

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ**

Інститут (факультет) Автоматизації комп'ютерних систем  
Кафедра Інформаційних систем

«До захисту в ЕК»  
Директор інституту(декан факультету)  
\_\_\_\_\_  
(підпис) Форсюк А.В.  
(прізвище та ініціали)

« \_\_\_ » червня 2020р.

«До захисту допущено»  
Завідувач кафедри  
\_\_\_\_\_  
(підпис) Чумаченко С.М.  
(прізвище та ініціали)

« \_\_\_ » червня 2020р.

**КВАЛІФІКАЦІЙНА РОБОТА  
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 Комп'ютерні науки та інформаційні технології  
(код та назва спеціальності)

освітньо-професійної програм Комп'ютерні науки

на тему: «Розроблення інформаційної системи обліку поставань ресторану "Суші Манія»»

Виконав: здобувач 4 курсу, групи б

\_\_\_\_\_  
Дідук Карина Віталіївна  
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник Ковалець Іван Васильович  
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти \_\_\_\_\_  
(прізвище та ініціали) (підпис)  
\_\_\_\_\_  
(прізвище та ініціали) (підпис)  
\_\_\_\_\_  
(прізвище та ініціали) (підпис)

Рецензент Смітюх Ярослав Володимирович  
(прізвище та ініціали) (підпис)

Засвідчую, що в цій кваліфікаційній  
роботі немає запозичень із праць  
інших авторів без відповідних  
посилань.

Здобувач \_\_\_\_\_  
(підпис)

# НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизація комп'ютерних систем

Кафедра Інформаційних технологій

Освітній ступінь Бакалавр

Спеціальність 122 комп'ютерні науки та інформаційні технології

(код і назва)

Освітньо-професійна програма Комп'ютерні науки

(назва)

**ЗАТВЕРДЖУЮ**

**Завідувач**

**кафедри**

**Чумаченко С.М.**

**“\_\_\_” червня 2020 року**

## **З А В Д А Н Н Я**

### **НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА**

Дідук Карини Віталіївни

(прізвище, ім'я, по батькові)

1. Тема роботи «Розроблення інформаційної системи обліку постачань ресторану "Суші Манія»»

керівник роботи Ковалець Іван Васильович,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 27”04 2020 року №269-к

2. Строк подання здобувачем роботи \_\_\_\_\_

3. Вихідні дані до роботи: Виконане замовлення, Чек, Звітність ресторану.

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити) Розділ 1. 1.1 Найменування компанії. 1.1 Загальна характеристика ТОВ “Суші Машія” 1.2 Організаційна структура підпри. 1.2. Режим і особливості функціонування 1.3 Аналіз вибору постачальників товару 1.4 Опис інформаційного і технічного забезпечення підприємства 1.5 Функціональна модель відділу постачання 1.5.1 Аналіз системи обліку постачань та порівняння з систем-аналогів 1.5.2 "jSolution Склад 1.5.4 Універсальна Система Обліку 1.5.5 Порівняння систем-аналогів 1.6 Обґрунтування доцільності проектування і розроблення системи 1.8 Постановка задачі на проектування функції та вимоги 1.8.1 Призначення та цілі створення системи 1.8.2 Вимоги до створюваної інформаційної системи 1.8.3. Функції, які виконує система. 1.8.4. Вхідні та вихідні дані системи 2.1 Інформаційне забезпечення проекту 2.2 Програмне забезпечення розробки 2.3 Технічне забезпечення розробки інформаційної системи 2.4 Обґрунтування структури та розробка логічної і фізичної моделей бази даних інформаційної системи 2.5 Обґрунтування вибору технічних засобів розробки 2.6 Створення інтерфейсу користувача

В данному пункті ми пишем код для створення 2.6.1 Підключення системи до БД 2.7 Реалізація функцій системи 2.8 Інструкція користувача 3.1 Вступ 3.2 Керівник відділу постачання повинен 3.3 Ряд небезпечних і шкідливих виробничих факторів при роботі з ПК 3.4 Приміщення, в якому розташовані ПК, відповідно до санітарних та інших норм безпеки має бути обладнений 3.5 Умови, яким повинна задовольняти робоча зона 3.6 Обов'язки оператора 3.7 До початку роботи оператор зобов'язаний 3.8 Вимоги до робочого місця **ВИСНОВКИ ДОДАТКИ СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ковалець І.В.		
2	Ковалець І.В.		
3	Ковалець І.В.		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Ознайомлення з об'єктом автоматизації	04.05-05.05 2020	Виконано
2	Аналіз об'єкту автоматизації	06.05 2020	Виконано
3	Аналіз ІС на об'єкту автоматизації	07.05-08.05 2020	Виконано
4	Обґрунтування доцільності проектування й розроблення системи	11.05-12.05 2020	Виконано
5	Постановка задачі	13.05-14.05 2020	Виконано
6	Розробка інтерфейсу	15.05-16.05 2020	Виконано
7	Розробка елементів керування	17.05-18.05 2020	Виконано
8	Реалізація функцій системи	19.05-25.05 2020	Виконано

