

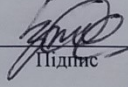
КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

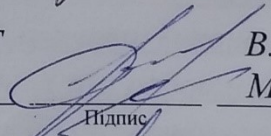
на тему Розробка програмного забезпечення «Облік товарів магазину»

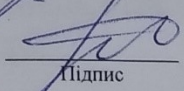
Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки
Назва освітньої програми

Виконав: студентка 4 курсу, група КН-17-2  Ю.В. Серветник
Курс, група виконавця Підпис Ініціали, прізвище

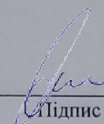
Керівник: к.фіз.-мат.н., доцент кафедри КНІТ  В.Д.
Міхалевський
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КНІТ  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КНІТ, д.т.н., професор

08 червня 2021 р.

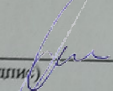

Підпис

О.В. Бармак
Ініціали, прізвище

Уведіть текст т

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерних наук та інформаційних технологій
Освітній ступінь бакалавр
Галузь знань 12 – Інформаційні технології
Спеціальність 122 – Комп'ютерні науки

Затверджую
Завідувач кафедри комп'ютерних наук та інформаційних технологій


(підпис)
д.т.н., професор О.В. Бармак
«08» лютого 2021 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: Розробка програмного забезпечення «Облік товарів магазину»

2. Завдання видано студентці Серветник Юлії Віталіївни

(прізвище, ім'я, по батькові)

3. Керівник роботи доцент кафедри КНІТ Міхалевський Віталій Цезарійович

(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від «05» лютого 2021 р. № 11

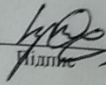
5. Зміст пояснювальної записки (перелік задач) та вихідні дані:

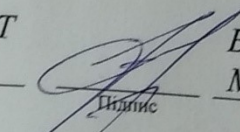
Мета роботи – розробка інформаційної системи автоматизованого обліку товарів магазину. При формуванні обліку слід враховувати параметри товарів, які прописуються в прихідній накладні. Слід забезпечити виконання функцій внесення, редагування та перегляду всіх необхідних для роботи системи даних, й одержання наступних результатів роботи системи: автоматизований облік товарів, надходження та видача товарів та послуг.

Виконавець: студентка 4 курсу, група КН-17-2
Курс, група виконавця

Керівник: к.фіз.-мат.н., доцент кафедри КНІТ

Науковий ступінь, посада


Підпис Ю.В. Серветник
Ініціали, прізвище


Підпис В.Ц. Міхалевський
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: Розробка програмного забезпечення «Облік товарів магазину»

Виконавець кваліфікаційної роботи бакалавра: студентка групи КН-17-2 Серветник Юлія Віталіївна

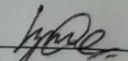
Керівник кваліфікаційної роботи бакалавра: к.фіз.-мат.н., доцент кафедри КНІТ Міхалевський Віталій Цезарійович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
72	30	6	13	1

Метою кваліфікаційної роботи бакалавра є розробка інформаційної системи автоматизованого обліку товарів в магазині або складі. Для розробки інформаційної системи було використано мову програмування Java, а також систему керування базами даних PostgreSQL.

Розроблена система призначена для менеджерів або працівників магазинів або складів. Реалізована автоматизація обліку товарів на складі дозволяє підвищити ефективність відповідної діяльності як працівників фірм, так і споживачів та клієнтів.

Виконавець: студентка 4 курсу, група КН-17 2  Ю.В. Серветник

Курс, група виконавця

Підпис

Ініціали, прізвище

Зміст

Вступ.....	3
Розділ 1 Характеристика предметної області та постановка задачі	4
1.1 Аналіз предметної області	4
1.2 Аналіз існуючого програмного забезпечення предметної області	6
1.2.1 Аналіз сучасних засобів створення програмного забезпечення	11
1.3 Постановка задачі	16
Розділ 2 Проектування структури інформаційної системи.....	18
2.1 Аналіз та автоматизація обробки інформаційних потоків.....	18
2.2 Розробка структури інформаційної системи	19
2.3 Вибір засобів розробки інформаційної системи	25
2.3.1 Вибір мови програмування	25
2.3.2 Вибір фреймворку.....	26
2.3.3 Вибір редактора програмного коду.....	27
2.3.4 Вибір СКБД	28
Розділ 3 Програмна реалізація	29
3.1 Структура та функціональне призначення складових системи	29
3.2 Розробка програмних модулів.....	33
3.3 Тестування інформаційної системи	33
3.4 Інструкція користувача.....	41
3.5 Вимоги до апаратних та програмних засобів.....	50
Висновки	51
Перелік посилань.....	52
Додатки	

ВСТУП

В сучасних умовах інформація – це один з основних товарів.

Без інформації, в сучасних умовах, в умовах епідемії, практично неможливо прийняти об'єктивного управлінського рішення, яке, зазвичай, є одним з найважливіших факторів функціонування і розвитку промислових фірм.

Така діяльність постійно рухається у відповідності з вимогами виробництва і реалізації товарів та послуг.

Коли змінюються умови виробничої діяльності, виникає необхідність пристосування системи управління. Всі перераховані ситуації вимагають від фірм адаптації до нових умов.

Варто зазначити, що інформаційна система досить істотно дозволяє скоротити усі витрати діяльності підприємства чи фірми та найбільш максимально оптимізувати управлінський процес. Застосування інформаційних систем у сфері управління підприємствами є важливим для збільшення прибутку.

У кваліфікаційній роботі бакалавра розроблено програмний продукт для автоматизації обліку товарів магазину. Він має також практичне застосування, оскільки ця програма впроваджується на цільовому підприємстві.

Метою роботи є розробка програми автоматизації обліку товарів магазину.

Також об'єктом цього дослідження є інформаційні технології та процеси обробки інформаційних ресурсів.

Предметом дослідження є алгоритми, моделі, структури та методи розробки автоматизованих систем обробки інформаційних ресурсів та створення баз даних, які допомагають в управлінні підприємством чи фірмою. Задачею розробки програмного забезпечення є створення структури конфігурації та її склад для керування нею.

Розділ 1

Характеристика предметної області та постановка задачі

1.1 Аналіз предметної області

Перспективи будь-якого бізнесу нерозривно пов'язані з автоматизацією бізнес-процесів, сьогодні це вже не викликає ніяких сумнівів. Ця необхідність – важлива складова ведення справ і обліку товарів та послуг будь-якої галузі. Довіривши комп'ютерам і технологіям виконання значної частини функцій, забезпечуються точні результати, але і звільняється час і ресурси.

Також зростання конкуренції мотивує роздрібних продавців шукати нові способи підвищення ефективності роботи магазинів і залучення потенційних покупців. Крім того, змінюються формати торгівлі: через ту ж конкуренцію значна частина ринку роздрібною торгівлі належить роздрібним мережам.

Для нормальної роботи будь-якого підприємства автоматизації потрібно підлягати майже усі бізнес-процеси, це дає можливість заощадити багато часу та надати зручності для робітників цього підприємства.

Фахівці радять керівникам підприємств/організацій проаналізувати і зрозуміти, що конкретно їх не влаштовує в поточній роботі. Виходячи з проаналізованих відповідей, приймається рішення про вибір автоматизованих систем.

Таким чином кожне підприємство/організація може вибрати ті бізнес-процеси, які потрібно оптимізувати.

Облік товарів часто вимагає не меншої концентрації уваги і зусиль керівників та власників підприємства, в конкурентних умовах кожна організація повинна ретельно аналізувати і планувати всі свої витрати, налагоджувати ефективну роботу навіть виключно внутрішніх, «тилових» процесів, вибудовувати грамотні відносини з усіма контрагентами.

Знаючи про можливості скорочення внутрішніх витрат, пов'язаних зі складськими операціями, шляхом автоматизації таких процесів, стає можливе резервування товару, відвантаження, сортування, зберігання і перевезення.

Одне з основних переваг автоматизації цього бізнес-процесу - скорочення часу циклу отримання товару на склад і потім – зі складу, відслідковувати товари, сортувати товари; також це відноситься до послуг, які надає підприємство. Шляхом прискорення процесів постановки в чергу, визначення замовлення, відвантаження під нього товару, транспортування, також відслідковування процесу послуг автоматизація може істотно скоротити час від замовлення до доставки. [2]

Переваги автоматизації складу:

- збільшення швидкості роботи складу і оптимізація площі зберігання;
- швидке і оперативне надходження і витрачання товарів;
- контроль над рухом товару і його залишків на складі;
- контроль постачальників;
- швидкий пошук необхідних товарів на складі;
- контроль складських залишків.

Як наслідок скорочення витрат і зростання прибутку вашої компанії. [2]

Висновок

Зараз – саме час проводити автоматизацію, оскільки велика кількість факторів, включаючи вартість автоматизації, робить її набагато доступною, ніж раніше.

Процес аналізу предметної області в розробці інформаційних систем передбачає виділення основних і допоміжних процесів, які покликані забезпечити виробництво продукту або послуги. Але, також виділяються і розглядаються бізнес-процеси, які надають можливість визначитися з бізнес-елементами і структурами даних, які повинні брати участь в обробці даних. Ці процеси заставляють розробляти бази даних, які використовуються для оптимізованого функціонування і розвитку промислових фірм.

1.2 Аналіз інформаційного забезпечення предметної області

1.2.1 Аналіз існуючого програмного забезпечення предметної області

Сучасне програмне забезпечення використовується в багатьох сферах, а вже залежно від потреб користувачів змінюються функції програмного забезпечення для роботи із ними. Але авторська програма зазвичай розробляється тільки для використання на певному(одному) підприємстві, тобто аналоги такої програми знайти важко і майже неможливо. Все таки розглянемо програмні продукти, які будуть подібним до нашої розробки. [3]

«Складський облік товарів» – програма, яка призначена для ведення оперативного складського обліку. Складський облік допомагає забезпечити оптимізацію бізнес-процесів та забезпечує ефективну роботу працівникам. Програма дозволяє по залишках отримувати звіти та відстежувати товари на складі. Інформація про товар зберігається на картках, на яких оснований товарний облік. Екранна форма програми наведена на Рисунок 1.1.

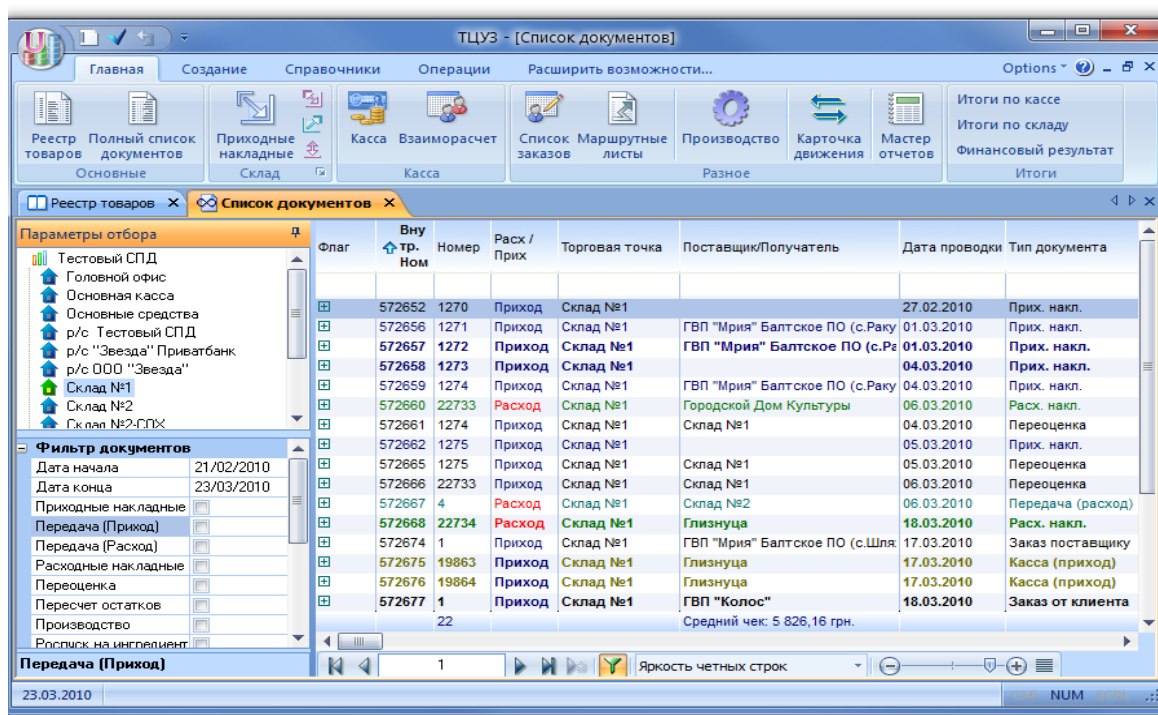


Рисунок 1.1 – Екранна форма програми «Складський облік товарів»

Ця програма надає можливість вести облік товару на складах, також відстежувати рух будь-яких товарів між складами. Також програма дозволяє вести облік розрахунків з постачальниками та покупцями. Також дає можливість друкувати цінників.

Такі звіти в себе включають інформацію про залишки товарів на початок і кінець періоду звітності. Програма «Складський облік товарів» підходить для середніх і малих підприємств оптової торгівлі. [4]

Система jSolutions – дозволяє керувати товарами на складі, їх реалізацією і послуг в різноманітних масштабах, вона підходить міні-складу, а також великим складським центрам. Програма розроблена на базі хмарної платформи jSolutions і має функції, які забезпечують об'єднання в єдину базу даних обліку товарів, в розрізі функцій і характеристик:

- номенклатура;
- облік первинних документів (вхідних/вихідних);
- штрих-коди товарів;
- обліку упакувань, серій, тари;
- облік термінів придатності;
- ведення договорів, в т.ч. з розбивкою на етапи і додаткові положення;
- оформлення вихідних податкових накладних;
- створення пов'язаних ланцюжків документів;
- формування рахунків на оплату з можливістю відправки електронних знижок;
- контроль оплати;
- повернення товарів та їх облік;
- резервування товару;
- формування відомостей товарних запасів;
- відслідковування наявності товарів на складах;
- проведення інвентаризації;
- формування журналу товарних операцій;

– акти комплектації для виробничих завдань.

Екранна форма програми "роздрібна торгівля Система jSolutions" Рисунок 1.2



Рисунок 1.2 – Екранна форма програми "роздрібна торгівля Система jSolutions"

Складський облік товарів в цій системі дає можливість інтеграції з різними системними модулями, також при інтеграції з торгово-касовим обладнанням (ваги, сканери штрих-кодів, POS-термінали, касові апарати, фіскальні реєстратори) можна створити систему, яка автоматизує точки продажів, магазини, супермаркети, торговельні мережі з прогнозованим бюджетом витрат. Поєднуючи деякі бізнес-процеси, в поєднанні з модулем "Бухгалтерія", можна автоматично виводити інформацію в бухгалтерському обліку про надходження, реалізацію та списання товарів зі складу, також розрахунки з постачальниками. [5]

«УкрСклад» – це програмне забезпечення для автоматизації роздрібної та оптової торгівлі. Також програма забезпечить контроль бухгалтерської звітності.

Опис і можливості програми Склад Україна

Виписка і друк таких документів:

- договір;
- прибуткова накладна + (штрих-код цінники);
- рахунок-форма замовлення;
- Рахунок-фактура;
- видаткова накладна + (товарний чек, товарно-транспортна накладна (ТТН), гарантійний талон);
- нова податкова накладна;
- внесення залишків;
- акт списання;
- накладна на переміщення;
- акт інвентаризації;
- акт виконаних робіт;
- поворотна накладна;
- нова коригування податкової;
- прибутковий касовий ордер;
- видатковий касовий ордер;
- платіжне доручення + (заява на переказ готівки, повідомлення);
- вхідне платіжне доручення;
- вікно-касира;
- виробництво;
- доручення;
- універсальний документ;
- виписка документів, ведення архіву документації (реєстру) і висновок їх на друк по декількох фірмах одночасно в одній програмі.

Звітність:

- залишки на складі;
- рух товару по складу;
- рух по клієнтах;

- взаєморозрахунки з контрагентами (постачальники і клієнти);
- статті витрат / приходу;
- виробництво, (вироблено за період, витрачено матеріалів за період, залишки матеріалів у виробництві, готова продукція у виробництві ...);
- прайс-листи, звіт по послугах, цінники (з штрих-кодами), каса, банк і багато інших звітів.

Екранна форма програми «УкрСклад» на Рисунок 1.3

СРД Петренко Ігор Володимирович (Основной счет) - УкрСклад

Документы Архив документов Справочники Отчеты Опции Помощь

Товари список Незарегистрированная версия

Расходная накладная

№ 411 Дата 11.11.2010 Клиент Гимназия №12 ім. М.Кравчука

Основное Общее Дополнительно Серийные номера

Данные о товаре на складе

Основной склад Поиск Фильтр Сортировка (поиск, фильтр) по ...

