

Міністерство освіти і науки України  
Національний університет водного господарства та природокористування  
Навчально-науковий інститут автоматики, кібернетики та  
обчислювальної техніки  
Кафедра комп'ютерних технологій та економічної кібернетики

**Допущено до захисту:**

Завідувач кафедри

\_\_\_\_\_ д. е. н., проф. П. М. Грицюк

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

## **КВАЛІФІКАЦІЙНА РОБОТА**

на здобуття ступеня «бакалавр»

за освітньо-професійною програмою «Інформаційні системи та технології»

спеціальності 122 «Інформаційні системи та технології»

на тему: «Інформаційна підтримка обліку торговельної діяльності

підприємства з виробництва верхнього одягу»

**Виконав:** здобувач вищої освіти 2 курсу,

групи ІСТ-21-інт

**Удод Діана Богданівна**

**Керівник:** к. т. н., доц. Барановський С. В.

**Рецензент:** к.е.н., доц. Бабич Т.Ю.

Рівне – 2022

## АНОТАЦІЯ

Удод Діана Богданівна «Інформаційна підтримка обліку торговельної діяльності підприємства з виробництва верхнього одягу» / Кваліфікаційна бакалаврська робота / м. Рівне: НУВГП, 2022. 93 ст. Українською мовою.

Бакалаврська робота складається зі вступу, 3 розділів, висновків, списку використаних джерел, додатку, 29 ілюстрацій, 2 таблиць.

**Метою бакалаврської дипломної роботи** є створення автоматизованої інформаційної системи підприємства з виробництва верхнього одягу, за допомогою якої буде здійснюватися взаємодія керівництва з різними інформаційними аспектами діяльності. Для досягнення поставленої мети були виділені і виконані наступні завдання:

- Вибір засобів і середовища розробки;
- Розробка користувальницького інтерфейсу проекту;
- Реалізація проекту за допомогою програмних засобів.

**Об'єкт дослідження** – підприємствоз виробництва верхнього одягу.

**Предмет дослідження** – інформатизація діяльності підприємства з виробництва верхнього одягу.

Розроблено інформаційну систему, що вирішує поставлені задачі за допомогою інструментів СУБД. Створена автоматизована інформаційна система забезпечує ефективне збереження та цілісність даних, за допомогою форм, кожна з яких відповідає певній таблиці чи сукупності таблиць, дозволяє легке введення, редагування, пошук даних в базі, за допомогою механізму запитів дозволяє пошук та відбір даних в базі за певними критеріями і ці дані можна перенаправити у відповідний звіт, який можна віддрукувати.

Створення автоматизованої інформаційної системи для обліку товарів з виробництва верхнього одягу для контролю виробництва.

## ANNOTATION

Udod Diana Bogdanovna "Information support for accounting of trade activities of the enterprise for the production of outerwear" / Qualifying bachelor's degree / Rivne: NUVGP, 2022. 93 p. In ukrainian.

The bachelor's thesis consists of an introduction, 3 chapters, conclusions, a list of sources used, an appendix, 29 illustrations, 2 tables.

The purpose of the bachelor's thesis is to create an automated information system of the enterprise for the production of outerwear, which will be the interaction of management with various information aspects of the activity. To achieve this goal, the following tasks were identified and performed:

- Choice of tools and development environment;
- Development of the user interface of the project;
- Implementation of the project using software.

The object of research is an outerwear company.

The subject of research - informatization of the enterprise for the production of outerwear.

An information system has been developed that solves the set tasks with the help of DBMS tools. The created automated information system provides effective preservation and integrity of data, by means of forms, each of which corresponds to a certain table or set of tables, allows easy input, editing, data retrieval and data retrieval in the database. and this data can be redirected to a corresponding report, which can be printed.

Creation of an automated information system for accounting of goods for the production of outerwear for production control.

## ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. ЗАГАЛЬНІ ПРИНЦИПИ ПОБУДОВИ ІНФОРМАЦІЙНИХ СИСТЕМ .	7
1.1 Призначення автоматизованих інформаційних систем .....	7
1.2 Класифікація АІС .....	9
1.3 Програмне забезпечення АІС.....	14
РОЗДІЛ 2.РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ .....	21
2.1 Аналіз предметної області.....	21
2.2 Логічна модель даних .....	22
2.3 База даних інформаційної системи.....	24
РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ .....	32
3.1 Довідники в ІС .....	32
3.2 Функціональні можливості системи .....	38
2.3 Аналіз вихідної інформації .....	46
3.4 Інтерфейс користувача.....	50
ВИСНОВОК.....	54
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	56
ДОДАТКИ.....	58

## ВСТУП

**Актуальність теми.** Роль інформаційних технологій у сучасному світі є стратегічно важливою для адекватного реагування на динаміку ринку, підтримувати і заглиблювати конкурентну перевагу з метою досягнення максимальної вигоди. Застосування інформаційних систем дозволяє радикально змінити стиль управління і значно поліпшити показники діяльності підприємства.

Для розвитку підприємства будь-якої галузі необхідне впровадження сучасних технологій обліку та контролю руху товарів, аналізу та збереження інформації. Використання терміналів збору даних допоможе швидко та ефективно вирішити питання, де потрібно проводити точний облік, контроль та обробку даних, при цьому людський чинник максимально усувається. Розвиток інформаційних технологій сприяє постійній модернізації терміналів збору даних, яка спрямована на покращення їх технічних параметрів та розширення сфери їх застосування [12].

Інформаційна система обліку містить різні елементи, важливі в обліковому циклі. Незважаючи на те, що інформація, що міститься в системі, різниться залежно від галузей та розмірів підприємств, типовий АІС включає дані, що стосуються інформації про клієнтів, інформацію про працівників та податкову інформацію. Конкретні дані включають замовлення на продаж та звіти про аналіз, заявки на купівлю, рахунки-фактури, реєстри чеків, товарно-матеріальні цінності, інформацію про звіт підприємства.

Інформаційна система обліку повинна мати структуру бази даних для зберігання інформації. Ця структура бази даних зазвичай запрограмована на мові запитів, що дозволяє маніпулювати таблицями та даними.

Процес створення автоматизованих інформаційних систем обліку - це сукупність робіт, що починаються формуванням вхідних вимог до цієї системи, а закінчуються введенням її в дію. Такий процес передбачає або способи індивідуального розроблення проектної документації.

**Метою бакалаврської дипломної роботи** є створення автоматизованої інформаційної системи підприємства з виробництва верхнього одягу, за допомогою якої буде здійснюватися взаємодія керівництва з різними інформаційними аспектами діяльності. Для досягнення поставленої мети були виділені і виконані наступні завдання:

- Вибір засобів і середовища розробки;
- Розробка користувальницького інтерфейсу проекту;
- Реалізація проекту за допомогою програмних засобів.

**Об'єкт дослідження** – підприємство з виробництва верхнього одягу.

**Предмет дослідження** – інформатизація діяльності підприємства з виробництва верхнього одягу.

## РОЗДІЛ 1. ЗАГАЛЬНІ ПРИНЦИПИ ПОБУДОВИ ІНФОРМАЦІЙНИХ СИСТЕМ

### 1.1 Призначення автоматизованих інформаційних систем

Система - це упорядкована сукупність різнорідних елементів або частин, взаємодіючих між собою і з зовнішнім середовищем, об'єднаних в єдине ціле і функціонують в інтересах досягнення єдиних цілей.

Інформаційна система - організований набір елементів, що збирає, обробляє, передає, зберігає та надає дані. Інформаційна система складається із людей, обладнання, процесів, процедур, даних та операцій. Кожна інформаційна система включає в себе наступні компоненти [15]:

- структура системи;
- функції кожного елемента системи;
- вхід і вихід кожного елемента і системи в цілому;
- мета і обмеження системи та її окремих елементів.

**Розробка інформаційної системи** передбачає вирішення двох таких завдань:

- наповнення системи даними певної предметної області;
- створення (бажано графічного) інтерфейсу користувача для отримання необхідної інформації.

Інформаційна система не тільки відображає функціонування об'єкта управління, а й впливає на нього через органи управління. Вона є сукупністю інформаційних процесів для задоволення потреби в інформації різних рівнів прийняття рішень.

**Автоматизована інформаційна система (АІС)** - система, що реалізує інформаційні технології у сфері управління за спільної роботи управлінського персоналу і комплексу технічних засобів.

Вона призначена для автоматизованого збирання, реєстрації, збереження, пошуку, оброблення та видачі інформації за запитами користувачів [9].

Автоматизована інформаційна система має забезпечувати:

- постійне спостереження за поточним станом об'єкта управління та його характеристик;
- адаптації, тобто пристосування до прийнятої практики бізнесу та модифікації, якщо така практика змінюється;
- підтримку професійної діяльності управлінських працівників;
- взаємодію з управлінським персоналом;
- здійснення збирання та аналізу даних для управління й автоматичного виконання програмних засобів при настанні заданого часу з формуванням необхідної звітності;
- реалізацію системи підказок і рекомендацій для користувачів;
- ефективне збереження даних у базі даних і можливість доступу до них будь-якого кінцевого користувача зі свого робочого місця.
- взаємодію користувачів між собою на основі безпаперової технології

Інформаційна система обліку - це сукупність інформації, апаратно-програмних і технологічних засобів, засобів телекомунікації, баз і банків даних, методів і процедур, персоналу управління, які реалізують функції збору, оброблення, нагромадження та оброблення інформації для підготовки та прийняття ефективних управлінських рішень. Це множина різних елементів та зв'язків між ними, що складають систему в цілому [10].

У сучасній концепції організації автоматизована інформаційна система виділяють такі типові функції програми:

- Прогнозування;
- Планування;
- Обліку;

- Контролю;
- Аналізу;
- Регулювання.

Прогнозування - функція, за допомогою якої здійснюються формування й обґрунтування наукових передумов перспективи розвитку об'єкта управління, можливих змін його стану до певного моменту часу в майбутньому.

Планування - функція, за допомогою якої в ідеальній формі реалізується мета управління. На вищому рівні управління планування орієнтовано на тривалий термін, на середньому - на більш короткий термін, при цьому план вищого рівня деталізується.

Облік - функція, спрямована на одержання інформації на оперативному та середньому рівнях. Здобуті результати передаються на вищий рівень.

Контроль - зіставлення фактичних показників із плановими і нормативними, визначення відхилень, що виходять за межі допустимих значень. Виконується на всіх трьох рівнях управління.

Аналіз - установлення причин відхилень, виявлення резервів, знаходження способів усунення неузгодженостей між фактичними показниками і запланованими. Виконується на середньому та вищому рівнях управління.

Регулювання - коригування ходу роботи об'єкта управління, прийняття рішень для виправлення ситуації, що утворилася, і виведення об'єкта управління на заплановані показники. Виконується на середньому та вищому рівнях управління. Сучасні АІС здатні обробляти інформацію за всіма функціями управління і передавати її на всі рівні управління. Для обміну інформацією між рівнями управління застосовуються засоби комунікації та мережні технології.

## **1.2 Класифікація АІС**

Створення та використання автоматизованих інформаційних систем сприяє підвищенню ефективності функціонування економічного об'єкта і забезпечує

якість управління. Найбільша ефективність АІС досягається під час оптимізації планів роботи підприємств, фірм і галузей, швидкого вироблення різних оперативних рішень, злагодженого маневрування матеріальними та фінансовими ресурсами тощо. Тому процес управління в умовах функціонування АІС побудований на використанні відповідних економіко-організаційних моделей, які певною мірою адекватно відображають характерні структурно-динамічні властивості об'єкта управління [16].

Досвід створення АІС, впровадження у практику економічної роботи оптимізаційних методів і моделей, формалізація ситуацій виробничих процесів, впровадження у державних та комерційних структурах сучасних обчислюваних засобів корінним чином видозмінили технологію інформаційних процесів у різних системах управління.

АІС за своїм призначенням різноманітні і можуть бути класифіковані за різними ознаками, зокрема:

- за сферою, призначенням чи характером дії об'єкта управління (АІС адміністративні, виробничо-технічні, соціальні, зв'язку, сільського господарства, транспорту тощо);
- за видами процесів управління (АІС управління технологічними процесами, управління організаційно-технологічними процесами, організаційного управління, наукових досліджень, навчаючі);
- за рівнем у системі державного управління (міжгалузеві, регіональні, галузеві АІС);
- за ступенем централізації опрацювання інформації (АІС централізовані, децентралізовані або системи колективного користування).

Кваліфікаційні складові за видами процесів управління мають таке призначення: АІС управління технологічними процесами – це людино-машинні системи, які забезпечують управління технологічними пристроями, станками, автоматичними лініями.

АІС управління організаційно-технологічними процесами – це багаторівневі системи, які поєднують АІС управління технологічними процесами і АІС управління підприємствами.

АІС організаційного управління – це АІС, об'єктом для яких слугують виробничо-господарські, соціально-економічні функціональні процеси. До них належать:

- Банківські АІС;
- АІС фондового ринку;
- Фінансові АІС;
- Страхові АІС;
- Податкові АІС;
- АІС митної служби;
- Статистичні АІС;
- Бухгалтерські АІС.

АІС наукових досліджень забезпечують високу якість та ефективність міжгалузевих розрахунків і наукових дослідів. Методичною базою таких систем слугують економіко-математичні методи, а базою забезпечення – комп'ютери і технічні засоби для проведення експериментальних робіт моделювання. Ці та організаційно-технологічні системи можуть додатково мати у своєму складі системи автоматизованого проектування робіт (САПР).

Навчаючі АІС отримали широке застосування під час навчання фахівців у системі освіти, а також під час перепідготовки та підвищення кваліфікації працівників різних галузей знань.

Кваліфікаційні складові за рівнем у системі державного управління мають таке призначення:

Міжгалузеві АІС забезпечують розроблення економічних прогнозів, державного бюджету, здійснюють контроль за результатами та регулюванням

діяльності всіх ланок господарства держави, а також контролюють наявність та розподіл ресурсів.

Регіональні АІС призначені для управління адміністративно-територіальними районами. Діяльність цих систем скерована на якісне виконання управлінських функцій у регіоні, формування звітності, видання оперативних відомостей місцевим державним та господарським органам.

Галузеві АІС функціонують у сферах промислового та агропромислового комплексів, у будівництві, на транспорті, у статистиці, у системі комунікацій тощо. Ці системи дають змогу розв'язувати завдання інформаційного обслуговування апарату управління відповідних відомств [15].

За роллю у виробничій діяльності та в процесі прийняття управлінських рішень АІС поділяють на:

- виробничі АІС, які призначені для забезпечення виробничої діяльності;
- управлінські АІС, які призначені для підтримування процесу прийняття

управлінських рішень.

До виробничих АІС належать:

- системи опрацювання інформації, призначені для опрацювання даних бізнес-транзакцій;
- системи керування технологічними (промисловими) процеами;
- системи автоматизації офісу, призначені для створення комунікацій і ведення діловодства.

До управлінських АІС належать:

- системи формування звітності;
- системи підтримки прийняття рішень (т. зв. інтерактивне підтримування);
- інформаційні системи для керівників (менеджерів), за допомогою яких продукується інформація для керівників вищого рівня.

Співвідношення класів АІС і рівнів менеджменту подаємо у табл. 1.1.

До інших класів АІС належать:

Експертні системи – клас систем штучного інтелекту, що здатні самостійно розв’язувати завдання у певній предметній ділянці, використовуючи базу знань.