**Здобувач**

\_\_\_\_\_

(підпис)

**Дідук К.В.**

\_\_\_\_\_

(прізвище та ініціали)

**Керівник роботи**

\_\_\_\_\_

( підпис )

**Ковалець І.В.**

\_\_\_\_\_

(прізвище та ініціала)

## АНОТАЦІЯ

Кваліфікаційна робота « Розроблення інформаційної системи обліку постачань ресторану «Суші Манія»» складається з 49 сторінок, містить 11 рисунків, 3 додатки та 10 літературного джерела. Автор Карина Дідук. Дана робота описує розробку інформаційної системи обліку постачань ресторану "Суші Манія".

Засобом для розробки проекту, інтегроване середовище розробки Microsoft Visual Studio для написання програмного коду веб сторінки (клієнтська та серверна частини). Використані .Net Core 2.2 для реалізації серверної частини та Angular 8 для клієнтської відповідно. Для розробки бази даних використовувалась MS SQL Server, що розроблялась у SQL Management Studio як середовищі розробки.

**Ключові слова:** інформаційна система, веб-сторінка, база даних, SQL SERVER, WEB AP.

## ANNOTATION

The qualification work "Development of the information system of the account of deliveries of restaurant" Sushi Mania "" consists of 49 pages, contains 11 drawings, 3 appendices and 10 literary sources. Author Karina Diduk. This work describes the development of an information system for accounting for the supply of the restaurant "Sushi Mania".

A project development tool, an integrated Microsoft Visual Studio development environment for writing web page code (client and server parts). Used .Net Core 2.2 to implement server chat and Angular 8 for client respectively. MS SQL Server was used to develop the database, which was developed in SQL Management Studio as a development environment.

**Keywords:** information system, web page, data base, SQL SERVER, WEB API,

## СКОРОЧЕНІ ТЕРМІНИ

**ТОВ** – Товариство з обмеженою відповідальністю

**ІС** – Інформаційна система

**ПЗ** – Програмне забезпечення

**БД** – База даних

**SQL** – Мова структурних запитів

**PHP** – Інструменти для створення персональних веб-сторінок

**POS** – Програмно-апаратний комплекс, що функціонує на базі фіскального реєстратора

**MS** – Microsoft

**ПК** – Персональний комп'ютер

**ЕОМ** – Електронна обчислювальна машина

# ЗМІСТ

АНОТАЦІЯ .....	4
ВСТУП.....	9
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ`ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ.....	10
1.1 Найменування компанії.....	10
1.1.1 Загальна характеристика ТОВ “Суші Машія” .....	10
1.2 Організаційна структура підприємства.....	10
1.2. Режим і особливості функціонування.....	13
1.3 Аналіз вибору постачальників товару.....	13
1.4 Опис інформаційного і технічного забезпечення підприємства .....	14
1.5 Функціональна модель відділу постачання .....	15
1.5.1 Аналіз системи обліку поставок та порівняння з систем-аналогів .....	15
1.5.2 "jSolution Склад" .....	15
1.5.4 Універсальна Система Обліку.....	15
1.5.5 Порівняння систем-аналогів .....	15
1.6 Обґрунтування доцільності проектування і розроблення системи.....	16
1.8 Постановка задачі на проектування функції та вимоги.....	16
1.8.1 Призначення та цілі створення системи.....	16
Система обліку поставки товарів прзначена тільки для внутрішнього використання в ресторані «Суші Манія».....	16
1.8.2 Вимоги до створюваної інформаційної системи.....	17
Система обліку поставки товарів буде використана у відділі поставок. ....	17
1.8.3. Функції, які виконує система. ....	17
1.8.4. Вхідні та вихідні дані системи.....	17
РОЗДІЛ 2. РОЗРОБКА АВТОМАТИЗАЦІЇ.....	18
2.1 Інформаційне забезпечення проекту .....	18
2.2 Програмне забезпечення розробки .....	18
2.3 Технічне забезпечення розробки інформаційної системи.....	20
2.4 Обґрунтування структури та розробка логічної і фізичної моделей бази даних інформаційної системи.....	21
2.5 Обґрунтування вибору технічних засобів розробки .....	22
2.6 Створення інтерфейсу користувача .....	23
2.6.1 Підключення системи до БД.....	24
При підключенні БД було використано ORM entity framework core.....	24
2.7 Реалізація функцій системи .....	25
2.8 Інструкція користувача .....	29
РОЗДІЛ 3. ОХОРОНА ПРАЦІ .....	31
3.1 Вступ .....	31

3.2 Керівник відділу постачання повинен .....	31
3.3 Ряд небезпечних і шкідливих виробничих факторів при роботі з ПК .....	31
3.4 Приміщення, в якому розташовані ПК, відповідно до санітарних та інших норм безпеки має бути обладнений .....	33
3.5 Умови, яким повинна задовольняти робоча зона.....	33
3.6 Обов'язки оператора.....	34
3.7 До початку роботи оператор зобов'язаний: .....	34
3.8 Вимоги до робочого місця.....	35
ВИСНОВКИ.....	36
ДОДАТКИ.....	37
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	!