Группы и подгруппы товаров

Код	Полное название товара	Ед. измерен	К-во	Цена	Валюта гр	Розн. цена	Опт. цена	Дрібний опт	Цена 2
00003	Ялинка біла 150 см	шт	74,00	360,000	грн.	468,000	432,000	450,000	0,000
00004	Ялинка біла 180 см	шт	81,00	400,000	грн.	520,000	480,000	500,000	0,000
00005	Ялинка зелена (стандарт) 150 см	шт	98,00	310,000	грн.	403,000	372,000	387,500	0,000
00006	Ялинка зелена (стандарт) 180 см	шт	90,00	370,000	грн.	481,000	444,000	462,500	0,000
00008	Ялинка срібна (декоративна з прикра	шт	38,00	444,000	грн.	577,200	532,800	555,000	0,000
00009	Ялинка срібна (декоративна з прикра	шт	20,00	514,000	грн.	668,200	616,800	642,500	0,000
00010	Ялинка срібна (декоративна з прикра	шт	20,00	596,000	грн.	774,800	715,200	745,000	0,000
00011	Ялинка кутова 140 см (трапеція)	шт	0,00	194,000	грн.	252,200	232,800	242,500	0,000
00012	Ялинка кутова 180 см (трапеція)	шт	0,00	250,000	грн.	325,000	300,000	312,500	0,000
00013	Ялинка червона (стандарт) 50 см	шт	40,00	80,000	грн.	104,000	96,000	100,000	0,000
00021	Блакитна ялина 1 м	шт	0,00	480,000	грн.	624,000	576,000	600,000	0,000
00022	Блакитна ялина 1,5 м	шт	0,00	620,000	грн.	806,000	744,000	775,000	0,000

Добавить Удалить К-во 1,00 Цена без НДС: 403,000 Цена с НДС: 483,600 Розн. цена Гривня

Код	Полное название товара	К-во	Цена	Цена с НДС	Сумма	Ед. измерения	Дополнительно
1	00016 Подарочковый комплект "Новый Рік"	1,00	625,040	750,050	625,04	шт	
2	0001 Гірлянда хвойна (срібна) 2,5м	5,00	18,200	21,840	91,00	шт	
3	0001 Гірлянда хвойна (зелена, срібні кінчики) 2,5м	3,00	28,600	34,320	85,80	шт	
4	0000 Електрогірлянда "Прозорі кульки" з лампами роз	2,00	143,000	171,600	286,00	шт	
5	0001 Куля срібна	7,00	7,800	9,360	54,60	шт	
6	0001 Куля скляна прозора	12,00	7,020	8,420	84,24	шт	
7	0002 Веріівка "Зірка" різнокольорова	1,00	23,400	28,080	23,40	шт	
8	00005 Ялинка зелена (стандарт) 150 см	1,00	403,000	483,600	403,00	шт	

Всего: Скид./надб. 0,00 Сумма без НДС 1028,04 НДС 205,61 Сумма с НДС 1233,65

Проводить документ Просмотреть Из шаблона Сохранить Отмена Закрыть

Рисунок 1.3 – Екранна форма програми «УкрСклад»

Для програм, які описувалися раніше складемо порівняльну таблицю, де покажемо сильні та слабкі сторони кожного з аналогів.

Таблиця 1.1 – Порівняння аналогів

Характеристики	Система jSolutions	«УкрСклад »	Складський облік
Зручність інтерфейсу	+	+	-
Робота зі штрих кодами	-	+	-
Резервне копіювання	+	+	+
Робота із зовнішніми файлами	+	-	+
Резервне копіювання	+	-	+

Усі попередні програми мають, як переваги так і недоліки, але всі вони надають базові можливості для роботи із базами даних: додавання, редагування та видалення даних, формування звітів, розподілене використання, тощо.

Висновок

В цьому підрозділі кваліфікаційної роботи бакалавра було проаналізовано предметну область. Виявилось, що в сучасній сфері торгівлі існує проблема обліку товарів, що обумовлює значна кількість документації, розподіленість споживачів та інформації.

Було розглянуто сучасні аналоги програмного продукту, а саме: «УкрСклад», Система jSolutions, «Складський облік». Після ознайомлення, було виявлено переваги та недоліки кожного з них.

1.2.2 Аналіз сучасних засобів створення програмного забезпечення

Для того, щоб розпочати розробку програми потрібно визначитись із типом СКБД, що має використовуватись у програмі. Щоб розробити базу даних буде застосовано реляційну модель.

Порівняємо деякі популярні види СКБД.

СКБД – це комплекс мовних та програмних засобів, призначений для створення, ведення та використання БД різними користувачами. Це складна програмна система для накопичення та маніпулювання даними. Кожній прикладній програмі СКБД надає інтерфейс для роботи з базою даних та має засоби безпосереднього доступу до неї. Таким чином, СКБД грає центральну та головну роль в функціонуванні баз даних. [6]

Основні можливості сучасних СКБД:

- підтримка цілісності бази даних;
- контроль за надлишковістю даних;
- спільне використання даних;
- несуперечливість даних;
- незалежність прикладних програм від даних;
- підвищений рівень безпеки, тощо.

Швидкодія сучасних СКБД це їхня основна вимога, також захищеність, розмежування рівнів доступу до даних, механізми резервного копіювання, тощо. [7]

Зараз розроблено велику кількість різних СКБД, для порівняння розглянемо деякі із них: SQLite, PostgreSQL, MySQL, Microsoft Access.

SQLite. Вона без утруднень вбудовується у програмні додатки. Ця система основана на файлах, через це вона надає досить широкий набір для роботи із нею, порівняно із мережевими СКБД.

При роботі із цією СКБД відбувається пряме звернення до файлів, у яких зберігаються дані. Тому SQLite дуже швидка та потужна. [8]

Переваги SQLite:

- підтримка стандартів SQL;
- легкість у розробці та тестуванні;
- файлова структура, вся база даних знаходиться в одному файлі, через це її не важко переносити на різні комп'ютери.

Недоліки SQLite:

- відсутність системи користувачів;

- відсутність можливості збільшення продуктивності.

Microsoft Access є настільною СУБД реляційного типу. Перевагою Access є те, що вона має дуже простий графічний інтерфейс, який дозволяє не тільки створювати власну базу даних, але і розробляти програми, використовуючи вбудовані засоби.

На відміну від інших настільних СУБД, Access зберігає всі дані в одному файлі, хоча і розподіляє їх по різних таблицях, як і належить реляційної СУБД. До цих даних відноситься не тільки інформація в таблицях, а й інші об'єкти бази даних. [9]

Переваги:

1. В першу чергу можна відзначити поширеність, яка обумовлена тим, що Access є продуктом компанії Microsoft, програмне забезпечення та операційні системи якої використовує більшість користувачів персональних комп'ютерів. MS Access повністю сумісний з операційною системою Windows, постійно оновлюється виробником, підтримує безліч мов.

2. MS Access надає велику кількість можливостей за порівняно невелику вартість. Також необхідно відзначити орієнтованість на користувача з різною фаховою підготовкою, що виражається в наявності великої кількості допоміжних засобів (Майстрів, як уже зазначалося), розвинену систему довідки і зрозумілий інтерфейс. Ці кошти полегшують проектування, створення БД і вибірку даних з неї.

3. MS Access надає в розпорядження непрограмуєчому користувачеві різноманітні діалогові засоби, які дозволяють йому створювати додатки не вдаючись до розробки запитів на мові SQL або до програмування макросів або модулів на мові VBA. Access володіє широкими можливостями по імпорту / експорту даних в різні формати, від таблиць Excel і текстових файлів, до практично будь-який серверної СУБД через механізм ODBC.

4. Ще одна важлива перевага MS Access полягає в розвинених вбудованих засобах розробки додатків. Більшість додатків, які розповсюджуються серед користувачів, містить той чи інший обсяг коду VBA (Visual Basic for Applications). Оскільки VBA є єдиним засобом для виконання

багатьох стандартних задач в Access (робота зі змінними, побудова команд SQL під час роботи програми, обробка помилок, використання Windows API ит. Д.), Для створення більш-менш складних додатків необхідно його знання і знання об'єктної моделі MS Access.

Недоліки:

1. Система управління базами даних MS Access відноситься до файл-серверних СУБД. Це означає, що обробка даних здійснюється безпосередньо на комп'ютері клієнта, користувача. Серверна частина тільки витягує необхідну інформацію і на вимогу передає її. Це призводить до того, що по локальній мережі передаються великі потоки даних. При великій кількості користувачів, які ведуть роботу з базою даних, може виникнути перевантаження мережі.

2. Класична файл-серверна архітектура може спричинити за собою низьку продуктивність при наявності великої кількості користувачів. Це питання частково вирішене в версії MS Access 2010.

3. Відсутність надійних засобів захисту інформації. По суті, тут використовується тільки пароль на базі даних і пароль користувача. Досвідчений фахівець може зняти такий захист без особливих зусиль.

4. Розрахований на багато користувачів режим можливий тільки в мережі з файловим сервером або в тимчасовій мережі.

5. Для розробки нестандартного інтерфейсу потрібно буде залучати програмістів. Також редактор SQL-запитів в даному випадку організований досить примітивно, тому використовувати його не дуже зручно.

6. Багато користувачів до недоліків СУБД MS Access відносять те, що вона не безкоштовна.

7. Ще одним мінусом є те, що Access заточений спеціально під операційні системи сімейства Windows. Щоб спробувати використовувати дані, створені в цій СУБД, в інших операційних системах, доведеться повозитися. Набагато простіше буде перенести інформацію в іншу СУБД.

PostgreSQL. Це одна з найбільш професійна із популярних СКБД. Вона вільно розповсюджується та максимально відповідає стандартам SQL. Від інших СКБД вона відрізняється підтримкою об'єктно-орієнтованого та

реляційного підходів до баз даних. Завдяки потужним технологіям PostgreSQL дуже продуктивна. Її можна доповнювати власними процедурами, щоб полегшити використання повторюваних операцій . [10]

Переваги PostgreSQL:

- відповідність стандартам та відкритий вихідний код;
- велика кількість розробників;
- можливість розширення функціоналу;
- об'єктна орієнтованість;
- кроссплатформенність.

Недоліки PostgreSQL:

- недостатня швидкодія;
- низька популярність;
- складність у вивченні.

MySQL. Найрозповсюдженіша повноцінна серверна СКБД. MySQL дуже функціональна, вільно розповсюджувана СКБД, яка успішно працює як із окремими додатками, так і з сайтами та web-додатками. Вона легка у вивченні, оскільки у неї зручна документація та їй присвячена велика кількість джерел, які стосуються СКБД та баз даних в цілому.

Завдяки популярності MySQL існує велика кількість різноманітних плагінів та розширень, які полегшують роботу із системою.

Незважаючи на те, що у ній не реалізовано весь функціонал SQL, MySQL пропонує доволі велику кількість інструментів для розробки додатків. Оскільки це серверна СКБД, то, на відміну від SQLite, доступ до даних відбувається через служби MySQL. [11]

Переваги MySQL:

- простота в роботі;
- багатий функціонал;
- масштабованість;
- висока продуктивність.

Недоліки MySQL:

- обмеження функціональності, необхідні для окремих задач;
- проблеми з надійністю;
- порівняно невисокі темпи розробки додатків.

Перелічивши ці особливості, для реалізації в авторській програмі буде використано СКБД PostgreSQL., що відповідає вимогам для нашої програми та в найбільшій мірі дозволить реалізувати необхідний функціонал.

1.3 Постановка задачі

Метою кваліфікаційної роботи бакалавра є програмного забезпечення, що виконує такі функції:

Довідники:

- облік користувачів;
- облік інтернет користувачів;
- облік співробітників;
- облік банків;
- облік контрагентів;
- довідники УКТ ЗЕД і одиниць виміру;
- довідник товарів.

Первинні документи:

- облік прихідних накладних;
- формування рахунків на оплату;
- формування видаткових накладних.

Сервіси:

- обрахунки поточних цін;
- обрахунки поточних залишків;
- імпортування всеукраїнських каталогів УКТ ЗЕД;
- імпортування всеукраїнських каталогів одиниць виміру;
- розсилка повідомлень на email користувачів;
- обрахунок поточних цін;

- формування обігової відомості по товарах;
- формування відомості реалізації товарів;
- формування відомості надходження товарів;
- вигразка даних у файлі .XML;
- вигразка даних у файлі .xlsx .

Звіти:

- відомість надходження;
- відомість реалізації;
- обігова відомість по контрагентам.

Для успішної розробки додатку необхідно використати наступні компоненти: Apache HttpComponent ,Apache Tomcat , Open HTML to PDF , Vaadin , YARG ...

Розділ 2

Проектування інформаційної системи

2.1 Аналіз та автоматизація обробки інформаційних потоків

Формування первинних документів і на їх основі повинен бути здійснений облік товарів, цін та залишків їх у магазині.

Бізнес-процеси підприємства, що підлягають автоматизації:

- 1) Формування накладних на надходження товарів.
- 2) Формування накладних на реалізацію товарів.

Бізнес-процес "Формування накладних на надходження товарів"

Даний бізнес-процес формує накладну про товар, який надійшов:

- постановка на облік кількості товару та автоматичний розрахунок роздрібною ціни згідно вказаної націнки.

Описаний вище бізнес-процес зображений нижче Рисунок 2.1

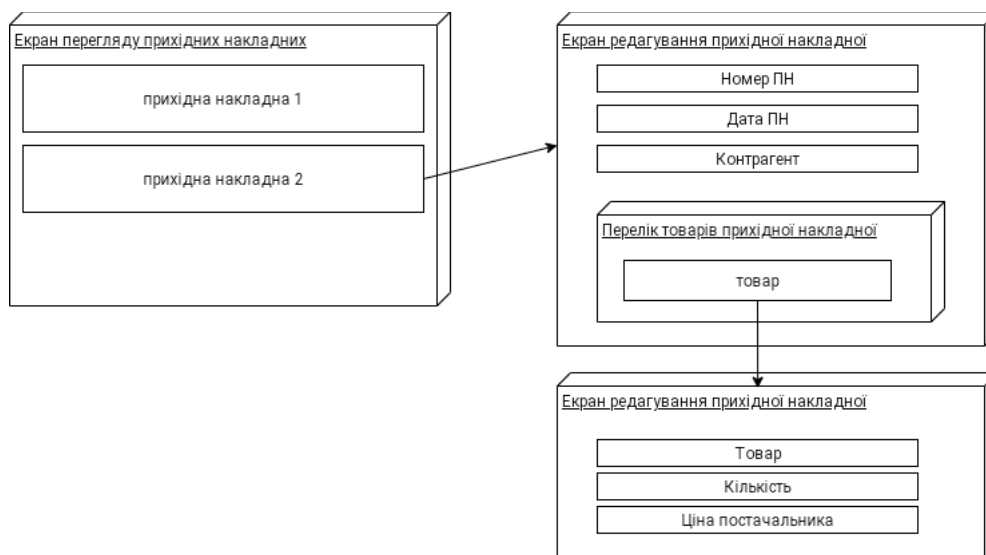


Рисунок 2.1 – Стадія бізнес-процесу "Формування накладних на надходження товарів"

Бізнес-процес "Формування накладних на реалізацію товарів"

Даний бізнес-процес формує накладну про товар, який був реалізований:

– списання (реалізація) кількості товару по роздрібних цінах контрагентам.

Описаний вище бізнес-процес зображений нижче Рисунок 2.2

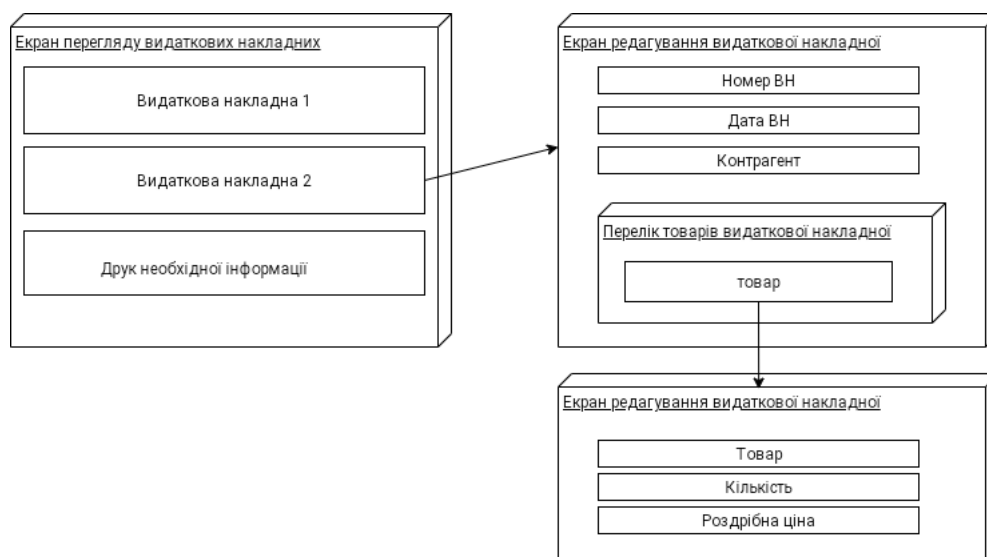


Рисунок 2.2 – Стадія бізнес-процесу "Формування накладних на реалізацію товарів"

2.2 Розробка структури інформаційної системи

У розроблюваній базі даних можна виділити такий набір основних сутностей: Unit(Одиниця виміру), Uktzed(Код зовнішньоекономічної діяльності), Customer (Контрагент), Employee(Співробітник), Firm(Наша фірма), Invoice(Рахунок), ProductGroup(Група товарів), Product(Товар), PurchaseInvoice(Прихідна накладна), WayInvoice(Видаткова накладна), User(Користувач), WebUser(Веб користувач), а також співставлені сутності NamedEntity, Requisite, Purchase, PurchaseDetail, Sale, SaleDetail.

