

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ «АНАЛІЗ
ЗАВОДСЬКОГО ЗАМОВЛЕННЯ» В ERP-СИСТЕМІ
SOFTWARE IMPLEMENTATION OF THE FACTORY ORDER ANALYSIS
FUNCTIONALITY IN THE ERP SYSTEM

Виконав: студент(ка) 4 курсу, групи КНТ-137
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

Сліпченко В.О.

(прізвище та ініціали)

Керівник Зеленьова І.Я.

(прізвище та ініціали)

Рецензент Терещенко Є.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти бакалавр
 Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. Субботін
 “ ” 20 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Сліпченко Владислави Олегівни

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація функціоналу "Аналіз заводського замовлення" в ERP-системі. Software Implementation of the Factory Order Analysis Functionality in the ERP System

керівник проєкту (роботи) Зеленьова Ірина Яківна, к.т.н., доцент,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) 5 травня 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Огляд аналогів й аналіз інструментів для розробки системи «Аналіз заводського Замовлення». 2. Розробка програмної частини системи «Аналіз заводського замовлення». 3. Тестування продукта та практичні рекомендації.

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

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-3 Основна частина	Зеленьова І.Я., доцент		
Нормоконтролер	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання “28” лютого 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	2 тиждень	Розділ 1
3	Розробка програми.	3 тиждень	Розділ 2
4	Тестування програмного забезпечення.	4 тиждень	Розділ 3
5	Оформлення пояснювальної записки та документів до неї.	5-6 тижні	Додатки
6	Нормоконтроль та рецензування.	7 тиждень	
7	Захист роботи.	8 тиждень	

Студент(ка)

(підпис)

Сліпченко В.О.

(прізвище та ініціали)

Керівник проєкту (роботи)

(підпис)

Зеленьова І.Я.

(прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 88 с., 27 рис., 29 табл., 2 дод., 12 джерел.

АНАЛІЗ ЗАВОДСЬКОГО ЗАМОВЛЕННЯ, ДЕКСТОПЗАСТОСУНОК, ОПТИМІЗАЦІЯ ПРОВЕДЕННЯ СПИСАНЬ ЗАВОДСЬКИХ ЗАМОЛЕНЬ, X++, MICROSOFT DYNAMICS AX.

Об'єкт дослідження – програмні засоби для розробки функціоналу, що допоможе оптимізувати проведення аналізу заводського замовлення.

Предмет дослідження – ERP-система Microsoft Dynamics AX.

Мета роботи – створення програмного застосунку для оптимізації витрат робочого часу при проведенні аналізу наявності ТМЦ (товарно-матеріальних цінностей) .

Матеріали, методи та технічні засоби: функціональне програмування, мови програмування X++, ноутбук з процесором Intel Core i5 під управлінням операційної системи Microsoft.

Результати. Створено оптимізований програмний застосунок на основі ERP системи Microsoft Dynamics AX та програмної мови X++, за допомогою якого користувач може проводити аналіз наявності ТМЦ для списань під виробничі замовлення.

Висновки. Розроблено програмний застосунок для пошуку, створення та перегляду аналізів наявності ТМЦ, з можливістю збереження або видалення даних, збереження налаштувань, тощо.

Галузь використання – працівниками та консультантами підрозділів підприємства, які працюють з системою MS Dynamics AX.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 88 pages, 27 figures, 29 tables, 2 appendixes, 12 sources.

FACTORY ORDER ANALYSIS, DEXT APPLICATION, OPTIMIZATION OF FACTORY REQUESTS, X ++, MICROSOFT DYNAMICS AX.

The object of research is software tools for functional development, which will help to optimize the analysis of the factory order.

The subject of research - ERP-system Microsoft Dynamics AX.

The purpose of the work - to create a software application to optimize the cost of working time when analyzing the availability of goods and materials.

Materials, methods and technical means: functional programming, X ++ programming languages, laptop with Intel Core i5 processor running Microsoft operating system.

Results. The most optimized software application based on ERP system Microsoft Dynamics AX and programming language X ++, with which the user can analyze the availability of goods for write-offs for production orders.

Conclusions. Developed a software application for finding, creating and viewing analyzes of the availability of goods and materials, with the ability to save or delete data, save settings, etc.

Field of use - employees and consultants of departments of the enterprise that work with the system MS Dynamics AX.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Огляд аналогів й аналіз інструментів для розробки системи «Аналіз заводського Замовлення».....	10
1.1 Аналіз предметної області.....	10
1.2 Аналогічні сучасні системи, що використовуються для розробки функціоналу.....	10
1.2.1 Критерії вибору системи.....	10
1.2.2 Огляд та порівняння ERP-систем.....	11
1.3 Основні теоретичні положення розробки системи.....	17
1.4 Архітектура системи.....	19
1.5 Висновок за розділом 1.....	26
2 Розробка програмної частини системи «Аналіз заводського замовлення»..	27
2.1 Постановка завдання розробки і дослідження.....	27
2.2 Загальна структура програми.....	27
2.2.1 Опис модифікації. Створення таблиць.....	27
2.2.2 Створення форми «Аналіз заводського замовлення»	31
2.2.3 Реалізація нижньої частини форми.....	45
2.3 Висновок за розділом 2.....	55
3 Тестування продукту та практичні рекомендації	56
3.1 Тестування роботи системи.....	56
3.2 Висновок за розділом 3.....	57
Висновки.....	59
Перелік джерел посилення.....	60
Додаток А Текст програми.....	62
Додаток Б Слайди презентації.....	81

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

AX	– Ахapta;
CRM	– Customer Relationship Management;
ERP	– Enterprise Resource Planning;
З/З	– заводське замовлення;
КД	– конструкторська документація;
ТМЦ	– товарно-матеріальних цінностей.

ВСТУП

Для успішної реалізації функціоналу «Аналіз заводського замовлення» буде використана ERP система Microsoft Dynamics AX. Це сучасна комплексна ERP система, що дозволяє управляти ресурсами підприємств корпоративного масштабу. Даний софт належить корпорації Microsoft і відноситься до групи додатків Microsoft Dynamics.

Основні споживачі AX складаються в основному з компаній, що використовують для своєї роботи програмне забезпечення на регулярній основі.

Microsoft Dynamics AX є системою ERP для середніх і великих підприємств. Це найнадійніша, масштабна і функціонально багата система планування ресурсів підприємства в сімействі продуктів Microsoft Dynamics. Вона має особливі переваги в виробництві і розподілі. Більш того, це функціонально широка система, здатна обслуговувати покупців в більшості галузей промисловості. Багатомовні і мультивалютні можливості роблять програму популярним варіантом для компаній, які прагнуть до централізації операцій в декількох місцях по всьому світу [1].

Ключові переваги Microsoft Dynamics AX 2009 надає розширений набір інструментів, дозволяючи співробітникам продуктивно працювати з інформацією і будувати більш ефективну взаємодію з клієнтами і партнерами, враховуючи постійно змінюючі потреби ринку. Ви зможете гнучко налаштувати бізнес-процеси і документообіг, розширюючи можливості управління організацією та спрощуючи підтримку корпоративних, законодавчих і клієнтських вимог.

Технології розподіленого планування, реалізовані в системі, дозволяють підтримувати складні холдингові структури, гарантуючи як ефективне співробітництво підприємств холдингу, так і прозору взаємодію з партнерами, клієнтами і постачальниками. Нові технологічні можливості,

призначені для розподілених структур, дозволяють планувати і розбирати ділячи матеріали і ресурси серед підрозділів холдингу незалежно від числа фізичних інсталяцій системи. Більш того, тісний контакт підрозділів дозволяє надавати точну інформацію про виконання замовлень з урахуванням готовності виробу до певного терміну (CTP - capable to promise) і доступності виробу до певного терміну (ATP - available to promise) [2].

Microsoft Dynamics AX може працювати на широкий спектр ринків, але має особливі можливості в сфері виробництва і розподілу. Галузева функціональність цього додатка задовольняє потреби окремих виробників в області високих технологій, електроніки, металів, промислового обладнання, аерокосмічного виробництва, споживчих товарів, медичних приладів та іншої великої промисловості. Програма підтримує повторювані ситуації, а також прогнозування з найрізноманітніших питань. Саме з цієї причини її широко використовують для регулювання виробничих процесів, що вимагають автоматизації партій, формул і рецептів. В результаті додаток дуже часто можна зустріти в компаніях, що займаються виробництвом фармацевтичної, хімічної та харчової продукції.

Іншими словами, відповіддю на питання: «Ахapta - що це за програма?» Служить її визначення як настроюється і гнучкої системи, здатної задовольнити потреби практично будь-якого типу виробництва.

Microsoft Dynamics AX підходить компаніям, розміри і функціональні вимоги яких можуть виправдати його вартість. Як правило, можна побачити, що додаток використовується організаціями, що мають не менше 50 мільйонів доларів річного доходу (хоча іноді трохи менше, коли мова йде про нестандартні сценарії) [1].

1 ОГЛЯД АНАЛОГІВ Й АНАЛІЗ ІНСТРУМЕНТІВ ДЛЯ РОЗРОБКИ СИСТЕМИ «АНАЛІЗ ЗАВОДСЬКОГО ЗАМОВЛЕННЯ»

1.1 Аналіз предметної області

З кожним роком ринок ERP-систем в Україні все більше розростається. Розробники пропонують різні рішення для автоматизації поточних бізнес-процесів і господарської діяльності підприємств. Такі програмні продукти помітно полегшують керування компанією і дозволяють вивести її на новий рівень.

На ринку представлені як галузеві рішення, так і універсальні ERP-системи для України. У міру зростання бізнесу їх можна масштабувати, удосконалювати і доповнювати з урахуванням потреб підприємства.

Впровадження ERP-систем сприяє раціональному зниженню витрат і розподілу ресурсів, а разом з тим - полегшує процес управління [3].

1.2 Аналогічні сучасні системи, що використовуються для розробки функціоналу

1.2.1 Критерії вибору системи

При виборі платформи ERP, на думку автора, слід брати до уваги такі критерії:

- рентабельність впровадження і планова сукупна вартість володіння (в неї входить не тільки вартість ліцензій, але також вартість супроводу (оновлень), вартість послуг інтегратора, послуг з доопрацювання, якщо така проводиться, вартість обладнання, вартість навчання співробітників та оплата фахівців, які будуть залучатися на етапі експлуатації);

- плановий термін впровадження і наявність готових рішень необхідної галузевої спрямованості (в разі їх відсутності для обраної

платформи, розробка нестандартного рішення призведе до збільшення і вартості і строків);

- сумісність впроваджуваного рішення з бізнес-процесами компанії (Можуть мати місце об'єктивні перешкоди адаптації конкретної ERP-системи до процесів конкретної компанії, або можуть бути присутніми додаткові функціональні вимоги. Наприклад, ряд зарубіжних аудиторів вимагає наявності спеціалізованого функціоналу в ERP. Це може бути важливо для компаній за участю іноземного капіталу) [4].

1.2.2 Огляд та порівняння найпоширеніх ERP-систем

У топ найбільш затребуваних продуктів для автоматизації бізнесу незмінно потрапляють:

- BAS ERP;
- SAP;
- Галактика ERP;
- SyteLine;
- Парус ERP;
- Oracle E-Business Suit;
- Microsoft Dynamics AX;
- DeloPro.

SAP

Попитом на українському ринку користується і ERP SAP. Продукт призначений для автоматизації ряду бізнес-процесів. У програмі передбачені модулі для:

- управління виробництвом;
- розподілу ресурсів;
- фінансового обліку;
- аналізу роботи підприємства та планування;
- управління персоналом;

- оптимізації взаємодії з постачальниками, партнерами, клієнтами.

Ще одна причина впровадити таку ІТ-систему - наявність власного хмарного сервісу SAP Business ByDesign. У ньому реалізовані модулі для управління фінансами, продажами, закупками і роботою персоналу.

Галактика

Розроблена однойменною компанією програма «Галактика» ще не вийшла на світовий ринок ERP. Впровадження цього продукту поки можливо тільки в Україні. Рішення заточене під українські підприємства і місцеве законодавство. Його головні переваги:

- багатофункціональність;
- можливість інтеграції з будь-якими програмними продуктами інших розробників;
- висока продуктивність платформи;
- можливість одночасної роботи великої кількості користувачів;
- зручність адміністрування системи;
- надійний захист персональних даних і всієї інформації, що надходить.

Причому розробники готові запропонувати українським підприємцям не тільки стандартну конфігурацію, але і ряд вузькоспеціалізованих галузевих рішень.

Oracle E-Business Suit

Цей створений іноземними програмістами продукт локалізована для України. Він являє собою повнофункціональний набір додатків для ефективного управління бізнесом.

В системі реалізовані модулі для ефективного управління:

- виробництвом продукції та експлуатацією обладнання;
- постачанням, збутом і роботою складів;
- доставкою замовлень;
- взаємодією з клієнтами;

- роботою персоналу (найм, виплата зарплат, організація навчання та інше);
- фінансами, прийомом платежів і розширенням матеріальних потоків.

Також в програмі передбачені модулі для управління життєвим циклом продукції. Є і підсистеми для аналізу роботи підприємства, податкового і фінансового обліку.

Microsoft Dynamics AX - продукт являє собою поєднання кількох підсистем, робота яких охоплює всі сфери управління компанією. В тому числі, виробництво товару, дистрибуцію, фінанси, ланцюжки поставок, обробку замовлень і роботу з клієнтами.

Як і інші продукти MS, система ретельно продумана. Її головними перевагами є:

- багатий функціонал;
- звичний інтуїтивно зрозумілий інтерфейс;
- масштабованість програми під потреби будь-якого бізнесу;
- можливість вбудовувати в систему інші продукти Microsoft (наприклад, Visual Studio .NET або OutLook);
- підтримка поширених бізнес-процесів і можливість швидко перебудовувати систему (за рахунок шарової архітектури коду);
- автоматичне резервне копіювання системи;
- розмежування прав доступу;
- простота взаємодії з клієнтами, партнерами, постачальниками.

Microsoft Dynamics AX - уніфікована система, локалізована під український ринок. Продукт підійде для держустанов, виробничих підприємств, фінансових організацій і компаній, що надають різні послуги. Використовувати програму можна і на підприємствах, що працюють в індустрії моди, будівництві або, наприклад, рекламу.

DeloPro

На ринку ЕРП представлені не тільки іноземні товари, а й ефективні рішення від українських розробників. Одне з них - програма DeloPro. Це гнучкий багатофункціональний хмарний сервіс для управління підприємством, реалізований на web-платформі.

Головними перевагами впровадження ERP DeloPro є:

- можливість використання безкоштовного програмного забезпечення;
- незалежність від базових операційних систем (може працювати як на MS Windows, так і на Linux);
- мінімальні вимоги до IT-інфраструктури підприємства;
- можливість підключення необмеженого числа користувачів;
- зручне розмежування прав доступу;
- простота масштабування і можливість удосконалити систему в міру необхідності;
- наявність інтегрованих CRM та інших підсистем для управління бізнес-процесами;
- у ERP передбачена робота з персоналом, управління поставками, електронна комерція та інше.

Система орієнтована на стартапи та компанії, що займаються оптовою та роздрібною торгівлею, наданням послуг, виробництвом.

BAS ERP

Незважаючи на таке розмаїття, одним з найбільш затребуваних на ринку рішень залишається BAS ERP.

Це багатопрофільний продукт, використовувати який можна як для оптимізації роботи великих підприємств, так і для управління дрібним бізнесом. До його головних переваг відносять:

- гнучкість і масштабованість системи;
- широкі функціональні можливості (від моніторингу показників роботи підприємства до управління закупівлями і бюджетування);

- наявність безлічі галузових рішень;
- можливість роботи через інтернет (підтримуються як «хмарні» сервіси, так і мобільні пристрої);
- доступну вартість впровадження.

BAS і ERP-системи, приклади яких наведені вище, дозволяють оптимізувати протікають на підприємстві процеси і ефективно розподілити наявні ресурси. Використовувати такі продукти можна практично для будь-якого бізнесу [3].

Основні ERP-системи та їх порівняння наведено нижче, у таблиці 1.1.

Таблиця 1.1 – Основні ERP-системи [5]

Рішення	Виробник	Сфера застосування	Термін впровадження	Вартість впровадження
SAP R/3	SAP AG (Германия)	Оборонні підприємства, компанії нафтогазового комплексу, металургія, енергетика телекомунікації, банківський сектор.	1-5 років і більше	Ліцензія на 50 робочих місць коштує близько \$ 350 тис. Вартість впровадження може в кілька разів перевищувати вартість рішення.
Oracle Applications	Oracle (США)	Важка промисловість (переважно металургія), телекомунікаційні компанії, фінансовий сектор, хімічна промисловість.	1-5 років і більше	Вартість рішення на одне робоче місце становить близько \$ 5 тис. Повна вартість істотно залежить від необхідної функціональності

Продовження таблиці 1.1

Рішення	Виробник	Сфера застосування	Термін впровадження	Вартість впровадження
Baan ERP	SSA Global (США)	Автомобілебудування, хімічна промисловість, фармацевтика, харчова промисловість.	6 міс-1,5 року і більше	Вартість одного робочого місця - \$ 3 тис. Співвідношення ціни рішення і витрат на впровадження 1: 1-1: 3.
iRenaissance	Ross Systems (США)	Харчова промисловість, хімічні компанії, металургійна промисловість нафтопереробні, целюлозно-паперові, фармацевтичні підприємства.	4 міс. - 1,3 року і більше	Вартість впровадження в середньому \$ 200 тис.
MBS Axapta, Navision	Microsoft (США)	Підприємства нафтової галузі, харчової промисловості, торгові компанії, металургія, дистрибуція, телекомунікаційна галузь.	6 міс. - 2 роки і більше	В середньому вартість рішення на одне робоче місце - \$ 3,5 тис. Вартість впровадження становить 100-250% вартості рішення.
iScala	Epicor (США)	Машинобудування, телекомунікаційна галузь, харчова промисловість.	3 міс. - 1,5 року і більше	Середня вартість iScala 2.1 становить \$ 2-5 тис. За одне робоче місце.

Продовження таблиці 1.1

Рішення	Виробник	Сфера застосування	Термін впровадження	Вартість впровадження
MFG/PRO	QAD (США)	Автомобільна, авіаційна, електронна, електротехнічна, хімічна, фармацевтична і харчова промисловість.	3 міс. - 1,5 року і більше	Вартість ліцензії на одне робоче місце \$ 2-5 тис. В залежності від конфігурації. Впровадження обходиться в 100-200% цієї суми.
J.D.Edwards OneWorld	J.D.Edwards (США)	Гірничодобувна промисловість, будівельні організації, торгові компанії, нафтогазовий сектор.	7 міс. - 1,5 року і більше	Вартість робочого місця OneWorld варіюється від \$ 400 до \$ 4000.
SyteLine ERP	MAPICS (США)	Виробники вимірювального і електрообладнання, деревообробка, поліграфія, машинобудування.	6 - 9 місяців і більше	Вартість ліцензії на одне робоче місце \$ 2-4 тис. Приблизно в стільки ж обійдеться впровадження.

1.3 Основні теоретичні положення розробки системи

Ахарта - це ERP-система, що працює в середовищі електронного бізнесу. Унікальність системи Ахарта полягає в тому, що її сучасна технологія забезпечує єдиний інформаційний простір підприємства, в якому

бек-офіс і фронт-офіс працюють як єдине ціле. Ахарта пропонує набір можливостей для бізнес-аналізу, що удосконалює процес прийняття рішень і комплексне управління відносинами з клієнтами (CRM).

Основними модулями системи Ахарта є фінанси; торгівля і логістика; виробництво; електронна комерція; управління персоналом; проекти; управління відносинами з клієнтами; управління знаннями; управління логістичними ланцюжками.

