

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

(повне найменування закладу вищої освіти)

Факультет математики, фізики та інформаційних технологій

(повне найменування факультету)

Кафедра інформаційних технологій

(повна назва кафедри)

Кваліфікаційна робота

на здобуття ступеня вищої освіти «Бакалавр»

**«Створення інформаційної бази для автоматизації процесів
та підвищення ефективності роботи підприємства»**

(тема кваліфікаційної роботи українською мовою)

**«Creating an Information Database for Automating Processes
and Increasing Enterprise Efficiency»**

(тема кваліфікаційної роботи англійською мовою)

Виконав: здобувач денної форми навчання
спеціальності 122 Комп'ютерні науки

(код, назва спеціальності)

Освітня програма Комп'ютерні науки

(назва)

Голюк Максим Ігорович

(прізвище, ім'я, по-батькові здобувача)

Керівник асистент Молчанова А.Ю.

(науковий ступінь, вчене звання, прізвище, ініціали)


(підпис)

Рецензент д.т.н., доцент Соколов А.В.

(науковий ступінь, вчене звання, прізвище, ініціали)

Рекомендовано до захисту:
Протокол засідання кафедри
Інформаційних технологій

№ 1 від 09 червня 2024 р.

Завідувачка кафедри


(підпис) КАЗАКОВА Надія
(прізвище, ім'я)

Захищено на засіданні ЕК № 13,
протокол № 15 від 20 червня 2024 р.

Оцінка задовільно / Е / 60
(за національною шкалою/шкалою ECTS/ бали)

Голова ЕК


(підпис) КОПИЧЕНКО Іван
(прізвище, ім'я)

Одеса 2024

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ	8
1.1 Актуальність теми	8
1.2 Мета та завдання дослідження	9
1.3 Етапи реалізації проекту	10
1.4 Оцінка ефективності	10
1.5 Аналіз вимог	11
1.6 Технічні аспекти	12
1.7 Ризики та їх мінімізація	12
1.8 Оцінка вартості та термінів	13
2 ЗАГАЛЬНИЙ ВИГЛЯД КОНФІГУРАЦІЇ	14
2.1 Мова програмування	14
2.2 Система захисту	14
2.3 Початковий інтерфейс	16
2.4 Порівняння з існуючими програмами	17
3 ПРОЕКТУВАННЯ СТРУКТУРИ	21
3.1 Проектування бази для компанії	21
3.2 Об'єкти конфігурації та їх призначення	22
3.3 Побудова структури на UML діаграмах	26
4 РЕАЛІЗАЦІЯ ОБ'ЄКТІВ КОНФІГУРАЦІЇ	29
4.1 Довідники	29
4.2 Документи	36
4.3 Перерахування	45
4.4 Звіти	47
4.5 Плани видів характеристик	50
4.6 Плани рахунків	53

4.7 Плани видів розрахунку та регістри розрахунку	55
4.8 Регістри розрахунків	57
4.9 Регістри накопичень	59
4.10 Регістри бухгалтерії	60
4.11 Регістри відомостей	62
4.12 План обміну	64
ВИСНОВКИ	66
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	67

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

РН – Регістр накопичень

РВ – Регістр відомостей

РБ – Регістр бухгалтерії

ПВХ – План видів характеристик

СКД – Система компонування даних

BAF - Business Application Framework

ERP - Enterprise Resource Planning

CRM - Customer Relationship Management

WMS - Warehouse Management System

API - Application Programming Interface

DBMS – Database Management System

UI/UX - User Interface/User Experience

SLA - Service Level Agreement

GDPR - General Data Protection Regulation

SaaS - Software as a Service

ВСТУП

В Україні дуже багато різноманітних бізнесів, і навіть під час війни вони продовжують працювати. Але все ще є підприємства, які працюють у Екселі, записуючи туди усі обробки заказів, перевезення, накладні і тд. Або гірше того – записують все це на блокнотах. І це продовжується в 21 столітті, коли є BAF [1]. BAF складається з технологічної платформи і розроблених на її основі прикладних рішень («конфігурацій»). Таке планування системи принесла їй високу затребуваність, оскільки гарантує відкритість прикладних рішень, їх функціональність і гнучкість, короткі терміни впровадження, високу продуктивність, масштабованість від одного до десятків тисяч робочих місць.

Мета даної дипломної роботи – розробка конфігурації на платформі BAF, призначеної для автоматизації ключових бізнес-процесів та підвищення ефективності роботи компанії, що спеціалізується на продажі телефонів, чохлів, аксесуарів, гаджетів та надання супутніх послуг. Для досягнення поставленої мети були сформульовані наступні завдання::

- визначення вимог до конфігурації
- розробка та впровадження конфігурації на платформі BAF
- тестування та оцінка ефективності впровадженої системи
- функціональні вимоги
- нефункціональні вимоги
- створення структури бази даних
- розробка модулів та інтерфейсів для користувачів
- інтеграція з існуючими системами

Структура кваліфікаційної роботи бакалавра складається з вступу, 4 розділів, висновків, переліку посилань на 10 найменувань. Повний обсяг проекту становить 67 сторінок, містить 72 рисунка і 1 таблицю

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ІСНУЮЧИХ ПРОГРАМНИХ СИСТЕМ

1.1 Актуальність теми

Тема ВАР є актуальною з кількох причин:

- Динамічні зміни в бізнес-середовищі: Сучасні компанії стикаються з постійними змінами на ринку, вимогами клієнтів та технологічними інноваціями. ВАР дозволяє компаніям більш ефективно адаптуватися до цих змін, структуруючи та систематизуючи аналіз бізнесу.

- Підвищення ефективності процесів: За допомогою ВАР компанії можуть покращити внутрішні бізнес-процеси, скоротити витрати та підвищити продуктивність. Це особливо важливо в умовах високої конкуренції та необхідності оптимізації ресурсів.

- Поліпшення якості рішень: ВАР допомагає бізнес-аналітикам збирати та аналізувати дані, що сприяє прийняттю більш обґрунтованих та якісних управлінських рішень. Це, своєю чергою, покращує результати діяльності компанії.

- Інтеграція нових технологій: Впровадження нових технологій, таких як штучний інтелект та машинне навчання, потребує ретельного аналізу та структурування. ВАР надає інструменти для аналізу та інтеграції цих технологій у бізнес-процеси.

- Дотримання стандартів та нормативів: Сучасні компанії повинні дотримуватись різних стандартів та нормативних вимог. ВАР допомагає структурувати та систематизувати ці процеси, забезпечуючи відповідність встановленим вимогам.

- Глобалізація бізнесу: В умовах глобалізації багато компаній працюють на міжнародних ринках. ВАР допомагає адаптувати бізнес-процеси під різні культурні, економічні та юридичні умови, що збільшує ефективність роботи на глобальному рівні.

Актуальність теми ВАР обумовлена її здатністю покращувати адаптацію бізнесу до змін, підвищувати ефективність процесів та якість управлінських рішень, інтегрувати нові технології, дотримуватись нормативних вимог та успішно працювати в умовах глобалізації.

1.2 Мета та завдання дослідження

Метою цієї роботи є створення інформаційної бази [2] на платформі ВАР, спрямованої на автоматизацію основних бізнес-процесів та підвищення ефективності роботи підприємства, яке займається продажем телефонів, чохлів, інших аксесуарів, гаджетів та наданням послуг зв'язаних с технікою.

Задачі які будуть розглянуті у цій роботі:

- визначення вимог до інформаційної бази.
- Розробка та впровадження інформаційної бази в ВАР
- Тестування та оцінка ефективності впровадження системи.

Основні компонентів предметної області

- Підприємство
- Характеристика підприємства
- Бізнес процеси
- Опис основних бізнес-процесів, що вимагають автоматизації (управління складом, бухгалтерський облік, управління персоналом, виробництво тощо).

Вимоги до інформаційної системи:

- Функціональні вимоги (автоматизація обліку, управління закупівлями та продажами, звітність).
- Нефункціональні вимоги (зручність використання, безпека даних, масштабованість).

Платформа ВАР

- Можливість платформи ВАР для автоматизації бізнес-процесів.

- Переваги використання VAF (інтеграція з іншими системами, гнучкість налаштування, підтримка законодавства).

1.3 Етапи реалізації проекту

Етапи реалізації проектування включають визначення вимог до системи, розробку архітектури та конфігурації інформаційної бази, і навіть інтеграцію з існуючими системами. Після цього слідує етап тестування, де проводиться перевірка функціональності та продуктивності системи, і завершується процес оцінкою ефективності впровадження та навчання співробітників.

1. Аналіз та проектування

- Збір та аналіз вимог
- Моделювання бізнес-процесів та проектування інформаційної бази

2. Розробка

- Створення структури бази даних
- Розробка модулів та інтерфейсів для користувачів
- Інтеграція з існуючими системами

3. Тестування

- Проведення функціонального та навантажувального тестування
- Внесення коригувань за результатами тестування

4. Впровадження та навчання персоналу

- Покрокове впровадження системи в експлуатацію
- Навчання користувачів роботі з новою системою

1.4 Оцінка ефективності

Оцінка ефективності включає аналіз продуктивності системи та порівняння показників до та після впровадження, а також збір відгуків

від користувачів. Крім того, проводиться аналіз фінансових показників, щоб визначити економічну доцільність та повернення інвестицій.

1. Оцінка ефективності

- Порівняльний аналіз показників роботи підприємства до та після впровадження системи
- Оцінка економічної ефективності (скорочення витрат, збільшення продуктивності)
- Анкетування та інтерв'ювання співробітників для оцінки зручності та якості роботи з системою

1.5 Аналіз вимог

Аналіз потреб включає збирання та вивчення вимог від усіх зацікавлених сторін, визначення ключових функцій і характеристик, необхідних для ефективної роботи системи. Також проводиться аналіз існуючих бізнес-процесів та виявлення проблемних областей, що потребують оптимізації та автоматизації.

Для створення конфігурації VAF необхідно провести аналіз вимог, що включає:

Функціональні вимоги:

- Автоматизація процесу замовлення та обліку техніки.
- Управління запасами та контроль складських залишків у реальному часі.
- Інтеграція із системами постачальників для автоматизації процесу закупівлі.
- Управління процесом доставки та відстеження вантажів.
- Підтримка та обслуговування клієнтів через централізовану систему заявок.

Нефункціональні вимоги

- Висока продуктивність та відмовостійкість системи.
- Масштабованість для підтримки зростання компанії.
- Безпека даних та дотримання законодавчих вимог (наприклад, GDPR).

1.6 Технічні аспекти

Технічні аспекти включають вибір відповідної архітектури системи, налаштування та інтеграцію бази даних, а також забезпечення сумісності з ІТ-інфраструктурами. Важливо також враховувати безпеку даних, продуктивність системи та можливість масштабування для підтримки зростання бізнесу. Під час створення конфігурації VAF необхідно враховувати такі технічні аспекти:

- Архітектура системи: Вибір відповідної архітектури (наприклад, мікросервісна архітектура для підвищення гнучкості та масштабованості).
- Інтеграція: Забезпечення інтеграції з існуючими системами (ERP, CRM, WMS тощо).
- Бази даних: Вибір та налаштування бази даних для зберігання та обробки даних.
- Інтерфейси: Розробка зручних та інтуїтивно зрозумілих інтерфейсів для співробітників компанії.
- Хмарні технології: Можливість використання хмарних технологій для підвищення доступності та масштабованості.

1.7 Ризики та їх мінімізація

Ризики під час впровадження системи включають можливі складності інтеграції з існуючими системами, високі початкові витрати та потенційні збої у роботі. Для їхньої мінімізації необхідно проводити поетапне

впровадження з ретельним тестуванням, навчання співробітників роботі з новою системою, а також розробка планів дій у разі непередбачених ситуацій та регулярне створення резервних копій даних.

Ризики під час впровадження ВАР:

- Складність інтеграції із існуючими системами.
- Високі початкові витрати на розробку та впровадження.
- Можливі збої та помилки у процесі переходу на нову систему.

Для мінімізації ризиків:

- Проведення поетапного застосування з тестуванням кожному етапі.
- Навчання співробітників роботі з новою системою.
- Створення резервних копій даних та розробка плану дій у разі збоїв.

1.8 Оцінка вартості та термінів

Оцінка вартості проекту включає врахування витрат на розробку та налаштування системи, закупівлю необхідного обладнання та програмного забезпечення, а також навчання персоналу. Додатково необхідно передбачити витрати на підтримку та обслуговування системи, а також можливі витрати на модернізацію та масштабування у майбутньому.

Оцінка вартості включає:

- Розробка та налаштування ВАР.
- Витрати обладнання та програмне забезпечення.
- Навчання персоналу.
- Підтримка та обслуговування системи.

Терміни залежать від масштабу проекту, складності інтеграції та кількості бізнес-процесів, які потрібно автоматизувати.

2 ЗАГАЛЬНИЙ ВИГЛЯД КОНФІГУРАЦІЇ

2.1 Мова програмування

Мова програмування VAF — вбудована мова платформи VAF. Він предметно-орієнтований, тобто на ньому не можна писати сайти та ігри, але можна швидко та економічно автоматизувати бухгалтерський та податковий облік, облік на складах, розробити CRM-систему та робити багато іншого.

Вбудована мова підтримує механізм ОВП, але у досить урізаному вигляді: є суворий набір класів із заданими властивостями та методами, наприклад, Довідники, Документи, Регістри, Звіти, Обробки. Нові класи створювати не можна, але для розробки достатньо тих, що є.

