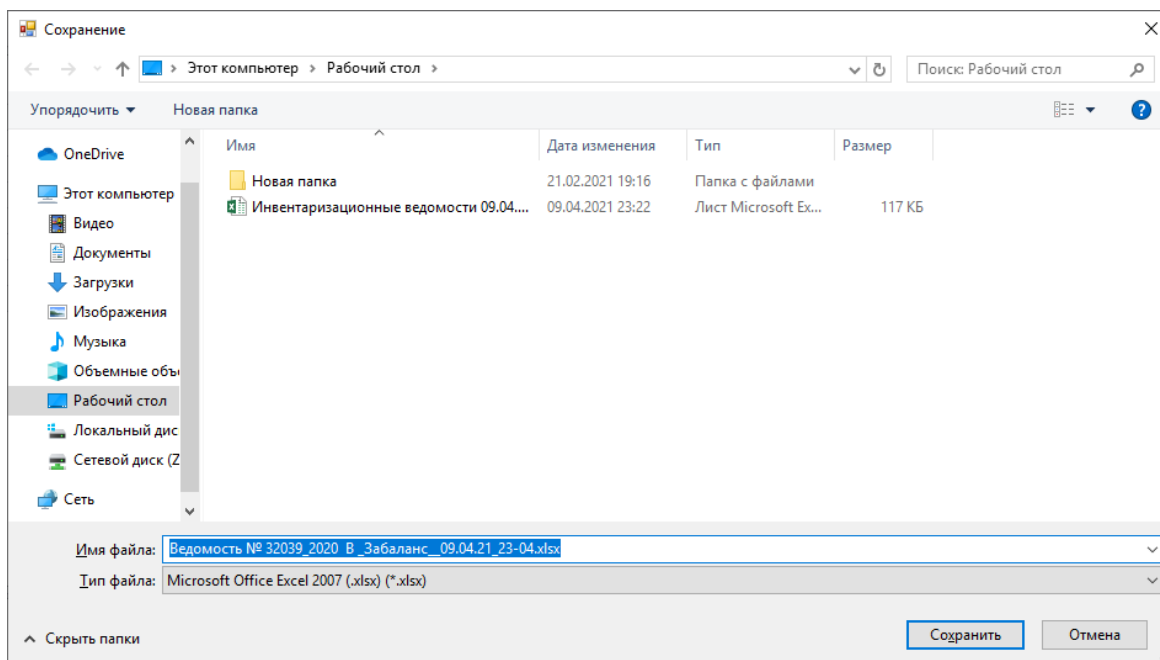


Рисунок 4.29. – Вибір шляху зберігання файлу

Далі відкриється експортований файл, який зберігав користувач в програмі Excel (див. рис. 4.30)

