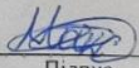




КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

на тему Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

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: студент 4 курсу, група КН-18-1  М.В. Запорожець
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: викладач кафедри КН  М.О. Молчанова
Науковий ступінь, посада Підпис Ініціали, прізвище

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

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор

13 червня 2022 р.

 О.В. Бармак
Підпис Ініціали, прізвище

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет інформаційних технологій

Кафедра комп'ютерних наук

Освітній ступінь бакалавр

Галузь знань 12 – Інформаційні технології

Спеціальність 122 – Комп'ютерні науки

ЗАТВЕРДЖУЮ
Завідувач кафедри комп'ютерних наук

(підпис)

д.т.н., професор О.В. Бармак

« 25 » березня 2022 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА**

1. Тема кваліфікаційної роботи бакалавра: «Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві»

2. Завдання видано студентці Запорожець Максимові Вадимовичу
(прізвище, ім'я, по батькові)

3. Керівник роботи викладач кафедри КН Молчанова Марина Олексіївна
(посада, прізвище, ім'я, по батькові)

4. Затверджено наказом університету від « 01 » березня 2022 р. № 18

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

Мета роботи – є розробка та практична реалізація на платформі PHP інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Потрібно забезпечити можливість працювати з багатьма видами документів для ведення складу та іншої інформації, мати можливість додавати, видаляти та редагувати інформацію.

Виконавець: студент 4 курсу, група КН-18-1
Курс, група виконавця

[Підпис]
Підпис

М.В. Запорожець
Ініціали, прізвище

Керівник: викладач кафедри КН
Науковий ступінь, посада

[Підпис]
Підпис

М.О. Молчанова
Ініціали, прізвище

Анотація

Тема кваліфікаційної роботи бакалавра: «Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-18-1 Запорожець Максим Вадимович

Керівник кваліфікаційної роботи бакалавра: викладач кафедри КН Молчанова Марина Олексіївна

Кваліфікаційна робота бакалавра містить:

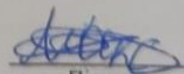
Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
72	52	11	38	4

Метою кваліфікаційної роботи бакалавра є розробка інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Для розробки інформаційної системи було використано мову програмування PHP, а також систему керування базами даних MySQL.

Розроблена система призначена для людей, яким потрібно працювати з різними видами документів на виробництві. Реалізована автоматизація прийняття рішень з контролю складських запасів матеріалів на виробництві дозволяє підвищити ефективність відповідної діяльності як людям в яких вже є досвід роботи з документами, так і тим хто тільки починає з ними працювати.

Ключові слова: накладна, документ, обліковий запис, інформаційна система, команда, виробництво.

Виконавець: студент 4 курсу, група КН-18-1
Курс, група виконавця


Підпис

М.В. Запорожець
Ініціали, прізвище

Зміст

Перелік скорочень	5
Вступ.....	6
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій.....	8
1.1 Аналіз інформаційних моделей.....	8
1.2 Огляд теоретичних підходів до розв’язку подібних задач	16
1.3 Аналіз існуючих програмних рішень.....	19
1.4 Аналіз сучасних засобів створення програмного забезпечення	25
1.5 Мета, задачі та вимоги до реалізації інформаційної системи	27
Розділ 2 Проектування інформаційної системи	29
2.1 Аналіз та автоматизація обробки потоків даних	29
2.1.1 Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві	29
2.1.2 Функціональна структура інформаційної системи	31
2.2 Інформаційна структура системи	37
2.2.1 Проектна архітектура системи та взаємозв’язок компонентів.....	37
2.2.2 Інформаційна модель.....	39
2.3 Вибір засобів розробки інформаційної системи	44
2.3.1 Вибір мови програмування	44
2.3.2 Вибір фреймворку	46
2.3.3 Вибір редактора програмного коду.....	48
2.3.4 Вибір СКБД	49
Розділ 3 Програмна реалізація інформаційної системи	51
3.1 Структура та функціональне призначення програмних складових системи.....	51
3.2 Особливості реалізації програмних складових системи.....	53
3.3 Тестування інформаційної системи	63
3.4 Інструкція користувача.....	67
3.5 Вимоги до розгортання інформаційної системи.....	76
Висновки	77
Перелік посилань.....	78
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
БД	База даних
ІС	Інформаційна система
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ПЗ	Пояснювальна записка
ПП	Програмний продукт
СКБД	Система керування базами даних
ХНУ	Хмельницький національний університет.
BCL	Base Class Library
FCL	Framework Class Library
MS	Microsoft

Вступ

На сьогоднішній день в епоху постійного технологічного прогресу у всіх сферах нашого життя є чітке розуміння, що світ не стоїть на одному місці, а постійно розвивається та прогресує у дуже шаленому темпі. Є унікальна можливість спостерігати з легкістю як одна технологія змінює іншу, спрощуючи та покращуючи ті чи інші сфери життєдіяльності людства. Для прикладу взяти найпростіше – телефонний зв'язок. В наш час люди не можуть собі уявити життя без мобільних телефонів, хоча, приблизно, 40 років назад їх ще навіть не існувало. За ці роки телефон пережив цілу еволюцію від великого апарату з антеною та кнопками до сучасних сенсорних моделей з постійним вдосконаленням. Завдяки цьому постійному прогресу з'явилося таке розуміння як “Інформаційні технології”.

Інформаційні технології – це певні методи які використовуються для того, щоб була можливість проводити певні дії з інформацією. Простими словами – це все, що пов'язано зі збором, обробкою, аналізом, передачею, зберіганням та поширенням даних [1].

Кожна сфера, яка на сьогодні існує використовує інформаційні технології, які відіграють важливу стратегічну роль у її розвитку. Завдяки їм у багатьох підприємств з'являється можливість для вдосконалення в управлінні та спрощення контролю обліку за матеріальними цінностями. Іншими словами, підприємству, яке виготовляє ту чи іншу продукцію потрібно вести облік товарно-матеріальних цінностей, які забезпечують безперебійну діяльність цього підприємства. А саме: складський облік матеріалів та комплектуючих для виробництва, складський облік та контроль реалізації готової продукції, взаєморозрахунки з постачальниками та клієнтами, тощо... Всі ці процеси повинні підтверджуватись відповідними документами, такими як: рахунок-фактура, товаротранспортна накладна, накладна-вимога, картка складського обліку, лімітно-забірна картка, різні відомості та журнали, шляховий лист,

тощо... Оформлення та ведення всіх цих документів потребує багато часу та ресурсів.

Іноді, завдяки такому поняттю як «людський фактор», інформація може бути не достовірною та містити багато помилок. А відсутність достовірних даних по залишкам матеріальних цінностей на складах може призвести до можливості у співробітників для їх розкрадання або до помилок у аналізі та плануванні виробництва. Але завдяки інформаційним технологіям є можливість впорядкувати, удосконалити та автоматизувати всі ці процеси, що забезпечить легкий доступ до будь-якого документа, безпеку та достовірність даних про наявність матеріальних цінностей підприємства.

Також хочеться відмітити ще один момент. Більшість підприємств малого та середнього бізнесу у нашій країні мають свій початок з так званих лихих 90-х часів, тобто часів перестройки. Спочатку підприємці, яких на той час називали «торгашами» привозили товари з-за кордону, а поступово по ходу зростання капіталу, почали відкривати власні виробництва цих товарів. Відповідно і почали виникати питання щодо управління підприємством та контролю над запасами на складах та над фінансовими потоками, бо більшість з цих підприємців, за браком часу на навчання, не володіють відповідними знаннями. Ось тут теж на допомогу приходять інформаційні технології. З'являється можливість розробити програмне забезпечення, за допомогою якого можна повністю вести та контролювати виробництво і використання якого не потребує якихось супер навичок та знань. В цьому і полягає основна мета інформаційних технологій – зробити так, щоб людині було якомога простіше користуватись різними сферами життєдіяльності.

Мета полягає у розробці та практичній реалізації інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Об'єкт – процес контролю запасів матеріалів на виробництві

Предмет – моделі, методи, для технології для прийняття рішень з контролю запасів матеріалів на виробництві

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізацій

1.1 Аналіз інформаційних моделей

Задачі прийняття рішень постійно виникають і розв'язуються в природі, у світі що нас оточує – в біологічних, екологічних, соціальних і економічних системах, різноманітних процесах та явищах, наприклад, у процесах функціонування живих організмів та їх колоній, проявах споживчих уподобань, природних катаклізмах тощо. Рішенням вважається обґрунтований набір дій з боку особи, що приймає рішення (ОПР), спрямованих на об'єкт чи систему управління, який надає можливість привести даний об'єкт чи систему до бажаного стану або досягнути поставленої мети. Рішення є одним із видів розумової діяльності і проявом волі людини. Характерними ознаками рішення є: можливість вибору з набору альтернативних варіантів; за відсутності альтернатив, відсутній і вибір, отже, відсутнє й рішення; наявність мети: безцільний вибір не розглядається як рішення; необхідність вольового акту ОПР при виборі рішення, тому що вона формує рішення при боротьбі мотивів і думок [2].

Прийняття рішення – це процес вибору найбільш преференційного рішення з множини допустимих рішень або упорядкування множини рішень. Прийняття рішень можливе на підставі знань про об'єкт управління, процеси, що в ньому відбуваються і можуть відбутися з перебігом часу, а також за наявності множини показників, що характеризують ефективність та якість прийнятого рішення. Тобто необхідні адекватна модель об'єкту і модель прийняття та оцінювання прийнятого рішення. Під моделлю прийняття рішень мається на увазі формальне подання поставленої задачі та процесу прийняття рішень [3].

Будь-який процес прийняття рішень проходить в декілька етапів. А саме:

– *Етап постановки задачі* – це той етап, який передбачає виявлення та опис певної проблемної ситуації, збір інформації та визначаються цілі рішення, яке має бути прийняте, що дає змогу напряму задати пошук рішення [4].

– *Етап формування рішень* – це той етап, де відбувається оцінка обмежень да надає змогу відокремити прийнятні варіанти від неприйнятних. Після цього відбувається альтернативних варіантів рішення [5].

– *Етап вибору рішення* – це той етап, де відбувається оцінка варіантів з множини допустимих альтернатив за обраними критеріями та подальший остаточний вибір рішення [6].

Суттєвий вплив на розв’язок задач прийняття рішень спричиняють умови та середовище, в яких відбувається прийняття рішень. **Визначеність** - рішення приймається в умовах визначеності, коли точно відомий результат кожного з альтернативних варіантів вибору. Відносно небагато рішень при управлінні бізнес-процесами приймаються в умовах визначеності. **Ризик** - до рішень, що приймаються в умовах ризику, відносяться такі, при формуванні яких результати альтернативних варіантів не є визначеними, але відомі їх імовірності. Сума імовірностей всіх результатів певної альтернативи повинна бути рівною одиниці [7].

Серед сучасних напрямів розробки людино-машинних систем - системи автоматичного керування, експертні системи та системи підтримки прийняття рішень. Найбільш придатними для розв’язання багатьох задач, зокрема задачі розподілу ресурсів, виявляються **СППР**. Саме за допомогою СППР ОПР має можливість безпосередньо за допомогою обчислювальних засобі проектувати, порівнювати та обирати альтернативні варіанти рішень у самі різноманітні способи. СППР інтегрують в собі такі якості, які роблять їх не тільки дуже корисними для системних задач управління і прийняття рішень, але й по суті незамінними інструментами аналізу даних в сучасних умовах економічного розвитку. На рівні користувача виділяють такі види СППР: **активна** – може зробити пропозицію, яке рішення варто вибрати; **пасивна** – допомагає у процесі ухвалення рішення, але не може ввести пропозицію, яке рішення прийняти; **кооперативні** – дозволяє ЛПР змінювати, поповнювати або поліпшувати рішення, пропоновані системою, посиляючи потім ці зміни в систему для перевірки [8].

В економічній теорії термін «виробництво» означає економічний процес, у якому виробник створює товари споживання матеріальних та суспільних благ. Товари можуть мати як матеріальну ознаку (житло, продукти харчування, одяг, побутова хімія, тощо...) так і не матеріальну, так як об'єктом виробництва може бути послуга (послуги: таксі, перукарні, готелю, тощо...) [9]. Якщо говорити простими словами, то в основному це процес створення якогось продукту на підприємстві, призначеного для продажу. Учасниками виробничого процесу є підприємства, які виробляють товари або послуги і покупці або споживачі, які їх отримують. Але у сучасному суспільстві не завжди торгівля іде напряму по схемі виробник – покупець. Найчастіше є посередники. Посередники зазвичай самостійно товари не виготовляють, але пропонують послуги як для виробника так і для покупця, створюючи комфорт для обох сторін. Наведемо простий приклад: танцювальне взуття. Є взуттєва фабрика, яка його виготовляє. Є покупець, танцюрист, якому потрібно придбати взуття для конкурсу. За однією парою взуття він іде у спеціалізований магазин, а не на фабрику. Магазин – є посередником між фабрикою та танцюристом. За часту виробництва надають перевагу у співпраці з посередниками, тому, що вони закупають великі партії товарів за раз. Для заохочення посередникам надаються різні умови: оптові ціни та знижки на продукцію, а також можливість отримувати товари під реалізацію. Іноді все ж таки можлива торгівля напряму з фабрики. При цьому на підприємстві створюється окремий підрозділ для роздрібною торгівлі. Це може бути або магазин, розташований біля виробництва чи на його території, або інтернет-магазин.

У більш глобальному значенні виробництво – один з факторів, які піднімають економіку країни, за рахунок того, що: на виробничих підприємствах створюються робочі місця, що призводить до зменшення безробіття в країні - люди багатіють, отримуючи зарплатню, і починають купувати товари - на товари починає зростати попит - товари починають виготовляти у більшій кількості, що призводить до розширення виробництва, створення нових підприємств і робочих місць - збільшуються прибутки підприємств - за рахунок сплати податків

підприємств збільшуються надходження до державного бюджету країни. Тому добробут нашої країни напряму залежить від господарської діяльності виробничо-комерційних підприємств.

Підприємство – це певний самостійний об’єкт, який зареєстрований компетентним органом державної влади з метою задоволення суспільних та особистих потреб [10]. Будь-яка діяльність, пов’язана з виробництвом, обміном та споживанням матеріальних і не матеріальних цінностей, наданням та використанням послуг, є господарською діяльністю підприємства країни. (Книжка) Для здійснення господарської діяльності кожне підприємство має у своєму розпорядженні необхідні господарські засоби (будівлі, транспорт обладнання, інструменти, виробничі запаси, грошові кошти...) та джерела цих засобів. Також поряд з господарськими засобами необхідною умовою діяльності підприємства є праця.

Праця – це діяльність людини, яка спрямована для того, щоб досягнути мети – виробництва певного продукту. Предмет праці – це певні об’єкти, на які діє людина. Для прикладу це може бути земля, певні матеріали і так далі. Коли людина діє на предмет праці, вона використовує для цього засоби праці. Засоби праці та предмет праці є складовими засобами виробництва [11].

Діяльність підприємства складаються із певних процесів, які поділяються на операції:

- процес заготівлі – операції із закупівлі матеріалів та комплектуючих для виробництва
- процес виробництва – операції з виготовлення виробів із сировини та інших виробничих запасів
- процес реалізації – операції з продажу вироблених виробів.

Запаси – активи підприємства, які утримуються для подальшого продажу, перебувають у процесі виробництва з метою подальшого продажу продукту виробництва або утримуються для споживання під час виробництва продукції, виконання робіт та надання послуг, а також управління підприємством [12].

Для правильної організації обліку запасів їх відповідно групують (класифікують) за їх призначенням та роллю в процесі виробництва. Формується **номенклатура запасів** – перелік запасів, згрупованих за характерною для них ознакою і відповідним чином зашифрованих із зазначенням одиниці виміру [13].

Тобто запаси поділяються на групи (наприклад взуття, одяг, аксесуари тощо). Кожна група поділяється на підгрупи (наприклад група взуття на чоловіче взуття, жіноче взуття, дитяче взуття тощо). У межах кожної групи запаси групуються за моделлю, матеріалом, кольором, розміром та ін. Також місцем зберігання та обліку виробничих запасів на підприємстві є склади.

Склад – це певна обладнана нерухомість, яка використовуються для зберігання запасів підприємства. Для прикладу це можуть бути матеріали, які необхідні для виробництва, або вже готова продукція, яку потім реалізують. Для спрощення обліку та контролю на підприємстві розрізняють склади за призначенням [14]. Наприклад:

- склад матеріалів та комплектації – зберігаються необхідні матеріали та комплектуючі вироби для виробництва потрібного продукту.

- склад заготовок – зберігаються напівфабрикати (незавершене виробництво).

- склад готової продукції – зберігається готова продукція.

Переміщення та облік товарів на складах здійснюється за допомогою відповідних документів – накладних.

Накладна – обліково-фінансовий документ (первинний документ) про приймання, видачу чи відправлення матеріальних цінностей [15]. Або якщо простішими словами – це документ, який використовується для передачі товару від однієї особи до іншої. Існує багато різновидів накладних.

Наприклад:

- товарна накладна або рахунок-фактура – виписується тоді, коли товар або приходить або відпускається зі складу.

- товаро-транспортна накладна – виписується тоді, коли товар необхідно транспортувати. (іноді, коли перевізника по дорозі можуть зупинити інспектора

патрульної служби – цей документ являється доказом того, що товар не вкрадено).

- Інвойс – використовується при міжнародній торгівлі
- лімітно-заборна накладна – виписується в момент запуску виробництва продукції для отримання необхідних матеріалів і комплектуючих згідно вказаних кількостей.

Бланки накладних можуть мати різний вигляд і містити різну інформацію. Накладна містить в собі наступні дані:

- назва та розрахункові реквізити установи яка видає накладну (наприклад ПП «КМ-Поділля», р/р UA7931454050000026000052319209 МФО 315405, код в ЄДРПО 36653351)
- назва виду накладної, номер та дату здійснення операції (наприклад Видаткова накладна №10 від 10 січня 2022р.)
- назва установи або клієнта, який отримує накладну (наприклад Яновська Олена Вікторівна)
- за допомогою трафаретного заповнення, заповнюються всі графи накладної, перераховується вся продукція яка поступає або відпускається, вказуються одиниці виміру, кількість, ціна за одиницю, сума за кількість одиниць, сума за весь товар прописом і цифрами (наприклад: 1. Чоловіче взуття – пара – 2 – 750,00 – 1500,00 Всього до сплати: Одна тисяча п’ятсот гривень 00 копійок)
- ставляться підписи матеріально-відповідальних осіб які відвантажують і які отримують товар, а також при наявності печатки.

На підприємстві ведеться відповідно складський облік запасів. Раніше, такий облік здійснювали за допомогою карток складського обліку. Вони заводились окремо на кожну номенклатуру продукції або матеріалу, де вписувались всі дані. Завідуючий складом щоденно мав записувати у відповідну картку всі операції надходження або витрати матеріалів, нумеруючи свої записи по порядку. Після кожного запису визначався залишок. Потім дані подавалися в бухгалтерію, де з них формували відповідні звіти. Зараз виробничі підприємства

вже не можуть працювати по старим стандартам – за допомогою паперових документів, ручного керування виробничими процесами та планування по графікам на міліметрівці. Враховуючи те, світ існує в епоху цифрових технологій та автоматизації, господарська діяльність кожного підприємства вимагає ретельного планування, чіткого управління та швидкого контролю, для вчасного реагування на зміни. Тому вона базується на найбільш раціональному використанні матеріальних благ і грошових ресурсів. А цього можливо досягнути тільки завдяки використанню інформаційних технологій, які зараз дуже стрімко розвиваються.

Інформаційні технології (ІТ) – це сукупна система методів, процесів та способів використання систем зв'язку і обчислювальної техніки для створення, збору, передачі, пошуку, оброблення та поширення інформації з метою ефективної організації діяльності людей [16].

Стрімкий розвиток інформаційних технологій, особливо за останній час, обумовлений не тільки досягненнями в галузі науки і техніки, а ще й збільшенням потреб сучасного ринку, які необхідно задовольнити. Попит на інформацію, інформаційні послуги веде до того, що сучасні ІТ орієнтовані на широке застосування різноманітних технічних засобів, зокрема, засобів комунікації. В результаті розробляються обчислювальні системи та мережі, мета яких не тільки накопичення, зберігання, обробка інформації, але й максимальне наближення термінальних пристроїв до робочого місця спеціаліста.

Наприклад, буквально 20 років назад, щоб підприємству розрахуватись з постачальником за певний матеріал потрібно було пройти наступне: бухгалтер на машинці друкував платіжне доручення через копірку в трьох екземплярах → доручення потрібно було підписати у керівництва і поставити печатку → тоді бухгалтер відносив платіж в банк. Якщо бухгалтер допустив хоч мінімальну помилку при друкуванні платежі (лишню кому поставив, чи букву пропустив), банк не міг провести дану операцію. І приходилося вертатись назад на підприємство і повторювати всі операції спочатку. Втрачалось багато часу. Зараз, завдяки комп'ютерній техніці, програмному забезпеченню та Інтернет

технологіям бухгалтер витрачає на ту саму операцію лише пару хвилин. Завдяки електронним підписам, ключам не має необхідності лишній раз турбувати дрібними питаннями керівника підприємства.

Отже, таким чином, спостерігаємо, що основними перевагами застосування інформаційних технологій на виробництві є:

- фіксування бухгалтерської інформації за єдиними принципами та вимогами, а також можливість синхронізації з іншими системами, що дозволяє сформувати єдину інформаційно-комунікативну систему підприємства;

- можливість здійснення налаштувань відповідно до запиту користувачів і надання інформації у необхідному для них вигляді, що впливає на оперативність прийняття управлінських рішень, тайм менеджмент, мінімізацію інформаційного шуму;

- взаємозв'язок бухгалтерського обліку й економічного аналізу як єдиної інформаційної бази, спрямованої на забезпечення вимог один одного, можливість налаштування інформаційних каналів щодо кожного елементу [17].

Програма автоматизації в частині аналізу запасів підприємства має відповідати ряду завдань: планування у довгостроковому та короткостроковому терміні на основі визначених показників, надання оперативної інформації відповідно до запиту, можливість вносити зміни, враховуючи фактори впливу, що у сукупності дозволить сформувати єдину концепцію управління. Зокрема, враховуються напрями мінімізації використання матеріальних ресурсів підприємства, а саме лише на основі сформованих планів, однак із можливістю коригування відповідно до наявних і потенційних змін, факторів внутрішнього та зовнішнього середовища, облік обмеження залучення ресурсів у зазначеному бізнес-процесі, формування логістичної стратегії та механізму взаємодії відповідних суб'єктів [18].