За синтаксисом мова VAF нагадує Pascal і BASIC у поєднанні з мовою запитів T-SQL, хоча його не можна назвати аналогом. Писати код у VAF можна двома мовами — російською та англійською. Програмісти, як правило, віддають перевагу російській. На нього швидко переходять і ті, хто раніше програмував англійською — так зрозуміліше.

2.2 Система захисту

Для захисту продукту «VAF» від несанкціонованого використання використовує апаратні ключі HASP. У базових конфігураціях версії 8 з'явився захист через електронні ключі, які потрібно активувати після придбання програми. В даний час використовується на всіх нових поставках, апаратні ключі продаються лише за запитом та версією дорожчою.

У самій конфігурації [3] для захисту від сторонніх осіб є вхід за логіном та паролем, як показано на рисунку 1 Також, навіть якщо людина зареєстрована у базі, їй можна заборонити входити до самої бази. Ще в конфігурації існують ролі, якими можливо заблокувати доступ до певних об'єктів конфігурації, якщо на те є потреба.

2.3 Початковий інтерфейс

На рисунку 4 можна побачити, що в початковому інтерфейсі виведені такі об'єкти як: План обміну «Філіали», документ «Надання послуг» та звіт про начислення робітникам.

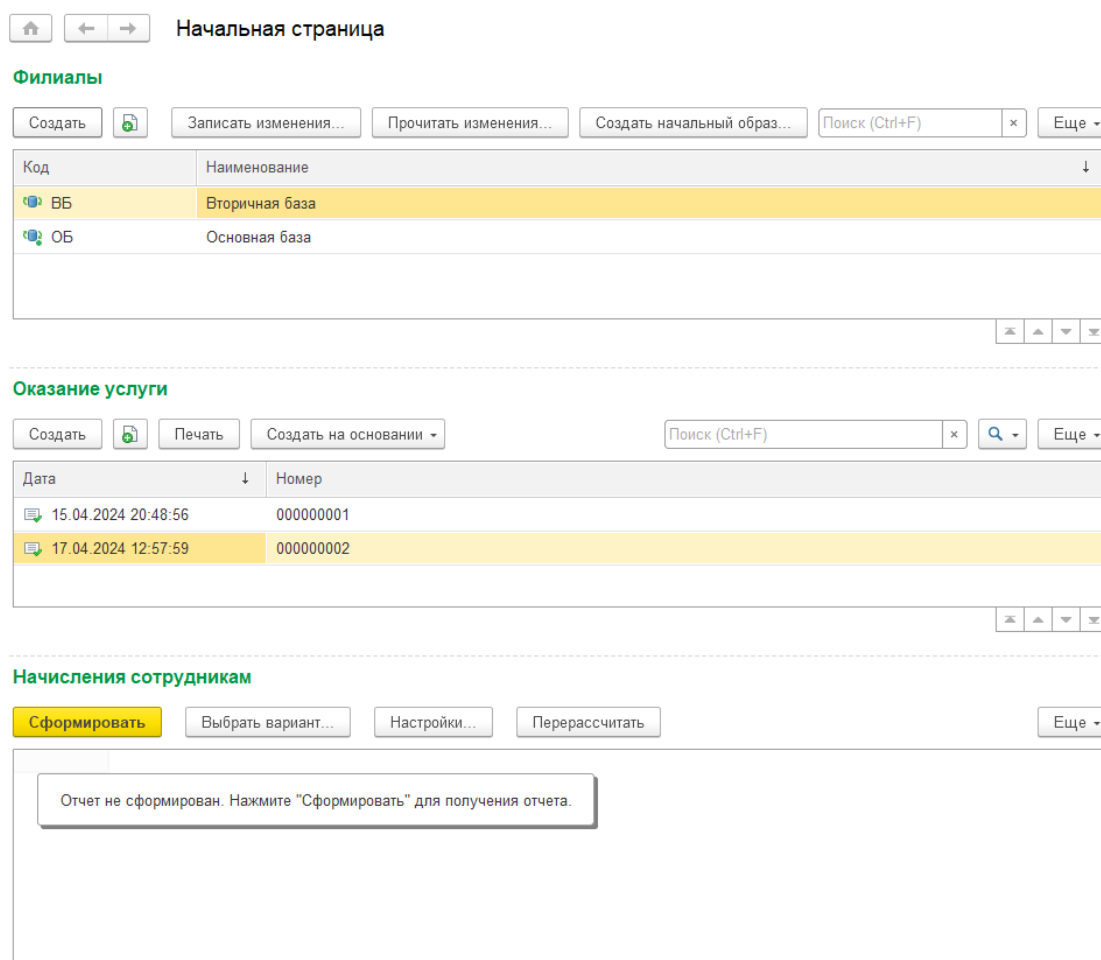


Рисунок 4 – Видяг початкового інтерфейсу

Також в конфігурації реалізовані підсистеми для того щоб велика кількість об'єктів не була в одному місці.

Підсистеми - це загальні об'єкти конфігурації. На їх основі платформа формує командний інтерфейс прикладного рішення та візуально поділяє всю функціональність програми на великі та дрібні блоки.

Підсистеми можуть мати ієрархічну структуру, тобто одна підсистема може включати кілька інших підсистем. Кожен конфігураційний об'єкт можна включити до складу однієї або декількох підсистем. Таким чином, у термінах підсистем можна описати всю структуру прикладного рішення

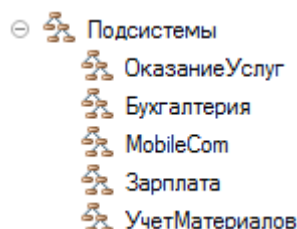


Рисунок 5 – Підсистеми у конфігураторі

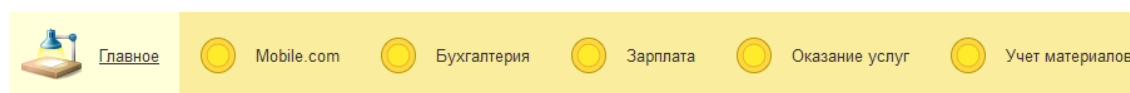


Рисунок 6 – Підсистеми у підприємстві

2.4 Порівняння з існуючими програмами

Порівняння VAF з аналогічними фреймворками дозволяє зрозуміти їх сильні та слабкі сторони, а також виявити найбільш відповідні рішення для конкретних бізнес-потреб. Розглянемо кілька популярних фреймворків, включаючи VAF, BABOK, TOGAF та Zachman Framework.

1. VAF

VAF є платформою для розробки та управління бізнес-додатками, призначеною для автоматизації та оптимізації різних бізнес-процесів. Вона забезпечує гнучкість та масштабованість, дозволяючи компаніям інтегрувати свої існуючі системи та адаптувати рішення під специфічні потреби.

Переваги:

Гнучкість: VAF дозволяє адаптуватися до різних бізнес-процесів та вимог.

Системний підхід: Забезпечує структурований метод аналізу та проектування бізнес-процесів.

Інтеграція з технологіями: Легко інтегрується із сучасними ІТ-рішеннями та методологіями.

Недоліки:

Об'ємність: Може бути надто складним для невеликих компаній або проектів.

Вимоги до кваліфікації: Вимагає високого рівня знань та досвіду для ефективного використання.

2. VABOK (Business Analysis Body of Knowledge)

VABOK – це акронім від Business Analysis Body of Knowledge, керівництво зведення знань з бізнес-аналізу від Міжнародного інституту ІВА (International Institute of Business Analysis) [1]. Примітно, що керівництво позиціонується не як професійний стандарт бізнес-аналітика, а як набір загальноприйнятих практик бізнес-аналізу.

Переваги:

Стандартизація: є глобально визнаним стандартом у сфері бізнес-аналізу.

Навчання та сертифікація: Пропонує програми сертифікації (CBAP, CCBA), які допомагають підвищити кваліфікацію спеціалістів.

Широке застосування: Застосовуємо у різних галузях та типах проектів.

Недоліки:

Обширність: Великий посібник може бути важким для освоєння та застосування.

Консерватизм: Найменша гнучкість у порівнянні з іншими фреймворками.

3. TOGAF (The Open Group Architecture Framework)

TOGAF - це структура корпоративної архітектури для розробки та управління ІТ-стратегією, орієнтованою на бізнес.

Це набір методів, інструментів та шаблонів для визначення того, як працює підприємство. Інструменти TOGAF допомагають визначити та змодельовати найважливіші ризики для організації, а також вивчити, як різні підходи можуть усунути чи зменшити ці ризики.

Переваги:

Повнота: Охоплює всі аспекти архітектури підприємства, включаючи бізнес, дані, програми та технології.

Модульність: дозволяє адаптувати методологію під конкретні потреби компанії.

Спільнота та підтримка: Широка підтримка з боку професійної спільноти та доступність ресурсів.

Недоліки:

Складність застосування: Вимагає значних зусиль для впровадження та підтримки.

Залежність від кваліфікації: Вимагає високої кваліфікації та досвіду спеціалістів.

4. Zachman Framework

Zachman Framework – це інструмент, який використовується для організації та класифікації архітектури підприємства. Це матриця, що складається з шести рядків і стовпців, де кожен осередок представляє певний аспект архітектури.

Переваги:

Універсальність: Застосовуємо різні типи підприємств і проектів.

Структурованість: Чітка ієрархія та структура допомагають систематизувати дані та процеси.

Гнучкість: Легко адаптується до вимог, що змінюються.

Недоліки:

Спільність: Можливо недостатньо конкретним реалізації специфічних завдань.

Потребує глибокого опрацювання: Вимагає детального аналізу та опрацювання всіх аспектів підприємства.

Критерий	BAF	BAВOK	TOGAF	Zachman Framework
Гибкость	Высокая	Средняя	Средняя	Высокая
Простота освоения	Средняя	Низкая	Низкая	Средняя
Полнота охвата	Высокая	Высокая	Очень высокая	Высокая
Область применения	Бизнес-анализ	Бизнес-анализ	Архитектура предприятия	Архитектура предприятия
Требования к квалификации	Высокие	Высокие	Высокие	Средние

Рисунок 7 - Порівняльна таблиця

Таким чином, вибір конкретного фреймворку залежить від специфіки бізнесу, масштабу проекту та кваліфікації спеціалістів. BAF підходить для компаній, які цінують гнучкість та інтеграцію із сучасними технологіями. BAVOK кращий для тих, хто прагне стандартизації та сертифікації в галузі бізнес-аналізу. TOGAF та Zachman Framework більш орієнтовані на архітектуру підприємства та підходять для великих організацій з комплексною структурою.

3 ПРОЕКТУВАННЯ СТРУКТУРИ

3.1 Проектування бази для компанії

1. Аналіз вимог

Перед початком проектування необхідно зрозуміти бізнес-процеси компанії, визначити, які дані зберігатимуться і як вони використовуватимуться. Основні категорії даних можуть включати:

- Продукти (назва, опис, ціна, кількість на складі тощо)
- Клієнти (ім'я, контактна інформація, історія покупок тощо)
- Замовлення (номер замовлення, дата, статус, інформація про клієнтів та продукти)
- Постачальники (інформація про постачальників, контакти, історія постачання)
- Співробітники (інформація про співробітників, роль, контактні дані)

2. Створення діаграми сутностей-зв'язків

Діаграми допомагають візуалізувати сутності та зв'язки між ними. Це полегшує розуміння структури бази даних та взаємодії даних.\

3. Нормалізація даних

Нормалізація допомагає усунути надмірність даних та забезпечити цілісність бази даних. Зазвичай нормалізація відбувається за кілька етапів (нормальних форм).

4. Тестування та оптимізація

Після створення бази даних необхідно провести тестування для виявлення та усунення можливих помилок. Також проводиться оптимізація бази даних для забезпечення її ефективності та швидкодії.

3.2 Об'єкти конфігурації та їх призначення

Об'єкти конфігурації — це складові елементи, «деталі», у тому числі складається будь-яке прикладне рішення.

Вони є проблемно-орієнтовані об'єкти, що підтримуються лише на рівні технологічної платформи. За великим рахунком, завдання розробника полягає в тому, щоб зібрати з цих об'єктів, як з конструктора, необхідну структуру прикладного рішення і потім описати специфічні алгоритми функціонування та взаємодії цих об'єктів, що відрізняються від їх типової поведінки.

Об'єкт конфігурації [4] Довідник призначений для роботи з списками даних. Як правило, у роботі будь-якої фірми використовуються списки співробітників, списки товарів, списки клієнтів, постачальників і т. д. Властивості та структура цих списків описуються в об'єктах конфігурації Довідник, на основі яких платформа створює в базі даних таблиці для зберігання інформації з цих довідників.

Довідник складається із елементів. Наприклад, для довідника співпраці-ником елементом є співробітник, для довідника товарів –товар і т. д. Користувач у процесі роботи може самостійно додавати нові елементи до довідника: наприклад, додати нових співробітників створити новий товар або внести нового клієнта.

У базі даних кожен елемент довідника є окремий запис в основній таблиці, що зберігає інформацію з цього довідника.

Більшість реквізитів розробник створює самостійно, однак кожен об'єкт конфігурації Довідник за замовчуванням існує набір стандартних реквізитів: Код та Найменування. Причому доступність стандартних реквізитів залежить від властивостей довідника. Наприклад, якщо ієрархічний довідник, у нього буде доступний стандартний реквізит Батько. Якщо довідник підпорядкований іншому об'єкта конфігурації, у нього буде доступний реквізит Власник. Якщо встановити довжину стандартного реквізиту Код

дорівнює нулю, то у довідника буде недоступний цей реквізит. Теж саме відноситься до реквізиту Найменування. Однак як мінімум або Код, або Найменування повинні бути присутніми в реквізитах довідника, інакше такий довідник не має сенсу. Таким чином, у базі даних довідник зберігається у вигляді таблиці, у рядках якої розташовані елементи списку, а кожному реквізиту (стандартному чи створеному розробником) у цій таблиці відповідає окремий стовпець. Відповідно, у осередках цієї таблиці зберігається значення конкретного реквізиту для конкретного елемента довідника.