Розглянемо, яким чином взаємопов'язані між собою ці сутності(Рисунок 2.3 – 2.9):

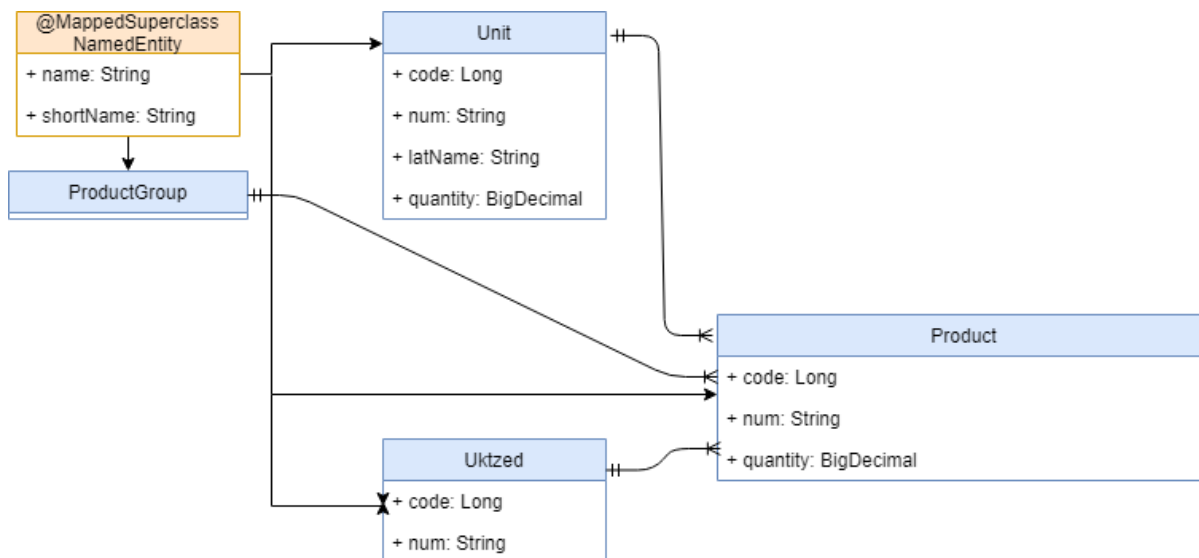


Рисунок 2.3 – Взаємодія сутностей

До Рисунок 2.3 складена таблиця 2.1

Таблиця 2.1 – Взаємодія сутностей

Група товарів	асоціація	Товар
Група товарів 1		Товар 1
Група товарів 2		Товар 2
		Товар 3
		Товар 4
		Товар 5
	Тип зв'язку 1:1	

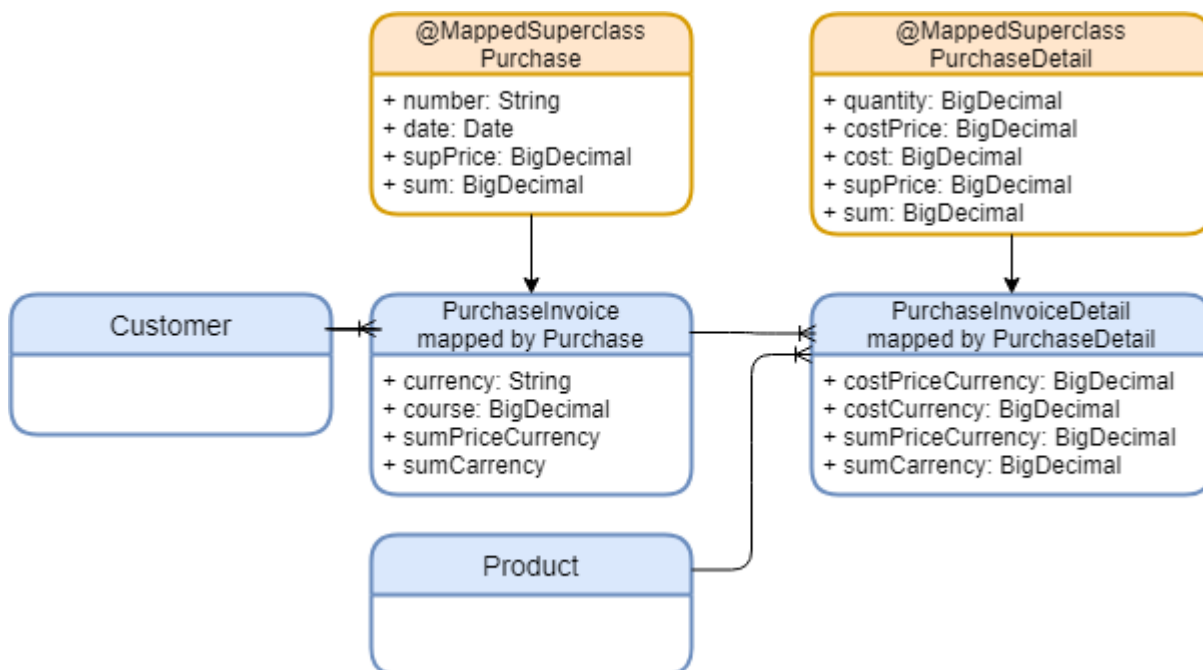


Рисунок 2.4 – Взаємодія сутностей

До Рисунок 2.4 складена таблиця 2.2

Таблиця 2.2 – Взаємодія сутностей

Контрагент	Прихідна накладна	Вихідна накладна	Товар
K1	ПН1	ВН1	T1
K2	ПН2	ВН2	T2
K3	ПН3	ВН3	T3
KN	ПНН	ВНН	TN

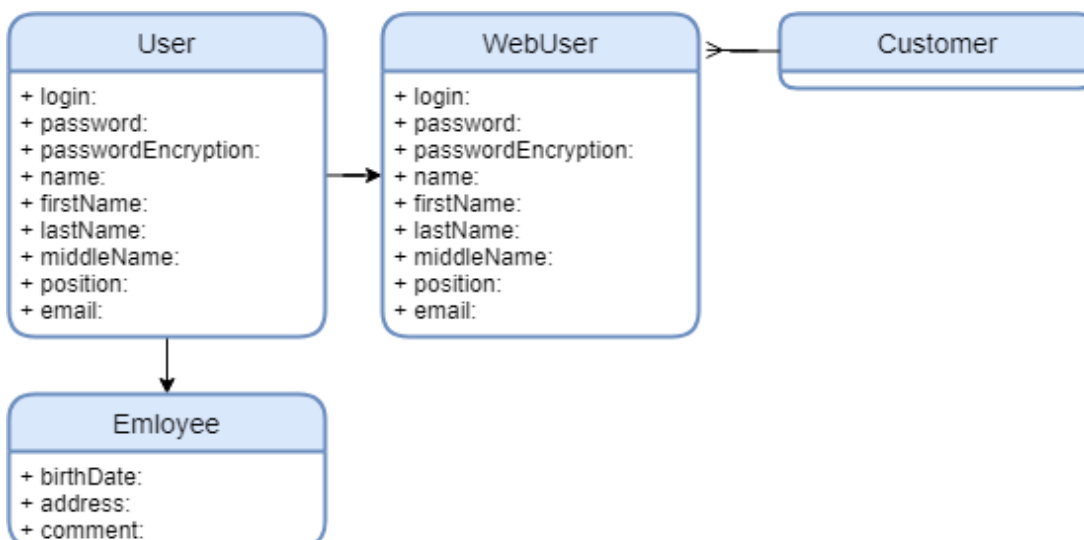


Рисунок 2.5 – Взаємодія сутностей

До Рисунок 2.5 складена таблиця 2.3

Таблиця 2.3 – Взаємодія сутностей

Користувач	WebКористувач	Контрагент
Кр1	Вк1	Кн1
Кр2	Вк2	Кн2
Кр3	ВкN	Кн3
Кр4	Працівник	Кн4
Кр5	П1	Кн5
Кр6	П2	Кн6
КрN	ПN	КнV

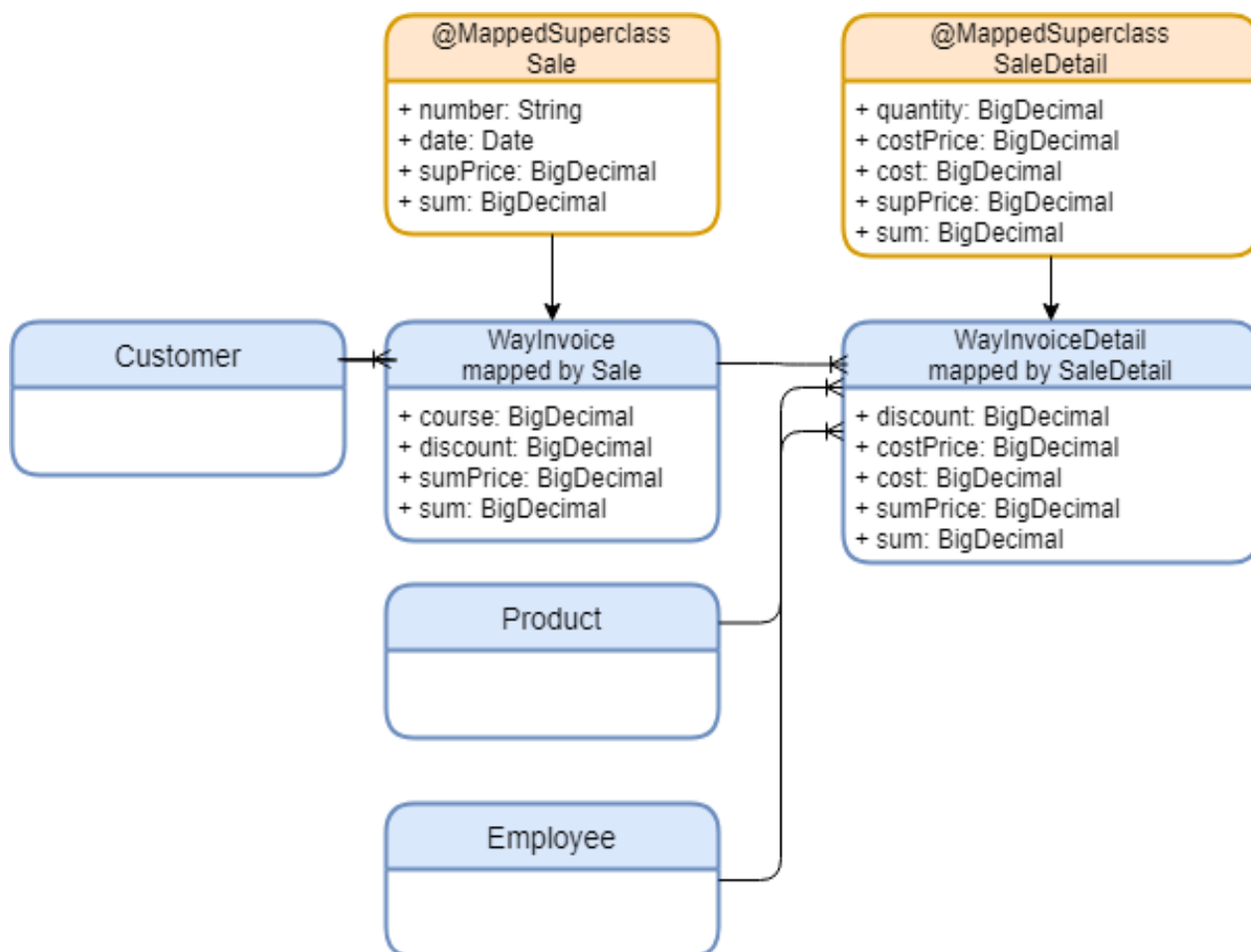


Рисунок 2.6 – Взаємодія сутностей

До Рисунок 2.6 складена таблиця 2.4

Таблиця 2.4 – Взаємодія сутностей

Контрагент	Прихідна накладна	Видаткова накладна	Товар
K1	ПН1	ВН1	Т1
K2	ПН2	ВН2	Т2
K3	ПН3	ВН3	Т3
K4	ПН4	ВН4	Працівник
K5	ПН5	ВН5	П1
K6	ПН6	ВН6	П2
KN	ПНН	ВНН	ПН

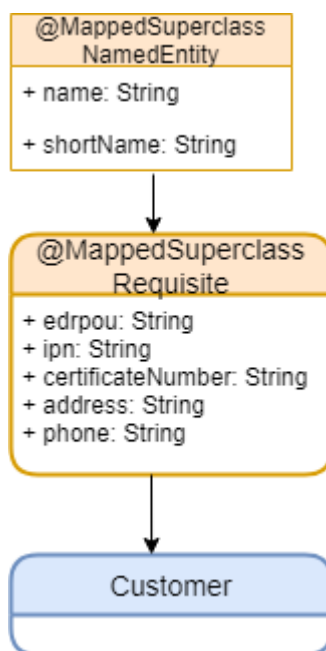


Рисунок 2.7 – Взаємодія сутностей

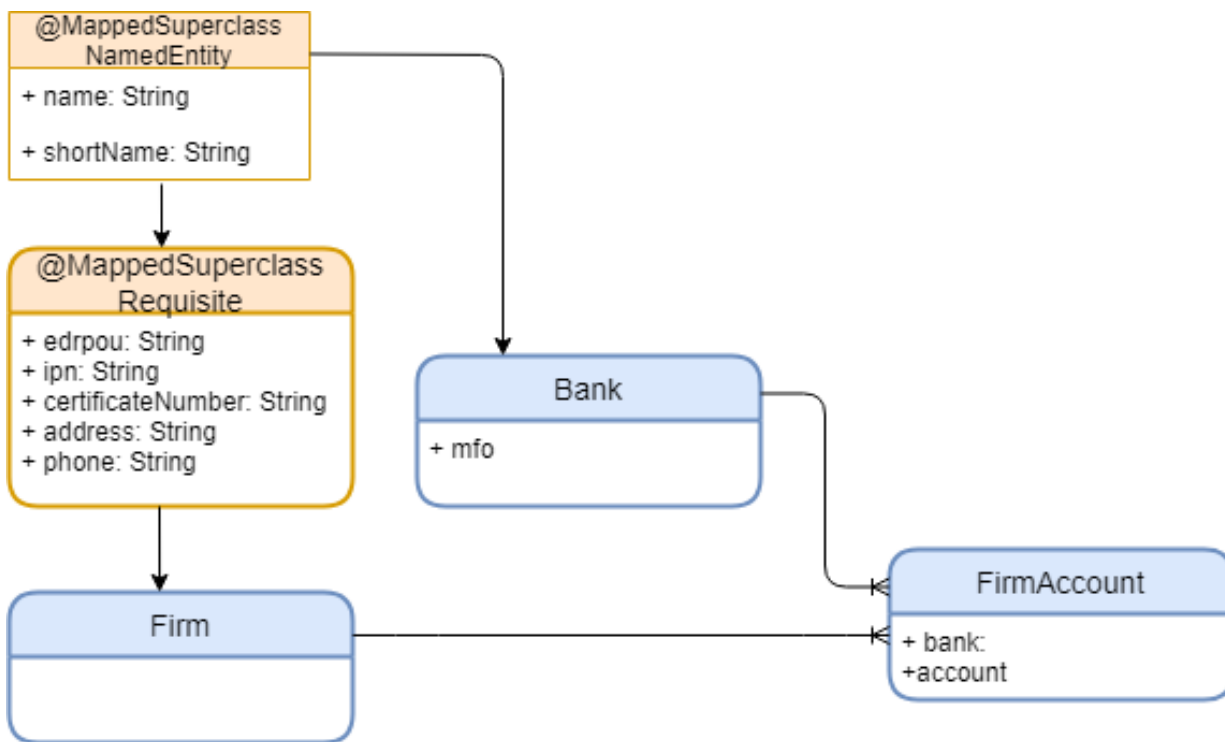


Рисунок 2.8 – Взаємодія сутностей

До Рисунок 2.8 складена таблиця 2.5

Таблиця 2.5 – Взаємодія сутностей

Банк	Акаунт фірми	Фірма
Б1	АФ1	Ф1
БN	АФN	ФN

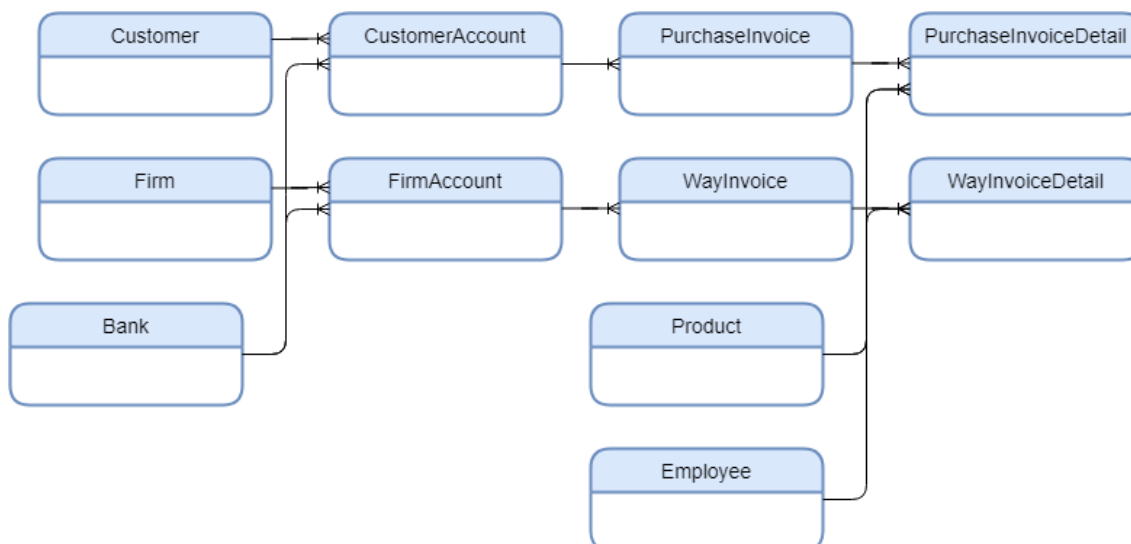


Рисунок 2.9 – Взаємодія сутностей

Співставлені сутності використовуються тільки на етапі формування структури БД для розширення переліку атрибутів основних сутностей. Цей механізм платформи дозволяє уникнути повторного опису однотипних атрибутів у схожих за своїм набором атрибутів основних сутностей. З цією ж ціллю застосоване наслідування сутностей WebUser та Employee від сутності User, що стандартно згенерована платформою.

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

Java – це мова програмування загального призначення, який слід парадигмі об'єктно-орієнтованого програмування і підходу «Написати один раз і використовувати скрізь». Java використовується для настільних, мережевих, мобільних і корпоративних додатків

Плюси програмування на Java в нашому випадку:

Java – вже не єдиний офіційно підтримуваний мову для розробки на різних платформах. Java далеко не єдиний вибір в веб-програмуванні. Проте, Java йде в ногу з часом. Вона досить лаконічно підлаштовується до потреб розробників, є досить безпечною і також може використовуватися на різних платформах, та багато інших плюсів. [12]

2.3.2 Вибір фреймворка

Cuba-platform – високорівнева Java-платформа з відкритим кодом для створення корпоративних інформаційних систем. [13]

Cuba-platform – це:

1) ПЕРЕВІРЕНА АРХІТЕКТУРА

CUBA пропонує модульну і масштабовану архітектуру на основі популярних фреймворків, створену для роботи в будь-якому оточенні. Модуль Generic UI в разі прискорює розробку інтерфейсів для внутрішніх користувачів.

2) Розвинута ЕКОСИСТЕМА

Нефункціональні вимоги - підводна частина айсберга корпоративних додатків, яка здатна перетворити автоматизацію навіть найпростішого бізнес-процесу в головний біль. Доповнення CUBA дозволяють практично в один клік реалізувати типові вимоги, такі як управління користувачами і доступом до даних, інструменти адміністрування, звітність і підтримка BPM.

3) ЕФЕКТИВНІ ЗАСОБИ РОЗРОБКИ

Інструменти забезпечують по-справжньому комфортний і продуктивний процес розробки завдяки генерації коду, візуальним редакторам, контекстним підказкам і інтуїтивної навігації.