Великий набір функціональних можливостей системи Ахарта дозволяє одержати низку певних переваг: менші витрати на створення і підтримку системи; легкість в оновленні застосунків; баланс надмірної інформації; повна інтеграція бізнес-процесів [6].

Три найголовніші переваги Microsoft Dynamics AX:

а) потужність:

1) багатий функціонал, включаючи передналаштованні галузеві та горизонтальні рішення в базовому функціоналі;

2) широкі можливості масштабування, за рахунок реалізації множин законодавчих та управлінських вимог в одному рішенні;

3) ролеорієнтований інтерфейс і швидке вбудовування різноманітних продуктів Microsoft в бізнес-процеси компанії;

б) гнучкість:

1) підтримка «природних» бізнес-процесів, які допомагають швидше і наочніше управляти бізнес-процесами компанії;

2) безліч різноманітних варіантів розгортання системи і швидке нарощування продуктивності;

3) шарова архітектура коду, що дозволяє швидко і гнучко змінювати бізнес-процеси;

в) зручність:

1) надання різноманітних варіантів доступу до даних ERP;

2) об'єднання співробітників, клієнтів, постачальників і партнерів компанії в рамках єдиного інформаційного простору;

3) легкість підтримки і супроводу системи на всіх життєвих циклах ERP [7].

На додаток до основної звітності, Microsoft Dynamics AX забезпечує повну функціональність для управління запасами, керівництва складами, транспортом, планування ланцюжків попиту і пропозиції, прогнозування потреби в матеріалах, управління виробництвом, контролю якості, управління життєвим циклом продукту, активами і бізнес-аналітикою. В даний час додаток повністю спирається на технології .Net від Microsoft. Воно частіше встановлюється безпосередньо, але може бути розміщено і віддалено (це зручно у випадках, коли клієнт віддає перевагу системі, безперебійно доступну через Інтернет) [1].

Microsoft Dynamics AX 2009 допомагає співробітникам підвищити ефективність роботи завдяки рольового інтерфейсу. Все, що Вам необхідно для роботи, знаходиться в одному вікні, включаючи збалансовані показники, план роботи, звітність, засновану на гнучких інструментах Microsoft SQL Server Reporting Services, і багато іншого.

Гнучка сервісно-орієнтована архітектура (SOA), легкість обміну документами і підтримка веб-сервісів дозволяють швидко і гнучко адаптувати Microsoft Dynamics AX до нових правил ведення бізнесу і до умов розширення інформаційної взаємодії з партнерами і клієнтами. За допомогою Інтуїтивно зрозумілих засобів бізнес-аналізу можна побачити те, що зазвичай вислизає від погляду. Це дозволяє вчасно помітити збої в роботі компанії [2].

1.4 Архітектура системи

В основі архітектури системи Ахартa знаходиться об'єктно-орієнтована система розробки, спочатку розроблена як базис для швидкої реалізації ділових додатків. Таким чином – здобуваючи цю систему з включеною ліцензією на розробку і доступ до вихідних текстів, ви отримуєте

можливість як завгодно адаптувати її для потреб вашої фірми. Фактично при такому варіанті покупки покупець отримує середовище розробки і виконання програм автоматизації виробництва і написану в цьому середовищі систему автоматизації типових бізнес-процесів підприємства. Можна ставитися до цієї системи як до даності і обмежитися розробкою модулів автоматизації бізнес-процесів, специфічних для даного підприємства. Однак можна ставитися до цієї системи і лише як до деякого прототипу, в якому можна модифікувати будь-які модулі, перекриваючи її архітектуру і підходи до автоматизації типових виробничих процесів під прийняті в автоматизованому підприємстві способи ведення справ. Звичайно, другий варіант підходу до впровадження Ахарта на підприємстві віддає деяким програмістським екстремізмом. Наявні в цій системі модулі автоматизації виробництва являють собою реалізацію кращих сформованих управлінських практик; проте - гнучкість цієї системи настільки висока, що до неї можна ставитися не тільки як до ERP-системі, а й як до середовища швидкої розробки додатків.

У Ахарта вся робота програміста виконується в середовищі MorphX, що входить до складу даного продукту. Morphx є візуальне середовище розробки об'єктів, по інтерфейсу і підходам до кодування нагадує, мабуть, ряд систем Visual Age фірми IBM. Слід відразу зазначити, що вихідні тексти об'єктів зберігаються в двійковому файлі упереміж з різноманітною службовою інформацією і р-кодом, в який вони компілюються. Відповідно, розробляти нові об'єкти в режимі "командного рядка" в принципі неможливо. Середовище розробки надає став вже типовим для аналогічних середовищ набір засобів для редагування вихідних текстів, методів налагодження програм, візуального конструювання форм і звітів. Існують рудиментарні засоби підтримки групової розробки - допускається блокування / розблокування об'єктів з метою запобігання їх паралельної модифікації декількома розробниками. Вбудованого засобу управління версіями в Ахарта, на жаль, немає, а використання зовнішніх продуктів (таких як Microsoft

Visual SourceSafe або Rational Clear Case) ускладнено в зв'язку з тим, що Ахарта не зберігає вихідних текстів об'єктів в окремих текстових файлах.

Для розробки додатків в Ахарта використовується мова програмування X ++. Хоча його прийнято називати Java-подібним, на мій погляд, він досить сильно відрізняється від Java за прийнятими в ньому підходам до розробки. У ньому немає, наприклад, таких фундаментальних для Java понять, як інтерфейс, реалізація інтерфейсу, пакет, абстрактний метод або клас. Тому, хоча синтаксис виразів і набір операторів мови X ++ нагадує Java або C ++, по підходу до розробки він, на мій погляд, ближче до Object Pascal. Набір типів в мові X ++ також відрізняється своєю орієнтацією на розробку ділових додатків. Наприклад, тип `real` зберігається не як традиційне число з плаваючою точкою (IEEE Double), а як Binary Coded Decimal, що дозволяє уникнути типових для формату зберігання IEEE Double помилок в точній поданні дробових чисел, так що ускладнюють життя тих розробників, які намагаються реалізувати грошову арифметику за допомогою мови C ++ або Java. Для обробки рядків в мові X ++ використовується примітивний тип, а не клас, як в мовах C ++ або Java. При порівнянні рядків регістр букв не відрізняється, що також типово для ділових додатків.

Для розробки додатків в Ахарта використовуються різноманітні елементи дизайну системи - класи, таблиці, форми, звіти і деякі інші.

Базовим модулем мови X ++ є клас. Підтримується одиночне спадкоємство, специфікатор області видимості методів, вкладені класи. На відміну від C ++ і Java доступ до змінних класу дозволений тільки з методів класу. Підтримуються статичні методи класу. Нарешті, дана мова програмування має специфічні для Ахарта специфікатор методу `server` і `client`, що дозволяють вказати, де даний метод повинен виконуватися при використанні триланкової конфігурації. Оскільки X ++ призначений для розробки ділових додатків, він безпосередньо підтримує включення в текст програми операторів мови SQL. Ахарта підтримує свій власний діалект мови

SQL, оператори якого транслюються в діалект SQL, що використовується цільовою базою даних - MS SQL або Oracle.

При проектуванні Ахарта однією з основних цілей була необхідність домогтися підтримки повної незалежності ділових додатків від використовуваної бази даних. У зв'язку з цим значна кількість функцій, виконуваних в звичайних системах клієнт-сервер сервером баз даних, виповнюється ядром Ахарта. У зв'язку з цим одним з базисних будівельних елементів програми в даній системі є таблиця. Розробник в середовищі розробки MorphX створює набір визначень таблиць. При цьому таблиця може мати власні методи, як звичайні так і статичні, що дозволяє розглядати таблицю як специфічний випадок класу. Частина методів є визначеним (наприклад методи update, delete, insert), що дозволяє використовувати їх як деякий аналог тригерів.

Крім того, визначення таблиці включає в себе визначення полів, індексів, обмежень посилальної цілісності. Після закінчення створення або модифікації визначення таблиці розробник може виконати операцію синхронізації, при якій в цільовій базі даних створюється або модифікується таблиця, відповідна даному опису. При цьому ні тригери, ні засоби підтримки посилальної цілісності цільової бази даних не використовуються - всі ці функції виконуються в Ахарті. Таким чином, розробник повністю звільняється від необхідності вникати в тонкощі роботи цільової бази даних.

Кожна таблиця в обов'язковому порядку містить поля, що містять унікальний числовий ідентифікатор цього запису, який присвоюється системою, дату і автора, який створив і модифікував даний запис. Нарешті, для кожного запису вказується компанія, до якої вона відноситься. (Ахарт влаштована таким чином, що можливе зберігання даних декількох незалежних компаній в одній базі даних).

Підтримується також поняття Extended Type - деякий аналог поняття домен реляційних баз даних. Extended Type створюється на основі примітивного типу або іншого Extended Type (але не класу). Всі поля

таблиць, що належать до даного Extended Type, успадковують характеристики зберігання та подання, визначені для нього. Крім того, в програмах можна створювати змінні, пов'язані з вашим розширеним типом. Крім того, розробник може створити enum-типи, які можуть використовуватися і для визначення типів полів. При цьому для кожного елемента перерахування визначається числове значення, ідентифікатор, який використовується в коді X ++, і, нарешті, словесний опис якого буде бачити користувач при редагуванні цього поля.

Для організації діалогу з користувачем Ахapta використовує поняття форми. Підхід до створення і організації з формами аналогічний підходу, використовуваному в системах Visual Basic або Delphi. Кожна форма може містити в собі один або більше джерел даних (таблиць), як незалежних один від одного, так і пов'язаних (join) в запит. Велика частина екранних кнопок може бути прив'язана до джерела даних і відображати вміст його поля, або значення, що повертається спеціальним методом, створеним для цього джерела даних. Можливо і чисто програмне керування вмістом екранної кнопки. Набір різноманітних екранних елементів в Ахapta досить типовий для систем подібного класу - він кілька бідніших ніж, наприклад, в Delphi або Visual Basic і багатше, ніж в PowerBuilder або Java AWT. Допускається підключення до Ахapta зовнішніх екранних кнопок, написаних за стандартом ActiveX. Для обробки подій в Ахapta необхідно створити метод плитки, що реагує на цю подію. При цьому форма або джерело даних форми може крім методів обробки подій мати абсолютно довільний набір методів, розроблених програмістом. Єдине обмеження, що діє відносно цих методів, - вони можуть викликатися тільки з інших методів даної форми. Для кожної таблиці в системі Ахapta можна задати форму, яка буде за замовчуванням використовуватися для редагування цієї таблиці. Завдяки цьому значно спрощуються механізми роботи з довідниками - при редагуванні поля, що є посиланням на довідник, користувач може легко викликати на екран форму, яка була вказана при створенні таблиці довідника.

Конструктор звітів в Ахарта аналогічний по набору можливостей аналогічним конструкторам в Delphi або Visal Basic.

Для організації доступу користувача до різної функціональності використовується механізм MenuItem. При створенні будь-якої плитки, що запускає деякі дії - екранної кнопки, пункту меню і так далі, з ним можна асоціювати певний елемент дизайну типу MenuItem. Створення та налагодження MenuItem проводиться глобально для всієї програми, але не залежить від конкретної форми або меню. При цьому для даного елемента MenuItem можна вказати конкретну форму, клас або звіт, які запускаються при його активації. Таким чином, досягається додатковий рівень настройки реакції системи на дії користувача.

Нарешті - слід розповісти про такий цікавий механізм розмежування повноважень, як FeatureKey. При розробці програми можна з кожним елементом дизайну системи - формою, звітом, MenuItem, екранним елементом зв'язати FeatureKey, який дозволяє або забороняє доступ до цього будівельного блоку. Далі при установці системи у кінцевого користувача можна для конкретної групи користувачів дозволити або заборонити доступ до елементів дизайну, асоційованим з даними FeatureKey.

Взаємодія системи Ахарта із зовнішніми програмами здійснюється за допомогою механізму COM / DCOM. З програм, розроблених на мові X ++, можна створювати будь-які COM-об'єкти, що підтримують механізм ActiveX Automation. Існує і зворотний механізм, що дозволяє працювати з внутрішніми об'єктами системи Ахарта через механізм ActiveX Automation. Правда, необхідно домовитися про те, що хоча цей механізм і дозволяє виконувати будь-які операції з цими об'єктами, але він кілька громіздкий; операції, які вимагають декількох рядків коду на X ++ при їх виклик через COM, скажімо, з C ++, можуть зажадати написання пари екранів коду. Тому я б рекомендував використовувати цей інтерфейс тільки для інтеграції існуючих систем. Спроба реалізувати з його допомогою систему, в якій Ахарта використовувалася б лише для реалізації рівня бізнес - об'єктів, а для

організації інтерфейсу використовувався б, скажімо C ++ або Delphi, була б протиприродною. Організація доступу до Ахарта через COM вимагає ліцензування додаткового компонента Ахарта COM Connector.

Особливо хотілося б згадати про механізми інтеграції Ахарта з Microsoft Internet Information Server. У першому наближенні цей механізм виглядає так: на додаток до стандартних елементів дизайну Ахарта має їх, так би мовити, web-еквівалент - web-форми, web-звіти і так далі. Від звичайних елементів дизайну вони відрізняються здатністю представляти свій вміст у вигляді HTML. Звичайно, при розробці подібних елементів розробник стикається з деякими обмеженнями в тих виразних засобах, які він може використовувати. Зрозуміло, що не всі стандартні екранні елементи можуть бути представлені в браузері користувача. Однак цей механізм дозволяє швидко розробити більш-менш пристойний Web-інтерфейс до ERP-системі замовника, не вимагаючи при цьому значного часу на вивчення всіх аспектів Web-програмування. Слід згадати, що підтримка зв'язку з MS IIS ліцензується окремо.

Ахарта спочатку проектувалася з метою полегшення розробки програм, які потребують триланкової архітектури. Для більшої частини елементів дизайну (крім власне елементів інтерфейсу користувача) можна вказати, де вони повинні виконуватися - на робочому місці користувача або на проміжному сервері додатків (Ахарта Object Server). В результаті при використанні триланкової архітектури можна використовувати дешевші комп'ютери для організації робочих місць користувачів (оскільки основне навантаження лягає на сервер додатків). Крім того, при використанні Ахарта Object Server можливе підключення робочих місць користувачів по досить повільним каналам - наприклад, по модемного з'єднання. Це дозволяє заощадити значні гроші на організації підключення віддалених робочих місць користувача [8-9].

1.5 Висновок за розділом 1

У ході розроблення програмного функціоналу для аналізу списань товарно-матеріальних цінностей, було обрано середовищем розробки ERP-систему Microsoft Dynamics AX. Адже дана система забезпечує повну функціональність для управління запасами, керівництва складами, транспортом, планування ланцюжків попиту і пропозиції, прогнозування потреби в матеріалах, управління виробництвом, контролю якості, управління життєвим циклом продукту, активами і бізнес-аналітикою. Microsoft Dynamics AX допомагає співробітникам підвищити ефективність роботи завдяки рольового інтерфейсу. Все, що Вам необхідно для роботи, знаходиться в одному вікні, включаючи збалансовані показники, план роботи та звітність.

Гнучка сервісно-орієнтована архітектура (SOA), легкість обміну документами і підтримка вебсервісів дозволяють швидко і гнучко адаптувати Microsoft Dynamics AX до нових правил ведення бізнесу і до умов розширення інформаційної взаємодії з партнерами і клієнтами.

2 РОЗРОБКА ПРОГРАМНОЇ ЧАСТИНИ СИСТЕМИ «АНАЛІЗ ЗАВОДСЬКОГО ЗАМОВЛЕННЯ»

2.1 Постановка завдання розробки і дослідження

На даний момент аналіз необхідних для списання під виробничі замовлення ТМЦ виконується вручну інженерами підрозділів директора з виробництва на підставі аналізу планової потреби в матеріалах і КІ по заводському замовленню, по цеху (звіт «Потреба в матеріалах і комплектуючих»), аналізу карт дозволів, аналізу таблиці взаємозамінності, аналізу залишків (звіт «Відомість руху»), аналізу дефіциту. Процес дуже трудомісткий. Тому необхідно оптимізувати витрати робочого часу при проведенні аналізу наявності ТМЦ (з урахуванням КД, карт дозволів, таблиці взаємозамінності, залишків на заводі, і ін.) Для списання під виробничі замовлення в системі MS Dynamics AX. Дана модифікація проводиться для русифікованої версії продукту Microsoft Dynamics AX.

2.2 Загальна структура програми

2.2.1 Опис модифікації. Розробка таблиць

Створити таблицю «Аналіз 3/3-Шапка» (табл. 2.1) (AnalysProdNumTable_ZTR).

Таблиця 2.1 – Аналіз 3/3-Шапка

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	AnalysId	EDT: AnalysId, створити	Обов'язкове, унікальне
Номер пр. замовлення по ЗТР	ProdNum	EDT: ProNum_ZTR	Обов'язкове
Робочий центр	WrkCtrGroupId	EDT: WrkCtrGroupId	Обов'язкове
Спільний вид/вузел	ItemIdBom	EDT: ItemIdBom_ZTR	Обов'язкове
Код схеми	SchemeCode	EDT: BOMSchemeCode_Z TR	Обов'язкове

Продовження таблиці 2.1

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Примітка	Note	Type: String String size: 500	

Створити таблицю «Аналіз 3/3-Рядки» (табл.2.2)
(AnalysProdNumLine_ZTR).

Таблиця 2.2 - Аналіз 3/3-Рядки

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	AnalysId	EDT: AnalysId	Обов'язкове
Тип аналізу 3/3	AnalysType	Enum Type: PotrbType	0 – без правил, 1 – правила застосовані
Код номенклатури	ItemId	EDT: ItemId	Обов'язкове
Кількість	Qty	EDT: BOMQty	
Таблиця взаємозамінності	TableReplacement	Enum Type: NoYes	
Карта розширення	Karta	Enum Type: NoYes	
Списано	SpisQty	EDT: BOMQty	
Замовлено	OrderQty	EDT: BOMQty	
Залишки 3/3, цех	OstZzQty	EDT: BOMQty	
Залишки без 3/3, цех	OstNoZzQty	EDT: BOMQty	
Залишки всі	OstAll	EDT: BOMQty	

Унікальний ключ: Код аналізу 3/3, Тип аналізу 3/3, Код номенклатури.

Створити рядок «Код аналізу 3/3» (рисунк 2.1) в формі «Параметри» (Потреба і Дефіцит - Налаштування - Параметри - закладка Номерні серії) зі стандартним функціоналом даної форми.

Параметры [01@AX-AX-SRV1 SQL-SRV1:AX_ZTR_20...

Номерные серии

Ссылка	Код номерн...
Код дефицита	Def
Код заявки	Z
Код анализа 3/3	A

Рисунок 2.1 – Приклад заповнення форми «Параметри»

Створити таблицю «Аналіз Правила» (табл. 2.3) (AnalysRules_ZTR).

Таблиця 2.3 – Аналіз-Правила

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу З/З	AnalysId	EDT: AnalysId	Обов'язкове
Код посилання на запис	RefRecId	EDT: RefRecId	Обов'язкове
Тип даних	TypeRule	Enum Type: PotrbTypeRule	1 – таблиця взаємозамінності 5 – карта розшир.
Номенклатура	ItemId	EDT: ItemId	
Номенклатура-дублікат	ItemIdDuplicate	EDT: ItemId	
Номер пр. заказа по ЗТР	ProdNum_ZTR	EDT: ProdNum_ZTR	
Кількість	Qty	EDT: Qty	
Кількість-Дублікіт	QtyDuplicate	EDT: Qty	
Цех	WrkCtrGroupId	EDT: WrkCtrGroupId	
Направлення	PotrbRoute	Enum Type: PotrbRoute_ZTR	1 – в обидві сторони 2 – в одну сторону
Коефіцієнт	Koef	Type: Numeric	
Примітка	Note	Type: String String size: 500	
Дата документу	NoteDate	Type: Date	
Автор створення правила	CreatedByRules	CreatedBy	
Дата створення правила	CreatedDateTimeRules	CreatedDateTime	
Направлення	PotrbRoute	Enum Type: PotrbRoute_ZTR	1 – в обидві сторони 2 – в одну сторону
Коефіцієнт	Koef	Type: Numeric	
Примітка	Note	Type: String String size: 500	
Дата документу	NoteDate	Type: Date	
Автор створення правила	CreatedByRules	CreatedBy	
Дата створення правила	CreatedDateTimeRules	CreatedDateTime	

Створити таблицю «Аналіз Детально» (табл. 2.4) (AnalysDetail_ZTR) .