## ВСТУП

Нові матеріали, технології та інструменти призводять до збільшення темпів виконання робіт в різних галузях і появи конкурентів серед фірм. Для того, щоб вижити на ринку і отримати хороший прибуток, необхідно не тільки відмінно виконувати роботи за основним видом діяльності, а й налагодити супутні функції - облік матеріалів, обслуговування та облік клієнтів, своєчасне формування звітності.

Від швидкості отримання потрібної інформації за матеріалами, клієнтам і замовленнями залежить вибір правильних рішень в управлінні фірмою. Необхідна автоматизація цих операцій шляхом розробки інформаційної системи, яка забезпечить введення, зберігання, редагування та отримання необхідної інформації.

Важлива роль в структурі сучасного малого бізнесу відводиться підприємствам громадського харчування, особливістю яких є організація одночасного виробництва готових страв і їх реалізація. Даний процес особливо важливий з позиції здійснення облікової та аналітичної діяльності, збільшення швидкості обробки інформації. Для досягнення даної завдання важливу роль відіграє автоматизація бізнес-процесів.

Основною особливістю сучасних закладів ресторанного господарства повинна бути їх локальність за характером діяльності, спрямованість на регіональний ринок, на задоволення індивідуальних потреб споживачів і як наслідок – наявність достатньої кількості закладів ресторанного господарства на ринку. При розробці кожного проекту повинен використовуватись індивідуальний підхід до об'єкту проектування.

# РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТУ АВТОМАТИЗАЦІЇ ТА ПОСТАНОВКА ЗАДАЧІ НА ПРОЕКТУВАННЯ

## 1.1 Найменування компанії

- повне – Товариство з обмеженою відповідальністю ТОВ ‘Суші Манія’  
Місце знаходження :
- адреса: Київська обл., м.Київ, вул Льва Толстого , 11/61

### 1.1.1 Загальна характеристика ТОВ “Суші Машія”

**ТОВ "Суші Манія"** - Ресторан «Суші Манія» був заснований в 2001 році. У ресторані витриманий інтер'єр з темними стінами, масивними дерев'яними меблями, яскравими світильниками.

До 40% меню засноване на червоній рибі.

У меню представлені:

- японські страви
- європейські страви

Та в тому числі суші, роли, піца, чізкейки.

Ресторан відомий не тільки своїм різноманітним меню і стабільно високою якістю страв, але і цікавими акціями і спецпропозиціями, що дозволяють пообідати або повечеряти з максимально вигідними цінами.

Особливої уваги заслуговує обслуговування та сервіс, що є принциповою позицією керівництва закладу.

Даний ресторан підійде для ділової зустрічі, для побачення під романтичну музику, для прекрасного вечері за келихом вина в колі друзів або сім'ї. Також тут пропонуються всі умови для проведення весіль, банкетів і корпоративних заходів.

## 1.2 Організаційна структура підприємства



Рис.1.2.1 Організаційна структура підприємства



Рис 1.2.2. Організаційна структура відділу постачання продуктів

## **1.2. Режим і особливості функціонування**

У відділі постачання основними напрямками є:

- Робота з постачальниками продуктів.
- Заповнення бланку потрібних продуктів.
- Перевірка правильності доставлених товарів.
- Ведення звітності.
- Контроль якості продуктів
- Аналіз цін.
- Моніторинг товару наявного на складі.
- Списання товару у якого збіг термін придатності.
- Аналіз динаміки збуту товару в ресторані та контроль кількості проданого товару задля майбутнього планування замовлень в постачальників

## **1.3 Аналіз вибору постачальників товару**

Ефективність закупівельної роботи прямо залежить від правильного вибору джерел оптових закупівель і конкретних постачальників товарів. Вибір джерел закупівель і конкретних постачальників здійснюється після складання плану надходження товарів на підприємство. В умовах ринку це — дуже важка справа, а помилки у виборі можуть коштувати для підприємств дуже дорого. Оптові підприємства можуть закуповувати товари з різних джерел, якими є:

— промислові і сільськогосподарські підприємства, місцева промисловість, індивідуальні виробництва;

— оптово-посередницькі структури;

— оптові бази;

— іноземні фірми і компанії.