Табличка 1.1

<b>Рівень менеджменту</b>	<b>Клас АІС</b>
Стратегічний менеджмент	АІС для керівників
Тактичний менеджмент	Системи підтримки прийняття рішень
Операційний менеджмент	Система формування звітності
Виробнича діяльність	Система автоматизації офісу, опрацювання даних, керування процесами

Персональні АІС – автоматизовані інформаційні системи на базі персональних комп’ютерів, з якими користувач працює безпосередньо.

Функціональні АІС – забезпечують підтримку виробничої та управлінської діяльності у головних функціональних сферах виробництва та бізнесу.

Стратегічні АІС – виробляють продукти або послуги, які сприяють досягненню стратегічних цілей організації, зокрема:

- підвищують ефективність бізнес-процесів;
- створюють нові товари, послуги та бізнес-процеси;
- нарощують стратегічні інформаційні ресурси.

АІС вдосконалюють наявні бізнес-процеси, зокрема:

- автоматизують бізнес-процес;
- прискорюють передання інформації на великі відстані, внаслідок чого процеси стають географічно незалежними;
- вносять комплексні аналітичні методи в бізнес-процес;
- залучають до бізнес-процесу великий обсяг деталізованої інформації;

- надають можливість змінювати послідовність робіт або виконувати декілька процесів паралельно;
- залучають до процесу спеціальні експертні знання;
- здійснюють моніторинг бізнес-процесу, надаючи конкретну інформацію про його стан.

### 1.3 Програмне забезпечення АІС

**Програмне забезпечення АІС** - це сукупність комп'ютерних програм та відповідної документації, необхідна для їхнього встановлення та експлуатації, і призначена для реалізації різнотипних завдань за допомогою обчислювальної техніки. За класами розв'язуваних завдань певних фінансово-господарських чи науково-технічних сфер діяльності та функціональним призначенням програмне забезпечення можна поділити на дві частини:

*загальносистемне*, що забезпечує функціонування та ефективну експлуатацію комп'ютерної техніки, і призначене для автоматизованого розроблення машинних програм та організації обчислювального процесу;

*спеціальне*, що призначене для розв'язування конкретних завдань у різних сферах застосування обчислювальної техніки.

До загальносистемного програмного забезпечення належать:

- операційні системи;
- системні сервісні програми;
- системи програмування різних рівнів;
- тестові та діагностичні програми.

Складовими частинами спеціального програмного забезпечення є:

- прикладні пакети програм;
- забезпечуючі або обслуговуючі програми.

Для створення програмного забезпечення використовують різні засоби, які узагальнено називають мовами програмування. Основне призначення сучасного комп'ютерного програмного забезпечення полягає в узгодженні різних програмних застосувань та способів роботи обчислювальної системи з метою збільшення її продуктивності, узгодження адаптування програм до умов зміни комп'ютерних ресурсів, проведення програмних змін і модифікацій та розширення можливостей обчислювальної системи, що дає змогу поліпшити загальний інтерфейс користувача з комп'ютерною технікою та підвищити продуктивність його праці у конкретній галузі діяльності.

Програмне забезпечення яким буде розроблятися проект:

- Visual Studio
- NET Framework
- Мова C #

Програмними для формування і керування базами даних:

- MySQL-Front;
- Navicat for MySQL;

**Visual Studio** – інтегроване середовище розробки програмного забезпечення від фірми Microsoft. Дане середовище дозволяє створювати різноманітні програмні продукти: консольні програми, програми з графічним інтерфейсом, наприклад віконні додатки Windows Forms, а також Web-додатки тощо [3].

Visual Studio включає в себе редактор коду, що підтримує IntelliSense (компонент завершення коду), а також рефакторинг коду. Вбудований відладчик працює як відладчик на рівні вихідного рівня, так і відладчик на рівні машини. Інші вбудовані інструменти включають програму для кодування, конструктор форм для побудови графічних інтерфейсів, веб-дизайнер, дизайнер класів та дизайнер схеми баз даних. Він приймає плагіни, які покращують функціональність практично на всіх рівнях, включаючи підтримку систем керування джерельними ресурсами

(наприклад, Subversion та Git) та додавання нових наборів інструментів, таких як редактори та візуальні розробники для мов або наборів інструментів для інших аспектів розробки програмного забезпечення життєвий цикл (наприклад, клієнт Team Foundation Server: Team Explorer).

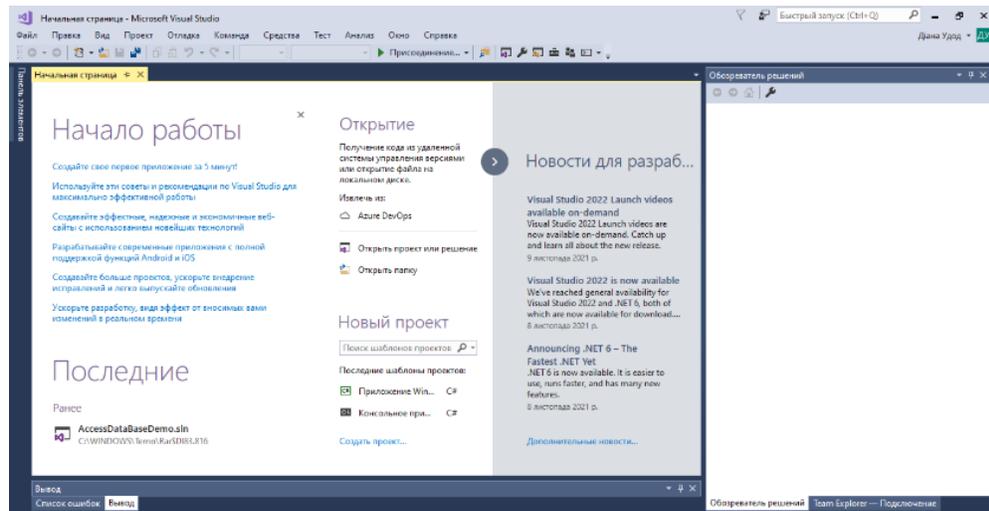


Рис.1.1 Головне вікно програми Microsoft Visual Studio

Visual Studio включає в себе редактор коду, що підтримує IntelliSense (компонент завершення коду), а також рефакторинг коду. Вбудований відладчик працює як відладчик на рівні вихідного рівня, так і відладчик на рівні машини. Інші вбудовані інструменти включають програму для кодування, конструктор форм для побудови графічних інтерфейсів, веб-дизайнер, дизайнер класів та дизайнер схеми баз даних. Він приймає плагіни, які покращують функціональність практично на всіх рівнях, включаючи підтримку систем керування джерельними ресурсами (наприклад, Subversion та Git) та додавання нових наборів інструментів, таких як редактори та візуальні розробники для мов або наборів інструментів для інших аспектів розробки програмного забезпечення життєвий цикл (наприклад, клієнт Team Foundation Server: Team Explorer).

Visual Studio підтримує 36 різних мов програмування і дозволяє редакторові коду та відладчику підтримувати (в тій чи іншій мірі) майже будь-яку мову програмування, якщо існує певна мова-служба. Вбудовані мови включають C, C++, C++ / CLI, Visual Basic, .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML та

CSS. Підтримка інших мов, таких як Python, Ruby, Node.js та M, серед інших, доступна через плагіни. Java (і J #) були підтримані в минулому. А також містить в собі сильний інструмент NuGet для завантаження різного роду плагінів та додатків.

**.NET Framework** - це програмне забезпечення, розроблене корпорацією Майкрософт, яке працює переважно на Microsoft Windows. Вона включає в себе бібліотеку великого класу з ім'ям Framework Class Library (FCL) і забезпечує сумісність мов (кожен мову може використовувати код, написаний іншими мовами) на кількох мовах програмування. Програми, написані для .NET Framework, виконуються в програмному середовищі (на відміну від апаратного середовища) називається Common Language Runtime (CLR) - віртуальна машина додатків, яка надає послуги, такі як безпека, керування пам'яттю та обробка виключень. Таким чином, комп'ютерний код, написаний з використанням .NET Framework, називається "керованим кодом". FCL і CLR разом складають .NET Framework [2].

FCL забезпечує користувальницький інтерфейс, доступ до даних, підключення до бази даних, криптографію, розробку веб-додатків, цифрові алгоритми та мережеві зв'язки. Програмісти створюють програмне забезпечення, об'єднуючи їх вихідний код з .NET Framework та іншими бібліотеками. Рамка призначена для використання більшістю нових додатків, створених для платформи Windows. Корпорація Майкрософт також випускає інтегроване середовище розробки, головним чином для .NET програмного забезпечення, яке називається Visual Studio.

.NET Framework розпочалася як фірмове програмне забезпечення, хоча фірма прагнула стандартизувати пакет програмного забезпечення практично відразу, навіть до її першого випуску. Незважаючи на зусилля щодо стандартизації, розробники, головним чином, у вільних та відкритих програмних спільнотах, висловлювали занепокоєння вибраними умовами та перспективами здійснення будь-якої вільної та відкритої версії, особливо стосовно патентів на програмне забезпечення.

**Мова C#** – проста, потужна, статично типізована, об’єктно орієнтована мова програмування від компанії Microsoft. C# входить до сімейства мов програмування C, синтаксис мови буде знайомим програмістам, що працювали з C, C++, Java та JavaScript [13].

Перша версія мови C# була створена в 1998-2001 роках, групою інженерів Microsoft під керівництвом *Андреса Гейлсберга* та *Скотта Вільтаумота*, як основна мова програмування платформи **Microsoft .Net**.

C# увібрав в себе найкращі властивості попередників – мов C, C++, Modula, Object Pascal, спираючись на практичний досвід їх використання. Деякі проблематичні моделі, що до цього використовувались у мовах програмування, зокрема множинне спадкування класів (яке використовується у мові C++), були свідомо виключені.

В багатьох відношеннях мова C# дуже схожа на Java, це відображено в синтаксисах та основних поняттях цих мов програмування.

### **Історія виникнення C#**

Назва мови C# трактується як наступне покоління розвитку C++, а символ # – символізує “++++”. Спочатку в назві фігурував дієз - “#” (англійською sharp), однак через відсутність цього символу на клавіатурі, використовується знак для позначення номеру “#”. Загалом назву мови можна позначати обома символами.

Розробка мови C# розпочалася в грудні 1998 року, та готувався до випуску разом з продуктами групи Millenium. Проект мав назву COOL (C-style **O**bject **O**riented **L**anguage), та розроблявся як аналог Java від компанії Oracle. C# був аносований, широкому загалу, в 2000 році, як основна мова платформи Microsoft .Net Framework. В цьому ж році з’явилася перша загальнодоступна бета-версія [13].

Перша фінальна версія мови програмування C# була випущена в 2002 році разом з середовищем інтегрованої розробки програмного забезпечення **Visual Studio .Net**.

Подібно до Java, C# отримав наступні концепції:

- віртуальна машина – платформа .Net виконує програму подібно до віртуальної машини від Java;
- байт-код – програмний код компілюється в проміжкову мову MSIL(Microsoft Intermediate Language), а вже потім перетворюється на машинну мову, в залежності від платформи на якій запускається програма;
- керований код – оскільки програми, написані на C#, виконуються виключно у віртуальному середовищі CLR (Common Language Runtime), це дає змогу контролювати виконання програми та у будь який момент зупинити її, а також контролювати використання пам'яті програмою, за необхідності - збільшувати, чи видаляти частини пам'яті, які використовує програма.

**MySQL-Front** – це популярний сервер баз даних, що використовується у різних додатках. SQL означає мову структурованих запитів - (S)tructured (Q)uery (L)anguage, яку MySQL використовує для комунікації з іншими програмами. Крім того, MySQL має власні розширені функції SQL для того, щоб забезпечити користувачам додатковий функціонал. Програма поєднує в собі простоту використання з потужною настільною програмою. Простий та інтуїтивно зрозумілий інтерфейс простий у використанні [5].

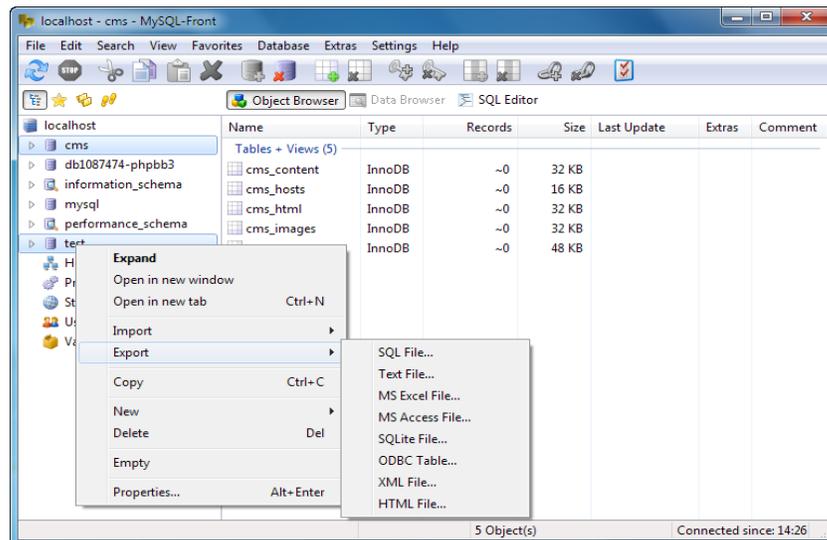


Рис.1.2 Середовище програми MySQL-Front

Navicat представляє зрозумілий інтерфейс для легкого керування базою даних MySQL. Ви можете швидко переглядати такі елементи, як таблиці, збережені процедури, події, запити та звіти. Ваші дані можна синхронізувати, щоб вони належним чином оновлювались у вашій базі даних. Navicat підтримує всі версії MySQL і дозволяє легко імпортувати та експортувати дані. Аналогічно, Navicat дозволяє легко створювати та роздруковувати звіти у вашій базі даних. Ви також можете керувати кількома користувачами та з'єднаннями.

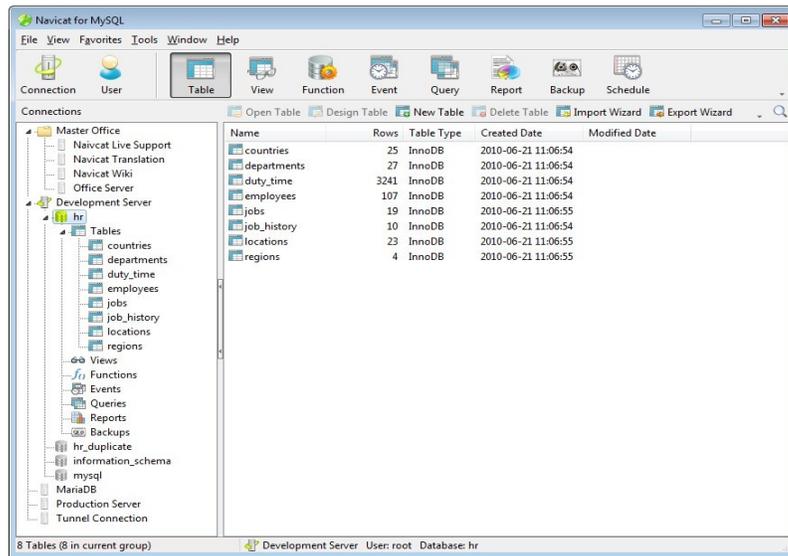


Рис. 1.3 Середовище програми Navicat for MySQL

## РОЗДІЛ 2. РОЗРОБКА ПРОЕКТУ ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 2.1 Аналіз предметної області

Особлива увага приділяється соціально орієнтованим галузям економіки, таким як легка промисловість. Більш впевнено розвивається швейна галузь, яка обслуговує весь господарський комплекс. В умовах ринкових відносин перед нею постає складна задача – задовольнити індивідуальні потреби людей не тільки у розмірах одягу, а й у забезпеченні його комфортності.