Таблиця 2.4 - Аналіз Детально

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	AnalysId	EDT: AnalysId	Обов'язкове
Код посилання на запис	RefRecId	EDT: RefRecId	Обов'язкове
Тип деталізації	DetailType	Enum Type: DetailType, создать	0 – списано, 1 – замовлено, 2 – залишки 3/3, цех, 3 – залишки без 3/3, цех, 4 – залишки все
Журнал	JournalId	EDT: InventJournalId	
Номер пр. заказа по ЗТР	ProdNum	EDT: ProdNum_ZTR	
Цех	InventLocationId	EDT: InventLocationId	
Код номенклатури	ItemId	EDT: ItemId	
Дата	DateFinancial	EDT: DateFinancial	
Кількість	Qty	EDT: BOMQty	
Партія	InventBatchId	EDT: InventBatchId	
Рівень	Level1	Type: Numeric (целое)	

Нижче буде наведено опис зв'язків таблиць (табл 2.5).

Таблиця 2.5 - Опис зв'язків таблиць

Таблиця 1	Таблиця 2
AnalysProdNumTable_ZTR.AnalysId	AnalysProdNumLine_ZTR.AnalysId
AnalysDetail_ZTR.RefRecId	AnalysProdNumLine_ZTR.RecId
AnalysRules_ZTR.RefRecId	AnalysProdNumLine_ZTR.RecId
InventLocation.WrkCtrGroupId_ZTR	WrkCtrTable.WrkCtrGroupId
AnalysProdNumLine_ZTR.ItemId	InventTable.ItemId
AnalysDetail_ZTR.ItemId	InventTable.ItemId
InventTable.ItemId	InventTableModule.ItemId
InventTrans.InventDimId	InventDim.InventDimId
InventSum.InventDimId	InventDim.InventDimId

2.2.2 Створення форми «Аналіз заводського замовлення»

(Потреба і Дефіцит - Аналіз заводського замовлення)

У верхній частині форми (шапка) - архів створених аналізів і 2 кнопки (рисунок 2.2). («Створити аналіз 3 / 3», «Створити журнал списань»).



Аналіз 3/3	Номер пр. заказа по ЗТР	Рабочий центр	Общий вид/узел	Код схемы	Примечание	Создан	Созданные дата и время
A_000026	050/6880	G003	ААКР.672233.071-03	59	Киях (таблич)дог K006168	KiyOV	25.08.2020 16:03:07
A_000027	050/6880	G005	ААКР.672233.071-03	59		VinAV	26.08.2020 10:37:10
A_000028	050/6880	G004	ААКР.672233.071-03	59		VinAV	26.08.2020 10:39:55
A_000029	050/6880	G002	ААКР.672233.071-03	59	Заявка на крепеж № 34 ПС...	PutNA	27.08.2020 12:20:30
A_000043	050/6880	G003	ААКР.672233.071-03	59	test	test	27.08.2020 13:57:11
A_000045	050/6880	G002	ААКР.672233.071-03	59		GedOS	31.08.2020 10:53:26
A_000048	050/6880	G003	ААКР.672233.071-03	59	Савенкова - 09Г2С (CA+A...	Savlr	31.08.2020 12:22:15
A_000049	050/6880	G007	ААКР.672233.071-03	59	Катанка, лак на Substation...	PutNA	31.08.2020 13:02:48

Рисунок 2.2 – Приклад створення форми «Аналіз заводського замовлення»

Нижче у таблиці (табл 2.6) буде наведено опис полів даної форми.

Таблица 2.6 – Аналіз заводського замовлення

Назва поля	Опис джерела даних (поле таблиці)	Примітка
Аналіз 3/3	AnalysProdNumTable_ZTR. AnalysId	Не редаговані
Номер пр. заказа	AnalysProdNumTable_ZTR. ProdNum	Не редаговані
Робочий центр	AnalysProdNumTable_ZTR. WrkCtrGroupId	Не редаговані
Загальний вид/вузол	AnalysProdNumTable_ZTR.I temIdBom	Не редаговані
Код схеми	AnalysProdNumTable_ZTR. SchemeCode	Не редаговані
Примітка	AnalysProdNumTable_ZTR. Note	Не редаговані
Створений	AnalysProdNumTable_ZTR. CreatedBy	Не редаговані
Створені дата й час	AnalysProdNumTable_ZTR. CreatedDateTime	Не редаговані

Створити екранну кнопку на формі, яка буде наведена нижче на рисунку 2.3, назвати її «Створити аналіз 3 / 3». Відкрити діалогове вікно.

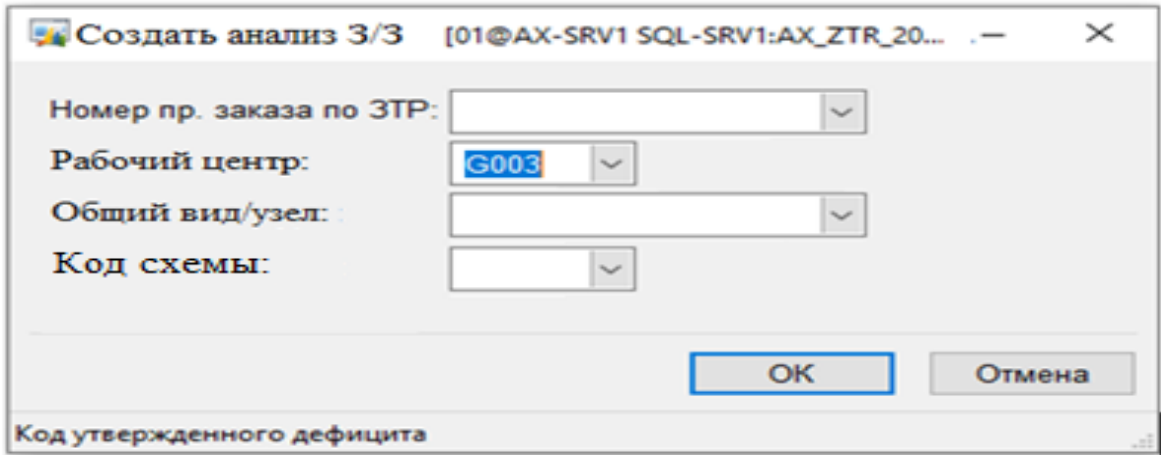


Рисунок 2.3 - Пример створення вікна «Створити аналіз 3 / 3»

В полі «Номер пр. Замовлення по ЗТР» вибрати зі списку номерів пр. Замовлень по ЗТР (таблиця ProdNumTable_ZTR).

В полі «Робочий центр» вибрати зі списку робочих центрів (таблиця WrkCtrTable.WrkCtrGroupId).

В полі «Загальний вид / вузол» вибрати зі списку номенклатур (таблиця InventTable) за умовою: InventTable.ItemType = 1.

В поле «Код схеми» вибрати зі списку схем покриттів (таблиця BOMCovScheme.SchemeCode)

Дані діалогу зберігати для поточного користувача, показувати при наступному виклику звіту.

За кнопкою ОК виконати:

Крок 1

Створити записи, що наведені у таблиці нижче (табл. 2.7) .

Таблиця 2.7 – Записи в таблиці AnalysProdNumTable_ZTR

Назва поля таблиці	Опис джерела даних	Примітка
AnalysProdNumLine_ZTR.A nalyzId	-	з номерний серії
AnalysProdNumLine_ZTR.Pr odNum	-	поле «Номер пр. Замовлення по ЗТР» діалогу

Продовження таблиці 2.7

Назва поля таблиці	Опис джерела даних	Примітка
AnalysProdNumLine_ZTR.WrkCtrGroupId	-	поле «Робочий центр» діалогу
AnalysProdNumLine_ZTR.ItemIdBom	-	поле «Загальний вид / вузол» діалогу

Крок 2

Виконати розрахунок потреби, див. Звіт «Потреба в матеріалах і комплектуючих» (Звіти ЗТР - Звіти ПДУ) , рисунок звіту наведений нижче (рисунок 2.4).

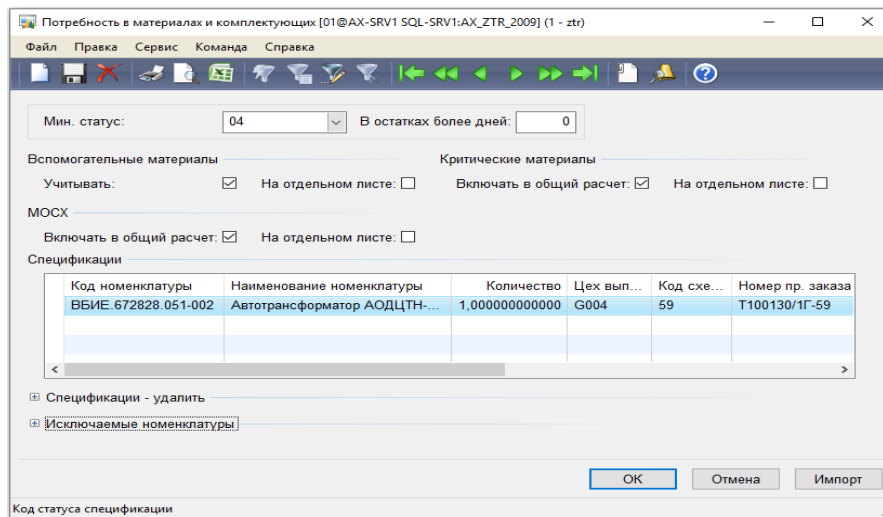


Рисунок 2.4 – Приклад заповнення звіту ЗТР

Опис полів даного звіту наведений нижчу у таблиці 2.8.

Таблиця 2.8 – Звіти ЗТР

Назва поля	Опис джерела даних	Примітка
Код номенклатури	-	поле «Загальний вид/вузол» діалогу
Кількість	-	1
Цех випуску	-	G004 (або G088, якщо поле «Загальний вид/вузол» діалогу = G088)
Код схеми	-	поле «Код схеми» діалогу
Номер вироб. замовлення ЗТР	-	поле «Номер пр. заказа по ЗТР» діалогу

Крок 3

Записи, отримані після виконання кроку 2 відфільтрувати за умовою:

WrkCtrGroupId (Цех випуску) звіту «Потреба в матеріалах і комплектуючих»= Поле «Робочий центр» діалогу.

Крок 4

Записи, отримані після виконання кроку 3, згрупувати по полю ItemId (Код номенклатури, див. Звіт «Потреба в матеріалах і комплектуючих») і підсумувати поле Qty (кількість, див. Звіт «Потреба в матеріалах і комплектуючих»).

Крок 5

Створити в таблиці AnalysProdNumLine_ZTR («Аналіз 3 / 3 - Рядки») записи, які наведені у таблиці 2.9.

Таблиця 2.9 - Аналіз 3 / 3 - Рядки

Назва поля	Опис джерела даних	Примітка
AnalysId	AnalysProdNumTable_ZTR.A nalysId	
AnalysType	0 – без правил	
ItemId	Код номенклатури	Див. Крок4
Qty	Кількість	Див. Крок4

Крок 6

Створити в таблиці AnalysProdNumLine_ZTR записи, застосувавши таблицю ПРАВИЛА (PotrbRules_ZTR), заповнивши AnalysProdNumLine_ZTR.AnalysType = «1 - правила застосовані»:

PotrbRules_ZTR.TypeRule = 5 (Карта розрішень)

Створити записи - копії вихідних, виконавши:

а) якщо PotrbRules_ZTR.Koef $\neq$ 1, створити 1 запис - копію вихідної, виконавши:

1) AnalysProdNumLine_ZTR.Qty=AnalysProdNumLine_ZTR.Qty * PotrbRules_ZTR.Koef;

2) `AnalysProdNumLine_ZTR.ItemId=PotrbRules_ZTR.ItemIdDu`

plicate;

б) якщо `PotrbRules_ZTR.Koef=1`, створити 2 записи:

1) Перший запис - копія вихідної, виконавши:

- `AnalysProdNumLine_ZTR.Qty=AnalysProdNumLine_ZTR.Qty-PotrbRules_ZTR.Qty` запис не створювати, якщо `AnalysProdNumLine_ZTR.Qty-PotrbRules_ZTR.Qty <= 0 !!!`;

2) Другий запис – копія вихідної, виконавши:

- `AnalysProdNumLine_ZTR.Qty=PotrbRules_ZTR.QtyDuplicate(AnalysProdNumLine_ZTR.ItemId=PotrbRules_ZTR.ItemIdDuplicate)`.

Джерело даних-таблиці `AnalysProdNumLine_ZTR`, `PotrbRules_ZTR` за умовами:

а) `AnalysProdNumLine_ZTR.Analys=AnalysProdNumTable_ZTR.AnalysId` (поточний) ;

б) `AnalysProdNumLine_ZTR.Type = 0` – без правил;

в) `PotrbRules_ZTR.ItemId = AnalysProdNumLine_ZTR.ItemId`;

г) `PotrbRules_ZTR.Actual = «так»`;

д) `PotrbRules_ZTR.ProdNum_ZTR=AnalysProdNumTable_ZTR.ProdNum_ZTR`;

е) `PotrbRules_ZTR.WrkCtrGroupId=AnalysProdNumTable_ZTR.WrkCtrGroupId`;

ж) `PotrbRules_ZTR.ItemId != «»`.

Створити записи в таблиці `AnalysProdNumLine_ZTR` (нові записи), виконавши:

а) `AnalysProdNumLine_ZTR.ItemId=PotrbRules_ZTR.ItemIdDuplicate`;

б) `AnalysProdNumLine_ZTR.Qty=PotrbRules_ZTR.QtyDuplicate`;

в) `AnalysProdNumLine_ZTR.PotrbType=«1–правила застосовані»`.

Джерело даних-таблиці `AnalysProdNumLine_ZTR`, `PotrbRules_ZTR` за умовами:

а) PotrbRules_ZTR.Actual = «так»;
 б) PotrbRules_ZTR.ProdNum_ZTR=AnalysProdNumTable_ZTR.ProdNum_ZTR;

в) PotrbRules_ZTR.ItemId = «».

PotrbRules_ZTR.TypeRule = 1 (Таблиця взаємозамінності)

Створити запис - копію вихідної, виконавши:

а) AnalysProdNumLine_ZTR.ItemId=PotrbRules_ZTR.ItemIdDuplicate.

Джерело даних-таблиці AnalysProdNumLine_ZTR, PotrbRules_ZTR за умовами:

а) PotrbRules_ZTR.ItemId = AnalysProdNumLine_ZTR.ItemId;

б) PotrbRules_ZTR.Actual = «так».

Якщо записи таблиці AnalysProdNumLine_ZTR немає ні в одному правилі, створити дублікат записи.

Для створених записів, які відповідають умові:

а) AnalysProdNumLine_ZTR.PotrbType = 1;

б) InventTable.IsCooperation_ZTR = 0;

в) InventTable.ItemType = 1 (номенклатури-специфікації).

Виконати розрахунок потреби, описаний у таблиці нижче (табл. 2.10) (розкрити номенклатури з типом Специфікація), див. звіт «Потреба в матеріалах і комплектуючих» (Звіти ЗТР - Звіти ПДУ).

Таблиця 2.10 – Опис полів для розрахунку потреби

Назва поля	Опис джерела даних	Примітка
Код номенклатури		AnalysProdNumLine_ZTR.ItemId
Кількість		AnalysProdNumLine_ZTR.Quantity
Цех випуску		G004 (або G088, якщо поле «Робочий центр» діалогу = G088)
Код схеми		поле «Код схеми» діалогу
Номер вироб. замовлення по ЗТР		поле «Номер вироб. замовлення по ЗТР» діалогу
Номер вироб. замовлення по ЗТР		поле «Номер вироб. замовлення по ЗТР» діалогу

У всіх створених записах підсумувати кількість (AnalysProdNumLine_ZTR.Qty), групуючи за:

- Код номенклатури (AnalysProdNumLine_ZTR.ItemId);
- Тип потреби (AnalysProdNumLine_ZTR.AnalysType).

Крок 7

Створити в таблиці AnalysRules_ZTR записи (табл 2.11).

Таблиця 2.11 – Записи AnalysRules_ZTR

Назва поля	Опис джерела даних	Примітка
AnalysId	AnalysProdNumTable_ZTR.AnalysId	
RefRecId	AnalysProdNumLine_ZTR.RecId	
ItemId	PotrbRules_ZTR.ItemId	
TypeRule	PotrbRules_ZTR.TypeRule	1 – таблиця взаємозамінності 5 – карта розрішень
ItemIdDuplicate	PotrbRules_ZTR.ItemIdDuplicate	
ProdNum_ZTR	PotrbRules_ZTR.ProdNum_ZTR	
Qty	PotrbRules_ZTR.Qty	
QtyDuplicate	PotrbRules_ZTR.QtyDuplicate	
WrkCtrGroupId	PotrbRules_ZTR.WrkCtrGroupId	
PotrbRoute	PotrbRules_ZTR.PotrbRoute	
Koef	PotrbRules_ZTR.Koef	
Note	PotrbRules_ZTR.Note	
NoteDate	PotrbRules_ZTR.NoteDate	
CreatedByRules	PotrbRules_ZTR.CreatedBy	
CreatedDateTimeRules	PotrbRules_ZTR.CreatedDateTime	

Джерело даних - таблиця PotrbRules_ZTR, використовувана в Крок 6.

Крок 8.

Створити в таблиці AnalysDetail_ZTR записи, які вказані далі у таблицях.

Створити в таблиці 2.12 записи Списано.

Таблиця 2.12 – А. Списано

Назва поля	Опис джерела даних	Примітка
AnalysId	AnalysProdNumTable_ZTR. AnalysId	
RefRecId	AnalysProdNumLine_ZTR.R ecId	
DetailType	-	0 – списано
JournalId	#tmp_t1.TransRefId	
ItemId	#tmp_t1.ItemId	
InventLoctionId	#tmp_t1.InventLocationId	
DateFinancial	#tmp_t1.DateFinancial	
Qty	#tmp_t1.Qty	
InventBatchId	#tmp_t1.InventBatchId	
Level1	#tmp_t1.Level1	

Джерело даних - таблиці AnalysProdNumLine_ZTR, # tmp_t1 за умовою:

- а) AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId (поточний);
- б) AnalysProdNumLine_ZTR.Type = 1 – правила застосовані;
- в) AnalysDetail_ZTR.InventLocationId in (Select InventLocationId from InventLocation where WrkCtrGroupId_ZTR = AnalysProdNumTable_ZTR.WrkCtrGroupId_ZTR);
- г) #tmp_t1.ItemId = AnalysProdNumLine_ZTR.ItemId;
- д) Додатково врахувати Списано для PotrbRules_ZTR.ItemId за умовою (Взаємозамінність!!!):

- 1) AnalysProdNumLine_ZTR.ItemId=PotrbRules_ZTR.ItemIdDuplicate;
- 2) PotrbRules_ZTR.PotrbRoute = «1 – в обидві сторони»;
- 3) PotrbRules_ZTR.Actual = 1;
- 4) PotrbRules_ZTR.TypeRule= 1 (Таблиця взаємозамінності).