Джерела товарного забезпечення визначаються виходячи зі структури конкретного ринку товарів, контингенту покупців, обсягу й асортименту товарів, товарообігу і фінансових можливостей оптового торгового підприємства.

Вибір конкретних постачальників товарів здійснюється в тому разі, коли серед них немає монополії. З цією метою складається список можливих постачальників кожного виду товарів. Джерелом інформації для складання

списку постачальників є вивчення ринку товарів. Ефективність закупівельної роботи прямо залежить від правильного вибору джерел оптових закупівель і конкретних постачальників товарів.

Рішення щодо вибору джерел закупівель і конкретних постачальників здійснюється після складання плану надходження товарів на підприємство. В умовах ринку це — дуже важка справа, а помилки у виборі можуть коштувати для підприємств дуже дорого.

Оптові підприємства можуть закуповувати товари із різних джерел, якими є:

- промислові і сільськогосподарські підприємства, місцева промисловість, індивідуальні виробництва;
- оптово-посередницькі структури;
- оптові бази;
- іноземні фірми і компанії.

Джерела товарного забезпечення визначаються виходячи зі структури конкретного ринку товарів, контингенту покупців, обсягу й асортименту товарів, товарообігу і фінансових можливостей оптового торгового підприємства.

Вибір конкретних постачальників товарів здійснюється в тому разі, коли серед них немає монополії. З цією метою складається список можливих постачальників кожного виду товарів. Джерелом інформації для складання списку постачальників є вивчення ринку товарів.

#### **1.4 Опис інформаційного і технічного забезпечення підприємства**

Компоненти робочої станції в ТОВ "Суші Манія":

- процесор Intel Pentium G4500, сокет LGA 1151;
- відеоадаптер Gigabyte GeForce GTX 1070 G1 ROCK [GV-N1070G1 ROCK-8GD], турбочастота 1797 МГц, обсяг 8 Гб;
- ОЗУ 16 Гб - Corsair Vengeance LPX DDR4 2x8Gb 2400 МГц (CMK16GX4M2A2400C16), тип пам'яті DDR4 частота 2400 МГц;
- монітор Samsung LC27F396FHIXCI 27 ", матриця VA;
- дисковий накопичувач: WD Red 8ТВ внутрішній жорсткий диск (WD80EFZX), інтерфейс диска SATA3 6.0 Гбіт / с,
- МФУ EPSON L366;
- блок живлення-400 Ватт.

## **1.5 Функціональна модель відділу постачання**

Одним з головних відділів є відділ постачання. Для моделювання бізнес-процесу застосовано було програму СА ERwin Process Modeler (раніше PRwin) в нотації IDEF0. PRwin допомагає чітко "документувати важливі аспекти будь-яких бізнес-процесів: дії, які необхідно зробити, способи їх здійснення і контролю, що вимагаються для цього ресурси, а також візуалізувати одержувані від цих дій результати.

### **1.5.1 Аналіз системи обліку постачань та порівняння з систем-аналогів**

#### **1.5.2 "jSolution Склад"**

Система jSolutions – дозволяє забезпечувати управління складською логістикою, запасами, реалізацією товарів і послуг в різних масштабах, від міні-складу до великих складських центрів. Рішення побудоване на базі хмарної платформи jSolutions і має функції, які дозволяють об'єднати в єдиній базі даних складський облік товарів, в розрізі характеристик і функцій:.

#### **1.5.4 Універсальна Система Обліку**

За допомогою такого потужного і професійного інструменту ви зможете повністю автоматизувати весь робочий процес. Розрахований на багато користувачів режим допоможе вам оперативно обмінюватися інформацією з усіма відділами і працівниками. Кожен з ваших співробітників, маючи свої права доступу до потрібної інформації, буде виконувати свої завдання, при цьому не відволікатися на інші функції в програмі. Наприклад, користувач може сформулювати заявку на закупівлю тих чи інших товарів або матеріалів, після підтвердження заявки можна заповнити і роздрукувати потрібні документи, а також порівняти вартість по всім постачальникам і вибрати, у кого закупівля буде найвигіднішою для компанії.

#### **1.5.5 Порівняння систем-аналогів**

У табл. 1.5.1 порівняння інших існуючих систем обліку постачання та зберігання на ринку, які можна використовувати окрім нашої системи обліку

постачання для ресторну 'Суші Манія'. Виявті недоліки та переваги всіх систем обліку.