Автоматизація передбачає першочергове визначення актуальних запитів і напрямів щодо кожного об'єкта. Як зазначає К. Багрій, параметрами системи управління запасами є:

- точка замовлення – мінімальний (контрольний) рівень запасів продукції, за умови досягнення якого необхідне їх поповнення;
- нормативний рівень запасів – розрахункова величина запасів, яка досягається під час чергової закупівлі;
- обсяг окремої закупівлі;
- частота здійснення закупок – тривалість інтервалу між двома можливими закупками продукції, тобто періодичність поповнення запасів продукції;
- кількість продукції, що поповнюється та за рахунок якої досягається мінімум витрат на зберігання запасу згідно із заданими витратами на поповнення і заданими альтернативними витратами інвестованого капіталу [19].

1.2 Огляд теоретичних підходів до розв'язку подібних задач

На сьогоднішній день існує безліч мов програмування, які можна використати для розробки програмного продукту. Для прикладу це C#, PHP, JAVA, C++ і так далі. Кожна з мов програмування має свої особливості, свій синтаксис. Але є одна проблема, яка торкається кожен мову і може завадити при колективній розробці певного проекту – це структура коду. Дуже часто для реалізації завдань, програмісти пишуть дуже багато коду для тих чи інших завдань. І чим далі вони рухаються, тим більше коду самого стає. А далі, коли підключається до програміста інший колега, то він починає продивлятися у те, що там написано щоб далі продовжити писати. І якраз через те, що коду багато, дуже легко загубитись і поки він розбереться що там написано, пройде дуже багато часу. Якщо коду біля 200 строк, то це одне, а коли коду більше 1000 строк, то це стає проблемою. Також слід враховувати і те, що програміст під час програмування використовує багато методів та технологій, в залежності від складності програмного продукту. Без певних правил, розібратись у коді буде складно. І це не говорячи про те, якщо виникне якась помилка і потрібно буде її

знайти. Щоб не було цих проблем та потенційно інших, було створено такий патерн як “MVC”.

Патерн – це конструкція архітектурна, яка вирішує проблеми проектування в рамках деякого контексту, що часто виникає. *MVC (Model-View-Controller)* – це фундаментальний патерн, який знайшов застосування у багатьох технологіях, дав розвиток новим технологіям і щодня полегшує життя розробникам. *Модель* – зазвичай розуміється частина, що містить у собі функціональну бізнес-логіку програми. Модель має бути повністю незалежною від інших елементів продукту. Модельний шар нічого не повинен знати про елементи дизайну, і як він відобразатиметься. *View* – його головне завдання полягає у тому, щоб відображати ті дані, які приходять від моделі [20].

Найпоширеніші види патерну MVC:

– MVVM (Model-View-View Model) – цей патерн рахується одним із найкращим патерном через те, що він дає змогу приділити основну увагу інтерфейсу або логіки програми. При цьому не потрібно прописувати обробник подій або роботу Controller як в MVC [21].

– MVP (Model-View-Presenter) – це патерн, який використовується для графічного інтерфейсу. Найчастіше цей патерн використовують саме для розробки мобільних додатків. Його головна відмінність від MVC полягає у тому, що в MVC оновлена модель сама каже View, що потрібно показати інші дані. А якщо цього не відбувається і додатку потрібен посередник як Present, то це патерн MVP [22].

Також при розробці програмного продукту є можливість зустрітися з проблемою при роботі з даними. Проблема у тому, що потрібно десь їх зберігати і при цьому мати можливість працювати з ними. Тому для цього створили базу даних.

База даних (скорочено БД) – це структура певна, яка дозволяє зберігати та працювати зі всіма даними, які там є. БД використовують практично во всіх програмних продуктах. Для прикладу це інтернет-магазин, портал, сайт і так далі. БД допомагає швидко використовувати ту інформацію, яка знаходиться на

БД а також швидкість. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках – це допомагає працювати з інформацією простіше і швидше. Щоб була можливість створити запит до БД для роботи з інформацією використовують SQL. *SQL (Structured Query Language)* дає змогу можливість додавати, редагувати або видаляти інформацію, що міститься у таблицях. Приклад роботи запиту на рисунку 1.1 СКБД (Система керування базами даних) – це певний набір програмних засобів призначених для створення та спільного використання БД двома і більше користувачами [23].

про інформації в БД.

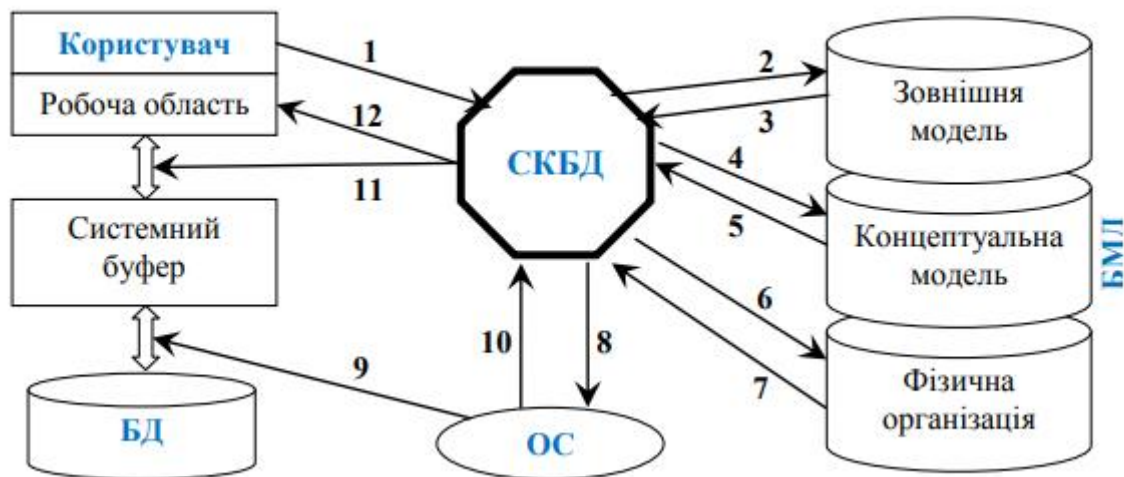


Рисунок 1.1 – Схема проходження запиту до БД [23]

Отже, головним висновком є те, що MVC є дуже важливою складовою у будь-якому серйозному проєкті. Він допомагає набагато якісніше писати програмний код та робити його доступним для інших програмістів, які будуть далі з ним працювати. Це стосується і БД. Вона відіграє важливу роль в розробці продукту. Взагалі вважається, що перед тим як розробляти буде-який продукт, потрібно чітко придумати базу даних з якою в подальшому буде робота.

1.3 Аналіз існуючих програмних рішень

Інформаційні технології допомогли автоматизувати ведення облікового контролю складських запасів матеріалів на виробництві. Якщо раніше для цього потрібно було від руки це все писати на папіру, то тепер це все можна робити за допомогою додатку, який взяв на себе вирішення більшості проблем і користувачу остається лише правильно водити інформація та працювати з нею. А за допомогою хмарних технологій користувач отримав змогу користуватись таким додатком з будь-якої точки світу. Єдина умова для цього – це наявність інтернету. І завдяки цьому, на сьогоднішній день є немало інтернет ресурсів, які надають таку можливість.

Прикладом такого ресурсу є “Dilovod”, головна сторінка якого зображена на рисунку 1.2.

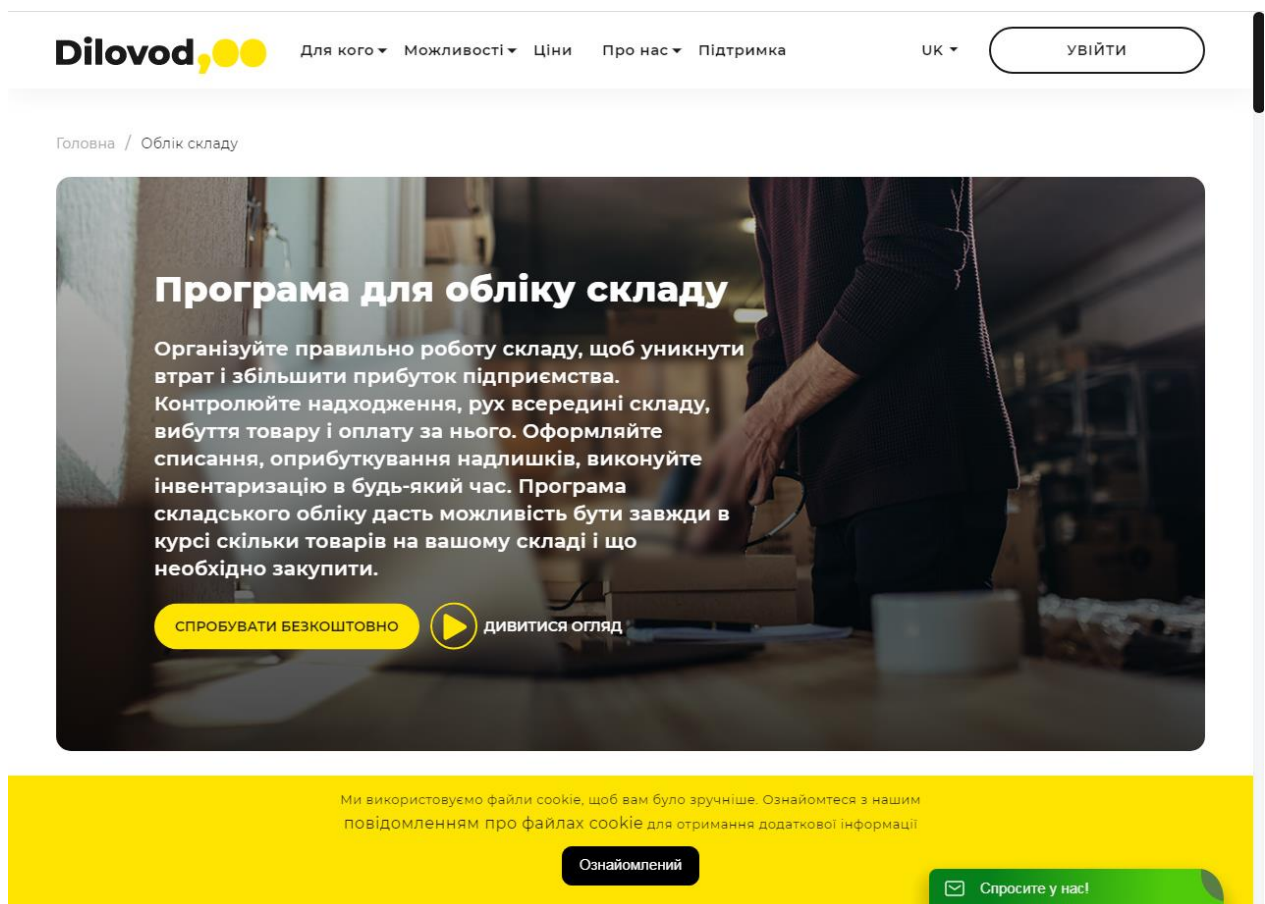


Рисунок 1.2 – Головна сторінка ресурсу “Dilovod” [24]

Dilovod – це інтернет ресурс, який надає можливість вести облік складу. Його головна перевага є у тому, що він надає дуже багато можливостей для ведення різного типу документації. Для прикладу це: облік доходів та витрат, облік самого складу і так далі [24]. Також перевагою є те, що ресурс є зручним у використанні та дуже легко розібратись, де що знаходиться. Також ще одна перевага – це зворотній зв'язок. Тобто, якщо у користувача виникли питання, то користувач має можливість звернутись до служби підтримки та задати питання, яке з'явилося або описати проблем з якою зіткнувся. Приклад служби підтримки зображено на рисунку 1.3

Головна / Підтримка

Служба підтримки

Фахівці служби підтримки Dilovod оперативно відреагують на ваш запит і допоможуть перенести дані з інших програм, виконати необхідні налаштування, завантажити початкові залишки. Проведуть демонстрацію функціоналу, проконсультують з усіх питань в ході роботи.

Підтримка не обмежена і безкоштовна для всіх користувачів Dilovod.

Зв'яжіться з нами!

У робочі дні з 9:00 до 19:00

Vodafone, Telegram	+38 (099) 174 66 11
Київстар, Viber	+38 (096) 174 66 11
Life :)	+38 (063) 174 66 11
Skype	dilovod.ua
Email	support@dilovod.com.ua




Рисунок 1.3 – Сторінка “Служба підтримки” [24]

З незначних недоліків можна відзначити те, що дуже багато функціонала платна. Приклад зображений на рисунку 1.4

Пакети послуг та ціни

Спробуйте будь-який пакет послуг протягом 7 днів. Банківська картка не потрібна.

Виберіть умови

для ФОП

для юр. осіб

1 місяць

6 місяців -5%

9 місяців -10%

12 місяців -15%

ПРОФІ

для одного ФОП без працівників на ЄП або загальній системі

Для 1-го користувача

299 грн/міс.

СПРОБУВАТИ

Включає:

Всі базові можливості + ПРРО

НАЙПОПУЛЯРНИШИЙ

ПРОФІ +

для одного ФОП або юр.особи на ЄП або загальній системі

Для 1-го користувача

399 грн/міс.

СПРОБУВАТИ

Всі можливості ПРОФІ

- + Звітність для юр.осіб
- + Облік ПДВ та реєстрація ПН
- + Бухгалтерські операції
- + Зарплата і управління персоналом
- + Облік роботи менеджерів
- + Валютний облік
- + Розрахунки з підзвітними особами

БІЗНЕС

для одного чи декількох ФОП або юр.осіб на ЄП

Для 3-х користувачів

699 грн/міс.

СПРОБУВАТИ

Всі можливості ПРОФІ +

- + Управління правами користувачів
- + Кілька юр.осіб та ФОП в одній базі
- + API

* За винятком звітності для юр.осіб на загальній системі

Спробуйте у нас!

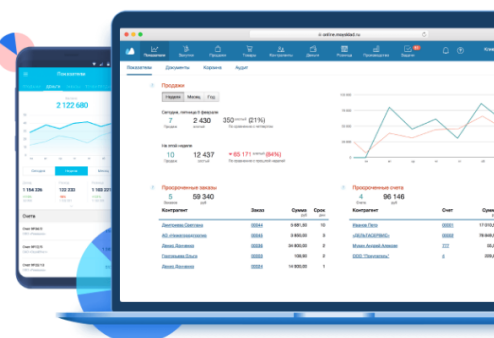
Рисунок 1.4 – Сторінка “Пакет послуг та ціни” [24]

Далі переходимо до наступного ресурсу під назвою “Мій склад”. Мій склад – це ресурс за допомогою якого користувач має змогу вести облік склад а також призначений для ведення бізнесу [25]. Головна сторінка зображена на рисунку 1.5

Торгівля, склад, CRM у хмарі

Все що потрібно — в одній системі: продаж, закупівля, склад, фінанси, клієнти та постачальники

[Почати роботу](#)



Кому вигідно

-
-
-
-


[Підтримка](#)

Рисунок 1.5 – Головна сторінка ресурсу “Мій склад” [25]

Найперша перевага – це гарно оформлений інтерфейс. Всі кнопки видно і нема проблеми в тому, щоб знайти щось. Також ще одна перевага у тому, що ресурс надає безліч інструментів для ведення облікового запису зображено на рисунку 1.6

Виберіть тип вашого бізнесу

Різниця
Онлайн-торгівля
Опт
Виробництво
Кафе



Продажі під контролем
Продавайте зі складу та на замовлення. Публікуйте прайслисти для онлайн-замовлення. Рахунки, взаєморозрахунки та договори, імпорт виписок з будь-якого банку.

На складі завжди порядок
Плануйте стан складу, закупівлі та відвантаження. Автоматичні замовлення постачальникам. Інвентаризація прямо зі смартфона. Адресне зберігання.

Відділ продажів - в офісі або віддалено
Організуйте роботу менеджерів. Ставте завдання, працюйте із базою клієнтів. Дивіться аналітику за замовленнями та клієнтами, підключіть телефонію та розсилки.

[Дізнатися більше](#)

Рисунок 1.6 – Сторінка для ведення облікового запису [25]

Усі можливості управління бізнесом

Без складного та дорогого впровадження

Складський облік

Приймання, відвантаження, інвентаризації. Робота з розмірами, кольорами, упаковками, серійними номерами, ваговим товаром. Друк документів та аналітика.

Обробка замовлень

Отримуйте повідомлення про нові замовлення, призначайте менеджерів, змінюйте статуси, створюйте відвантаження. Легко звіряйте комплектацію при збиранні замовлень на складі.

Роздрібні продажі

Для касира чи кур'єра, роздрібу, пункту видачі, розносною торгівлі – безкоштовна касова програма під Windows, Linux, Android, iOS.

Складання та виробництво

Оформляйте операції зі складання та розбирання, контролюйте рух комплектуючих, вражайте собівартість та плануйте потреби в матеріалах.

Робота з клієнтами

Зберіть у картці все про клієнта: статистику, документи, завдання, зустрічі. Телефонуйте та робіть розсилки по SMS та email. Сегментуйте базу, будуйте вирви продажів.

Управління фінансами

Контролюйте взаєморозрахунки із клієнтами. Отримуйте виписки з будь-якого банку, надсилайте та друкуйте платіжні доручення.

[Чат](#)

Рисунок 1.7 – Сторінка з інформацією, які можливості надає ресурс [25]

Також потрібно відзначити те, що ресурс надає можливість не тільки вести обліковий запис, але і багато чого іншого. Для прикладу це те, як потрібно керувати фінансами, як правильно строїти плани на виробництві і так далі зображені на рисунку 1.7

Також ресурс надає службу підтримки для допомоги вирішення проблем та надання відповідей на запитання. Це важлива перевага тому, що ресурс надає дуже багато можливостей. І через це у користувача може виникнути різні питання або можуть зіткнутись з помилками. Тому це важлива перевага ресурсу зображена на рисунку 1.8

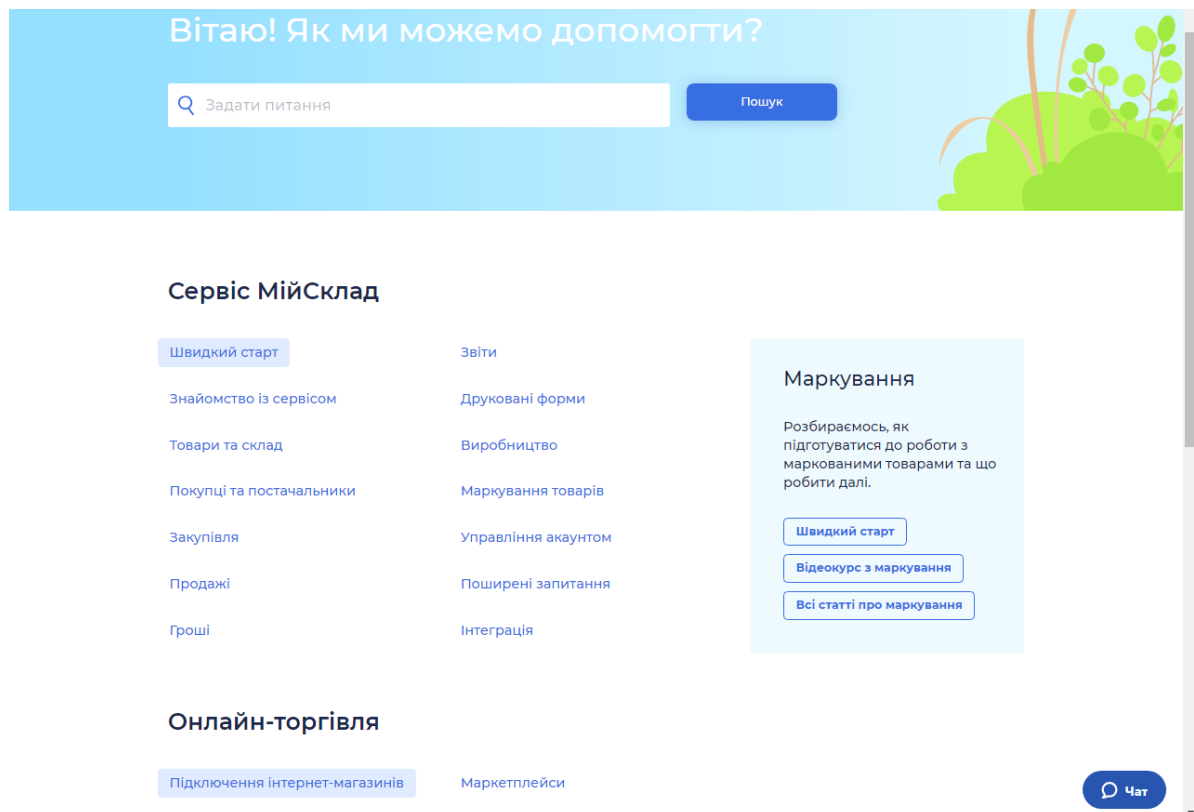


Рисунок 1.8 – Сторінка служби підтримки [25]

Недоліки також є і найголовніший – це те, що більшість можливостей платні зображено на рисунку 1.9

Тарифи для вашого бізнесу

[50% знижка для малого бізнесу](#)

Популярний

	Безкоштовний	Старт	Базовий	Проф	Корп
При оплаті за рік	0 Р	450 Р	850 Р	2465 Р	5865 Р
При оплаті за місяць		-	1000 Р	2900 Р	6900 Р
Інші варіанти					
	Підключити	Підключити	Підключити	Підключити	Підключити
	Завжди безкоштовно	Безкоштовно 14 днів	Безкоштовно 14 днів	Безкоштовно 14 днів	Безкоштовно 14 днів
Кількість співробітників	1	1	2 +	5 +	10 +
Точки продажу	1	1	0 +	0 +	5 +
Підключення інтернет-магазину	0	0	0 +	0 +	0 +

[Підтримка](#)

Ожидание msuindex.ru...