Створити в таблиці 2.13 записи Замовлено.

Таблиця 2.13 – Б. Замовлено

Назва поля	Опис джерела даних	Примітка
AnalysId	AnalysProdNumTable_ZTR. AnalysId	

Продовження таблиці 2.13

Назва поля	Опис джерела даних	Примітка
RefRecId	AnalysProdNumLine_ZTR.RecId	
JournalId	#tmp_t1.TransRefId	
InventLocationId	#tmp_t1.InventLocationId	
ItemId	#tmp_t1.ItemId	
DateFinancial	#tmp_t1.DateFinancial	
Qty	#tmp_t1.Qty	
InventBatchId	#tmp_t1.InventBatchId	
Level1	#tmp_t1.Level1	

Джерело даних – таблиця AnalysProdNumLine_ZTR, #tmp_t1 за умовою:

- а) AnalysProdNumLine_ZTR.AnalysIdя=AnalysProdNumTable_ZTR.AnalysId (поточний);
- б) AnalysProdNumLine_ZTR.Type = 1 – правила застосовані;
- в) AnalysDetail_ZTR.InventLocationId in (Select InventLocationId from InventLocation; where WrkCtrGroupId_ZTR=AnalysProdNumTable_ZTR.WrkCtrGroupId_ZTR);
- г) #tmp_t1.ItemId = AnalysProdNumLine_ZTR.ItemId;
- д) Додатково врахувати Замовлено для PotrbRules_ZTR.ItemId (див. вище Взаємозамінність).

Створити в таблиці 2.14 записи Залишки 3/3.

Таблиця 2.14 – В. Залишки 3/3,цех

Назва поля	Опис джерела даних	Примітка
AnalysId	AnalysProdNumTable_ZTR.AnalysId	
RefRecId	AnalysProdNumLine_ZTR.RecId	
DetailType	-	2 – залишки 3/3, цех
InventLocationId	InventDim.InventLocationId	InventSum.InventDimId = InventDim.InventDimId
ProdNum	InventDim.InventGtdId_RU	InventSum.InventDimId = InventDim.InventDimId
ItemId	InventSum.ItemId	
Qty	$\sum$ InventSum.AvailPhysical	Підсумувати за групуванням

Групування:

- а) `AnalysDetail_ZTR.RefRecId`;
- б) `AnalysDetail_ZTR.ItemId`;
- в) `AnalysDetail_ZTR.InventLocationId`;
- г) `AnalysDetail_ZTR.ProdNum`.

Джерело даних – таблиці `AnalysProdNumLine_ZTR`, `InventSum` за умовою:

- а) `AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId` (поточний) ;
- б) `AnalysProdNumLine_ZTR.Type = 1` – правила застосовані;
- в) `InventSum.ItemId = AnalysProdNumLine_ZTR.ItemId`;
- г) `InventSum.InventDimId = InventDim.InventDimId`;
- д) `AnalysDetail_ZTR.InventLocationId in (Select InventLocationId from InventLocation` where `WrkCtrGroupId_ZTR=AnalysProdNumTable_ZTR.WrkCtrGroupId_ZTR)`;
- е) `InventDim.InventGtdId_RU=AnalysProdNumTable_ZTR.ProdNum`.

Додатково врахувати Залишки 3/3,цех для `PotrbRules_ZTR.ItemId` (див. вище Взаємозамінність).

Створити в таблиці 2.15 записи Залишки без 3/3.

Таблиця 2.15 - Г. Залишки без 3/3,цех

Назва поля	Опис джерела даних	Примітка
<code>AnalysId</code>	<code>AnalysProdNumTable_ZTR.AnalysId</code>	
<code>RefRecId</code>	<code>AnalysProdNumLine_ZTR.RefRecId</code>	
<code>DetailType</code>	-	3 – залишки без 3/3, цех
<code>InventLocationId</code>	<code>InventDim.InventLocationId</code>	<code>InventSum.InventDimId = InventDim.InventDimId</code>
<code>ProdNum</code>	<code>InventDim.InventGtdId_RU</code>	<code>InventSum.InventDimId = InventDim.InventDimId</code>
<code>ItemId</code>	<code>InventSum.ItemId</code>	
<code>Qty</code>	$\sum$ <code>InventSum.AvailPhysical</code>	Підсумувати за групуванням

Групування:

- а) `AnalysDetail_ZTR.RefRecId`;
- б) `AnalysDetail_ZTR.ItemId`;
- в) `AnalysDetail_ZTR.InventLocationId`;
- г) `AnalysDetail_ZTR.ProdNum`.

Джерело даних – таблиці `AnalysProdNumLine_ZTR`, `InventSum` за умовою:

- а) `AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId` (поточний);
- б) `AnalysProdNumLine_ZTR.Type = 1` – правила застосовані;
- в) `InventSum.ItemId = AnalysProdNumLine_ZTR.ItemId`;
- г) `InventSum.InventDimId = InventDim.InventDimId`;
- д) `AnalysDetail_ZTR.InventLocationId in (Select InventLocationId from InventLocation where WrkCtrGroupId_ZTR = AnalysProdNumTable_ZTR.WrkCtrGroupId_ZTR)`;
- е) `InventDim.InventGtdId_RU = «Без з/з»`;
- ж) Додатково врахувати Залишки без 3/3, цех для `Potrbrules_ZTR.ItemId` (див. вище Взаємозамінність).

Створити в таблиці 2.16 записи Залишки всі.

Таблиця 2.16 – Д. Залишки всі

Назва поля	Опис джерела даних	Примітка
<code>AnalysId</code>	<code>AnalysProdNumTable_ZTR.AnalysId</code>	
<code>RefRecId</code>	<code>AnalysProdNumLine_ZTR.RefRecId</code>	
<code>DetailType</code>	-	4 – залишки всі
<code>InventLocationId</code>	<code>InventDim.InventLocationId</code>	<code>InventSum.InventDimId = InventDim.InventDimId</code>
Назва поля	Опис джерела даних	Примітка
<code>ProdNum</code>	<code>InventDim.InventGtdId_RU</code>	<code>InventSum.InventDimId = InventDim.InventDimId</code>
<code>ItemId</code>	<code>InventSum.ItemId</code>	
<code>Qty</code>	$\sum$ <code>InventSum.AvailPhysical</code>	Підсумувати за групуванням

Групування:

- а) `AnalysDetail_ZTR.RefRecId`;
- б) `AnalysDetail_ZTR.ItemId`;
- в) `AnalysDetail_ZTR.InventLocationId`;
- г) `AnalysDetail_ZTR.ProdNum`.

Джерело даних – таблиці `AnalysProdNumLine_ZTR`, `InventSum` за умовою:

- а) `AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId` (поточний);
- б) `AnalysProdNumLine_ZTR.Type = 1` – правила застосовані;
- в) `InventSum.ItemId = AnalysProdNumLine_ZTR.ItemId`;
- г) `InventSum.InventDimId = InventDim.InventDimId`;
- д) `AnalysDetail_ZTR.InventLocationId in (Select InventLocationId from InventLocation where InventLocation.InPotrb = 1)` для `InventDim.InventGtdId_RU = «Без з/з»`;
- е) Додатково врахувати Списано для `PotrbRules_ZTR.ItemId` за умовою (Взаємозамінність).

Крок 9.

Заповнити в таблиці `AnalysProdNumLine_ZTR` поля, дивитись нижче у таблиці 2.17.

Таблиця 2.17 – `AnalysProdNumLine_ZTR`

Назва поля	Опис джерела даних	Примітка
<code>SpisQty</code>	$\sum \text{AnalysDetail_ZTR.Qty}$	<code>AnalysDetail_ZTR.DetailType = 0</code>
<code>TableReplacement</code>		<code>=1</code> , якщо існує запис у таблиці <code>AnalysRules_ZTR</code> за умовою: <code>AnalysRules_ZTR.AnalysId = AnalysProdNumTable_ZTR.AnalysId</code> <code>AnalysRules_ZTR.RefRecId = AnalysProdNumLine_ZTR.RefRecId</code> <code>AnalysRules_ZTR.TypeRule</code>

Продовження таблиці 2.17

Назва поля	Опис джерела даних	Примітка
Karta		=1, якщо існує запис у таблиці AnalysRules_ZTR за умовою: AnalysRules_ZTR.AnalysId = AnalysProdNumTable_ZTR.AnalysId AnalysRules_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId AnalysRules_ZTR.TypeRule = 5
OrderQty	$\sum$ AnalysDetail_ZTR.Qty	AnalysDetail_ZTR.DetailType = 1
OstZzQty	$\sum$ AnalysDetail_ZTR.Qty	AnalysDetail_ZTR.DetailType = 2
OstNoZzQty	$\sum$ AnalysDetail_ZTR.Qty	AnalysDetail_ZTR.DetailType = 3
OstAll	$\sum$ AnalysDetail_ZTR.Qty	AnalysDetail_ZTR.DetailType = 4
DefQty	AnalysProdNumLine_ZTR.Qty - (SpisQty + OrderQty + OstZzQty)	

Джерело даних – таблиця AnalysProdNumLine_ZTR за умовою:

- а) AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId (поточний);
- б) AnalysProdNumLine_ZTR.Type = 1 – правила застосовані.

Створити екранна кнопку «Створити журнал списань», дивитись нижче рисунок 2.5.

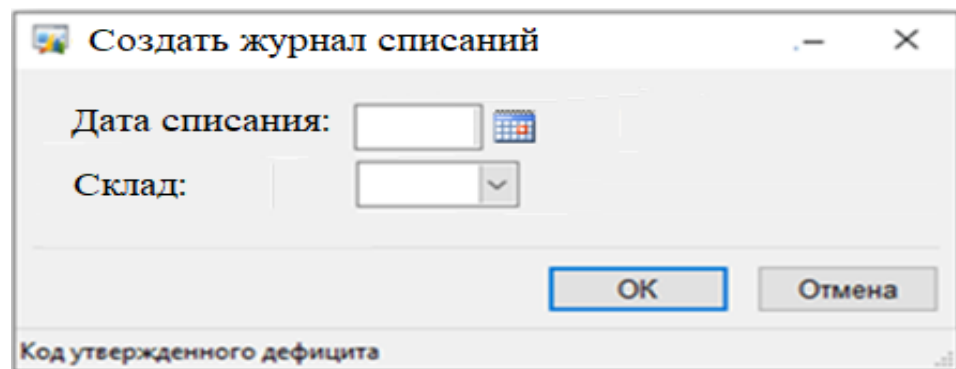


Рисунок 2.5 - Приклад створення вікна «Створити журнал списань»

Відкрити діалогове вікно

В полі «Склад» (обов'язкове поле) вибрати зі списку складів (таблиця InventLocation) за умовою:

а) InventLocation.WrkCtrGroupId_ZTR=AnalysProdNumTable_ZTR.WrkCtrGroupId_ZTR;

б) Substring ((InventLocation.InventLocationId, 1, 1) = «К».

В полі «Дата списання» внести інформацію в вигляді ДД.ММ.РРРР.

За кнопкою ОК виконати:

Створити журнал з типом «Проводка» (таблиця InventJournalTable) з полями (таблиця 2.18):

Таблиця 2.18 - InventJournalTable з полями

Назва поля таблиці InventJournalTable		Опис джерела даних
Журнал	JournalId	з номерний серії (Управління запасами - Налаштування - Параметри - закладка Номерні серії - поле Журнал)
Опис	Description	InventJournalName.Description
Ім'я	JournalNameId	Substring(поле «Склад» діалогу, 2, 3) + «_ПР_Списання»
Заводське замовл.	Dimention_ZTR7	AnalysProdNumTable_ZTR.ProdNum
Всі інші поля (Reservation, JournalType, VoucherDraw, VoucherChange, VoucherSeqId, DetailSummary, LedgerAccountIdOffset і інші) заповнити згідно стандарту.		

Створити рядки журналу (таблиця InventJournalTrans) з полями (табл. 2.19):

Таблиця 2.19 - InventJournalTrans з полями

Назва поля таблиці InventJournalTable		Опис джерела даних
Журнал	JournalId	InventJournalTable.JournalId
Дата	TransDate	поле «Дата списання» діалогу
Код номенклатури	ItemId	AnalysProdNumLine_ZTR.ItemId
Кількість	Qty * (-1)	AnalysProdNumLine_ZTR.OstZzQty

Всі інші поля (LineNum, JournalType, InventTransId, LedgerAccountIdOffset, Dimension_ZTR7_, InventDimId, ProdNum_ZTR, CurrencyCode_ZTR, Name_ZTR і інші) заповнити згідно зі стандартною функціональністю створення рядків журналу.

Створити автоматично рядки в таблицях InventTrans, InventTransPosting, InventSum згідно з стандартною функціональністю створення рядків журналу.

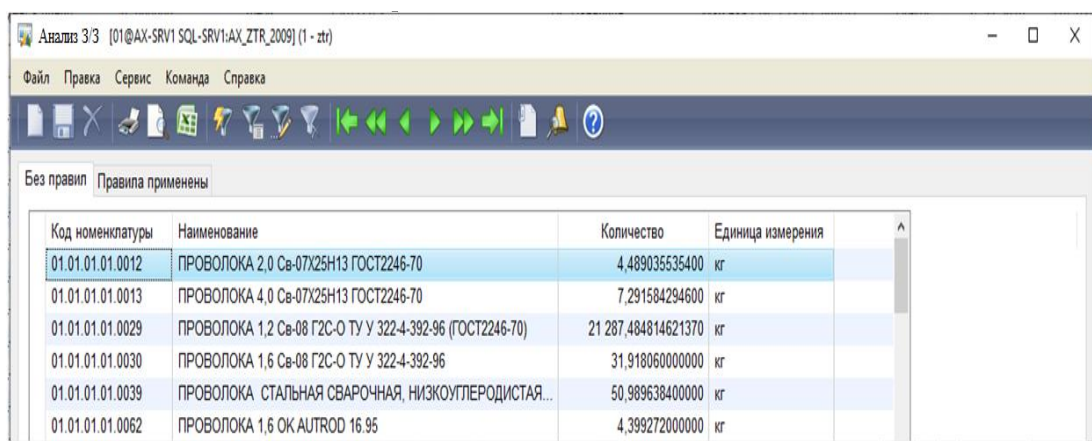
Джерело даних – таблиця AnalysDetail_ZTR за умовою:

- а) AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId (поточний);
- б) AnalysDetail_ZTR.InventLocationId = поле «Склад» діалогу;
- в) AnalysProdNumLine_ZTR.DefQty >= 0;
- г) AnalysProdNumLine_ZTR.OstZzQty > 0.

2.2.3 Реалізація нижньої частини форми

У нижній частині форми розмістити інформацію про створений аналізі 3 /3 і 2 вкладки («Без правил», «Правила застосовані»).

Вкладка «Без правил» - рисунок 2.6, вкладка «Правила застосовані» - рисунок 2.7.



Код номенклатуры	Наименование	Количество	Единица измерения
01.01.01.01.0012	ПРОВОЛОКА 2,0 Св-07Х25Н13 ГОСТ2246-70	4.489035535400	кг
01.01.01.01.0013	ПРОВОЛОКА 4,0 Св-07Х25Н13 ГОСТ2246-70	7.291584294600	кг
01.01.01.01.0029	ПРОВОЛОКА 1,2 Св-08 Г2С-О ТУ У 322-4-392-96 (ГОСТ2246-70)	21.287.484814621370	кг
01.01.01.01.0030	ПРОВОЛОКА 1,6 Св-08 Г2С-О ТУ У 322-4-392-96	31.918060000000	кг
01.01.01.01.0039	ПРОВОЛОКА СТАЛЬНАЯ СВАРОЧНАЯ, НИЗКОУГЛЕРОДИСТАЯ...	50.989638400000	кг
01.01.01.01.0062	ПРОВОЛОКА 1,6 ОК АУТРОД 16.95	4.399272000000	кг

Рисунок 2.6 - Приклад створення вкладки «Без правил»

Нижче у таблиці 2.20 наведений опис полів вкладки «Без правил».

Таблиця 2.20 – Без правил

Назва поля	Опис джерела даних	Примітка
Код номенклатури	AnalysProdNumLine_ZTR.ItemId	Не редагує
Найменування	InventTable.ItemName	Не редагує
Кількість	AnalysProdNumLine_ZTR.Quantity	Не редагує
Одиниця виміру	InventTableModule.UnitId	Не редагує

Сортування: Код номенклатури

На вкладку вибрати записи таблиці `AnalysProdNumLine_ZTR` за умовою:

- а) `AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId` (поточний);
- б) `AnalysTypeType = «0 – без правил»`.

Нижче наведений рисунок 2.7 - «Правила застосовані».

Код номенклатуры	Наименование	Количество	Единица	Карта разрезов	Таблица вза	Списано	Заказано	Остатки 3/3д	Остатки без 3/3д	Остатки все	Дефицит
01.01.01.01.0012	ПРОВОЛОКА 2,0 Св-07Х25Н13 ГОСТ2246-70	4,489036535400	кг	☐	☐						
01.01.01.01.0013	ПРОВОЛОКА 4,0 Св-07Х25Н13 ГОСТ2246-70	7,291504294600	кг	☐	☐						
01.01.01.01.0029	ПРОВОЛОКА 1,2 Св-08 Г2С-О ТУ У 322-4-392-96 (ГОСТ	287,484814621370	кг	☒	☐						
01.01.01.01.0030	ПРОВОЛОКА 1,6 Св-08 Г2С-О ТУ У 322-4-392-96	31,918060000000	кг	☒	☒						
01.01.01.01.0039	ПРОВОЛОКА СТАЛЬНАЯ СВАРЧНАЯ, НИЗКОУГЛЕ	50,989638400000	кг	☐	☐						
01.01.01.01.0062	ПРОВОЛОКА 1,6 ОК.АИТРОД 16.95	4,399272000000	кг	☐	☐						

Рисунок 2.7 - Приклад створення вкладки «Правила застосовані»

На вкладці розмістити 2 кнопки («Правила», «Детально»).

Нижче на таблиці 2.21 наведено опис полів вкладки «Правила застосовані».

Таблиця 2.21 – Правила застосовані

Назва поля	Опис джерела даних	Примітка
Код номенклатури	AnalysProdNumLine_ZTR.ItemId	Не редагуємо
Найменування	InventTable.ItemName	AnalysProdNumLine_ZTR.ItemId = InventTable.ItemId
Кількість	AnalysProdNumLine_ZTR.Qty	Не редагує
Одиниця виміру	InventTableModule.UnitId	InventTableModule.ItemId = AnalysProdNumLine_ZTR.ItemId InventTableModule.ModuleType = 0
Карта розрішень	AnalysProdNumLine_ZTR.Karta	Не редагує
Таблиця взаємозамінності	AnalysProdNumLine_ZTR.TableReplacement	Не редагує
Списано	AnalysProdNumLine_ZTR.SpisQty	Не редагує
Замовлено	AnalysProdNumLine_ZTR.OrderQty	Не редагує
Залишки 3/3, цех	AnalysProdNumLine_ZTR.OstZzQty	Не редагує
Залишки без 3/3, цех	AnalysProdNumLine_ZTR.OstNoZzQty	Не редагує
Залишки всі	AnalysProdNumLine_ZTR.OstAll	Не редагує
Дефіцит	AnalysProdNumLine_ZTR.DefQty	Не редагує
Примітка	AnalysProdNumLine_ZTR.Note	Редагує

Сортування: Код номенклатури

На вкладку обрати записи таблиці `AnalysProdNumLine_ZTR` за умовою:

а) `AnalysProdNumLine_ZTR.AnalysId=AnalysProdNumTable_ZTR.AnalysId` (поточний);

б) `AnalysTypeType = «1 – правила застосовані»`.

Екранна кнопка «Правила» (Карта розрішень, Таблиця взаємозамінності).