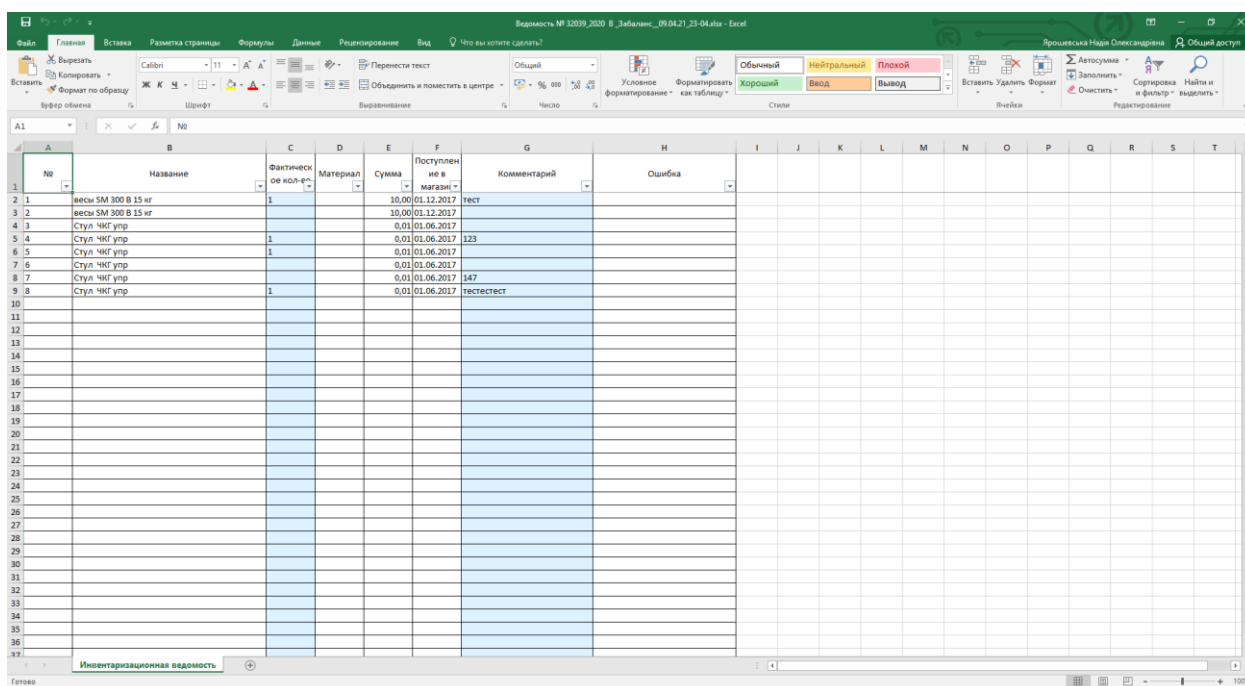


Рисунок 4.30. – Експортований файл

У даній вкладці також доступна фільтрація даних в таблиці, розглянемо приклад: на стовпчику «Надходження в магазин» натискаємо на значок фільтру, з'являється вікно, в якому ми можемо вибрати вид вибірки по стовпцю з датою. На вкладці «date filters» можна вибрати один з запропонованих періодів, виберемо «after» та введемо потрібну дату, в даному випадку «12.04.2018» (див. рис. 4.31).

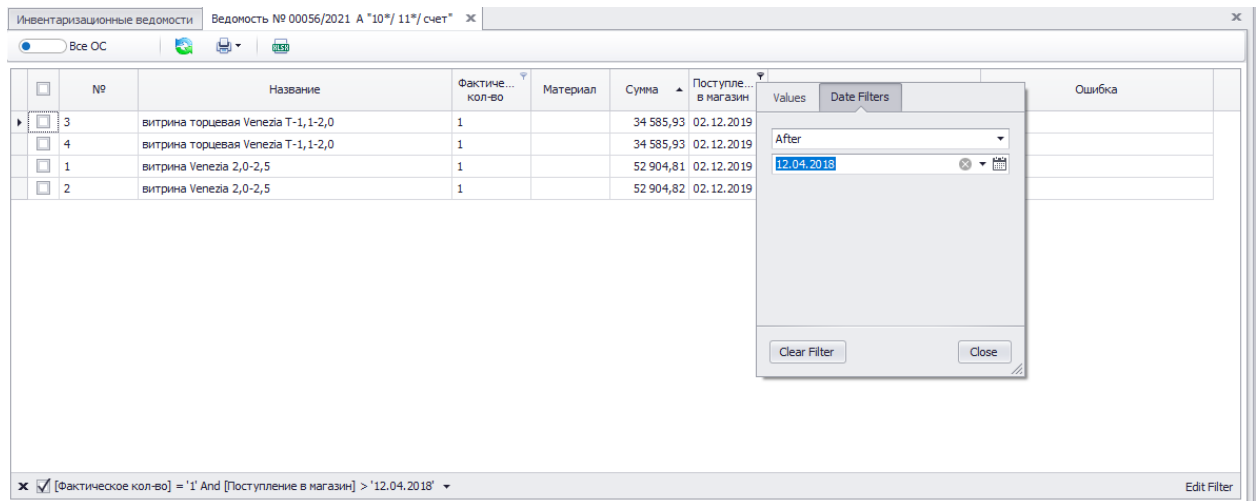


Рисунок 4.31. – Застосування фільтру по даті

4.3 Формування звітів

Розглянемо, як користувач може сформувати звіт по вибраним відомостям:

1) Для формування інвентаризаційної відомості потрібно відкрити меню друку (іконка принтеру на панелі) та вибрати пункт «Друк інвентаризаційної відомості» (див. рис. 4.32).