Сьогодні моделюючі підприємства розробляють заходи, що спрямовані на підвищення інтересу і задоволення кожної групи споживачів за рахунок виявлення важливих для них сподівань та наступних реалізацій цих сподівань в продукції.

Головна задача швейної промисловості – задоволення потреб людей в одязі високої якості й різноманітного асортименту. Рішення цієї задачі здійснюється на основі підвищення ефективності виробництва, підвищення продуктивності праці, поліпшення якості роботи, удосконалення праці й виробництва [17].

Швейна промисловість відіграє важливу роль у загальній системі народного господарства країни. Після текстильної вона є найважливішою підгалуззю легкої промисловості. Забезпечення населення нашої країни якісним одягом є досить впливовим чинником як підвищення ефективності виробництва та збільшення добробуту населення, так і забезпеченості робочими місцями наявних трудових ресурсів, особливо жіночих. Адже на підприємствах легкої промисловості, в тому числі і швейної, у виробництві зайняті переважно жінки.

Основною діяльністю швейного підприємства є виробництво верхнього одягу. У виробничу структуру швейного підприємства входять основні цехи:

Експериментальний;

Швейний;

Склад.

## 2.2 Логічна модель даних

Введення в експлуатацію даної АІС дозволить автоматизувати збір і зберігання інформації про виробництво верхнього одягу, а саме про товар, сапр (система автоматизованого проектування і розрахунку), моделі, інформацію про цехи і видачу на пошиття в цехи. В таблиці про виробництво міститься інформація який саме товар та про склад товару, а також є дані про відправку по завершенню пошиття даної моделі одягу [10].

У разі замовлення на іншу модель пошиття, існує можливість швидкого додавання назву моделі кількість і відправку товару. Так само існує можливість швидкого призначення відправки на цех пошиття даного товару.

Крім цього, надається можливість швидкого редагування даних, що стосуються самих товарів, моделей для пошиття на цех і відправки даного товару.

Завдяки подібним технічним рішенням система стає більш універсальною і зрозумілою для користувача, що і є основним завданням проектування.

Таблиця 2.1

Таблиці БД

Назва таблиці	Назви полів	Властивості полів
Товари	Код товару	int(11) unsigned <aut_increment>
	Назва товару	varchar(100) <NULL>
	Кількість замовлень	varchar(50) <NULL>
	Код відправки	int(11) unsigned <NULL>
САПР	Код сапр	int(11) unsigned <aut_increment>
	Код товару	int(11) unsigned <NULL>
	Код моделі	int(11) unsigned <NULL>
	Дата отримання товару	date <NULL>
	Назва тканини	varchar(100) <NULL>
	Розмір моделі	varchar(55) <NULL>

	Кількість моделей	varchar(50) <NULL>
Моделі	Код моделі	int(11) unsigned <NULL>
	Код сапр	varchar(100) <NULL>
	Назва моделі	int(11) unsigned <NULL>
	Ціна моделі	varchar(100) <NULL>
Видача на цех	Код видачі на цех	int(11) unsigned <aut_increment>
	Код моделі	int(11) unsigned <NULL>
	Дата отримання	date <NULL>
	Кількість	varchar(55) <NULL>
	Хто прийняв	varchar(255) <NULL>
	Код цеху	int(11) unsigned <NULL>
Цех	Код цеху	int(11) unsigned <aut_increment>
	Номер цеху	varchar(50) <NULL>
	Назва цеху	varchar(100) <NULL>
Виробництво	Код виробника	int(11) unsigned <aut_increment>
	Код цеху	int(11) unsigned <NULL>
	Назва товару	varchar(100) <NULL>
	Дані товару	varchar(255) <NULL>
	Кількість	varchar(50) <NULL>
Відправка	Код відправки	int(11) unsigned <aut_increment>
	Хто відправив	varchar(100) <NULL>
	Код виробника	int(11) unsigned <NULL>
	Кількість відправлень	varchar(50) <NULL>
	Місце відправки товару	varchar(255) <NULL>
	Дата відправки	date <NULL>

Логічна модель є основою бази даних, вона повинна відображати взаємозв'язки між реляційними таблицями. Між реляційними таблицями можуть

бути наступні типи зв'язків 1:1, 1:Б та Б:Б. Найбільшим поширеним зв'язком є зв'язок 1:Б.

Для створення зв'язку необхідно, щоб у головній таблиці були визначенні первинні ключі. Встановлення первинного ключа для зв'язаної (підпорядкованої) таблиці не є обов'язковою умовою. Для підпорядкованої таблиці необхідно визначити поле вторинного ключа, тип даних і розмір якого повинні збігатись з полем первинного ключа головної таблиці. Імена полів первинного та вторинного ключів, між якими встановлюється зв'язок можуть не збігатися. Вторинні ключі відрізняються від первинних тим, що для них допускається дублювання значень.

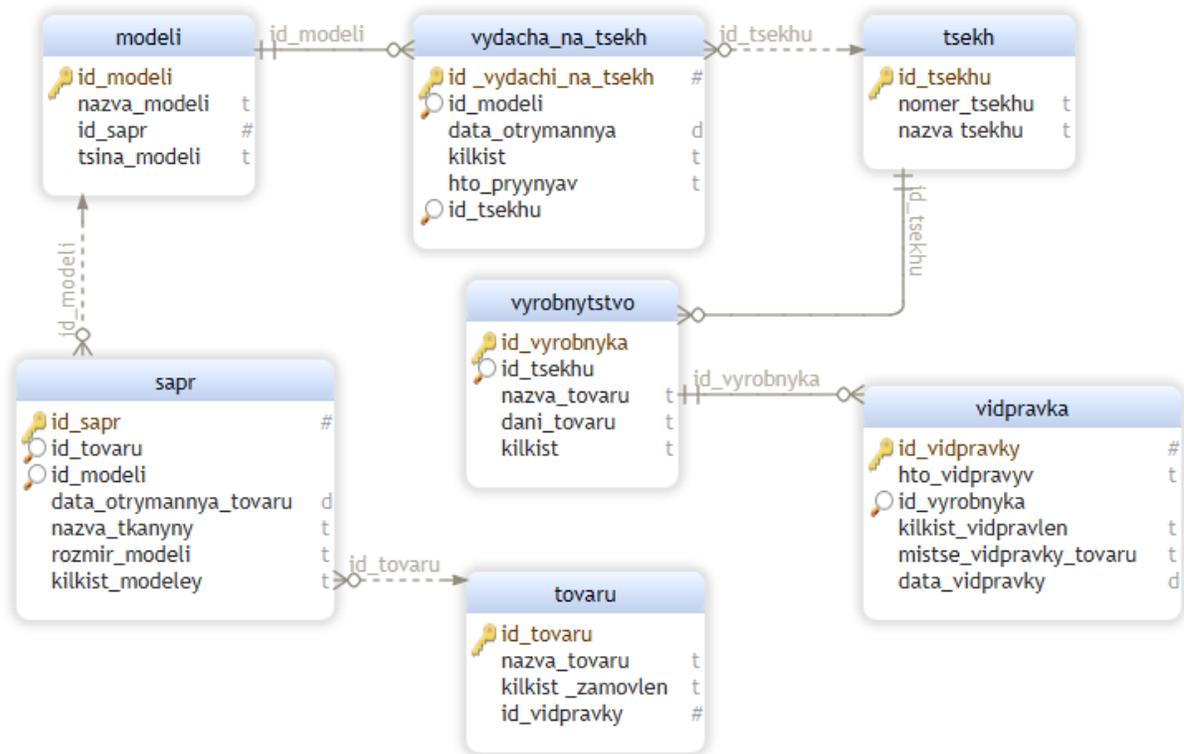


Рис. 2.1 Логічна модель даних АІС «Обліку товарів»

### 2.3 База даних інформаційної системи

За допомогою програмних засобів для формування і керування базами даних: MySQL-Front, Navicat for MySQL було розроблено інформаційна систему «Обліку товарів з виробництва верхнього одягу» яка налічує сім таблиць:

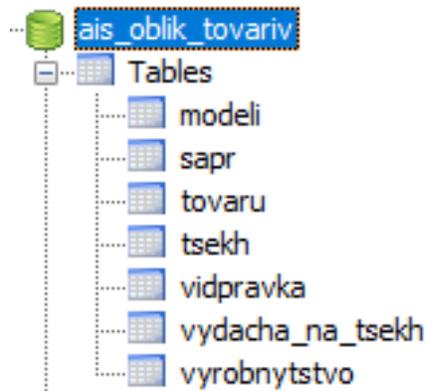


Рис. 2.2 Таблиці БД

id_modeli	nazva_modeli	id_sapr	tsina_modeli
1	Бежевий тренчкот		15 000 тисяч грн.
2	пальто «оверсайз»		12 500 тисяч грн.
3	Честерфильд		12 000 тисяч грн.
4	Коверкот		10 000 тисяч грн.
5	Поло		12 500 тисяч грн.
6	Балмакаан		10 000 тисяч грн.
7	Бушлат		12 500 тисяч грн.
8	Шинель		10 000 тисяч грн.
9	Пальто-халат		10 000 тисяч грн.
10	Кейп-пальто		11 000 тисяч грн.
11	Плащ в клетку		10 000 тисяч грн.
12	Двубортний плащ		10 000 тисяч грн.
13	Тренч		10 000 тисяч грн.
14	Френч		12 000 тисяч грн.
15	Парка		12 000 тисяч грн.

Рис. 2.3 Таблиця «Моделі»

Дана таблиця служить для вводу інформації даних про моделі для введення інформації використовують зручні форми, а в таблицях зберігається інформація. Таблиця містить такі поля як код моделі, назву моделі, код сапр та ціну моделі та зв'язана з таблицею «Видача на цех» за допомогою полів код моделі.

Таблиця «Моделі» створена за допомогою наступного SQL запиту та містить такі властивості полів:

```
CREATE TABLE `modeli` (
  `id_modeli` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `nazva_modeli` varchar(100) DEFAULT NULL,
  `id_sapr` int(11) unsigned DEFAULT NULL,
```

```

`tsina_modeli` varchar(100) DEFAULT NULL,
PRIMARY KEY (`id_modeli`),
CONSTRAINT `vydacha_na_tsekh` FOREIGN KEY (`id_modeli`)
REFERENCES `vydacha_na_tsekh` (`id_modeli`) ON DELETE CASCADE ON UPDATE
CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=17 DEFAULT CHARSET=utf8;

```

id_sapr	id_tovaru	id_modeli	data_otrymannya_tovaru	nazva_tkanyny	rozmir_modeli	kilkist_modeley
1	1	1	2022-01-03	7 рулона кашемиру	42-SM	15 шт.
2	2	2	2022-01-17	5 рулона драп	44 -ML	10 шт.
3	3	3	2022-01-31	5 рулона твід	46-ML	10 шт.
4	4	4	2022-02-14	7 рулона букле	48-XL	15 шт.
5	5	5	2022-02-28	5 рулона габардин	50-XL	10 шт.
6	6	6	2022-03-21	16 рулона драп	52-XXL	20 шт.
7	7	7	2022-04-04	8 рулона кашемиру	54-XXL	15 шт.
8	8	8	2022-04-18	8 рулона нейлон	56-3XL	10 шт.
9	9	9	2022-05-02	8 рулона твід	58-3XL	10 шт.
10	10	10	2022-05-16	8 рулона букле	44-ML	15 шт.
11	11	11	2021-12-20	10 рулона кашемиру	48-XL	10 шт.
12	12	12	2021-12-06	12 рулона драп	50-XL	10 шт.
13	13	13	2021-11-29	8 рулона бондинг	48	10 шт.
14	14	14	2021-11-15	12 рулона кашемиру	44	20 шт.
15	15	15	2021-10-25	14 рулона поларфліс	50	20 шт.

Рис. 2.4 Таблиця «САПР»

Таблиця «САПР» містить введення інформації використовують зручні форми, а в таблицях зберігається інформація. В таблиці є такі поля: код сапр, код товару, код моделі, дата отримання товару, назва тканини в рулонах, розмір моделі, кількість моделей яка дана для пошиття. Дана таблиця зв'язана з таблицею «Моделей» та «Товру» за допомогою полів код товару та код моделі.

Таблиця «САПР» створена за допомогою наступного SQL запити та містить такі властивості полів:

```

CREATE TABLE `sapr` (
  `id_sapr` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_tovaru` int(11) unsigned DEFAULT NULL,
  `id_modeli` int(11) unsigned DEFAULT NULL,
  `data_otrymannya_tovaru` date DEFAULT NULL,
  `nazva_tkanyny` varchar(100) DEFAULT NULL,

```

```

`rozmir_modeli` varchar(55) DEFAULT NULL,
`kilkist_modeley` varchar(50) DEFAULT NULL,
PRIMARY KEY (`id_sapr`),
KEY `tovaru` (`id_tovaru`),
KEY `modeli` (`id_modeli`),
CONSTRAINT `modeli` FOREIGN KEY (`id_modeli`) REFERENCES `modeli`
(`id_modeli`) ON UPDATE CASCADE,
CONSTRAINT `tovaru` FOREIGN KEY (`id_tovaru`) REFERENCES `tovaru`
(`id_tovaru`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

```

id_tovaru	nazva_tovaru	kilkist_zamovlen	id_vidpravky
1	Жіноче пальто	15 шт.	1
2	Жіноче пальто	10 шт.	2
3	Чоловыче пальто	10 шт.	3
4	Чоловыче пальто	15 шт.	4
5	Чоловыче пальто	10 шт.	5
6	Чоловыче пальто	20 шт.	6
7	Чоловыче пальто	15 шт.	7
8	Жіноче пальто	10 шт.	8
9	Жіноче пальто	10 шт.	9
10	Жіноче пальто	15 шт.	10
11	Чоловычий плащ	10 шт.	11
12	Чоловычий плащ	10 шт.	12
13	Жіночий плащ	10 шт.	13
14	Жіночий плащ	20 шт.	14
15	Жіночий плащ	20 шт.	15

Рис. 2.5 Таблица «Товари»

Таблица «Товари» служить для вводу інформації даних про: код товару, назву товару та кількість замовлень, код відправки. Для введення інформації використовують зручні форми, а в таблицях зберігається інформація.

Таблица створена за допомогою наступного SQL запиту та містить такі властивості полів:

```

CREATE TABLE `tovaru` (
  `id_tovaru` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `nazva_tovaru` varchar(100) DEFAULT NULL,

```

```

`kilkist_zamovlen` varchar(50) DEFAULT NULL,
`id_vidpravky` int(11) unsigned DEFAULT NULL,
PRIMARY KEY (`id_tovaru`)
) ENGINE=InnoDB AUTO_INCREMENT=16 DEFAULT CHARSET=utf8;

```

id_tsekhu	nomer_tsekhu	nazva tsekhu
1	№1	Швейний цех 1
2	№2	Швейний цех 2
3	№3	Швейний цех 3
4	№4	Експериментальний цех
5	№5	Розкрийний цех

Рис. 2.6 Таблиця «Цех»

Таблиця «Цех» служить для вводу інформації даних про номер цеху та його назву та містить такі поля: код цеху, номер меху, назва цеху, а також зв'язана з таблицею «Виробництво» за допомогою поля код цеху. Для введення інформації використовують зручні форми, а в таблицях зберігається інформація.