4) МОДУЛЬНІСТЬ

Архітектура платформи дозволяє включити будь-СUBA-додаток до складу іншого. Це дає можливість легко реалізувати модульність, розробляючи окремо компоненти програми і об'єднуючи їх в подальшому в єдину систему.

5) МАСШТАБОВАНІСТЬ

Платформа дозволяє створювати горизонтально і вертикально масштабовані рішення. Виходячи з планованого навантаження додатки і допустимого часу простою, підтримуються різні варіанти розгортання.

6) СУМІСНІСТЬ

CUBA-додатки сумісні з популярними реляційними СКБД і працюють в будь-якому Java контейнері сервлетів. Додатки можна поширювати у вигляді WAR, Docker image, UberJar або розгортати в хмарі.

Крім того Платформа надає можливість взаємодії з цією СКБД за допомогою механізму ORM(Object-Relational Mapping):

Object-Relational Mapping – об'єктно-реляційне відображення – технологія зв'язування таблиць реляційної бази даних з об'єктами мови програмування. У платформі CUBA використовується реалізація ORM на основі фреймворку EclipseLink. [14]

Використання ORM дає ряд очевидних переваг:

- дозволяє працювати з даними реляційної СКБД, маніпулюючи об'єктами Java;
- спрощує програмування, позбавляючи від рутини написання тривіальних SQL-запитів;
- спрощує програмування, дозволяючи отримувати і зберігати цілі графи об'єктів однією командою;
- забезпечує легке портування додатків на різні СКБД;
- дозволяє використовувати лаконічний мову запитів JPQL.

Аналізуючи вище перераховані пункти, можна сказати, що ця платформа підходить для виконання даного завдання.

2.3.3 Вибір системи керування базами даних

PostgreSQL є найбільш професійною, як зазначено вище, вона вільно розповсюджується та максимально відповідає стандартам SQL, також має відкритий вихідний код, через це має можливість розширення функціоналу. Являється дуже потужною та продуктивно, тому вона в найбільш повній мірі задовольняє наші потреби для виконання поставленої задачі.

2.3.4 Вибір редактору програмного коду

Глибокий аналіз коду:

IntelliJ IDEA шукає зв'язки між символами у всіх файлах, аналізуючи код, та на всіх мовах, що були використані в проєкті. При цьому аналізі IntelliJ IDEA забезпечує допомогу при написанні коду, зручність при навігації, перевірка помилок в коді.

Все що потрібно прямо в редакторі:

У IntelliJ IDEA можна здійснювати велику частину дій, не пов'язаних напряму з написанням коду, прямо з редактора.

Можливо бачити додаткову інформацію у спливаючих вікнах, не відволікаючись від контексту. Наприклад, Ctrl + Shift + I покаже визначення елемента, на якому стоїть курсор.

Для того, щоб згенерувати код не потрібно проходити через діалоги або заповнювати величезні форми.

Ергономічний інтерфейс:

У IntelliJ IDEA є швидкий пошук по всіх списках, і спливаючих вікнах. Можна миттєво перейти до списку об'єктів, що містять текст, який вибирається.

Аналіз потоків даних

Для автодоповнення, IntelliJ IDEA аналізує весь потік даних, щоб передбачити можливий тип символу під час виконання, і на основі цієї інформації уточнює варіанти, автоматично додаючи приведення класів.

Мовні вставки

Всі розумні функції допомоги при написанні коду працюють не тільки для вибраної мови, а й для виразів і строкових літералів іншою мовою. Наприклад, можна вставляти в рядкові літерали Java фрагменти коду на SQL, XPath, HTML, CSS або JavaScript.

Розділ 3

Програмна реалізація інформаційної системи

3.1 Структура і функціональне призначення модулів системи

Підсистема безпеки Рисунок 3.1 – Підсистема безпеки:

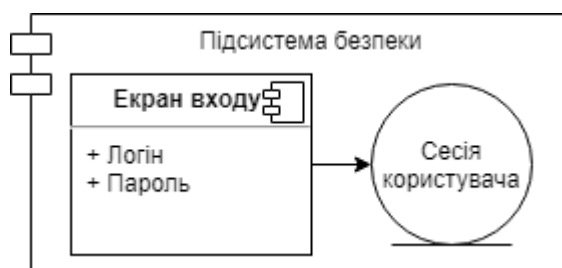


Рисунок 3.1 – Підсистема безпеки

Екран входу – вікно входу в систему. У цьому вікні проводиться аутентифікація користувача по імені облікового запису та паролю. У базі даних замість пароля, в цілях його безпеки, зберігається хеш.

Після входу користувача в систему створюється об'єкт Сесія користувача – призначена для користувача сесія. Це центральний елемент забезпечення безпеки, об'єкт, асоційований з аутентифікованими в даний момент в системі користувачем і містить інформацію про права доступу користувача до даних.

Підсистема довідників Рисунок 3.2 – Підсистема довідників:



Рисунок 3.2 – Підсистема довідників

У формах даної підсистеми є можливість створення записів з детальним описом параметрів об'єктів довідників.

Підсистема первинних документів Рисунок 3.3 – Підсистема первинних документів:

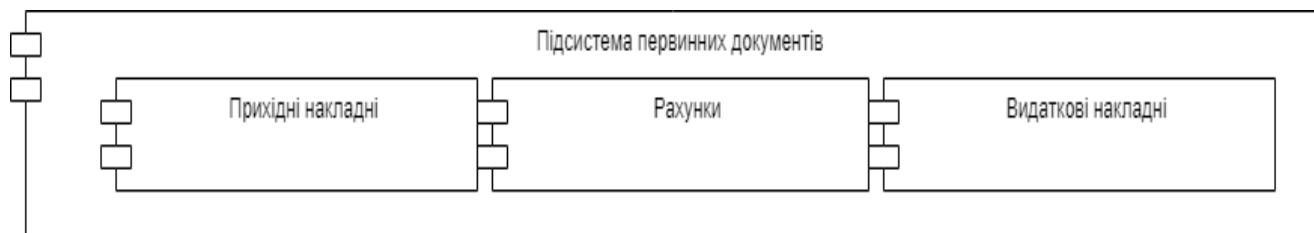


Рисунок 3.3 – Підсистема первинних документів

У формах даної підсистеми здійснюється облік руху товарів у кількісному та грошовому виразі.

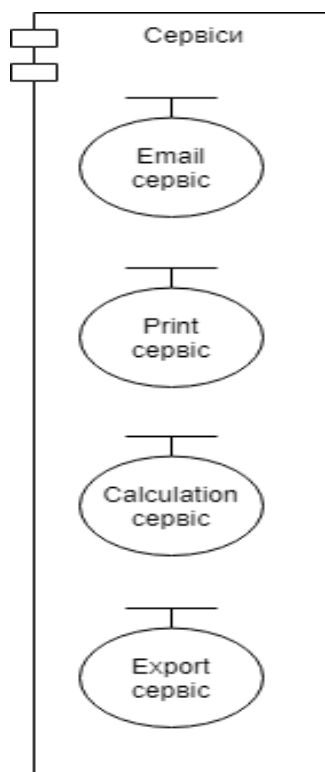


Рисунок 3.4 – Сервіси

Сервіси

Email сервіс надає засоби відправки повідомлень електронної пошти з наступними можливостями:

1. Синхронна або асинхронна відправка. У разі синхронної відправки викликаючий код очікує, поки повідомлення не буде віддане на SMTP сервер. При асинхронній відправці повідомлення зберігається в базі даних, і управління негайно повертається викликаючому коду. Відправлення здійснюється пізніше шляхом виклику з призначеного завдання.

2. Надійна фіксація факту відправки і помилок в базі даних, як для синхронної, так і для асинхронної відправки.

3. Інтерфейс для пошуку і перегляду інформації про повідомлення, включаючи всі атрибути і вміст повідомлень, а також статус відправки і кількість зроблених спроб.

Print servic надає можливість:

1. Створювати шаблони в звичних і доступних всім середовищах (Microsoft Office і LibreOffice / OpenOffice) і описувати джерела даних в інтерфейсі додатка, використовуючи модель даних програми, SQL, JPQL або виконувати скрипти.

2. Print servic надає можливість.

3. Швидко створювати шаблони звітів безпосередньо з програми за допомогою покрокового помічника.

4. Формувати звіти в DOC, DOCX, ODT, XLS, XLSX, HTML і довільних текстових форматах.

5. Створювати XLS (X) звіти складної структури - багаторівневі, з агрегацією даних або перехресними таблицями.

6. Використовувати графіки і формули в XLS (X) звітах.

7. Конвертувати звіти з офісних форматів або HTML в PDF.

Calculation servic надає засоби для різноманітних маніпуляцій та обчислень даних:

1. Форматування дат у символічні строки у різних форматах.

2. Переведення числа у форму прописом.

3. Обрахування підсумкових сум із застосування різноманітних фільтрів, накладених на первинні документи.

4. Формування штрих-кодів, використовуючи як первинні так і обчислені дані.

5. Створення коректної нумерації первинних документів, враховуючи їх приналежність до батьківських сутностей.

6. Різноманітні маніпуляції з датами, як то: обчислення стандартних періодів, дат у минулому та майбутньому.

Export servic забезпечує ефективну обробку готових об'єктів що знаходяться у БД або сформовані у вигляді неперсистентних сутностей, а саме:

1. Копія по зразку первинних документів.
2. Створення різних типів документів з одного вихідного, наприклад створення видаткової накладної, використовуючи рахунок як вихідний документ.
3. Експорт сформованих зведених документів у файли різних форматів (xlsx, xml, csv).

Підсистема зведених документів Рисунок 3.5 – Підсистема зведених документів:

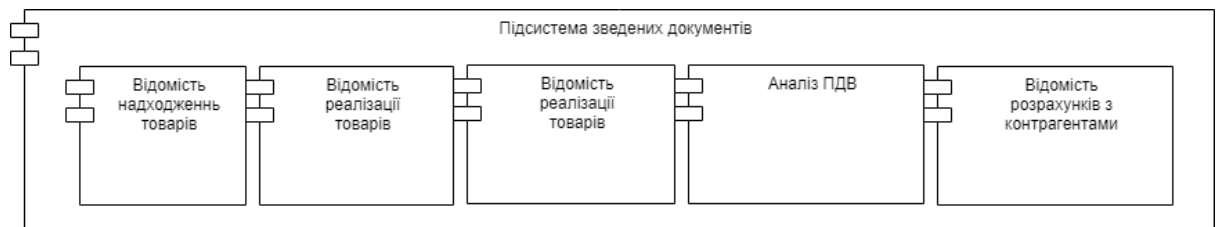


Рисунок 3.5 – Підсистема зведених документів

У формах даної підсистеми є можливість доступу до різних зведених відомостей з різноманітними фільтрами вибору діапазонів та ознак об'єктів БД.
Рисунок.3.5 – Зведена підсистема

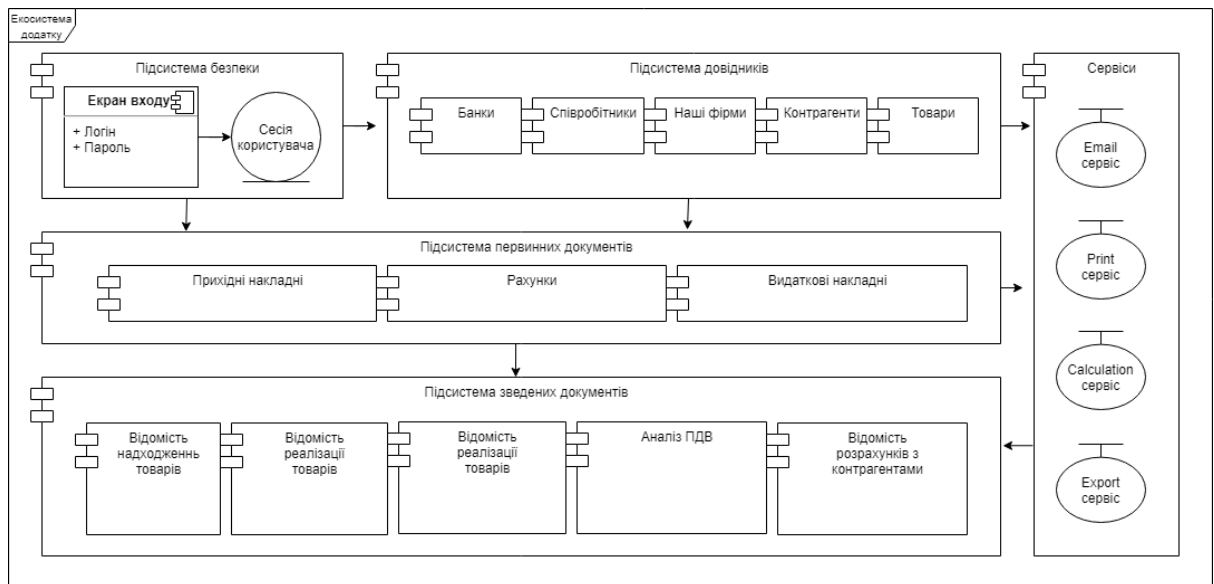


Рисунок 3.5 – Зведена підсистема

3.2 Розробка програмних модулів , інтерфейсу

При розробці модулів були задіяні можливості платформи по генерації програмного коду. Найбільш ефективно вони застосовуються при створенні модулів первинних документів, де присутній великий об'єм стандартного коду по обробці даних.

3.3 Інструкція користувача

1. Для того, щоб користуватися програмою потрібно:

1.1 Зайти в будь-який браузер і в місце пошуку ввести url-адресу <http://192.168.1.31:8080/csk/#main> та натиснути пошук. Рисунок 3.6 – url-адресу порталу

http://192.168.1.31:8080/csk/#main

Рисунок 3.6 – url-адресу порталу

1.2 Відкривається головна сторінка порталу, потрібно вибрати значок, який означає «Вхід» Рисунок 3.7 – Головна сторінка Порталу ЦСК

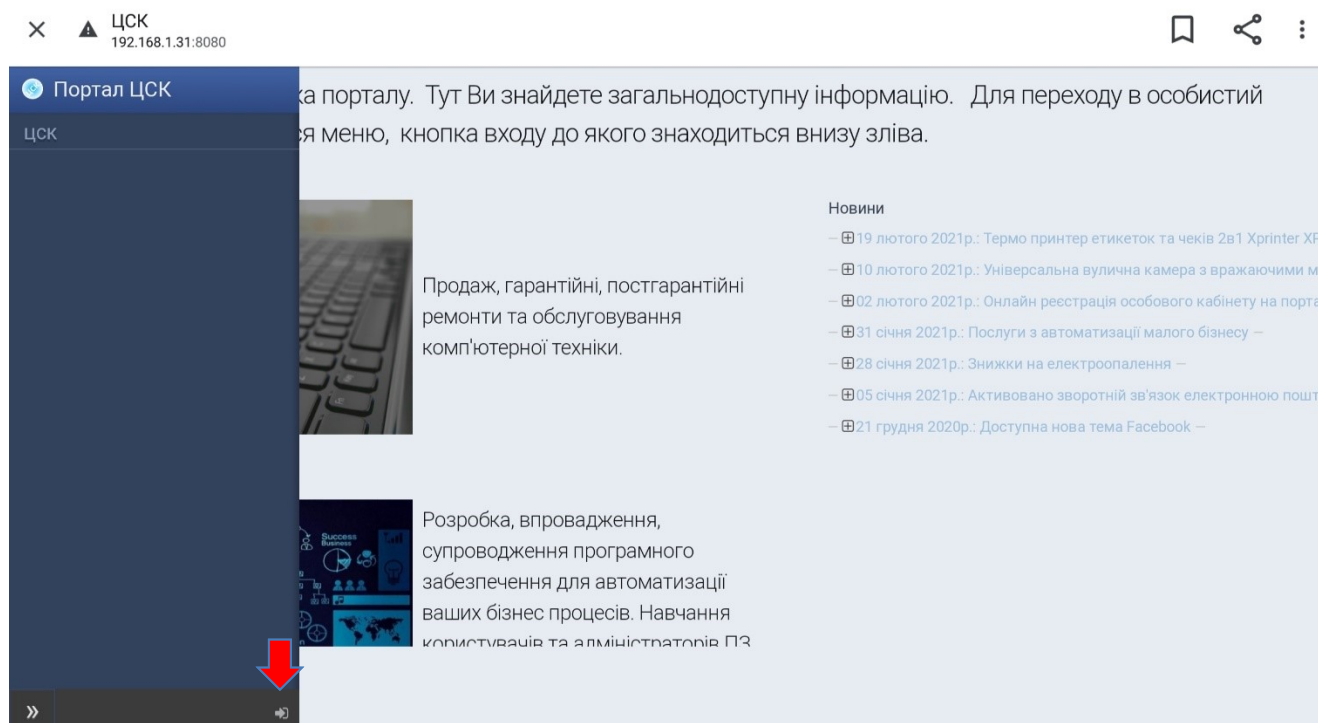


Рисунок 3.7 – Головна сторінка Порталу ЦСК

1.3 Відкривається вікно для входу. Потрібно ввести логін і пароль, вибрати «Запам'ятати мене», вибрати мову та натиснути «Надіслати». Рисунок 3.8 – Вхід у систему ЦСК