Рисунок 1.9 – Сторінка тарифів [25]

ма р: x Diagram x orderdra x перевод x Склады x Служба x Служба x

Можливості Тарифи Підтримка Маркетплейси Блог [Увійти](#) [Реєстрація](#)

ля, склад, хмарі

— в одній системі:
з, склад, фінанси,
альники

[Боту](#)

Кому вигідно

[Чат](#)

Рисунок 1.10 – Приклад того, як при натисканні на “Підтримка” відкривається нова сторінка у браузері [25]

Також ще один недолік полягає у тому, що при натисканні на кнопку – відкривається нове вікно у браузері з тою інформацією, яку він обрав зображено на рисунку 1.10

Отже, проаналізувавши ресурси онлайн-обліку ведення складу, кімнат було виділено наступні елементи, які необхідно реалізувати: особистий кабінет, можливість роботи з усіма видами документів, можливість робити звіт по тій інформації, яка потрібна, зручний інтерфейс.

1.4 Аналіз сучасних засобів створення програмного забезпечення

Одна із проблем яка постає перед розробником – це вибрати на якій платформі він буде розробляти свій продукт та на якій мові програмування. На сьогоднішній день основним платформами для розробки є десктоп, веб-платформа та мобільна. Тому перед тим як розробляти продукт, потрібно чітко обрати платформу на якій буде розроблятися продукт. А для цього, краще поговорити про кожну із них і виділити позитивні сторони та негативні.

Десктоп – це певний набір функцій які потрібні для того, щоб була можливість керувати ресурсами комп'ютера та взаємодіяти з користувачем [26].

Переваги десктопа: безпека, швидкість роботи додатку

Недоліки десктопа: відсутність доступу до інтернету, використання тільки з того пристрою, на якому воно встановлене.

Веб-платформа – це платформа, яка потрібна для того, щоб мати можливість розробляти веб-додаток і мати можливість керувати його інформацією [27]. На сьогоднішній день є одною з найбільш популярних платформ яку використовують розробники.

Переваги веб-платформи: нам не потрібно зберігати додаток на пристрої, є можливість користуватись додатком з будь-якої точки світу, зручність, простота.

Недоліки веб-платформи: для користування потрібен інтернет, залежність від браузерів та їх вимог для використання продукту, так як дані знаходяться на певному сервері, то нема гарантій, що злодій не викраде

інформацію з сервера або не станеться помилка, через яку станеться витік даних з серверу [28].

Мобільна платформа – це платформа, яка використовується для розробки додатків на мобільні пристрої. На сьогоднішній день знаходиться на одному рівні з веб-платформою по популярності використання [29].

Переваги мобільної платформи: зручність, швидкість.

Недоліки мобільної платформи: розробка додатку під декілька операційних системам, підтримка додатку є дуже дорогою у плані фінансів.

Після того, як було розглянуто платформи де буде працювати додаток, потрібно визначитись з платформою на якій буде розроблятися додаток. На сьогодні їх існує безліч. Основними є: PHP, Mono, Java.

PHP – це платформа, яка дає змогу розробляти веб-додатки. На сьогоднішній день є актуальним та часто використовується у розробці веб-додатку [30].

Переваги PHP: надається багато можливостей керувати сайтом, висока продуктивність, простий у розумінні

Недоліки PHP: не є самим безпечним, не підходить для створення десктопних додатків [31].

Mono – це платформа, яка дозволяє розробляти з відкритим кодом на основі .Net Framework і за допомогою цього у розробників є можливість робити крос-платформні програми [32].

Переваги Mono: зручність, можливість роботи з будь-якою БД, наявність вбудованого коду для оформлення звітів [33].

Недоліки Mono: великі фінансові витрати, на розробку йде багато часу.

Java – це платформа, яка надає нам можливість розробляти прикладне програмне забезпечення для крос-платформного програмного забезпечення. Також ця платформа надає нам можливість розробляти не тільки на Java, але і на інших мовах [34].

Переваги Java: зручність, безпека, можливість крос-платформної розробки

Недоліки Java: продуктивність низька, важкий у написанні та розумінні код

Проаналізувавши всю інформацію, було прийнято рішення вибрати для розробки інформаційної технології контролю складських запасів матеріалів на виробництві веб-технології.

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка та практична реалізація на платформі РНР інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

Для досягнення мети ставляться наступні завдання:

- Дослідити предметну область: огляд теоретичних підходів та існуючих програмних реалізацій до вирішення задачі контролю складських запасів матеріалів на виробництві.
- Провести аналіз сучасних засобів створення програмних продуктів.
- Виконати проєктування інформаційної системи підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.
- Створити інформаційну технологію підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.
- На базі запропонованої інформаційної технології створити програмну реалізацію та виконати тестування створеного програмного продукту.

Програмний продукт буде виконувати наступні функції:

- автоматизація ведення даних щодо потреб виробництва у визначених матеріалах для виготовлення визначених виробів;
- забезпечення формування планів виробництва;
- автоматизоване визначення потреб виробництва у визначених матеріалах згідно сформованих планів виробництва;

– реєстрація надходжень матеріалів на склад виробництва та автоматизоване визначення залишкових потреб для подальших надходжень у визначених матеріалах згідно сформованих планів виробництва;

– автоматизоване визначення надлишкових запасів матеріалів на складі згідно сформованих планів виробництва;

– автоматизація ведення даних щодо використання визначених матеріалів для виготовлення визначених виробів та ведення даних щодо інших втрат і витрат матеріалів;

– автоматизоване визначення очікуваних залишків матеріалів на складі згідно зареєстрованих втрат матеріалів і витрат матеріалів на виробництво для проведення ревізійної діяльності.

В якості області практичного застосування розробленої інформаційної технології контролю складських запасів матеріалів на виробництві слід використати взуттєве виробництво.

Розділ 2 Проектування інформаційної системи

2.1 Аналіз та автоматизація обробки потоків даних

2.1.1 Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві призначена для того, щоб правильно розрахувати матеріал для виробництва певного продукту.

Для коректного розрахунку інформаційна технологія передбачає наступні кроки виконання:

- Крок виконання при складанні плану виробництва враховує кількість продукту, який має йти на виробництво, враховує чи на складі вистачає матеріалу для продукту для ефективного розрахунку, щоб не було проблем з тим, що матеріалу не вистачило для того плану, який склали.

- Крок виконання при складанні плану виробництва враховує залишки матеріалу для подальшої простоти при проведенні ревізійної діяльності.

- Крок виконання при складанні плану виробництва проводить перевірку складських запасів і якщо матеріалу нема, то повідомляє про це. Також якщо потрібного матеріалу мало, повідомляє про це, щоб в подальшому не було проблем з складанням плану.

- Крок виконання при складанні плану виробництва враховує кількість кожного виду матеріалу, який потрібен для виробництва і якщо є такий випадок, коли кількість матеріалу більша, ніж зазначено у плані по виробництву. Тоді система опрацьовує цю кількість матеріалу і повертає залишок на склад. Це допомагає уникнути проблеми, коли забагато матеріалу для виробництва і є чіткий контроль за цим.

На рисунку 2.1 представлено схему інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

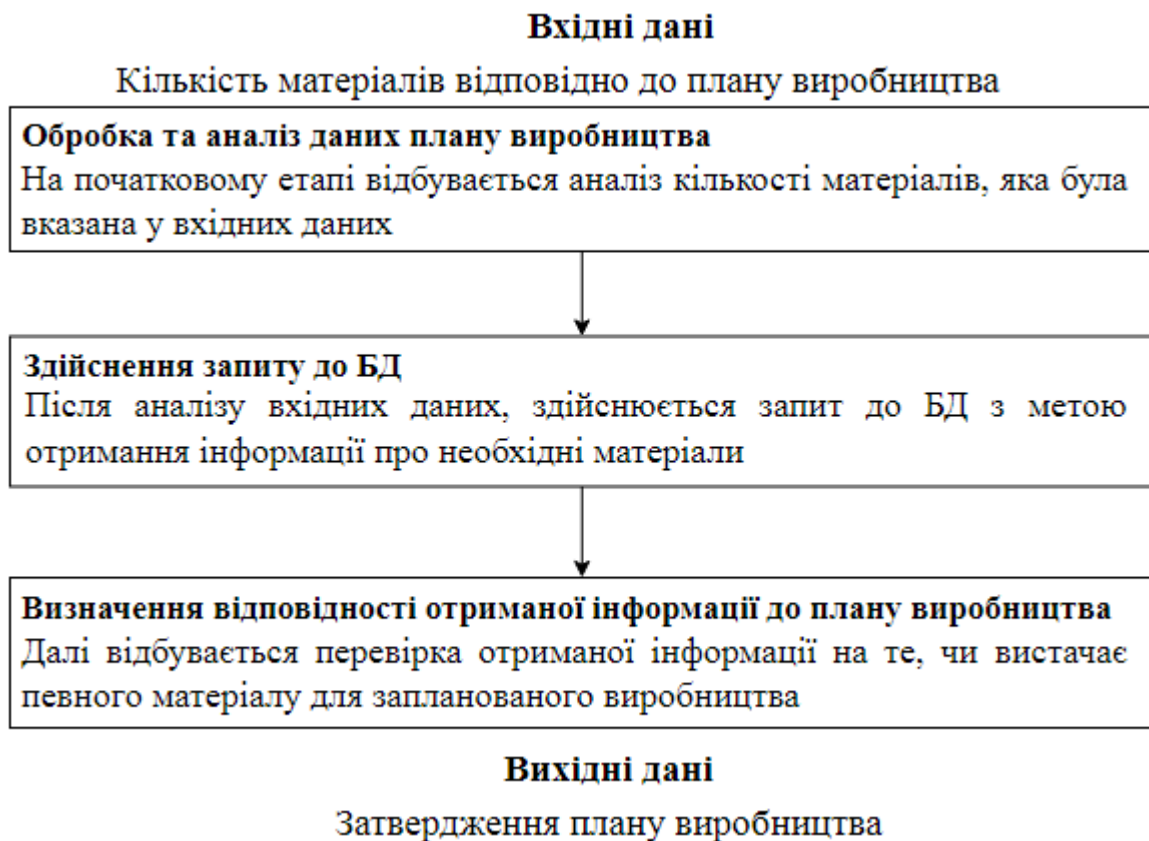


Рисунок 2.1 – Схема інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Ці кроки виконання є важливими компонентами інформаційної технології, які допомагають правильно приймати рішення по складанню планів на виробництво. Також інформаційна технологія потребує визначати середній товарний запас. Він потрібен для врахування обсягу товару на складі за певний проміжок часу. Розраховується він за наступною формулою:

$$C = \frac{T_1 + T_2}{2}, \quad (2.1)$$

де C – Середній товарний запас, T_1 – залишки на початку періоду, T_2 – залишки на кінець періоду

Ще один важливий пункт є вантажообіг. Він потрібен для того, щоб контролювати обсяг товарів отриманих на склад і відвантажених з нього за певний період. Визначається він за формулою:

$$B = \frac{T}{C} \quad (2.2)$$

де B – вантажообіг, T – товарообіг, C – вартість

Отже, інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві потрібна при роботі зі складськими запасами матеріалів та формуванні планів виробництва через те, що набагато спрощує роботу для користувача та допомагає ефективніше робити роботу та виконувати відповідні завдання. Якщо говорити про математичні формули, які застосовані у даній технології, то вони виглядають простими у розумінні але відіграють важливу роль у даному методу тому, що завдяки їм проводяться важливі розрахунки, де потрібна чітка відповідь та максимально правильна.

2.1.2 Функціональна структура інформаційної системи

Відповідно до поставленого завдання потрібно автоматизувати групи функцій для користувача. Відповідні групи функцій представлені в таблиці 2.2

Таблиця 2.1 – Групи функцій

Номер групи функцій	Назва групи функцій
1	Робота користувача з головною сторінкою
2	Реєстрація користувача в системі
3	Авторизація користувача в системі
4	Робота користувача з документами
5	Робота користувача зі складом для виконання замовлення

Функція “Робота користувача зі складом для виконання замовлення” призначена для того, щоб користувач коли записує в програму, скільки матеріалу було взято зі складу, програма аналізує скільки осталося матеріалу певного на складі і у випадку його закінчення, сповіщає про це користувача. Це відбувається наступним чином:

1. Для початку користувачу потрібно зайти в розділ “Склад” та вибрати пункт “Матеріали”

2. Після чого користувача переадресовує на таблиці з інформацією про те, який матеріал є на складі та його кількість

3. Далі користувачу потрібно натиснути на кнопку “Редагувати” і після чого йому відкриється форма для цього

4. Форма містить наступні поля: “Вид матеріалу”, “Кількість”. Далі користувачу потрібно вибрати який саме матеріал йому потрібен та його кількість

5. Після цього, натиснути на кнопку “Зберегти”. Далі форма переадресовує інформацію на сервер, де вона обробляється і порівнює, чи є на складі такий матеріал та кількість на складі.

6. Якщо нема матеріалу або кількість, яку вводить користувач перевищує ту, що на складі, програма повідомить про помилку

7. Також якщо кількість певного матеріалу невелика, то програма попередить про це.

8. Якщо все пройшло успішно, то данні будуть збережені, а користувача переадресує на головну сторінку

Функція “Робота користувача з особистою сторінкою” призначений для редагування особистої інформації. Редагування особистої інформації користувача проводиться наступним чином:

1. Знаходячись на головній сторінці сайту, користувачу потрібна в правому верхньому куті натиснути на кнопку “Особистий кабінет”. Після натискання, його переадресовує на сторінку, де він має змогу переглядати свою особисту інформацію

2. Після цього, користувачу потрібно натиснути на кнопку “Редагувати інформацію” і після цього користувача переадресовує на сторінку для редагування

3. У формі для редагування, користувачу виводить його інформація в полях: “Ім’я”, “Прізвище”, “Електронна адреса”, “Номер телефону”, “Пароль”.

4. Далі користувачу потрібно обрати відповідні поля та змінити ту інформацію, яку потрібно

5. Після того, як користувач змінив всю інформацію, потрібно натиснути на кнопку “Зберегти зміни”. Після чого, форму відправить на сервер з оновленою інформацією

6. Після цього, користувача переадресує назад на сторінку з особистою інформацією.

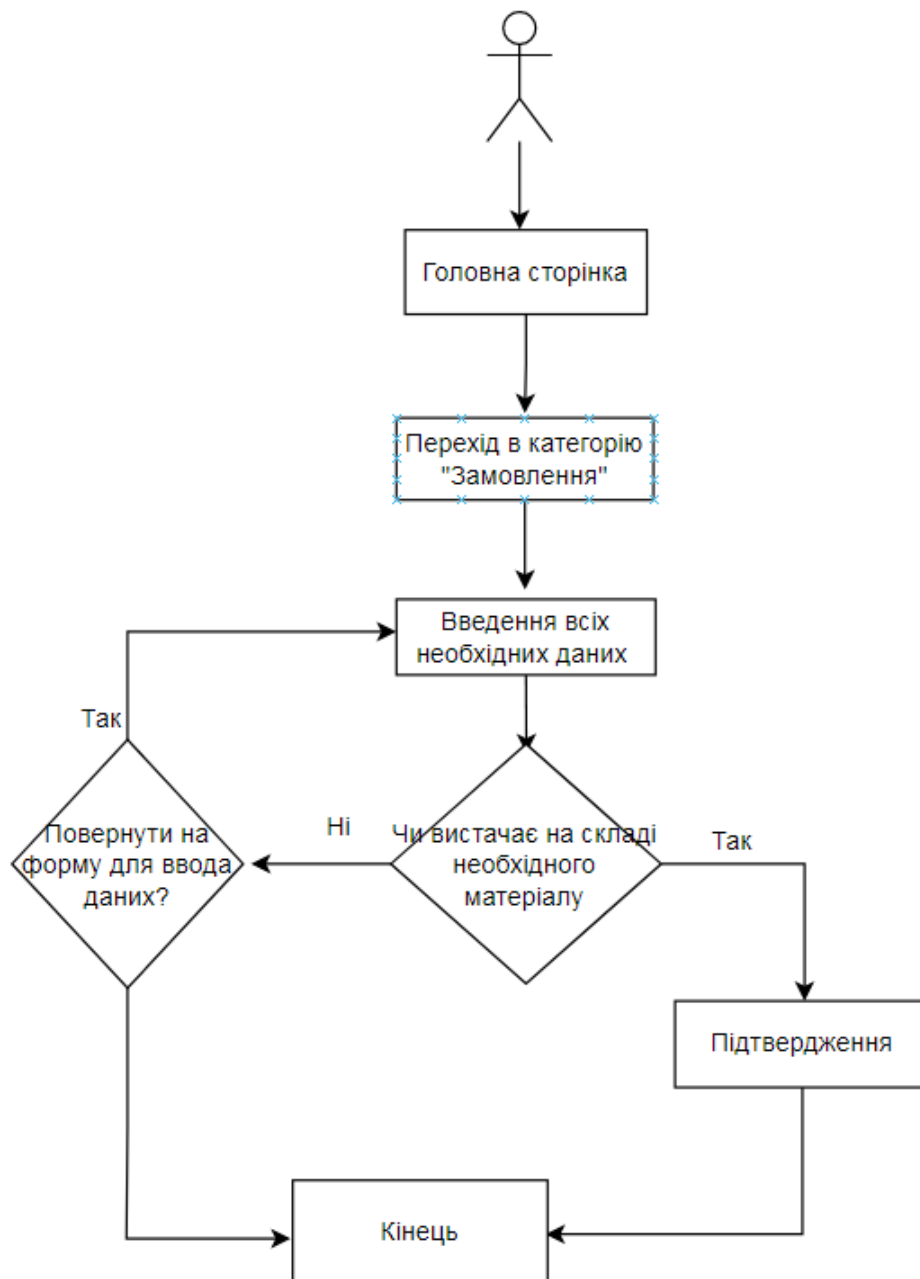


Рисунок 2.2 – Діаграма функції “Робота користувача зі складом для виконання замовлення”

Функція “Реєстрація користувача в системі”:

1. Функція “Реєстрація користувача в системі” потрібна для того, щоб мати змогу зареєструватись та мати можливість користуватись можливостями ресурсу. Для цього потрібно відтворити наступні кроки:

2. Знаходячись на головній сторінці, потрібно натиснути на кнопку в правому нижньому кутку “Реєстрація”. Після натискання, користувача переадресовує на форму для реєстрації

3. Після переадресування, користувачу потрібно ввести наступну інформацію в полях: “Ім’я”, “Прізвище”, “Електронна адреса”, “Номер телефону”, “Пароль”. Мінімальна кількість символів у полі “Пароль” – 6 символів, “Електронний адрес” повинна бути валідність. Після заповнення, користувачу потрібно натиснути на кнопку “Зареєструватись”.

4. Після натискання, форму буде відправлено на сервіс де буде збережено дані користувача

5. Потім користувачу виведе вікно “Реєстрація пройшла успішно” і переадресує користувача на головну сторінку.

Функція “Авторизація користувача в системі” призначена для того, щоб користувач мав змогу авторизуватись в системі для користування можливостями ресурсу. Для цього потрібно пройти наступні кроки:

1. Знаходячись на головній сторінці, потрібно натиснути в правому верхньому кутку на кнопку “Вхід” і після чого відбудеться переадресація на форму для входу в систему.

2. Форма авторизація передбачає наступні поля які потрібно ввести користувачу для успішної авторизації: “Електронна адреса” та “пароль”. Мінімальна кількість символів у полі “Пароль” – 6 символів, “Електронний адрес” повинен бути валідним.

3. Після того, як користувач закінчив заповнювати поля, потрібно натиснути на кнопку “Вхід” і форма буде надіслана на сервіс для перевірки інформації щодо правильності введених даних та чи є взагалі такий користувач в системі

4. Якщо система підтвердить правильність інформації, то користувач отримає повідомлення “Успішний вхід” і буде авторизований, і далі переадресує на головну сторінку

5. Якщо система виявить помилку, то користувач отримає повідомлення “Електронна адреса або пароль були введені некоректно” і користувачу потрібно буде повторити процедуру.



Рисунок 2.3 – Діаграма функції “Реєстрація користувача в системі”

Функція “Робота користувача з документа” призначений для того, щоб надати можливість користувачу працювати з документами по виробництву. Процес передбачає автоматизацію процесу ведення даних надходження матеріалу на склад, визначення кількості матеріалів для виробництва виробів.

Для ведення даних про надходження матеріалу на склад потрібно виконати наступні кроки:

1. Користувачу потрібно авторизуватись у системі. Після успішної авторизації потрібно перейти на головній сторінці у меню “Операції з документами” та обрати категорію “Надходження”

2. Після цього, користувача буде переадресований на сторінку, де буде представлено таблицю з полями: “Назва товару”, “Назва установи”, “Кількість матеріалу”

3. Далі користувачу потрібно натиснути на кнопку “Додати інформацію” і після цього буде переадресований на форму для додавання надходження в таблицю з відповідними полями

4. Після успішного заповнення, користувач повинен натиснути на кнопку “Зберегти” і після того дані будуть збережені на сервісі.

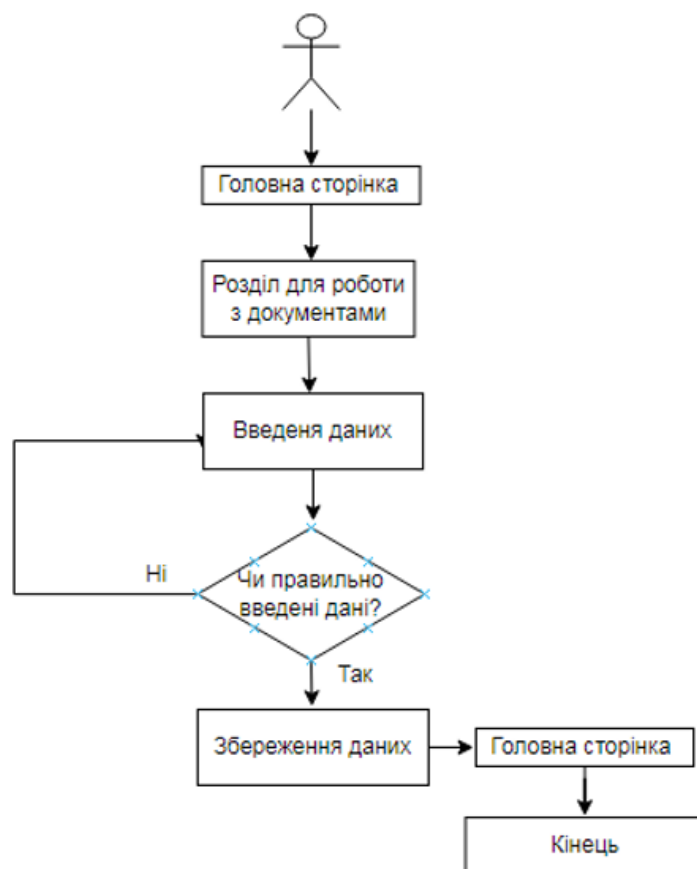


Рисунок 2.4 – Діаграма функції “Робота користувача з документами”

Визначення кількості матеріалів для виробництва виробів проводиться наступним чином:

1. Знаходячись на головній сторінці, користувачу потрібно перейти на головній сторінці у меню “Операції з документами” та обрати категорію “Виробництво”

2. Після цього користувача буде переадресовано на сторінку з таблицею, де будуть доступні наступні поля: “Назва виробу”, “Назва матеріалу”, “Кількість виробу”, “Кількість матеріалу”

3. Далі користувачу потрібно натиснути з правої сторони на кнопку “Добавити” і після цього його буде переадресовано на форму для додавання інформації. Поле “Кількість виробу” повинне залежати від того, скільки є на складі матеріалу.