Таблиця «Цех» створена за допомогою наступного SQL запити та містить такі властивості полів:

```

CREATE TABLE `tsekh` (
  `id_tsekhu` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `nomer_tsekhu` varchar(50) DEFAULT NULL,
  `nazva tsekhu` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`id_tsekhu`),
  CONSTRAINT `vyrobnytstvo` FOREIGN KEY (`id_tsekhu`) REFERENCES
`vyrobnytstvo` (`id_tsekhu`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=6 DEFAULT CHARSET=utf8;

```

id_vidpravky	hto_vidpravyv	id_vyrobnika	kilkist_vidpravlen	mistse_vidpravky_tovaru	data_vidpravky
1	Чмут Ірина Миколаївна		1 30 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-01-10
2	Саванчук Ірина Олегівна		2 50 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-01-24
3	Чмут Ірина Миколаївна		3 30 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-02-07
4	Матяшук Наталія Миколаївна		4 50 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-02-21
5	Герасимчук Оксана Михайлівна		5 50 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-03-07
6	Леснік Аліна Ігорівна		6 50 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-03-28
7	Чмут Ірина Миколаївна		7 50 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-04-11
8	Саванчук Ірина Олегівна		8 45 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-04-25
9	Чмут Ірина Миколаївна		9 40 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-05-09
10	Матяшук Наталія Миколаївна		10 45 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2022-05-23
11	Матяшук Наталія Миколаївна		11 45 шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2021-12-27
12	Леснік Аліна Ігорівна		12 50шт.	Франція, м.Хазебрук, вулиця де Мервіль, 39	2021-12-13

Рис. 2.7 Таблиця «Відправка»

Таблиця «Відправка» служить для вводу інформації даних про відправку товару. Яка містить інформацію хто відправив даний товар, дату, кількість, місце відправки товару. Таблиця містить такі поля: код відправки, хто відправив, код виробника, кількість відправки, місце відправки товару, дата відправки товару. Для введення інформації використовують зручні форми, а в таблицях зберігається інформація.

Таблиця «Відправка» створена за допомогою наступного SQL запити та містить такі властивості полів:

```
CREATE TABLE `vidpravka` (
  `id_vidpravky` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `hto_vidpravyv` varchar(100) DEFAULT NULL,
  `id_vyrobnika` int(11) unsigned DEFAULT NULL,
  `kilkist_vidpravlen` varchar(50) DEFAULT NULL,
  `mistse_vidpravky_tovaru` varchar(255) DEFAULT NULL,
  `data_vidpravky` date DEFAULT NULL,
  PRIMARY KEY (`id_vidpravky`),
  KEY `id_vyrobnika` (`id_vyrobnika`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
```

id_vydachi_na_tsekh	id_modeli	data_otrymannya	kilkist	hto_pryynyav	id_tsekh
1	1	2021-12-31	10 шт.	Миронець Марина Андріївна	4
2	2	2022-01-14	10 шт.	Крупко Алла Сергіївна	4
3	3	2022-01-28	10 шт.	Сорока Марина Юрівна	4
4	4	2022-02-11	15 шт.	Сорока Юля Юрівна	5
5	5	2022-02-25	10 шт.	Крук Карина Андріївна	5
6	6	2022-03-11	10 шт.	Ковбар Ольга Анатоліївна	5
7	7	2022-03-25	15 шт.	Нестерчук Анастасія Андріївна	5
8	8	2022-04-08	10 шт.	Мельник Наталія Сергіївна	1
9	9	2022-04-22	10 шт.	Саванчук Ірина Олександрівна	1
10	10	2022-05-06	7 шт.	Лясковець Ольга Валентинівна	2
11	11	2022-05-20	10 шт.	Полейчук Марія Вікторівна	2
12	12	2021-11-05	10 шт.	Плескач Аліна Андріївна	2
13	13	2021-11-19	10 шт.	Сидорок Ірина Олегівна	3
14	14	2021-12-03	10 шт.	Гамза Світлана Олегівна	3
15	15	2021-12-17	10 шт.	Поправка Анна Ігорівна	4
16	16	2021-12-30	6 шт.	Федас Юлія Андріївна	4

Рис. 2.8 Таблиця «Видача на цех»

Таблиця «Видача на цех» служить для вводу інформації даних про дату отримання, кількість, хто прийняв та який цех. Та містить наступні поля: код видачі на цех, код моделі, дата отримання, кількість, хто прийняв, код цеху, а також зв'язана з таблицею «Цех» за допомогою поля код цеху. Для введення інформації використовують зручні форми, а в таблицях зберігається інформація.

Таблиця «Видача на цех» створена за допомогою наступного SQL запити та містить такі властивості полів:

```
CREATE TABLE `vydacha_na_tsekh` (
  `id_vydachi_na_tsekh` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_modeli` int(11) unsigned DEFAULT NULL,
  `data_otrymannya` date DEFAULT NULL,
  `kilkist` varchar(55) DEFAULT NULL,
  `hto_pryynyav` varchar(255) DEFAULT NULL,
  `id_tsekh` int(11) unsigned DEFAULT NULL,
  PRIMARY KEY (`id_vydachi_na_tsekh`),
  KEY `id_modeli` (`id_modeli`),
  KEY `tsekh` (`id_tsekh`),
  CONSTRAINT `tsekh` FOREIGN KEY (`id_tsekh`) REFERENCES `tsekh`
  (`id_tsekh`) ON DELETE CASCADE ON UPDATE CASCADE
```

) ENGINE=InnoDB AUTO\_INCREMENT=17 DEFAULT CHARSET=utf8;

id_vyrobnyka	id_tsekhu	nazva_tovaru	dani_tovaru	kilkist
1	1	Жіноче пальто	Бежевий тренчкот- жіноче пальто. Склад:35 % вискоза 35 % хлопок 30 % поліестер ; 100 % поліестер	10 шт.
2	2	Жіноче пальто	Пальто «оверсайз» - жіноче пальто. Склад:60%, поліамід - 40%.	10 шт.
3	3	Чоловиче пальто	Честерфілд (Chesterfield)-чоловіче пальто. Склад:вовна - 70%, поліамід - 30%.	10 шт.
4	4	Чоловиче пальто	Коверкот (Covert coat)-чоловіче пальто. Склад:97% поліестер 3% еластан	15 шт.
5	5	Чоловиче пальто	Поло (Polo)-чоловіче пальто. Склад:60% шерсть 40% поліестер	10 шт.
6	1	Чоловиче пальто	Балмакан (Balmacaan)-чоловіче пальто. Склад:90% поліестер 10% шерсть	10 шт.
7	2	Чоловиче пальто	Бушлат-чоловіче пальто. Склад:59% поліестер 41% шерсть	15 шт.
8	3	Жіноче пальто	Шинель-пальто жіноче. Склад: 55% поліестер 45% віскоза	10 шт.
9	1	Жіноче пальто	Пальто-халат, жіноче. Склад:100% поліестер	10 шт.
10	5	Жіноче пальто	Кейп-пальто жіноче. Склад:65% шерсть альпаки 20% поліамід 15% шерсть	7 шт.
11	5	Чоловичий плащ	Плащ в клетку-чоловічий. Склад:100% поліестер	10 шт.
12	5	Чоловичий плащ	Двубортний плащ-чоловічий. Склад:48% поліестер 46% поліамід 6% еластан	10 шт.

Рис. 2.9 Таблиця «Виробництво»

Таблиця «Виробництво» служить для вводу інформації даних про назву товару, дані про товар та кіскість. Та містить наступні поля: код виробника, код цеху, назва товару, тані товару, кількість, а також звязана з таблицею «Відправка» за допомогою поля код відправки. Для введення інформації використовують зручні форми, а в таблицях зберігається інформація.

Таблиця «Виробництво» створена за допомогою наступного SQL запиту та містить такі властивості полів:

```
CREATE TABLE `vyrobnytstvo` (
  `id_vyrobnyka` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_tsekhu` int(11) unsigned DEFAULT NULL,
  `nazva_tovaru` varchar(100) DEFAULT NULL,
  `dani_tovaru` varchar(255) DEFAULT NULL,
  `kilkist` varchar(50) DEFAULT NULL,
  PRIMARY KEY (`id_vyrobnyka`),
  KEY `id_tsekhu` (`id_tsekhu`),
  CONSTRAINT `vidpravka` FOREIGN KEY (`id_vyrobnyka`) REFERENCES
`vidpravka` (`id_vyrobnyka`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;
```

## РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЕКТУ

### 3.1 Довідники в ІС

За допомогою програмного забезпечення Visual Studio та NET Framework в якості платформи розробки через можливість сприйняття мови С#, великої бібліотеки класів і середовища виконання Windows Forms. Та за допомогою програмних продуктів для формування і керування базами даних, використані програми: MySQL-Front та Navicat for MySQL. Було розроблено автоматизовану інформаційну систему з «Обліку товарів з виробництва верхнього одягу. Яка містить головне меню: файл, довідники, моделі, звіт та довідка.

Запустивши проект на головному меню додатку, користувачу доступні довідники:

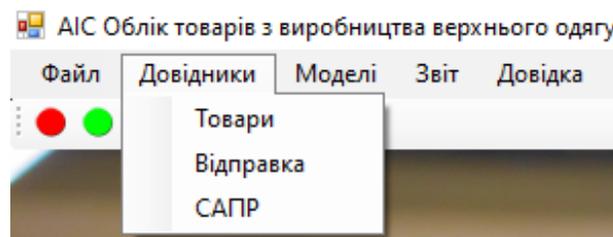


Рис. 3.1 Довідники ІС

Для забезпечте відкриття таблиць:

- Товари;
- Відправка;
- САПР.

Потрібно у модальному режимі після натиснення на пункт головного меню «Довідники» виділіть пункт таблиці яку потрібно відкрити написати наступний програмний код:

```
private void товаруToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 newForm = new Form2();
    newForm.ShowDialog();
}
```

```

private void відправкаToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form3 newForm = new Form3();
    newForm.ShowDialog();
}
private void canpToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 newForm = new Form4();
    newForm.ShowDialog();
}
}

```

Код товару	Назва товару	Кількість замовлень
8	Жіноче пальто	10 шт.
1	Жіноче пальто	15 шт.
2	Жіноче пальто	10 шт.
15	Жіночий плащ	20 шт.
14	Жіночий плащ	20 шт.
13	Жіночий плащ	10 шт.
5	Чоловоче пальто	10 шт.
3	Чоловоче пальто	10 шт.
4	Чоловоче пальто	15 шт.
7	Чоловоче пальто	15 шт.
6	Чоловоче пальто	20 шт.

Код товару:

Назва товару:

Кількість замовлень:

Рис. 3.2 Довідник «Товари»

Довідник «Товари» дозволяє переглядати дані про товар, також адміністратор може редагувати дані, додавати нові та видаляти. Для відображення даних в таблиці створений за допомогою SQL запиту, а також можна див. Додаток (Form2):

```

public Form2()
{
    InitializeComponent();
}

```

```
query = @"SELECT
tovaru.id_tovaru,
tovaru.nazva_tovaru,
tovaru.`kilkist_zamovlen`
FROM
tovaru
ORDER BY
tovaru.nazva_tovaru ASC";
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "tovaru");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "tovaru";
dataGridView1.Columns[0].HeaderText = "Код товару";
dataGridView1.Columns[1].HeaderText = "Назва товару";
dataGridView1.Columns[2].HeaderText = "Кількість замовлень ";
dataGridView1.Columns[0].Width = 150;
dataGridView1.Columns[1].Width = 300;
dataGridView1.Refresh();
}
```

Код відправки	Хто відправив ПІБ	Кількість відправлень	Місце відправки товару
5	Герасимчук Оксана Михайлівна	50 шт.	Франція, м....
6	Леснік Аліна Ігорівна	50 шт.	Франція, м....
12	Леснік Аліна Ігорівна	50шт.	Франція, м....
4	Матяшук Наталія Миколаївна	50 шт.	Франція, м....
10	Матяшук Наталія Миколаївна	45 шт.	Франція, м....
11	Матяшук Наталія Миколаївна	45 шт.	Франція, м....
2	Саванчук Ірина Олегівна	50 шт.	Франція, м....
8	Саванчук Ірина Олегівна	45 шт.	Франція, м....
1	Чмут Ірина Миколаївна	30 шт.	Франція, м....
3	Чмут Ірина Миколаївна	30 шт.	Франція, м....
7	Чмут Ірина Миколаївна	50 шт.	Франція, м....

Код відправки:

Хто відправив ПІБ:

Кількість відправлень:

Місце відправки товару:

Дата відправки : 31 травня 2022 р.

Рис. 3.3 Довідник «Відправка»

Довідник «Відправка» дозволяє переглядати дані про відправку, також адміністратор може редагувати дані, додавати нові та видаляти. Для відображення даних в таблиці створений за допомогою SQL запиту, а також можна див. Додаток (Form3):

```
public Form3()
{
    InitializeComponent();
    query = @"SELECT
    vidpravka.id_vidpravky,
    vidpravka.hto_vidpravyv,
    vidpravka.kilkist_vidpravlen,
    vidpravka.mistse_vidpravky_tovaru,
    vidpravka.data_vidpravky
    FROM
    vidpravka
```

*ORDER BY*

*vidpravka.hto\_vidpravyyv ASC";*

*MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);*

*DataSet ds = new DataSet();*

*da.Fill(ds, "vidpravka");*

*dataGridView1.DataSource = ds;*

*dataGridView1.DataMember = "vidpravka";*

*dataGridView1.Columns[0].HeaderText = "Код відправки";*

*dataGridView1.Columns[1].HeaderText = "Хто відправив ПИБ";*

*dataGridView1.Columns[2].HeaderText = "Кількість відправлень ";*

*dataGridView1.Columns[3].HeaderText = "Місце відправки товару ";*

*dataGridView1.Columns[4].HeaderText = "Дата відправки";*

*dataGridView1.Columns[0].Width = 150;*

*dataGridView1.Columns[1].Width = 300;*

*dataGridView1.Refresh();}*

Виберіть назву товару:

	Код сапр	Назва товару	Назва тканини	Розмір моделі	Кількість моеделей	id
▶	4	Чоловче пальто	7 рулона букле	48-XL	15 шт.	4
*						

Назва товару:

Назва тканини:

Розмір моделі:

Кількість моделей:

Рис. 3.4 Довідник «САПР»

Довідник «САПР» дозволяє переглядати інформацію про товар, задаючи назву товару, також адміністратор може редагувати дані, додавати нові та видаляти.

Для відображення даних в таблиці створений за допомогою SQL запити, також можна див. Додаток (Form4):

```
private void comboBox1_SelectionChangeCommitted(object sender, EventArgs e)
{
    query = @"SELECT
sapr.id_sapr,
tovaru.nazva_tovaru,
sapr.nazva_tkanyny,
sapr.rozmir_modeli,
sapr.kilkist_modeley,
```

```

    tovaru.id_tovaru
FROM
    tovaru
Inner Join sapr ON tovaru.id_tovaru = sapr.id_tovaru
WHERE
    tovaru.id_tovaru = " + Convert.ToString(comboBox1.SelectedValue);
query = query + @" ORDER BY
    sapr.nazva_tkanyny ASC";
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "sapr");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "sapr";
dataGridView1.Columns[0].HeaderText = "Код сарп";
dataGridView1.Columns[1].HeaderText = "Назва товару";
dataGridView1.Columns[2].HeaderText = "Назва тканини";
dataGridView1.Columns[3].HeaderText = "Розмір моделі";
dataGridView1.Columns[4].HeaderText = "Кількість моеделей";
dataGridView1.Columns[4].Width = 150;
dataGridView1.Refresh();
}

```

Отже, було створено три довідника для перегляду інформації про товар та для редагування даних адміністратором.