Об'єкт конфігурації Документ призначений для опису інформації про скоєні господарські операції або про події, що відбулися у житті організації взагалі. Як правило, у роботі будь-якої фірми використовуються такі документи, як прибуткові накладні, накази про прийом на роботу, платіжні доручення, рахунки і т. д. Властивості та структура цих документів описуються в об'єктах конфігурації Документ, на основі яких платформа створює базу даних таблиці для зберігання інформації з цих документів.

Логіка роботи документів відрізняється від логіки роботи інших об'єктів конфігурації. Документ має здатність проведення. Факт проведення документа означає, що подія, яку він відбиває, вплинув стан обліку. Доки документ не проведено, стан обліку незмінний, і документ не більше ніж чернетка, заготівля. Як тільки документ буде проведено, зміни, що вносяться документом в облік, набудуть чинності в силу та стан обліку буде змінено.

Оскільки документ вносить зміни до стану обліку, він завжди "прив'язаний" до конкретного моменту часу. Це дозволяє відобразити у базі даних фактичну послідовність подій.

Об'єкт конфігурації Перелік [5] призначений для опису структури зберігання постійних наборів значень, що не змінюються у процесі конфігурації. На основі об'єкта конфігурації Перерахування платформи створює в базі даних таблицю, в якій може зберігатись набір деяких постійних

значень. У реальному житті цьому об'єкту може відповідати, наприклад, перелік варіантів вказівки ціни («включаючи ПДВ», «без ПДВ»). Набір всіх можливих значень, які містить перерахування, задається при конфігуруванні системи, і користувач не може змінювати їх, видаляти чи додавати нові.

Об'єкт конфігурації План видів характеристик призначений для опису структури зберігання інформації про характеристики, створюваних користувачем. На основі об'єкта конфігурації План Види параметрів платформа створює в базі даних набір таблиць, в яких зберігатиметься інформація про існуючі види характеристик і тип значення значення характеристики кожного виду. По суті, план видів характеристик дуже нагадує довідник, проте має більш вузьку спеціалізацію: зберігає, по суті, інформацію лише у тому, якими видами характеристик може описувати будь-який об'єкт бази даних. План видів показників складається з видів показників. Кожен вид характеристики обов'язково описується найменуванням та типом значення. Розробник і, що найважливіше, користувач можуть поставити в ньому будь-яку необхідну їм кількість видів характеристик.

Об'єкт конфігурації Регістр накопичення призначений для опису структури накопичення даних. На основі об'єкта конфігурації «Регістр накопичення» платформа створює у базі даних таблиці, в яких накопичуватимуться дані, що постачаються різними об'єктами бази даних.

Ці дані зберігатимуться у таблицях у вигляді окремих записів, кожна з яких має однакову задану конфігураторі структуру.

На підставі таблиці рухів реєстру накопичення система розраховує таблицю підсумків реєстру, що зберігає у базі даних підсумки на даний момент часу останнього руху (актуальні результати).

Відмінною особливістю реєстру накопичення є те, що він не призначений для інтерактивного редагування користувачем. Розробник може за необхідності надати користувачеві можливість редагувати реєстр накопичення. Але призначення реєстру накопичення полягає в тому, щоб

його модифікація вироблялася на основі алгоритмів роботи інших об'єктів бази даних, а чи не внаслідок безпосередніх дій користувача.

Основним призначенням реєстру накопичення є накопичення числової інформації в розрізі кількох вимірів, які описуються розробником у відповідному об'єкті конфігурації Реєстр накопичення та є підпорядкованими об'єктами конфігурації. Види числової інформації, що накопичується реєстром накопичення, називаються ресурсами, також є підлеглими об'єктами та описуються в конфігураторі.

Об'єкт конфігурації Реєстр відомостей [7] призначений для опису структури зберігання даних у межах декількох вимірів. На основі об'єкта конфігурації Реєстр відомостей платформа створює в базі даних таблицю, в якій може зберігатися довільна інформація.

Принципова відмінність реєстру відомостей від реєстра накопичення у тому, що кожен рух реєстру відомостей встановлює нове значення ресурсу, тоді як рух реєстру Нагромадження змінює існуюче значення ресурсу. За цією причини реєстр відомостей може зберігати будь-які дані (а не тільки числові, як реєстр накопичення).

Наступною важливою особливістю реєстру відомостей є його здатність (за потреби) зберігати дані з прив'язкою на час. Завдяки цьому реєстр відомостей може зберігати як актуальні значення даних, а й історію їх зміни в часі. Реєстр відомостей, що використовує прив'язку до часу, називають періодичним реєстром відомостей.

Періодичність реєстру відомостей можна визначити одним з наступних значень:

- в межах секунди;

- в межах дня;

- в межах місяця;

- в межах кварталу;

- в межах року;

в межах реєстратора (якщо встановлено режим запису «Підпорядкування реєстратору»).

Періодичний реєстр відомостей завжди містить службове поле Період автоматично додається системою. Воно має тип Дата і служить для вказівки факту належності запису до будь-якого періоду. При записі даних у реєстр платформа завжди наводить значення цього поля на початок того періоду, в який він потрапляє.

3.3 Побудова структури на UML діаграмах

Для кращого розуміння та візуалізації внутрішньої структури, використовувалися UML діаграми.

Діаграми послідовностей - це один з типів діаграм UML, які використовуються для моделювання взаємодії об'єктів у системі в певний момент часу. Ці діаграми відображають, як об'єкти взаємодіють один з одним шляхом надсилання повідомлень у певній послідовності.

Основні елементи діаграми послідовностей:

Актори - зовнішні сутності, які взаємодіють із системою (наприклад, користувачі).

Об'єкти – елементи системи, які обмінюються повідомленнями.

Лінії життя - вертикальні лінії, які показують існування об'єкта протягом часу.

Повідомлення – стрілки між об'єктами, які показують обмін інформацією.

Активні області — це прямокутники на лініях життя, які показують, коли об'єкт виконує операцію.

Діаграми послідовностей потрібні для:

1. Розуміння взаємодій: Допомагають зрозуміти, як різні частини системи взаємодіють одна з одною, що особливо корисно на етапі проектування.

2. Аналіз вимог: Дозволяють уточнити та проаналізувати вимоги до системи, демонструючи, як мають працювати різні сценарії.

3. Документація системи: Служать засобом документування, яке легко розуміють розробники, аналітики та інші зацікавлені сторони.

4. Налаштування та тестування: Допомагають виявити потенційні проблеми та помилки у логіці взаємодій між компонентами системи.



Рисунок 8 — Діаграма послідовності входу до бази

Далі, наприклад, якщо користувач хоче провести якусь операцію, то діаграма буде виглядати наступним чином:

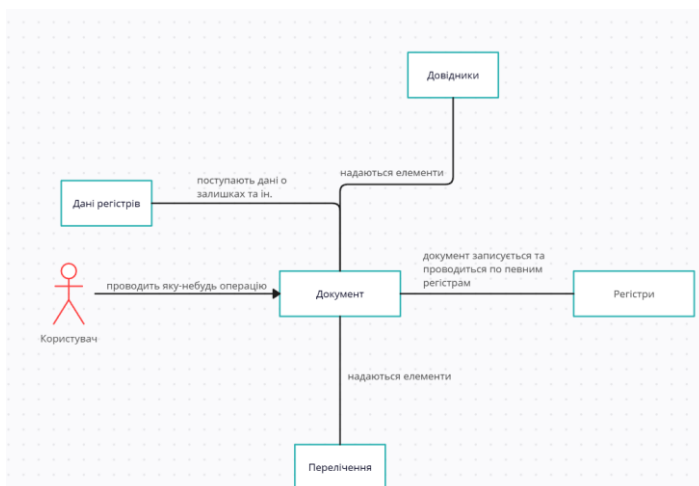


Рисунок 9 — Діаграма послідовності проведення документа

На рис. 9 можна побачити, що в документі можна взаємодіяти з елементами певних довідників і тд., якщо у документа є реквізити з посиланням на них. Далі документ записується і проводиться по реєстрам, створюючи нові записи.

На рис 10 показано як збирається звіт по реєстру:

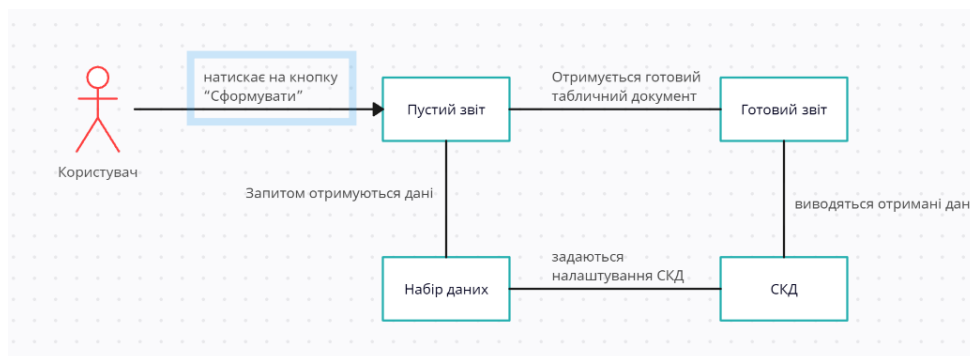


Рисунок 10 — Діаграма послідовності формування звіту

Користувач натискає на кнопку формування, далі за допомогою набору даних в СКД отримуються дані з БД, формуються та виводяться в готовий табличний документ.

4 РЕАЛІЗАЦІЯ ОБ'ЄКТІВ КОНФІГУРАЦІЇ

4.1 Довідники

Перелік довідників які були створені у даній конфігурації:













- ⊕  Клиенты
- ⊕  Поставщики
- ⊕  Номенклатура
- ⊕  Сотрудники
- ⊕  Должности
- ⊕  Магазины
- ⊕  МоделиТелефонов
- ⊕  ВариантыТоваров
- ⊕  ДопСвойстваТоваров
- ⊕  Субконто
- ⊕  ВидыГрафиковРаботы

Рисунок 11 – Довідники конфігурації

Довідник «Клієнти» має в собі інформацію про клієнтів нашої бази. Подалі, у документах можна вказувати ці елементи.

🏠 ← → ☆ Клиенты

Создать  Создать на основании ▾

ФИО	↓	Адрес	Номер телефона
—	Гайдан Чан		
—	Дворник Сергей Николаевич	Пироговская 3/1	380932507220
—	Жмышенко Анна Игоревна	Гулшко 2/А	380934441859
—	Людмила Ивановна		380952167320
—	Филинюк Григорий	Ильфа и Петрова 20	
—	Чмырь Игорь Михайлович	Львовская 27	

Рисунок 12 – Форма списку довідника «Клієнти»

Обов'язковим реквізитом для заповнення є «ФІО». Інші реквізити, такі як «Адреса» та «Номер телефону» є інформаційними та заповнюються за бажанням. Тип цих реквізитів – рядок.

☆ Дворник Сергей Николаевич (Клиент) 🔗 ⋮ □ ✕

Основное [Оказание услуги](#)

Записать и закрыть Записать Создать на основании ▾ Еще ▾

Код:

ФИО:

Адрес:

Номер телефона:

Рисунок 13 – Форма об'єкта довідника «Клієнти»

У довіднику «Постачальники» зберігається інформація про постачальників.

Єдиний та обов'язковий реквізит – «Найменування»

🏠 ← → ☆ Поставщики

Создать

Наименование
Altmir
Avroga
Bigl.ua
BigOpt
Rozetka
Zakupka.com
ВсеЕсть
МирТехники

Рисунок 14 – Форма об'єкта довідника «Постачальники»

У довіднику «Номенклатура» знаходиться інформація про всю номенклатуру, її вид та групи. Цей довідник є ієрархічним. Тобто, у довіднику можуть бути як елементи, так і групи елементів. Також кожен елемент має додатковий реквізит «Власник».

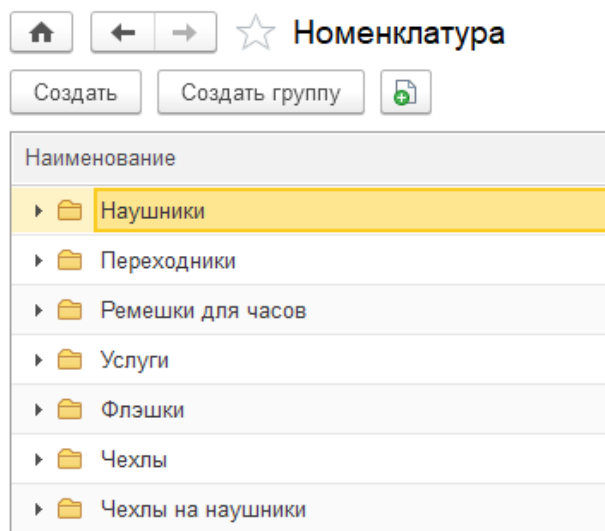


Рисунок 15 – Форма списка довідника «Номенклатура»

У даному довіднику обов'язковими реквізитами є «Найменування» та «Вид номенклатури». Номенклатура може бути товаром чи послугою. Реквізит «Вид» є посиланням на перелічення «Види номенклатури», щоб розрізняти товари та послуги в даному довіднику та подалі у конфігурації так кодів.