4. Після того, як користувач закінчив водити, йому потрібно натиснути на кнопку “Додати” і після того, форма буде надіслана на сервер. Потім користувачу виведе вікно з текстом “Дані збережено успішно”

Зважаючи на ті групи функцій, які були описані вище, можна переходити до проектування БД.

2.2 Інформаційна структура системи

2.2.1 Проектна архітектура системи та взаємозв’язок компонентів

Інформаційна система підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві включає в собі модулі, які призначені для вирішення певних завдань. На схемі 2.6 зображено структура інформаційної системи.



Рисунок 2.5 – Схема модулів інформаційної системи підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Далі необхідно описати кожен з модулів та їх призначення:

– Модуль авторизації та реєстрації відповідає безпосередньо за можливість користувача пройти без проблем реєстрацію або якщо він вже зареєстрований, то надати можливість авторизуватись, тобто щоб користувач мав можливість отримати доступ до використання інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

– Модуль визначення надходжень матеріалів на склад призначений для того, щоб користувач мав можливість добавляти на склад ту кількість матеріалу, яка прийшла. Обов'язковим критерієм користування даною можливістю є проходження авторизації або реєстрації.

– Модуль визначення потреб у виробництві за планом призначений для того, щоб користувач мав можливість правильно розподіляти матеріали для виробництва. План дій полягає у наступному: користувач формує план виробництва. Після цього, система робить перевірку на складі, чи є достатня кількість матеріалу. Далі, після перевірки кількості потрібного матеріалу, якщо достатньо, то план буде прийнятий. Якщо, матеріалу не вистачає, то система сповістить про це користувача та проінформує скільки саме не вистачає певного

матеріалу. Далі, на основі цієї інформації, користувач має змогу сформувати перелік матеріалу, який потрібно замовити для виконання плану виробництва.

– Модуль визначення надлишкових запасів матеріалів призначений для того, щоб конкретно визначити надлишки матеріалу, який залишиться після плану виробництва. План дій полягає у наступному: спочатку формується план виробництва. Далі після того, як був сформований план, система визначає витрати матеріалів відповідно до плану. Якщо матеріалу було задіяно менше ніж спочатку це було зазначено в плані по виробництву, то система визначить точну кількість матеріалу та поверне її на склад. Також даний модуль призначений для того, щоб мати можливість після складання плану на виробництво визначити скільки різного виду матеріалу залишиться на складі, тобто залишок. Це потрібно для того, щоб можна було краще визначати залишок по матеріалу на складі та чітко визначити потребу для певного виду матеріалу та знати чітку кількість, яку потрібно замовити.

Отже, кожен модуль відповідає за свої функції у даній системі і кожен із них є необхідний для ефективної роботи інформаційної технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

2.2.2 Інформаційна модель

Перед тим, як приступити до розробки, потрібно добре вивчити предметну область і дослідити всі нюанси, з якими можна зустрітись. Після цього можна зайнятись проектування БД

Кожна з таблиць в БД існує для того, щоб зберігати певну інформацію. Архітектура БД зображена на рисунку 2.6 і має наступні таблиці та атрибути.

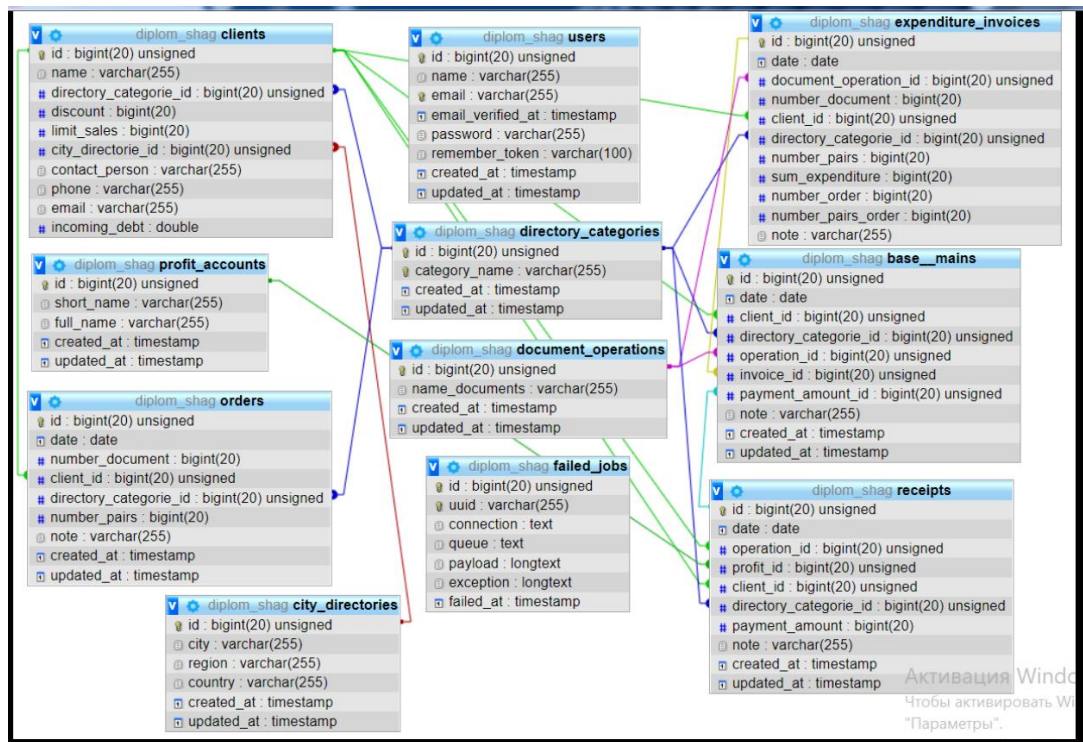


Рисунок 2.6 – Архітектура БД

Таблиця “Customer_Directory” призначена для збереження даних про клієнтів і має наступні атрибути: ClientID, NameClient, Category, Discount, LimitRealizations, City, FullNameClient, Number, Email, IncominDebt(таблиця 2.2)

Таблиця 2.2 – Атрибути таблиці Customer_Directory

№ п/п	Назва	Тип даних	Опис
1	ClientID	int	Унікальний ідентифікатор
2	NameClient	nvarchar(255)	Прізвище
3	Category	nvarchar(255)	Назва категорії
4	Discount	int	Вторинний ключ (знижка)
5	LimitRealizations	double	Ліміт на реалізацію
6	City	nvarchar(255)	Вторинний ключ (місто)
7	Number	int	Телефон
8	Email	nvarchar(255)	Електронна адреса
9	IncominDebt	double	Борг

Таблиця Category призначена для того, щоб зберігати дані по категоріям і має наступні атрибути: CategoryID, NameCategory (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці Category

№ п/п	Назва	Тип даних	Опис
1	CategoryID	int	Унікальний ідентифікатор
2	NameCategory	nvarchar(255)	Назва категорії

Таблиця City_Directory призначена для того, щоб зберігати інформацію про те, з якого місця клієнт робить замовлення або приходить товари має наступні атрибути: ID, City, Region, Country (таблиця 2.4)

Таблиця 2.4 – Атрибути таблиці City_Directory

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	City	nvarchar(255)	Назва міста
3	Region	nvarchar(255)	Назва області
4	Country	nvarchar(255)	Назва країни

Таблиця Order призначена для того, щоб зберігати інформацію про замовлення і має наступні атрибути: ID, Date, Name_Document, NameClient, Category, Number, Sum, Note (таблиця 2.5)

Таблиця 2.5 – Атрибути таблиці Order

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	Date	date	Дата оформлення замовлення
3	Name_Document	nvarchar(255)	Вторинний ключ (назва операції)
4	NameClient	nvarchar(255)	Вторинний ключ (назва клієнта)
5	Category	nvarchar(255)	Вторинний ключ (назва категорії)
6	Number	int	Кількість
7	Sum	double	Сума
8	Note	nvarchar(255)	Примітка

Таблиця Document_Operation призначена для того, щоб зберігати інформацію про назву операції і містить наступні атрибути: ID, NameDocument (таблиця 2.6)

Таблиця 2.6 – Атрибути таблиці Document_Operation

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	NameOperation	nvarchar(255)	Назва операції

Таблиця Expenditure_Document має наступні атрибути: ID, Date, NameDocument, NumberDocument, NameClient, Category, Number, Sum, NumberOrder, Note (таблиця 2.7)

Таблиця Receipt призначена для того, щоб зберігати дані про надходження матеріалів на склад і містять наступні атрибути: ID, Date, NameDocument, NameClient, City, Category, Sum, Note (таблиця 2.8)

Таблиця 2.7 – Атрибути таблиці Expenditure_Document

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	Date	date	Дата оформлення замовлення
3	NameDocument	nvarchar(255)	Вторинний ключ (назва операції)
4	NumberDocument	nvarchar(255)	Номер документа
5	NameClient	nvarchar(255)	Вторинний ключ (назва клієнта)
6	Category	nvarchar(255)	Вторинний ключ (назва категорії)
7	Number	int	Кількість
8	Sum	double	Сума
9	NumberOrder	int	Вторинний ключ (номер замовлення)
10	Note	nvarchar(255)	Примітка

Таблиця 2.8 – Атрибути таблиці Receipt

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	Date	date	Дата
3	NameDocument	nvarchar(255)	Вторинний ключ (назва операції)
4	NameClient	nvarchar(255)	Вторинний ключ (назва клієнта)
5	City	nvarchar(255)	Вторинний ключ (назва міста)
6	Category	nvarchar(255)	Вторинний ключ (назва категорії)
7	Sum	double	Сума
8	Note	nvarchar(255)	Примітка

Таблиця Warehouse призначена для того, щоб зберігати інформацію про предмети, які знаходяться на складі і має наступні атрибути: ID, Name, Numbers, NameWorker (таблиця 2.9)

Таблиця 2.9 – Атрибути таблиці warehouse

№ п/п	Назва	Тип даних	Опис
1	ID	int	Унікальний ідентифікатор
2	Name	nvarchar(255)	Назва
3	Numbers	int	кількість
4	NameWorker	nvarchar(255)	Назва постачальника

Таким чином, представлена база даних підходить для функціонування веб-застосунку для інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Далі необхідно обрати засоби для розробки даної системи

2.3 Вибір засобів розробки інформаційної системи

2.3.1 Вибір мови програмування

Для мого проекту було вибрано мову програмування PHP. PHP – це мова програмування, яка використовується для створення веб-сайтів. На сьогоднішній день є одна з найпопулярніших у використанні і має багато переваг над іншими мовами такі як: [35].

– Простота. Головна проблема з якою зіштовхується кожен програміст – це синтаксис, функція, та бібліотеки, які надає мова програмування. Якщо говорити про PHP, то всі ці 3 аспекти є не складними у вивченні та опануванні людиною, яка має бажання опанувати цю мову. А якщо цієї людиною є досвідчений програміст, то опанувати мову йому не складе труднощів.

– Підтримка ОС. Взагалі це є важливим пунктом тому, що людина може програмувати на різних операційних системах. Наприклад один може програмувати на Windows, другий на Linux і так далі. Якщо говорити про PHP, то дана мова підтримується практично всіма операційними системами і це дає змогу без проблем розгорнути програмний продукт без помилок.

– Швидкість. Мова програмування РНР є найшвидшою порівняно з іншими мовами. Це дуже гарно видно, коли потрібно завантажити якийсь РНР додаток з повільного інтернету. Також коли доходить діло до підключення будь-якої мови програмування до БД, то на це йде багато часу і це вже є невеликою, але проблемою. Але якщо говорити про РНР, то тут цієї проблеми нема. Підключення до БД та отримання необхідної інформації з неї відбувається дуже швидко.

– Відкритий код. РНР є відкритим кодом та найголовніше – це безкоштовним. Це надає можливість кожному без проблем використовувати у своїх цілях. Також завдяки цьому, кожен може користуватись фреймворками, які написані на мові РНР.

– Безпека. Коли йде мова про безпеку використання інформації та даних з БД, то є мови програмування, де потрібно вручну це все розробляти. РНР має достатню кількість вбудованих функцій та готових бібліотек для цього, які гарно справляються з безпекою програмного продукту та всією необхідною інформацією, яка використовується.

– Підтримка. Найважливішим пунктом, коли програмуєш на будь-якій мові програмування – це її документація. В документації знаходиться практично вся інформація про мову програмування та які можна використовувати функції, бібліотеки, синтаксис і так далі. В РНР є дуже добре описана документація, яка допомагає вирішити багато яких проблем при програмуванні. Також важливим фактором підтримки який присутній в РНР є не тільки документація, а інтернет спільнота, яка готова допомагати друг другу у вирішенні різних задач та проблем. Також допомагати початківцям ефективно та просто опанувати мову програмування.

– Стабільність. Мова програмування РНР є доволі старою мовою. Але не дивлячись на це, по сьогоднішній день займає одну з найвищих сходинок при виборі мови програмування для програмного продукту. Також підтримка з боку розробників не закінчилась, а продовжується.

Отже, дивлячись на всі переваги PHP можна зробити висновок, що саме ця мова гарно підходить для розробки веб-додатку інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві якісно та зручним у використанні

2.3.2 Вибір фреймворку

Кожна людина прагне покращити щось до такого стану, щоб на це потрібно було витратити менше часу. Наприклад раніше для того, щоб прочитати книгу, потрібно було йти в бібліотеку і брати там книгу або навіть там і читати її. А сьогодні є можливість будучи дома зайти в Інтернет і знайти будь-яку книжку і при цьому не потрібно нікуди йти. Так і з програмуванням. Раніше для вирішення певних задач потрібно було писати дуже великий за розміром код і підключати багато яких можливостей для взаємодії наприклад БД з проектом. Але на це витрачається дуже багато часу і потім іншому розробнику прочитати код буде дуже складно. Тому для того, щоб спростити цей процес та зробити його більш зручнішим та кращим існує фреймворк.

Фреймворк – це готова модель, шаблон для програмної платформи, на основі якого дає змогу дописувати свій власний код. Фреймворк бере на себе безліч нюансів які зв'язані з розробкою. Для прикладу, якщо є певні помилки, то фреймворк їх обробляє і надає чітку відповідь про те, що саме спричинило цю помилку і де її можна знайти та рекомендації по її усуненню. Далі – це підключення до БД. Якщо подивитись на код, який потрібно написати для підключення до БД, то можна побачити, що його доволі багато і це не так просто так як багато чого в коді написати, щоб з'єднання було успішним. А це в свою чергу займає певний час, який можна було використати для вирішення іншої задачі. Тому фреймворк прибирає від нас цю проблему тим, що в його структурі вже написаний код для підключення до БД і користувачу потрібно лише написати назву своєї БД і далі вже напряму з нею працювати [36]. Практично

кожна мова програмування має свої готові фреймворки. В даному проекті було обрано фреймворк Laravel.

Laravel – це фреймворк, який написаний на мові програмування PHP для зручної розробки програмного продукту та веб-додатків. В даного фреймворку є ряд переваг, через які його і обирають для розробки. Наприклад:

– MVC. Напевно це найголовніша перевага цього фреймворку через те, що є чітке розділення програмного коду по правилам та коли починаєш виправляти помилку в одній частині, іншу частину це не ламає і логіка, яка там була написана залишається.

– Безпека. При розробці крупних та серйозних проектів безпека ставиться на самому першому місці. Що стосується даного фреймворку, то можна бути спокійним через те, що фреймворк передбачає безпеку при роботі з даними та необхідною інформацією.

– Сумісність. Основний язык даного фреймворку є PHP, але це не заважає підключати інші язики до проекту. Для прикладу це може бути JS.

– Документація. Так як і в мові програмування, в фреймворку Laravel є своя документація, де чітко описаний всі необхідні функції і бібліотеки та як їх правильно підключити до проекту та використати. Документація написана доступно і не виникає важкості в її опануванні.

– Технологія пов'язання моделі з БД. Даний фреймворк має особливий шаблон для роботи з БД. Тобто, даний фреймворк забирає безліч коду, який потрібно було написати на мові програмування для підключення до БД.

Тому, можна зробити висновок про те, що фреймворк займає важливе місце при розробці великого або навіть маленького програмного продукту. Також за допомогою фреймворку є можливість не допустити багатьох помилок, які можуть виникнути при розробці. А фреймворк Laravel зважаючи на всі його переваги ідеально підходить для розробки інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

2.3.3 Вибір редактора програмного коду

Дуже часто при створенні проекту, потрібно написати дуже великий код який буде містити безліч методів, функцій, назв змінних і так далі. І тут виникає головна проблема, яка може завадити завершити розробку. Це помилка, яку програміст допустив під час написання коду. І щоб її знайти, потрібно дуже добре перевіряти кожен рядок. І добре, якщо їх буде тільки 50. А якщо біля 1000, то не факт, що він її знайде. Тому для цього створили спеціальні редактори коду, які допомагають вирішити цю проблему і не тільки її.

Редактор програмного коду – це текстовий редактор, який допомагає створювати та редагувати програмний код. Редактори мають деякі функції, які допомагають спрощують та прискорюють написання коду [37]. Для прикладу це виділення помилки та вказування користувачеві де саме вона і розробник не потрібно витратити час на її пошук. Також редактор під час написання коду пропонує свої варіанти того, що хоче розробник (наприклад коли пише назву методу і редактор його пропонує підставити повністю). Є безліч редакторів, які підходять для розробки веб-додатків. Для прикладу це може бути Sublime Text, Visual studio Code, WebStorm і багато інших. В даному проекті буде використовуватись редактор Visual Studio Code.

На сьогоднішній день Visual Studio Code займає одне із перших місць по популярності використання розробниками. Він був створений компанією Microsoft і його можна використовувати на операційній системі Windows, Linux [38].

Найголовнішими його перевагами є:

– Можливість працювати одразу з декількома мовами програмування в одному редакторі. Дуже добре, коли редактор коду може підтримувати дві і більше мов програмування. Якщо говорити про Visual Studio Code, то цей редактор підтримує майже всі мови програмування і це дає зручність при розробці, бо не потрібно відкривати додаткові редактори для тих чи інших мов.

– Додаткові можливості Visual Studio Code роблять так, щоб розробка програмного продукту була цікавіша та ефективніша. Наприклад можливість підключати додаткові бібліотеки, або є можливість змінити колір яким пишуть синтаксис мови програмування. Також додатково можна включити можливість автодоповнення що ж дуже добре через те, що коли пишеться код, буває потрібно написати назву функції, а вона по розмірам не маленька, а автодоповнення вже по першим літерам пропонує варіанти і залишається лише обрати ту функцію або зміну, яка потрібна

– Також головною перевагою цього редактор є те, що можна напряму взаємодіяти з GitHub. Простими слова, GitHub – це платформа, яка надає можливість зберігати проекти, які в процесі розробці. Дуже часто ним користуються тоді, коли йде саме команда розробка.

Тому, можна зробити висновок про те, що обирати редактор коду потрібно уважно через те, що від вибору напряму залежить ефективність та зручність написання коду. Visual Studio Code гарний вибір для розробки інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

2.3.4 Вибір СКБД

Один із найважливіших етапів створення додатку є вибір БД. Це є важливим аспектом через те, що йде мова про роботу з інформацією, яка буде застосовуватись в програмному продукту. І потрібно, щоб вибрана БД дозволяла це зробити без проблем та помилок. Один із варіантів є MYSQL.

MYSQL – це реляційна система управління базами даних із відкритим вихідним кодом. Вона одна із найпопулярніших БД, яка використовується у веб-додатках. В неї є ряд переваг, які роблять її популярною у використанні серед програмістів та інших користувачів. Для прикладу

– У час існування фреймворків, більшість користувачів використовує їх у свої проектах і добре, коли БД підтримує певний фреймворк і не видає помилок.

MYSQL є одною з тих БД, яка підтримується при роботі з багатьма фреймворками [39].

– При роботі з БД, дуже важливим критерієм є простота та зручність у використанні. MYSQL відповідає цим характеристикам через те, що не є складним у розумінні та має просто команди для роботи з нею

– Завжди де є робота з інформацією та її подальшим зберіганням постає питання безпеки використання даних. Якщо програмний продукт не є великого масштабу, то на це не приділяється особливої уваги. Але якщо проект серйозний, то безпека ставиться на перше місце. MYSQL справляється з цим на найвищому рівні через функції, які шифрують дані.

– Коли є багато інформації, з якою постійно проводиться певна робота, то важливим критерієм є швидкість роботи з інформацією. MYSQL дозволяє працювати з інформацією без лишніх затримок у загрузці, або коли потрібно щось змінити або додати, то не буде затримок.

Отже, можна зробити висновок про те, MYSQL є правильним вибором через вище сказані переваги. MYSQL була використана для роботи з інформаційною технологією підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

Розділ 3 Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення програмних складових системи

Метою є розробка прикладної реалізації інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Для розробки інформаційної системи було використано мову програмування PHP, а також систему керування базами даних MySQL.

Розроблена система призначена для людей, яким потрібно працювати з різними видами документів на виробництві. Реалізована автоматизація прийняття рішень з контролю складських запасів матеріалів на виробництві дозволяє підвищити ефективність відповідної діяльності як людям в яких вже є досвід роботи з документами, так і тим хто тільки починає з ними працювати.