Таблиця 1.5.1. Результати порівняння систем

	<i>УСО</i>	jSolution Склад
Зручність інтерфейсу	-	+
Захист доступу	+	+
Звітність	+/-	-
Надійність даних	висока	середня
Бескоштовна	-	-

## 1.6 Обґрунтування доцільності проектування і розроблення системи

Система обліку постачання для ресторану «Суші Манія» надійна, безкоштова, має систему захисту та дуже проста в інтерфейсі тому в порівнянні всіх критеріїв систем-аналогів які ми порівнял, найкращий варіант це наша система обліку постачань. Запропонована мною система реалізує функціонал що задовольняє потреби саме ресторану «Суши Манія»

## 1.8 Постановка задачі на проектування функції та вимоги

### 1.8.1 Призначення та цілі створення системи

Система обліку поставки товарів прзначена тільки для внутрішнього використання в ресторані «Суші Манія»

#### Користувачі:

- Директор ресторану
- Адміністратор
- IT відділ
- Бухгалтер

- Керівник відділу постачання
- Менеджер з закупівлі

**Цілі які є основними:**

- Полегшення роботи зі звітністю
- Перегляд всіх замовлених товарів
- Перегляд товарів які потрібно замовити
- Списання зіпсованої продукції
- Моніторинг динаміки кількості наявного товару в часі

### **1.8.2 Вимоги до створюваної інформаційної системи**

Система обліку поставки товарів буде використана у відділі поставок.

Користувачами буде персонал ресторану.

Вимоги які були основними до системи яка створювалась :

- Зручність інтерфейсу
- Захист доступу
- Надійність
- Внесення та редагування даних
- Підтримка всіх операційних систем
- Доступ з мобільних гаджетів

### **1.8.3. Функції, які виконує система.**

Система обробки поставок виконує такі функції:

- Запис та корегування товарів;
- Перелік всіх товарів які знаходяться на складі;
- Кожен користувач з доступом може вносити зміни;
- Контроль за товарами які потрібно замовити.

Система має вікно входу данні до якого є тільки у користувачів які мають доступ.

### **1.8.4. Вхідні та вихідні дані системи.**

Вхідними даними системи є:

- Перелік найменувань товарів;
- Кількість товарів в відповідних одиницях виміру.
- Відомості про постачальника.
- Вартість закуплених товарів.

Вихідні дані:

- Залишок наявних товарів на складі;
- Товарні накладні;
- Дата та час поставки.

### 2.1 Інформаційне забезпечення проекту

Інформаційне забезпечення вимагає чіткого планування етапів реалізації проекту, тому створення системи обліку товару можна розділити на певні етапи, а саме:

- Визначення структури бази даних в якій ведеться облік товару, що надходить на склад;
- Логіка внесення змін до бази даних, реалізована в програмі з функцією захисту доступу тільки для авторизованих користувачів застосунку, що мають відповідні паролі для входу в систему;
- Інтерфейс програми втілений у веб-сторінці, що має чіткий та зрозумілий функціонал достатній для ведення звітності відділу поставки, формування накладних тощо.

### 2.2 Програмне забезпечення розробки

Розробка новітніх інформаційних систем потребує використання професійного програмного забезпечення, що дає змогу пришвидшити процес реалізації та підвищити якість кінцевого програмного продукту та мати можливість подальшого покращення та розширення функціоналу системи, якщо виникне відповідна потреба. Тому при виборі засобів розробки враховано вище наведені вимоги.

Підчас розробки інформаційної системи було використано .Net Core Framework версії 2.2, SQL Server Management Studio версії 17.9.1, Visual Studio Code версії 1.46, Microsoft Visual Studio Community 2019 версії 16.4.4, Angular CLI версії 8.0.0.

NET Core - це модульна реалізація бібліотеки і середовища виконання, до складу якої входить поднабір .NET Framework. .NET Core працює на Windows, Mac і Linux. Версія складається з колекції бібліотек CoreFX і невеликий оптимізованої середовища виконання CoreCLR. .NET Core - проект з відкритим вихідним кодом, тому ви можете спостерігати за його розвитком і підтримувати його на GitHub.

Середовище виконання CoreCLR (Microsoft.CoreCLR) і бібліотеки CoreFX поширюються через NuGet. Оскільки версія .NET Core є компонентізований набір бібліотек, ви можете обмежити кількість API для свого застосування і використовувати тільки необхідні елементи. Крім того, можна запускати програми на базі .NET Core в набагато більш обмежених середовищах (наприклад, в ASP.NET Core на Nano Server).

Функціональна структура середовища Microsoft Visual Studio Community включає в себе:

- редактор вихідного коду, який включає безліч додаткових функцій, як автодоповнення IntelliSense, рефракторинг коду і т. д. ;

- відладчик коду;
- редактор форм, призначений для спрощеного конструювання графічних інтерфейсів;
- веб-редактор;
- дизайнер класів;
- дизайнер схем баз даних.