Для активної номенклатури форми відкрити не редаговане обране правило Карта розрішень, рисунок наведений нижче (рисунок 2.8).

Код номенклатури	Найменування	Кількість	Єдиниця	Цех	Номенклатура-заміна	Найменування	Кількість-за...	Єдиниця...	Номер пр. заказа по ЗТР	Коефіцієнт	Примечание
		0.000000		G007	01.02.02.02.0002	ЛЕНТА ДПРНТ 0.50x250 НД БРЮМЦ...	0.048000	кг	160 69/35Н-ПК-04		входить в ПУ
		0.000000		G004	03.05.03.02.1829	Наконечник КСИ 16/8	4.000000	Штук	A333765/ЗК-67	1.00	Зп. сообщ. ОПК
01.01.09.01.01.0002	КРУГ В-12 ГОСТ2590-2006/СТЗ...	10.070000	кг	G003	01.01.09.01.01.0152	КРУГ В-20 ГОСТ2590-2006/СЗмс5-3...	10.070000	кг	A333765/ЗК-67	1.00	Вовлечение TML
01.01.09.01.01.0002	КРУГ В-12 ГОСТ2590-2006/СТЗ...	0.390000	кг	G005	01.01.09.01.01.0152	КРУГ В-20 ГОСТ2590-2006/СЗмс5-3...	0.390000	кг	A333765/ЗК-67	1.00	Вовлечение TML
01.01.08.03.0160	ТРУБА 152x8.0 ГОСТ8732-78/В...	1.011574	м	G003	01.01.08.02.0070	ТРУБА 152x8.0 12x18x110Т ГОСТ 994...	1.011574	м	A333765/ЗК-67	1.00	Вовлечение TML
01.09.01.02.0029	КАРТОН Б-3.00-3200x1200 ГОСТ...	529.700000	кг	G008	01.09.01.03.0261	Опорное кольцо НД 2040 мм ВН 185...	2.000000	Штук	A333765/ЗК-67	1.00	№218(Х1)
01.09.01.03.0099	Картон ламинатный ИЕС 60763-3...	78.946000	кг	G008	01.09.01.03.0262	Опорное кольцо НД 1788 мм ВН 155...	2.000000	Штук	A333765/ЗК-67	1.00	№218(Х1)
01.09.01.03.0105	Картон ламинатный ИЕС 60763-3...	60.206000	кг	G008	01.09.01.03.0263	Опорное кольцо НД 1793 мм ВН 155...	2.000000	Штук	A333765/ЗК-67	1.00	№218(Х1)
01.09.01.03.0111	Картон ламинатный ИЕС 60763-3...	198.728000	кг	G008	01.09.01.03.0264	Опорное кольцо НД 1331 мм ВН 121...	2.000000	Штук	A333765/ЗК-67	1.00	№218(Х1)

Рисунок 2.8 – Приклад створення вкладки «Карта розрешень»

Нижче, у таблиці 2.22 наведений опис полів правила Карта розрешень.

Таблиця 2.22 – Карта розрешень

Назва поля	Опис джерела даних	Примітка
Код номенклатури	AnalysRules_ZTR.ItemId	
Найменування	InventTable.ItemName	PotrbRules_ZTR.ItemId = InventTable.ItemId
Кількість	AnalysRules_ZTR.Qty	
Одиниця виміру	InventTableModule.UnitId	InventTableModule.ItemId =PotrbRules_ZTR.ItemId InventTableModule.ModuleType = 0
Цех	AnalysRules_ZTR.WrkCtrGroupId	
Номенклатура-заміна	AnalysRules_ZTR.ItemIdDuplicate	
Найменування	InventTable.ItemName	PotrbRules_ZTR.ItemIdDuplicate =InventTable.ItemId
Кількість-заміна	AnalysRules_ZTR.QtyDuplicate	
Одиниця виміру	InventTableModule.UnitId	InventTableModule.ItemId =PotrbRules_ZTR.ItemIdDuplicate InventTableModule.ModuleType = 0
Номер вироб. замовлення по ЗТР	AnalysRules_ZTR.ProdNum	
Коефіцієнт	AnalysRules_ZTR.Koef	

Продовження таблиці 2.22

Назва поля	Опис джерела даних	Примітка
Примітка	AnalysRules_ZTR.Note	
Автор створення правила	AnalysRules_ZTR.CreatedByRules	
Дата створення правила	AnalysRules_ZTR.CreatedDateTimeRules	

На вкладку обрати записи таблиці PotrbRules_ZTR за умовою:

- а) `AnalysRules_ZTR.TypeRule = 5;`
- б) `PotrbRules_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId.`

Для активної номенклатури форми відкрити не редаговане обране правило Карта розрішень, рисунок наведений нижче (рисунок 2.9).

Код номенклатуры	Наименование	Номенклатура согласно КД	Наименование	Направление	Примечание	Дата
01.01.09.01.01.0022	КРУГ В-66 ГОСТ2590-2006/СТЗСП2-ЗГП ДСТУ4484.20...	01.01.04.01.0005	УГОЛОК В-50Х50Х5 ДСТУ 2251-93 (Г...	в обе стороны	распоряжение №1/68-254 от 15.08.19	15.08.2019
01.17.23.07.0443	Сальник 25х42х10-RSTANBR	01.01.04.01.0003	УГОЛОК В-40Х40Х4 ДСТУ 2251-93 (Г...	в обе стороны	распоряжение №1/68-254 от 15.08.19	15.08.2019
05.10.12.0029	ЗМ 947D 25х457 P80 Шлифовальная лента 25х457 мм...	05.10.12.0024	Лента шлифовальная 947D 25х457 P40			
ВБМЕ.322412.002-17	МЕШОК	ВБМЕ.322412.002-20	МЕШОК		распоряжение №50 от 14.08.18	14.08.2018
01.03.08.01.0003	КАБЕЛЬ КТВВ 4х2,5 ТУ 16-505.665-74	01.03.07.01.0397	КАБЕЛЬ КВВГ нд-LS 4х2,5 (IV кл.) Т...		распоряжение №62 от 18.09.18	18.09.2018
01.04.05.0004	ГЕТИНАКС I 1с-6,0 ГОСТ2718-74	01.04.05.0006	ГЕТИНАКС I 1с-10,0 ГОСТ2718-74	в одну сторону	распоряжение №51 от 14.08.18	14.08.2018
01.04.05.0017	ГЕТИНАКС V 1с-15,0 ГОСТ2718-74	01.04.05.0018	ГЕТИНАКС V 1с-20,0 ГОСТ2718-74		распоряжение №51 от 14.08.18	14.08.2018
01.03.07.01.0400	КАБЕЛЬ КВВГ нд-LS 7х2,5 (IV кл.) ТУ У 31.3-19275366...	01.03.08.01.0005	КАБЕЛЬ КТВВ 7 х2,5 ТУ 16-505.665-74		распоряжение №62 от 18.09.18	18.09.2018
01.04.05.0011	ГЕТИНАКС I 1с-30,0 ГОСТ2718-74	01.04.05.0020	ГЕТИНАКС V 1с-30,0 ГОСТ2718-74		распоряжение №51 от 14.08.18	14.08.2018
01.02.13.02.0037		01.02.13.02.0015	ШИНА АДО ВХ40 НД ГОСТ15176-89	в обе стороны	распоряжение №19 от 28.02.19	28.02.2019

Рисунок 2.9 – Приклад створення вкладки «Таблица взаємозамінності»

Нижче, у таблиці 2.23 описати поля правила Таблица взаємозамінності.

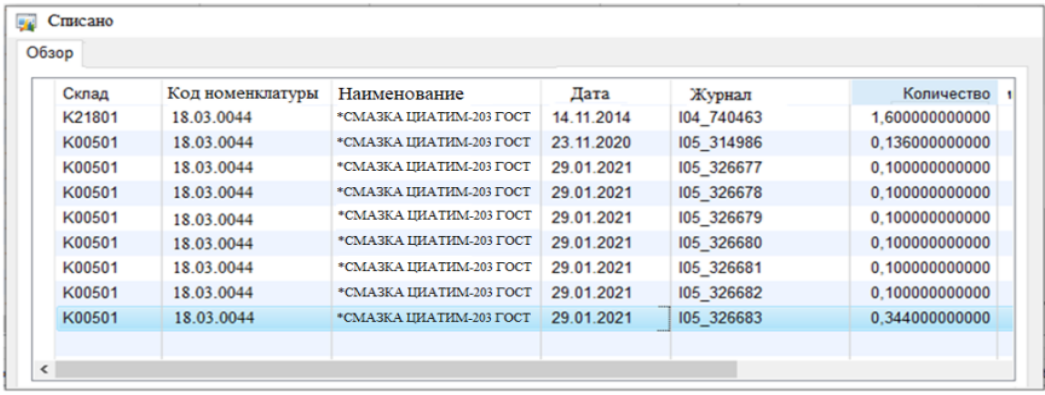
Таблица 2.23 – Таблица взаємозамінності

Назва поля	Опис джерела даних	Примітка
Код номенклатуры	AnalysRules_ZTR.ItemId	
Найменування	InventTable.ItemName	AnalysRules_ZTR.ItemId = InventTable.ItemId
Номенклатура згідно КД	AnalysRules_ZTR.ItemIdDuplicate	
Направлення	AnalysRules_ZTR.PotrbRoute	
Примітка	AnalysRules_ZTR.Note	
Дата	AnalysRules_ZTR.NoteDate	

На вкладку вибрати записи таблиці PotrbRules_ZTR за умовою:

- а) `AnalysRules_ZTR.TypeRule = 1;`
- б) `AnalysRules_ZTR.AnalysId = AnalysProdNumTable_ZTR.AnalysId;`
- в) `AnalysRules_ZTR.ItemIdDuplicate=AnalysProdNumLine_ZTR.ItemId` або `AnalysRules_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId .`

Екранна кнопка «Детально» (Списано, Замовлено, Залишки 3 / 3, цех, Залишки без 3 / 3, цех, Залишки всі). Для активної номенклатури форми відкрити не редаговані обрану деталізацію Списано (рисунок 2.10).



The screenshot shows a window titled 'Списано' with a tab 'Обзор'. It contains a table with the following data:

Склад	Код номенклатуры	Наименование	Дата	Журнал	Количество
K21801	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	14.11.2014	I04_740463	1,600000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	23.11.2020	I05_314986	0,136000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326677	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326678	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326679	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326680	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326681	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326682	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	29.01.2021	I05_326683	0,344000000000

Рисунок 2.10 – Приклад створення вкладки «Списано»

Нижче, в таблиці 2.24 наведено опис полів деталізації Списано.

Таблиця 2.24 – Списано

Назва поля	Опис джерела даних	Примітка
Склад	<code>AnalysDetail_ZTR.InventLocationId</code>	
Назва поля	Опис джерела даних	Примітка
Код номенклатури	<code>AnalysDetail_ZTR.ItemId</code>	
Найменування	<code>InventTable.ItemName</code>	<code>AnalysDetail_ZTR.ItemId = InventTable.ItemId</code>
Одиниця виміру	<code>InventTableModule.UnitId</code>	<code>InventTableModule.ItemId = AnalysDetail_ZTR.ItemId</code> <code>InventTableModule.ModuleType=0</code>
Дата	<code>AnalysDetail_ZTR.DateFinancial</code>	
Журнал	<code>AnalysDetail_ZTR.JournalId</code>	
Кількість	<code>AnalysDetail_ZTR.Qty</code>	

Продовження таблиці 2.24

Назва поля	Опис джерела даних	Примітка
Партія	AnalysDetail_ZTR.InventBatchId	
Рівень	AnalysDetail_ZTR.Level1	

На вкладку вибрати записи таблиці `AnalysDetail_ZTR` за умовою:

- а) `AnalysDetail_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId;`
- б) `AnalysDetail_ZTR.DetailType = 0.`

Для активної номенклатури форми відкрити не редаговані обрану деталізацію Замовлено (рисунок 2.11).

Склад	Код номенклатуры	Наименование	Дата	Журнал	Количество
K21801	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I04_740463	1.600000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_314986	0.136000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326677	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326678	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326679	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326680	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326681	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326682	0.100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ		I05_326683	0.344000000000

Рисунок 2.11 – Приклад створення вкладки «Замовлено»

Нижче, в таблиці 2.25 наведено опис полів деталізації Замовлено.

Таблиця 2.25 – Замовлено

Назва поля	Опис джерела даних	Примітка
Склад	AnalysDetail_ZTR.InventLocationId	
Код номенклатури	AnalysDetail_ZTR.ItemId	
Найменування	InventTable.ItemName	AnalysDetail_ZTR.ItemId = InventTable.ItemId
Назва поля	Опис джерела даних	Примітка
Одиниця виміру	InventTableModule.UnitId	InventTableModule.ItemId = AnalysDetail_ZTR.ItemId InventTableModule.ModuleType = 0
Дата	AnalysDetail_ZTR.DateFinancial	

Продовження таблиці 2.25

Назва поля таблиці	Опис джерела даних	Примітка
Журнал	AnalysDetail_ZTR.JournalId	
Кількість	AnalysDetail_ZTR.Qty	
Партія	AnalysDetail_ZTR.InventBatchId	
Назва поля	Опис джерела даних	Примітка
Рівень	AnalysDetail_ZTR.Level1	

На вкладку вибрати записи таблиці `AnalysDetail_ZTR` за умовою:

- а) `AnalysDetail_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId;`
- б) `AnalysDetail_ZTR.DetailType = 1.`

Для активної номенклатури форми відкрити не редаговані обрану деталізацію Залишки 3/3,цех (рисунок 2.12).



Склад	Код номенклатури	Наименование	Номер пр. заказа по ЗТР	Количество
K21801	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	1,600000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	0,136000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	0,100000000000

Рисунок 2.12 – Приклад створення вкладки «Залишки 3/3,цех»

Нижче, в таблиці 2.26 наведено опис полів деталізації Залишки 3/3,цех.

Таблиця 2.26 – Залишки 3/3

Назва поля	Опис джерела даних	Примітка
Склад	AnalysDetail_ZTR.InventLocationId	
Код номенклатури	AnalysDetail_ZTR.ItemId	
Найменування	InventTable.ItemName	AnalysDetail_ZTR.ItemId = InventTable.ItemId
Одиниця виміру	InventTableModule.UnitId	InventTableModule.ItemId = AnalysDetail_ZTR.ItemId InventTableModule.ModuleType = 0
Номер вироб. замовлення по ЗТР	AnalysDetail_ZTR.Prodnum	

Продовження таблиці 2.26

Назва поля	Опис джерела даних	Примітка
Кількість	AnalysDetail_ZTR.Qty	

На вкладку вибрати записи таблиці `AnalysDetail_ZTR` за умовою:

- а) `AnalysDetail_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId;`
- б) `AnalysDetail_ZTR.DetailType = 2.`

Для активної номенклатури форми відкрити не редаговані обрану деталізацію Залишки без З/З,цех (рисунок 2.13).



Склад	Код номенклатуры	Наименование	Номер пр. заказа по ЗТР	Количество
K21801	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	1,600000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,136000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,100000000000

Рисунок 2.13 – Приклад створення вкладки «Залишки без З/З,цех»

Нижче, в таблиці 2.27 наведено опис полів деталізації Залишки без З/З,цех.

Таблиця 2.27 – Залишки без З/З

Назва поля	Опис джерела даних	Примітка
Склад	<code>AnalysDetail_ZTR.InventoryLocationId</code>	
Код номенклатури	<code>AnalysDetail_ZTR.ItemId</code>	
Найменування	<code>InventoryTable.ItemName</code>	<code>AnalysDetail_ZTR.ItemId = InventoryTable.ItemId</code>
Одиниця виміру	<code>InventoryTableModule.UnitId</code>	<code>InventoryTableModule.ItemId = AnalysDetail_ZTR.ItemId</code> <code>InventoryTableModule.ModuleType = 0</code>
Номер вироб. замовлення по ЗТР	<code>AnalysDetail_ZTR.Prodnum</code>	
Кількість	<code>AnalysDetail_ZTR.Qty</code>	

На вкладку вибрати записи таблиці `AnalysDetail_ZTR` за умовою:

- а) `AnalysDetail_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId;`
- б) `AnalysDetail_ZTR.DetailType = 3.`

Для активної номенклатури форми відкрити не редаговані обрану деталізацію Залишки всі (рисунок 2.14).



Склад	Код номенклатуры	Наименование	Номер пр. заказа по ЗТР	Количество
K21801	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	1,600000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,136000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	050/6880	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,100000000000
K00501	18.03.0044	*СМАЗКА ЦИАТИМ-203 ГОСТ	Без з/заказа	0,100000000000

Рисунок 2.14 – Приклад створення вкладки «Залишки всі»

Нижче, в таблиці 2.28 наведено опис полів деталізації Залишки всі.

Таблиця 2.28 – Залишки всі

Назва поля	Опис джерела даних	Примітка
Склад	<code>AnalysDetail_ZTR.InventLocationId</code>	
Код номенклатури	<code>AnalysDetail_ZTR.ItemId</code>	
Найменування	<code>InventTable.ItemName</code>	<code>AnalysDetail_ZTR.ItemId = InventTable.ItemId</code>
Одиниця виміру	<code>InventTableModule.UnitId</code>	<code>InventTableModule.ItemId = AnalysDetail_ZTR.ItemId</code> <code>InventTableModule.ModuleType = 0</code>
Номер вироб. замовлення по ЗТР	<code>AnalysDetail_ZTR.Prodnum</code>	
Кількість	<code>AnalysDetail_ZTR.Qty</code>	

На вкладку вибрати записи таблиці `AnalysDetail_ZTR` за умовою:

- а) `AnalysDetail_ZTR.RefRecId = AnalysProdNumLine_ZTR.RecId;`
- б) `AnalysDetail_ZTR.DetailType = 4.`

2.3 Висновок за розділом 2

Для створення даного функціоналу був обраний ефективний та сучасний набір інструментів. Середовищем розробки є ERP – система Microsoft Dynamics AX, мова що використовується для даної системи – X++.

Був розроблений користувацький інтерфейс, основним елементом якого є форма, яка складається з верхньої шапки, нижніх рядків та чотирьох кнопок. В якості сховища інформації використовується реляційна база даних. Даний програмний продукт було реалізовано за допомогою створення чотирьох класів, методи яких виконують поставлені задачі.

3 ТЕСТУВАННЯ ПРОДУКТУ ТА ПРАКТИЧНІ РЕКОМЕНДАЦІЇ

3.1 Тестування роботи системи

Була створена форма, яка складається з верхньої шапки, куди записується потрібний аналіз заводського замовлення та нижніх рядків, куди записуються усі номенклатури даного аналізу, які сортуються за правилами в наступній вкладці.

Для створення аналізу, потрібно натиснути на кнопку «Створити аналіз 3/3», після цього відкривається діалогове вікно, де записується:

- номер виробничого замовлення по підприємству;
- назва цеху;
- код номенклатури;
- код схеми.

Після натискання кнопки «ОК», йде оброблення даних та створення аналізу. Після закінчення оброблення у верхній шапці з'являється аналіз, який обрав користувач, у нижній рядки з усіма номенклатурами у першій вкладці «Без правил» та рядки відфільтровані за правилами у другій вкладці «Правила застосовані».

В нижній частині форми, у вкладці «Правила застосовані» розміщено дві кнопки «Правила», «Детально».

Кнопка «Правила» відкриває нове вікно з нашими відфільтрованими правилами, розподіленими на дві вкладки «Таблиця взаємозамінності», «Карта розрішень», й виводє всю інформацію за цими правилами (актуальність, код номенклатури, найменування, одиниця виміру, цех і т.д) .