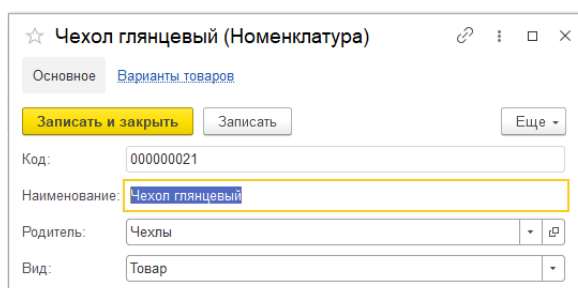


Рисунок 16 – Форма об'єкта довідника «Номенклатура»

На рис. 16 можна побачити, що є напис «Варіанти товарів». Це гіперссылка на довідник «Варіанти товарів». Натиснувши на неї, нас перекидає до форми списку варіантів товару.

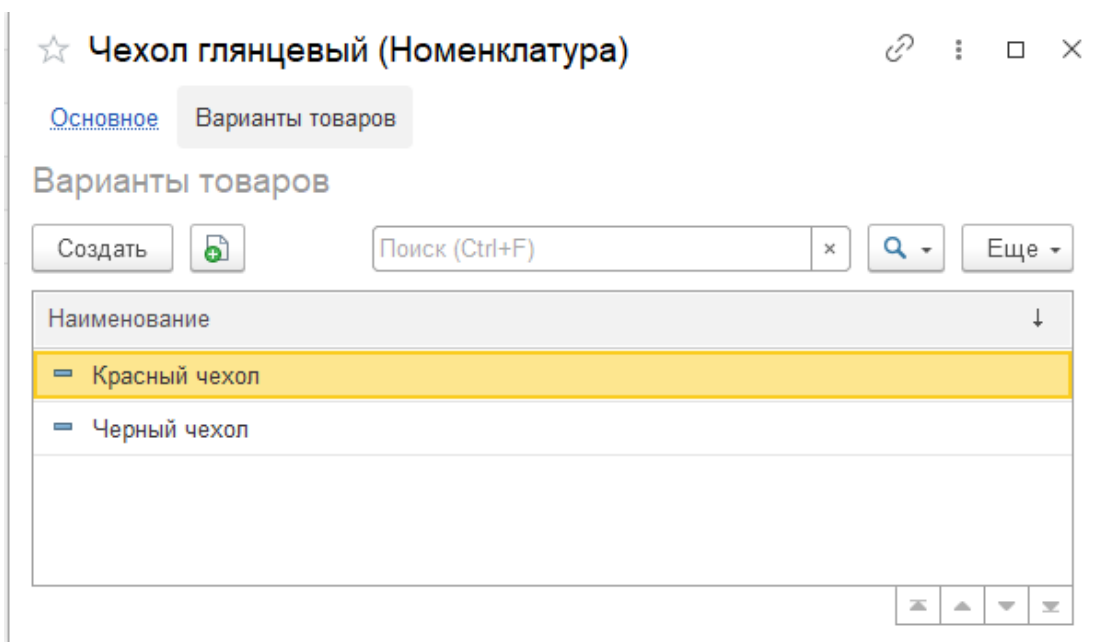


Рисунок 17 – Форма списку довідника «Варіанти товарів»

У цьому довіднику є лише два реквізиту – «Найменування» та «Власник».

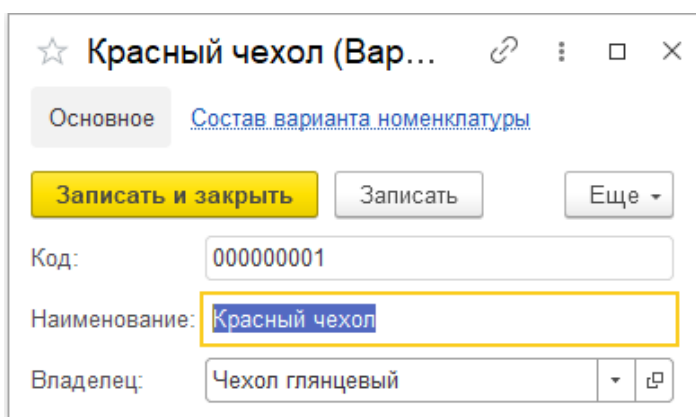


Рисунок 18 – Форма об'єкта довідника «Варіанти товарів»

На рисунку 18 теж можна побачити гіпер-ссылку. Вона приведе нас до довідника «Склад варіантів номенклатури», який зв'язаний з варіантами товарів.

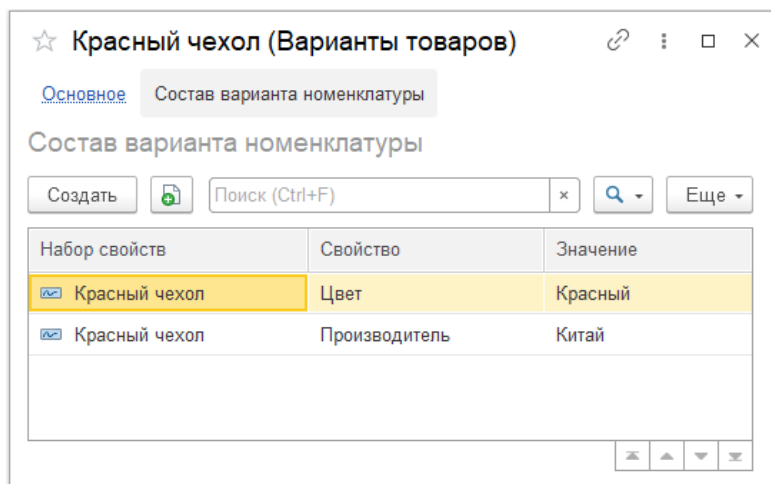


Рисунок 19 – Форма елемента довідника «Склад варіантів номенклатури»

Інформація про всіх співробітників знаходиться у довіднику «Співробітники». Обов'язковими реквізитами є «ФІО» та «Посада». Усі реквізити окрім реквізиту «Посада» мають типи – рядок. «Посада» як очевидно є посиланням на довідник «Посади».

ФИО	Должность	Номер телефона	Адрес
Быстрый Леонид	Продавец	380 931 802 330	Вильямса 7
Ечин Даниил	Продавец	380 974 501 622	
Корчаковский Кирилл	Мастер по ремонту	380 974 302 893	Архитекторская 14/1
Тарасенко Елена Петровна	Менеджер	380 937 772 215	
Швед Евгений Эдуардович	Менеджер	380 934 505 764	
Шевченко Дмитрий	Мастер по ремонту	380 975 508 012	Глушко 1/В

Рисунок 20 – Форма списку довідника «Співробітники»

Довідник «Посади». Обов'язковим реквізитом є «Найменування».

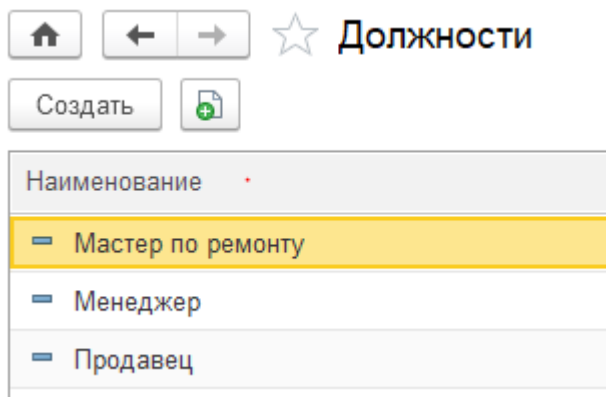


Рисунок 21 – Форма списку довідника «Посади»

Інформація про магазини нашої компанії теж зберігаються у довіднику.

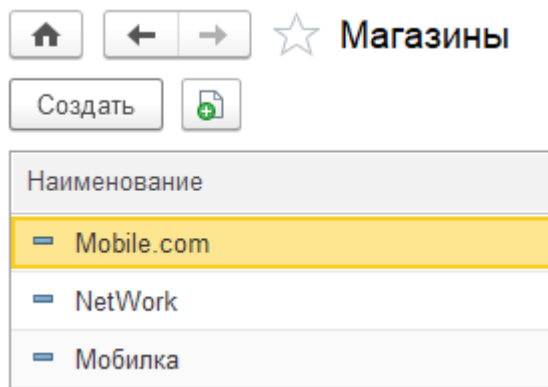


Рисунок 22 – Форма списку довідника «Магазини»

Тут обов'язковими реквізитами є «Найменування» та «Адреса». Також тут можна вказати в які дні працює магазин, початок та кінець дня. Таблична частина, яка вказує робочі дні магазину є посиланням на переліченням «Дні тижня».

☆ Mobile.com (Магазин)

Записать и закрыть Записать Еще ▾

Код: 000000001

Наименование: Mobile.com

Номер телефона: 380933051889

Адрес: Архитекторская 1

<input checked="" type="checkbox"/>	День недели	Время начала	Время окончания
<input checked="" type="checkbox"/>	Понедельник	09:00	19:00
<input checked="" type="checkbox"/>	Вторник	09:00	19:00
<input checked="" type="checkbox"/>	Среда	09:00	19:00
<input checked="" type="checkbox"/>	Четверг	09:00	19:00
<input checked="" type="checkbox"/>	Пятница	09:00	19:00
<input checked="" type="checkbox"/>	Суббота	09:00	19:00
<input type="checkbox"/>	Воскресенье	09:00	19:00

Рисунок 23 – Форма элементу довідника «Магазини»

Для виду графіків працівників також є довідник. Для цього довідника є два зумовлених реквізиту «Графік адміністрації» та «Графік інших».

☆ Виды графиков работы

Создать +

Наименование

- График администрации
- График остальных

Рисунок 24 – Форма списку довідника «Види графіків роботи»

Довідник «Моделі телефонів». Зберігає моделі товарів, якщо така є.

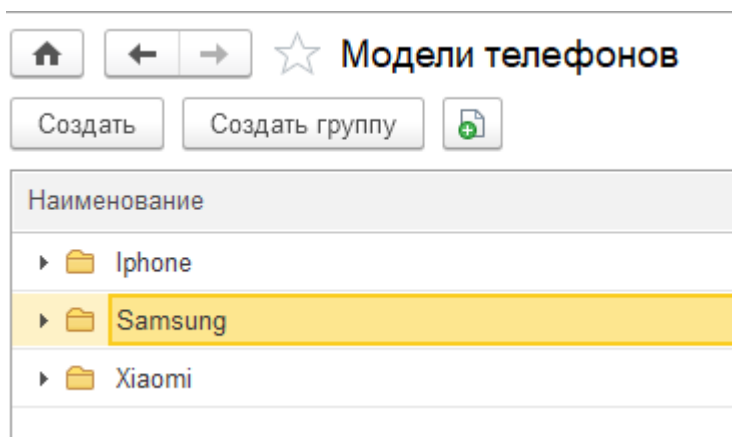


Рисунок 25 – Форма списка довідника «Моделі телефонів»

4.2 Документи

Документ «Прибуткова накладна» потрібен для поповнення залишків. Грає роль партії у документі «Надання послуг», з якої запишуться товари.

Тут вказується постачальник, товари, їх модель, кількість та ціна.

Також на формі списку зроблена кнопка для завантаження даних з Екселя.

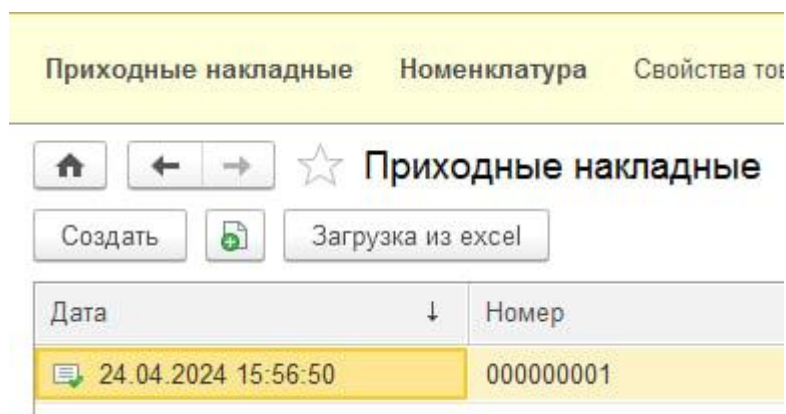


Рисунок 26 – Форма списку документа «Прибуткова накладна»

На рис 27 натиснувши на кнопку, буде відкрита форма завантаження, як це видно на рис. 28, де треба вибрати файл, який зображено на рис 1.6, для

завантаження інформації. У коді задано відбір тільки на файли з типом «Ексель».

Форма загрузки

Путь к файлу: K:\Work1C\GolyukMI\Универ\Rozetka.xlsx

Загрузить

Рисунок 27 – Форма завантаження даних

	1	2	3	4	5
1	Мак бук			100	20000
2	Iphone 12			100	15000
3	Iphone 13			100	17000
4					
5					
6					
7					

Рисунок 28 – Файл завантаження

За цими даними буде створено новий документ, с датою сьогоднішнього дня, постачальник береться з назва файлу та у циклі будуть заповнені товари.