Visual Studio також дозволяє створювати і підключати сторонні додатки (плагіни) для розширення функціональності практично на кожному рівні, включаючи додавання підтримки систем контролю версій вихідного коду (Subversion і VisualSourceSafe), додавання нових наборів інструментів (для редагування і візуального проектування коду на предметно-орієнтованих мовах програмування або інструментів для інших аспектів процесу розробки програмного забезпечення).

Інтегроване середовище розробки (IntegratedDevelopmentEnvironment - IDE) Visual Studio пропонує ряд високорівневих функціональних можливостей, які виходять за рамки базового управління кодом.

Нижче перераховані основні переваги IDE-середовища Visual Studio.

- Вбудований Web-сервер. Для обслуговування Web-додатки ASP.NET необхідний Web-сервер, який буде очікувати Web-запити і обробляти відповідні сторінки. Наявність в Visual Studio інтегрованого Web-сервера дозволяє запускати Web-сайт прямо з середовища проектування, а також підвищує безпеку, виключаючи ймовірність отримання доступу до тестовому Web-сайту з якого-небудь зовнішнього комп'ютера, оскільки тестовий сервер може приймати з'єднання лише з локального комп'ютера.
- Підтримка безлічі мов при розробці. Visual Studio дозволяє писати код своєю рідною мовою чи будь-яких інших бажаних мовами, використовуючи весь час один і той же інтерфейс (IDE). Більш того, Visual Studio також ще дозволяє створювати Web-сторінки на різних мовах, але поміщати їх все в один і той же Web-додаток. Єдиним обмеженням є те, що в кожній Web-сторінці можна використовувати тільки якийсь один мову (очевидно, що в іншому випадку проблем при компіляції було б просто не уникнути).
- Інтуїтивний стиль кодування. За замовчуванням Visual Studio форматує код у міру його введення, автоматично вставляючи необхідні відступи і застосовуючи колірне кодування для виділення елементів типу коментарів. Такі незначні відмінності

роблять код більш зручним для читання і менш схильним до помилок. Застосовувані Visual Studio автоматично параметри форматування можна навіть налаштувати, що дуже зручно у випадках, коли розробник вважає за краще інший стиль розміщення дужок (наприклад, стиль K & R, при якому відкриває дужка розміщується на тому самому рядку, що і оголошення, якому вона передує).

- Більш висока швидкість розробки. Багато з функціональних можливостей Visual Studio спрямовані на те, щоб допомагати розробнику робити свою роботу якомога швидше. Зручні функції, на зразок функції IntelliSense (яка вміє перехоплювати помилки і пропонувати правильні варіанти), функції пошуку і заміни (яка дозволяє відшукувати ключові слова як в одному файлі, так і в усьому проекті) і функції автоматичного додавання і видалення коментарів (яка може тимчасово приховувати блоки коду), дозволяють розробнику працювати швидко і ефективно.
- Можливості налагодження. Пропоновані в Visual Studio інструменти налагодження є найкращим засобом для відстеження загадкових помилок і діагностування дивної поведінки. Розробник може виконувати свій код по рядку за раз, встановлювати інтелектуальні точки переривання, при бажанні зберігаючи їх для використання в майбутньому, і в будь-який час переглядати поточну інформацію з пам'яті.

Angular - JS фреймворк з відкритим вихідним кодом, розроблений і підтримуваний google. Інструмент дає все необхідне для створення і управління динамічними front end сторінками для веб-додатки. Модульний підхід до веб-дизайну і величезна підтримує спільнота роблять AngularJS популярним серед професійних розробників. Насправді, на Angular працюють деякі з найбільш навантажених сайтів в мережі, в тому числі google і Virgin America.

### **2.3 Технічне забезпечення розробки інформаційної системи**

Для використання створеної системи обліку користувачі повинні задовольняти наступним вимогам:

- Пристрій повинен мати можливість підключення до мережі інтернет;
- Завантажити та встановити одну з останніх версій найпоширеніших веб-браузерів (наприклад Chrome, Safari, Mozilla Firefox);
- Мати елементарний досвід користування веб-сторінками на початковому рівні;

## 2.4 Обґрунтування структури та розробка логічної і фізичної моделей бази даних інформаційної системи

Для форм пошуків та звітів необхідних для відділу поставання головним є зберігання інформації. Використовуємо систему управління базами даних MySQL.

Логічне проектування – перетворення концептуального представлення в логічну структуру бази даних, включаючи проектування відношень.

Фізичне проектування – ухвалення рішення про те, як логічна модель буде фізично реалізована у базі даних, створюваної за допомогою обраної СКБД.