У проекті було реалізовано контролери для роботи з даними. Кожен з них відповідає за свої обов'язки. Відповідно коротка характеристика кожної моделі:

- Client – відповідає за ведення списку клієнтів, які роблять оформляють замовлення.
- Document_Operation – відповідає за операції які проводяться з матеріалами(тобто для виробництва або навпаки прихід на склад)
- Expenditure_Invoice – відповідає за витрати на закупку матеріалів
- Order – відповідає за оформлення замовлення
- Warehouse – відповідає безпосередньо за склад, де зберігаються матеріали
- Production_Plan – відповідає за зберігання інформації про план виробництва.
- City – відповідає за зберігання інформації про місто, область та країну.
- Profit – відповідає за зберігання інформації про те, яку операцію провів замовник.
- User – відповідає за зберігання інформації про користувача

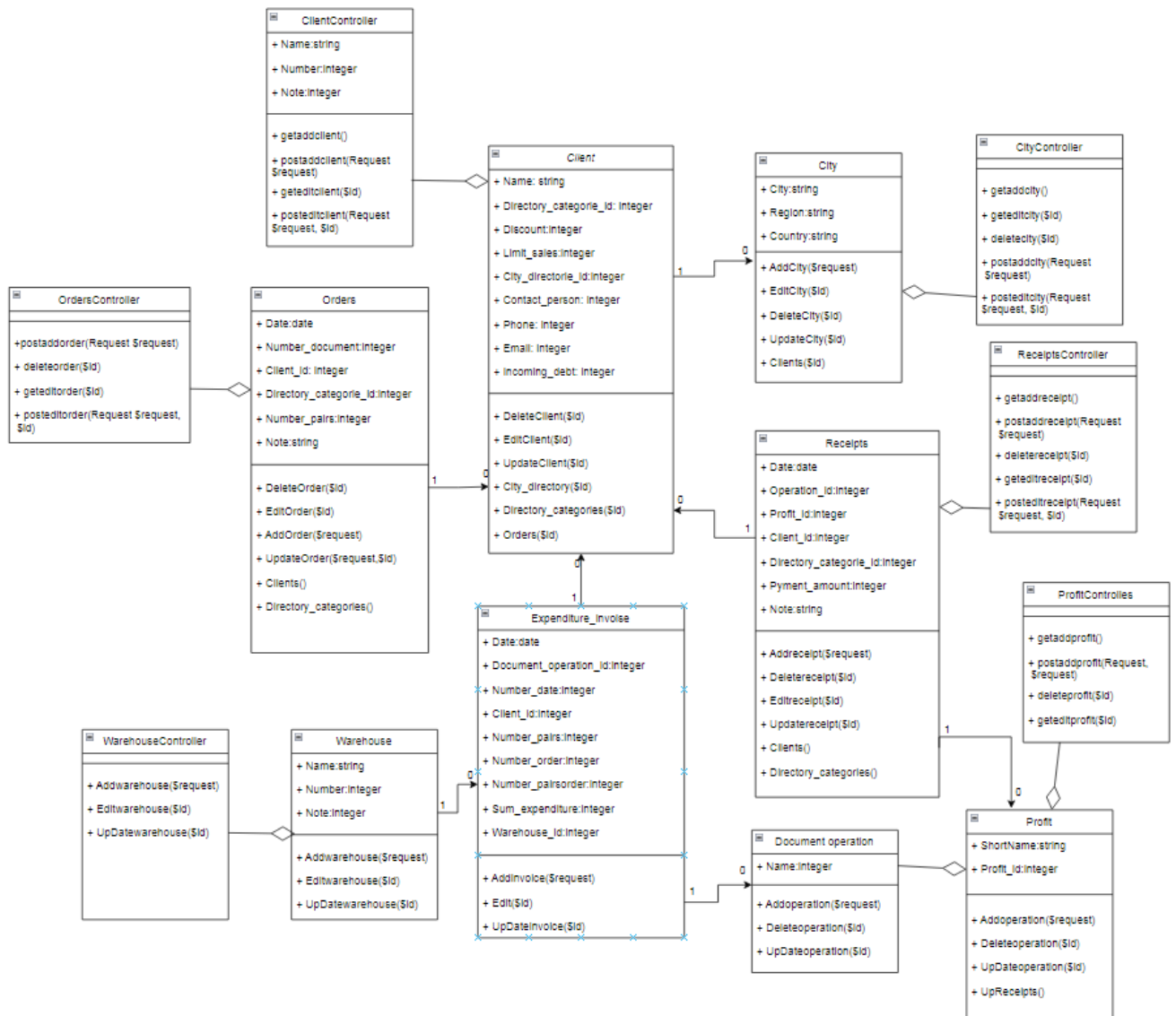


Рисунок 3.1 – Діаграма класів

Також до кожної моделі є контролер у які забезпечують взаємодію для моделі з представленням. Основні контролери наступні:

- UserController – відповідає за забезпечення взаємодії моделі User з представленням ViewUser.
- OrderController – відповідає за забезпечення взаємодії моделі Order з представленням ViewOrder
- ClientController – відповідає за забезпечення взаємодії моделі Client з представленням ViewClient
- InvoiceController – відповідає за забезпечення взаємодії моделі Expenditure_Invoice з представленням ViewInvoice

– Production_PlanController – відповідає за забезпечення взаємодії моделі Production_Plan з представленням ViewProduction

– CityController – відповідає за забезпечення взаємодії моделі City з представленням ViewCity

– ProfitController – відповідає за забезпечення взаємодії моделі Profit з представленням ViewProfit

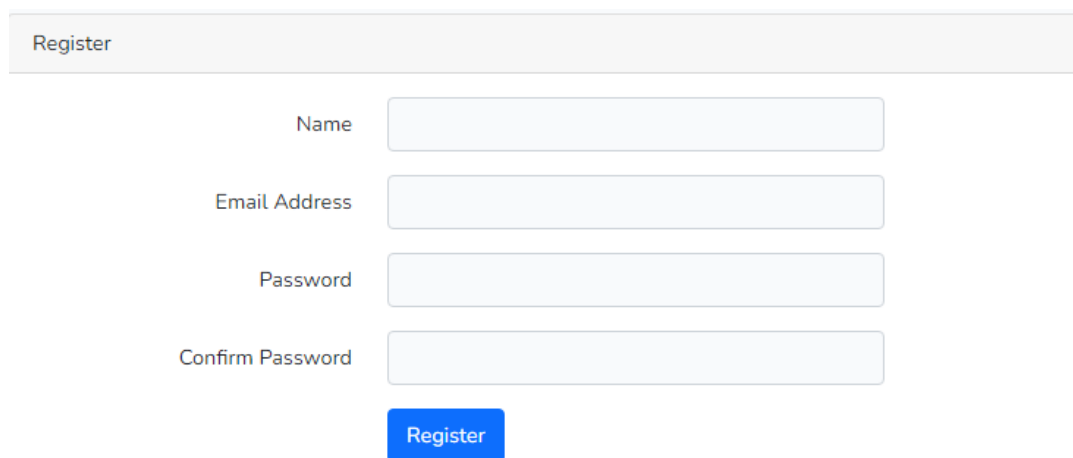
– OperationController – відповідає за забезпечення взаємодії моделі Document_Operation з представленням ViewOperation

Детальніше їх взаємодію між собою представлено на діаграмі класів на рисунку 3.1.

Таким чином було досліджено моделі та контролери, які використовуються у технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві.

3.2 Особливості реалізації програмних складових системи

Для початку, щоб користувач міг користуватись програмним продуктом потрібно пройти авторизацію або реєстрацію. Реалізація реєстрації зображена на рисунку 3.2



The image shows a registration form with a title bar labeled 'Register'. Below the title bar, there are four input fields stacked vertically, each with a label to its left: 'Name', 'Email Address', 'Password', and 'Confirm Password'. At the bottom of the form, there is a blue button with the text 'Register' in white.

Рисунок 3.2 - Форма для реєстрації

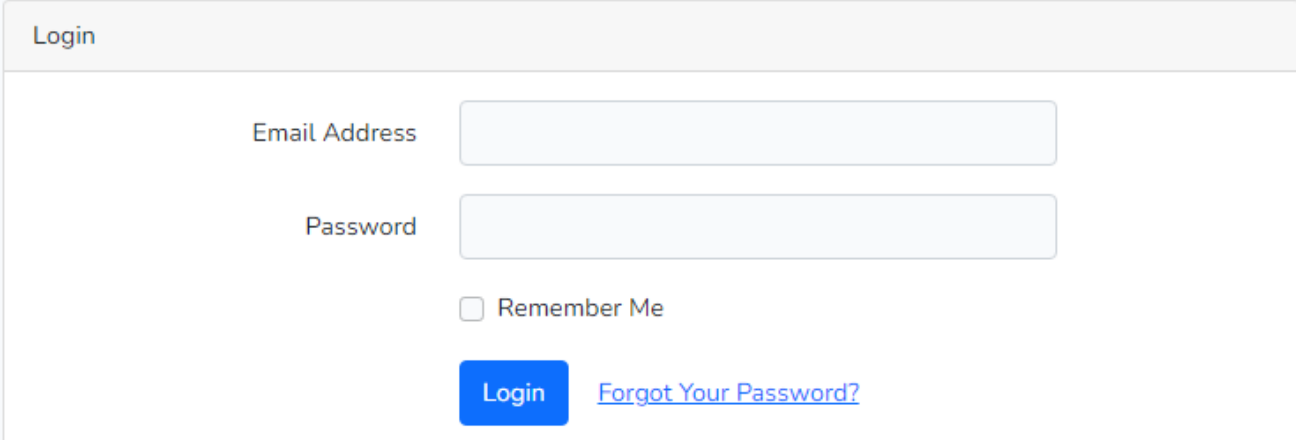
Ця форма була реалізована за допомогою наступного коду:

```
public function registerProcess(Request $request)
{
    try
    {
        $messages = [
            'email.required'          ==> "Ведіть пошту"
            'email.email'             ==> "Помилка при введенні пошти"
            'password.required'       ==> "Ведіть пароль"
            'password.between'        ==> "мінімальна довжина пароля - 8 символів"
        ];
        $this->validate($request, [
            'email' ==> 'required|email',
            'password' ==> 'required|between:8, 100',
        ], $messages);
        $input = $request->all();

        if (Session::get('user') && Session::get('user')->id) {
            $input[Session::get('net').'_id'] = Session::get('user')->id;
        }

        $pas = ['email' => $request->email];
        if($user = Sentinel::findByCredentials($pas))
        {
            return Redirect::to('register')
                ->withErrors('Така пошта вже є')
        }
    }
}
```

Далі була реалізована форма для авторизації яка зображена на рисунку 3.3.



The image shows a web form for logging in. At the top, there is a header bar with the word "Login". Below this, there are two input fields: "Email Address" and "Password". Under the "Password" field, there is a checkbox with the text "Remember Me". At the bottom of the form, there is a blue button labeled "Login" and a blue link labeled "Forgot Your Password?".

Рисунок 3.3 – Форма для авторизації

Ця форма була реалізована за допомогою наступного коду:

```
public function loginProcess(Request $request)
{
    try
    {
        $messages = [
            'email.required' ==> "Ведіть пошту"
            'email.email' ==> "Помилка при введенні пошти"
            'password.required' ==> "Ведіть пароль"
            'password.between' ==> "мінімальна довжина пароля - 8 символів"
        ];
        $this->validate($request, [
            'email' ==> 'required|email',
            'password' ==> 'required|between:8, 100',
        ], $messages)
        $errors = "Неправильний логін або пароль.";
        return Redirect::back()
            ->withInput()
            ->withErrors()
    }
}
```

Далі було реалізована можливість взаємодіяти з документами в електронному вигляді. Тобто у вигляді таблиць. На рисунку 3.4 представлено візуальний довідник.

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Клей	Юлія	976893928	alkes234@gmail.com	1500
Макасон	тендер	25	1235	Клей	Максім	315841813	mafjghb@gmail.com	200000
Коля	тендер	13	2540	Устілка	Власюк	065451565	fasfafs@gmail.com	200000

Рисунок 3.4 – Довідник клієнтів

Довідник клієнтів був реалізований за допомогою наступного коду:

```

@foreach($list as $item)
    <tr>
        <td >{{ $item->name}}</td>
        <td >{{ $item->Directory_categorys['category_name']}}</td> <!-- связь для вывода
в таблиці назви категории -->
        <td >{{ $item->discount}}</td>
        <td >{{ $item->limit_sales}}</td>
        <td >{{ $item->City_directorys['city']}}</td>
        <td >{{ $item->contact_person}}</td>
        <td >{{ $item->phone}}</td>
        <td >{{ $item->email}}</td>
        <td >{{ $item->incoming_debt}}</td>

        <td><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{route('deleteclient',$item->id)}}">Видалити</a></td>
        <td><a class='update' href="{{route('editclient',$item->id)}}">Змінити</a></td>
    </tr>
@endforeach

```

Також було реалізовано можливість додавати, видаляти та редагувати інформацію. Додавання інформації було реалізовано за допомогою наступного коду:

```

public function addclient(Request $request)
{
    $this->create([
        'name'=>$request->input('name'),
        'directory_categorie_id'=>$request->input('categoriesselect'),
        'discount'=>$request->input('discount'),
        'limit_sales'=>$request->input('limit_sales'),
        'city_directorie_id'=>$request->input('cityselect'),
        'contact_person'=>$request->input('contact_person'),
        'phone'=>$request->input('phone'),
        'email'=>$request->input('email'),
        'incoming_debt'=>$request->input('incoming_debt'),
    ]);
}

```

Візуальний вигляд форми для додавання інформації про клієнта зображена на рисунку 3.5.

Клієнт

Категорія

Знижка

Ліміт на реалізацію

Контактна особа

Телефон

Електронна адреса

Вхідний борг на початку року

Рисунок 3.5 – Форма для додавання інформації про клієнта

Візуальний вигляд форми для редагування інформації зображений на рисунку 3.6:

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10
Максим	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0

Рисунок 3.6 – Результат успішного виконання форми для додавання

Можливість видалення інформації було реалізовано за допомогою наступного коду:

```
public function deleteclient($id)
{
    $deleteclient = $this->find($id);
    $deleteclient->delete();
}
```

Можливість редагування інформації було реалізовано за допомогою наступного коду:

```
public function updateclient(Request $request, $id)
{
    $updateclient = $this->find($id);
    $updateclient->name = $request->get('name');
    $updateclient->directory_categorie_id = $request->get('categoriesselectedit');
    $updateclient->discount = $request->get('discount');
    $updateclient->limit_sales = $request->get('limit_sales');
    $updateclient->city_directorie_id = $request->get('cityselectedit');
    $updateclient->contact_person = $request->get('contact_person');
    $updateclient->phone = $request->get('phone');
    $updateclient->email = $request->get('email');
    $updateclient->incoming_debt = $request->get('incoming_debt');
    $updateclient->save();
}
```

Клієнт

▾

Знижка

Ліміт на реалізацію

▾

Контактна особа

Телефон

Електронна адреса

Вхідний борг на початку року

Рисунок 3.7 – Форма для редагування інформації про клієнта

Візуальний вигляд форми для редагування інформації зображений на рисунку 3.7.

Щоб переконатись, що форма працює, було змінено ім'я клієнта. Результат зображений на рисунку 3.8:

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10
Артем	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0

Рисунок 3.8 – Результат успішного виконання форми для редагування

Також було реалізовано ряд інших електронних документів у вигляді таблиць та з можливістю додавати, видаляти та редагувати інформацію, яка зберігається у них. Далі представлено їх візуальних вигляд та код за допомогою якого вони були реалізовані.

Довідник замовлень

Дата	Номер документа	Клієнт	Категорія	Кількість пар	Примітка
2021-05-19	36	Самчинська	тренер	56	нічого
2019-06-10	12	Макасон	тендер	32	нічого
2019-06-15	13	Макасон	тренер	12	нічого

Рисунок 3.9 – Довідник замовлень

Довідника замовлень був реалізований за допомогою наступного коду:

```

@foreach($list as $item)
    <tr>
        <td >{{$item->date}}</td>
        <td >{{$item->number_document}}</td>
        <td >{{$item->Clients['name']}}</td>
        <td >{{$item->Directory_categorys['category_name']}}</td>
        <td >{{$item->number_pairs}}</td>
        <td >{{$item->note}}</td>

        <td><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{route('deleteorder',$item->id)}}">Видалити</a></td>
        <td><a class='update' href="{{route('editororder',$item->id)}}">Змінити</a></td>
    </tr>
@endforeach

```

[+ Додати](#)

Видаткова накладна

Дата	Назва документу	Клієнт	Категорія	Номер документу	Кількість пар	Сума видатку	Номер замовлення	Кількість пар по замовленню	Примітка
2018-06-10	Платіжне доручення	Самчинська	тендер	11	10	5410	1	50	нічого
2009-06-10	Видаткова накладна	Коля	тренер	18	40	25040	3	40	нічого

Рисунок 3.10 – Видаткова накладна

Видаткова накладна була реалізована за допомогою наступного коду:

```

@foreach($list as $item)
    <tr>
        <td >{{$item->date}}</td>
        <td> {{$item->Document_operations['name_documents']}}</td>
        <td> {{$item->Clients['name']}}</td>
        <td> {{$item->Directory_categorys['category_name']}}
        <td >{{$item->number_document}}</td>
        <td >{{$item->number_pairs}}</td>
        <td >{{$item->sum_expenditure}}</td>
        <td >{{$item->number_order}}</td>
        <td >{{$item->number_pairs_order}}</td>
        <td >{{$item->note}}</td>
        <td><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{route('deleteinvoice',$item->id)}}">Видалити</a></td>
        <td><a class='update' href="{{route('editinvoice',$item->id)}}">Змінити</a></td>
    </tr>
@endforeach

```

[+ Додати](#)

Довідник надходжень

Дата	Назва документа	Назва	Клієнт	Категорія	Сума оплати	Примітка
2017-06-10	Видаткова накладна	Приват	Самчинська	тендер	250220	нічого

Рисунок 3.11 – Довідник надходжень

Видаткова накладна була реалізована за допомогою наступного коду:

```
@foreach($list as $item)
    <tr>
        <td >{{ $item->date }}</td>
        <td >{{ $item->Document_operations['name_documents'] }}</td>
        <td >{{ $item->Profit_accounts['full_name'] }}</td>
        <td >{{ $item->Clients['name'] }}</td>
        <td >{{ $item->Directory_categorys['category_name'] }}</td>
        <td >{{ $item->payment_amount }}</td>
        <td >{{ $item->note }}</td>

        <td ><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{ route('deleterecipt', $item->id) }}">Видалити</a></td>
        <td ><a class='update' href="{{ route('editreceipt', $item->id) }}"
">Змінити</a></td>
    </tr>
@endforeach
```

Далі було реалізовано склад для зберігання матеріалів на виробництво. На рисунку зображений візуальний вигляд складу:

[+ Додати](#)

Склад матеріалів

Назва	Розмір	Кількість	Дія	Дія
Устілка	36	5	Видалити	Змінити
Клей	маленький	5	Видалити	Змінити

Рисунок 3.12– Склад матеріалів

Склад матеріалів був реалізований за допомогою наступного коду:

```
@foreach($list as $item)
  <tr>
    <td >{{ $item->city}}</td>
    <td >{{ $item->region}}</td>
    <td >{{ $item->country}}</td>
    <td><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{route('deletecity',$item->id)}}">Видалити</a></td>
    <td><a class='update' href="{{route('editcity',$item->id)}}">Змінити</a></td>
  </tr>
@endforeach
```

Також було реалізовано роботу з таблицею надходжень матеріалів на склад. Цей документ зберігає інформацію про те, коли та звідки прийшов матеріал на склад. На рисунку 3.13 зображено візуальний вигляд:

Довідник надходжень

Дата	Постачальник	Назва матеріалу	Розмір	Кількість	Сума оплати	Примітка
2017-06-10	Club Dance	Підшва	25	125	40000	нічого

Рисунок 3.13 – Довідник надходжень матеріалів

Довідник був реалізований за допомогою наступного коду:

```
@foreach($list as $item)
  <tr>
    <td >{{ $item->date}}</td>
    <td >{{ $item->Document_operations['name_documents']}}</td>
    <td >{{ $item->Profit_accounts['full_name']}}</td>
    <td >{{ $item->Clients['name']}}</td>
    <td >{{ $item->Directory_categorys['category_name']}}</td>
    <td >{{ $item->payment_amount}}</td>
    <td >{{ $item->note}}</td>
    <td><a class='delete' onclick="return confirm('Do you really want to delete
it?');" href="{{route('deleterecipt',$item->id)}}">Видалити</a></td>
    <td><a class='update' href="{{route('editreceipt',$item->id)}}">
">Змінити</a></td>
  </tr>
@endforeach
```

Таким чином, було розглянуто основні можливості даного програмного продукту та їх реалізацію за допомогою програмного коду

3.3 Тестування інформаційної системи

Тестування займає важливе місце у будь-якій розробці. Саме на цьому етапі є можливість протестувати кожен функцію та метод, який є в програмному продукті та якщо є нюанси та помилки, то допрацювати їх. В даному проекті було проведено декілька тестувань. Перше тестування проводиться по додаванню певної інформації у таблицю та коректного запису у БД. Тестування проводиться у наступному порядку:

- Для початку користувачу потрібно перейти до роботи з таблицею “Довідник клієнтів”
- Далі перейти до додавання нового клієнта та заповнити відповідну форму
- Після заповнення, форма відправляється на сервер та проходить обробку. Після чого інформація записується до БД. Наступний код відповідає за реалізацію:

```
public function addclient(Request $request)
{
    $this->create([
        'name'=>$request->input('name'),
        'directory_categorie_id'=>$request->input('categoriesselect'),
        'discount'=>$request->input('discount'),
        'limit_sales'=>$request->input('limit_sales'),
        'city_directorie_id'=>$request->input('cityselect'),
        'contact_person'=>$request->input('contact_person'),
        'phone'=>$request->input('phone'),
        'email'=>$request->input('email'),
        'incoming_debt'=>$request->input('incoming_debt'),
    ]);
}
```

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Скаржинці	Юлія	976893928	alkes234@gmail.com	1500

Рисунок 3.14 – Результат

Другий тест повинен перевірити коректність функції редагування інформації та успішне зберігання її в БД. Для цього потрібно виконати наступні кроки:

- Потрібно перейти до роботи з таблицею “Довідник клієнтів”. Після чого перейти на форму для редагування
- Далі потрібно замінити необхідні інформацію. Для прикладу змінено місто Скаржинці на Київ. Після чого потрібно підтвердити дане редагування.
- Після підтвердження, форма відправляється на сервер, де обробляється та вносить зміни у БД. Приклад роботи та результат зображені на рисунках 3.15 та 3.16