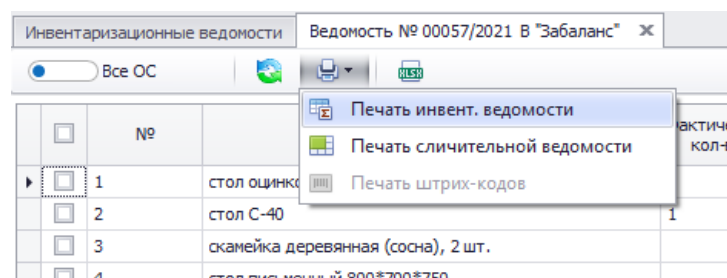


Рисунок 4.32. – Друк інвентаризаційної відомості

Після цього відкривається вікно, в якому можна переглянути інвентаризаційну відомість по відкритому документу (див. рис. 4.33 – 4.34).

[illegible]

Рисунок 4.33. – Інвентаризаційна відомість

[illegible]

Рисунок 4.34. – Інвентаризаційна відомість

2) Для формування порівняльної відомості потрібно відкрити меню друку (іконка принтера на панелі) та вибрати пункт «Друк порівняльної відомості» (див. рис. 4.35).

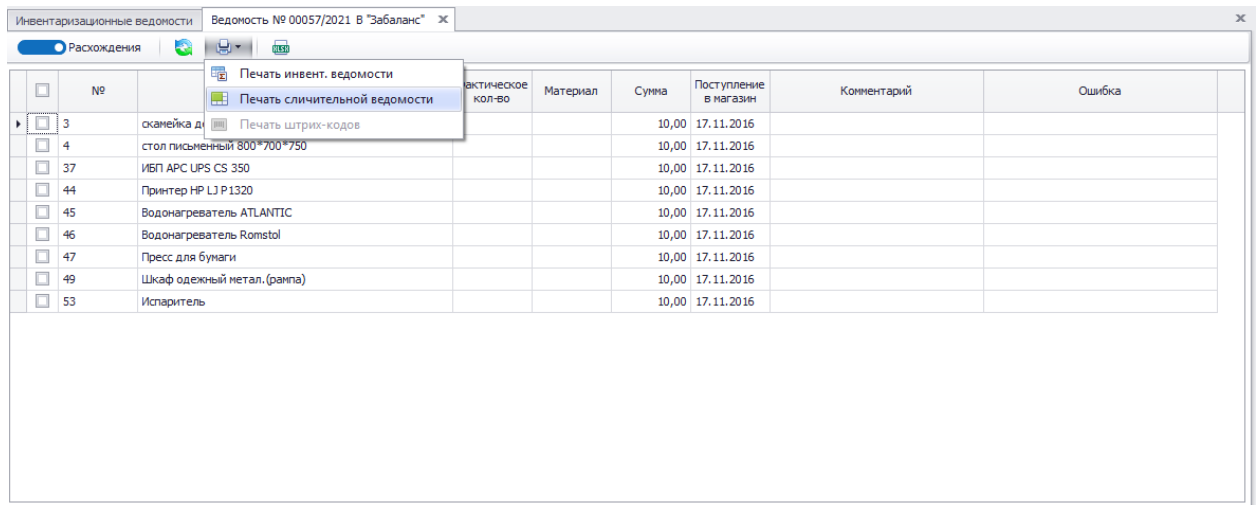


Рисунок 4.35. – Друк порівняльної відомості

Після цього відкривається вікно, в якому можна переглянути порівняльну відомість по відкритому документу (див. рис. 4.36).