### 3.2 Функціональні можливості системи

Також ми можемо отримати інформацію про моделі та про видачу товару на цехи.

The screenshot shows a window titled 'Моделі' with a table of models and a form below it. The table has the following data:

Код моделі	Назва моделі	Ціна моделі	Код сапр
1	Бежевий тренчкот	15 000 тисяч грн.	1
2	пальто «оверсайз»	12 500 тисяч грн.	2
3	Честерфилд	12 000 тисяч грн.	3
4	Коверкот	10 000 тисяч грн.	4
5	Поло	12 500 тисяч грн.	5

Below the table, there are input fields for 'Назва моделі:', 'Ціна моделі:', and 'Код сапр:'. There are two buttons: 'Зберегти' (Save) and 'Відміна' (Cancel). Below these is a large grey rectangular area. At the bottom, there are more input fields: 'Код моделі:', 'Хто прийняв ПІБ:', 'Дата отримання:' (with a date picker showing '31 травня 2022 р.'), 'Код цеху:', and 'Кількість:'. There is also a 'Цех' button. At the very bottom, there are two more buttons: 'Зберегти' and 'Відміна'.

Рис. 3.5 Інформація про моделі

Для відображення даних в таблиці «Моделі» створений за допомогою SQL запиту, а також можна див. Додаток (Form5):

```
private void Form5_Shown(object sender, EventArgs e)
```

```
{
    insert_mode = true;
    query = @"SELECT
    modeli.id_modeli,
    modeli.nazva_modeli,
    modeli.tsina_modeli,
    modeli.id_sapr
    FROM
    modeli
    ORDER BY
    modeli.id_modeli ASC";
```

```

MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "modeli");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "modeli";
dataGridView1.Columns[0].HeaderText = "Код моделі";
dataGridView1.Columns[1].HeaderText = "Назва моделі";
dataGridView1.Columns[2].HeaderText = "Ціна моделі";
dataGridView1.Columns[3].HeaderText = "Код сапр";
dataGridView1.Columns[1].Width = 250;
dataGridView1.Refresh();
}

```

Натиснувши на відповідну модель, ми можемо отримати додаткову інформацію про видачу товару на цехи:

The screenshot shows a window titled "Моделі" with the following content:

Код моделі	Назва моделі	Ціна моделі	Код сапр
1	Бежевий тренкот	15 000 тисяч грн.	1
2	пальто «оверсайз»	12 500 тисяч грн.	2
3	Честерфільд	12 000 тисяч грн.	3
4	Коверкот	10 000 тисяч грн.	4
5	Поло	12 500 тисяч грн.	5

Below the table are input fields for "Назва моделі:", "Ціна моделі:", and "Код сапр:". There are "Зберегти" and "Відміна" buttons.

Код видачі на цех	Дата отримання	Кількість	Хто прийняв ПІБ	Код цеху
2	14.01.2022	10 шт.	Крупко Алла Се...	4
*				

Below this table are input fields for "Код моделі:", "Хто прийняв ПІБ:", "Дата отримання:" (with a date picker set to 31 травня 2022 р.), and "Код цеху:". There are also "Зберегти" and "Відміна" buttons.

Рис. 3.6 Інформація про видачу на цехи

Для відображення даних в таблиці «Видача на цех» при натисканні на назву товару був створений SQL запит, а також можна див. Додаток (Form5):

```
private void dataGridView1_Click(object sender, EventArgs e)
{
    string query3;
    {
        string data_b;
        DateTime data1 = dateTimePicker1.Value;
        data_b = data1.ToString("yyyy-MM-dd");
        query3 = @"SELECT
vydacha_na_tsekh.`id_vydachi_na_tsekh`,
vydacha_na_tsekh.id_modeli,
vydacha_na_tsekh.data_otrymannya,
vydacha_na_tsekh.kilkist,
vydacha_na_tsekh.hto_pryynyav,
vydacha_na_tsekh.id_tsekhu
FROM
vydacha_na_tsekh
WHERE
vydacha_na_tsekh.id_modeli           =           "           +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        MySqlDataAdapter da = new MySqlDataAdapter(query3, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "vydacha_na_tsekh");
        dataGridView2.DataSource = ds;
        dataGridView2.DataMember = "vydacha_na_tsekh";
        dataGridView2.Columns[0].HeaderText = "Код видачі на цех";
    }
}
```

```

        dataGridView2.Columns[1].HeaderText = "Код моделі";
        dataGridView2.Columns[2].HeaderText = "Дата отримання";
        dataGridView2.Columns[3].HeaderText = "Кількість";
        dataGridView2.Columns[4].HeaderText = "Хто трійняв ПІБ";
        dataGridView2.Columns[5].HeaderText = "Код цеху";
        dataGridView2.Columns[1].Visible = false;
        dataGridView2.Refresh();
    }
}

```

Адміністратор може редагувати дані, додавати нові та видаляти. Також ми можемо побачити інформацію про цехи які є і назву цеху, натиснувши на кнопку цех на формі рис 3.6 «Інформація про видачу на цехи».

Для відображення даних в таблиці «Цехи» на формі створений за допомогою SQL запиту, а також можна див. Додаток (Form6):

```

private void Form6_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    query = @"SELECT
tsekh.id_tsekhu,
tsekh.nomer_tsekhu,
tsekh.nazva_tsekhu
FROM
tsekh
ORDER BY
tsekh.nazva_tsekhu ASC";
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "tsekh");
}

```

```

dataGridView1.DataSource = ds;
dataGridView1.DataMember = "tsekh";
dataGridView1.Columns[0].HeaderText = "Код цеху";
dataGridView1.Columns[1].HeaderText = "Номер цеху";
dataGridView1.Columns[2].HeaderText = "Назва цеху";
dataGridView1.Columns[0].Width = 130;
dataGridView1.Columns[1].Width = 200;
dataGridView1.Columns[2].Width = 250;
dataGridView1.Refresh();
}

```

	Код цеху	Номер цеху	Назва цеху
	4	№4	Експериментальний цех
▶	5	№5	Розкрійний цех
	2	№2	Швейний цех 2
	3	№3	Швейний цех 3
	1	№1	Швейний цех1
*			

Рис. 3.7 Таблиця «Цех» на тиснувши на кнопку на формі «Інформація видача на цехи»

Можемо побачити інформацію про виробників в таблиці «Виробництво», натиснувши на пункт контекстного меню таблиці Цехи на формі рис 3.8:

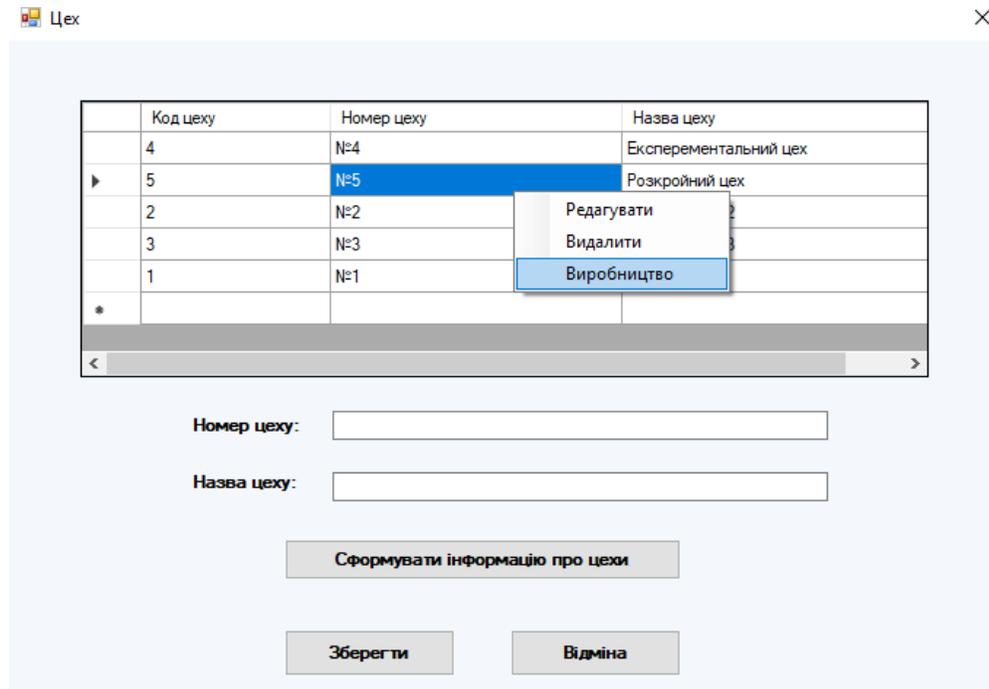


Рис. 3.8 Пункт контекстного меню таблиці «Цехи»

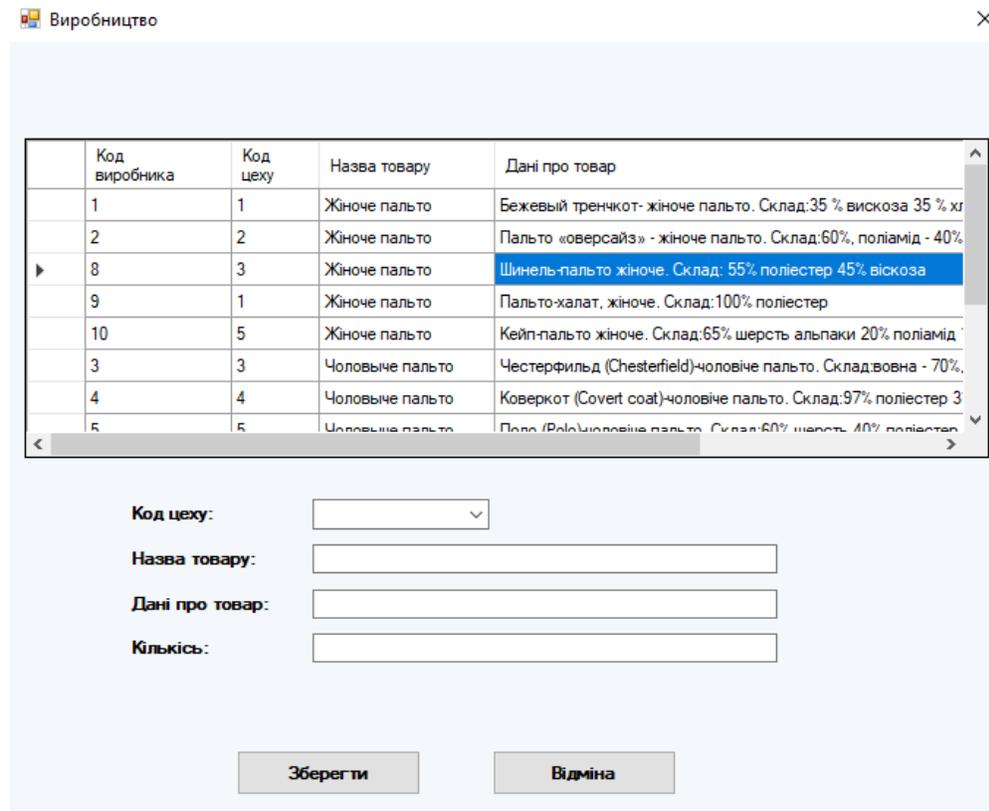


Рис. 3.9 Таблиця «Виробництво»

Ми можемо переглянути інформацію про виробництво про назву товару, дані про тавар і в якому цеху даний товар шиється.

Для відображення даних в таблиці «Виробництво» створений за допомогою SQL запиту, а також можна див. Додаток (Form7):

```
private void Form7_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    query = @"SELECT
vyrobnytstvo.id_vyrobnyka,
vyrobnytstvo.id_tsekhu,
vyrobnytstvo.nazva_tovaru,
vyrobnytstvo.dani_tovaru,
vyrobnytstvo.kilkist
FROM
vyrobnytstvo
ORDER BY
vyrobnytstvo.nazva_tovaru ASC";
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "vyrobnytstvo");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "vyrobnytstvo";
    dataGridView1.Columns[0].HeaderText = "Код виробника";
    dataGridView1.Columns[1].HeaderText = "Код цеху";
    dataGridView1.Columns[2].HeaderText = "Назва товару";
    dataGridView1.Columns[3].HeaderText = "Дані про товар";
    dataGridView1.Columns[4].HeaderText = "Кількість";
    dataGridView1.Columns[0].Width = 100;
```

```

dataGridView1.Columns[1].Width = 60;
dataGridView1.Columns[2].Width = 120;
dataGridView1.Columns[3].Width = 420;
dataGridView1.Columns[4].Width = 120;
dataGridView1.Refresh();
}

```

### 2.3 Аналіз вихідної інформації

На формі таблиці «Цех» див. рис. 3.8 розташована кнопка Сформувати інформацію про цехи, натиснувши на неї формується інформація про цех який натиснули:



Рис. 3.10 Інформація про цех– звіт з Microsoft Word

Для відображення даних в звіті Microsoft Word на кнопку «Сформувати інформацію про цех» є наступний програмний код :

```

private void button3_Click(object sender, EventArgs e)
{
    //Копіювання файлів
}

```

```

string fileName = "tsekh.docx";
string fileName2 = "tsekh" +
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value) + ".docx";
string sourcePath = Environment.CurrentDirectory + @"\"rep_copy";
string targetPath = Environment.CurrentDirectory + @"\"rep";
string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
string destFile = System.IO.Path.Combine(targetPath, fileName2);
System.IO.File.Copy(sourceFile, destFile, true);
//Завершення копіювання
string filename = Environment.CurrentDirectory + @"\"rep\" +
fileName2;
Microsoft.Office.Interop.Word.Application word = new
Microsoft.Office.Interop.Word.Application();
Microsoft.Office.Interop.Word.Document doc = new
Microsoft.Office.Interop.Word.Document();
word.Visible = false;
doc = word.Documents.Open(filename);
doc.Activate();
Microsoft.Office.Interop.Word.Range rng =
doc.Bookmarks["nomer_tsekhu"].Range;//Закладка що попередньо створена у Word
rng.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
Microsoft.Office.Interop.Word.Range rng2 =
doc.Bookmarks["nazva_tsekhu"].Range;//Закладка що попередньо створена у Word
rng2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
word.Visible = true;
}

```

Також ми можемо сформувати певний звіт вибравши пункт головного меню «Звіт»\Звіт про моделі:

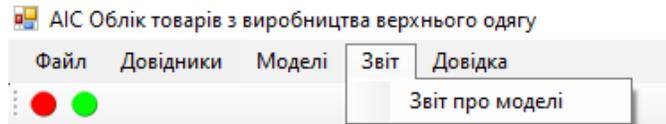


Рис.3.11.Звіт

№ з/п	Назва моделі	Ціна моделі
1	Бежевий тренчкот	15 000 тисяч грн.
2	пальто «оверсайз»	12 500 тисяч грн.
3	Честерфільд	12 000 тисяч грн.
4	Коверкот	10 000 тисяч грн.
5	Поло	12 500 тисяч грн.
6	Балмакаан	10 000 тисяч грн.
7		
8		
9		
10		