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Скаржинці	Юлія	976893928	alkes234@gmail.com	1500

Рисунок 3.15 – Початкова інформація

Код який відповідає за реалізацію можливості редагування:

```
public function updateclient(Request $request, $id)
{
    $updateclient = $this->find($id);
    $updateclient->name = $request->get('name');
    $updateclient->directory_categorie_id = $request->get('categoriesselectedit');
    $updateclient->discount = $request->get('discount');
    $updateclient->limit_sales = $request->get('limit_sales');
    $updateclient->city_directorie_id = $request->get('cityselectedit');
    $updateclient->contact_person = $request->get('contact_person');
    $updateclient->phone = $request->get('phone');
    $updateclient->email = $request->get('email');
    $updateclient->incoming_debt = $request->get('incoming_debt');
    $updateclient->save();
}
```

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500

Рисунок 3.16 – Результат виконання функції редагування

Третій тест повинен перевірити коректність функції для видалення інформації. Для цього потрібно виконати наступні кроки:

- Потрібно перейти до роботи з таблицею “Довідник клієнтів”. Після чого обрати необхідну інформацію та підтвердити видалення

- Після цього, запит буде успішно виконаний та інформація, яка зберігалась в БД буде змінена і видалена та інформація, яка була обрана. Приклад реалізації зображений на рисунках 3.17 та 3.18

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000
Коля	тендер	13	2540	Скаржинці	Власюк	065451565	fasfafs@gmail.com	200000
Едік	тренер	15	2500	Хмельницький	Муляр	+3055402280	makmkaf@laafslm.com	150000

Рисунок 3.17 – Початкова інформація

Код який відповідає за можливість видалення інформації:

```
public function deleteclient($id)
{
    $deleteclient = $this->find($id);
    $deleteclient->delete();
}
```

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000
Коля	тендер	13	2540	Скаржинці	Власюк	065451565	fasfafs@gmail.com	200000

Рисунок 3.18 – Результат виконання функції видалення

Далі потрібно провести Test Case для того, щоб переконатись у правильності функціонування додатку. В таблиці 3.1 та 3.2 проведені наступні тести:

Таблиця 3.1 – Результат виконання тесту “Перевірка відображення головної сторінка після проходження авторизації”

Перевірка відображення головної сторінка після проходження авторизації		
Дія	Очікуваний результат	Результат тесту
Відкрити сторінку “Вхід”	<ul style="list-style-type: none"> – Сторінка “Вхід” відкрита – На формі 2 поля: Ім’я та пароль – Назва форми “Вхід” зображена по центрі у верхньому куті – Кнопка “Вхід” доступна – Поле з гіперпосиланням доступне та знаходиться внизу під формою 	Головна сторінка після проходження авторизація успішно відобразилась

Таблиця 3.2 – Результат виконання тесту “Перевірка на коректність вводу помилки при заповненні поля типу “String” числом на формі “Додавання міста”

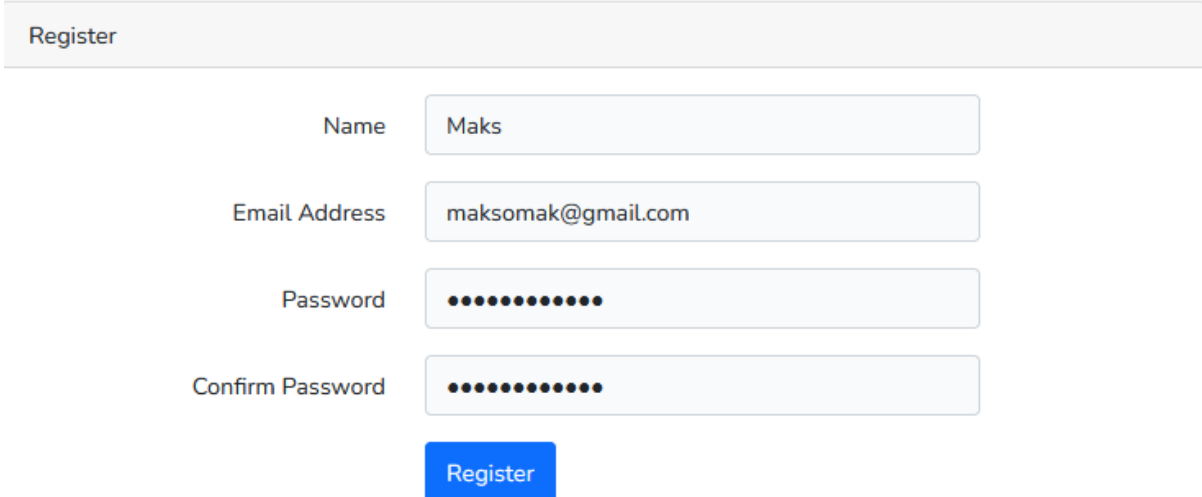
Перевірка на коректність вводу помилки при заповненні поля типу “String” числом на формі “Додавання міста”		
Дія	Очікуваний результат	Результат тесту
Відкрити форму “Додавання міста”	<ul style="list-style-type: none"> – Сторінка “Додавання міста” відкрита – На формі поля: назва міста, області, країни” – Вводимо число в одне з полів зі значенням “String” 	Помилка через некоректне заповнення поля типу “String”

Можна зробити висновок про те, що програмний продукт пройшов успішно всі види тестування.

3.4 Інструкція користувача

Для початку, користувачу потрібно пройти авторизацію або якщо користувача нема в БД, то потрібно пройти реєстрацію для того, щоб мати змогу користуватись можливостями продукту.

На рисунку 3.19 зображено приклад заповнення форми для реєстрації



The image shows a registration form with the following fields and values:

Field Label	Value
Name	Maks
Email Address	maksomak@gmail.com
Password	••••••••••
Confirm Password	••••••••••

Below the fields is a blue button labeled "Register".

Рисунок 3.19 – Приклад заповнення форми для реєстрації

Після того, як користувач заповнив усі поля, потрібно натиснути на кнопку “Register”. Після чого, користувача буде добавлено в БД та переадресує на головну сторінку.

Якщо користувач уже зареєстрований, перед тим як користуватись можливостями продукту, потрібно пройти авторизацію. На рисунку 3.20 зображено приклад заповнення форми для авторизації

Login

Email Address

Password

Remember Me

[Forgot Your Password?](#)

Рисунок 3.20 – Приклад заповнення форми для авторизації

Після успішного проходження реєстрації та авторизації, користувача переадресовує на головну сторінку яка зображена на рисунку 3.21.

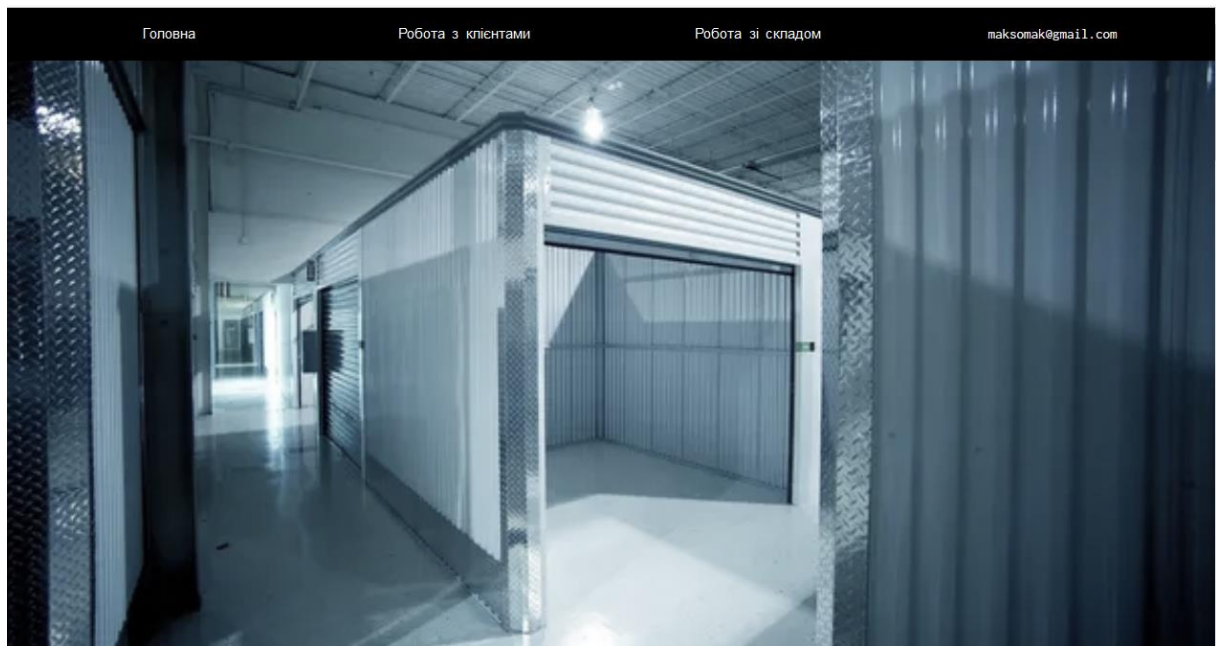


Рисунок 3.21 – Головна сторінка

На головній сторінці у користувача є дві категорії роботи:

1. Робота з документами клієнтів
2. Робота з документами по складу

Розглянемо роботу з документами клієнтів. Для цього користувачу потрібно перейти у розділ “Робота з клієнтами”. Після цього, користувача

переадресує на сторінки для роботи з відповідними документами. На рисунку 3.22 зображено сторінку для роботи з документами клієнтів.

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року	Видалення	Редагування
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500	Видалити	Змінити
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000	Видалити	Змінити
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000	Видалити	Змінити
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10	Видалити	Змінити
Артем	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0	Видалити	Змінити

Рисунок 3.22 – Візуальний вигляд сторінки “Робота з клієнтами”

Користувач має змогу працювати з багатьма видами документів. Для прикладу, розглянемо роботу з довідником клієнтів. В даному документі є ряд важливих полів з інформацією про клієнта. У користувача є можливість взаємодіяти з цим документом наступним чином: додавати інформацію, редагувати інформацію, видаляти інформацію.

Для додавання інформації у довідник користувачу потрібно натиснути на кнопку “Добавити”. Після чого, користувача буде переадресовано на форму для додавання інформації. Приклад зображено на рисунку 3.23:

Клієнт

Категорія
тренер ▾

Знижка

Ліміт на реалізацію

Хмельницький ▾

Контактна особа

Телефон

Електронна адреса

Вхідний борг на початку року

Добавить

Рисунок 3.23 – Візуальний вигляд форми для додавання

Далі користувачу потрібно заповнити інформацію у відповідних полях.
Приклад зображений на рисунку 3.24

Клієнт

Категорія

Знижка

Ліміт на реалізацію

Контактна особа

Телефон

Електронна адреса

Вхідний борг на початку року

Рисунок 3.24 – Приклад заповнення форми

Після того, як користувач успішно заповнив форму, інформація зберігається у БД і надалі буде відображатись у довіднику клієнтів. Приклад зображений на рисунку 3.25:

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року	Видалення	Редагування
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500	Видалити	Змінити
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000	Видалити	Змінити
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000	Видалити	Змінити
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10	Видалити	Змінити
Артем	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0	Видалити	Змінити
Аня	Покупець	15	2500	Хмельницький	Запорожець	0976585122	ana_4949@gmail.com	0	Видалити	Змінити

Рисунок 3.25 – Результат виконання додавання

Далі, у користувача є можливість редагувати інформацію. Для цього у кожній строці є текст з посиланням на форму для редагування. На рисунку 3.26 зображено форму для редагування:

Клієнт

 ▾

Знижка

Ліміт на реалізацію

▾

Контактна особа

Телефон

Електронна адреса

Вхідний борг на початку року

Рисунок 3.26 – Форма для редагування

У цій формі користувачу надається всі інформація строки, яку він обрав та є можливість змінити будь-яке поле. Для прикладу було змінено ім'я клієнта на Іван і результат зображено на рисунку 3.27:

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року	Видалення	Редагування
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500	Видалити	Змінити
Макасон	тендер	25	1235	Київ	Максим	315841813	mafjghb@gmail.com	200000	Видалити	Змінити
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000	Видалити	Змінити
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10	Видалити	Змінити
Артем	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0	Видалити	Змінити
Іван	Покупець	15	2500	Хмельницький	Запорожець	0976585122	ana_4949@gmail.com	0	Видалити	Змінити

Рисунок 3.27 – Результат виконання форми для редагування

І останнє – це видалення інформації. Тобто, якщо користувачу потрібно видалити будь-яку строку з інформацією про клієнта, потрібно натиснути на “Видалити” після чого інформація буде видалена з БД та таблиці. Результат виконання – це видалення клієнта Іван і результат виконання зображено на рисунку 3.28:

Довідник клієнтів

Клієнт	Категорія	Знижка	Ліміт на реалізацію	Місто	Контактна особа	Телефон	Електронна адреса	Вхідний борг на початку року	Видалення	Редагування
Самчинська	тренер	0	0	Київ	Юлія	976893928	alkes234@gmail.com	1500	Видалити	Змінити
Макасон	тендер	25	1235	Київ	Максім	315841813	mafjghb@gmail.com	200000	Видалити	Змінити
Коля	тендер	13	2540	Хмельницький	Власюк	065451565	fasfafs@gmail.com	200000	Видалити	Змінити
Алекс	Покупець	25	24	Хмельницький	10	097645231	andrey@gmail.com	10	Видалити	Змінити
Артем	Покупець	20	2500	Хмельницький	Запорожець	097651235	maksimka228@gmail.com	0	Видалити	Змінити

Рисунок 3.28 – Результат виконання видалення інформації

Далі на рисунках 3.29-3.32 зображено візуальний вигляд інших документів.

+ Додати

Довідник надходжень

Дата	Назва документа	Назва	Клієнт	Категорія	Сума оплати	Примітка
2017-06-10	Видаткова накладна	Приват	Самчинська	тендер	250220	нічого

Рисунок 3.29 – Візуальний вигляд документа “Довідник надходжень”

Довідник замовлень

Дата	Номер документа	Клієнт	Категорія	Кількість пар	Примітка
2021-05-19	36	Самчинська	тренер	56	нічого
2019-06-10	12	Макасон	тендер	32	нічого
2019-06-15	13	Макасон	тренер	12	нічого

Рисунок 3.30 – Візуальний вигляд документа “Довідник замовлень”

[+ Додати](#)

Видаткова накладна

Дата	Назва документу	Клієнт	Категорія	Номер документу	Кількість пар	Сума видатку	Номер замовлення	Кількість пар по замовленню	Примітка
2018-06-10	Платіжне доручення	Самчинська	тендер	11	10	5410	1	50	нічого
2009-06-10	Видаткова накладна	Коля	тренер	18	40	25040	3	40	нічого

Рисунок 3.31 – Візуальний вигляд документа “Видаткова накладна”

[+ Додати](#)

Довідник операцій

Документи по операціям	Видалення	Редагування
Платіжне доручення	Видалити	Змінити

Рисунок 3.32 – Візуальний вигляд документа “Довідник операцій”

Як у випадку з довідником клієнтів, так і з іншими, у користувача є можливість додавати інформацію, видаляти інформацію, редагувати інформацію. Це все, що стосується розділу “Робота з клієнтами”.

Далі у користувача є можливість роботи з документами складу. Для цього потрібно натиснути на розділ “Робота з документами по складу”. Після чого, користувача переадресує на сторінку. Для роботи зі складом у користувача є наступні документи, які зображені на рисунку 3.33:

Таблиці>>>	Матеріали	Довідник надходжень	Довідник постачальників	Готове взуття	<<< Таблиці
------------	-----------	---------------------	-------------------------	---------------	-------------

Рисунок 3.33 – Види документів для роботи зі складом

На рисунках 3.34-3.36 зображено візуальний вигляд документів.

[+ Додати](#)

Склад матеріалів

Назва	Розмір	Кількість	Дія	Дія
Устілка	36	5	Видалити	Змінити
Клей	маленький	5	Видалити	Змінити

Рисунок 3.34 – Візуальний вигляд документа “Склад матеріалів”

Довідник надходжень

Дата	Постачальник	Назва матеріалу	Розмір	Кількість	Сума оплати	Примітка
2017-06-10	Club Dance	Підшва	25	125	40000	нічого

Рисунок 3.35 – Візуальний вигляд документа “Довідник надходжень”

Довідник постачальників

Назва	Видалення	Редагування
Fidellity	Видалити	Змінити
Aticco	Видалити	Змінити
ТопіТоп	Видалити	Змінити

Рисунок 3.36 – Візуальний вигляд документа “Довідник постачальників”

У кожній таблиці користувач має можливість проводити наступні дії: додавати інформацію, редагувати інформацію та видаляти інформацію.

Таким чином було розглянути основні моменти даного програмного продукту та дії, які має можливість проводити користувач.

3.5 Вимоги до розгортання інформаційної системи

Для коректного розгортання та функціонування необхідно:

- Браузер Google Chrome, від версії 91 – поточна версія;
- ІІS, від версії 7.0 – поточна версія;
- Windows Server, від версії 2012 – поточна версія;
- Microsoft SQL Server, від версії 2012 – поточна версія;
- Дисківий простір – не менше 1000 Мб;
- Composer – поточна версія;
- Laravel – поточна версія;
- Open Server – поточна версія;

Висновки

В результаті виконання кваліфікаційної роботи бакалавра було розроблено інформаційну технологію підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві, а також було спроектовано та розроблено на платформі PHP відповідну автоматизовану інформаційну систему підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Для створення програмного продукту було використано мову програмування PHP, фреймворк Laravel, систему керування базами даних MySQL.

Система виконує усі необхідні функції. А саме:

- автоматизує ведення даних щодо потреб виробництва у визначених матеріалах для виготовлення визначених виробів;
- забезпечує формування планів виробництва;
- автоматизує визначення потреб виробництва у визначених матеріалах згідно сформованих планів виробництва;
- проводить реєстрацію надходжень матеріалів на склад виробництва та автоматизує визначення залишкових потреб для подальших надходжень у визначених матеріалах згідно сформованих планів виробництва;
- автоматизує визначення надлишкових запасів матеріалів на складі згідно сформованих планів виробництва;
- автоматизує ведення даних щодо використання визначених матеріалів для виготовлення визначених виробів та ведення даних щодо інших втрат і витрат матеріалів;
- автоматизує визначення очікуваних залишків матеріалів на складі згідно зареєстрованих втрат матеріалів і витрат матеріалів на виробництво для проведення ревізійної діяльності.

Програмний продукт пройшов тестування на коректність роботи та на основі результатів тестувань повністю відповідає поставленому завданню.

Перелік посилань

1. Що таке інформаційні технології. URL: <http://apeps.kpi.ua/shcho-take-informatsiini-technologii/en>
2. Основні положення систем підтримки прийняття рішень. Що таке прийняття рішень. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
3. Основні положення систем підтримки прийняття рішень. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
4. Основні положення систем підтримки прийняття рішень. Етап постановки завдання. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
5. Основні положення систем підтримки прийняття рішень. Етап формування рішень. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
6. Основні положення систем підтримки прийняття рішень. Ризик та визначеність URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
7. Основні положення систем підтримки прийняття рішень. Етап вибору рішення. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
8. Основні положення систем підтримки прийняття рішень. Класифікація СППР. URL: <https://lib.chmnu.edu.ua/pdf/posibnuku/313/3.pdf>
9. Виробництво: сутність, чинники та зміст. URL: https://ru.osvita.ua/vnz/reports/econom_theory/21552/
10. Підприємство. URL: <https://uk.wikipedia.org/wiki/Підприємство>
11. Виробництво та його фактори. Що таке праця? Що таке предмет праці? Що таке засоби праці? URL: <https://ukreferat.com/chapters/ekonomichna-teoriya/virobnitstvo-ta-jogo-faktori-referat.html>
12. Левченко О. П. Шляхи удосконалення організації обліку виробничих запасів. Збірник наукових праць Таврійського державного агротехнологічного університету (економічні науки). 2014. № 4. С. 266–270.
13. Ткаченко Н.М., Бухгалтерський (фінансовий) облік, оподаткування і звітність: Підручник. – 6-те вид. допов. і перероб. – К.:Алерта, 2013. – 982 с.