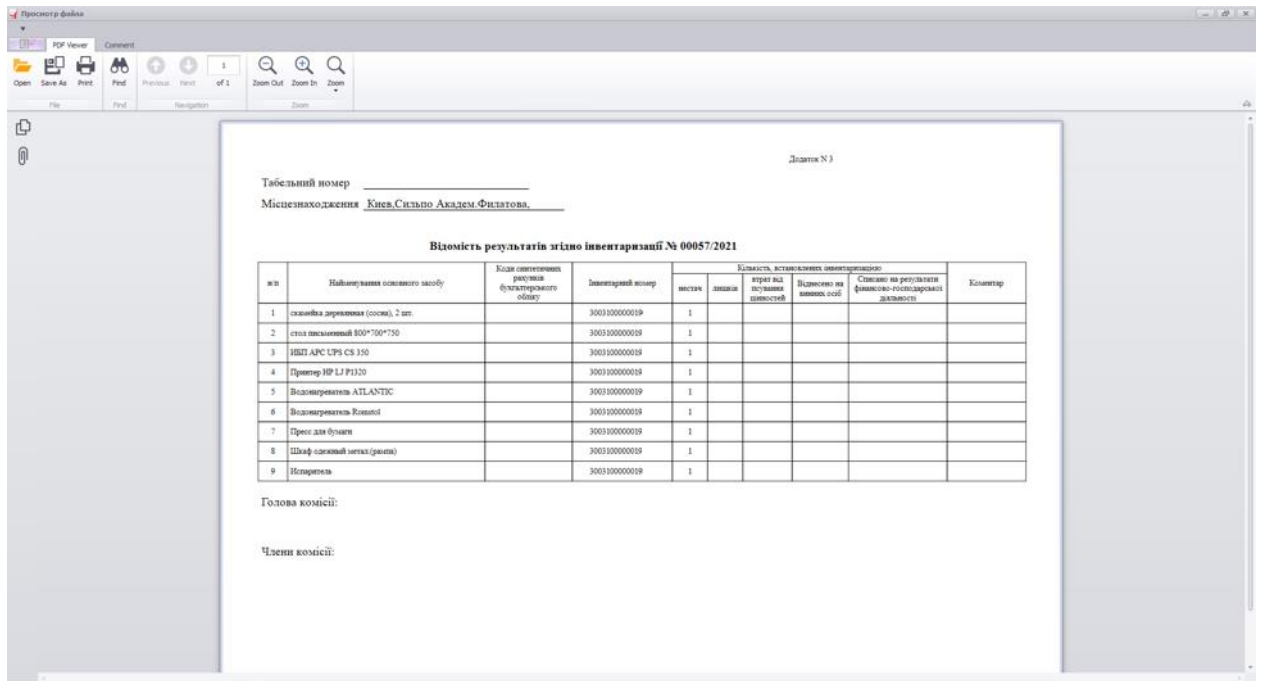


Рисунок 4.36. – Порівняльна відомість

Для формування штрих-кодів вибраних товарів, потрібно вибрати в таблиці потрібні товари, відкрити меню друку (іконка принтера на панелі) та вибрати пункт «Друк штрих-кодів» (див. рис. 4.37).

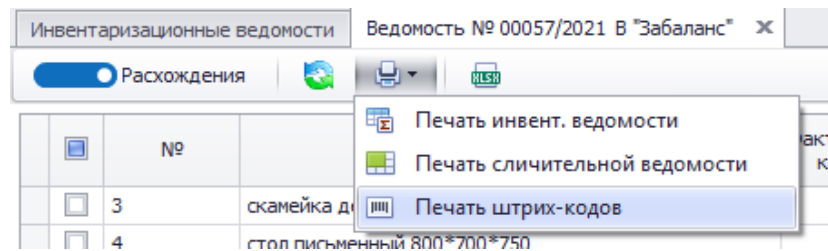


Рисунок 4.37. – Друк штрих-кодів

Після цього відкривається вікно, в якому можна переглянути штрих-коди вибраних товарів (див. рис. 4.38).