Перейдемо до лістингу коду.

```

&НаКлиенте
Процедура ПутьКфайлуНачалоВыбора (Элемент, ДанныеВыбора, СтандартнаяОбработка)
»
» Диалогфайла := Новый ДиалогВыборафайла (РежимДиалогаВыборафайла.Открытие);
» Диалогфайла.фильтр := "Книга Excel (*.xlsx; *.xls) | *.xlsx; *.xls";
» Диалогфайла.Заголовок := "Выберите файл";
»
» Если Диалогфайла.Выбрать () Тогда
» » ПутьКфайлу := Диалогфайла.ПолноеИмяфайла;
» » КонечЕсли;
»
»
КонечПроцедуры

```

Рисунок 29 – Процедура «Початок вибору» реквізиту форми «Шлях до файлу»

У процедурі на рис 30 обмежується тип файлу. Будемо завантажувати данні з Excel файлів.

```

&НаКлиенте
Процедура Загрузить (Команда)
»
» файл := Новый файл (ПутьКфайлу);
» Если НЕ файл.Существует () Тогда
» » Сообщить ("Выбранный файл не существует!");
» » Возврат;
» » КонечЕсли;
»
»
» ДокументСсылка := ЗагрузитьНаСервере ();
» Этаформа.Закреть ();
»
» ПараметрыДляформы := Новый Структура;
» ПараметрыДляформы.Вставить ("Ключ", ДокументСсылка);
»
» Открытьформу ("Документ.ПриходнаяНакладная.формаОбъекта", ПараметрыДляформы);
»
»
КонечПроцедуры

```

Рисунок 30 – Обробник команди «Завантажити»

На рис 31 перевірка на існування файлу, якщо такий файл не існує, буде виведена помилка. Далі йде сама процедура обробки даних файлу.

```

&НаСервере
Функция ЗагрузитьНаСервере ()
»
» ДокЭксель := ПолучитьСОБъект (ПутьКФайлу) ;
»
» ЛистЭкселя := ДокЭксель.Sheets (1) ;
»
» НомерСтроки := 0 ;
» НомерКолонкиНоменклатура := 1 ;
» НомерКолонкиМодель := 2 ;
» НомерКолонкиНаборСвойств := 3 ;
» НоменКолонкиКоличество := 4 ;
» НомерКолонкиЦена := 5 ;
»
» МассивЗнаков := СтрРазделить (ДокЭксель.Name, "." ) ;
»
» ПоставщикИзФайла := МассивЗнаков [0] ;

```

Рисунок 31 — Процедура завантаження даних з файлу

На рис. 32 зображено як відкривається файл, отримуємо лист на якому є інформація. Заздалегідь домовившись з постачальником заповнюються номери колонок, у яких будуть знаходитись данні. Також отримується назва постачальника.

```

ЗапросПоставщик := Новый Запрос ;

ЗапросПоставщик.Текст :=
"ВЫБРАТЬ
|> Поставщики.Ссылка КАК Ссылка
| ИЗ
|> Справочник.Поставщики КАК Поставщики
| ГДЕ
|> Поставщики.Наименование = &Наименование" ;

ЗапросПоставщик.УстановитьПараметр ("Наименование", ПоставщикИзФайла) ;

ВыборкаПоставщик := ЗапросПоставщик.Выполнить ().Выбрать () ;

Если НЕ ВыборкаПоставщик.Следующий () Тогда
»
» Поставщик := Справочники.Поставщики.СоздатьЭлемент () ;
» Поставщик.Наименование := ПоставщикИзФайла ;
» Поставщик.Записать ()
»
Иначе
»
» Поставщик := ВыборкаПоставщик.Ссылка ;
»
КонецЕсли ;

```

Рисунок 32 — Процедура завантаження даних з файлу

Запитом перевіряємо чи є в нас у базі такий постачальник. Якщо ні, то створюємо його. Далі у циклі отримуємо товари з файлу.

Далі йде перевірка чи є в нас у базі такий товар. Також, якщо цей товар новий, йому буде присвоєна група. Перевірка на існування групи теж присутня.

Документ «Надання послуги».

Він потрібен для запису інформації про покупку чи надання послуги клієнту.

У даному документі зберігається інформація про те, який магазин надав послугу, сам клієнт, співробітник який обслуговував клієнта, спосіб оплати.

Також інформація про товар та його модель, або послуга, кількість, партія з якої спишеться кількість товару, ціна та сума.

Номенклатура Клиенты Сотрудники Филиалы Варианты товаров Доп свойства товаров Субконто Виды графиков работы До

Оказание услуги 000000001 от 10.03.2023 10:47:47

Основное [Остатки товаров](#) [Стоимость товаров](#) [Управленческий](#)

Провести и закрыть Записать Провести Печать

Номер: 000000001

Дата: 10.03.2023 10:47:47

Магазин: Mobile.com

Клиент: Гайдан Чан

Сотрудник: Ечин Даниил

Способ оплаты: Наличные

Добавить

Номенклатура	Модель	Набор свойств
Чехол пуховик	Iphone 14 Pro Max	

Поиск (Ctrl+F) × Еще -

Количество	Партия	Цена	Сумма
1	Приходная накладная 000000001 от 10.03.2023 12:29:02	470,00	470,00

Рисунок 33 – Форма элемента документа «Надання послуги»

Для полів «Кількість» та «Ціна» є обробники подій «ПриИзменении», де розраховується сума, де ціна * кількість номенклатури.

Для поля «Номенклатура» теж є обробник події «ПриИзменении». Там ми аналізуємо чи є номенклатура товаром чи послугою. Якщо не є послугою, тоді запитом звертаємось до бази даних, де беремо ціну на номенклатуру з урахуванням моделі, якщо вона вказана. Якщо це послуга, тоді модель не ураховується. Також при змінні моделі теж йде запит до БД.

```

<НаСервереБезКонтекста
Функция·ПолучитьИнформациюМодели(Номенклатура, ·Модель, ·Дата)
»
»   Результат := ·Новый·Структура("Цена", ·0);
»
»   Запрос := ·Новый·Запрос;
»   Запрос.Текст := "ВЫБРАТЬ
»   ..... |»   ЦеныСрезПоследних.Цена·КАК·Цена
»   ..... |»   ИЗ
»   ..... |»   РегистрСведений.Цены.СрезПоследних (
»   ..... |»   »   &Дата,
»   ..... |»   »   Модель := ·&Модель
»   ..... |»   »   И·ТоварыИУслуги := ·&Номенклатура)·КАК·ЦеныСрезПоследних";
»
»   Запрос.УстановитьПараметр("Номенклатура", ·Номенклатура);
»   Запрос.УстановитьПараметр("Модель", ·Модель);
»   Запрос.УстановитьПараметр("Дата", ·Дата);
»
»   РезультатЗапроса := ·Запрос.Выполнить();
»
»   Если·НЕ·РезультатЗапроса.Пустой()·Тогда
»   »
»   »   Выборка := ·РезультатЗапроса.Выбрать();
»   »   Выборка.Следующий();
»   »   ЗаполнитьЗначенияСвойств(Результат, ·Выборка)
»   »
»   »   КонечЕсли;
»   »
»   Возврат ·Результат;
»
КонечФункции

```

Рисунок 34 – Лістинг коду функції

Також у документі є друк документа, який показано на рис. 35.

Оказание услуги						
Номер	00000003					
Дата	15.03.2023 13:10:19					
Магазин	Mobile.com					
Клиент	Дворник Сергей Николаевич					
Структура	Быстрый Платеж					
Список оплат	Бар					
№	Номенклатура	Партия	Набор характеристик	Количество	Цена	Сумма
1	Чехол с аккумулятором	Примодная накладная 000000003 от 15.03.2023 13:09:50		2	370.00	740.00
2	Чехол пленочный	Примодная накладная 000000002 от 15.03.2023 12:58:27		2	360.00	720.00
3	Зачепок защитного стекла			2	100.00	200.00
ВСЕГО						1 660

Рисунок 35 – Роздрукований документ

Запитом беремо данні документа, отримуємо області макета та виводимо їх послідовно.

При створенні нового документа йому буде встановлений новий унікальний номер. На рис. 37 видно, що до нього буду додаватись префікс, який буду потрібен для обміну даним між базами.

```

□ Процедура ПриУстановкеНовогоНомера (СтандартнаяОбработка, Префикс)
  »
  » Префикс := Обмен.ПолучитьПрефиксНомера ();
  »
  КонечПроцедуры
  
```

Рисунок 36 – Процедура, яка встановлює префікс до номера

Реалізована можливість створення документа на основі клієнта з форми самого клієнта.

При створенні документа буду зразу вказаний клієнт.

Також при проведенні документа, він буду робити рух по певним регістрам.

```

□ Процедура ОбработкаПроведения (Отказ, Режим)
  Движения.ОстаткиТоваров.Записывать = Истина;
  Движения.СтоимостьТоваров.Записывать = Истина;
  Движения.Продажи.Записывать = Истина;
  Движения.Управленческий.Записывать = Истина;
  МенеджерВТ = Новый МенеджерВременныхТаблиц;
  □ НоменклатураДокумента
  □ ДвиженияДокумента
  □ КонтрольОстатков
  КонечПроцедуры
  
```

Рисунок 37 – Процедура обробки проведення

На рис. 38 видно, що тут є три області у процедурі. Спочатку ми отримуємо данні номенклатури документа, які помістяться у тимчасову таблицю даних, яка подалі буде використана у кодї.

```

#Область НоменклатураДокумента
Запрос = Новый Запрос;
Запрос.МенеджерВременныхТаблиц = МенеджерВТ;

Запрос.Текст =
"ВЫБРАТЬ
| ОказаниеУслугиСоставУслуги.Номенклатура КАК Номенклатура,
| ОказаниеУслугиСоставУслуги.Номенклатура.Вид КАК ВидНоменклатуры,
| ОказаниеУслугиСоставУслуги.НаборСвойств КАК НаборСвойств,
| СУММА(ОказаниеУслугиСоставУслуги.Количество) КАК КоличествоВДокументе,
| СУММА(ОказаниеУслугиСоставУслуги.Сумма) КАК СуммаВДокументе,
| ОказаниеУслугиСоставУслуги.Партия КАК Партия,
| ОказаниеУслугиСоставУслуги.Модель КАК Модель
|ПОМЕСТИТЬ НоменклатураДокумента
|ИЗ
| Документ.ОказаниеУслуги.СоставУслуги КАК ОказаниеУслугиСоставУслуги
|ГДЕ
| ОказаниеУслугиСоставУслуги.Ссылка = &Ссылка
|
|СГРУППИРОВАТЬ ПО
| ОказаниеУслугиСоставУслуги.Номенклатура,
| ОказаниеУслугиСоставУслуги.Номенклатура.Вид,
| ОказаниеУслугиСоставУслуги.Партия,
| ОказаниеУслугиСоставУслуги.Модель,
| ОказаниеУслугиСоставУслуги.НаборСвойств";

Запрос.УстановитьПараметр("Ссылка", Ссылка);

РезультатЗапроса = Запрос.Выполнить();
#КонецОбласти

```

Рисунок 38 – Запит з отриманням номенклатури документа

На рис. 39 отримуються залишки по товарам документа и записуються рухи по регістрам.

Також реалізовано контроль залишків. Якщо не буде достатньо якогось товару, документ не проведеться та буде виведена помилка.

```

#Область КонтрольОстатков
//Если Режим = РежимПроведенияДокумента.Оперативный Тогда

Запрос3 = Новый Запрос;
Запрос3.МенеджерВременныхТаблиц = МенеджерВТ;
Запрос3.Текст = "ВЫБРАТЬ
| ОстаткиТоваров.Остатки.Номенклатура КАК Номенклатура,
| ОстаткиТоваров.Остатки.Модель КАК Модель,
| ОстаткиТоваров.Остатки.Партия КАК Партия,
| ОстаткиТоваров.Остатки.НаборСвойств КАК НаборСвойств,
| ОстаткиТоваров.Остатки.КоличествоОстаток КАК КоличествоОстаток,
| НоменклатураДокумента.КоличествоВДокументе КАК КоличествоВДокументе
| ИС
| НоменклатураДокумента КАК НоменклатураДокумента,
| РегистрНакопления.ОстаткиТоваров.Остатки(
| (Номенклатура, Модель, НаборСвойств) В
| (ВЫБРАТЬ
| НоменклатураДокумента.Номенклатура,
| НоменклатураДокумента.Модель,
| НоменклатураДокумента.НаборСвойств
| ИС
| НоменклатураДокумента)
| И Партия = «Партия» КАК ОстаткиТоваров.Остатки
| ГДЕ
| ОстаткиТоваров.Остатки.КоличествоОстаток < 0";
Запрос3.УстановитьПараметр("Партия", ВыборкаДетальныеЗаписи.Партия);
РезультатЗапроса = Запрос3.Выполнить();
ТЗ = РезультатЗапроса.Выгрузить();
ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.КоличествоВДокументе > ВыборкаДетальныеЗаписи.КоличествоОстаток Тогда

Отказ = Истина;

Если НЕ ЗначениеЗаполнено(ВыборкаДетальныеЗаписи.НаборСвойств)
ИЛИ ВыборкаДетальныеЗаписи.НаборСвойств = Неопределено Тогда
ТекстСообщения = "Не хватает " + Строка(ВыборкаДетальныеЗаписи.КоличествоОстаток)
+ " единиц товара "" + ВыборкаДетальныеЗаписи.Номенклатура + """";
Иначе
ТекстСообщения = "Не хватает " + Строка(ВыборкаДетальныеЗаписи.КоличествоОстаток)
+ " единиц товара "" + ВыборкаДетальныеЗаписи.Номенклатура + """"
+ " из набора свойств "" + ВыборкаДетальныеЗаписи.НаборСвойств + """";
КонецЕсли;

Сообщение = Новый СообщениеПользователю();
Сообщение.Текст = ТекстСообщения;
Сообщение.Сообщить();

КонецЕсли;

КонецЦикла;

```

Рисунок 39 – Область перевірки залишків

На рис. 39 отримаємо залишки товарів документа, якщо кількість у документі перевищить кількість реєстру залишків, то виведеться помилка користувачу.