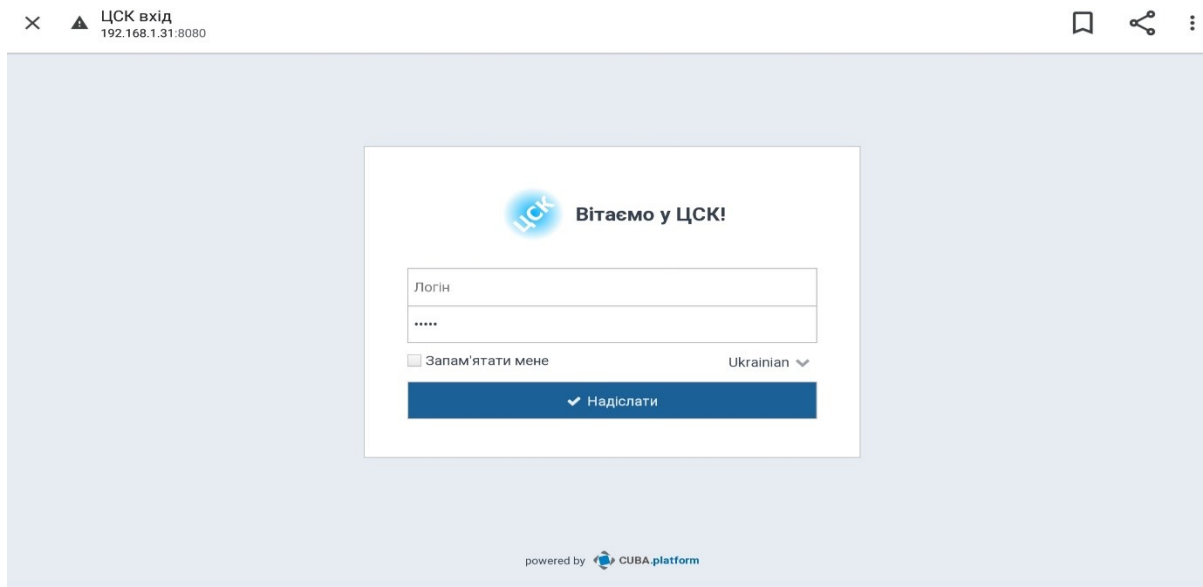


Рисунок 3.8 – Вхід у систему ЦСК

1.4 Після входу відкривається вікно, де видно останні оновлення. Зліва потрібно відкрити панель управління «Портал ЦСК». Рисунок 3.9 – Головна сторінка для користувачів Порталу ЦСК

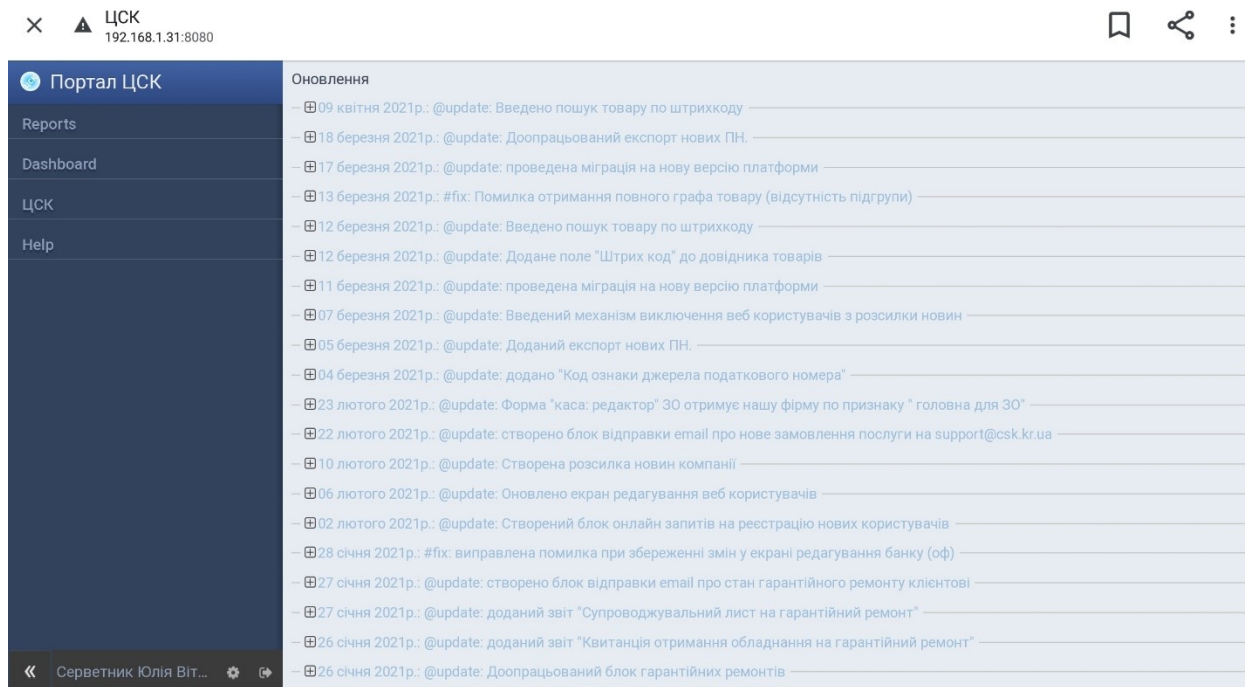


Рисунок 3.9 – Головна сторінка для користувачів Порталу ЦСК

1.5 Щоб додати новий товар на склад потрібно на панелі вибрати «ЦСК»; «Первинні документи»; «Офіційні»; «Прихідні накладні(оф)».

Рисунок 3.10 – Шлях до вікна «Прихідні накладні(оф)»

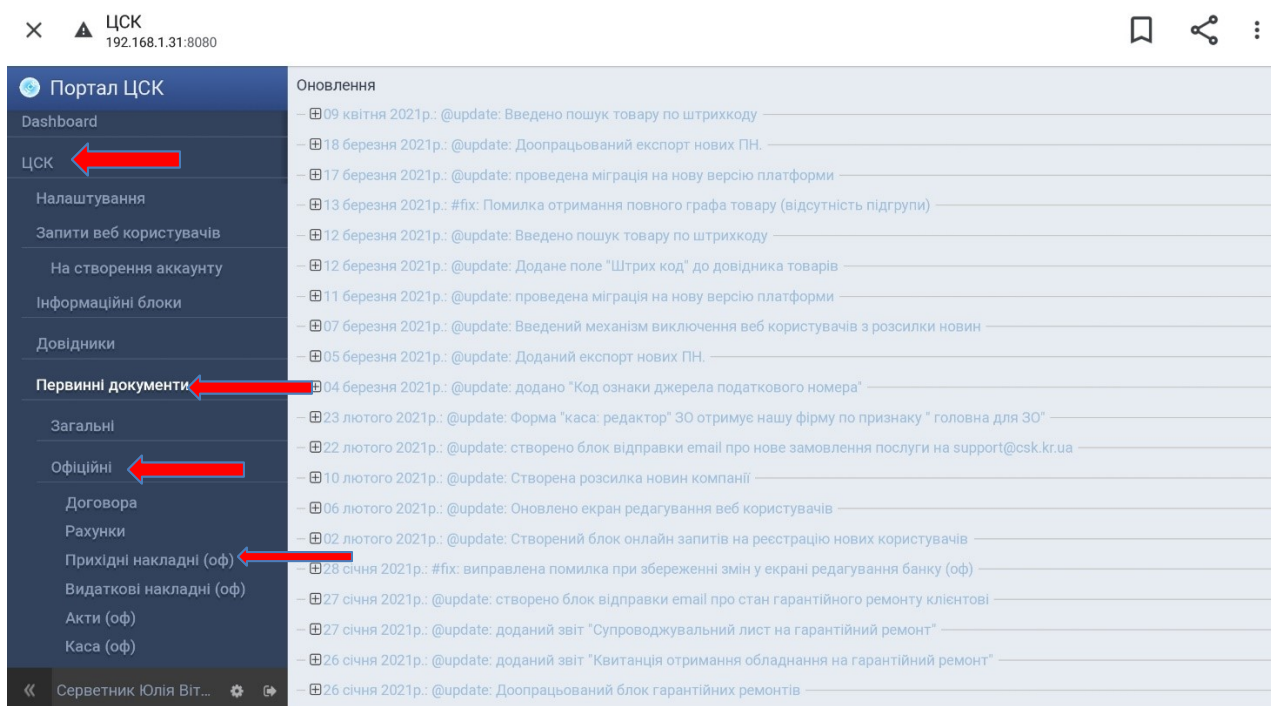


Рисунок 3.10 – Шлях до вікна «Прихідні накладні(оф)»

1.6 Відкривається вікно з переліком прихідних накладних. Щоб створити нову прихідну накладну вибрати «Створити» в верхньому лівому куті.

Рисунок. 3.11 – Вікно прихідних накладних

Прихідні накладні (оф) ×

Filter

Оновити Додати умову пошуку Show rows 50

Створити Редагувати Видалити Excel 1-50 rows of [?] > <

Наш рахунок	Контрагент	Номер	Дата	Номер ПН	Дата ПН	Місяць року	Сума
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ТОВ "СИСТЕМИ ДЛЯ БІЗНЕСУ 2018"	1903	18/03/2021 14:02				0,000000
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ТОВ "СИСТЕМИ ДЛЯ БІЗНЕСУ 2018"	123	04/03/2021 11:20				0,000000
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ТОВ "СИСТЕМИ ДЛЯ БІЗНЕСУ 2018"	13	16/11/2020 09:10				0,000000
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ТОВ "СИСТЕМИ ДЛЯ БІЗНЕСУ 2018"	123	10/11/2020 15:38			01/11/2020	0,000000
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ	ТОВ "Центр сертифікації ключів "Україна"	ХМС14147184	05/08/2020 09:13	10142	05/08/2020	01/08/2020	0,000000
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ТОВ "ТК"	15/01/2021					

Рисунок 3.11 – Вікно прихідних накладних

1.7 Для створення прихідної накладної потрібно ввести обов'язкові поля(виділені червоним), потім перейти до «Товари прихідної накладної» та натиснути «Створити». Рисунок 3.12 – Заповнення прихідної накладної

Прихідна накладна (оф) редактор

Прихідні накладні (оф) > Прихідна накладна (оф) редактор

Наш рахунок: 7840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ

Контрагент

Номер

Дата: 12/04/2021 13:04

Номер ПН

Дата ПН

Місяць року

Коментар

Товари прихідної накладної (оф)

Створити Редагувати Видалити

Товар	Кількість	Собівартість без ПДВ	ПДВ	Собівартість	Ціна	Сума собівартості без ПДВ	ПДВ по сумі собівартості	Собівартість
						0	0	0

OK Відмінити

Рисунок 3.12 – Заповнення прихідної накладної

1.8 Обов'язкове введення/вибирання товару, опис. Можлива функція «По останньому надходженню», щоб не вводити однакові данні щоразу. Щоб зберегти вибрати «ОК». Рисунок 3.13 – Заповнення прихідної накладної про товар

ЦСК
192.168.1.31:8080

Портал ЦСК

Запити веб користувачів

На створення аккаунту

Інформаційні блоки

Довідники

Первинні документи

Загальні

Офіційні

Договора

Рахунки

Прихідні накладні (оф)

Видаткові накладні (оф)

Акти (оф)

Каса (оф)

Кошти в банку

РРО

ПРРО

Серветник Юлія Віт...

Товар прихідної накладної... x

Прихідні накладні (оф) > Прихідна накладна (оф) редактор > Товар прихідної накладної (оф)

Товар

Кількість 0,000

Собівартість без ПДВ 0,00

ПДВ 0,000000

Собівартість 0,00

Ціна 0,00

Відмінити

По останньому надходженню

OK

Сума собівартості

ПДВ по сумі собівартості

Собівартість (сума)

Сума

Рисунок 3.13 – Заповнення прихідної накладної про товар

2 Створення видаткової накладної:

2.1 Для створення видаткової накладної потрібно в панелі управління вибрати «Видаткові накладні (оф)», та аналогічним шляхом вибрати «Створити». Рисунок 3.14 – Вікно видаткових накладних

ЦСК 192.168.1.31:8080

Портал ЦСК

Запити веб користувачів
На створення аккаунту
Інформаційні блоки
Довідники
Первинні документи
Загальні
Офіційні
Договора
Рахунки
Прихідні накладні (оф)
Видаткові накладні (оф)
Акти (оф)
Каса (оф)
Кошти в банку
РРО
ПРРО

Прихідні накладні (оф) x Видаткові накладні оф x

Filter
Q Оновити Додати умову пошуку Show rows 50

Створити Видати Excel Друк XML 1-50 rows of [?] >>

Рахунок фірми	Рахунок контрагента	Номер	Дата	Представник виконавця	П
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	СТОВ Колос, 26006303211503 у ПАТ"Промінвестбанк"	42	29/12/2020 17:45		
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ	Товариство з обмеженою відповідальністю "Агропартнер-1", UA08380805000000026003379082 у АТ "РАЙФАЗЕН БАНК АВАЛЬ"	27	27/10/2020 09:10		
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	"КРАСИЛІВСЬКЕ ПІДПРИЄМСТВО ТЕПЛОВИХ МЕРЕЖ", UA753154050000026004052422062 у ПАТ КБ "Приватбанк" м.Хмельницький	19	10/11/2020 15:39		
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ	"КРАСИЛІВСЬКЕ ПІДПРИЄМСТВО ТЕПЛОВИХ МЕРЕЖ", UA753154050000026004052422062 у ПАТ КБ "Приватбанк" м.Хмельницький	10	01/09/2020 09:47		
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ДП "Красилівський агрегатний завод", UA273223130000026007000006480 у Ф-я ВАТ "УКРЕКСІМБАНК" м.Хмельниц	227	06/04/2021 17:05		
ФО-П Серветник Віталій Віталійович,	ДП "Красилівський агрегатний завод",		06/04/2021		

Рисунок 3.14 – Вікно видаткових накладних

2.2 Потрібно ввести інформацію у відповідні поля Рисунок. 3.15 – «Заповнення відповідних полів», перейти до поля «Товари» та натиснути «Створити». Рисунок 3.16 – Заповнення інформації для видаткової накладної

ЦСК 192.168.1.31:8080

Портал ЦСК

Запити веб користувачів
На створення аккаунту
Інформаційні блоки
Довідники
Первинні документи
Загальні
Офіційні
Договора
Рахунки
Прихідні накладні (оф)
Видаткові накладні (оф)
Акти (оф)
Каса (оф)
Кошти в банку
РРО
ПРРО

Прихідні накладні (оф) x Видаткова накладна оф редактор x

Видаткові накладні оф > Видаткова накладна оф редактор

Рахунок фірми ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛ

Рахунок контрагента

Номер 63 Віддруковано Сум соб

Дата 12/04/2021 13:05 Оформленні Собівартість прописом

Платник Оформлено Сума

Замовлення В наявності Сума прописом

Умова продажу

Представник виконавця

Принняв

Довіреність

Місце складання

Номер ПН 18

Дата ПН 12/04/2021

Номер договору

Дата договору

Місце зберігання

Коментар

OK Відмінити

Рисунок 3.15 – Заповнення відповідних полів

Портал ЦСК

Запити веб користувачів

На створення аккаунту

Інформаційні блоки

Довідники

Первинні документи

Загальні

Офіційні

Договора

Рахунки

Прихідні накладні (оф)

Видаткові накладні (оф)

Акти (оф)

Каса (оф)

Кошти в банку

РРО

ПРРО

Серветник Юлія Віт...

Прихідні накладні (оф) x Видаткова накладна оф редактор x

Видаткові накладні оф > Видаткова накладна оф редактор

Прийняв

Довірненість

Місце складання

Номер ПН

Дата ПН

Номер договору

Дата договору

Місце зберігання

Коментар

Сума ПДВ

Сума без ПДВ

Товари

Створити Редагувати Видалити

Товар	Собівартість	Кількість	Ціна б/ПДВ	ПДВ	Ціна	Собівартість сума	Сума
						0	0

OK Відмінити

Рисунок 3.16 – Заповнення інформації для видаткової накладної

2.3 У вікні про товари ввести необхідну інформацію про товар та суми товару. Щоб все зберегти – вибрати «ОК». Рисунок 3.17 – «Заповнення інформації про товар вихідної накладної»

Портал ЦСК

Запити веб користувачів

На створення аккаунту

Інформаційні блоки

Довідники

Первинні документи

Загальні

Офіційні

Договора

Рахунки

Прихідні накладні (оф)

Видаткові накладні (оф)

Акти (оф)

Каса (оф)

Кошти в банку

РРО

ПРРО

Серветник Юлія Віт...

Прихідні накладні (оф) x Товар вид наклад (оф) редактор x

Видаткові накладні оф > Видаткова накладна оф редактор > Товар вид наклад (оф) редактор

Товар

Кількість

Собівартість

Ціна б/ПДВ

ПДВ

Ціна

Собівартість сума

Сума без ПДВ

Сума ПДВ

Сума

OK Відмінити По останньому надходженню

Рисунок 3.17 – Заповнення інформації про товар вихідної накладної

2.4 Щоб роздрукувати необхідну інформацію видаткової накладної, в вікні «Видаткові накладні(оф)» вибрати потрібну накладну та вибрати «Друк» Рисунок 3.18 – Друк необхідної інформації вихідної накладної

The screenshot shows the 'Портал ЦСК' (ЦСК 192.168.1.31:8080) interface. The main content area displays a table of 'Видаткові накладні (оф)'. The table has columns for 'Рахунок фірми', 'Рахунок контрагента', 'Номер', 'Дата', 'Представник виконавця', and 'Підпис'. The second row is highlighted in blue, and a red arrow points to it. In the top toolbar, a red arrow points to the 'Друк' (Print) button.

Рахунок фірми	Рахунок контрагента	Номер	Дата	Представник виконавця	Підпис
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	СТОВ Колос, 26006303211503 у ПАТ "Промінвестбанк"	42	29/12/2020 17:45		
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ	Товариство з обмеженою відповідальністю "Агропартнер-1", UA08380805000000026003379082 у АТ "РАЙФАЗЕН БАНК АВАЛЬ"	27	27/10/2020 09:10		
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	"КРАСИЛІВСЬКЕ ПІДПРИЄМСТВО ТЕПЛОВИХ МЕРЕЖ", UA753154050000026004052422062 у ПАТ КБ "Приватбанк" м.Хмельницький	19	10/11/2020 15:39		
ПРИВАТНЕ МАЛЕ ПІДПРИЄМСТВО "Універсал", UA113157840000026001300799535 у ФХМЕЛЬНИЦЬКЕ ОБЛАСН ВАТОЩАД М.ХМЕЛЬНИЦ	"КРАСИЛІВСЬКЕ ПІДПРИЄМСТВО ТЕПЛОВИХ МЕРЕЖ", UA753154050000026004052422062 у ПАТ КБ "Приватбанк" м.Хмельницький	10	01/09/2020 09:47		
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ДП "Красилівський агрегатний завод", UA273223130000026007000006480 у Ф-я ВАТ "УКРЕКСІМБАНК" м.Хмельниц	227	06/04/2021 17:05		
ФО-П Серветник Віталій Віталійович, UA293154050000026002052408602 у ПАТ КБ "Приватбанк" м.Хмельницький	ДП "Красилівський агрегатний завод", UA273223130000026007000006480 у Ф-я ВАТ "УКРЕКСІМБАНК" м.Хмельниц		06/04/2021		

Рисунок 3.18 – Друк необхідної інформації вихідної накладної

3.4 Тестування інформаційної системи

Інтеграційні тести веб-рівня

Інтеграційні тести веб-рівня виконуються в контейнері Spring блоку Web Client. Тестовий контейнер працює незалежно від середнього шару, так як фреймворк автоматично створює заглушки (stubs) для всіх сервісів. Тестова інфраструктура складається з наступних класів, розташованих в пакеті com.haulmont.cuba.web.testsupport і його вкладених пакетах:

1. TestContainer – обгортка навколо контейнера Spring для використання в якості базового класу в проектах.
2. TestServiceProху – надає заглушки за замовчуванням для сервісів середнього шару. Даний клас можна також використовувати для реєстрації моков сервісів, специфічних для тіста: см. Статичний метод mock ().