Кнопка «Детально» розкривається на п'ять додаткових кнопок (Списано, Замовлено, Залишки 3 / 3, цех, Залишки без 3 / 3, цех, Залишки всі). Кожна кнопка відкриває нове вікно де йде фільтрація номенклатур за

статусами(Списано, Замовлено, Залишки 3 / 3, цех, Залишки без 3 / 3, цех, Залишки всі).

Порівнюючи даний процес с роботою вручну, витрачається набагато менше часу, адже даний аналіз виконується не більше 10 хвилин.

В ході експлуатації критичних помилок не було виявлено. Форма дозволяє зручно знаходити потрібний аналіз необхідних для списання під виробництво замовлень ТМЦ (товарно-матеріальних цінностей). А точніше:

- знаходити потрібні номенклатури та проводити аналіз заводського замовлення;
- сортувати потрібні номенклатури за правилами (Карта розрішення та Таблиця взаємозамінності);
- створити журнал списань;
- для активної номенклатури форми відкривати не редаговану обрану деталізацію за правилами (Карта розрішення, Таблиця взаємозамінності);
- для активної номенклатури форми відкривати не редаговану обрану деталізацію (Списано, Замовлено, Залишки 3 / 3, цех, Залишки без 3 / 3, цех, Залишки всі).

Також хотілось б нагадати, що даний функціонал можуть використовувати тільки працівники (консультанти) підприємств, які працюють з ERP – системою, адже дана програма реалізована саме для цього середовища.

3.2 Висновок за розділом 3

Головною метою написання даного функціоналу було оптимізувати витрати робочого часу робітників підприємства, при проведенні аналізу наявності ТМЦ(з урахуванням КД, карт дозволів, таблиці взаємозамінності, залишків на заводі, і ін.) для списання під виробничі замовлення в системі

Microsoft Dynamics AX. Адже, раніше даний аналіз виконувався вручну, цей процес є дуже трудомістким. Тепер даний аналіз проводиться автоматично.

Основним недоліком програми є час виконання поставленої задачі(хоч це всеодно швидше, ніж виконувати вручну), адже задачею програми є обробка дуже великого об'єму даних. Оскільки об'єми інформації великі і число користувачів, які працюють з даними досить велике, час виконання може збільшуватись, в залежності від вище описаних факторів. Проте даний програмний продукт повністю виконує поставлені, користувачем задачі.

Продукт можна назвати оптимізованим та «легким» для використання у роботі консультантами підприємств.

ВИСНОВКИ

Під час виконання дипломної кваліфікаційної роботи бакалавра було розроблено програмне забезпечення та інтерфейс для проведення аналізу товарно-матеріальних цінностей. Було проведено аналіз предметної області та сучасних технологій, враховані можливі труднощі при розробці програмного забезпечення, та на їх основі поставлено та вирішено наступні задачі:

- обрано технології та інструменти, за допомогою яких був створений програмний функціонал аналізу заводського замовлення;
- розширення функціоналу ERP-системи;
- створення зрозумілого та простого користувацького інтерфейсу;
- налаштування взаємодії сервера та користувача через систему АХАРТА;

Використовуючи такі технології як Microsoft Dynamics AX, X++, MorphX і MS SQL Server, було розроблено програмне забезпечення, яке включає в себе програмний інтерфейс засобу для взаємодії між користувачем та сервером. Таким чином, мета розробки, поставлена в технічному завданні, досягнута, і всі задачі вирішені.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ахарт: що це за програма і для чого вона потрібна? [Електрон. ресурс]. – Режим доступу: <https://ukr.kagutech.com/4293134-axapta-what-is-this-program-and-what-is-it-for>.
2. Microsoft Dynamics AX 2009 [Електрон. ресурс]. – Режим доступу: http://www.giustoconsulting.com/wp-content/uploads/2009/10/02_whats-new-Microsoft-Dynamics-AX2009.pdf.
3. Обзор рынка ERP-систем в Украине [Електрон. ресурс]. – Режим доступу: <https://kamala-soft.com/blog/obzor-rynka-erp-sistem-v-ukraine/>.
4. Шашлова И. ERP-системы: Сравнительный анализ и основные особенности [Електрон. ресурс] / И. Шашлова. – Режим доступу: <http://sdo.rea.ru/cde/conference/28/file.php?fileId=36>.
5. Сравниваем ERP по ключевым характеристикам [Електрон. ресурс]. – Режим доступу: https://www.cnews.ru/articles/sravnivaem_erp_po_klyuchevym_harakteristikam.
6. ERP-система (планування ресурсів підприємства) [Електрон. ресурс]. – Режим доступу: https://pidru4niki.com/1171062647760/informatika/erp-sistema_planuvannya_resursiv_pidpriyemstva.
7. Microsoft Dynamics AX 2012 обзор возможностей [Електрон. ресурс]. – Режим доступу: <http://www.omegaplus.ru/bitrix/templates/omega3/download/AX2012.pdf>.
8. AXForum [Електрон. ресурс]. – Режим доступу: <https://axforum.info/forums/showthread.php?t=360>.
9. АНД Проджект (СПб) [Електрон. ресурс]. – Режим доступу: <http://www.andproject.spb.ru/andproj...B-F82B1AB94EE8>.

10. Офіційний веб-сайт Microsoft Dynamics AX в Україні [Електрон. ресурс]. – Режим доступу: <http://www.microsoft.com/ukraine/dynamics/axapta/default.mspx>.

11. MorphX IT Введение в язык программирования Ахapta X++ и среду разработки MorphX [Электрон. ресурс]. – Режим доступа: <http://book.axforum.info/morphx.pdf>.

12. Обзор среды разработки Microsoft Dynamics AX 4.0 Где используется программа Microsoft Dynamics Ахapta и как в ней работают [Электрон. ресурс]. – Режим доступа: <http://composs.ru/axapta-что-eto-za-programma/>.

ДОДАТОК А
Текст програми

Клас AnalysProdNumTableCreate_ZTR

ClassDeclaration:

```
class AnalysProdNumTableCreate_ZTR extends RunBase
{
```

```
    DialogRunbase        dialog;
    DialogField           prodNumFld;
    DialogField           wrkCtrGroupFld;
    DialogField           itemIdTypeBOMFld;
    DialogField           bomSchemeCodeFld;
    BomTreeGenerator_ZTR  bomTreeGenerator;
    TmpBOMWrkTreeItems_ZTR  localItems;
    AnalysDetail_ZTR      analysDetail;
    AnalysProdNumLine_ZTR  analysProdNumLine;
    AnalysProdNumTable_ZTR  analysProdNumTable;
    WrkCtrTable           wrkCtrTable;
    ProdNum_ZTR           prodNum;
    WrkCtrGroupId         wrkCtr;
    ItemIdBom_ZTR         itemIdTypeBOM;
    BOMSchemeCode_ZTR     bomSchemeCode;
    PotrbRules_ZTR        potrbRules;

    boolean               useSubsidiaryMaterial;
    CodeSatusBom_ZTR      minCodeSatusBom;
    TmpBomTreeItems_ZTR   tmpBomTreeItems, tmpBomtreeItemsFromreport;
    RecordInsertList      tmpBomTreeItemsInsertList;
    int64                 tmpBomTreeItemsCounter;
    TmpBomTree_ZTR        tmpBomTree;
    Bom                   bom;
    InventDim             inventDim;
    InventLocation         inventLoc;
    AnalysID              analysId;
    int                   sesId;
    NoYes                 isFound;
    #DEFINE.CurrentVersion(4)
    #LOCALMACRO.CurrentList
        prodNum,
        wrkCtr,
        itemIdTypeBOM,
        bomSchemeCode
    #ENDMACRO
    #AviFiles
```

```
    }
    public void insertHeader()
    {
```

```
        ;
        analysId
```

=

```
        NumberSeq::newGetNum(PotrbDefParameters_ZTR::numRefAnalysId()).num();
        analysProdNumTable.clear();
        analysProdNumTable.initValue();
        analysProdNumTable.AnalysId = analysId;
        analysProdNumTable.ProdNum = prodnum;
        analysProdNumTable.WrkCtrGroupId = wrkCtr;
        analysProdNumTable.ItemIdBom = itemIdTypeBOM;
        analysProdNumTable.SchemeCode = bomSchemeCode;
        if (analysProdNumTable.validateWrite())
```

```

        {
            analysProdNumTable.insert();
        }
    }
protected void createData()
{
    bomTreeGenerator = new BomTreeGenerator_ZTR();
    bomTreeGenerator.parmUseTemporaryTables(false);

    if (bomTreeGenerator.parmUseTemporaryTables())
    {
        tmpBomTreeItems =
        bomTreeGenerator.getCopyTmpBomTreeItemsTemporary();
        //tmpBomTreeIgnoreItems =
        bomTreeGenerator.getCopyTmpBomTreeIgnoreItemsTemporary();
    }
    else
    {
        tmpBomTreeItemsInsertList = new
        RecordInsertList(tablenum(TmpBomTreeItems_ZTR), true, true);
        //tmpBomTreeIgnoreItemsInsertList = new
        RecordInsertList(tablenum(TmpBomTreeIgnoreItems_ZTR), true, true);
    }
    bomTreeGenerator.clearTables();

    bomTreeGenerator.parmExplActOnNotPhantomNotStopExpl(BomTreeExplodeAction_Z
TR::Explode);

    bomTreeGenerator.parmExplActOnPhantomNotStopExpl(BomTreeExplodeAction_ZTR
::Explode);

    bomTreeGenerator.parmExplActOnPhantomStopExpl(BomTreeExplodeAction_ZTR::Ex
plode);

    bomTreeGenerator.parmExplActOnStopExplNotPhantom(BomTreeExplodeAction_ZTR
::Explode);
        bomTreeGenerator.parmCalcBomQtyTree(true);

        if (useSubsidiaryMaterial)
            //
            bomTreeGenerator.parmUseSubsidiaryMaterial(BomTreeUseSubsidiaryMaterial_
ZTR::Type2, BomTreeExplodeAction_ZTR::ReplaseByChildrends);

    bomTreeGenerator.parmUseSubsidiaryMaterial(BomTreeUseSubsidiaryMaterial_ZTR::T
ype5, BomTreeExplodeAction_ZTR::ReplaseByChildrends);
        bomTreeGenerator.parmMinCodeSatusBom(minCodeSatusBom);
        if (!BomStatusTable_ZTR::find(minCodeSatusBom).Approved)
        {
            bomTreeGenerator.parmUseNotApprovedBoms(true);
            bomTreeGenerator.parmUseNotActivatedBoms(true);
        }
        bomTreeGenerator.parmCalcInventLocationConsumer(true, 3);

```

```

//while select ItemIdBom
//{
//tmpBomTreeItemsCounter++;
tmpBomTreeItems.clear();
tmpBomTreeItems.initValue();
tmpBomTreeItems.initFromBomTreeGenerator(bomTreeGenerator);
tmpBomTreeItems.ItemId = itemIdTypeBOM;
tmpBomTreeItems.WrkCtrIssueId = wrkCtr;
tmpBomTreeItems.SchemeCode = bomSchemeCode;
tmpBomTreeItems.ProdNum = prodNum;
tmpBomTreeItems.Qty = 1;
tmpBomTreeItems.Num = 1;
// if (tmpBomTreeItems.ProdNum)
//
//                                     tmpBomTreeItems.SchemeCode      =
ProdNumTable_ZTR::find(tmpBomTreeItems.ProdNum).SchemeCode;
//    if (localItems.TmpNoYes)
//
//                                     tmpBomTreeItems.ExplActOnCooperation  =
BomTreeExplodeAction_ZTR::Explode;
//    else
//
//                                     tmpBomTreeItems.ExplActOnCooperation  =
BomTreeExplodeAction_ZTR::ConvertToMaterial;
// VALY, Task2217, 27.04.2011 -->
// закоментирован код (заменен):
// tmpBomTreeItems.RecId = tmpBomTreeItemsCounter;
if (!bomTreeGenerator.parmUseTemporaryTables())
    tmpBomTreeItems.RecId = TmpBomTreeItems_ZTR::nextRecId();
// VALY, Task2217, 27.04.2011 <--
tmpBomTreeItems.insertUsingInsertList(tmpBomTreeItemsInsertList);
//}

if (tmpBomTreeItemsInsertList)
{
    tmpBomTreeItemsInsertList.insertDatabase();
    tmpBomTreeItemsInsertList = null;
}
bomTreeGenerator.run();
sesId = bomtreegenerator.parmSesId();
}
public void insertLines()
{
;

while select sum(BomQty) from tmpBomTree
group by ItemId
where tmpBomTree.SesId == sesId
join bom
where tmpBomTree.BomRecId == bom.RecId
join inventDim
where bom.InventDimId == inventDim.inventDimId
join inventLoc
where inventDim.InventLocationId == inventLoc.InventLocationId
&& inventLoc.WrkCtrGroupId_ZTR == analysProdNumTable.WrkCtrGroupId
{

```

```

        analysProdNumLine.clear();
        analysProdNumLine.initValue();
        analysProdNumLine.AnalysId = analysId;
        analysProdNumLine.AnalysType = PotrbType_ZTR::NoRules;
        analysProdNumLine.ItemId = tmpBomTree.ItemId;
        analysProdNumLine.Qty = tmpBomTree.BomQty;
        if (analysProdNumLine.validateWrite())
        {
            analysProdNumLine.insert();
        }
    }
    Bomtreegenerator.clearTables();
}

public void applyRulesDublicate()
{
    AnalysProdNumLine_ZTR  analysProdNumLineNew,
                          analysProdNumLineCopy;
    AnalysRules_ZTR        analysRules;
    ;
    ttsbegin;
    while select analysProdNumLine
        order          by          analysProdNumLine.AnalysId,
        analysProdNumLine.ItemId,analysProdNumLine.Qty, analysProdNumLine.AnalysType
        where analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join ItemIdDuplicate from potrbRules
        where potrbRules.TypeRule  == PotrbTypeRule_ZTR::Duplicate
        && potrbRules.Actual  == NoYes::Yes
        && potrbRules.ItemId  == analysProdNumLine.ItemId
    {
        select firstly forupdate analysProdNumLineNew where
        analysProdNumLineNew.RecId == analysProdNumLine.RecId;
        analysProdNumLineNew.ItemId = potrbRules.ItemIdDuplicate;
    }
    ttscommit;
    this.groupActualRulesYes();
}

public void applyRulesPermissionCard()
{
    AnalysProdNumLine_ZTR  analysProdNumLineNew,
                          analysProdNumLineCopy;

    AnalysRules_ZTR        analysRules;
    ;
    ttsbegin;
    while select RecId from analysProdNumLine
        group by RecId

```

```

//group by ItemId, BudgetCode_ZTR, ProdNum_ZTR, WrkCtrGroupId
    where  analysProdNumLine.AnalysId == analysId  &&
    analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join ItemId from potrbRules
    where  potrbRules.TypeRule == PotrbTypeRule_ZTR::PermissionCard //Капра
разрешений
        && potrbRules.ItemId == analysProdNumLine.ItemId
        && potrbRules.Actual == NoYes::Yes
        join AnalysId from analysProdNumTable
        where  analysProdNumTable.AnalysId ==
analysProdNumLine.AnalysId
        && potrbRules.ProdNum_ZTR ==
analysProdNumTable.ProdNum
        && potrbRules.WrkCtrGroupId ==
analysProdNumTable.WrkCtrGroupId
    {
        select firstly forupdate analysProdNumLineNew where
        analysProdNumLineNew.RecId == analysProdNumLine.RecId;
        //analysProdNumLineNew.RecId =
analysProdNumLine.RecId;

        if(potrbRules.Koef != 1)
        {
            analysProdNumLineNew.ItemId =
            potrbRules.ItemIdDuplicate;
            analysProdNumLineNew.Qty =
            analysProdNumLineNew.Qty * potrbRules.Koef;
        }
        else
        {
            if(analysProdNumLineNew.Qty - potrbRules.Qty >
0)
            {
                analysProdNumLineCopy.clear();
                analysProdNumLineCopy.initValue();

                buf2Buf(analysProdNumLineNew, analysProdNumLineCopy);
                analysProdNumLineCopy.Qty = analysProdNumLineCopy.Qty -
potrbRules.Qty;
                analysProdNumLineCopy.AnalysType =
PotrbType_ZTR::YesRules;
                if (analysProdNumLineCopy.validateWrite())
                {
                    analysProdNumLineCopy.insert();

                    this.insertintoAnalysRules(analysProdNumLineCopy,potrbRules);
                }
            }
            analysProdNumLineNew.ItemId =
            potrbRules.ItemIdDuplicate;
            analysProdNumLineNew.Qty = potrbRules.QtyDuplicate;

```

```

        analysProdNumLineNew.AnalysType
PotrbType_ZTR::YesRules;
        if (analysProdNumLineNew.validateWrite())
        {
            analysProdNumLineNew.insert();

this.insertintoAnalysRules(analysProdNumLineNew,potrbRules);
        }
    }

// Б
while select potrbRules
    where potrbRules.TypeRule == PotrbTypeRule_ZTR::PermissionCard
        && potrbRules.Actual == NoYes::Yes
        && !potrbRules.ItemId
        && analysProdNumLine.ItemId == potrbRules.ItemIdDuplicate
        && analysProdNumLine.Qty == potrbRules.QtyDuplicate
exists join ProdNum from analysProdNumTable
    where potrbRules.ProdNum_ZTR == analysProdNumTable.ProdNum
    {
        analysProdNumLineNew.clear();
        analysProdNumLineNew.initValue();
        analysProdNumLineNew.AnalysType
PotrbType_ZTR::YesRules;
        analysProdNumLineNew.ItemId
potrbRules.ItemIdDuplicate;
        analysProdNumLineNew.Qty = potrbRules.QtyDuplicate;
        analysProdNumLineNew.AnalysId = analysid;
        if (analysProdNumLineNew.validateWrite())
        {
            analysProdNumLineNew.insert();

this.insertintoAnalysRules(analysProdNumLineNew,potrbRules);
        }
    }
    isFound = true;
    ttscommit;
    this.groupActualRulesYes();
}
public void insertintoAnalysRules(AnalysProdNumLine_ZTR _line, PotrbRules_ZTR
_potrbRules)
{
    AnalysRules_ZTR      analysRules;
    ;

    analysRules.clear();
    analysRules.initValue();
    analysRules.AnalysId = _line.AnalysId;
    analysRules.RefRecId = _line.RecId;
    analysRules.PotrbTypeRule_ZTR = _potrbRules.TypeRule;

```

```

        analysRules.ItemId = _potrbRules.ItemId;
        analysRules.ItemIdDuplicate = _potrbRules.ItemIdDuplicate;
        analysRules.ProdNum_ZTR = _potrbRules.ProdNum_ZTR;
        analysRules.Qty = _potrbRules.Qty;
        analysRules.QtyDuplicate = _potrbRules.QtyDuplicate;
        analysRules.WrkCtrGroupIdAR_ZTR = _potrbRules.WrkCtrGroupId;
        analysRules.PotrbRoute_ZTR = _potrbRules.PotrbRoute;
        analysRules.Koef = _potrbRules.Koef;
        analysRules.Notes = _potrbRules.Note;
        analysRules.NoteDate = _potrbRules.NoteDate;
        analysRules.CreatedByRules = _potrbRules.createdBy;
        analysRules.createdDateTimeRules = _potrbRules.createdDateTime;

        if (analysRules.validateWrite())
        {
            analysRules.insert();
        }
    }
    public server static void createAnalysDetailA(ProdNum_ZTR _prodNum, AnalysID
    _analysID)
    {
        NoYes          isFound;
        DataAreaId      dataAreaId = CompanyInfo::find().dataAreaId;

        UserConnection  con;
        Statement        stmt;
        str              sqlStr, tmpStr;
        ResultSet        myResult;
        AnalysDetail_ZTR analysDetail;
        int              startRunTmp, endRunTmp;
        AnalysProdNumLine_ZTR analysProdNumLine;
        PotrbRules_ZTR   potrbRules;

        ;

        tmpStr = ( @"
        SET NOCOUNT ON
        SET ANSI_WARNINGS OFF
        DECLARE @kol int, @num int, @zz as varchar(max)
        SELECT @kol = 1000
        SELECT @num = 1
        set @zz = '%1'
        IF OBJECT_ID('tempdb..#tmp_t1') IS NOT NULL
        drop table #tmp_t1
        --все списание на основное производство - журнал проводка
        select dimension7_, IT.ITEMID, IT.QTY, ID.INVENTBATCHID,
it.TRANSREFID, it.DateFinancial, id.INVENTLOCATIONID, @num as level1
        into #tmp_t1
        from INVENTTRANS IT
        join INVENTTRANSPOSTING ITP on IT.INVENTTRANSID =
ITP.INVENTTRANSID
                                and ITP.DATAAREAID = '%2'
        and ITP.INVENTTRANSPOSTINGTYPE = 1