Рис. 3.12 Звіт про моделі і ціну в Microsoft Excel

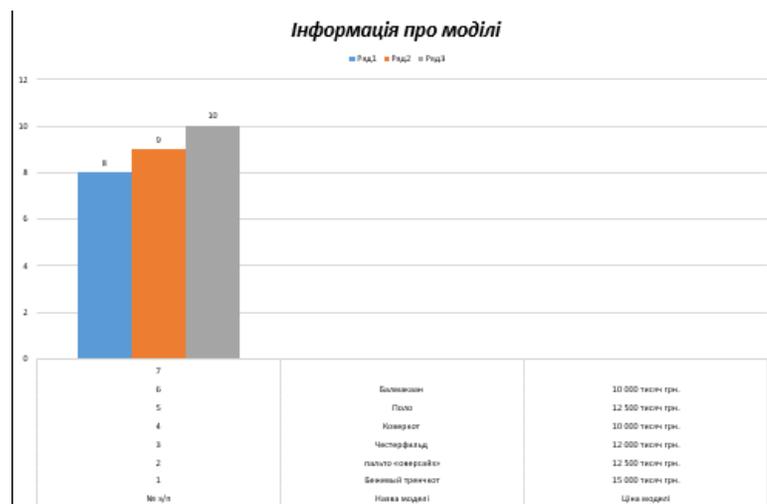


Рис. 3.12. Гістаграма про моделі

Для відображення даних в звіті Microsoft Excel на кнопку «Звіт про моделі» створений наступний програмний код:

```

private void звимПроМоделіToolStripMenuItem_Click(object sender, EventArgs
e)
{
    //копіювання файлів
    string fileName = "informatsiyamodeli.xlsx";
    string sourcePath = Environment.CurrentDirectory + @"\"rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\"rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName);
    System.IO.File.Copy(sourceFile, destFile, true);
    //завершення копіювання
    Excel.Application app = new Excel.Application();
    app.Visible = false;
    string filename = Environment.CurrentDirectory +
@"\"rep\"informatsiyamodeli.xlsx";
    Excel.Workbook wb = app.Workbooks.Open(filename,
Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing, Type.Missing, Type.Missing,
Type.Missing, Type.Missing);
    Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];
    //формуємо таблицю з даними
    string query2 = @"SELECT
        modeli.nazva_modeli,
        modeli.tsina_modeli
    FROM
        modeli";
    int i = 8;

```

```

using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query2,
Form1.con))
{
    DataTable table = new DataTable();
    table.Locale = System.Globalization.CultureInfo.InvariantCulture;
    da_sp.Fill(table);
    foreach (DataRow row in table.Rows)
    {
        ws.Cells[i, 2] = row["nazva_modeli"].ToString();
        ws.Cells[i, 3] = row["tsina_modeli"].ToString();
        i = i + 1;
    }
}

app.Visible = true;
}

```

Отже, за допомогою інформаційної системи можна сформувати звіти за якими можемо отримати інформацію про цех та моделі.

### 3.4 Інтерфейс користувача

Після запуску додатку відкривається головна форма:



Рис. 3.14 Головна форма

Пункт головного меню «Файл» забезпечую «Вхід в систему» та «Вихід»:

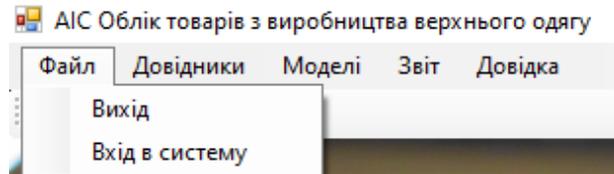


Рис. 3.14 Пункт головного меню Файл

Щоб у вийти в систему нам потрібно вести користувача, пароль та сервер для цього нам потрібно прописати наступний код в пункті головного меню «Файл» пункт підменю «Вхід в систему»:

```
private void button1_Click(object sender, EventArgs e)
{
    Form1.con.Close();
    string user_name = textBox1.Text;
    string user_password = textBox2.Text;
    string user_connection_string;
    string user_server = textBox3.Text;
    if (user_password == "")
    {
        user_connection_string = "server=" + user_server + ";User Id=" +
user_name + ";Persist Security Info=True;database=ais_oblik_tovariv";
    }
    else
    {
        user_connection_string = "server=" + user_server + ";User Id=" +
user_name + ";password=" + user_password + ";Persist Security
Info=True;database=ais_oblik_tovariv";
    }
    try
    {
```

```
Form1.con.ConnectionString = user_connection_string;  
Form1.con.Open();  
MessageBox.Show("Підключення відбулось успішно",  
"Підключення", MessageBoxButtons.OK, MessageBoxIcon.Information);  
this.Close();  
}  
catch  
{  
    MessageBox.Show("Помилка при підключенні", "Увага",  
MessageBoxButtons.OK, MessageBoxIcon.Question);  
}
```

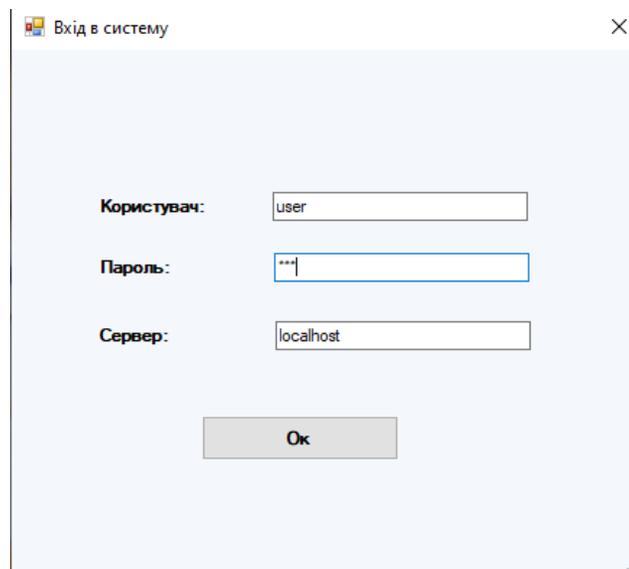


Рис. 3.15 Вхід для директора

Оскільки адміністратор може редагувати, додавати та видаляти записи, він заходить під іншим логіном:

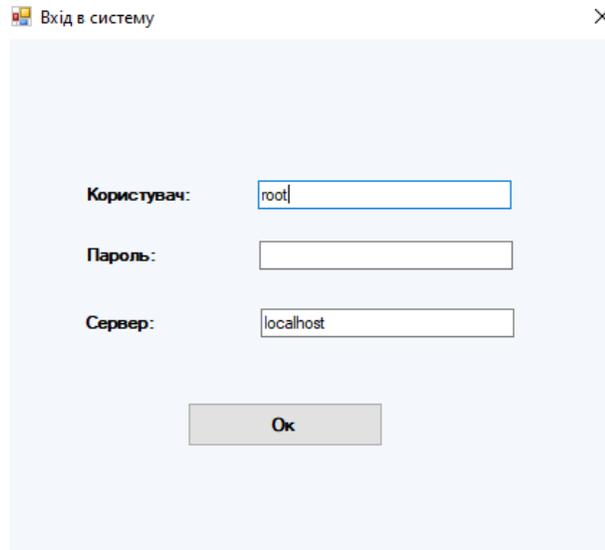


Рис. 3.16 Вхід для адміністратора

Також ми можемо побачити інформацію про розробника проекту, натиснувши на пункт головного меню «Довідка»:

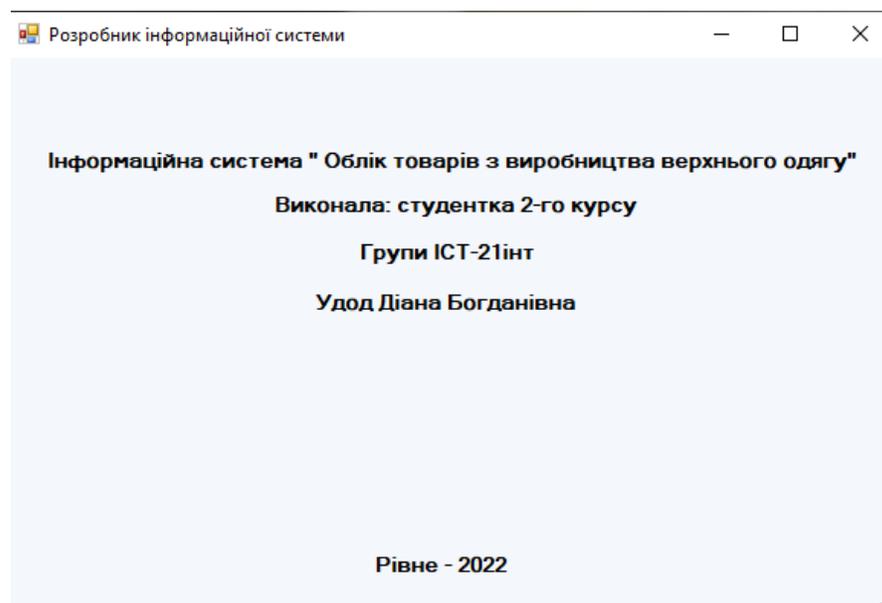


Рис. 3.17 Інформація про розробника

Отже, під час розробки інтерфейсу ІС «Облік товарів з виробництва верхнього одягу» було створено головне меню, за допомогою якого можна швидко отримати доступ до форм із даними та забезпечена можливість входу для директора та адміністратора.

## ВИСНОВОК

В ході виконання даної роботи було розроблено автоматизовану інформаційну систему для обліку товарів з виробництва верхнього одягу.

Автоматизована інформаційна система (АІС) - це комплекс, який включає комп'ютерне і комунікаційне обладнання, програмне забезпечення, лінгвістичні засоби, інформаційні ресурси, а також системний персонал, що забезпечує підтримку динамічної інформаційної моделі деякої частини реального світу для задоволення інформаційних потреб користувачів і для прийняття рішень [9].

Додаток було написано в програмі Visual Studio на мові програмування С# з використанням графічного інтерфейсу Windows Forms та програмними засобами для формування і керування базами даних: MySQL-Front; Navicat for MySQL;

С# - це мова програмування створена спеціально для роботи у середовищі Microsoft .NET Framework [13].

В ході роботи було проведено огляд та зроблено аналіз засобів, що були використані для створення даного програмного забезпечення (середовища розробки Visual Studio 2017, інтерфейсу Windows Forms, MySQL-Front, Navicat for MySQL та відповідних бібліотек мови С#).

Результатом виконаної роботи є desktop-додаток, обліку товарів верхнього одягу.

При проектуванні системи були визначені засоби розробки. Аналіз цих засобів допоміг встановити переваги їх розробки.

Спроектвана система допоможе скоротити час роботи і підвищити продуктивність праці в роботі швейного виробництва по контролю виробництва.

Розроблена система є дуже зручним доповненням для роботи з великими обсягами даних з товарами

Даний продукт дозволяє директору, переглядати інформацію про облік товарів.

Адміністратори можуть керувати основними складовими мережі. Вони зможуть переглянути інформацію, та додати інші моделі товару, а також з легкістю редагувати будь-які дані бази.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Database Administrator's Companion: Microsoft SQL Server.- Microsoft Corporation, 1998.- с. xi-35.
2. Jeff Prosis Programming Microsoft .NET – Microsoft Press Redmond, WA, USA, 2002. – 800с
3. Mickey Williams Microsoft Visual C# (Core Reference) – Microsoft Press Redmond, WA, USA, 2002. – 750с.
4. Microsoft DataGridView Class URL адреса: DataGridView Class (System.Windows.Forms) | Microsoft Docs (дата звернення на сайт 20.05.2022).
5. MySQL/Посібник для початківців URL адреса: [https://wiki.gentoo.org/wiki/MySQL/Startup\\_Guide/ru#.D0.9D.D0.B0.D1.87.D0.B0.D0.BB.D0.BE\\_.D1.80.D0.B0.D0.B1.D0.BE.D1.82.D1.8B\\_.D1.81\\_MySQL](https://wiki.gentoo.org/wiki/MySQL/Startup_Guide/ru#.D0.9D.D0.B0.D1.87.D0.B0.D0.BB.D0.BE_.D1.80.D0.B0.D0.B1.D0.BE.D1.82.D1.8B_.D1.81_MySQL) (дата звернення на сайт 1.04.2022).
6. SQL база даних. Для чого призначена база даних? URL адреса:<https://www.ukraine.com.ua/uk/blog/programming/sql-baza-dannih-dlya-chego-prednaznachena-baza-dannih.html>(дата звернення на сайт 20.05.2022).
7. SQL в Access: основні поняття, глосарій і синтаксисURL адреса: <https://support.microsoft.com/uk-ua/office/sql-в-access-основні-поняття-глосарій-і-синтаксис-444d0303-cde1-424e-9a74-e8dc3e460671> (дата звернення на сайт 10.04.2022).
8. SQL-Урок 1. Мова SQL. Основні поняття. URL адреса: [http://moonexcel.com.ua/уроки-sql1-мова-основні-поняття\\_ua](http://moonexcel.com.ua/уроки-sql1-мова-основні-поняття_ua) (дата звернення на сайт 10.04.2022).
9. Автоматизована система URL адреса:<https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%BE%D0%BC%D0%B0%D1%82%D0%B8%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%>

B0\_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0 (дата звернення на сайт 15.03.2022).

10. Автоматизовані інформаційні системи для підприємств та організацій.URL

адреса:[https://pidru4niki.com/12370107/informatika/avtomatizovani\\_informatsiyni\\_sistemi\\_dlya\\_pidpriyemstv\\_organizatsiy](https://pidru4niki.com/12370107/informatika/avtomatizovani_informatsiyni_sistemi_dlya_pidpriyemstv_organizatsiy) (дата звернення на сайт 17.03.2022).

11. Ахаян Р., Горев А., Макашарипов С. Эффективная работа с СУБД.- СПб.: Питер, 1997.- с. 100-105.

12. Вовчак І. С. Інформаційні системи та комп'ютерні технології в менеджменті: Навч. посібн. –Тернопіль: Карт-бланш, 2001.

13. Вступ в С#.URL адреса:<https://programm.top/uk/c-sharp/tutorial/introduction/> (дата звернення на сайт 2.04.2022).

14. Горев А., МакашариповС., Владимиров Ю. Microsoft SQL Server для профессионалов.- СПб.: Питер, 1998.- с.80-190.

15. Інформаційна технологія як основна складова ІС підприємства URL адреса: <https://studopedia.org/7-97423.html> (дата звернення на сайт 20.03.2022).

16. Класифікація інформаційних систем та інформаційних технологій URL адреса:[https://stud.com.ua/21185/informatika/klasifikatsiya\\_informatsiynih\\_sistem\\_informatsiynih\\_tehnologiy](https://stud.com.ua/21185/informatika/klasifikatsiya_informatsiynih_sistem_informatsiynih_tehnologiy) (дата звернення на сайт 20.03.2022).

17. Писаревська Т. А. Інформаційні системи обліку та аудиту: Навчальний посіб.— К.: КНЕУ, 2004. —с. 9 – 10.

18. Юринець В., Крупка М., Сажинець С., Смаль З. Основи інформаційних систем бухгалтерського обліку. – Львів: ЛНУ імені Івана Франка, 2002.