3. `DataServiceProxy` – заглушка за замовчуванням для `DataManager`. Містить реалізацію методу `commit ()`, яка імітує поведінку реального сховища: робить нові екземпляри сутностей `detached`, інкрементує версії, і т.д. Методи завантаження нічого не роблять і повертають `null` і порожні колекції.

4. `TestUiEnvironment` – надає набір методів для конфігурації і отримання `TestContainer`. Примірник даного класу повинен бути використаний в тестах як `JUnit 5 extension`.

5. `TestEntityFactory` – фабрика для зручного створення тестових екземплярів сутностей. Може бути отримана з `TestContainer`.

Тестовий контейнер для інтеграційних тестів веб-рівня:

```
package com.company.sc.web;

import com.haulmont.cuba.web.testsupport.TestContainer;

import java.util.Arrays;

public class ScWebTestContainer extends TestContainer {
    public ScWebTestContainer() {
        appComponents = Arrays.asList(
            "com.haulmont.cuba"
            // add CUBA add-ons and custom app components here
        );
        appPropertiesFiles = Arrays.asList(
            // List the files defined in your web.xml
            // in appPropertiesConfig context parameter of the web module
            "com/company/demo/web-app.properties",
            // Add this file which is located in CUBA and defines some properties
            // specifically for test environment. You can replace it with your own
            // or add another one in the end.
            "com/haulmont/cuba/web/testsupport/test-web-app.properties"
        );
    }
}

public static class Common extends ScWebTestContainer {

    // A common singleton instance of the test container which is initialized once for all tests
    public static final ScWebTestContainer.Common INSTANCE = new ScWebTestContainer.Common();

    private static volatile boolean initialized;

    private Common() {
    }
}
```

```

@Override
public void before() throws Throwable {
    if (!initialized) {
        super.before();
        initialized = true;
    }
    setupContext();
}

@Override
public void after() {
    cleanupContext();
    // never stops - do not call super
}
}
}

```

Тест, який перевіряє стан редагованої в екрані суті (Customer) після деякої дії користувача:

```

package com.company.sc.web;

import com.company.sc.entity.catalogs.customer.Customer;
import com.company.sc.web.screens.catalogs.customer.CustomerEdit;
import com.haulmont.cuba.gui.Screens;
import com.haulmont.cuba.gui.screen.OpenMode;
import com.haulmont.cuba.web.app.main.MainScreen;
import com.haulmont.cuba.web.testsupport.TestEntityFactory;
import com.haulmont.cuba.web.testsupport.TestEntityState;
import com.haulmont.cuba.web.testsupport.TestUiEnvironment;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.RegisterExtension;

import java.util.Collections;

public class CustomerEditInteractionTest {
    @RegisterExtension
    TestUiEnvironment environment =
        new
    TestUiEnvironment(ScWebTestContainer.Common.INSTANCE).withUserLogin("admin");

    private Customer customer;

    @BeforeEach
    public void setUp() throws Exception {

```

```

TestEntityFactory<Customer> customersFactory =
    environment.getContainer().getEntityFactory(Customer.class, TestEntityState.NEW);

customer = customersFactory.create(Collections.emptyMap());
}

@Test
public void testGenerateName() {
    Screens screens = environment.getScreens();

    screens.create(MainScreen.class, OpenMode.ROOT).show();

    CustomerEdit customerEdit = screens.create(CustomerEdit.class);
    customerEdit.setEntityToEdit(customer);
    customerEdit.show();

    Assertions.assertNull(customerEdit.getEditedEntity().getName());

    Button generateBtn = (Button) customerEdit.getWindow().getComponent("generateBtn");
    customerEdit.onGenerateBtnClick(new Button.ClickEvent(generateBtn));

    Assertions.assertEquals("Generated name", customerEdit.getEditedEntity().getName());
}
}

```

Тест, який перевіряє коректність завантаження даних в екран UI:

```

package com.company.sc.web;

import com.company.sc.entity.catalogs.customer.Customer;
import com.company.sc.web.screens.catalogs.customer.CustomerEdit;
import com.haulmont.cuba.core.app.DataService;
import com.haulmont.cuba.core.entity.Entity;
import com.haulmont.cuba.core.global.LoadContext;
import com.haulmont.cuba.gui.Screens;
import com.haulmont.cuba.gui.model.InstanceContainer;
import com.haulmont.cuba.gui.screen.OpenMode;
import com.haulmont.cuba.gui.screen.UiControllerUtils;
import com.haulmont.cuba.web.app.main.MainScreen;
import com.haulmont.cuba.web.testsupport.TestEntityFactory;
import com.haulmont.cuba.web.testsupport.TestEntityState;
import com.haulmont.cuba.web.testsupport.TestUiEnvironment;
import com.haulmont.cuba.web.testsupport.proxy.TestServiceProxy;
import mockit.Delegate;
import mockit.Expectations;
import mockit.Mocked;

```

```

import org.junit.jupiter.api.AfterEach;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.RegisterExtension;

import static org.junit.Assert.assertEquals;

public class CustomerEditLoadDataTest {
    @RegisterExtension
    TestUiEnvironment environment =
        new
TestUiEnvironment(ScWebTestContainer.Common.INSTANCE).withUserLogin("admin");

    @Mocked
    private DataService dataService;

    private Customer customer;

    @BeforeEach
    public void setUp() throws Exception {
        new Expectations() {{
            dataService.load((LoadContext<? extends Entity>) any);
            result = new Delegate() {
                Entity load(LoadContext lc) {
                    if ("demo_Customer".equals(lc.getEntityMetaClass())) {
                        return customer;
                    } else
                        return null;
                }
            };
        }};

        TestServiceProxy.mock(DataService.class, dataService);

        TestEntityFactory<Customer> customersFactory =
            environment.getContainer().getEntityFactory(Customer.class,
TestEntityState.DETACHED);

        customer = customersFactory.create(
            "name", "Homer", "email", "homer@simpson.com");
    }

    @AfterEach
    public void tearDown() throws Exception {
        TestServiceProxy.clear();
    }
}

```

```

@Test
public void testLoadData() {
    Screens screens = environment.getScreens();

    screens.create(MainScreen.class, OpenMode.ROOT).show();

    CustomerEdit customerEdit = screens.create(CustomerEdit.class);
    customerEdit.setEntityToEdit(customer);
    customerEdit.show();

    InstanceContainer customerDc =
UiControllerUtils.getScreenData(customerEdit).getContainer("customerDc");
    assertEquals(customer, customerDc.getItem());
}
}

```

Інтеграційні тести Middleware

Інтеграційні тести Middleware виконуються в повнофункціональному контейнері Spring з підключенням до бази даних. У тестах такого типу можна виконувати код будь-якого шару всередині Middleware – від сервісів до ORM.

Тестовий контейнер для інтеграційних тестів Middleware:

```

package com.company.sc;

import com.haulmont.cuba.testsupport.TestContainer;
import org.junit.jupiter.api.extension.ExtensionContext;

import java.util.ArrayList;
import java.util.Arrays;

public class ScTestContainer extends TestContainer {

    public ScTestContainer() {
        super();
        //noinspection ArraysAsListWithZeroOrOneArgument
        appComponents = Arrays.asList(
            "com.haulmont.cuba",
            "com.haulmont.reports",
            "de.diedavids.cuba.dataimport",
            "com.haulmont.addon.emailtemplates",
            "com.haulmont.addon.helium",
            "com.haulmont.addon.dashboard",
            "com.haulmont.addon.restapi");
    }
}

```

```

appPropertiesFiles = Arrays.asList(
    // List the files defined in your web.xml
    // in appPropertiesConfig context parameter of the core module
    "com/company/sc/app.properties",
    // Add this file which isAn located in CUBA and defines some properties
    // specifically for test environment. You can replace it with your own
    // or add another one in the end.
    "com/company/sc/test-app.properties");
autoConfigureDataSource();
}

public static class Common extends ScTestContainer {

    public static final ScTestContainer.Common INSTANCE = new ScTestContainer.Common();

    private static volatile boolean initialized;

    private Common() {
    }

    @Override
    public void beforeAll(ExtensionContext extensionContext) throws Exception {
        if (!initialized) {
            super.beforeAll(extensionContext);
            initialized = true;
        }
        setupContext();
    }

    @SuppressWarnings("RedundantThrows")
    @Override
    public void afterAll(ExtensionContext extensionContext) throws Exception {
        cleanupContext();
        // never stops - do not call super
    }

}
}

```

Тест, який перевіряє коректність створення, завантаження та видалення сутності Customer:

```

package com.company.sc.core;

import com.company.sc.ScTestContainer;
import com.company.sc.entity.catalogs.customer.Customer;
import com.haulmont.cuba.core.entity.contracts.Id;
import com.haulmont.cuba.core.global.AppBeans;
import com.haulmont.cuba.core.global.DataManager;
import org.junit.jupiter.api.BeforeAll;
import org.junit.jupiter.api.Test;
import org.junit.jupiter.api.extension.RegisterExtension;

import static org.junit.Assert.assertEquals;

public class CustomerTest {
    // Using the common singleton instance of the test container which is initialized once for all tests
    @RegisterExtension
    static ScTestContainer cont = ScTestContainer.Common.INSTANCE;

    static DataManager dataManager;

    @BeforeAll
    static void beforeAll() {
        // Get a bean from the container
        dataManager = AppBeans.get(DataManager.class);
    }

    @Test
    void testCreateLoadRemove() {
        Customer customer = cont.metadata().create(Customer.class);
        customer.setName("c1");

        Customer committedCustomer = dataManager.commit(customer);
        assertEquals(customer, committedCustomer);

        Customer loadedCustomer = dataManager.load(Id.of(customer)).one();
        assertEquals(customer, loadedCustomer);

        dataManager.remove(loadedCustomer);
    }
}

```

Тест, який перевіряє коректність завантаження сутності User з БД:

```

package com.company.sc.core;

import com.company.sc.ScTestContainer;

```

```

import com.haulmont.cuba.core.EntityManager;
import com.haulmont.cuba.core.Persistence;
import com.haulmont.cuba.core.Transaction;
import com.haulmont.cuba.core.TypedQuery;
import com.haulmont.cuba.core.global.AppBeans;
import com.haulmont.cuba.core.global.DataManager;
import com.haulmont.cuba.core.global.Metadata;
import com.haulmont.cuba.security.entity.User;
import org.junit.jupiter.api.*;
import org.junit.jupiter.api.extension.RegisterExtension;

import java.util.List;

public class UserTest {

    @RegisterExtension
    public static ScTestContainer cont = ScTestContainer.Common.INSTANCE;

    private static Persistence persistence;

    @BeforeAll
    public static void beforeAll() throws Exception {
        Metadata metadata = cont.metadata();
        persistence = cont.persistence();
        DataManager dataManager = AppBeans.get(DataManager.class);
    }

    @AfterAll
    public static void afterAll() throws Exception {
    }

    @Test
    public void testLoadUser() {
        try (Transaction tx = persistence.createTransaction()) {
            EntityManager em = persistence.getEntityManager();
            TypedQuery<User> query = em.createQuery(
                "select u from sec$User u where u.login = :userLogin", User.class);
            query.setParameter("userLogin", "admin");
            List<User> users = query.getResultList();
            tx.commit();
            Assertions.assertEquals(1, users.size());
        }
    }
}

```

3.5 Вимоги до апаратних та програмних засобів

Системні вимоги:

- 64-бітна операційна система – Windows, Linux або macOS;
- оперативна пам'ять – 8 ГБ;
- вільне місце на жорсткому диску – 10 ГБ.

Вимоги до ПЗ:

- встановлений сервер додатків Apache Tomcat не старше 8-ї версії;
- встановлена СУБД PostgreSQL (протестовано на 12-й версії);
- встановлений Java SE development Kit (JDK) не старше 8-ї версії.

Веб-браузер:

Веб-інтерфейс додатків, створюваних на основі платформи, підтримує всі популярні сучасні браузери, в тому числі Google Chrome, Mozilla Firefox, Safari, Opera 15+, Internet Explorer 11, Microsoft Edge.

Висновки

В ході виконання кваліфікаційної роботи бакалавра було розроблено функціональне програмне забезпечення, повністю готове до застосування. Дане програмне забезпечення орієнтоване для організації складу в магазинах та інтернет магазинах. З його допомогою працівники зможуть отримувати необхідну та розширену інформацію про надходження та реалізування товару або послуги.

При розробці були проаналізовані сучасні програмне забезпечення, які забезпечують швидкість та комфорт на фірмах, магазинах. Це допомогло оцінити функціонал різноманітного програмного забезпечення, та дійти до створення свого.

Розроблене програмне забезпечення задовольняє всім, поставленим на етапі постановки задачі, вимогам. При розробці було проаналізовані інформаційні потоки та визначено бізнес-процеси, які необхідно автоматизувати. Також були розроблені структури інформаційних систем для того, щоб уникати повторного опису однотипних атрибутів у схожих за своїм набором атрибутів основних сутностей. Для вибору мови програмування були виділені плюси вибраної мови. Також був проаналізований фреймворк та вибрана платформа, на якій реалізовувалось програмне забезпечення.

Для програмної реалізації було розроблено схеми та діаграми, щоб показати структура та функціонал модулів системи, також створювались сервіси для інформаційного обміну в системі. В одному з підрозділів була розписана інструкція для користувача, яка забезпечує легке користування системою.

Також програма була протестована, для забезпечення справності.

Зібравши всю інформацію та написаний код, було розроблено повнофункціональну програму для автоматизації складу малих та середніх підприємств та фірм.

Перелік посилань

- 1 Мишин В.М. Исследование систем управления [Текст] : учеб пособие / В.М.Мишин – М. : Юнити, 2010
- 2 Аналіз предметної області, чому потрібна автоматизація підприємств та складів/ <https://inteltech.com.ua/ru/blogs/v-chem-preimushchestva-avtomatizacii-roznicnogo-magazina>
- 3 Новак В. О. Інформаційне забезпечення менеджменту : навч. посіб. / В.О. Новак, Л. Г. Макаренко, М. Г. Луцький. – К. : Кондор, 2008.
- 4 Програма «Складський облік товарів» [Електронний ресурс] // Режим доступу: <http://business-soft.net/uk/1c-inventory-control.html> – Назва з екрану.
- 5 Система jSolutions / https://jsolutions.ua/ua/sistema-upravleniya-skladom?gclid=CjwKCAiApNSABhAlEiwANuR9YJZT8z_gKKzCPZkF664QspzH2c eA7mBmITqKD6kVURTvMHspStv2BoCYCoQAvD_BwE / Система jSolutions
- 6 Програма УкрСклад / Про програма для магазина и склада/ <https://acctech.com.ua/product/ukrsklad-pro-programma-dlya-magazina-i-sklada/>
- 7 Конспект лекцій з дисципліни «Організація баз даних» / <http://dspace.wunu.edu.ua/jspui/bitstream/316497/9109/1/%D0%BE%D0%BF%D0%B E%D1%80%D0%BD%D0%B8%D0%B9%20%D0%BA%D0%BE%D0%BD%D1%81 %D0%BF%D0%B5%D0%BA%D1%82%20%D0%BB%D0%B5%D0%BA%D1%86% D1%96%D0%B9.pdf>
- 8 Фаро С., Лерми П. Рефакторинг SQLite-приложений/ С. Фаро, П. Лерми – Симваол-Плюс, 2006. – 180 с.
- 9 Microsoft Access обзор, плюси та мінуси/ <https://fruitsekta.ru/access/ms-access-dostoinstva-i-nedostatki.html>
- 10 Гешвинде. PostgreSQL рук-во разраб и администратора/ Гешвинде – ДиаСофт, 2010. – 344 с.
- 11 Тахагхогхи С., Вильянс Хью Е. Руководство по MySQL/ С. Тахагхогхи, Е. Вильянс Хью – Русская редакция, 2006. – 544 с.
- 12 Мова програмування Java/ <https://medium.com/nuances-of-programming/плюсы-и-минусы-программирования-на-java-2861f4c2a0d5/>

13 Cuba-platform/ <https://www.cuba-platform.ru/>

14 Object-Relational Mapping/ [https://ru.bmstu.wiki/ORM_\(Object-Relational_Mapping\)](https://ru.bmstu.wiki/ORM_(Object-Relational_Mapping))

ДОДАТКИ

Додаток А: Лістинг

При розробці модулів були задіяні можливості платформи по генерації програмного коду. Найбільш ефективно вони застосовуються при створенні модулів первинних документів, де присутній великий об'єм стандартного коду по обробці даних.