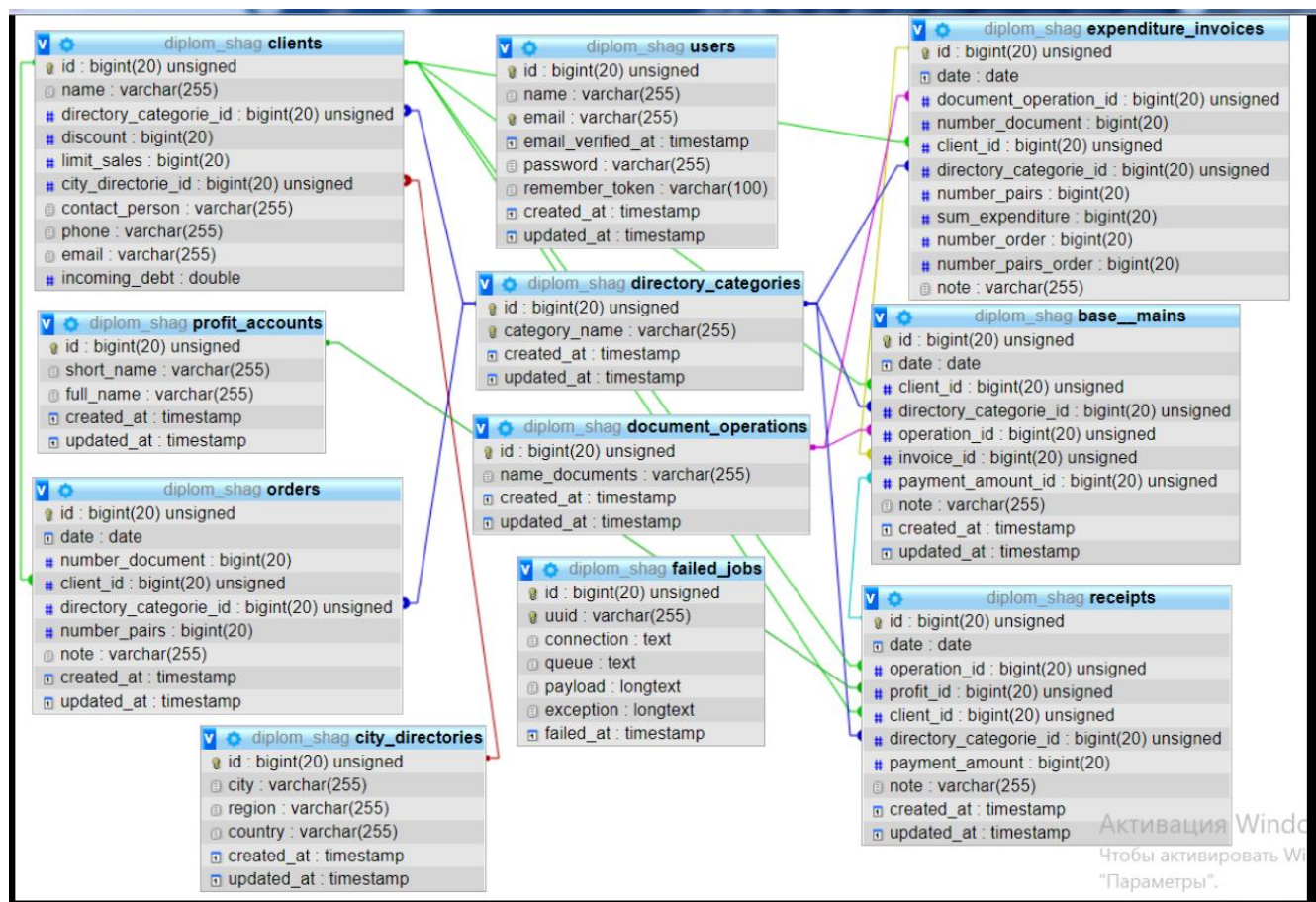
14. Що таке склад(будівля). URL:
[https://uk.wikipedia.org/wiki/Склад_\(будівля\)](https://uk.wikipedia.org/wiki/Склад_(будівля))
15. Накладна. URL:
<https://www.pharmencyclopedia.com.ua/article/1165/nakladna>
16. Інформаційні технології. [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Інформаційні_технології
17. Застосування комп'ютерних технологій в економічному аналізі запасів підприємства. URL: https://www.problecon.com/export_pdf/problems-of-economy-2017-1_0-pages-244_249.pdf
18. Автоматизація. URL: https://www.problecon.com/export_pdf/problems-of-economy-2017-1_0-pages-244_249.pdf
19. Багрій К.Л., Удосконалення класифікації матеріальних запасів у системі управління підприємством. Вісник Чернівецького торговельно-економічного інституту. Серія: Економічні науки. 2009. Вип. 3. С. 288–293.
20. Патерни для початківців: MVC vs MVP vs MVVM. URL:
<https://habr.com/ru/post/215605/>
21. MVVM. URL:
https://skillbox.ru/media/code/mvvm_proektirovanie_prilozheniy_dlya_windows/
22. Що таке MVP і як це використовувати. URL:
https://skillbox.ru/media/code/chto_takoe_mvp_i_kak_eto
23. Що таке БД? URL: <http://apeps.kpi.ua/shco-take-basa-danykh>
24. Dilovod. URL: <https://dilovod.ua/>
25. Мій склад. URL: <https://www.moysklad.ru/>
26. Операційна система. URL:
https://uk.wikipedia.org/wiki/Операційна_система
27. Веб-платформа. URL: <http://x-site.by/products/web-platform>
28. Web-додаток – переваги і недоліки. URL:
https://mydiv.net/arts/view-web-prilozhenija_preimuxhestva_i_nedostatki.html

29. Архітектура мобільної платформи 1С:Підприємство 8.3. URL: <https://tqm.com.ua/ua/likbez/1%D1%81-platform/arkhitektura-mobilnoyi-platformy-1s-pidpryyemstvo-8-3>
30. Чому PHP є лідером серед платформ розробки веб-додатків. URL: <https://msiter.ru/articles/pochemu-php-do-sih-por-yavlyaetsya-liderom-sredi-platform-razrabotki-veb-prilozheniy>
31. PHP переваги и недоліки. URL: <https://web-shpargalka.ru/php-dostoinstva-i-nedostatki.php>
32. FastReport .Mono. Що це і для чого. URL: <https://www.fastreport.ru/ru/blog/show/what-is-mono/>
33. Переваги Mono. URL: <https://www.fastreport.ru/ru/blog/show/what-is-mono/>
34. Java як язык и платформа. URL: <https://comaqa.gitbook.io/java-automation/java.-vvedenie/java-kak-yazyk-i-platforma>
35. Чому PHP до сих пор являється лідером среди платформ розробки веб-приложений. URL: <https://msiter.ru/articles/pochemu-php-do-sih-por-yavlyaetsya-liderom-sredi-platform-razrabotki-veb-prilozheniy>
36. Що таке фреймворк. URL: <https://secretmag.ru/enciklopediya/chto-takoe-freimvork-obyasnyаем-prostymi-slovami.htm>
37. Редактор початкового коду. URL: https://uk.wikipedia.org/wiki/Редактор_початкового_коду
38. Найкращі IDE та редактори коду для веб-розробки у 2021 році. URL: <https://techrocks.ru/2021/01/03/21-best-ide-and-code-editors-for-web-development/>
39. MySQL – система управління базами даних. URL: <https://web-creator.ru/articles/mysql>

ДОДАТКИ

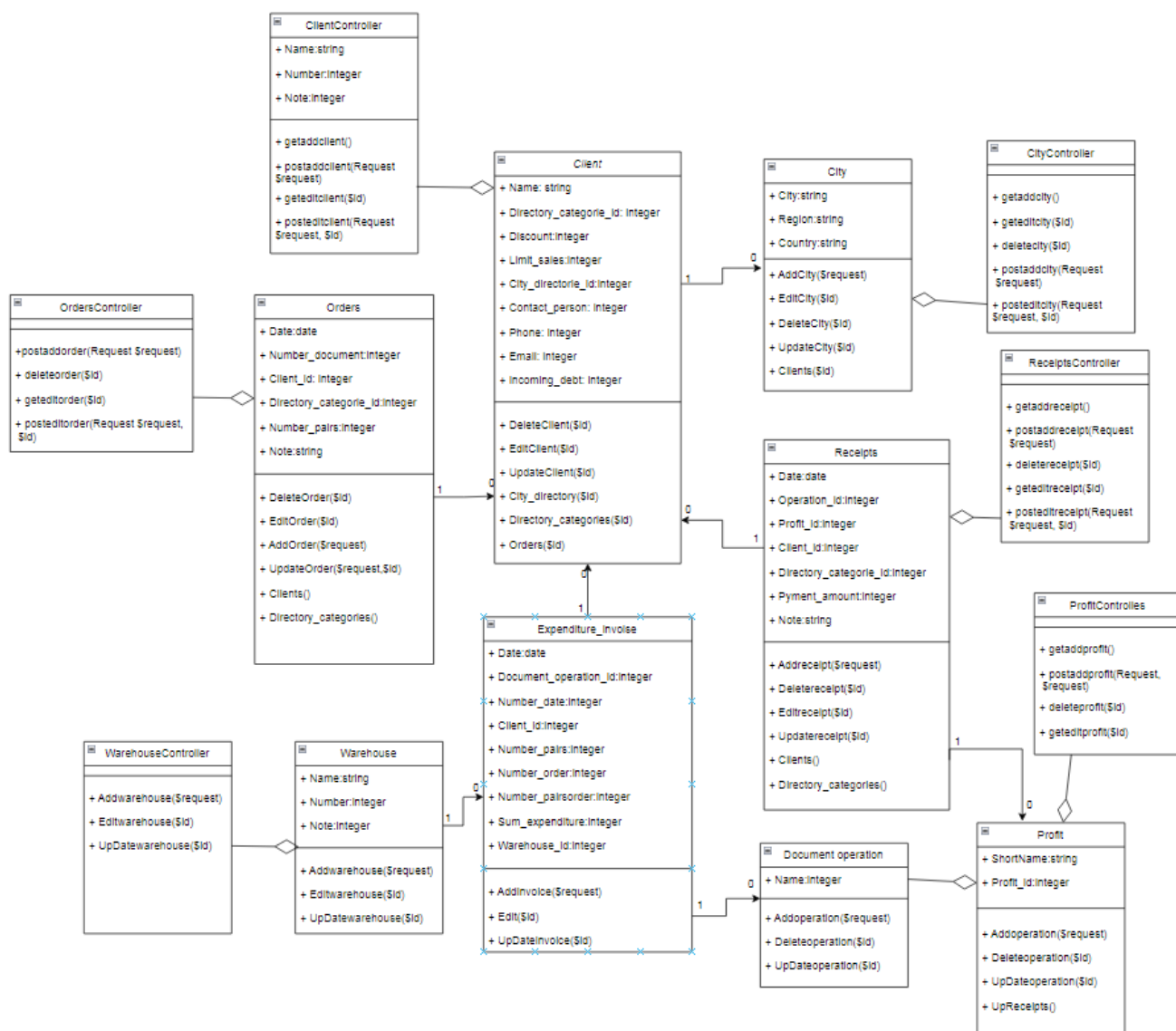
Додаток А

Структура бази даних



Додаток Б

Розгорнута структура класів



Додаток В

Програмні коди

Лістинг City_directory.php:

```
<?php
namespace App\Models;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
class City_directory extends Model
{
    use HasFactory;
    protected $table = "city_directories";
    *
    * @var array
    */
    protected $fillable = ['city', 'region', 'country'
];
    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }
    public function addcity(Request $request)
    {
        $this->create([
            'city'=>$request->input('city'),
            'region'=>$request->input('region'),
            'country'=>$request->input('country'),
        ]);
    }
    public function deletecity($id)
    {
        $deletecity = $this->find($id);
        $deletecity->delete();
    }
    public function editcity($id)
    {
        $mass = $this->where('id',$id)->get();
        return $mass;
    }
    public function updatecity(Request $request, $id)
    {
        $updatecity = $this->find($id);
        $updatecity->city = $request->get('city');
        $updatecity->region = $request->get('region');
        $updatecity->country = $request->get('country');
        $updatecity->save();
    }
    public function Clients()//для связи
    {
        return $this->hasMany(Client::class,'city_directorie_id','id'); //для связи
    }
}
```

Лістинг Client.php:

```
<?php
namespace App\Models;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Client extends Model
{
    use HasFactory;

    protected $fillable = [ //поля з якими можна щось робити
        'name', 'directory_categorie_id', 'discount','limit_sales',
        'city_directorie_id','contact_person','phone','email','incoming_debt'
    ];
    public $timestamps = false;
    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }
    public function addclient(Request $request)
    {

```

```

$this->create([
    'name'=>$request->input('name'),
    'directory_categorie_id'=>$request->input('categoriesselect'),
    'discount'=>$request->input('discount'),
    'limit_sales'=>$request->input('limit_sales'),
    'city_directorie_id'=>$request->input('cityselect'),
    'contact_person'=>$request->input('contact_person'),
    'phone'=>$request->input('phone'),
    'email'=>$request->input('email'),
    'incoming_debt'=>$request->input('incoming_debt'),
]);
}

public function deleteclient($id)
{
    $deleteclient = $this->find($id);
    $deleteclient->delete();
}

public function editclient($id)
{
    $mass = $this->where('id',$id)->get();
    return $mass;
}

public function updateclient(Request $request, $id)
{
    $updateclient = $this->find($id);
    $updateclient->name = $request->get('name');
    $updateclient->directory_categorie_id = $request->get('categoriesselectedit');
    $updateclient->discount = $request->get('discount');
    $updateclient->limit_sales = $request->get('limit_sales');
    $updateclient->city_directorie_id = $request->get('cityselectedit');
    $updateclient->contact_person = $request->get('contact_person');
    $updateclient->phone = $request->get('phone');
    $updateclient->email = $request->get('email');
    $updateclient->incoming_debt = $request->get('incoming_debt');
    $updateclient->save();
}

// public function clientselect($id)
// {
//     $clientselect = $this->where('id', $id)->get();
//     return $clientselect;
// }

public function City_directorys() //для связи
{
    return $this->belongsTo(City_directory::class,'city_directorie_id','id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Directory_categorys() //для связи
{
    return $this->belongsTo(Directory_category::class,'directory_categorie_id','id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Orders()//для связи
{

return $this->hasMany(Order::class,'client_id','id'); //для связиclient_id
// return $this->hasMany(Order::class,'directory_categorie_id','id'); //для связи
}

public function Expenditure_invoices()//для связи
{
return $this->hasMany(Expenditure_invoice::class,'client_id','id'); //для связиclient_id
//для связи
}
public function Receipts() //для связи
{
    return $this->hasMany(Receipt::class,'client_id','id'); //по якому полю связь буде
}

public function Main_Bases() //для связи
{
    return $this->hasMany(Base_Main::class,'client_id','id'); //по якому полю связь буде
}
}

```

Лістинг Directory_Categories.php:

```

<?php
namespace App\Models;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

```

```

class Directory_category extends Model
{
    use HasFactory;

    protected $fillable = [ //поля з якими можна щось робити
        'category_name'
    ];

    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }

    public function addcategorie(Request $request)
    {
        $this->create([
            'category_name'=>$request->input('category_name')
        ]);
    }
    public function deletecategorie($id)
    {
        $deletecategorie = $this->find($id);
        $deletecategorie->delete();
    }
    public function editcategorie($id)
    {
        $mass = $this->where('id',$id)->get();
        return $mass;
    }
    public function updatecategorie(Request $request, $id)
    {
        $updatecategorie = $this->find($id);
        $updatecategorie ->category_name = $request->get('category_name');

        $updatecategorie ->save();
    }

    public function Clients()//для зв'язи
    {
        return $this->hasMany(Client::class,'directory_category_id','id'); //для зв'язи
    }
    public function Orders()//для зв'язи
    {
        return $this->hasMany(Order::class,'directory_category_id','id'); //для зв'язи
        // return $this->hasMany(Order::class,'directory_category_id','id'); //для зв'язи
    }
    public function Expenditure_invoices()//для зв'язи
    {
        return $this->hasMany(Expenditure_invoice::class,'directory_category_id','id'); //для зв'язи
        // return $this->hasMany(Order::class,'directory_category_id','id'); //для зв'язи
    }
    public function Receipts()//для зв'язи
    {
        return $this->hasMany(Receipt::class,'directory_category_id','id');
    }
}
}

```

Лістинг Document_operations.php:

```

<?php

namespace App\Models;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Document_operation extends Model
{
    use HasFactory;

    protected $fillable = [ //поля з якими можна щось робити
        'name_documents'
    ];

    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }

    public function addoperation(Request $request)

```

```

{
    $this->create([
        'name_documents'=>$request->input('name_documents'),
    ]);
}

public function deleteoperation($id)
{
    $deleteoperation = $this->find($id);
    $deleteoperation->delete();
}

public function editoperation($id)
{
    $mass = $this->where('id',$id)->get();
    return $mass;
}

public function updateoperation(Request $request, $id)
{
    $updateoperation = $this->find($id);
    $updateoperation ->name_documents = $request->get('name_documents');

    $updateoperation ->save();
}

public function Expenditure_invoices()//для связи
{
    return $this->hasMany(Expenditure_invoice::class,'document_operation_id','id'); //для связисclient_id
    // return $this->hasMany(Order::class,'directory_categorie_id','id'); //для связи
}
public function Receipts()//для связи
{
    return $this->hasMany(Receipt::class,'document_operation_id','id'); //для связисclient_id
}
}
}

```

Лістинг Expenditure_operations.php:

```

<?php
namespace App\Models;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use Illuminate\Http\Request;

class Expenditure_invoice extends Model
{
    use HasFactory;
    protected $fillable = [ //поля з якими можна щось робити
        'date', 'document_operation_id', 'number_document','client_id',
        'directory_categorie_id','number_pairs','sum_expenditure','number_order','number_pairs_order', 'note'
    ];
    public $timestamps = false;
    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }
    public function addinvoice(Request $request)
    {
        $this->create([
            'date'=>$request->input('date'),
            'document_operation_id'=>$request->input('operationsselect'),
            'number_document'=>$request->input('number_document'),
            'client_id'=>$request->input('clientsselect'),
            'directory_categorie_id'=>$request->input('categoriesselect'),
            'number_pairs'=>$request->input('number_pairs'),
            'sum_expenditure'=>$request->input('sum_expenditure'),
            'number_order'=>$request->input('number_order'),
            'number_pairs_order'=>$request->input('number_pairs_order'),
            'note'=>$request->input('note')
        ]);
    }

    public function deleteinvoice($id)
    {
        $deleteinvoice = $this->find($id);
        $deleteinvoice->delete();
    }
    public function editinvoice($id)
    {
        $mass = $this->where('id',$id)->get();
        return $mass;
    }
}

```

```

public function updateinvoice(Request $request, $id)
{
    $updateinvoice = $this->find($id);
    $updateinvoice->date = $request->get('date');
    $updateinvoice->document_operation_id = $request->get('operationselectedit');
    $updateinvoice->number_document = $request->get('number_document');
    $updateinvoice->client_id = $request->get('clientselectedit');
    $updateinvoice->directory_categorie_id = $request->get('categoriesselectedit');
    $updateinvoice->number_pairs = $request->get('number_pairs');
    $updateinvoice->sum_expenditure = $request->get('sum_expenditure');
    $updateinvoice->number_order = $request->get('number_order');
    $updateinvoice->number_pairs_order = $request->get('number_pairs_order');
    $updateinvoice->note = $request->get('note');
    $updateinvoice->save();
}

public function Document_operations()
{
    return $this->belongsTo(Document_operation::class, 'document_operation_id', 'id');
}

public function Clients() //для связи
{
    return $this->belongsTo(Client::class, 'client_id', 'id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Directory_categorys() //для связи
{
    return $this->belongsTo(Directory_category::class, 'directory_categorie_id', 'id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Main_Bases()//для связи
{
    return $this->hasMany(Main_Base::class, 'invoice_id', 'id');
    //для связиclient_id
    // return $this->hasMany(Order::class, 'directory_categorie_id', 'id'); //для связи
}
}

```

Лістинг Order.php:

```

<?php

namespace App\Models;

use Directory;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Order extends Model
{
    use HasFactory;
    protected $fillable = [ //поля з якими можна щось робити
        'date', 'number_document', 'client_id',
        'directory_categorie_id', 'number_pairs', 'note'
    ];

    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }

    public function addorder(Request $request)
    {
        $this->create([
            'date'=>$request->input('date'),
            'number_document'=>$request->input('number_document'),
            'client_id'=>$request->input('clientsselect'),
            'directory_categorie_id'=>$request->input('categoriesselect'),
            'number_pairs'=>$request->input('number_pairs'),
            'note'=>$request->input('note'),
        ]);
    }

    public function deleteorder($id)
    {
        $deleteorder = $this->find($id);
        $deleteorder->delete();
    }

    public function editor($id)
    {
        $mass = $this->where('id', $id)->get();
    }
}

```

```

        return $mass;
    }

    public function updateorder(Request $request, $id)
    {
        $updateorder = $this->find($id);
        $updateorder ->date = $request->get('date');
        $updateorder ->number_document = $request->get('number_document');
        $updateorder ->client_id = $request->get('clientsselectedit');
        $updateorder ->directory_categorie_id = $request->get('categoriesselectedit');
        $updateorder ->number_pairs = $request->get('number_pairs');
        $updateorder ->note = $request->get('note');

        $updateorder ->save();
    }

    public function Clients() //для связи
    {
        return $this->belongsTo(Client::class,'client_id','id');
        // return $this->belongsTo(Client::class,'directory_categorie_id', 'id'); //по якому полю связь буде
    } //belongsTo - там де в нас є зовнішній ключ

    public function Directory_categorys() //для связи
    {
        return $this->belongsTo(Directory_category::class,'directory_categorie_id','id'); //по якому полю связь буде
    } //belongsTo - там де в нас є зовнішній ключ
}

```

Лістинг Profit_account.php:

```

<?php

namespace App\Models;

use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Profit_account extends Model
{
    use HasFactory;

    protected $fillable = [ //поля з якими можна щось робити
        'short_name','full_name'
    ];

    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }

    public function addprofit(Request $request)
    {
        $this->create([
            'short_name'=>$request->input('short_name'),
            'full_name'=>$request->input('full_name'),
        ]);
    }

    public function deleteprofit($id)
    {
        $deleteprofit = $this->find($id);
        $deleteprofit->delete();
    }

    public function editprofit($id)
    {
        $mass = $this->where('id',$id)->get();
        return $mass;
    }

    public function updateprofit(Request $request, $id)
    {
        $updateprofit = $this->find($id);
        $updateprofit ->short_name = $request->get('short_name');
        $updateprofit ->full_name = $request->get('full_name');
        $updateprofit ->save();
    }

    public function Receipts() //для связи
}

```

```

{
    return $this->hasMany(Receipt::class, 'profit_id', 'id'); //по якому полю зв'язь буде
}

```

Лістинг Receipt.php:

```

<?php

namespace App\Models;
use Illuminate\Http\Request;
use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;
use League\CommonMark\Block\Element\Document;

class Receipt extends Model
{
    use HasFactory;
    protected $fillable = [ //поля з якими можна щось робити
        'date', 'operation_id', 'profit_id',
        'client_id', 'directory_categorie_id', 'payment_amount', 'note'
    ];
    public $timestamps = false;
    public function listDate()
    {
        $listbd = $this->all();
        return $listbd; //достаємо дані з бд
    }

    public function addreceipt(Request $request)
    {
        $this->create([
            'date'=>$request->input('date'),
            'operation_id'=>$request->input('operationselect'),
            'profit_id'=>$request->input('profitselect'),
            'client_id'=>$request->input('clientsselect'),
            'directory_categorie_id'=>$request->input('categoriesselect'),
            'payment_amount'=>$request->input('payment_amount'),
            'note'=>$request->input('note')
        ]);
    }

    public function deletereceipt($id)
    {
        $deletereceipt = $this->find($id);
        $deletereceipt->delete();
    }

    public function editreceipt($id)
    {
        $mass = $this->where('id', $id)->get();
        return $mass;
    }

    public function updatereceipt(Request $request, $id)
    {
        $updatereceipt = $this->find($id);
        $updatereceipt ->date = $request->get('date');
        $updatereceipt ->operation_id = $request->get('operationsselectedit');
        $updatereceipt ->profit_id = $request->get('profitselectedit');
        $updatereceipt ->client_id = $request->get('clientsselectedit');
        $updatereceipt ->directory_categorie_id = $request->get('categoriesselectedit');
        $updatereceipt ->payment_amount = $request->get('payment_amount');
        $updatereceipt ->note = $request->get('note');

        $updatereceipt ->save();
    }

    public function Document_operations() //для зв'язи
    {
        return $this->belongsTo(Document_operation::class, 'operation_id', 'id');
    }

    public function Profit_accounts() //для зв'язи
    {
        return $this->belongsTo(Profit_account::class, 'profit_id', 'id');
    }

    public function Clients() //для зв'язи
    {
        return $this->belongsTo(Client::class, 'client_id', 'id');
    }
}

```

```
    // return $this->belongsTo(Client::class,'directory_categorie_id', 'id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Directory_categorys() //для связи
{
    return $this->belongsTo(Directory_category::class,'directory_categorie_id','id'); //по якому полю связь буде
} //belongsTo - там де в нас є зовнішній ключ

public function Main_Bases()//для связи
{
    return $this->hasMany(Main_Base::class,'payment_amount_id','id'); //для связиclient_id
    // return $this->hasMany(Order::class,'directory_categorie_id','id'); //для связи
}
}
```

Додаток Г
Презентаційний матеріал



КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Виконав: студент 4 курсу, група КН-18-1, М.В.Запорожець
Керівник: викладач кафедри КН, М.М. Молчанова



АКТУАЛЬНІСТЬ

Зважаючи на те, що наша країна зараз перебуває у стані війни, багато власників виробничих підприємств були змушені перенести виробничі потужності у інші більш безпечні регіони або тимчасово змінити профіль виробництва. Це в свою чергу потягнуло за собою певні проблеми з персоналом, матеріально-технічним забезпеченням та налагоджуванням безперебійного виробничого процесу. Іноді власникам самотужки, без допомоги кваліфікованих працівників, потрібно швидко аналізувати, планувати та приймати рішення. Тому створення відповідної інформаційної технології підтримки прийняття рішень забезпечить власнику вести належний контроль процесів та складських запасів на виробництві:

- За допомогою модуля, який використовуючи текстову та числову інформацію клієнта, за певним алгоритмом допоможе сформулювати план виробництва та визначити потреби у матеріалах для виготовлення необхідної продукції.
- За допомогою модулів, які виконують контроль руху запасів на складі та виробництві, визначаючи їхню наявність, надлишок або дефіцит, забезпечується вчасне постачання матеріалів на склад та у виробництво.