## ДОДАТКИ

Код головної форми Form1:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
using Excel = Microsoft.Office.Interop.Excel;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        static public MySqlConnection con = new MySqlConnection();
        MySqlCommand cmd;

        public Form1()
        {
            InitializeComponent();
            // try
            // {
            //     con.ConnectionString = "server=localhost;user
id=root;database=ais_oblik_tovariv";
            // con.Open();
            // MessageBox.Show("Підключення відбулось успішно");
            }
```

```

    //}
    //catch
    // {
    //     MessageBox.Show("Помилка при підключення");
    //}
}

```

```

private void вихідToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show
        ("Ви впевнені, що хочете завершити роботу?", "Вихід",
        MessageBoxButtons.YesNo, MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        Close(); // System.Environment.Exit(0);
    }
}

```

```

private void товариToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form2 newForm = new Form2();
    newForm.ShowDialog();
}

```

```

private void відправкаToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form3 newForm = new Form3();
    newForm.ShowDialog();
}

```

```

private void сапрToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form4 newForm = new Form4();
    newForm.ShowDialog();
}

```

```

private void конференціїToolStripMenuItem_Click(object sender, EventArgs e)
{

```

```

Form5 newForm = new Form5();
newForm.ShowDialog();
}

private void звітПроМоделіToolStripMenuItem_Click(object sender, EventArgs e)
{
    //копіювання файлів
    string fileName = "informatsiyamodeli.xlsx";
    string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
    string targetPath = Environment.CurrentDirectory + @"\rep";
    string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
    string destFile = System.IO.Path.Combine(targetPath, fileName);
    System.IO.File.Copy(sourceFile, destFile, true);
    //завершення копіювання
    Excel.Application app = new Excel.Application();
    app.Visible = false;
    string filename = Environment.CurrentDirectory +
    @"\rep\informatsiyamodeli.xlsx";
    Excel.Workbook wb = app.Workbooks.Open(filename,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing, Type.Missing, Type.Missing,
    Type.Missing, Type.Missing);
    Excel.Worksheet ws = (Excel.Worksheet)wb.Worksheets[1];
    //формуємо таблицю з даними
    string query2 = @"SELECT
        modeli.nazva_modeli,
        modeli.tsina_modeli
    FROM
        modeli";
    int i = 8;
    using (MySqlDataAdapter da_sp = new MySqlDataAdapter(query2,
    Form1.con))
    {
        DataTable table = new DataTable();
        table.Locale = System.Globalization.CultureInfo.InvariantCulture;
        da_sp.Fill(table);
        foreach (DataRow row in table.Rows)

```

```

        {
            ws.Cells[i, 2] = row["nazva_modeli"].ToString();
            ws.Cells[i, 3] = row["tsina_modeli"].ToString();
            i = i + 1;
        }
    }
    app.Visible = true;

}

private void файлToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form8 newForm = new Form8();
    newForm.ShowDialog();
}

private void розробникToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form9 newForm = new Form9();
    newForm.ShowDialog();
}
}
}

```

Код формы «Товари» Form2:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace WindowsFormsApp1

```

```

{
public partial class Form2 : Form
{
    MySqlCommand cmd;
    string query; // зміна для тексту запиту на SELECT
    Boolean insert_mode; // режим додавання нового запису

    public Form2()
    {
        InitializeComponent();
        query = @"SELECT
        tovaru.id_tovaru,
        tovaru.nazva_tovaru,
        tovaru.`kilkist_zamovlen`
        FROM
        tovaru
        ORDER BY
        tovaru.nazva_tovaru ASC";
        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "tovaru");
        dataGridView1.DataSource = ds;
        dataGridView1.DataMember = "tovaru";
        dataGridView1.Columns[0].HeaderText = "Код товару";
        dataGridView1.Columns[1].HeaderText = "Назва товару";
        dataGridView1.Columns[2].HeaderText = "Кількість замовлень ";
        dataGridView1.Columns[0].Width = 150;
        dataGridView1.Columns[1].Width = 300;
        dataGridView1.Refresh();
    }

    private void Form2_Shown(object sender, EventArgs e)
    {
        insert_mode = true;
    }

    private void button1_Click(object sender, EventArgs e)
    {

```

```

string query2;
if (insert_mode == true)
{
    query2 = "INSERT INTO `tovaru` (`nazva_tovaru`, `kilkist_zamovlen`)
VALUES (";
    query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox3.Text) + """); //
convert.ToInt64(text)

}
else
{
    query2 = "UPDATE `tovaru` SET";
    query2 = query2 + "`nazva_tovaru`=" + Convert.ToString(textBox2.Text) +
    ",";
    query2 = query2 + "`kilkist_zamovlen`=" +
Convert.ToString(textBox3.Text) + " ";
    query2 = query2 + " WHERE `id_tovaru`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);

}

cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
//Оновлюємо дані в таблиці dataGridView1
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "tovaru");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();
//Виділення даного запису
for (int i = 0; i < dataGridView1.RowCount; i++)
{
    dataGridView1.Rows[i].Selected = false;
}

```

```

        if
(dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))
        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }

```

```

    }
    //Очищення
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;

```

```

}

```

```

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        textBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox3.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

```

```

}

```

```

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{

```

```

        DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
        if (dialogResult == DialogResult.Yes)
        {
            try
            {
                string query2;
                query2 = "DELETE FROM `tovaru` WHERE `id_tovaru`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
                cmd = new MySqlCommand();
                cmd.Connection = Form1.con;
                cmd.CommandText = query2;
                cmd.CommandType = CommandType.Text;
                cmd.ExecuteNonQuery();

                MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
                DataSet ds = new DataSet();
                da.Fill(ds, "tovaru");
                dataGridView1.DataSource = ds;
                dataGridView1.Refresh();
                dataGridView1.Enabled = true;
                textBox1.Text = "";
                textBox2.Text = "";
                textBox3.Text = "";
                insert_mode = true;
            }
            catch

            {
                MessageBox.Show("Видалення неможливо. Товари має пов'язані
записи", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }

        }
    }
}
private void button2_Click(object sender, EventArgs e)

```

```

    {
        dataGridView1.Enabled = true;
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        insert_mode = true;
    }
}

```

### Код форми «Відправки» Form3:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace WindowsFormsApp1
{
    public partial class Form3 : Form
    {
        MySqlCommand cmd;
        string query; // зміна для тексту запиту на SELECT
        Boolean insert_mode; // режим додавання нового запису

        public Form3()
        {
            InitializeComponent();
            query = @"SELECT
            vidpravka.id_vidpravky,

```

```

vidpravka.hto_vidpravyyv,
vidpravka.kilkist_vidpravlen,
vidpravka.mistse_vidpravky_tovaru,
vidpravka.data_vidpravky
FROM
vidpravka
ORDER BY
vidpravka.hto_vidpravyyv ASC";
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "vidpravka");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = "vidpravka";
dataGridView1.Columns[0].HeaderText = "Код відправки";
dataGridView1.Columns[1].HeaderText = "Хто відправив ПІБ";
dataGridView1.Columns[2].HeaderText = "Кількість відправлень ";
dataGridView1.Columns[3].HeaderText = "Місце відправки товару ";
dataGridView1.Columns[4].HeaderText = "Дата відправки";
dataGridView1.Columns[0].Width = 150;
dataGridView1.Columns[1].Width = 300;
dataGridView1.Refresh();
}

private void Form3_Shown(object sender, EventArgs e)
{
    insert_mode = true;
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `vidpravka` (`hto_vidpravyyv`, `kilkist_vidpravlen`,
`mistse_vidpravky_tovaru`, `data_vidpravky`) VALUES (";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + ",";
    }
}

```

```

        query2 = query2 + "" + Convert.ToString(textBox4.Text) + """; //
convert.toint16(text)
        query2 = query2 + "" + dateTimePicker1.Value.ToString("yyyy-MM-dd") +
"",";

    }
    else
    {
        query2 = "UPDATE `vidpravka` SET";
        query2 = query2 + "`hto_vidpravvyv`=" + Convert.ToString(textBox2.Text) +
"",";

        query2 = query2 + "`kilkist_vidpravlen`=" +
Convert.ToString(textBox3.Text) + "" ";
        query2 = query2 + "`mistse_vidpravky_tovaru`=" +
Convert.ToString(textBox4.Text) + "" ";
        query2 = query2 + "`data_vidpravky`=" +
dateTimePicker1.Value.ToString("yyyy-MM-dd") + "",";
        query2 = query2 + " WHERE `id_vidpravky`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);

    }

    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "vidpravka");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();

    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        if
(dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox2.Text))

```

```

        {
            dataGridView1.Rows[i].Selected = true;

            break;
        }

    }

    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    textBox4.Text = "";
    insert_mode = true;

}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        textBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox3.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        textBox4.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[4].Value);

        dataGridView1.Enabled = false;
        insert_mode = false;

    }

}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `vidpravka` WHERE `id_vidpravky`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "vidpravka");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            textBox3.Text = "";
            textBox4.Text = "";
            insert_mode = true;

        }

        catch

        {
            MessageBox.Show("Видалення неможливо. Гуртожиток має пов`язані
записи", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

    }

    private void button2_Click(object sender, EventArgs e)
    {
        dataGridView1.Enabled = true;
        dataGridView1.Enabled = true;
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        textBox4.Text = "";
        insert_mode = true;
    }
}
}

```

Код форми «САПР» Form4:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace WindowsFormsApp1
{
    public partial class Form4 : Form
    {
        MySqlCommand cmd;
        string query; // зміна для тексту запиту на SELECT
        Boolean insert_mode; // режим додавання нового запису
        public Form4()
        {

```

```

        InitializeComponent();
    }

    private void Form4_Shown(object sender, EventArgs e)
    {
        insert_mode = true;
        string query2 = @"SELECT
        tovaru.id_tovaru,
        tovaru.nazva_tovaru
        FROM
        tovaru";
        using (MySqlDataAdapter da_kaf = new MySqlDataAdapter(query2,
Form1.con))
        {
            DataTable table = new DataTable();
            table.Locale = System.Globalization.CultureInfo.InvariantCulture;
            da_kaf.Fill(table);
            comboBox1.DataSource = table;
            comboBox1.DisplayMember = table.Columns["nazva_tovaru"].ColumnName;
            comboBox1.ValueMember = table.Columns["id_tovaru"].ColumnName;
            comboBox1.Text = "";
            DataTable table2 = new DataTable();
            table2.Locale = System.Globalization.CultureInfo.InvariantCulture;
            da_kaf.Fill(table2);
            comboBox2.DataSource = table2;
            comboBox2.DisplayMember =
table2.Columns["nazva_tovaru"].ColumnName;
            comboBox2.ValueMember = table2.Columns["id_tovaru"].ColumnName;
            comboBox2.Text = "";

        }

    }

    private void comboBox1_SelectionChangeCommitted(object sender, EventArgs e)
    {
        query = @"SELECT
        sapr.id_sapr,

```

```

    tovaru.nazva_tovaru,
    sapr.nazva_tkanyny,
    sapr.rozmir_modeli,
    sapr.kilkist_modeley,
    tovaru.id_tovaru
FROM
    tovaru
Inner Join sapr ON tovaru.id_tovaru = sapr.id_tovaru
WHERE
    tovaru.id_tovaru = " + Convert.ToString(comboBox1.SelectedValue);
query = query + @" ORDER BY
    sapr.nazva_tkanyny ASC";
MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, " sapr");
dataGridView1.DataSource = ds;
dataGridView1.DataMember = " sapr";
dataGridView1.Columns[0].HeaderText = "Код сапр";
dataGridView1.Columns[1].HeaderText = "Назва товару";
dataGridView1.Columns[2].HeaderText = "Назва тканити";
dataGridView1.Columns[3].HeaderText = "Розмір моделі";
dataGridView1.Columns[4].HeaderText = "Кількість моделей";
dataGridView1.Columns[4].Width = 150;
dataGridView1.Refresh();

}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `sapr` (`id_tovaru`, `nazva_tkanyny`,
`rozmir_modeli`, `kilkist_modeley`) VALUES (";
        query2 = query2 + "" + Convert.ToString(comboBox2.SelectedValue) + ",";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox3.Text) + """;
    }
}

```

```

    }
    else
    {
        query2 = "UPDATE `sapr` SET";
        query2 = query2 + "`id_tovaru`=" +
Convert.ToString(comboBox2.SelectedValue) + ",";
        query2 = query2 + "`nazva_tkanyny`=" + Convert.ToString(textBox1.Text) +
"";
        query2 = query2 + "`rozmir_modeli`=" + Convert.ToString(textBox2.Text) +
"";
        query2 = query2 + "`kilkist_modeley`=" + Convert.ToString(textBox3.Text)
+ "";
        query2 = query2 + " WHERE `id_sapr`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);

    }

    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "sapr");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();

    try
    {
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if
(dataGridView1.Rows[i].Cells[1].Value.ToString().Contains(textBox1.Text))
            {

```

```

        dataGridView1.Rows[i].Selected = true;

        break;
    }

}

catch
{
    dataGridView1.Rows[0].Selected = true;
}
comboBox1.Enabled = true;
dataGridView1.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";
comboBox2.Text = "";
insert_mode = true;
}

private void button2_Click(object sender, EventArgs e)
{
    comboBox1.Enabled = true;
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    comboBox2.Text = "";
    insert_mode = true;
}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        comboBox2.SelectedValue = dataGridView1.CurrentRow.Cells[5].Value;
    }
}

```

```

        comboBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox1.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        textBox3.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[4].Value);
        dataGridView1.Enabled = false;
        comboBox1.Enabled = false;
        insert_mode = false;

    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try
        {
            string query2;
            query2 = "DELETE FROM `sapr` WHERE `id_sapr`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();
            //Оновлюємо дані в таблиці dataGridView1
            MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "sapr");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
        }
    }
}

```



```
{
    MySqlCommand cmd;
    string query; // зміна для тексту запиту на SELECT
    Boolean insert_mode; // режим додавання нового запису
    public Form5()
    {
        InitializeComponent();
    }

    private void Form5_Shown(object sender, EventArgs e)
    {
        insert_mode = true;
        query = @"SELECT
        modeli.id_modeli,
        modeli.nazva_modeli,
        modeli.tsina_modeli,
        modeli.id_sapr
        FROM
        modeli
        ORDER BY
        modeli.id_modeli ASC";
        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "modeli");
        dataGridView1.DataSource = ds;
        dataGridView1.DataMember = "modeli";
        dataGridView1.Columns[0].HeaderText = "Код моделі";
        dataGridView1.Columns[1].HeaderText = "Назва моделі";
        dataGridView1.Columns[2].HeaderText = "Ціна моделі";
        dataGridView1.Columns[3].HeaderText = "Код сапр";
        dataGridView1.Columns[1].Width = 250;
        dataGridView1.Refresh();

    }

    private void button1_Click(object sender, EventArgs e)
    {
        string query2;
```

```

if (insert_mode == true)
{
    query2 = "INSERT INTO `modeli` (nazva_modeli, tsina_modeli, id_sapr)
VALUES (";
    query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox2.Text) + ",";
    query2 = query2 + "" + Convert.ToString(textBox3.Text) + " )";

}
else
{
    query2 = "UPDATE `modeli` SET";
    query2 = query2 + "`nazva_modeli`=" + Convert.ToString(textBox1.Text) +
",";
    query2 = query2 + "`tsina_modeli`=" + Convert.ToString(textBox2.Text) +
",";
    query2 = query2 + "`id_sapr`=" + Convert.ToString(textBox3.Text) + " ";
    query2 = query2 + " WHERE `id_modeli`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);

}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();

MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "modeli");
dataGridView1.DataSource = ds;
dataGridView1.Refresh();

dataGridView1.Enabled = true;
textBox1.Text = "";
textBox2.Text = "";
textBox3.Text = "";