До прикладу модуль «Банки» практично не містить коду, написаного вручну:

- Дескриптор екрану

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<windowxmlns="http://schemas.haulmont.com/cuba/screen/window.xsd"
  xmlns:c="http://schemas.haulmont.com/cuba/screen/jpql_condition.xsd"
  caption="msg://browseCaption"
  focusComponent="banksTable"
  messagesPack="com.company.sc.web.screens.catalogs.bank">
<data readOnly="true">
  <collection id="banksDc"
    class="com.company.sc.entity.catalogs.Bank"
    view="_local">
    <loader id="banksDI">
      <query>
        <![CDATA[select e from sc_Bank e]]>
      </query>
    </loader>
    </collection>
</data>
<dialogMode height="600"
  width="800"/>
<layout expand="banksTable"
  spacing="true">
  <filter id="filter"
    applyTo="banksTable"
    dataLoader="banksDI">
    <properties include=".*"/>
  </filter>
  <groupTable id="banksTable"
    width="100%"
    dataContainer="banksDc">
    <actions>
      <action id="create" type="create"/>
      <action id="edit" type="edit"/>
      <action id="remove" type="remove"/>
    </actions>
    <columns>
      <column id="name"/>
      <column id="shortName"/>
      <column id="mfo"/>
```

```

</columns>
<rowCount/>
<buttonsPanel id="buttonsPanel"
    alwaysVisible="true">
    <button id="createBtn" action="banksTable.create"/>
    <button id="editBtn" action="banksTable.edit"/>
    <button id="removeBtn" action="banksTable.remove"/>
</buttonsPanel>
</groupTable>
<hbox id="lookupActions" spacing="true" visible="false">
    <button action="lookupSelectAction"/>
    <button action="lookupCancelAction"/>
</hbox>
</layout>
</window>

```

- Контролер екрану:

```
package com.company.sc.web.screens.catalogs.bank;
```

```
import com.haulmont.cuba.gui.screen.*;
import com.company.sc.entity.catalogs.Bank;
```

```

@UiController("sc_Bank.browse")
@UiDescriptor("bank-browse.xml")
@LookupComponent("banksTable")
@LoadDataBeforeShow
public class BankBrowse extends StandardLookup<Bank> {
}

```

- Єдине на чому потрібно зосередитись – ретельно описати клас моделі таблиці у БД:

-

```
□ □ package com.company.sc.entity;
```

```
import com.haulmont.chile.core.annotations.NamePattern;
import com.haulmont.cuba.core.entity.StandardEntity;
```

```
import javax.persistence.Column;
import javax.persistence.MappedSuperclass;
import javax.validation.constraints.NotNull;
```

```

@MappedSuperclass
@NamePattern("%s|name")
public class NamedEntity extends StandardEntity {
    private static final long serialVersionUID = -4023663287077857461L;

```

```

    @NotNull
    @Column(name = "NAME", nullable = false)
    protected String name;

```

```

    @Column(name = "SHORT_NAME")
    protected String shortName;

```

```

public String getShortName() {
    return shortName;
}

public void setShortName(String shortName) {
    this.shortName = shortName;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}
}

```

```

package com.company.sc.entity.catalogs;

```

```

import com.company.sc.entity.NamedEntity;
import com.haulmont.chile.core.annotations.NamePattern;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Table;
import javax.validation.constraints.NotNull;

@Table(name = "SC_BANK")
@NamePattern("%s|name")
@Entity(name = "sc_Bank")
public class Bank extends NamedEntity {
    private static final long serialVersionUID = -3148163825277727308L;

    @Column(name = "CLARA_ID")
    protected Long claraId;

    @NotNull
    @Column(name = "MFO", nullable = false)
    protected String mfo;

    public Long getClaraId() {
        return claraId;
    }

    public void setClaraId(Long claraId) {
        this.claraId = claraId;
    }

    public String getMfo() {
        return mfo;
    }
}

```

```

public void setMfo(String mfo) {
    this.mfo = mfo;
}
}

```

На відміну від цих модулів більшість сервісів вимагають «ручного» коду. Сервіси дозволяють обробляти великі масиви даних на сервері, віддаючи клієнтові тільки результати обчислень. Цей механізм радикально зменшує мережевий трафік та дозволяє ефективно працювати з додатком як у локальній мережі організації так і через мережу Інтернет.

Наведений нижче приклад ілюструє сервіс, що здійснює створення різних типів документів з одного вихідного

```

package com.company.sc.service.copyTo;

import com.company.sc.entity.catalogs.Department;
import com.company.sc.entity.catalogs.Option;
import com.company.sc.entity.catalogs.Stat;
import com.company.sc.entity.catalogs.firm.Firm;
import com.company.sc.entity.catalogs.product.Product;
import com.company.sc.entity.source.money.MoneyCash;
import com.company.sc.entity.source.proposal.Proposal;
import com.company.sc.entity.source.proposal.ProposalDetail;
import com.company.sc.entity.source.serviceact.ServiceAct;
import com.company.sc.entity.source.serviceact.ServiceActDetail;
import com.company.sc.entity.source.wayinvoice.WayInvoice;
import com.company.sc.entity.source.wayinvoice.WayInvoiceDetail;
import com.company.sc.service.GetMainEntityService;
import com.company.sc.service.GetOptionsService;
import com.company.sc.service.NextValueService;
import com.company.sc.service.NumbersInWordsService;
import com.company.sc.service.manipulationdata.CalcSalaryService;
import com.haulmont.cuba.core.global.*;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import org.springframework.stereotype.Service;

import javax.inject.Inject;
import java.math.BigDecimal;
import java.util.Date;
import java.util.List;
import java.util.UUID;

@Service(CopyToService.NAME)
public class CopyToServiceBean implements CopyToService {

```

```

private static final Logger log = LoggerFactory.getLogger(CopyToServiceBean.class);
@Inject
private Metadata metadata;
@Inject
private NextValueService nextValueService;
@Inject
private TimeSource timeSource;
@Inject
private NumbersInWordsService numbersInWordsService;
@Inject
private GetMainEntityService getMainEntityService;
@Inject
private DataManager dataManager;
@Inject
private CalcSalaryService calcSalaryService;

BigDecimal sumPrice = BigDecimal.ZERO;
BigDecimal sum = BigDecimal.ZERO;
@Inject
private GetOptionsService getOptionsService;

/**
 * Copying an entity to general documents
 * @param uuid UUID of the entity being copied
 * @param nameEntity name of the target entity
 * @param login current login
 * @param dateDoc document date
 */
@Override
public Boolean copyTo(UUID uuid, String nameEntity, String login, Date dateDoc) {
    boolean boolResult = false;
    try {
        Department dep = getMainDepartment();
        Proposal source = null;
        WayInvoice wayInvoice = null;
        ServiceAct serviceAct = null;

        switch (nameEntity) {
            case "SC_SERVICE_ACT":
            case "SC_WAY_INVOICE":
                source = getProposalById(uuid);
                break;
            case "SC_MONEY_CASH":
                wayInvoice = getWayInvoiceById(uuid);
                break;
            case "SERVICE_ACT_TO_SC_MONEY_CASH":
                serviceAct = getServiceActById(uuid);
                break;
        }
    }

    //CommitContext commitContext;

```

```

switch (nameEntity) {
    case "SC_SERVICE_ACT":
        if (!source.getCopyToAct()) {
            sumPrice = BigDecimal.ZERO;
            sum = BigDecimal.ZERO;
            CommitContext commitContext = new CommitContext();
            ServiceAct result = metadata.create(ServiceAct.class);
/*
            result.setVersion(1);
            result.setCreateTs(timeSource.currentTimestamp());
*/
            result.setCreatedBy(login);
/*
            result.setUpdatedBy(null);
            result.setUpdateTs(null);
*/
            result.setService(source.getService());
            result.setCustomer(source.getCustomer());
            if (dep != null) result.setDepartment(dep);
            else break;
            result.setNumber(nextValueService.NextValue("SC_SERVICE_ACT", "NUMB
ER"));
            if (dateDoc != null) result.setDate(dateDoc);
            else result.setDate(source.getReleasedDate());
            result.setComment("Копія заявки № "
                + source.getNumber()
                + " від " + numbersInWordsService.DateToStringUa(source.getDate())
+ "р.");
            result.setDiscount(source.getDiscount());
            if (source.getFirmRepresentative() != null)
                result.setFirmRepresentative(source.getFirmRepresentative());
            result.setCustomerRepresentative(source.getWebUser().getName());

            for (ProposalDetail detail : source.getProposalDetail()) {
                BigDecimal sumDetail;
                BigDecimal sumPriceDetail;
                ServiceActDetail newDetail = metadata.create(ServiceActDetail.class);
                newDetail.setServiceAct(result);
                //newDetail.setVersion(1);
                newDetail.setCreatedBy(login);
/*
                newDetail.setCreateTs(timeSource.currentTimestamp());
                newDetail.setUpdatedBy(null);
                newDetail.setUpdateTs(null);
*/
                newDetail.setQuantity(detail.getQuantity());
                newDetail.setCostPrice(detail.getCostPrice());
                Product p = detail.getProduct();
                newDetail.setProduct(p);
                newDetail.setEmployee(detail.getEmployee());
                newDetail.setDiscount(detail.getDiscount());
                newDetail.setCost(detail.getCost());

```

```

        sumPriceDetail = newDetail.getCostPrice().multiply(newDetail.getQuantity());
        newDetail.setSumPrice(sumPriceDetail);
        sumDetail = newDetail.getCost().multiply(newDetail.getQuantity());
        newDetail.setSum(sumDetail);
        if(sumDetail.compareTo(BigDecimal.valueOf(0)) > 0) {
            List<BigDecimal> salaryList = calcSalaryService.calcSalaryData(newDetail
);
                newDetail.setSalaryPercent(salaryList.get(0));
                newDetail.setSalary(salaryList.get(1));
            }
            sumPrice = sumPrice.add(sumPriceDetail);
            sum = sum.add(sumDetail);
            commitContext.addInstanceToCommit(newDetail);
        }
        result.setSumPrice(sumPrice);
        result.setSumPriceInWords(numbersInWordsService.inWords(sumPrice.doubleV
alue()));
        result.setSum(sum);
        result.setSumInWords(numbersInWordsService.inWords(sum.doubleValue()));
        result.setDifference(sum.subtract(sumPrice));
        result.setDifferenceInWords(numbersInWordsService.inWords(sum.subtract(sum
Price).doubleValue()));
        commitContext.addInstanceToCommit(result);
        dataManager.commit(commitContext);
        modifySource(nameEntity, source, result.getNumber(), result.getDate());
        boolResult = true;
        break;
    }
    case "SC_WAY_INVOICE":
        if (!source.getCopyToWay()) {
            sumPrice = BigDecimal.ZERO;
            sum = BigDecimal.ZERO;
            CommitContext commitContext = newCommitContext();
            WayInvoice resultWi = metadata.create(WayInvoice.class);
/*
            resultWi.setVersion(1);
            resultWi.setCreateTs(timeSource.currentTimestamp());
*/
            resultWi.setCreatedBy(login);
/*
            resultWi.setUpdatedBy(null);
            resultWi.setUpdateTs(null);
*/
            resultWi.setCustomer(source.getCustomer());
            if (dep != null) resultWi.setDepartment(dep);
            else break;
            resultWi.setNumber(nextValueService.NextValue("SC_WAY_INVOICE", "NU
MBER"));
            if (dateDoc != null) resultWi.setDate(dateDoc);
            elseresultWi.setDate(source.getReleasedDate());
            resultWi.setComment("Копія заявки № "
                + source.getNumber()
                + " від " + numbersInWordsService.DateToStringUa(source.getDate())
+ "р.");

```

```

resultWi.setDiscount(source.getDiscount());
if (source.getFirmRepresentative() != null)
    resultWi.setFirmRepresentative(source.getFirmRepresentative());
resultWi.setCustomerRepresentative(source.getWebUser().getName());

for (ProposalDetail detail : source.getProposalDetail()) {
    BigDecimal sumDetail;
    BigDecimal sumPriceDetail;
    WayInvoiceDetail newDetail = metadata.create(WayInvoiceDetail.class);
    newDetail.setWayInvoice(resultWi);
    newDetail.setVersion(1);
    newDetail.setCreatedBy(login);
    newDetail.setCreateTs(timeSource.currentTimestamp());
    newDetail.setUpdatedBy(null);
    newDetail.setUpdateTs(null);

    newDetail.setQuantity(detail.getQuantity());
    newDetail.setCostPrice(detail.getCostPrice());
    Product p = detail.getProduct();
    newDetail.setProduct(p);
    newDetail.setEmployee(detail.getEmployee());
    newDetail.setDiscount(detail.getDiscount());
    newDetail.setCost(detail.getCost());
    sumPriceDetail = newDetail.getCostPrice().multiply(newDetail.getQuantity());
    newDetail.setSumPrice(sumPriceDetail);
    sumDetail = newDetail.getCost().multiply(newDetail.getQuantity());
    newDetail.setSum(sumDetail);
    if(sumDetail.compareTo(BigDecimal.valueOf(0)) > 0) {
        List<BigDecimal> salaryList = calcSalaryService.calcSalaryData(newDetail
);
        newDetail.setSalaryPercent(salaryList.get(0));
        newDetail.setSalary(salaryList.get(1));
    }
    sumPrice = sumPrice.add(sumPriceDetail);
    sum = sum.add(sumDetail);
    commitContext.addInstanceToCommit(newDetail);
}
resultWi.setSumPrice(sumPrice);
resultWi.setSumPriceInWords(numbersInWordsService.inWords(sumPrice.double
eValue()));
resultWi.setSum(sum);
resultWi.setSumInWords(numbersInWordsService.inWords(sum.doubleValue()
));
resultWi.setDifference(sum.subtract(sumPrice));
resultWi.setDifferenceInWords(numbersInWordsService.inWords(sum.subtract(s
umPrice).doubleValue()));
commitContext.addInstanceToCommit(resultWi);
dataManager.commit(commitContext);
modifySource(nameEntity, source, resultWi.getNumber(), resultWi.getDate());
boolResult = true;
break;
}
case "SC_MONEY_CASH":

```

```

assert wayInvoice != null;
if (!wayInvoice.getCopyToMoneyCash()) {
    sumPrice = BigDecimal.ZERO;
    sum = BigDecimal.ZERO;
    CommitContext commitContext = new CommitContext();
    MoneyCash resultMc = metadata.create(MoneyCash.class);
/*
    resultMc.setVersion(1);
    resultMc.setCreateTs(timeSource.currentTimestamp());
*/
    resultMc.setCreatedBy(login);
/*
    resultMc.setUpdatedBy(null);
    resultMc.setUpdateTs(null);
*/
    resultMc.setCustomer(wayInvoice.getCustomer());
    resultMc.setDepartment(wayInvoice.getDepartment());
    resultMc.setFirm(getFirmById(getMainEntityService.getMainEntityUuid("SC_F
IRM", "MAIN"))));
    Option op = getOptionsService.GetOptions();
    resultMc.setEmployee(op.getEmployeeProduct());
    resultMc.setStat(getStatId(getMainEntityService.getMainEntityUuid("SC_STAT
", "IS_MAIN_TO_PROPOSAL"))));
    resultMc.setNumber(nextValueService.NextValue("SC_MONEY_CASH", "NU
MBER"));
    if (dateDoc != null) resultMc.setDate(dateDoc);
    else resultMc.setDate(wayInvoice.getDate());
    resultMc.setComment("Копія накладної № "
        + wayInvoice.getNumber()
        + " від " + numbersInWordsService.DateToStringUa(wayInvoice.getDate())
+ "р.");

    resultMc.setSum(wayInvoice.getSum());
    resultMc.setSumInWords(wayInvoice.getSumInWords());
    commitContext.addInstanceToCommit(resultMc);
    dataManager.commit(commitContext);
    modifyWayInvoiceSource(wayInvoice, resultMc.getNumber(), resultMc.getDate
());

    boolResult = true;
    break;
}
case "SERVICE_ACT_TO_SC_MONEY_CASH":
    assert serviceAct != null;
    Boolean copyToMoneyCash = serviceAct.getCopyToMoneyCash();
    if (copyToMoneyCash == null) copyToMoneyCash = false;
    if (!copyToMoneyCash) {
        sumPrice = BigDecimal.ZERO;
        sum = BigDecimal.ZERO;
        CommitContext commitContext = new CommitContext();
        MoneyCash resultMc = metadata.create(MoneyCash.class);
        //resultMc.setVersion(1);
        //resultMc.setCreateTs(timeSource.currentTimestamp());
        resultMc.setCreatedBy(login);
        //resultMc.setUpdatedBy(null);

```

```

        //resultMc.setUpdateTs(null);
        resultMc.setCustomer(serviceAct.getCustomer());
        resultMc.setDepartment(serviceAct.getDepartment());
        resultMc.setFirm(getFirmById(getMainEntityService.getMainEntityUuid("SC_F
IRM", "MAIN"))));
        Option op = getOptionsService.GetOptions();
        resultMc.setEmployee(op.getEmployeeProduct());
        resultMc.setStat(getStatId(getMainEntityService.getMainEntityUuid("SC_STAT
", "IS_MAIN_TO_PROPOSAL")));
        resultMc.setNumber(nextValueService.NextValue("SC_MONEY_CASH", "NU
MBER"));
        if (dateDoc != null) resultMc.setDate(dateDoc);
        elseresultMc.setDate(serviceAct.getDate());
        resultMc.setComment("Копія акту № "
            + serviceAct.getNumber()
            + " від " + numbersInWordsService.DateToStringUa(serviceAct.getDate())
+ "р.");

        resultMc.setSum(serviceAct.getSum());
        resultMc.setSumInWords(serviceAct.getSumInWords());
        commitContext.addInstanceToCommit(resultMc);
        dataManager.commit(commitContext);
        modifyServiceActSource(serviceAct, resultMc.getNumber(), resultMc.getDate())
;

        boolResult = true;
        break;
    }
    default:
        boolResult = false;
        break;
    }
} catch (Exception exception) {
    log.error("Error", exception);
    boolResult = false;
}
return boolResult;
}

/**
 * Update the source service act if the operation was successful
 * @param source source service act
 * @param number new entity number
 * @param date new entity date
 */
private void modifyServiceActSource(ServiceActsource, Long number, Date date) {
    CommitContext commitContext = new CommitContext();
    String comment;
    source.setCopyToMoneyCash(true);
    comment = "кacy ";
    if (source.getCommentCopy() == null) {
        source.setCommentCopy("Рознесено в " + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    }
}

```

```

    } else {
        source.setCommentCopy(source.getCommentCopy() + "," + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    }
    commitContext.addInstanceToCommit(source);
    dataManager.commit(commitContext);
}

/**
 * Update the source way invoice if the operation was successful
 * @param source source way invoice
 * @param number new entity number
 * @param date new entity date
 */
private void modifyWayInvoiceSource(WayInvoicesource, Long number, Date date) {
    CommitContext commitContext = new CommitContext();
    String comment;
    source.setCopyToMoneyCash(true);
    comment = "кacy ";
    if (source.getCommentCopy() == null) {
        source.setCommentCopy("Рознесено в " + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    } else {
        source.setCommentCopy(source.getCommentCopy() + "," + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    }
    commitContext.addInstanceToCommit(source);
    dataManager.commit(commitContext);
}