Документ «Нарахування співробітникам».

Потрібен для нарахування заробітної платні працівникам.

N	Сотрудник	График работы	Дата начала	Дата окончания	Вид расчета	Начислено
1	Ечин Даниил	График остальных	01.05.2024	31.05.2024	Оклад	
2	Хлюстин Денис	График остальных	01.05.2024	31.05.2024	Оклад	
3	Молочко Даниил	График администрации	01.05.2024	31.05.2024	Оклад	

Рисунок 40 – Форма елемента документа «Нарахування співробітникам»

Тут заповнюються працівники, графік їх роботи, дата, вид розрахунку та суму нарахування.

Також при проведенні, документ утворює запис у реєстрі розрахунків «Нарахування».

4.3 Перерахування

Перерахування утворені у конфігурації:

- ⊖ {...} Перечисления
- ⊕ {...} ДниНедели
- ⊕ {...} СпособОплаты
- ⊕ {...} ВидыНоменклатуры

Рисунок 41 – Перерахування

Перерахування «Дні неділі» застосовані у довіднику «Магазини» щоб відобразити в які дні працюють магазини.

☆ Mobile.com (Магазин) 🔗 ⋮ □ ×

Записать и закрыть Записать Еще ▾

Код:

Наименование:

Номер телефона:

Адрес:

<input checked="" type="checkbox"/>	День недели	Время начала	Время окончания
<input checked="" type="checkbox"/>	Понедельник	09:00	19:00
<input checked="" type="checkbox"/>	Вторник	09:00	19:00
<input checked="" type="checkbox"/>	Среда	09:00	19:00
<input checked="" type="checkbox"/>	Четверг	09:00	19:00
<input checked="" type="checkbox"/>	Пятница	09:00	19:00
<input type="checkbox"/>	Суббота	09:00	19:00
<input type="checkbox"/>	Воскресенье	09:00	19:00

Рисунок 42 – Приклад використання перерахування

Перерахування «Способи оплати».

Це перерахування потрібно для документа «Надання послуг», щоб відобразити який буде розрахунок.

Оказание услуги 000000002 от 05.06.2024 13:02:06

Основное [Остатки товаров](#) [Стоимость товаров](#) [Управленческий](#)

Провести и закрыть Записать Провести Печать

Номер: 000000002

Дата: 05.06.2024 13:02:06

Магазин: Mobile.com

Клиент:

Сотрудник: Хлюстин Денис

Способ оплаты: Карта

Добавить

Наличные

Карта

Номенклатура Набор свойств

Рисунок 43 – Приклад використання перерахування

Перерахування «Види номенклатури»

Потрібен для розрізу номенклатури. Застосовується у довіднику «Номенклатура».

Ирhone 12 (Номенклатура)

Записать и закрыть Записать Еще

Код: 00000000003

Наименование: Ирhone 12

Родитель: Товары

Вид: Товар

Товар

Услуга

Рисунок 44 – Приклад використання перерахування

4.4 Звіти

Звіти – це прикладні об'єкти конфігурації. Вони призначені для обробки накопиченої інформації та отримання зведених даних у зручному для перегляду та аналізу вигляді.

В конфігурації реалізовані такі звіти:

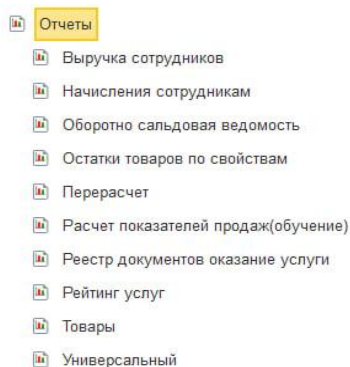


Рисунок 45 – Звіти конфігурації

Усі звіти зроблені за допомогою СКД [12]. СКД призначена для створення довільних звітів та складається з кількох основних елементів. Вихідні дані для компонування звіту містить у собі схема компонування даних. Це набори даних [13] та методи роботи з ними.

Розробник створює схему компонування даних, у якій описує текст запити, набори даних, зв'язок між ними, доступні поля, параметри отримання даних, та задає початкові налаштування компонування – структуру звіту, макет оформлення даних та ін.

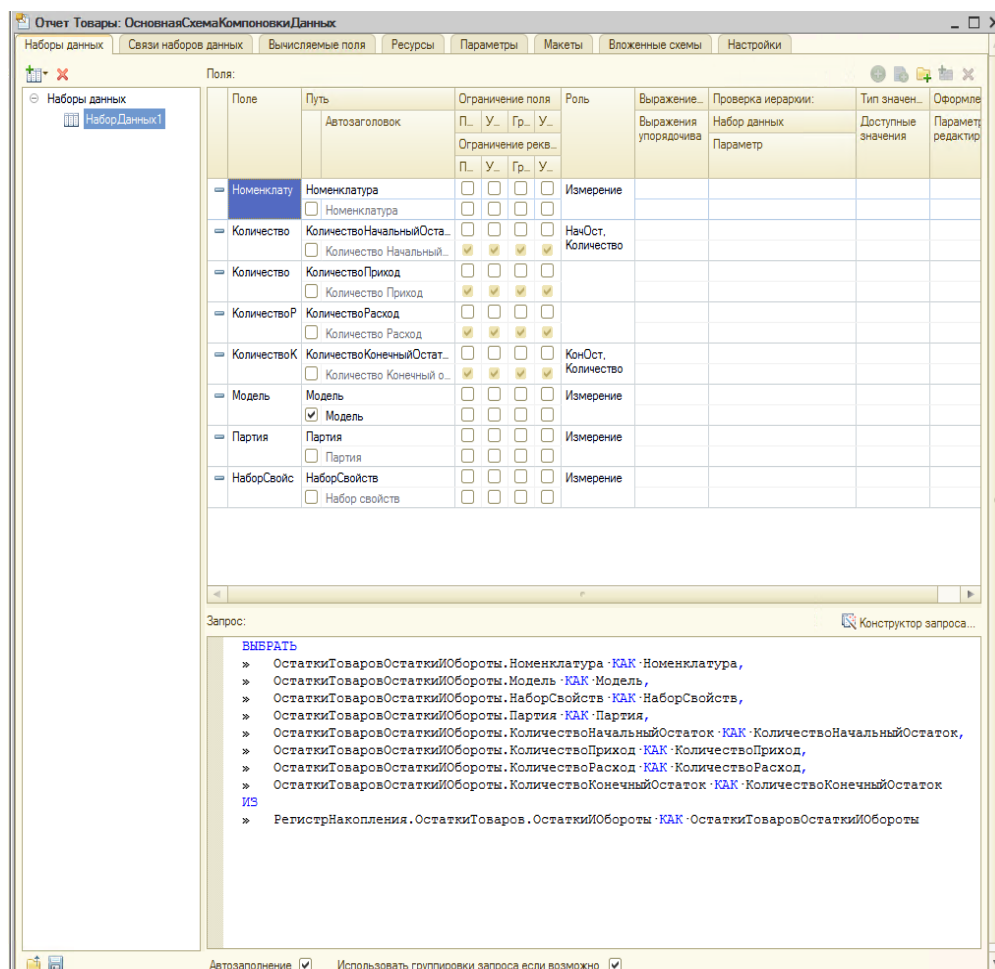


Рисунок 46 — Приклад схеми компоновки (набір даних та запит)

На рис. 47 показано вікно конструктора схеми компоновки даних, в якому міститься джерело даних, текст запиту та поля, вибрані запитом. Звіт системи компонування даних, який отримає користувач, є не просто таблицею записів, що задовольняють умовам запиту. Звіт системи компонування має складну ієрархічну структуру і може складатися з різних елементів, таких як угруповання, таблиці та діаграми.

При цьому користувач може змінити існуючу структуру звіту або взагалі створити нову структуру звіту. Може налаштувати необхідний відбір, оформлення елементів структури звіту, отримати розшифрування по кожному елементу і т.д.

☆ Начисления сотрудникам

Сформировать | Выбрать вариант... | Настройки... | Перерассчитать

Начисления сотрудникам

Сотрудник	Вид расчета	Начало	Окончание	Регистратор	Исходные данные
Ечин Даниил	Оклад	01.05.2024 0:00:00	31.05.2024 23:59:59	Начисление сотрудникам 1 от 05.06.2024 12:10:29	15 000,00
Молочко Даниил	Оклад	01.05.2024 0:00:00	31.05.2024 23:59:59	Начисление сотрудникам 1 от 05.06.2024 12:10:29	30 000,00
	Премия	01.05.2024 0:00:00	31.05.2024 23:59:59	Начисление сотрудникам 2 от 05.06.2024 12:10:47	3 000,00
Хлюстин Денис	Оклад	01.05.2024 0:00:00	31.05.2024 23:59:59	Начисление сотрудникам 1 от 05.06.2024 12:10:29	20 000,00
Итого					68 000,00

Рисунок 47 — Приклад сформованого звіту

Цей звіт має в собі данні з РР «Нарахування». А в цей реєстр попадають данні з документів «Нарахування співробітникам».

Звіт «Виручка майстрів» бере інформацію реєстра «Продажи», який в собі містить данні документів «Надання послуг». На рисунку можна побачити скільки кожен співробітник приніс виручки до компанії.

Виручка мастеров

Сотрудник	Период	Выручка
Итого		23 400,00
Ечин Даниил		23 000,00
	05.06.2024 0:00:00	23 000,00
		23 000,00
Хлюстин Денис		400,00
	05.06.2024 0:00:00	400,00
		400,00

Рисунок 48 — Приклад звіту «Виручка майстрів»

Звіт «Залишки товарів» існує для відображення інформацію про поточний залишок товарів. Данні беруться з РН «Залишки товарів».

Партия	Номенклатура	Модель	Набор свойств	Количество Начальный остаток	Количество Приход	Количество Расход	Количество Конечный остаток
Приходная накладная 000000001 от 24.04.2024 15:56:50	Мак Бук				100		100
Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 12				100	1	99
Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 13				100		100
Итого					300	1	299

Рисунок 49 — Пример звіту «Залишки товарів»

Звіт «Реєстр документів надання послуг» служить для відображення інформації який документ, в якому магазині та яким працівником був опрацьований.

Магазин	Документ	Сотрудник
Mobile.com	Оказание услуги 000000001 от 05.06.2024 13:01:10	Ечин Даниил
Mobile.com	Оказание услуги 000000002 от 05.06.2024 13:02:06	Хлюстин Денис

Рисунок 50 — Пример звіту «Реєстр документів надання послуг»

4.5 Плани видів характеристик

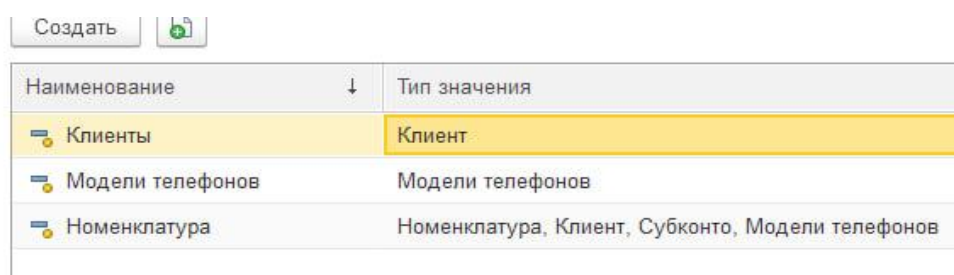
В даній конфігурації існують такі ПВХ:

- ☐ Плани видов характеристик
- ☐ Виды субконто
- ☐ Свойства товаров

Рисунок 51 — ПВХ конфігурації

ПВХ «Види субконто».

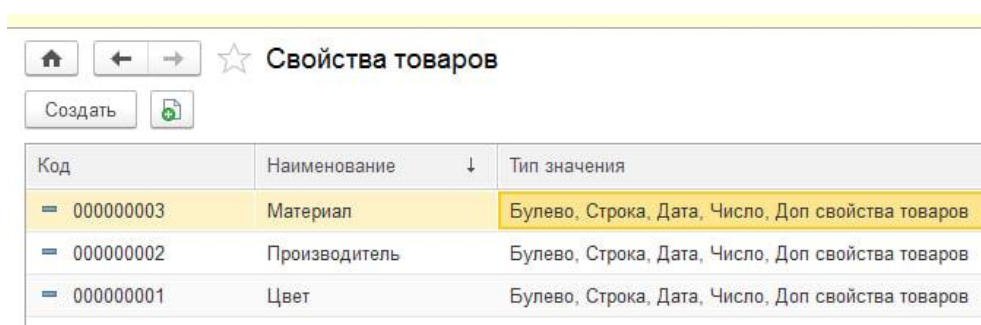
Бухгалтерський облік, як правило, має на увазі ведення аналітичного обліку на більшості рахунків. Для позначення розрізів аналітичного обліку використовуватиметься термін "Види субконто". Тобто на кожному рахунку облік може вестись у розрізі кількох видів субконто. А для позначення конкретних об'єктів аналітичного обліку використовуватиметься термін субконто. Наприклад, на 41 рахунку (Товари) облік ведеться зазвичай у розрізі Номенклатури та склади, які є видами субконто.



Наименование	Тип значения
Клиенты	Клиент
Модели телефонов	Модели телефонов
Номенклатура	Номенклатура, Клиент, Субконто, Модели телефонов

Рисунок 52 — ПВХ «Види субконто»

ПВХ «Властивості номенклатури».