На основі фізичної моделі ми зробили таблиці:

**ProductNames** (Назва продукту) – Таблиця в якій містяться назви всіх продуктів

**Products** (Продукт) – В цій таблиці вказана ціна за одиницю товару, ціна за одиницю товару яка помножена на кількість, коментар, назва продукту та станні корегування. Під час замовлення ми вказуємо за одиницю, ціна помножена на кількість рахується автоматично

**Recieves** (Отримання) – В даній таблиці буде інформація про весь отриманий товар, а саме:

- Кількість продукту
- Назва продукту
- Ціна за одиницю
- Ціна за всю кількість
- Стан корегування
- Коментар

Наступні дві таблиці відповідають за повернення товару

**ReturnedDetails** (Повернення товарів) В таблиці описані товари які необхідно повернути, але тільки ті що є на складі. Вказано: Кількість продуктів що потрібно повернути, назва, та останні корегування

**Returneds** (Кому) – Таблиця в якій вказано кому саме потрібно повернути товар та дата останнього корегування

Всі дані в таблиці та самі таблиці створились автоматично у вигляді фізичної моделі та графічної схеми з відповідними ключовими полями, та зв'язками між полями таблиць.

База даних буде складатись з 5 таблиць (див. Додаток Б-В):

## 2.5 Обґрунтування вибору технічних засобів розробки

У виборі технічних засобів я переглянула безліч програм. Потрібно було вибрати більш зручнішу у використанні, зрозумілу в інтерфейсі та інтуїтивно легку.

Після пошуку найкращих, було вибрано ось ці технічні засоби:

**Web API** - це інтерфейс програмування додатків для веб-сервера або веб-браузера. Це концепція веб-розробки, яка зазвичай обмежується клієнтською стороною веб-програми (включаючи будь-які використовувані веб-рамки), і, отже, зазвичай не включає деталі веб-сервера чи веб-переглядача, такі як SAPI або API, за винятком загальнодоступного доступу через віддалений веб-додаток.

**.Net Core** - це модульна платформа для розробки програмного забезпечення з відкритим вихідним кодом.

Сумісна з такими операційними системами як Windows, Linux и macOS. Була випущена компанією Microsoft.

На платформі GitHub існує власне ПОВІДОМЛЕННЯ. Підтримує наступні мовні програми: C #, Visual Basic .NET (частково) и F #

**Angular** - (версія 2 і вище) - це відкрита і вільна платформа для розробки веб-додатків, написана на мові TypeScript, що розробляється командою з компанії Google, а також спільнотою розробників з різних компаній. Angular - це повністю переписаний фреймворк від тієї ж команди, яка написала AngularJS.

Спочатку створювався як друга версія AngularJS. Angular 2 був переписаний з нуля на TypeScript, володіє іншою архітектурою і не є назад сумісним з AngularJS, в зв'язку з чим для запобігання плутанини було вирішено розвивати його як окремий фреймворк, нумерація версій якого починається з 2.

**Система контролю версій Git** - дозволяє та заохочує вас мати кілька локальних відділень, які можуть бути повністю незалежними один від одного. Створення, об'єднання та видалення цих напрямків розвитку займає секунди.

## 2.6 Створення інтерфейсу користувача

**В данному пункті ми пишем код для створення інтерфейсу**

```
import { Component, OnInit } from '@angular/core';
import { ActivatedRoute, Router } from '@angular/router';
import { switchMap } from 'rxjs/operators';
import { Observable } from 'rxjs';
import { IProduct } from './product';
import { ProductNameService } from './product-name.service';

@Component({
  selector: 'app-product-list',
  templateUrl: './product-list.component.html',
  styleUrls: ['./product-list.component.css']
})
export class ProductListComponent implements OnInit {
  public products$: Observable<IProduct[]>;
  public newProductId: number;
  public newProduct: IProduct;
  public products: IProduct[];
  public loading: boolean = true;

  constructor(private productNameService: ProductNameService,
    private route: ActivatedRoute,
    private router: Router) { }

  ngOnInit() {
    this.products$ = this.route.paramMap.pipe(
      switchMap(params => {
        this.newProductId = +params.get('id');
        return this.productNameService.getAll();
      })
    );

    this.products$.subscribe(p => {
      this.products = p;
      this.loading = false;
    });
  }
}
```

```

public delete(id: number) {
  this.loading = true;
  this.productNameService.delete(id).subscribe(_ => {
    this.products = this.products.filter(p => p.id !== id);
    this.loading = false;
  });
}

public edit(id: number) {
  this.router.navigate(['/products/${id}']);
}
}

<router-outlet></router-outlet>
<div *ngIf="loading" class="m-3 text-center">
  
</div>
<div *ngIf="!loading">
  <h2>Список товарів</h2>
  <table class="table">
    <tr>
      <th>Назва товару</th>
      <th></th>
    </tr>
    <tr *ngFor="let product of products">
      <td>{{ product.name }}</td>
      <td class="text-right">
        <i class="fa fa-edit mr-2" (click)="edit(product.id)"
style="cursor:pointer">
          </i>
        <i class="fa fa-trash" (click)="delete(product.id)"
style="cursor:pointer">
          </i>
      </td>
    </tr>
  </table>
</div>