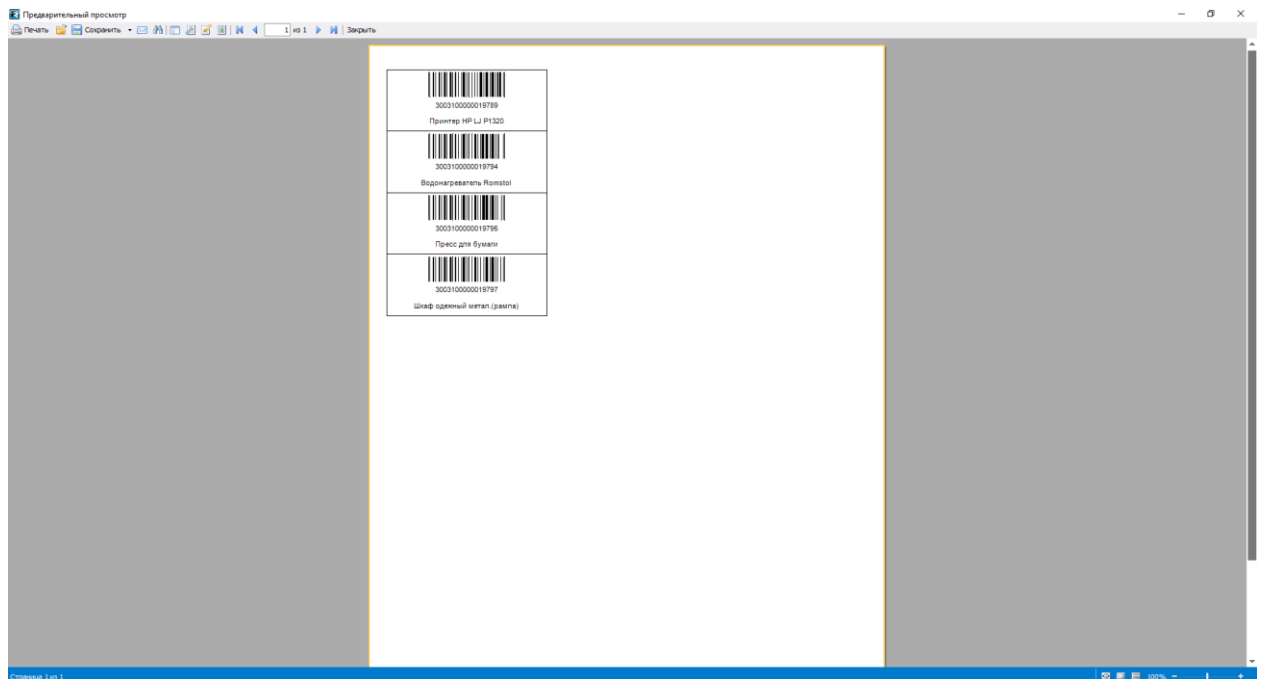


Рисунок 4.38. – Штрих-коди

Висновки

У цьому розділі описана інструкція для користувача по роботі у вкладках: кроки для отримання даних, експортування їх у Excel-документ та формування звітів.

ВИСНОВКИ

Під час розробки дипломного проєкту було досліджено існуючі на ринку застосунки, виявлено їх переваги і недоліки, розроблено загальну концепцію програми, поставлено задачу, проведено аналіз об'єкту програмування та визначено вимоги до апаратного та програмного забезпечення. Були спроектовані шаблони документів для застосунку, розроблений інтерфейс та дизайн застосунку, призначеного для роботи зі звітною документацією підприємства торгово-промислової групи. Також розроблено сам програмний продукт та створено інструкції до нього.