Код	Наименование	Тип значения
000000003	Материал	Булево, Строка, Дата, Число, Доп свойства товаров
000000002	Производитель	Булево, Строка, Дата, Число, Доп свойства товаров
000000001	Цвет	Булево, Строка, Дата, Число, Доп свойства товаров

Рисунок 53 — ПВХ «Властивості номенклатури»

ПВХ потрібен для додаткового опису товару. Наприклад колір товару, з якого матеріалу зроблено або хто постачальник чи виробник.\

У кожної номенклатури свої окремі властивості. Щоб їх побачити, треба обрати номенклатуру та натиснути на «Варіанти товарів».



☆ Iphone 12 (Номенклатура) [ссылка] [меню] [окно] [закрыть]

Основное **Варианты товаров**

Записать и закрыть Записать [Еще ▾]

Код: 00000000003

Наименование: Iphone 12

Родитель: Товары [выбор] [ссылка]

Вид: Товар [выбор]

Рисунок 54 – Довідник «Номенклатура»



☆ Iphone 12 (Номенклатура) [ссылка] [меню] [окно] [закрыть]

Основное **Варианты товаров**

Варианты товаров

Создать [ссылка]

Поиск (Ctrl+F) [поиск] [Еще ▾]

Наименование	Владелец
Белый телефон	Iphone 12

[стрелки]

Рисунок 55 – Довідник «Варіанти товарів»

Відкривається варіанти нашого товару, обираємо потрібний варіант. Відкриється форма елемента. Далі натискаємо на «Склад варіантів номенклатури».

☆ Белый телефон (Варианты товаров)

Основное **Состав варианта номенклатуры**

Записать и закрыть Записать

Наименование: Белый телефон


Владелец: Iphone 12

Рисунок 56 – Форма элементу варианту товару

☆ Белый телефон (Варианты товаров) 🔗 ⋮ □ ×

[Основное](#) Состав варианта номенклатуры

Состав варианта номенклатуры

Создать  Поиск (Ctrl+F) × 🔍 Еще ▾

Свойство	Значение
🔗 Цвет	Белый

⏪ ⏩ ⏴ ⏵

Рисунок 57 – Значення ПВХ для нашого товару

4.6 Плани рахунків

Об'єкт конфігурації План рахунків призначений для опису структури зберігання інформації про сукупність синтетичних рахунків підприємства, які створені для угруповання даних про його господарської діяльності.

На основі об'єкта конфігурації План рахунків платформа створює у базі даних таблиці, в яких зберігатиметься інформація про те, які рахунки та яким

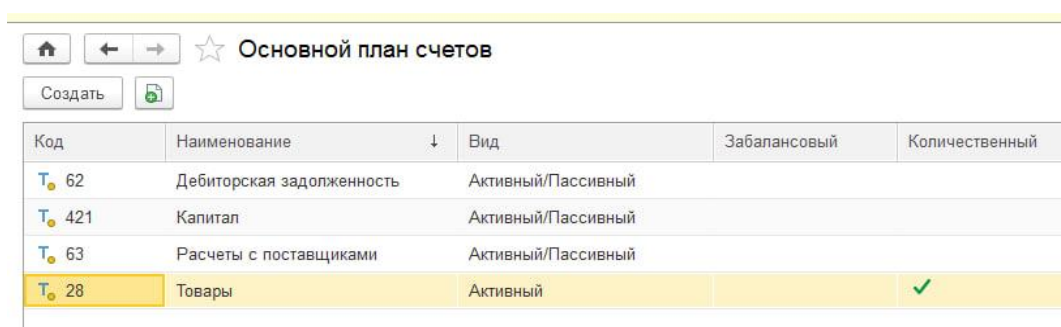
чином використовуватиме підприємство. Це може бути система бухгалтерських рахунків, встановлена державою, план управлінських рахунків чи довільний набір рахунків, що використовуються для аналізу тих чи інших видів діяльності підприємства.

План рахунків у системі «ВAF» підтримує ієрархію субрахунків: до кожного рахунку першого рівня може бути відкрито кілька субрахунків, які, у свою чергу, можуть мати свої субрахунки, і тд.

За будь-яким рахунком чи субрахунком може вестись аналітичний облік у розрізі субконто, описаних у плані видів показників. Зв'язок між планом рахунків і планом видів характеристик задається розробником на етапі конфігурування.

Для опису субконто система створює в плані рахунків спеціальну табличну частину ВидиСубконто, яка не видно у конфігураторі (але доступна засобами вбудованої мови). Для кожного рахунку можна задати кілька видів обліку (наприклад, кількісний та валютний). Види обліку задаються при допомозі підлеглих об'єктів зміни ознака обліку.

Також існує можливість визначити декілька видів обліку субконто (наприклад, сумовий, валютний чи кількісний). Види обліку субконто задаються за допомогою підлеглих об'єктів Зміна ознака обліку субконто.



Код	Наименование	↓ Вид	Забалансовый	Количественный
Т. 62	Дебиторская задолженность	Активный/Пассивный		
Т. 421	Капитал	Активный/Пассивный		
Т. 63	Расчеты с поставщиками	Активный/Пассивный		
Т. 28	Товары	Активный		✓

Рисунок 58 — План рахунків

4.7 План видів розрахунку та реєстр розрахунку

Об'єкт конфігурації План видів розрахунку призначений для опису структури зберігання інформації про можливі види розрахунків. За підсумками об'єкта конфігурації План видів розрахунку платформа створює у базі даних таблицю, у якій зберігатиметься інформація про те, які існують види розрахунку і які взаємодії зв'язку між ними.

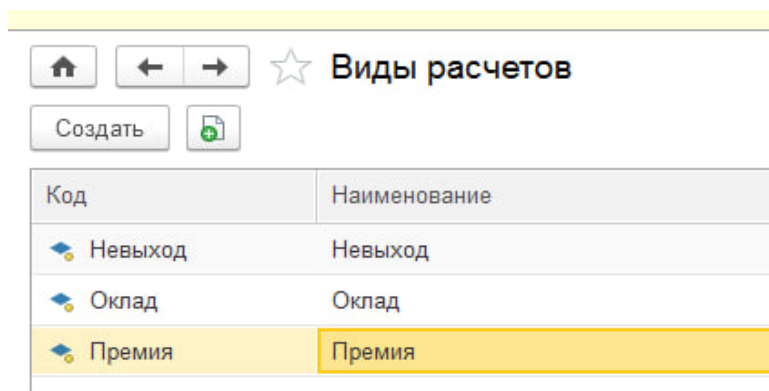
Відмінною особливістю плану видів розрахунку є те, що користувач у процесі роботи може додавати нові види розрахунку. Така можливість робить механізм періодичних розрахунків більш гнучким і дозволяє користувачеві створювати власні види розрахунку, окрім тих, які задані розробником як зумовлені.

Об'єкт конфігурації План видів розрахунку має властивість «Використовує період дії». З його допомогою визначається, чи будуть у цьому плані знаходитись види розрахунку, які можуть бути витіснені за періодом дії. Якщо ця властивість встановлена, то розробник отримує можливість вказати для кожного виду розрахунку ті види, які витісняють його за періодом дії.

Наступною важливою властивістю об'єкта конфігурації План видів Розрахунком є залежність від бази. Воно визначає, чи будуть у цьому плані перебувати залежні за базовим періодом види розрахунку. Якщо ця властивість встановлена, з'являється можливість вказати, у якому плані видів розрахунку будуть базові види розрахунку і, крім цього, як визначатиметься ця залежність.

Ще однією важливою особливістю плану видів розрахунку є можливість створення зумовлених видів розрахунку та опису їхнього взаємного впливу. При цьому у загальному випадку розробник має можливість вказати три категорії видів розрахунку, що впливають на зумовлений вид розрахунку:

- Базові – їх результати мають бути використані під час перерахунку цього виду розрахунку;
- Витісняючі – витісняють цей вид розрахунку за періодом дії;
- Ведучі – зміна їх результатів має призводити до необхідності перерахунку цього виду розрахунку.



Код	Наименование
Невыход	Невыход
Оклад	Оклад
Премия	Премия

Рисунок 59 — План видів розрахунку

4.8 Регістри розрахунків

Об'єкт конфігурації Регістр розрахунку призначений для опису структури накопичення даних, що є результатами розрахунків. На основі об'єкта конфігурації Регістр розрахунку платформа створює у базі даних таблиці, в яких накопичуватимуться дані, формовані різними об'єктами бази даних.

Відмінною особливістю реєстру розрахунку є те, що він не призначено для інтерактивного редагування користувачем. Розробник може за необхідності надати користувачеві можливість редагувати реєстр розрахунку, але призначення реєстру розрахунку у тому, щоб його модифікація проводилася з урахуванням алгоритмів роботи об'єктів бази даних, а чи не внаслідок безпосередніх дій користувача.

Як і інші реєстри, реєстр розрахунку має ресурси, у яких зберігає числові дані; має виміри, у розрізі яких можна отримувати значення ресурсів реєстру; має реквізити, які характеризують кожен запис реєстру розрахунку. Відмінними ж особливостями реєстру розрахунку є його періодичність, можливість використання механізмів витіснення за періодом дії та залежності за базовим періодом, а також зв'язок із планом видів розрахунку. Розглянемо ці особливості по порядку.

🏠 ⏪ ⏩ ☆ Движения в регистре Начисления

Период регистрации	Регистратор	Номер строки	Вид расчета	Сторно	Сотрудник	Результат	График работы
01.06.2024 0:00:00	Начисление сот...	1	Оклад		Ечин Даниил		График остальных
01.06.2024 0:00:00	Начисление сот...	2	Оклад		Хлюстин Денис		График остальных
01.06.2024 0:00:00	Начисление сот...	3	Оклад		Молочко Даниил		График администрации
01.06.2024 0:00:00	Начисление сот...	1	Премия		Молочко Даниил		График администрации

Рисунок 60 — Рух документів у реєстрі розрахунку «Нарахування»

У даному реєстрі зроблено вимір «Співробітник», ресурс «Результат» та реквізити «Графік роботи» та «Вихідні дані». Вимір «Співробітник» зсилається на довідник «Співробітники», ресурс «Результат» має тип число.

Реквізит «Графік роботи» зсилається на перелічення «Графіки співробітників», а реквізит «Вихідні дані» має тип число. При проведенні документів які зв'язані з реєстром буде утворений запис у реєстрі, виміри, ресурси та реквізити якого будуть заповнені з даних документа.

4.9 Регістри накопичень

В моїй конфігурації реалізовані такі РН:

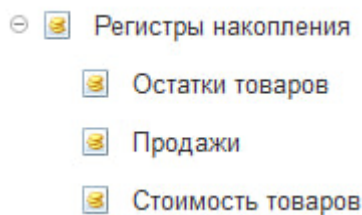


Рисунок 61 — РН конфігурації

РН «Залишки товарів»

☆ Движение по регистру Остатки товаров

Период	Регистратор	Номенклатура	Модель	Набор свойств	Количество
- 05.06.2024 13:01:10	Оказание услуги 000000001 от 05.06.2024 13:01:10	Iphone 12			
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Мак бук			
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 12			
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 13			

Рисунок 62 — Форма руху документів реєстра

Даний реєстр має виміри: Номенклатура - посилання на довідник «Номенклатура», Регістратор – це документ який створив даний запис у реєстрі, Модель – посилання на довідник «Моделі телефонів» та Набір властивостей – посилання на довідник «Варіанти товарів». Ресурси: Кількість, тип – число.

Також даний реєстр має документи реєстратори, які утворюють записи. Це «Надання послуг» та «Прибуткова накладна». Як я показав раніше, цей реєстр заповнюється даними документа.

РН «Вартість товарів»

Период	Регистратор	Товар	Модель	Стоимость
+ 05.06.2024 13:01:10	Оказание услуги 000000001 от 05.06.2024 13:01:10	Iphone 12		15 000,00
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 12		1 500 000,00
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Iphone 13		1 700 000,00
+ 24.04.2024 15:56:50	Приходная накладная 000000001 от 24.04.2024 15:56:50	Мак бук		2 000 000,00

Рисунок 63 — Форма руху документів реєстра

Даний реєстр має виміри: Номенклатура - посилання на довідник «Номенклатура», Регистратор – це документ який створив даний запис у реєстрі, Модель – посилання на довідник «Моделі телефонів». Ресурси: Вартість, тип – число.

На рис 64 видно, що даний реєстр має документи регістратори, які утворюють записи. Це «Надання послуг» та «Прибуткова накладна». Як я показав раніше, цей реєстр заповнюється даними документа.