```

```

and ITP.DIMENSION7_ = @zz
and ITP.ACCOUNTOFFSET like '231888%'

join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID and
ID.DATAAREAID = '%2'
and id.INVENTLOCATIONID in (select InventLocationId from InventLocation
where INPOTRB = 1)
where IT.DATAAREAID = '%2'
and IT.TRANSTYPE = 4
and IT.STORNO_ZTR = 0
and IT.QTY <> 0");

info(strfmt("%1",strlen(tmpstr)));
sqlStr=sqlStr+tmpStr;

tmpStr= ( @" -- Начинается ЦИКЛ
WHILE @kol > 0
BEGIN
    IF OBJECT_ID('tempdb..#tmp_t2') IS NOT NULL
drop table #tmp_t2
    ---Ищем Спецификацию или Производство
    select  t1.DIMENSION7_,  IT.TRANSREFID  as  firstTRANSREFID,
t1.inventbatchid, t1.itemid,
    ABS(cast(t1.qty as dec(26,12))/cast(IT.QTY as dec(26,12))) as koef, @num as
level1
    into #tmp_t2
    from INVENTTRANS IT
    left join INVENTJOURNALTRANS IJT on IT.INVENTTRANSID =
IJT.INVENTTRANSID
        and IJT.DATAAREAID = '%2'
        and IT.INVENTDIMID <> IJT.INVENTDIMID
join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID
and ID.DATAAREAID = '%2'
and ID.inventbatchid not in ('#Партия_НЗП', '#Партия_МДО', '#МГД')
and ID.inventbatchid in (select INVENTBATCHID from #tmp_t1)
join #tmp_t1 t1 on t1.INVENTBATCHID = ID.inventbatchid
and t1.itemid = it.itemid
    ");

sqlStr=sqlStr+tmpStr;

    tmpStr= (@"
    and t1.level1 = @num
    where IT.PackingSlipReturned = 0
    and ( (IT.TRANSTYPE = '2') or (IT.TRANSTYPE = '10' and
IT.INVENTDIMID <> IJT.INVENTDIMID) )

    delete from #tmp_t1 where inventbatchid in (select INVENTBATCHID from
#tmp_t2) -- удаление того, что раскрываем

    ---Ищем строки Спецификации или строки Производства
    insert into #tmp_t1

```

```

        select    dimension7_,    IT.itemid,    IT.qty*#tmp_t2.koef    as    qty,
ID.INVENTBATCHID, it.TRANSREFID, it.DateFinancial,
        id.INVENTLOCATIONID, @num+1 as level1
    from #tmp_t2
    join INVENTTRANS IT on IT.TRANSREFID = #tmp_t2.firstTRANSREFID
        and IT.TRANSTYPE in ('8','9') ---Ищем строки Спецификации или
строки Производства
        and IT.DATAAREAID = '%2'
    join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID and
ID.DATAAREAID = '%2'
    where IT.qty*#tmp_t2.koef <> 0
    and IT.PackingSlipReturned = 0
    SELECT @kol = (select count(*) from #tmp_t2)
    SELECT @num = @num + 1
END-- WHILE
select LINE.RECID, * from #tmp_t1
join AnalysProdNumLine_ZTR AS LINE
on #tmp_t1.ItemId = LINE.ItemId
SET NOCOUNT OFF
SET ANSI_WARNINGS ON
");

    sqlStr=sqlStr+tmpStr;

    sqlStr = strfmt(sqlStr,_prodNum, dataAreaId);
    con      = new UserConnection();
    stmt     = con.createStatement();
    myResult = callSqlOnServer_ZTR(stmt, sqlStr, false);

    while (myResult.next())
    {
        analysDetail.clear();
        analysDetail.initValue();

        analysDetail.RefRecId      = myResult.getInt64(1);
        analysDetail.AnalysId      = _analysID;
        analysDetail.DetailType    = DetailType::TakenOff;
        analysDetail.ItemId        = myResult.getString(3);
        analysDetail.Qty           = myResult.getreal( 4);
        analysDetail.InventBatchId = myResult.getstring( 5);
        analysDetail.JournalId     = myResult.getstring( 6);

        //analysDetail.DetailType    = myResult.getReal( 3);
        analysDetail.DateFinancial  = myResult.getDate( 7);
        analysDetail.InventLocationId = myResult.getstring( 8);
        //analysDetail.DateFinancial = myResult.getDateTime( 6);

        analysDetail.Level1        = myResult.getstring( 9);

        if(analysDetail.validateWrite())
        {

```

```

        analysDetail.Insert();

    }
    else
    {
        throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
    }
    stmt.close();
}
while select analysProdNumLine
join potrbRules
where analysProdNumLine.ItemId == potrbRules.ItemIdDuplicate
&& potrbRules.PotrbRoute == PotrbRoute_ZTR::BothSides
&& potrbRules.Actual == NoYes::Yes
&& potrbRules.TypeRule == PotrbTypeRule_ZTR::Duplicate
{
    analysDetail.clear();
    analysDetail.initValue();
    analysDetail.AnalysId = analysProdNumLine.AnalysId;
    analysDetail.ItemId = analysProdNumLine.ItemId;
    analysDetail.RefRecId = analysProdNumLine.RecId;

    if (analysDetail.validateWrite())
    {
        analysDetail.insert();
    }
    else
    {
        throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
    }
    isFound = true;
}
//
}

public server static void createAnalysDetailB(ProdNum_ZTR _prodNum, AnalysID
_analysID)
{
    NoYes        isFound;
    DataAreaId    dataAreaId = CompanyInfo::find().dataAreaId;

    UserConnection    con ;
    Statement          stmt ;
    str                sqlStr, tmpStr;
    ResultSet          myResult;
    AnalysDetail_ZTR    analysDetail;
    int                startRunTmp, endRunTmp;
    AnalysProdNumLine_ZTR    analysProdNumLine;
    PotrbRules_ZTR      potrbRules;

```

```

;
// Б. Заказано
tmpStr = ( @"
SET NOCOUNT ON
SET ANSI_WARNINGS OFF
DECLARE @kol int, @num int, @zz as varchar(max)
SELECT @kol = 1000
SELECT @num = 1
set @zz = '%1'
IF OBJECT_ID('tempdb..#tmp_t1') IS NOT NULL
    drop table #tmp_t1

--все списание на основное производство - журнал проводка (статус-заказано)
select dimension7_, IT.ITEMID, IT.QTY, ID.INVENTBATCHID, it.TRANSREFID,
it.DateFinancial, id.INVENTLOCATIONID, @num as level1
into #tmp_t1
from INVENTTRANS IT
join inventjournaltrans ijt on ijt.inventtransid = it.inventtransid and
ijt.DIMENSION7_ = @zz
join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID and
ID.DATAAREAID = '%2'
and id.INVENTLOCATIONID in (select InventLocationId from InventLocation
where INPOTRB = 1)
where IT.DATAAREAID = '%2'
and it.StatusIssue = 6
and IT.TRANSTYPE = 4
and IT.STORNO_ZTR = 0
and IT.QTY <> 0");

sqlStr=sqlStr+tmpStr;

tmpStr= ( @"-- Начинается ЦИКЛ
WHILE @kol > 0
BEGIN
    IF OBJECT_ID('tempdb..#tmp_t2') IS NOT NULL
        drop table #tmp_t2
    ---Ищем Спецификацию или Производство
    select t1.DIMENSION7_, IT.TRANSREFID as firstTRANSREFID,
t1.inventbatchid, t1.itemid,
ABS(cast(t1.qty as dec(26,12))/cast(IT.QTY as dec(26,12))) as koef,
@num as level1
into #tmp_t2
from INVENTTRANS IT
left join INVENTJOURNALTRANS IJT on IT.INVENTTRANSID =
IJT.INVENTTRANSID
and IJT.DATAAREAID = '%2'
and IT.INVENTDIMID <> IJT.INVENTDIMID
join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID
and ID.DATAAREAID = '%2'
and ID.inventbatchid not in ('#Партия_НЗП', '#Партия_МДО', '#МГД')
and ID.inventbatchid in (select INVENTBATCHID from #tmp_t1)

```

```

join #tmp_t1 t1 on t1.INVENTBATCHID = ID.inventbatchid
      and t1.itemid = it.itemid ");

      sqlStr=sqlStr+tmpStr;

      tmpStr= (@"
      and t1.level1 = @num
      where IT.PackingSlipReturned = 0
      and ((IT.TRANSTYPE = '2') or
      (IT.TRANSTYPE = '10' and IT.INVENTDIMID <> IJT.INVENTDIMID))

      delete from #tmp_t1 where inventbatchid in (select INVENTBATCHID from
      #tmp_t2) -- удаление того, что раскрываем

      ---Ищем строки Спецификации или строки Производства
      insert into #tmp_t1
      select      dimension7_,      IT.itemid,      IT.qty*#tmp_t2.koef      as      qty,
      ID.INVENTBATCHID, it.TRANSREFID, it.DateFinancial,
      id.INVENTLOCATIONID, @num+1 as level1
      from #tmp_t2
      join INVENTTRANS IT on IT.TRANSREFID = #tmp_t2.firstTRANSREFID
      and IT.TRANSTYPE in ('8','9') ---Ищем строки Спецификации или строки
      Производства
      and IT.DATAAREAID = '%2'
      join INVENTDIM ID on IT.INVENTDIMID = ID.INVENTDIMID and
      ID.DATAAREAID = '%2'
      where IT.qty*#tmp_t2.koef <> 0
      and IT.PackingSlipReturned = 0

      SELECT @kol = (select count(*) from #tmp_t2)
      SELECT @num = @num + 1
      END-- WHILE
      select LINE.RECID, * from #tmp_t1
      join AnalysProdNumLine_ZTR AS LINE
      on #tmp_t1.ItemId = LINE.ItemId
      SET NOCOUNT OFF
      SET ANSI_WARNINGS ON
      ");
      sqlStr=sqlStr+tmpStr;

      sqlStr = strfmt(sqlStr,_prodNum, dataAreaId);
      con      = new UserConnection();
      stmt      = con.createStatement();
      myResult = callSqlOnServer_ZTR(stmt, sqlStr, false);

      while (myResult.next())
      {
      analysDetail.clear();
      analysDetail.initValue();

      analysDetail.RefRecId      = myResult.getInt64(1);
      analysDetail.AnalysId      = _analysID;

```

```

        analysDetail.DetailType      = DetailType::Ordered;
        analysDetail.ItemId          = myResult.getString(3);
        analysDetail.Qty             = myResult.getreal( 4);
        analysDetail.InventBatchId    = myResult.getstring( 5);
        analysDetail.JournalId        = myResult.getstring( 6);

        //analysDetail.DetailType      = myResult.getReal( 3);
        analysDetail.DateFinancial    = myResult.getDate( 7);
        analysDetail.InventLocationId = myResult.getstring( 8);
        //analysDetail.DateFinancial    = myResult.getDateTime( 6);

        analysDetail.Level1           = myResult.getstring( 9);

        if(analysDetail.validateWrite())
        {
            analysDetail.Insert();
        }
        else
        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
        }
        stmt.close();
    }

    while select analysProdNumLine
    join potrbRules
    where analysProdNumLine.ItemId == potrbRules.ItemIdDuplicate
    && potrbRules.PotrbRoute == PotrbRoute_ZTR::BothSides
    && potrbRules.Actual == NoYes::Yes
    && potrbRules.TypeRule == PotrbTypeRule_ZTR::Duplicate
    {
        analysDetail.clear();
        analysDetail.initValue();
        analysDetail.AnalysId = analysProdNumLine.AnalysId;
        analysDetail.ItemId = analysProdNumLine.ItemId;
        analysDetail.RefRecId = analysProdNumLine.RecId;

        if (analysDetail.validateWrite())
        {
            analysDetail.insert();
        }
        else
        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
        }
        isFound = true;
    }
}

```

```

public void createAnalysDetailC()
{
    InventTrans      inventTrans;
    AnalysDetail_ZTR  analysDetailNew, analysDetailCopy;
    InventSum         inventSum;
    ;
    while select  analysProdNumLine
        group by analysProdNumLine.RecId,
                inventSum.ItemId,
                inventDim.InventLocationId,
                inventDim.InventGtdId_RU
    where analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join sum(AvailPhysical) from inventSum
    where inventSum.ItemId == analysProdNumLine.ItemId
    join inventDim
    where inventSum.InventDimId == inventDim.InventDimId
    join analysProdNumTable
    where analysProdNumLine.AnalysId == analysProdNumTable.AnalysId
        && inventDim.InventGtdId_RU == analysProdNumTable.ProdNum
    exists join inventLoc
    where inventDim.InventLocationId == inventLoc.InventLocationId
        && inventLoc.WrkCtrGroupId_ZTR == analysProdNumTable.WrkCtrGroupId
    {
        analysDetail.clear();
        analysDetail.initValue();
        analysDetail.AnalysId
AnalysProdNumLine_ZTR::findRecId(analysProdNumLine.RecId).AnalysId;
        analysDetail.RefRecId = analysProdNumLine.RecId;
        analysDetail.InventLocationId = inventDim.InventLocationId;
        analysDetail.ProdNum_ZTR = inventDim.InventGtdId_RU;
        analysDetail.ItemId = inventSum.ItemId;
        analysDetail.Qty = inventSum.AvailPhysical;
        if (analysDetail.validateWrite())
        {
            analysDetail.insert();
        }
        else
        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
        }
    }
}

public void createAnalysDetailD()
{
    InventTrans      inventTrans;
    AnalysDetail_ZTR  analysDetailNew, analysDetailCopy;
    InventSum         inventSum;
    ;
    while select  analysProdNumLine
        group by analysProdNumLine.RecId,
                inventSum.ItemId,

```

```

        inventDim.InventLocationId,
        inventDim.InventGtdId_RU
    where analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join sum(AvailPhysical) from inventSum
        where inventSum.ItemId == analysProdNumLine.ItemId
    join inventDim
        where inventSum.InventDimId == inventDim.inventDimId
    join analysProdNumTable
        where analysProdNumLine.AnalysId == analysProdNumTable.AnalysId
    exists join inventLoc
        where inventDim.InventLocationId == inventLoc.InventLocationId
            && inventLoc.WrkCtrGroupId_ZTR ==
analysProdNumTable.WrkCtrGroupId
    {
        analysDetail.clear();
        analysDetail.initValue();
        analysDetail.AnalysId =
AnalysProdNumLine_ZTR::findRecId(analysProdNumLine.RecId).AnalysId;
        analysDetail.RefRecId = analysProdNumLine.RecId;
        analysDetail.DetailType = DetailType::RemainswoPNSite;
        analysDetail.InventLocationId = inventDim.InventLocationId;
        analysDetail.ProdNum_ZTR = inventDim.InventGtdId_RU;
        analysDetail.ItemId = inventSum.ItemId;
        analysDetail.Qty = inventSum.AvailPhysical;
        if (analysDetail.validateWrite())
        {
            analysDetail.insert();
        }
        else
        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
        }
        isFound = true;
    }
    while select analysProdNumLine
    join potrbRules
        where analysProdNumLine.ItemId == potrbRules.ItemIdDuplicate
            && potrbRules.PotrbRoute == PotrbRoute_ZTR::BothSides
            && potrbRules.Actual == NoYes::Yes
            && potrbRules.TypeRule == PotrbTypeRule_ZTR::Duplicate
        {
            analysDetail.clear();
            analysDetail.initValue();
            analysDetail.AnalysId = analysProdNumLine.AnalysId;
            analysDetail.ItemId = analysProdNumLine.ItemId;
            analysDetail.RefRecId = analysProdNumLine.RecId;
            if (analysDetail.validateWrite())
            {
                analysDetail.insert();
            }
            else

```

```

        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
        }
        isFound = true;
    }
}

public void createAnalysDetailE()
{
    InventTrans      inventTrans;
    AnalysDetail_ZTR analysDetailNew, analysDetailCopy;
    InventSum        inventSum;

;

    while select analysProdNumLine
    group by analysProdNumLine.RecId,
            inventSum.ItemId,
            inventDim.InventLocationId,
            inventDim.InventGtdId_RU
    where analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join sum(AvailPhysical) from inventSum
    where inventSum.ItemId == analysProdNumLine.ItemId
    join inventDim
    where inventSum.InventDimId == inventDim.inventDimId
        && inventDim.InventGtdId_RU ==
    join analysProdNumTable
    where analysProdNumLine.AnalysId == analysProdNumTable.AnalysId
    exists join inventLoc
    where inventDim.InventLocationId == inventLoc.InventLocationId
        && inventLoc.WrkCtrGroupId_ZTR == analysProdNumTable.WrkCtrGroupId
    {

        analysDetail.clear();
        analysDetail.initValue();
        analysDetail.AnalysId
AnalysProdNumLine_ZTR::findRecId(analysProdNumLine.RecId).AnalysId;
        analysDetail.RefRecId = analysProdNumLine.RecId;
        analysDetail.DetailType = DetailType::RemainsAll;
        analysDetail.InventLocationId = inventDim.InventLocationId;
        analysDetail.ProdNum_ZTR = inventDim.InventGtdId_RU;
        analysDetail.ItemId = inventSum.ItemId;
        analysDetail.Qty = inventSum.AvailPhysical;
        if (analysDetail.validateWrite())
        {
            analysDetail.insert();
        }
        else
        {
            throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));

```

```

    }
    isFound = true;
}

while select analysProdNumLine
join potrbRules
where analysProdNumLine.ItemId == potrbRules.ItemIdDuplicate
&& potrbRules.PotrbRoute == PotrbRoute_ZTR::BothSides
&& potrbRules.Actual == NoYes::Yes
&& potrbRules.TypeRule == PotrbTypeRule_ZTR::Duplicate
{
    analysDetail.clear();
    analysDetail.initValue();
    analysDetail.AnalysId = analysProdNumLine.AnalysId;
    analysDetail.ItemId = analysProdNumLine.ItemId;
    analysDetail.RefRecId = analysProdNumLine.RecId;

    if (analysDetail.validateWrite())
    {
        analysDetail.insert();
    }
    else
    {
        throw error(strfmt("Record cannot be inserted into %1",
tablename(AnalysDetail_ZTR)));
    }
    isFound = true;
}
}

public void fillinAnalysLine()
{
    AnalysDetail_ZTR analysDetail0, analysDetail1, analysDetail2,
    analysDetail3, analysDetail4 ;
    AnalysRules_ZTR analysRules1, analysRules5;
    ;
    while select analysProdNumLine
    where analysProdNumLine.AnalysId == analysProdNumTable.AnalysId
    && analysProdNumLine.AnalysType == PotrbType_ZTR::YesRules
    join analysRules1
    where analysRules1.AnalysId == analysProdNumTable.AnalysId
    && analysRules1.RefRecId == analysProdNumLine.RecId
    && analysRules1.PotrbTypeRule_ZTR == PotrbTypeRule_ZTR::Duplicate
    join analysRules5
    where analysRules5.AnalysId == analysProdNumTable.AnalysId
    && analysRules5.RefRecId == analysProdNumLine.RecId
    && analysRules5.PotrbTypeRule_ZTR ==
PotrbTypeRule_ZTR::PermissionCard
    join sum(Qty) from analysDetail0
    where analysDetail0.DetailType == DetailType::TakenOff
    join sum(Qty) from analysDetail1
    where analysDetail1.DetailType == DetailType::Ordered

```

```

join sum(Qty) from analysDetail2
where analysDetail2.DetailType == DetailType::RemainsPNSire
join sum(Qty) from analysDetail3
where analysDetail3.DetailType == DetailType::RemainswoPNSite
join sum(Qty) from analysDetail4
where analysDetail4.DetailType == DetailType::RemainsAll
{

    analysProdNumLine.clear();
    analysProdNumLine.initValue();
    analysProdNumLine.TableReplacement = NoYes::Yes;
    analysProdNumLine.Karta = NoYes::Yes;
    analysProdNumLine.SpisQty = analysDetail0.Qty;
    analysProdNumLine.OrderQty = analysDetail1.Qty;
    analysProdNumLine.OstZzQty = analysDetail2.Qty;
    analysProdNumLine.OstNoZzQty = analysDetail3.Qty;
    analysProdNumLine.OstAll = analysDetail4.Qty;
    analysProdNumLine.DefQty          =          analysProdNumLine.Qty      -
(analysProdNumLine.SpisQty          +          analysProdNumLine.OrderQty  +
analysProdNumLine.OstZzQty          +          analysProdNumLine.OstNoZzQty +
analysProdNumLine.OstAll);

    if (analysProdNumLine.validateWrite())
    {
        analysProdNumLine.insert();
    }
    else
    {
        throw error(strfmt("Record cannot be inserted into %1",
tablepname(AnalysDetail_ZTR)));
    }
}
}

```

ДОДАТОК Б
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

ДИПЛОМНА РОБОТА БАКАЛАВРА

ТЕМА РОБОТИ: ПРОГРАМНА РЕАЛІЗАЦІЯ ФУНКЦІОНАЛУ «АНАЛІЗ ЗАВОДСЬКОГО ЗАМОВЛЕННЯ»

Виконала студентка групи КНТ-137

Сліпченко В. О.