Для цього було досліджено існуючі на ринку застосунки роботи зі звітною документацією, обрано та описано інформаційне, організаційне, програмне та технічне забезпечення, мову програмування та середовище створення застосунку, спроектовано шаблони документів, розроблено інтерфейс та дизайн програмного продукту, призначеного для роботи зі звітною документацією підприємства торгово-промислової групи.

Обраною мовою розробки стала – C#. Для розробки ПЗ використовувалась технологія WF (Windows Form Application) в середовищі розробки Microsoft Visual Studio 2019 з застосуванням DevExpress. Для створення шаблонів документів був використаний генератор звітів FastReport.

Платформа .NET Framework дозволяє розробляти інтелектуальні клієнтські програми з багатьма можливостями, уникаючи проблем.

Windows Forms вже вплинув на розробки для Windows. Тепер, коли додаток знаходиться на початковій стадії проєктування, прийняти рішення про те, чи потрібно будувати веборієнтовані додатки або ж повністю клієнтські, стало трохи важче. Клієнтські програми Windows можуть бути розроблені швидко і ефективно, при цьому вони пропонують користувачам набагато ширші можливості.

Робота має практичне значення і може використовуватися користувачами для знаходження інформації про документи, їх редагування, створення та отримання звітів.

Було спроектовано та розроблено додаток для користувача, який використовується для ведення документації по основним засобам і генерації аналітичних звітів підприємства торгово-промислової групи.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Програма «1С:Підприємство» [Електронний ресурс] // Програма «1С:Підприємство. Бухгалтерія для України». – 2018. – Режим доступу до ресурсу: <https://accounting.ua/ua/1C-pidpriyemstvo-8.-buhgalteriya-dlya-ukrayini/>.
2. Бесплатные программы для сдачи отчетности в Украине [Електронний ресурс] // ІТСua. – 2021. – Режим доступу до ресурсу: <https://itc.ua/blogs/besplatnyie-programmyi-dlya-sdachi-otchetnosti-v-ukraine/>.
3. TFORMer Designer: Печать переменных данных [Електронний ресурс] // Tec-it. – 2021. – Режим доступу до ресурсу: <https://www.tec-it.com/ru/software/report-generator/report-designer/variable-data-printing/Default.aspx>.
4. Mr. Doc [Електронний ресурс] // Mr.Doc. – 2021. – Режим доступу до ресурсу: <http://mrdoc.org>.
5. Power BI [Електронний ресурс] // Microsoft Power BI. – 2020. – Режим доступу до ресурсу: <https://powerbi.microsoft.com/ru-ru/business-analysts/>.
6. Power BI [Електронний ресурс] // Microsoft Power BI. – 2020. – Режим доступу до ресурсу: <https://powerbi.microsoft.com/ru-ru/what-is-power-bi/>.
7. Руководство по Entity Framework [Електронний ресурс] // METANIT.COM. – 2016. – Режим доступу до ресурсу: <https://metanit.com/sharp/entityframework/>.
8. Microsoft ADO.NET / Девід Скеппа – Петербург: Русская Редакция, 2007. – 784 с.
9. Фленов М. Е. Библия C#/ М. Е. Фленов. – Петербург: БХВ, 2010. – 560 с.
10. C# 7.0 in a Nutshell: The Definitive Reference/ Joseph Albahari. – O'Reilly Media, 2017. – 1150 с.

11. Що таке бібліотека DLL? [Електронний ресурс] // Підтримка від Microsoft. – 2018. – Режим доступу до ресурсу: <https://support.microsoft.com/uk-ua/help/815065/what-is-a-dll>.

12. Введение в MS SQL Server и T-SQL [Електронний ресурс] // METANIT.COM. – 2017. – Режим доступу до ресурсу: <https://metanit.com/sql/sqlserver/1.1.php>.

13. DevExpress [Електронний ресурс] // Allsoft. – 2021. – Режим доступу до ресурсу: <https://allsoft.ru/software/vendors/developer-express/devexpress-winforms/>.