ЗАВДАННЯ

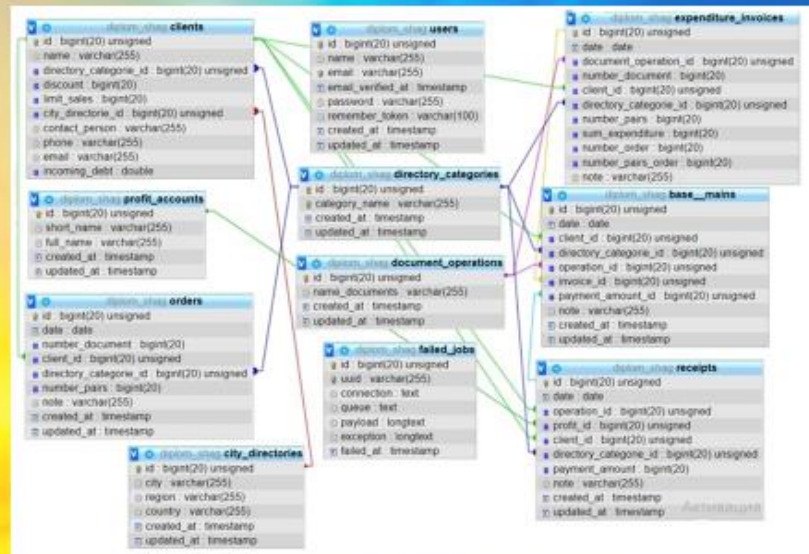
Мета кваліфікаційної роботи бакалавра полягає у розробці та практичній реалізації інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві, яка:

- автоматизує ведення даних щодо потреб виробництва у визначених матеріалах для виготовлення визначених виробів;
- забезпечує формування планів виробництва;
- автоматизує визначення потреб виробництва у визначених матеріалах згідно сформованих планів виробництва;
- проводить реєстрацію надходжень матеріалів на склад виробництва та автоматизує визначення залишкових потреб для подальших надходжень у визначених матеріалах згідно сформованих планів виробництва;

- автоматизує визначення надлишкових запасів матеріалів на складі згідно сформованих планів виробництва;
- автоматизує ведення даних щодо використання визначених матеріалів для виготовлення визначених виробів та ведення даних щодо інших втрат і витрат матеріалів;
- автоматизує визначення очікуваних залишків матеріалів на складі згідно зареєстрованих втрат матеріалів і витрат матеріалів на виробництво для проведення ревізійної діяльності.

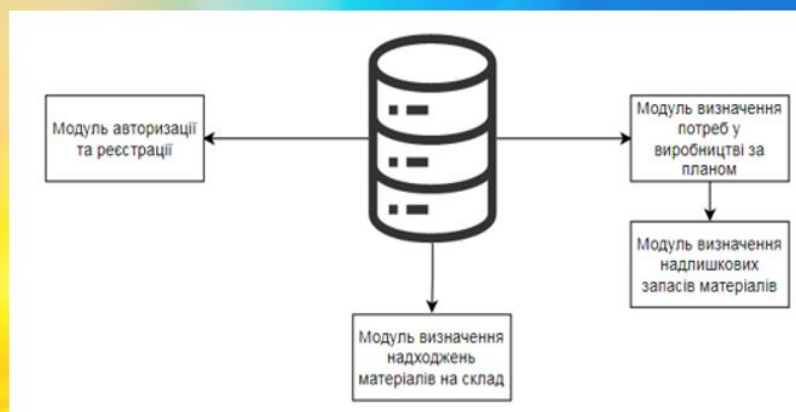


ДАТАЛОГІЧНА МОДЕЛЬ БАЗ ДАНИХ



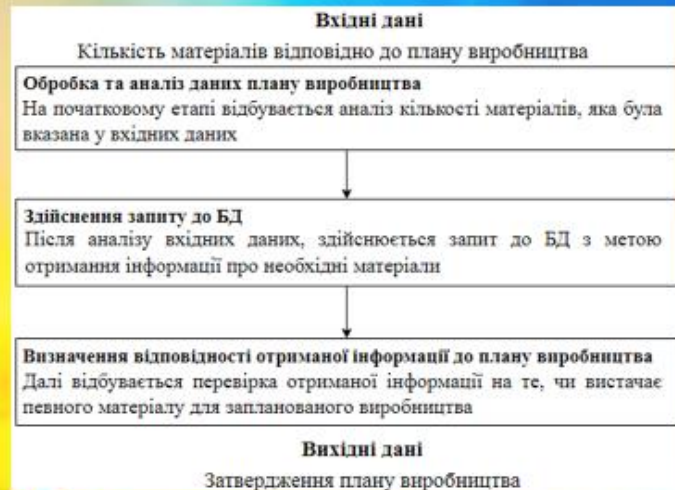
АРХІТЕКТУРА

Схема модулів інформаційної системи підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві



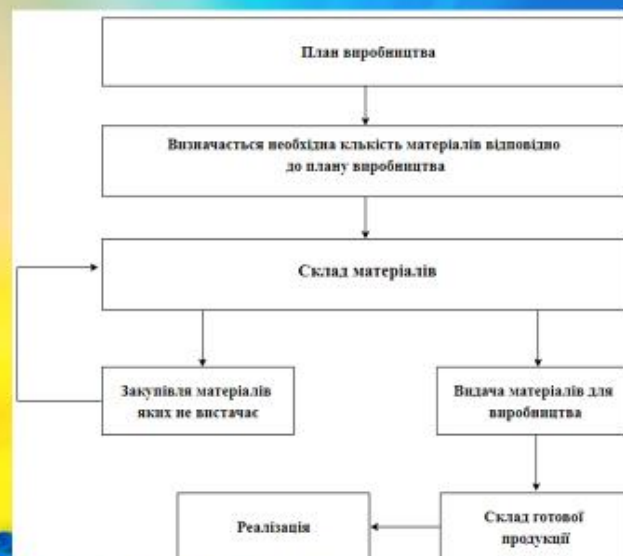
СТРУКТУРА

Схема інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві



СТРУКТУРА

Схема складання плану виробництва



Демонстрація роботи програми



LogIn

Email Address:

Password:

Remember Me

[LogIn](#) [Forgot Your Password!](#)

Register

Name:

Email Address:

Password:

Confirm Password:

[Register](#)

Демонстрація роботи програми

Таблиця >> Места Категорії клієнтів Резюме на роботу Документи по операції Додатки клієнтів Завантаження Відаткова накладна

Надходження Бюла <<< Таблиця

+ Додати

Довідник клієнтів

Ідентифікатор	Категорія	Відомості	Середній рейтинг	Місто	Ідентифікатор клієнта	Телефон	Електронна пошта	Розмір грошової суми (грн)	Статус	Додатки
Самківська	тренер	0	0	Київ	Київ	978899928	akcs254@gmail.com	1500	Відкрито	Закрито
Маском	тренер	25	1235	Київ	Маском	315841813	malghb@gmail.com	200000	Відкрито	Закрито
Колл	тренер	13	2540	Хмельницький	Власко	065451565	kolahb@gmail.com	200000	Відкрито	Закрито
Алекс	Повар	25	24	Хмельницький	10	097645231	andrey@gmail.com	18	Відкрито	Закрито
Артем	Повар	20	2500	Хмельницький	Закарюць	097851235	arukomsk22@gmail.com	0	Відкрито	Закрито

+ Додати

Відаткова накладна

Дата	Тип операції	Місце	Категорія	Середній рейтинг	Середній рейтинг	Сума грошової суми (грн)	Кількість клієнтів	Кількість грошової суми (грн)	Статус
2018-06-10	Платіжне доурахування	Самківська	тренер	11	10	5410	1	50	нчого
2020-06-10	Відаткова накладна	Київ	тренер	18	40	25040	3	40	нчого

Демонстрація роботи програми

Таблиця >>> Матеріали Готова продукція Калькуляція Документ по операції Довідник постачальників План виробництва <<< Таблиця

Калькуляційна таблиця

Номер операції в технології	Продукція	Співвідн.	Підприємство	Ширин.	Забарвлення	Код продукції
1	Берці шкірані	1	1	5 см	Видалити	Змінити
2	Берці спилок	1	1	3 см	Видалити	Змінити
3	Кросівки шкірані	1	1	4 см	Видалити	Змінити

Продукція

Продукція	Код	Розмір	Кількість	Вартість ц/фа	Сума	Видалити	Змінити
Берці шкірані	Олива	м2	20	1500	30000	Видалити	Змінити
Кросівки шкірані	Хакі	м2	20	1880	37600	Видалити	Змінити
Кросівки шкірані	Койот	м2	20	1750	35000	Видалити	Змінити
Берці шкірані	Чорна	м2	20	1700	34000	Видалити	Змінити
Кросівки шкірані	Олива	м2	20	1730	34600	Видалити	Змінити
Кросівки шкірані	Чорна	м2	20	1720	34400	Видалити	Змінити

Демонстрація роботи програми

План виробництва

Продукція

Берці шкірані

Кількість продукції на виробництво

5

Назва матеріалу

Спінлок

Кількість

5

Назва матеріалу

Ширин.

Кількість

20 см

Назва матеріалу

Пішоки

Кількість

5

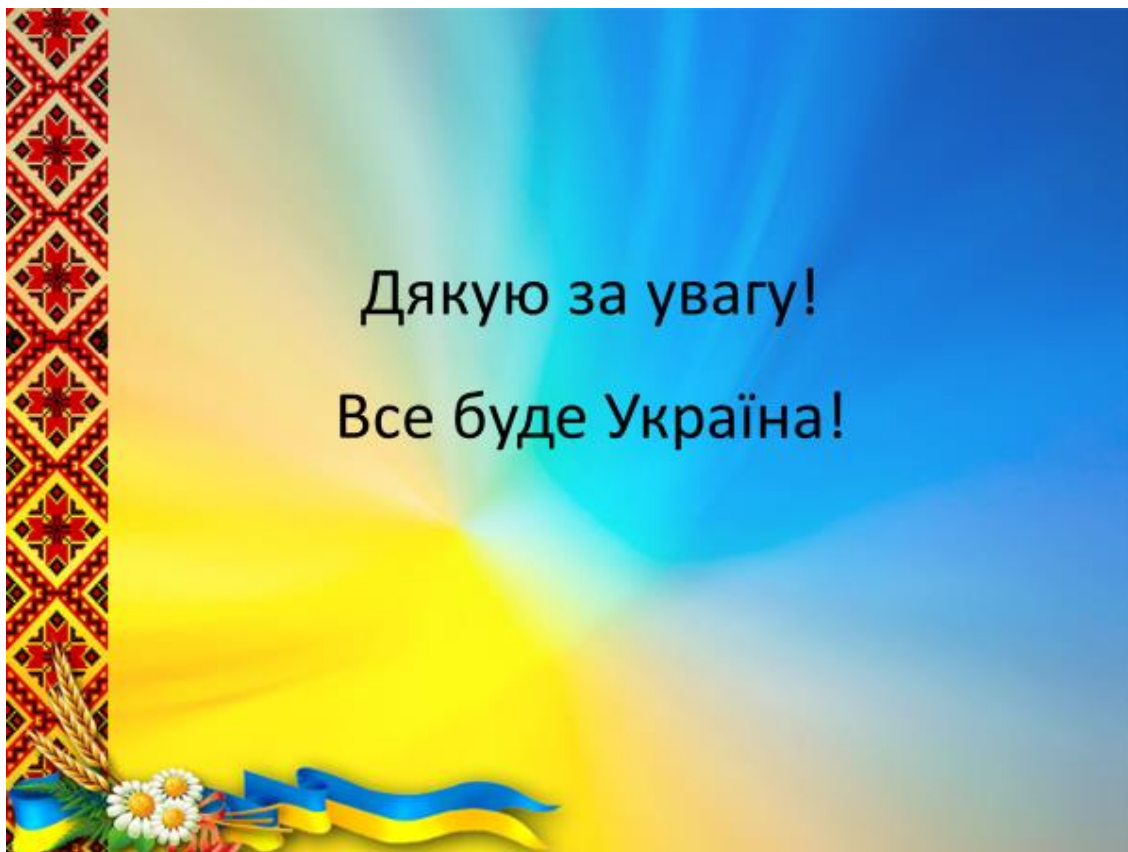
Початкова дата

14.06.2022

Кінцева дата

25.06.2022

Підтвердити



Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 2.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 8%**

ID: 104968 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА на тему Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві Додано в БД: 2022-06-10 Автори: М.В. Запорожець Руководители: М.О. Молчанова Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	68195	1020	4816 (7%)	70 (7%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы



Ім'я користувача:
Кафедра КН

ID перевірки:
1011533989

Дата перевірки:
10.06.2022 11:53:29 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
10.06.2022 11:54:58 EEST

ID користувача:
100005671

Назва документа: Запорожець_Записка_short

Кількість сторінок: 76 Кількість слів: 11394 Кількість символів: 83792 Розмір файлу: 3.06 MB ID файлу: 1011406964

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.7% Схожість

Найбільша схожість: 3.77% з Інтернет-джерелом (<http://mmsa.kpi.ua/sites/default/files/publications/%D0%91%D1%96%>).

7.7% Джерела з Інтернету 194 Сторінка 78

4.63% Джерела з Бібліотеки 94 Сторінка 79

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.

Підозріле форматування 32 сторінки

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

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

Назва: Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

Автор: студент групи КН-18-1 Запорожець Максим Вадимович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: викладач кафедри КН Молчанова М.О.

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.	<i>відповідає</i>
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:

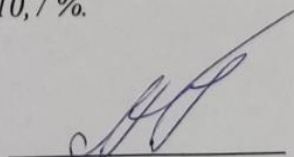
Запозичення, виявлені в роботі Запорожця М.В., не є плагіатом, оскільки: запозичення розміщені в розділі огляду існуючих підходів, не описують безпосередньо авторську роботу і не стосуються її результатів; усі запозичення фрагментарні; серед запозичень знаходяться загальновідомі терміни, скорочення та матеріали оглядових джерел.

Обсяг запозичень, визначений системами виявлення збігів/ідентичності/схожості, складає:

- за системою Anti-Plagiarism: 7%;

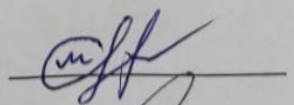
- за системою Unicheck: 10,7 %.

Керівник роботи



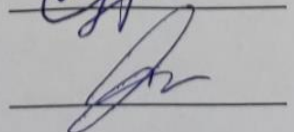
Марина МОЛЧАНОВА

Гарант ОП



Олександр МАЗУРЕЦЬ

Завідувач кафедри КН



Олександр БАРМАК



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента *гр. КН-18-1* Запорозжця Максима Вадимовича

за темою: Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

1. Актуальність обраної теми

Зважаючи на різючі зміни у суспільно-політичному житті країни, багато власників підприємств були змушені перенести виробничі потужності у інші більш безпечні регіони або тимчасово змінити профіль виробництва. Це в свою чергу спричинило певні проблеми матеріально-технічним забезпеченням та налагодженням безперебійного виробничого процесу. Тому створення відповідної інформаційної технології підтримки прийняття рішень забезпечить власнику вести належний контроль процесів та складських запасів на виробництві, зокрема наступні функції: формування плану виробництва та визначення потреб у матеріалах для виготовлення необхідної продукції та контроль руху запасів на складі та виробництві, визначаючи їхню наявність, надлишок або дефіцит, забезпечується вчасне постачання матеріалів на склад та у виробництво

2. Повнота розкриття мети та завдань роботи

Метою кваліфікаційної роботи бакалавра була реалізація інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Для досягнення поставленої мети виконано наступні задачі: автоматизація ведення даних щодо потреб виробництва у визначених матеріалах для виготовлення визначених виробів; забезпечення формування планів виробництва; автоматизація визначення потреб виробництва у визначених матеріалах згідно сформованих планів виробництва; проведення реєстрації надходжень матеріалів на склад виробництва та автоматизація визначення залишкових потреб для подальших надходжень у визначених матеріалах згідно сформованих планів виробництва; автоматизація визначення надлишкових запасів матеріалів на складі згідно сформованих планів виробництва; автоматизація ведення даних щодо використання визначених матеріалів для виготовлення визначених виробів та ведення даних щодо інших втрат і витрат матеріалів; автоматизація визначення очікуваних залишків матеріалів на складі згідно

заресстрованих витрат матеріалів і витрат матеріалів на виробництво для проведення ревізійної діяльності. Зміст роботи виконано в повному обсязі, належним чином. Тема є змістовно розкритою.

3. Зміст кожного розділу роботи

Перший розділ присвячений проведенню аналізу предметної області та визначенню основних параметрів для розв'язку поставленої задачі. Другий розділ присвячений проєктуванню функціональної структури інформаційної системи. Третій розділ присвячений програмній реалізації спроектованої функціональної структури інформаційної системи.

4. Оцінка розробленої інформаційної системи, її практична цінність

Спроектвана інформаційна система дозволяє скласти план виробництва з врахуванням усіх нюансів, проводити реєстрацію надходжень матеріалів на склад виробництва, дозволяє проводити ведення даних щодо використання визначених матеріалів для виготовлення визначених виробів, визначати потреби виробництва у визначених матеріалах для виготовлення необхідної продукції. Цінність даної системи полягає в повній автоматизації її функціональних можливостей.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота виконана на належному науково-методичному рівні та відповідає встановленим вимогам щодо оформлення такого роду праць.

6. Недоліки кваліфікаційної роботи бакалавра

Кваліфікаційна робота бакалавра виглядала б привабливіше, якби у ній було розглянуто більше математичних підходів щодо вирішення задачі взаємодії документів між собою.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент Бедратюк Л.П. р. ф.и. н.





ВІДГУК НАУКОВОГО КЕРІВНИКА на кваліфікаційну роботу бакалавра

студента гр. КН-18-1 Запорожця Максима Вадимовича

за темою Інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві

1. Актуальність теми

Актуальність теми обґрунтована, оскільки створення інформаційної технології дозволить автоматизувати систему підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. Однією із складових для досягнення мети є формуванням плану виробництва з врахуванням усіх особливостей, що є особливо актуальним у період потужного пришвидшення масової діджиталізації.

2. Відповідність роботи предметній області Стандарту спеціальності 122 Комп'ютерні науки

Тема кваліфікаційної роботи відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра, оскільки метою роботи є створення інформаційної технології підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій.

3. Професійні та особистісні якості бакалавра

Студент Запорожець М.В. під час роботи над кваліфікаційною роботою бакалавра продемонстрував високий рівень знань та умінь за спеціальністю "Комп'ютерні науки", проявив себе відповідальним та сумлінним студентом. Опанував необхідні професійні навички за напрямком «Комп'ютерні науки».

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Одержані в роботі результати є наслідком особистої діяльності студента, який самостійно виконував всі поставлені задачі.

5. Ступінь оволодіння методами дослідження

При реалізації кваліфікаційної роботи студент Запорожець М.В. показав достатній рівень компетентностей та володіння необхідними інструментами та

обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Усі поставлені вимоги до роботи виконані в повному обсязі, проведено аналіз актуальності та відомих досліджень в межах обраної теми, також реалізоване відповідне програмне забезпечення.

7. Логічність, послідовність, аргументованість, літературна грамотність викладення матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

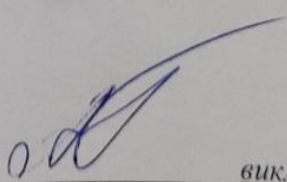
8. Можливість практичного застосування кваліфікаційної роботи бакалавра, окремих її частин

Реалізована інформаційна технологія підтримки прийняття рішень з контролю складських запасів матеріалів на виробництві може бути використана на різного роду підприємствах, що безперервно взаємодіють із обліком та контролем складських запасів.

9. Висновок про можливість допуску кваліфікаційної роботи бакалавра до захисту, на яку оцінку заслуговує робота

Враховуючи високий рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Керівник _____



викладач каф. КН Марина МОЛЧАНОВА