Керівник Зеленьова І.Я.

Рисунок Б.1 - Тема роботи

Мета роботи, об'єкт та предмет

- Об'єкт дослідження – новий функціонал в ERP - системі для пошуку аналізу заводського замовлення.
- Предмет дослідження – програмні способи реалізації аналізу заводського замовлення.
- Мета роботи – створення функціоналу для пошуку аналізу заводського замовлення.

Рисунок Б.2- Мета роботи

Порівняння ERP - систем

Рішення	Виробник	Сфера застосування	Термін впровадження	Вартість впровадження
SAP R/3	SAP AG (Германия)	Оборонні підприємства, компанії нафтогазового комплексу, металургія, енергетика телекомунікації, банківський сектор.	1-5 років і більше	Ліцензія на 50 робочих місць коштує близько \$ 350 тис. Вартість впровадження може в кілька разів перевищувати вартість рішення.
Oracle Applications	Oracle (США)	Важка промисловість (переважно металургія), телекомунікаційні компанії, фінансовий сектор, хімічна промисловість.	1-5 років і більше	Вартість рішення на одне робоче місце становить близько \$ 5 тис. Повна вартість істотно залежить від необхідної функціональності і складності впровадження.
IFS Application	IFS (Швеція)	Підприємства машинобудівного комплексу, енергетика, харчова промисловість, фармацевтика, кабельна промисловість.	0,8-3 роки і більше	Повна вартість впровадження може досягати \$ 250 тис. І більше. Також істотно залежить від необхідної функціональності.
Baan ERP	SSA Global (США)	Автомобілебудування, хімічна промисловість, фармацевтика, харчова промисловість.	6 міс-1,5 року і більше	Вартість одного робочого місця - \$ 3 тис. Співвідношення ціни рішення і витрат на впровадження 1: 1-1: 3.
iRenaissance	Ross Systems (США)	Харчова промисловість, хімічні компанії, металургійна промисловість нафтопереробні, целюлозно-паперові, фармацевтичні підприємства.	4 міс. - 1,3 року і більше	Вартість впровадження в середньому \$ 200 тис.
MBS Axapta, Navision	Microsoft (США)	Підприємства нафтової галузі, харчової промисловості, торгівлі компанії, металургія, дистрибуція, телекомунікаційна галузь.	6 міс. - 2 роки і більше	В середньому вартість рішення на одне робоче місце - \$ 3,5 тис. Вартість впровадження становить 100-250% вартості рішення.

Рисунок Б.3 – Порівняння ERP-систем

Функції програми

- Знаходити потрібні номенклатури та проводити аналіз заводського замовлення.
- Сортувати потрібні номенклатури за правилами(Карта розрішення та Таблиця взаємозамінності).
- Створювати журнал списань.
- Для активної номенклатури форми відкривати не редаговану обрану деталізацію за правилами (Карта розрішення, Таблиця взаємозамінності).
- Для активної номенклатури форми відкривати не редаговану обрану деталізацію (Списано, Замовлено, Залишки 3 / 3, цех, Залишки без 3 / 3, цех, Залишки всі).

Рисунок Б.4 – Функції програми

Класи програми

<u>AnalysProdNumTableCreate_ZTR</u>	<u>AnalysProdNumJournalCreate_ZTR</u>	<u>AnalysPotrbRules_ZTR</u>	<u>AnalysDetailCreate_ZTR</u>
➤ Створення записів в шапці(таблиці) форми ➤ Виконати розрахунок потреби ➤ Створити в таблиці «Рядки» записи ➤ Створити в таблиці «Рядки» записи, застосувавши ПРАВИЛА(Карта розрешень, Таблиця взаємозамінності) ➤ Створити в таблиці «Детально» записи: Списано, Замовлено, Залишки 3/3, цех, Залишки без 3/3, цех, Залишки всі.	➤ Створити журнал з типом «Проводка» (таблиця InventJournalTable) ➤ Створити рядки журналу (таблиця InventJournalTrans)	➤ Для активної номенклатури форми відкрити не <u>редаговане</u> правило «Карта розрешень» ➤ Для активної номенклатури форми відкрити не <u>редаговане</u> правило «Таблиця взаємозамінності»	➤ Для активної номенклатури форми відкрити не <u>редаговану</u> деталізацію за <u>вибором</u>: • Списано • Замовлено • Залишки 3/3, цех • Залишки без 3/3, цех • Залишки всі

5

Рисунок Б.5 - Класи програми

Розподіл доступу до функціоналу застосунку

6

Рисунок Б.6 – Розподіл доступу до функціоналу застосунку

База даних

AnalysProdNumTable_ZTR

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	<u>AnalysId</u>	EDT: <u>AnalysId</u>	Обов'язкове
Номер пр. замовлення по ЗТР	<u>ProdNum</u>	EDT: <u>ProdNum_ZTR</u>	Обов'язкове
Робочий центр	<u>WrkCtrGroupId</u>	EDT: <u>WrkCtrGroupId</u>	Обов'язкове
Спільний вид/вузел	<u>ItemIdBom</u>	EDT: <u>ItemIdBom_ZTR</u>	Обов'язкове
Код схеми	<u>SchemeCode</u>	EDT: <u>BOMSchemeCode_ZTR</u>	Обов'язкове
Примітка	<u>Note</u>	Type: String String size: 500	

AnalysProdNumLine_ZTR

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	<u>AnalysId</u>	EDT: <u>AnalysId</u>	Обов'язкове
Тип аналізу 3/3	<u>AnalysType</u>	Enum Type: <u>PotrType</u>	0 – без правил, 1 – правила застосовані
Код номенклатури	<u>ItemId</u>	EDT: <u>ItemId</u>	Обов'язкове
Кількість	<u>Qty</u>	EDT: <u>BOMQty</u>	
Таблиця взаємозамінності	<u>TableReplacement</u>	Enum Type: <u>NoYes</u>	
Карта розширення	<u>Karta</u>	Enum Type: <u>NoYes</u>	
Списано	<u>SpisQty</u>	EDT: <u>BOMQty</u>	
Замовлено	<u>OrderQty</u>	EDT: <u>BOMQty</u>	
Залишки 3/3, цех	<u>OstZzQty</u>	EDT: <u>BOMQty</u>	
Залишки без 3/3, цех	<u>OstNoZzQty</u>	EDT: <u>BOMQty</u>	
Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Залишки всі	<u>OstAll</u>	EDT: <u>BOMQty</u>	
Дефіцит	<u>DefQty</u>	EDT: <u>BOMQty</u>	
Примітка	<u>Note</u>	Type: String String size: 500	

Рисунок Б.7 – База даних

База даних

AnalysRules_ZTR

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	<u>AnalysId</u>	EDT: <u>AnalysId</u>	Обов'язкове
Код посилання на запис	<u>RefRecId</u>	EDT: <u>RefRecId</u>	Обов'язкове
Тип даних	<u>TypeRule</u>	Enum Type: <u>PotrTypeRule</u>	1 – таблиця взаємозамінності 5 – карта розширення
Номенклатура	<u>ItemId</u>	EDT: <u>ItemId</u>	
Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Номенклатура-дублікат	<u>ItemIdDuplicate</u>	EDT: <u>ItemId</u>	
Номер пр. заказа по ЗТР	<u>ProdNum_ZTR</u>	EDT: <u>ProdNum_ZTR</u>	
Кількість	<u>Qty</u>	EDT: <u>Qty</u>	
Кількість-дублікат	<u>QtyDuplicate</u>	EDT: <u>Qty</u>	
Цех	<u>WrkCtrGroupId</u>	EDT: <u>WrkCtrGroupId</u>	
Направлення	<u>PotrRoute</u>	Enum Type: <u>PotrRoute_ZTR</u>	1 – в обидві сторони 2 – в одну сторону
Коефіцієнт	<u>Koef</u>	Type: Numeric	
Примітка	<u>Note</u>	Type: String String size: 500	
Дата документу	<u>NoteDate</u>	Type: Date	
Автор створення правила	<u>CreatedByRules</u>	CreatedBy	
Дата створення правила	<u>CreatedDateTimeRules</u>	<u>CreatedDateTime</u>	

AnalysDetail_ZTR

Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Код аналізу 3/3	<u>AnalysId</u>	EDT: <u>AnalysId</u>	Обов'язкове
Код посилання на запис	<u>RefRecId</u>	EDT: <u>RefRecId</u>	Обов'язкове
Тип деталізації	<u>DetailType</u>	Enum Type: <u>DetailType</u> , <u>create</u>	0 – списано, 1 – замовлено, 2 – залишки 3/3, цех, 3 – залишки без 3/3, цех, 4 – залишки все
Журнал	<u>JournalId</u>	EDT: <u>InventJournalId</u>	
Назва поля (RU)	Назва поля (ENG)	Тип	Примітка
Номер пр. заказа по ЗТР	<u>ProdNum</u>	EDT: <u>ProdNum_ZTR</u>	
Цех	<u>InventLocationId</u>	EDT: <u>InventLocationId</u>	
Код номенклатури	<u>ItemId</u>	EDT: <u>ItemId</u>	
Дата	<u>DateFinancial</u>	EDT: <u>DateFinancial</u>	
Кількість	<u>Qty</u>	EDT: <u>BOMQty</u>	
Партія	<u>InventBatchId</u>	EDT: <u>InventBatchId</u>	
Рівень	<u>Level1</u>	Type: Numeric (целое)	

Рисунок Б.8 – База даних

База даних

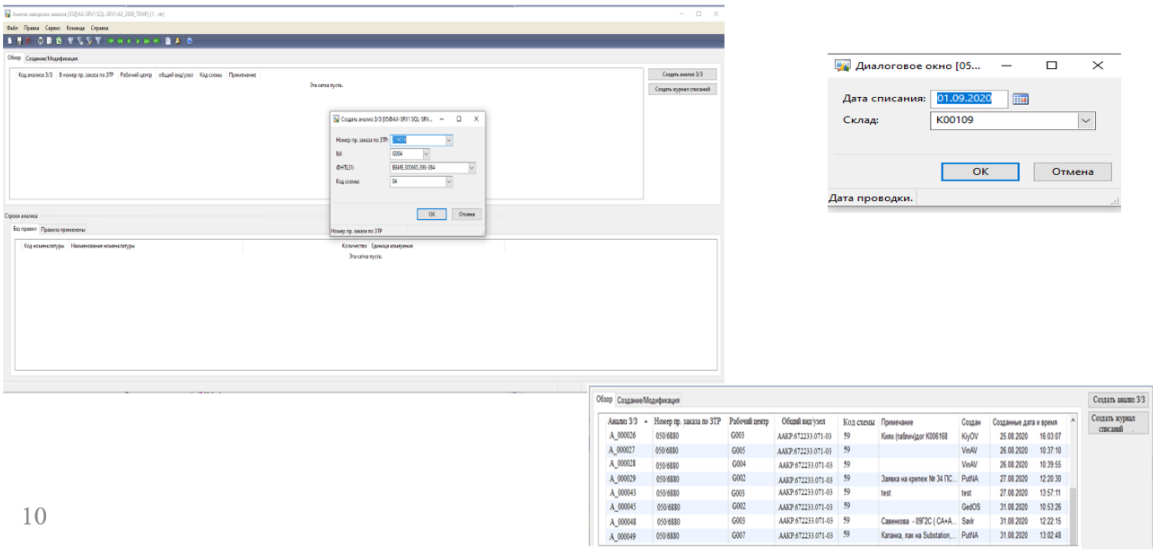
Опис зв'язків таблиць

Таблиця 1	Таблиця 2
AnalysProdNumTable_ZTR.AnalysId	AnalysProdNumLine_ZTR.AnalysId
AnalysDetail_ZTR.RefRecId	AnalysProdNumLine_ZTR.RecId
AnalysRules_ZTR.RefRecId	<u>AnalysProdNumLine_ZTR.RecId</u>
<u>InventLocation.WrkCtrGroupId_ZTR</u>	WrkCtrTable.WrkCtrGroupId
AnalysProdNumLine_ZTR.ItemId	InventTable.ItemId
AnalysDetail_ZTR.ItemId	InventTable.ItemId
InventTable.ItemId	InventTableModule.ItemId
InventTrans.InventDimId	InventDim.InventDimId
InventSum.InventDimId	<u>InventDim.InventDimId</u>

9

Рисунок Б.9 – База даних

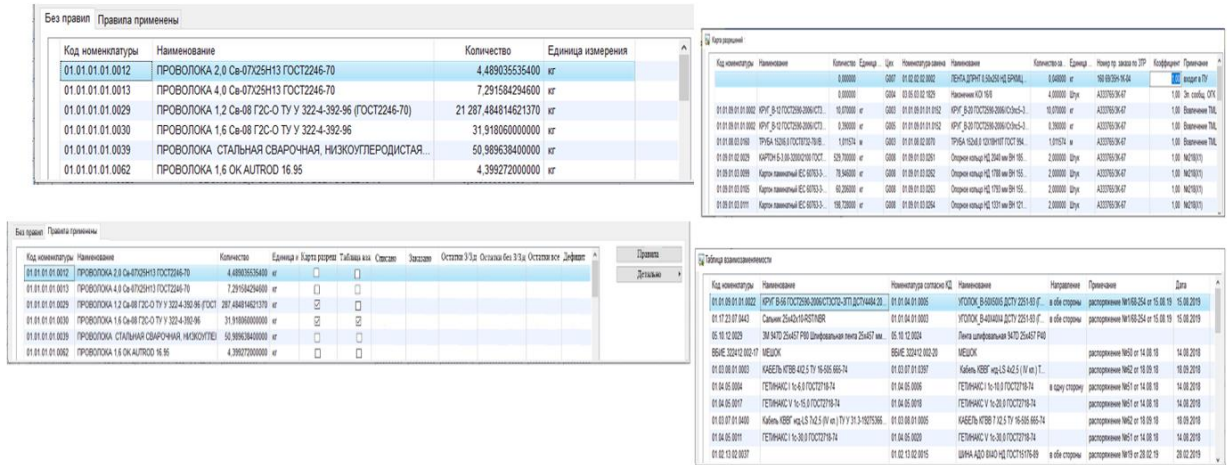
Інтерфейс застосунку



10

Рисунок Б.10 – Интерфейс застосунку

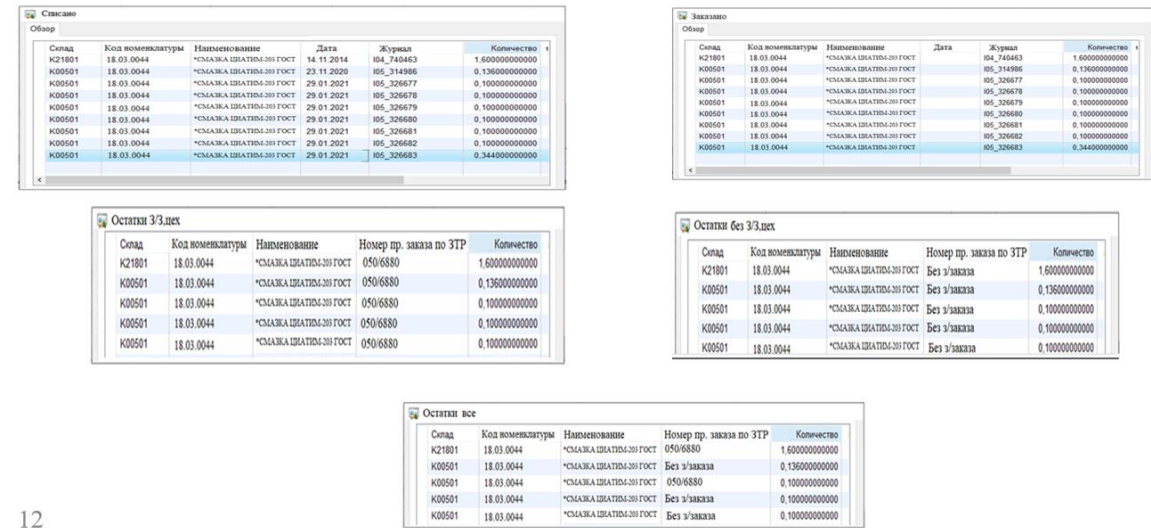
Интерфейс застосунку



11

Рисунок Б.11 – Интерфейс застосунку

Интерфейс застосунку



12

Рисунок Б.12 – Интерфейс застосунку

Висновки

- Головною метою написання даного функціоналу було оптимізувати витрати робочого часу **робітників** підприємства, адже, раніше даний аналіз виконувався вручну, цей процес є дуже трудомістким. Тепер даний аналіз проводиться автоматично.
- Була створена форма, яка дозволяє зручно знаходити потрібний аналіз необхідних для списання під виробництво замовлень ТМЦ (товарно-матеріальних цінностей).
- Порівнюючи даний процес з роботою вручну, **витрачається набагато менше часу**, адже даний аналіз виконується не більше 10 хвилин. В ході тестування **критичних помилок не було виявлено**.
- **Основним недоліком програми** є час виконання поставленої задачі (хоч це всеодно швидше, ніж виконувати вручну), адже задачею програми є обробка дуже великого об'єму даних. Проте даний програмний продукт **повністю виконує поставлені користувачем задачі та може бути використаний на виробництві**.
- Даний функціонал можуть використовувати тільки працівники(консультанти) підприємств, які працюють з ERP – системою, адже дана програма реалізована саме для цього середовища.
- Продукт можна вважати оптимізованим та «легким» для використання у роботі консультантами підприємств.

13

Рисунок Б.13 – Висновки

Дякую за увагу!

14

Рисунок Б.14 – Подяка за увагу