/**
 * Update the source proposal if the operation was successful
 * @param nameEntity the name of the entity being created
 * @param source source proposal
 * @param number new entity number
 * @param date new entity date
 */
private void modifySource(String nameEntity, Proposal source, Long number, Date date) {
    CommitContext commitContext = new CommitContext();
    String comment = "";

    switch (nameEntity) {
        case "SC_SERVICE_ACT":
            source.setCopyToAct(true);
            comment = "акт ";
            break;
        case "SC_WAY_INVOICE":

```

```

        comment = "накладну ";
        source.setCopyToWay(true);
        break;
    default:
        break;
    }
    if (source.getCommentCopy() == null) {
        source.setCommentCopy("Рознесено в " + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    } else {
        source.setCommentCopy(source.getCommentCopy() + ", " + comment + "№ "
            + number + " від "
            + numbersInWordsService.DateToStringUa(date)
            + "р.");
    }
    commitContext.addInstanceToCommit(source);
    dataManager.commit(commitContext);
}

/**
 * Returning active Proposal
 * @param uuid Proposal UUID
 * @return active Proposal
 */
private Proposal getProposalById(UUID uuid) {
    returndataManager.load(Proposal.class).id(uuid).view("proposal-edit").one();
}

/**
 * Returning main Firm
 * @param uuid Firm UUID
 * @return main Firm
 */
private Firm getFirmById(UUID uuid) {
    returndataManager.load(Firm.class).id(uuid).view("_minimal").one();
}

/**
 * Getting Main Department
 * @return the main department
 */
private Department getMainDepartment() {
    try {
        UUID mainDepartmentUuid = getMainEntityService.getMainEntityUuid("SC_DEPART
MENT"
            , "MAIN");
        returndataManager.load(Department.class).id(mainDepartmentUuid).view("_minimal").
one();
    } catch (Exception exception) {
        log.error("Error", exception);
        return null;
    }
}

```

```

}

/**
 * Returning active Way Invoice
 * @param uuid Way Invoice UUID
 * @return active Way Invoice
 */
private WayInvoice getWayInvoiceById(UUID uuid) {
    returndataManager.load(WayInvoice.class).id(uuid).view("wayInvoice-browse").one();
}

/**
 * Returning active Service Act
 * @param uuid Service Act UUID
 * @return active Service Act
 */
private ServiceAct getServiceActById(UUID uuid) {
    returndataManager.load(ServiceAct.class).id(uuid).view("serviceAct-browse").one();
}

/**
 * Returning main Stat
 * @param uuid Stat UUID
 * @return main Stat
 */
private Stat getStatId(UUID uuid) {
    returndataManager.load(Stat.class).id(uuid).view("_local").one();
}
}

```

Опис методів цього класу міститься у спеціальних блоках документації **javaDoc** програмного коду.

У даному класі задіяне ORM (об'єктно-реляційне відображення) платформи.

Але досить часто доводиться використовувати SQL-запити для взаємодії з БД.

Наприклад так виглядає один із класів **CalculationService** по підрахунку денного балансу:

```

package com.company.sc.service;

```

```

import com.haulmont.cuba.core.Persistence;
import com.haulmont.cuba.core.Query;
import com.haulmont.cuba.core.Transaction;
import org.springframework.stereotype.Service;

```

```

import javax.inject.Inject;
import java.awt.*;
import java.math.BigDecimal;
import java.util.Iterator;
import java.util.List;
import java.util.UUID;

```

```

/**
 * Calculating Day Balance
 */

```

```

@Service(MoneyDayBalanceService.NAME)
public class MoneyDayBalanceServiceBeanimplements MoneyDayBalanceService {

    @Inject
    private Persistence persistence;

    /**
     * Calculating Day Balance
     * @param nameEntity - Cash or Bank
     * @param date Balance Date
     * @return 1. Total at Day Start
     *           2. Profit per day
     *           3. Expense per day
     *           4. Total at Day End
     */
    @Override
    public List GetDayBalance(String nameEntity, String date, UUID firmAccountId) {
        List result;
        StringBuilder sb = new StringBuilder();
        String strQuery = "";
        if (firmAccountId == null) {
            strQuery = sb.append("Select " +
                "Sum(A.total) as total_begin" +
                ",Sum(A.Profit) as Profit" +
                ",Sum(A.Expense) as Expense" +
                ",Sum(A.total) - Sum(A.Profit) + Sum(A.Expense) as total_end " +
                "From " +
                "(Select " +
                "Sum(Case when stat.direction = 'Profit' then cash.sum else (-1
* cash.sum) end) as total" +
                ",0 as Expense" +
                ",0 as Profit " +
                "From ")
                .append(nameEntity)
                .append(" as cash " +
                    "Inner join " +
                    "sc_stat as stat on cash.stat_id = stat.id " +
                    "Where 1=1 " +
                    "and cash.delete_ts is null " +
                    "and cash.date < ")
                .append(date)
                .append(" " +
                    "union all " +
                    "Select " +
                    "0 as total" +
                    ",Sum(Case when stat.direction = 'Profit' then cash.sum else 0 end) as Profit" +
                    ",Sum(Case when stat.direction =
'Expense' then cash.sum else 0 end) as Expense " +
                    "From ")
                .append(nameEntity)
                .append(" as cash " +
                    "Inner join " +
                    "sc_stat as stat on cash.stat_id = stat.id " +
                    "Where 1=1 " +

```

```

        "and cash.delete_ts is null " +
        "and cash.date = ""
    .append(date).append("") as A")
    .toString();
} else {
    strQuery = sb.append
    (
        "Select " +
        "Sum(A.total) as total_begin" +
        ",Sum(A.Profit) as Profit" +
        ",Sum(A.Expense) as Expense" +
        ",Sum(A.total) - Sum(A.Profit) + Sum(A.Expense) as total_end " +
        "From " +
        "(Select " +
        "Sum(Case when stat.direction = 'Profit' then cash.sum else (-1
* cash.sum) end) astotal" +
        ",0 as Expense" +
        ",0 as Profit " +
        "From "
    )
    .append(nameEntity)
    .append
    (
        " as cash " +
        "Inner join " +
        "sc_stat as stat on cash.stat_id = stat.id "
    )
    .append
    (
        "Where 1=1 " +
        "and cash.FIRM_ACCOUNT_ID= ""
    .append(firmAccountId)
    .append(
        "" and cash.delete_ts is null " +
        "and cash.date < ""
    )
    .append(date)
    .append
    (
        "" " +
        "union all " +
        "Select " +
        "0 as total" +
        ",Sum(Case when stat.direction =
'Profit' then cash.sum else 0 end) as Profit" +
        ",Sum(Case when stat.direction =
'Expense' then cash.sum else 0 end) as Expense " +
        "From "
    )
    .append(nameEntity)
    .append
    (
        " as cash " +
        "Inner join " +

```

```

        "sc_stat as stat on cash.stat_id = stat.id " +
        "Where 1=1 " +
        "and cash.FIRM_ACCOUNT_ID= "
    )
    .append(firmAccountId)
    .append(
        " and cash.delete_ts is null " +
        "and cash.date = "
    )
    .append(date).append(") as A")
    .toString();
}

try(Transaction tx = persistence.createTransaction()){
    Query query = persistence.getEntityManager()
        .createNativeQuery(strQuery);

    result = query.getResultList();

    tx.commit();
}
return result;
}
}

```

ПРЕЗЕНТАЦІЯ

Кваліфікаційна робота бакалавра

Розробка програмного забезпечення «Облік товарів магазину»

Виконала: студентка 4 курсу групи КН-17-2 Серветник Ю.В.

Керівник: к.ф.-м.н., доцент кафедри КНІТ Міхалевський В.Ц.

Актуальність

В сучасних умовах зберігати інформацію та швидко керувати нею надзвичайно важливо для невеликих магазинів та фірм.

Один із важливих аспектів оптимізації робочого процесу магазину або фірми є автоматизований облік товарів.



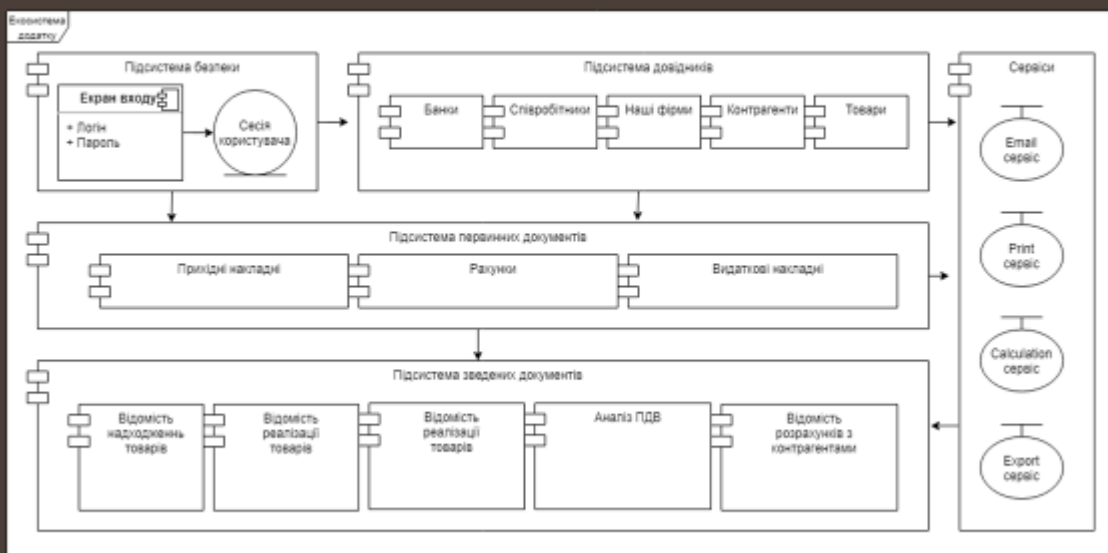
Завдання та мета

Метою створення програмного забезпечення є спростити облік товарів та послуг на фірмі або в магазині та забезпечити доступ до повної інформації про товар, який знаходиться на складі.

Функції, які підлягають автоматизуванню

1. Формування накладних на надходження товарів.
2. Формування накладних на реалізацію товарів.

Діаграма зведеної підсистеми



Сторінка для входу до порталу

...а порталу. Тут Ви знайдете загальнодоступну інформацію. Для переходу в особистий ... меню, кнопка входу до якого знаходиться внизу зліва.

Продаж, гарантія, постгарантійні ремонти та обслуговування комп'ютерної техніки.

Розробка, впровадження, супроводження програмного забезпечення для автоматизації ваших бізнес процесів. Навчання імплементів та адміністративів ПЗ

Новини

- 19 лютого 2021р. - Термінал інтернет-екранів та нова Зв'язок 3G
- 10 лютого 2021р. - Універсальна музична камера з вивантаженням
- 02 лютого 2021р. - Онлайн реєстрація особового кабінету на портал
- 21 січня 2021р. - Послуги з автоматизації вказівки банку
- 25 січня 2021р. - Завантаження на електронне завантаження
- 05 січня 2021р. - Активовано зворотній зв'язок електронною поштою
- 27 грудня 2020р. - Доступна нова тема Facebook

Пошук вікон прихідних та видаткових накладних

Оновлення

- 29 квітня 2021р. - @fardale: Введено пошук товару по штрихкоду
- 18 березня 2021р. - @fardale: Доданий експорт нових ПН
- 17 березня 2021р. - @fardale: проведена миграція на нову версію платформи
- 13 березня 2021р. - @f: Помилка отримання повного графа товару (відсутність підгруп)
- 12 березня 2021р. - @fardale: Введено пошук товару по штрихкоду
- 12 березня 2021р. - @fardale: Додано поле "Штрих код" до довідника товарів
- 11 березня 2021р. - @fardale: проведена миграція на нову версію платформи
- 07 березня 2021р. - @fardale: Введений механізм виключення веб користувачів з розсилки новин
- 05 березня 2021р. - @fardale: Доданий експорт нових ПН
- 04 березня 2021р. - @fardale: додано "код ознаки джерела податкового номера"
- 23 лютого 2021р. - @fardale: Форма "каса: редактор" 30 отримує назву фірми по привязці "головна для 30"
- 22 лютого 2021р. - @fardale: створено блок відправки email про нове замовлення послуги на support@csk.kiev.ua
- 10 лютого 2021р. - @fardale: Створено розсилку новин компанії
- 06 лютого 2021р. - @fardale: Оновлено екран редагування веб користувачів
- 02 лютого 2021р. - @fardale: Створений блок онлайн заявки на реєстрацію нових користувачів
- 28 січня 2021р. - @f: оптимізована поведінка при збереженні змін у екран редагування банку (оф)
- 27 січня 2021р. - @fardale: створено блок відправки email про стан гарантійного ремонту клієнтами
- 27 січня 2021р. - @fardale: доданий звіт "Супровіджувальний лист на гарантійний ремонт"
- 26 січня 2021р. - @fardale: доданий звіт "Квитанція отримання обслуговування на гарантійний ремонт"
- 26 січня 2021р. - @fardale: Доданий екран гарантійного ремонту

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ

Направляється студент Серветник Ю. В. на захист дипломного проекту (роботи)
(прізвище, ініціали)

за спеціальністю 122 - Комп'ютерні науки

На тему: Розробка програмного забезпечення "Облік товарів магазину"

Дипломний проект (робота), рецензія і довідка про перевірку на плагіат додаються.

Декан факультету



САВЕНКО О.С.

(прізвище та ініціали)

ДОВІДКА УСПІШНОСТІ

Серветник Ю. В. за період навчання на факультеті програмування та комп'ютерних і телекомунікаційних систем з 2017 по 2021 роки повністю виконав навчальний план спеціальності з такими розподілом оцінок за:

національною шкалою: відмінно 0,00 %, добре 34,38 %, задовільно 65,62 %.

шкалою ЄКТС: А 1,82 %, В 9,09 %, С 25,45 %, D 16,36 %, E 47,27 %.

Методист факультету

[Signature]

(підпис)

(прізвище та ініціали)

ВИСНОВОК КЕРІВНИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ) ТА ОБГРУНТУВАННЯ ОЦІНКИ

Студент Серветник Ю. В. одягав роботу по виконанню

кваліфікаційної роботи бакалавра вищого рівня. Темою роботи аналіз, розробка моделі та алгоритми оптимізації обліку товарів. Інформаційна система використовує сучасні технології. КРБ виконана згідно методичних вимог. Зазначаю, що виконавцю на роботу програмного забезпечення, це зроблено.

Оцінка дипломного проекту (роботи) добре

Керівник дипломного проекту (роботи)

[Signature]

(підпис)

Михайлик В.С.

(прізвище та ініціали)

" 8 " 06 2021 р.

ВИСНОВОК КАФЕДРИ ПРО ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ)

Дипломний проект (роботу) розглянуто. Студент Серветник Ю. В. допускається до захисту цього

Завідувач кафедри

КНІТ

(назва)

[Signature]

Барман О.В.

(підпис, прізвище, ініціали)

" 9 " 06 2021 р.

09.06.2021

result_1076318629340470415.html

Wed Jun 09 11:49:29 EEST 2021, Петровський Сергій Степанович, Хмельницький національний університет, ХНУ

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%
Словари проверки: en_US, ru_RU, ua_UA. Ошибок в документах: 12%

ID: 92856 Название: Розробка програмного забезпечення «Облік товарів магазину» Добавлено в БД: 2021-06-09 Авторы: Ю.В. Серветник Руководители: В.Ц. Міхалевський Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	36968	387	2707 (7%)	36 (9%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:
Кафедра КН

ID перевірки:
1008245253

Дата перевірки:
09.06.2021 13:32:40 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
09.06.2021 13:36:12 EEST

ID користувача:
100005671

Назва документа: 1623233869317831 New

Кількість сторінок: 53 Кількість слів: 5528 Кількість символів: 41572 Розмір файлу: 3.96 MB ID файлу: 1008317609

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

13.5%
Схожість

Найбільша схожість: 5.25% з Інтернет-джерелом (<http://um.co.ua/8/8-2/8-221997.html>)

11.3% Джерела з Інтернету

100

Сторінка 55

4.2% Джерела з Бібліотеки

57

Сторінка 55

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

26

Підозріле форматування

22
сторінки

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: Розробка програмного забезпечення «Облік товарів магазину»

Автор: Серветник Ю.В., студентка групи КН-17-2

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.фіз.-мат.н., доцент кафедри КНІТ Міхалевський В.Ц.

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої та електронної версії роботи	-
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	-
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	-

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

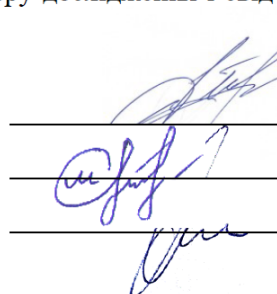
- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні;
- 3) до запозичень входять фрагменти програмного коду, що на мають авторства і містять поширені конструкції;
- 4) серед запозичень знаходяться загальновідомі терміни, скорочення та визначення.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 13,5% і адресується до першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КНІТ



В. Ц. Міхалевський

О. В. Мазурець

О. В. Бармак

Хмельницький національний університет
Кафедра комп'ютерних наук та інформаційних технологій

РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студентки групи КН-17-2 Серветник Юлії Віталіївни

за темою Розробка програмного забезпечення «Облік товарів магазину».

Актуальність і значення теми: програма дозволить оптимізувати час та спростити облік товарів в магазині, забезпечить доступ до потрібної інформації про товар.

1. Оцінка запропонованих моделей, підходів, алгоритмів, інформаційної складової та засобів розробки: моделі та алгоритми оптимізації обліку товарів використані доцільно, тому, що такий облік дозволяє швидко отримати доступ до потрібної інформації

2. Оцінка розробленої інформаційної системи, її практична цінність та економічна доцільність: розроблене програмне забезпечення «Облік товарів магазину» використовується для автоматизації контролю руху товарів на складі магазину, оптимізації обліку, та доступ до потрібної інформації про товар

3. Загальний висновок та оцінка: вимоги поставленої задачі виконані в повному обсязі, розроблена програма «Облік товарів магазину», працює вірно.

Робота заслуговує на оцінку « добре »

Рецензент В. Г. Н., доцент каф. ТМІТ ДНУ Олександр О. Р.