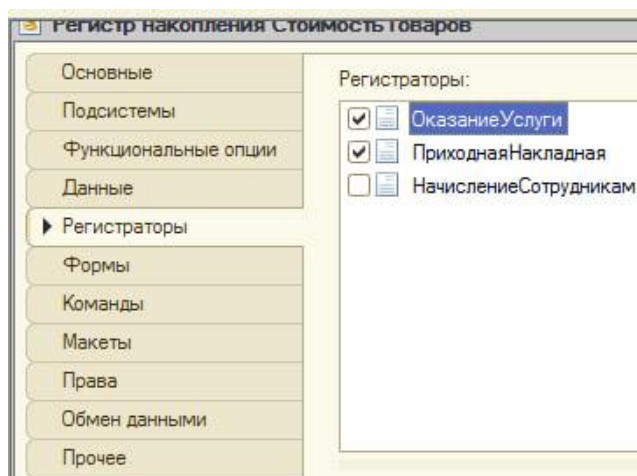
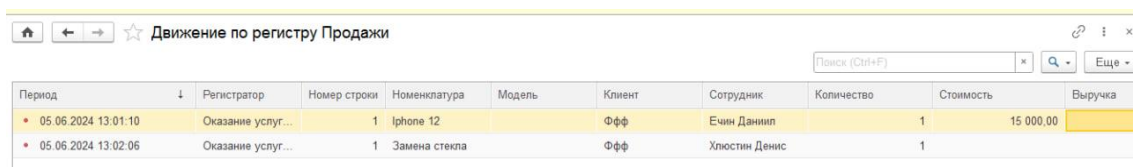


Рисунок 64 — Документи регістратор РН «Вартість товарів»

РН «Продажі»



Период	Регистратор	Номер строки	Номенклатура	Модель	Клиент	Сотрудник	Количество	Стоимость	Выручка
05.06.2024 13:01:10	Оказание услуг...	1	Iphone 12		Ффф	Ечин Даниил	1	15 000,00	
05.06.2024 13:02:06	Оказание услуг...	1	Замена стекла		Ффф	Хлюстин Денис	1		

Рисунок 65 — Форма руху документів реєстра

Даний реєстр має виміри: Номенклатура - посилання на довідник «Номенклатура», Регистратор – це документ який створив даний запис у реєстрі, Модель – посилання на довідник «Моделі телефонів», Клиент – посилання на довідник «Клієнти» та Співробітник – посилання на довідник «Співробітники». Ресурси: Кількість, тип – число, Вартість та Виручка теж мають тип - число.

На рис. 66 видно, що даний реєстр має єдиний документ регистратор, який утворює записи. Це «Надання послуг».

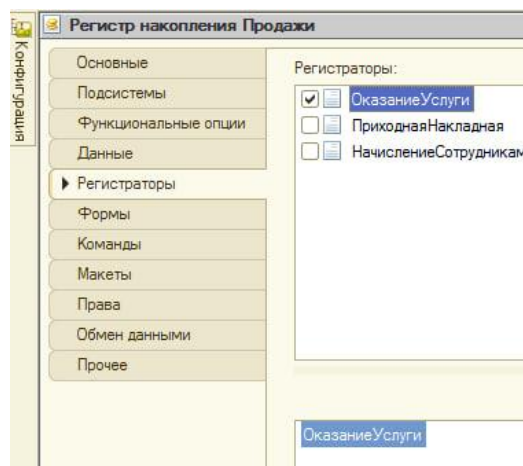


Рисунок 66 — Документи регистратор РН «Продажі»

4.10 Регістри бухгалтерії

Об'єкт конфігурації Регістр бухгалтерії призначений для опис структури накопичення даних, облік яких ведеться виходячи з деякого плану

рахунків. За підсумками об'єкта конфігурації Регістр бухгалтерії платформа створює у базі даних таблицю, в якій накопичуватимуться дані про господарські операції, відображаються у бухгалтерському обліку. За своїм видом реєстр бухгалтерії нагадує реєстр на-плювання – він також має ресурси, може мати виміри та реквізити. Вимірювання дозволяють розділяти ведення обліку (наприклад, використовуючи вимір Організація, можна вести облік у розрізі кількох юридичних осіб). Реквізити є ознакою, за якою одні записи реєстру можна відокремити від інших (наприклад, як реквізит може використовувати номер журналу, що дозволить відбирати проводки, мають однаковий зміст).

Значна відмінність від реєстру накопичення [6] у тому, що реєстр бухгалтерії має жорсткий зв'язок із використовуваним планом рахунків. Тому кожен запис реєстра бухгалтерії містить додаткові поля, що визначаються налаштуванням використовуваного плану рахунків.

Наприклад, запис реєстру може містити додаткові поля для зазначення кореспондуючих рахунків, сум, об'єктів аналітичного обліку (субконто), кількості, виду валюти тощо. Крім цього, відмінною рисою реєстра бухгалтерії є можливість підтримки механізму подвійного запису, за якого кожен запис реєстру містить обов'язкові поля для вказівки рахунки дебету та рахунки кредиту.

В конфігурації реалізовано єдиний РБ – «Управлінський»

🏠 ⏪ ⏩ ☆ Движение в регистре Управленческий

Період	↓	Реєстратор	Номер рядку	Рахунок Дт	Субконто1 Дт	Субконто2 Дт	Субконто3 Дт
At Кр	24.04.2024 15:56:50	Приходная накл...	1	28	Мак бук		
At Кр	24.04.2024 15:56:50	Приходная накл...	2	28	Iphone 12		
At Кр	24.04.2024 15:56:50	Приходная накл...	3	28	Iphone 13		

Субkonto3 Дт	Колічество Дт	Рахунок Кт	Субkonto1 Кт	Субkonto2 Кт	Субkonto3 Кт	Колічество Кт	Сумма
	100,000	63					2 000 000,00
	100,000	63					1 500 000,00
	100,000	63					1 700 000,00

Рисунок 67 — Форма руху документів реєстра

Реєстр має тільки ресурси: Сума з типом число та кількість. Реєстратором для РБ виступають документи «Надання послуг» та «Прибуткова накладна».

4.11 Регістри відомостей

Регістри відомостей конфігурації:





- ☐  Регістри відомостей
 -  Графики работы
 -  Значения свойств товаров
 -  Цены

Рисунок 68 — РВ конфігурації

РВ «Ціни» потрібен для збереження інформації про товар.

Период	Товары и услуги	Модель	Набор свойств
01.06.2024	Мак бук		
01.06.2024	Iphone 12		
01.06.2024	Iphone 13		

Цена
40 000,00
20 000,00
30 000,00

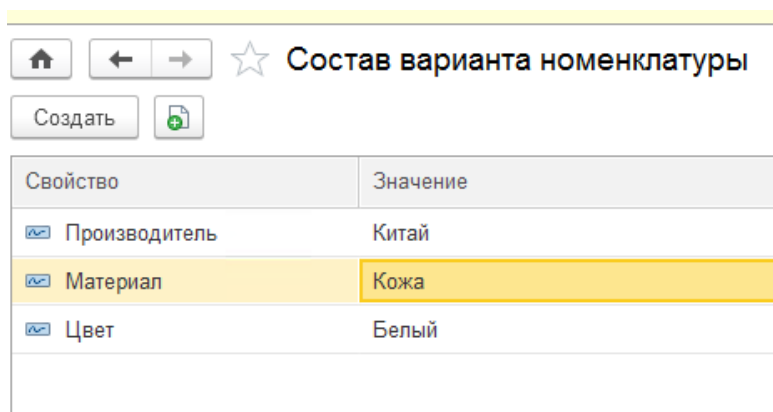
Рисунок 69 – Форма списка реєстру

При створенні запису треба вказати товар для якого буде вказана ціна та саму ціну. Також можна вказати модель товару, якщо така є та набір властивостей.

Рисунок 70 – Форма елемента РВ «Ціни»

РВ «Значення властивостей товарів»

Регістр потрібен для збереження значень властивостей ПВХ.



Свойство	Значение
Производитель	Китай
Материал	Кожа
Цвет	Белый

Рисунок 71 – Форма реєстру РВ «Значення властивостей товарів»

4.12 План обміну

Механізми обміну [8] даними «BAF» дозволяють організувати обмін інформацією, що зберігається в базі даних, з іншими програмними системами. Як такі системи можуть виступати як інші інформаційні бази «BAF» (мають аналогічну або відмінну конфігурацію), так і програмні системи, які не засновані на "BAF".

Така гнучкість обміну даними досягається за рахунок того, що засоби обміну даними «BAF» можуть використовуватись у різних комбінаціях. Крім цього, формат обміну даними заснований на мові XML, що є на сьогоднішній день загальноприйнятим засобом представлення даних.

Для того, щоб існувала можливість обміну будь-якими даними з будь-ким, необхідно певним чином ідентифікувати тих, з ким ми обмінюватимемося, і для кожного з них описати Список обміну. Обидві ці завдання дозволяє вирішувати об'єкт конфігурації План обміну. Подібно до того, як елементами даних довідника є елементи довідника, елементами даних плану обміну є вузли план обміну. Кожен вузол ідентифікує учасника обміну

за цим планом обміну. Крім цього, у кожному плані обміну завжди існує один зумовлений вузол, що ідентифікує цю інформаційну базу.

В одній конфігурації може існувати кілька планів обміну. Кожен план обміну визначає набір даних, яким здійснюватиметься обмін у рамках цього плану, і сам механізм цього обміну. Наявність кількох планів обміну може знадобитися, якщо з різними вузлами ведеться обмін різним складом даних, або коли схема організації обміну з одними вузлами відрізняється від схеми організації обміну коїться з іншими вузлами.

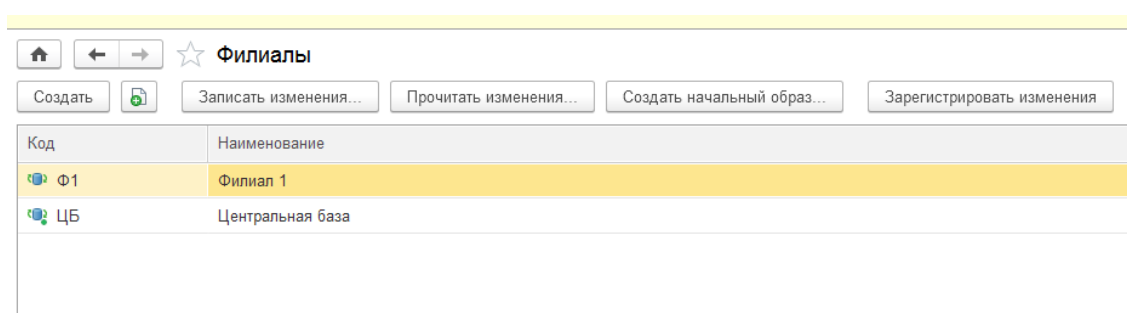


Рисунок 72 – План обміну конфігурації

При натисканні на кнопку «Записати змінення» починає записувати новостворенні або змінені об'єкти у файл.

В процедурі спочатку оголошується користувачу, що починається вивантаження [9] до вузла який був вибран. Далі створюється файл XML, починається запис, користувачу доповідається про процес вивантаження. Отримуються данні змінень та в циклі записуються. Коли вивантаження завершена, користувачу виводиться повідомлення.

Далі отримується каталог тимчасових файлів [10], де шукається потрібний файл, якщо такого файлу немає, виведеться помилка про відсутність файлу. Далі починається зачитування даних у циклі.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи бакалавра було розроблена конфігурація для призначеної для автоматизації ключових бізнес-процесів та підвищення ефективності роботи компанії, що спеціалізується на продажі телефонів, чохлів, аксесуарів, гаджетів та надання супутніх послуг. Також були виконані поставлені завдання, такі як: визначення вимог до конфігурації, розробка та впровадження конфігурації на платформі VAF, тестування та оцінка ефективності впровадженої системи та розробка модулів та інтерфейсів для користувачів.

Платформа [11] була обрана тому що має широкий функціонал. Вона дозволяє вирішити безліч завдань, пов'язаних з управлінням клієнтською базою та продажами. Також VAF відрізняється простотою та зручністю використання. Вона має інтуїтивно зрозумілий інтерфейс, що дозволяє швидко освоїтися навіть недосвідченим користувачам. Завдяки зручній навігації та зрозумілим пунктам меню, робота з VAF стає легкою та приємною. Компаніям не потрібний тривалий і складний процес навчання співробітників, що допомагає заощадити час та ресурси.

Код був написан на вбудованій мові, яка нагадує SQL та C#. Сама на цих мовах була написана сама програма. В конфігурації було реалізовано взагалом усе необхідне для роботи у базі. На даний момент користувачі можуть спокійно вносити данні, заповняти ціни та проводити бухгалтерські операції. Також реалізован план обміну, якщо клієнт захоче мати дві або більше баз. Дана конфігурація безкоштовна, але всі інші типові конфігурації платні, бо мають більший функціонал. Але не дивлячись на це, дану конфігурацію можна доробити, якщо на те є потреба.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Що таке BAF https://uk.wikipedia.org/wiki/Business_Automation_Framework (дата звернення 25.05.2024)
2. Рішення BAS <https://www.bas-soft.eu/> (дата звернення 26.05.2024)
3. Програма BAS для автоматизації бізнесу <https://aktiv.ua/ua/bas> (дата звернення 30.05.2024)
4. Платформа BAF <https://ingenum.ua/platforma-baf> (дата звернення 01.06.2024)
5. Огляд BAF <https://kamala-soft.com/uk/blog/chto-takoe-platforma-business-automation-framework-baf/> (дата звернення 02.06.2024)
6. Інструкція з використання зовнішньої обробки для облікової системи <https://medoc.ua/files/uploads/bc80577c8618c40b95dac304fb0230b6.pdf> (дата звернення 02.06.2024)
7. Встановлення програми BAS та створення інформаційної бази <https://lopan.com.ua/instrukcii/bas/ustanovka-baf-i-sozdanie-informacionnoy-bazi/> (дата звернення 03.06.2024)
8. Цікаві матеріали для програмістів BAF <https://csoprocom.com.ua/info/ed-mat/programming/> (дата звернення 07.06.2024)
9. Платформи BAF <https://www.youtube.com/watch?v=g-tbdYdbkJo> (дата звернення 08.06.2024)
10. Курс програмування <https://medias.com.ua/trivaje-nabir-na-bezkostovnij-kurs-programuvannia-baf-business-automation-framework-vid-kompaniyi-media-servis> (дата звернення 07.06.2024)
11. Радченко Максим Григорович, Хрустальова Олена Юріївна , BAF. Практичний посібник розробника, 2013, 964 ст.
12. Ажеронок В.А., Островерх О. В., Радченко М. Г., Хрустальова Є. Ю , Розробка керованого інтерфейсу, 2010, 728 ст.