```

```

insert_mode = true;

}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    textBox3.Text = "";
    insert_mode = true;
}

private void dataGridView1_Click(object sender, EventArgs e)
{
    string query3;
    {
        string data_b;
        DateTime data1 = dateTimePicker1.Value;
        data_b = data1.ToString("yyyy-MM-dd");

        query3 = @"SELECT
vydacha_na_tsekh.`id_vydachi_na_tsekh`,
vydacha_na_tsekh.id_modeli,
vydacha_na_tsekh.data_otrymannya,
vydacha_na_tsekh.kilkist,
vydacha_na_tsekh.hto_pryynyav,
vydacha_na_tsekh.id_tsekhu
FROM
vydacha_na_tsekh
WHERE
vydacha_na_tsekh.id_modeli = " +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        MySqlDataAdapter da = new MySqlDataAdapter(query3, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "vydacha_na_tsekh");
        dataGridView2.DataSource = ds;
        dataGridView2.DataMember = "vydacha_na_tsekh";
    }
}

```

```

        dataGridView2.Columns[0].HeaderText = "Код видачі на цех";
        dataGridView2.Columns[1].HeaderText = "Код моделі";
        dataGridView2.Columns[2].HeaderText = "Дата отримання";
        dataGridView2.Columns[3].HeaderText = "Кількість";
        dataGridView2.Columns[4].HeaderText = "Хто трійняв ПІБ";
        dataGridView2.Columns[5].HeaderText = "Код цеху";
        dataGridView2.Columns[1].Visible = false;
        dataGridView2.Refresh();
    }

}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        textBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox3.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        dataGridView1.Enabled = false;
        dataGridView2.Enabled = false;
        insert_mode = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)
    {
        try

```

```

    {
        string query2;
        query2 = "DELETE FROM `modeli` WHERE `id_modeli`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
        cmd = new MySqlCommand();
        cmd.Connection = Form1.con;
        cmd.CommandText = query2;
        cmd.CommandType = CommandType.Text;
        cmd.ExecuteNonQuery();

        MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
        DataSet ds = new DataSet();
        da.Fill(ds, "modeli");
        dataGridView1.DataSource = ds;
        dataGridView1.Refresh();

        dataGridView1.Enabled = true;
        textBox1.Text = "";
        textBox2.Text = "";
        textBox3.Text = "";
        insert_mode = true;
    }
    catch
    {
        MessageBox.Show("Видалення неможливо. Моделі має пов`язані
записи", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}
}

private void button3_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        string data_b;
        DateTime data1 = dateTimePicker1.Value;

```

```

data_b = data1.ToString("yyyy-MM-dd");

query2 = "INSERT INTO `vydacha_na_tsekh` (`id_modeli`,
data_otrymannya, kilkist, hto_pryynyav, id_tsekhu) VALUES (";

query2 = query2 + "" + Convert.ToString(textBox4.Text) + ",";
query2 = query2 + "" + datePicker1.Value.ToString("yyyy-MM-dd") +
", ";

query2 = query2 + "" + Convert.ToString(textBox5.Text) + ",";
query2 = query2 + "" + Convert.ToString(textBox6.Text) + ",";
query2 = query2 + " " + Convert.ToString(comboBox1.SelectedValue) + " )";

}
else
{
query2 = "UPDATE `vydacha_na_tsekh` SET";
query2 = query2 + "`id_modeli`=" + Convert.ToString(textBox4.Text) + ",";
query2 = query2 + "`data_otrymannya`=" +
datePicker1.Value.ToString("yyyy-MM-dd") + ",";
query2 = query2 + "`kilkist`=" + Convert.ToString(textBox5.Text) + ",";
query2 = query2 + "`hto_pryynyav`=" + Convert.ToString(textBox6.Text) +
", ";

query2 = query2 + "`id_tsekhu`=" +
Convert.ToString(comboBox1.SelectedValue) + " ";
query2 = query2 + " WHERE ``id_vydachi_na_tsekh``=" +
Convert.ToString(dataGridView2.CurrentRow.Cells[0].Value);

}
cmd = new MySqlCommand();
cmd.Connection = Form1.con;
cmd.CommandText = query2;
cmd.CommandType = CommandType.Text;
cmd.ExecuteNonQuery();
MySQLDataAdapter da = new MySQLDataAdapter(query, Form1.con);
DataSet ds = new DataSet();
da.Fill(ds, "vydacha_na_tsekh");
dataGridView2.DataSource = ds;
dataGridView2.Refresh();

```

```

dataGridView2.Enabled = true;
textBox4.Text = "";
dataGridView2.Enabled = true;
textBox5.Text = "";
textBox6.Text = "";
comboBox1.Text = "";
insert_mode = true;
}

```

```

private void button4_Click(object sender, EventArgs e)
{
    dataGridView2.Enabled = true;
    textBox4.Text = "";
    dataGridView2.Enabled = true;
    textBox5.Text = "";
    textBox6.Text = "";
    comboBox1.Text = "";
    insert_mode = true;
}

```

```

private void редагуватиToolStripMenuItem1_Click(object sender, EventArgs e)
{
    textBox4.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[1].Value);
    dataGridView2.Enabled = false;
    textBox5.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[2].Value);
    textBox6.Text = Convert.ToString(dataGridView2.CurrentRow.Cells[3].Value);
    comboBox1.Text =
Convert.ToString(dataGridView2.CurrentRow.Cells[4].Value);
    dataGridView2.Enabled = false;
    insert_mode = false;
}

```

```

private void видалитиToolStripMenuItem1_Click(object sender, EventArgs e)
{

```

```

        DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
        if (dialogResult == DialogResult.Yes)
        {
            try
            {
                string query2;
                query2 = "DELETE FROM `vydacha_na_tsekh` WHERE `id
_vydachi_na_tsekh`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
                cmd = new MySqlCommand();
                cmd.Connection = Form1.con;
                cmd.CommandText = query2;
                cmd.CommandType = CommandType.Text;
                cmd.ExecuteNonQuery();

                MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
                DataSet ds = new DataSet();
                da.Fill(ds, "vydacha_na_tsekh");
                dataGridView2.DataSource = ds;
                dataGridView2.Refresh();

                dataGridView2.Enabled = true;
                textBox4.Text = "";
                dataGridView2.Enabled = true;
                textBox5.Text = "";
                textBox6.Text = "";
                comboBox1.Text = "";
                insert_mode = true;
            }
            catch
            {
                MessageBox.Show("Видалення неможливо. Видача на цех має
пов'язані записи", "Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

```

private void button5_Click(object sender, EventArgs e)
{
    Form6 newForm = new Form6();
    newForm.ShowDialog();

}
}
}

```

Код формы «Цех» Form6:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
using Word = Microsoft.Office.Interop.Word;
using Excel = Microsoft.Office.Interop.Excel;

```

```

namespace WindowsFormsApp1
{
    public partial class Form6 : Form
    {
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        static public string id_tsekhu;
        static public string nomer_tsekhu;
        static public string nazva_tsekhu;

        public Form6()
        {

```

```

    InitializeComponent();
}

private void Form6_Shown(object sender, EventArgs e)
{
    insert_mode = true;
    query = @"SELECT
tsekh.id_tsekhu,
tsekh.nomer_tsekhu,
tsekh.nazva_tsekhu
FROM
tsekh
ORDER BY
tsekh.nazva_tsekhu ASC";
    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "tsekh");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "tsekh";
    dataGridView1.Columns[0].HeaderText = "Код цеху";
    dataGridView1.Columns[1].HeaderText = "Номер цеху";
    dataGridView1.Columns[2].HeaderText = "Назва цеху";
    dataGridView1.Columns[0].Width = 130;
    dataGridView1.Columns[1].Width = 200;
    dataGridView1.Columns[2].Width = 250;
    dataGridView1.Refresh();
}

private void button1_Click(object sender, EventArgs e)
{
    string query2;
    if (insert_mode == true)
    {
        query2 = "INSERT INTO `tsekh` (`nomer_tsekhu`, `nazva_tsekhu`) VALUES
(";
        query2 = query2 + "" + Convert.ToString(textBox1.Text) + ",";
        query2 = query2 + "" + Convert.ToString(textBox2.Text) + """;
    }
}

```

```

    }
    else
    {
        query2 = "UPDATE `tsekh` SET";
        query2 = query2 + "`nomer_tsekhu`=" + Convert.ToString(textBox1.Text) +
        ",";
        query2 = query2 + "`nazva_tsekhu`=" + Convert.ToString(textBox2.Text) +
        ")";
        query2 = query2 + "WHERE `id_tsekhu`=" +
        Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
    }
    cmd = new MySqlCommand();
    cmd.Connection = Form1.con;
    cmd.CommandText = query2;
    cmd.CommandType = CommandType.Text;
    cmd.ExecuteNonQuery();

    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "tsekh");
    dataGridView1.DataSource = ds;
    dataGridView1.Refresh();
    try
    {
        for (int i = 0; i < dataGridView1.RowCount; i++)
        {
            dataGridView1.Rows[i].Selected = false;
            if
            (dataGridView1.Rows[i].Cells[2].Value.ToString().Contains(textBox1.Text))
            {
                dataGridView1.Rows[i].Selected = true;

                break;
            }
        }
    }
    catch
    {

```

```

        dataGridView1.Rows[0].Selected = true;
    }

    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    insert_mode = true;

}

private void button2_Click(object sender, EventArgs e)
{
    dataGridView1.Enabled = true;
    textBox1.Text = "";
    textBox2.Text = "";
    insert_mode = true;
}

private void редагуватиToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (dataGridView1.RowCount != 0)
    {
        textBox1.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        textBox2.Text =
Convert.ToString(dataGridView1.CurrentRow.Cells[3].Value);
        dataGridView1.Enabled = false;
        insert_mode = false;
    }
}

private void видалитиToolStripMenuItem_Click(object sender, EventArgs e)
{
    DialogResult dialogResult = MessageBox.Show("Ви впевнені, що хочете
видалити запис?", "Видалення", MessageBoxButtons.YesNo,
MessageBoxIcon.Information);
    if (dialogResult == DialogResult.Yes)

```

```

    {
        try
        {
            string query2;
            query2 = "DELETE FROM `tsekh` WHERE `id_tsekhu`=" +
Convert.ToString(dataGridView1.CurrentRow.Cells[0].Value);
            cmd = new MySqlCommand();
            cmd.Connection = Form1.con;
            cmd.CommandText = query2;
            cmd.CommandType = CommandType.Text;
            cmd.ExecuteNonQuery();

            MySqlDataAdapter da = new MySqlDataAdapter(query2, Form1.con);
            DataSet ds = new DataSet();
            da.Fill(ds, "tsekh");
            dataGridView1.DataSource = ds;
            dataGridView1.Refresh();
            dataGridView1.Enabled = true;
            textBox1.Text = "";
            textBox2.Text = "";
            insert_mode = true;

        }

        catch

        {
            MessageBox.Show("Видалення неможливо. Цех має пов`язані записи",
"Помилка", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }

    }
}

private void виробництвоToolStripMenuItem_Click(object sender, EventArgs e)
{
    Form7 newForm = new Form7();
    newForm.ShowDialog();
}

```

```

    }

    private void button3_Click(object sender, EventArgs e)
    {
        //Копіювання файлів
        string fileName = "tsekh.docx";
        string fileName2 = "tsekh" +
Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value) + ".docx";
        string sourcePath = Environment.CurrentDirectory + @"\rep_copy";
        string targetPath = Environment.CurrentDirectory + @"\rep";
        string sourceFile = System.IO.Path.Combine(sourcePath, fileName);
        string destFile = System.IO.Path.Combine(targetPath, fileName2);
        System.IO.File.Copy(sourceFile, destFile, true);
        //Завершення копіювання
        string filename = Environment.CurrentDirectory + @"\rep\" + fileName2;
        Microsoft.Office.Interop.Word.Application word = new
Microsoft.Office.Interop.Word.Application();
        Microsoft.Office.Interop.Word.Document doc = new
Microsoft.Office.Interop.Word.Document();
        word.Visible = false;
        doc = word.Documents.Open(filename);
        doc.Activate();
        Microsoft.Office.Interop.Word.Range rng =
doc.Bookmarks["nomer_tsekhu"].Range;//Закладка що попередньо створена у Word
        rng.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[1].Value);
        Microsoft.Office.Interop.Word.Range rng2 =
doc.Bookmarks["nazva_tsekhu"].Range;//Закладка що попередньо створена у Word
        rng2.Text = Convert.ToString(dataGridView1.CurrentRow.Cells[2].Value);
        word.Visible = true;

    }
}
}

```

Код форми «Вмробника» Form7:

```
using System;
```

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;
using MySql.Data.Types;
```

```
namespace WindowsFormsApp1
{
```

```
    public partial class Form7 : Form
    {
```

```
        MySqlCommand cmd;
        string query;
        Boolean insert_mode;
        public Form7()
        {
            InitializeComponent();
        }
```

```
        private void Form7_Shown(object sender, EventArgs e)
        {
```

```
            insert_mode = true;
            query = @"SELECT
vyrobnytstvo.id_vyrobnyka,
vyrobnytstvo.id_tsekhu,
vyrobnytstvo.nazva_tovaru,
vyrobnytstvo.dani_tovaru,
vyrobnytstvo.kilkist
FROM
vyrobnytstvo
ORDER BY
vyrobnytstvo.nazva_tovaru ASC";
```

```

    MySqlDataAdapter da = new MySqlDataAdapter(query, Form1.con);
    DataSet ds = new DataSet();
    da.Fill(ds, "vyrobnytstvo");
    dataGridView1.DataSource = ds;
    dataGridView1.DataMember = "vyrobnytstvo";
    dataGridView1.Columns[0].HeaderText = "Код виробника";
    dataGridView1.Columns[1].HeaderText = "Код цеху";
    dataGridView1.Columns[2].HeaderText = "Назва товару";
    dataGridView1.Columns[3].HeaderText = "Дані про товар";
    dataGridView1.Columns[4].HeaderText = "Кількість";
    dataGridView1.Columns[0].Width = 100;
    dataGridView1.Columns[1].Width = 60;
    dataGridView1.Columns[2].Width = 120;
    dataGridView1.Columns[3].Width = 420;
    dataGridView1.Columns[4].Width = 120;
    dataGridView1.Refresh();
}
}
}

```

Код форми «Вхід у систему» Form8:

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApp1
{
    public partial class Form8 : Form
    {
        public Form8()
        {
            InitializeComponent();

```

```

    }

private void button1_Click(object sender, EventArgs e)
{
    Form1.con.Close();
    string user_name = textBox1.Text;
    string user_pasword = textBox2.Text;
    string user_connection_string;
    string user_server = textBox3.Text;
    if (user_pasword == "")
    {
        user_connection_string = "server=" + user_server + ";User Id=" + user_name
+
        ";Persist Security Info=True;database=ais_oblik_tovariv";
    }
    else
    {
        user_connection_string = "server=" + user_server + ";User Id=" + user_name
+
        ";password=" + user_pasword + ";Persist Security
Info=True;database=ais_oblik_tovariv";
    }
    try
    {
        Form1.con.ConnectionString = user_connection_string;
        Form1.con.Open();
        MessageBox.Show("Підключення відбулось успішно", "Підключення",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        this.Close();
    }
    catch
    {
        MessageBox.Show("Помилка при підключенні", "Увага",
        MessageBoxButtons.OK, MessageBoxIcon.Question);
    }
}
}

```

```
}
```

Код форми «Розробник інформаційної системи» Form9:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
  
namespace WindowsFormsApp1  
{  
    public partial class Form9 : Form  
    {  
        public Form9()  
        {  
            InitializeComponent();  
        }  
    }  
}
```