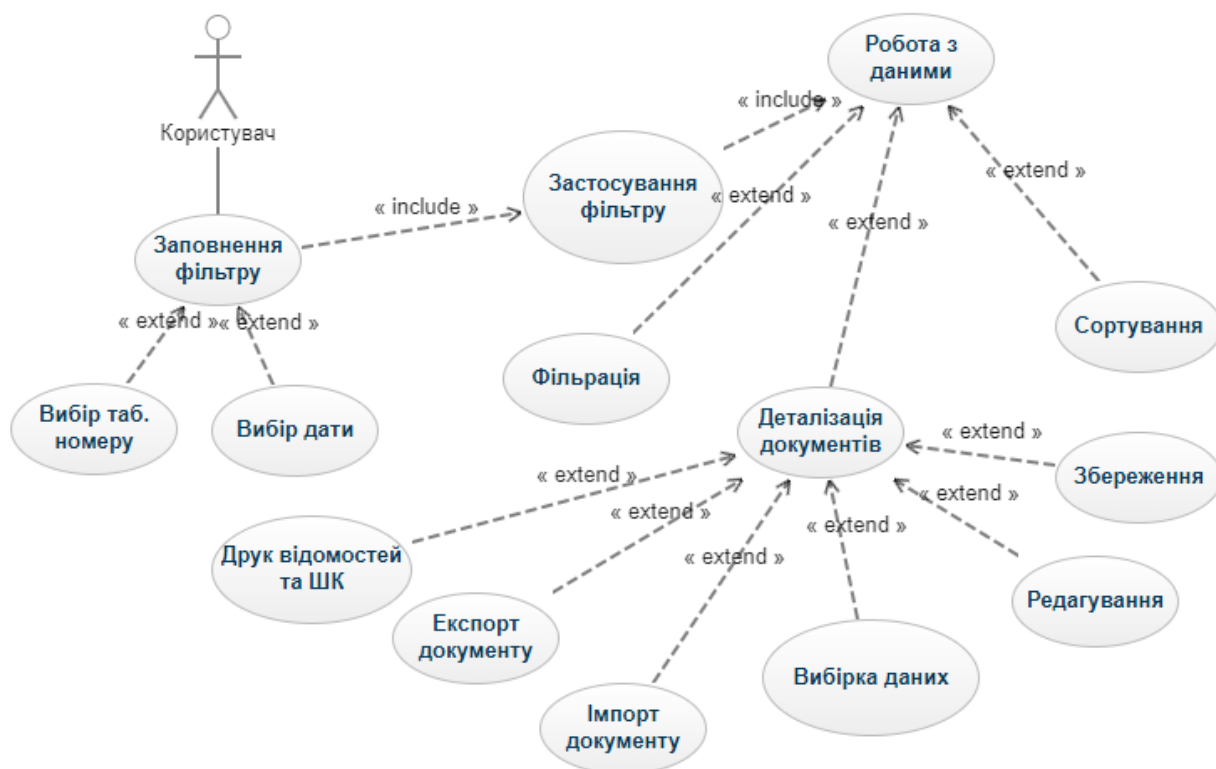
14. Библиотека генерации отчётов и создания документов FastReport .NET [Електронний ресурс] // FastReport .NET. – 2021. – Режим доступу до ресурсу: <https://www.fastreport.ru/ru/>.

15. Инвентаризация [Електронний ресурс] // Главная книга. – 2018. – Режим доступу до ресурсу: <https://glavkniga.ru/situations/s503086>.

ДОДАТКИ

ДОДАТОК А

Діаграма прецедентів



ДОДАТОК Б

Метод експортування даних в Excel-документ

```

private void ExportToFile(GridView grid, string ExportHeaderName, string SheetName)
{
    try
    {
        if (grid.DataSource == null) return;
        ExportHeaderName = string.Format("{0}_{1}",
StaticHelpers.ReplaceInvalidChars(ExportHeaderName),
DateTime.Now.ToString("dd.MM.yy_HH-MM"));
        using (SaveFileDialog sfdExport = new SaveFileDialog
        {
            Filter = "Microsoft Office Excel 2007 (.xlsx) | *.xlsx | Текст (*.txt) | *.txt"
        })
        {
            if (grid.DataSource is DataView dt)
            {
                sfdExport.FilterIndex = 1;
                sfdExport.RestoreDirectory = true;
                string exportHeader = string.Format(ExportHeaderName,
DateTime.Now.ToString("yyyy-MM-dd"));
                sfdExport.FileName = exportHeader;
                if (sfdExport.ShowDialog() == DialogResult.OK)
                {
                    if (sfdExport.FileName.ToLower().EndsWith(".xlsx"))
                    {
                        DevExpress.XtraPrinting.XlsxExportOptionsEx options = new
DevExpress.XtraPrinting.XlsxExportOptionsEx()
                        {
                            ExportHyperlinks = true,
                            SheetName = SheetName,
                            ShowGridLines = true,
                            ExportMode =
DevExpress.XtraPrinting.XlsxExportMode.SingleFile,
                            TextExportMode =
DevExpress.XtraPrinting.TextExportMode.Text
                        };
                        options.CustomizeDocumentColumn += (e) =>
                    {

```

```

        if ((e.ColumnFieldName == "inventoryNumber") ||
(e.ColumnFieldName == "barcode") || (e.ColumnFieldName == "quantity"))
        {
            e.DocumentColumn.Formatting.NumberFormat = "@";
        }
    };
    grid.ExportToXlsx(sfdExport.FileName, options);
    using (System.Diagnostics.Process p = new
System.Diagnostics.Process())
    {
        p.StartInfo.FileName = sfdExport.FileName;
        p.Start();
    }
    }
    if (!sfdExport.FileName.ToLower().EndsWith(".xls") &&
!sfdExport.FileName.ToLower().EndsWith(".xlsx"))
    {
        grid.ExportToText(sfdExport.FileName);
    }
}
}
}
}
catch (Exception ex)
{
    MB.error(ex.Message);
}
}

```

ДОДАТОК В

Метод для друку звіту інвентаризаційної відомості

```
private void PrintFastReport(DocumentType _printedFormId, string _listPath)
{
    try
    {
        using (DataSet ds = new DataSet())
        {
            using (DataTable dtFilters = new DataTable("Filters"))
            {
                dtFilters.Columns.Add("documentId", typeof(int));
                dtFilters.Columns.Add("printedFormId", typeof(int));
                dtFilters.Rows.Add(_documentID, _printedFormId);
                ds.Tables.Add(dtFilters);
                Helpers.CheckRightsVsExecuteAsync(this,
                    (o, d) =>
                    {
                        try
                        {
                            using (DataSet dbDS = Helpers.ParseExecuteReaderContract(o
as IDefaultContract))
                            {
                                DataTable src = dbDS.Tables[0];
                                ResourceManager.ResourceHandlers handlers = new
ResourceManager.ResourceHandlers()
                                {
                                    DataHandler = (x, y) =>
                                    {
                                        if (x == null)
                                            throw new Exception(string.Format("Ресурс не
найден: {0}", y));

                                        {
                                            rep.Report.Load(new MemoryStream(x));
                                            rep.GetDataSource("Table")?.Delete();

                                            rep.RegisterData(src, "Table");
                                            sys.WaitFor(() => rep.GetDataSource("Table") != null);
                                            rep.GetDataSource("Table").Enabled = true;
                                            rep.Prepare();
                                        }
                                    }
                                }
                            }
                        }
                    }
                )
            }
        }
    }
}
```

```

        rep.ShowPrepared(true, this);
    }
}
};
ResourceManager.GetResourceAsync(_listPath, handlers);
}
}
catch (Exception ex)
{
    MB.error(Ex.Message(ex));
}
}, _printingOperation, Serialization.Serialize(ds),
null, 600);
}
}
}
catch (Exception ex)
{
    MB.error(ex);
}
}

```