```

### 2.6.1 Підключення системи до БД

При підключенні БД було використано ORM entity framework core

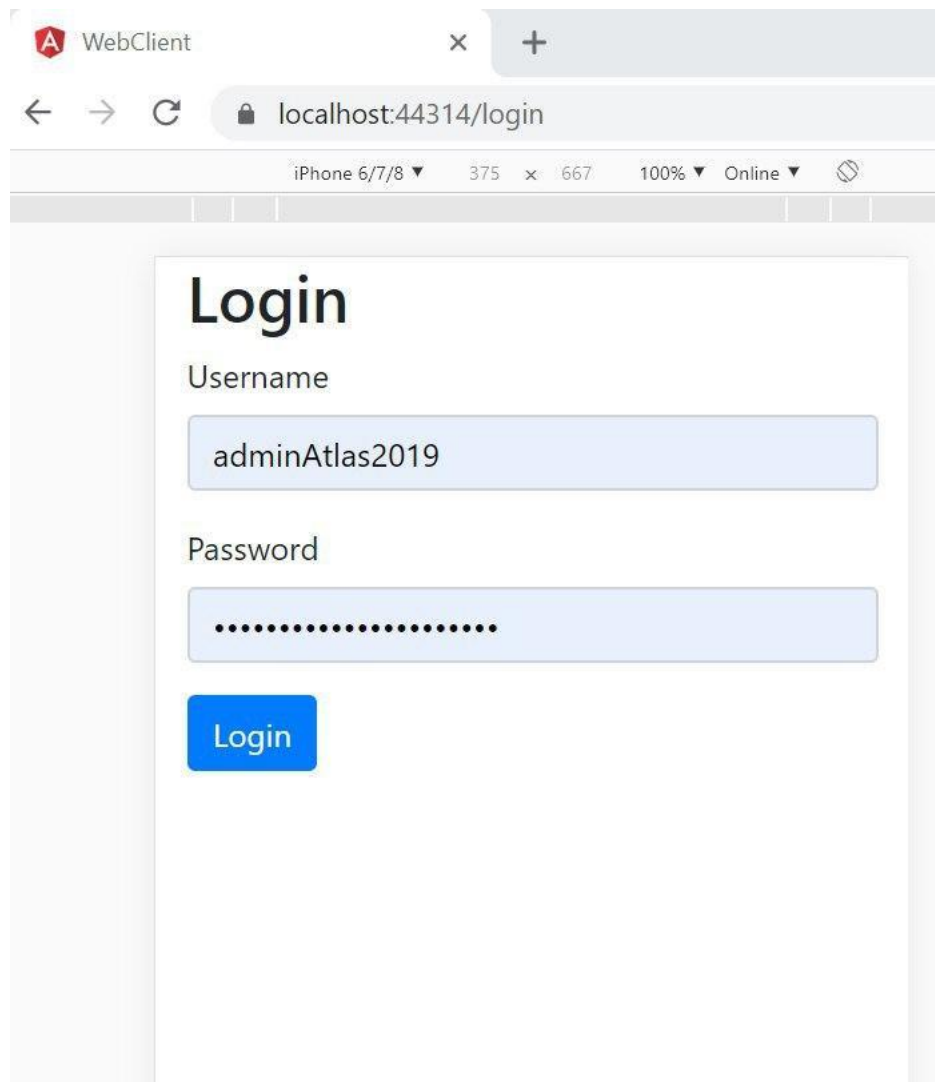
Додаток А

## 2.7 Реалізація функцій системи

Розроблена система обліку для ресторану “Суші Манія” дозволяє нам

- Повний захист даних
- Зручний список товарів
- Повернення товарів
- Список товарів що вивантажили
- Формування звітів за певний період за кожним співробітником-користувачем

### Захист даних



The image shows a screenshot of a web browser window. The browser's address bar displays 'localhost:44314/login'. The page content includes a login form with the following elements:

- Title:** Login
- Username:** A text input field containing 'adminAtlas2019'.
- Password:** A password input field with masked characters (dots).
- Login Button:** A blue button labeled 'Login'.

Рис. 2.7.1 Форма авторизації

## Перегляд списку товарів

Склад Товари Вихід

Список товарів Створити новий товар

### Список товарів

















Назва товару	
Авокадо ваговий	 
Картопля молода	 
Кока Кола оріджинал	 
Кунжут в зернах	 
Лосось ваговий	 
Морква	 
Огірок український	 
Пиво Хугарден нефільтроване	 

Рис. 2.7.2 Список Товарів  
Перегляд товарів на складі

Товари на складі

Вигрузка товару

Продати товар

Видати/Списати товар

Повернення товару

## Список товарів на складі

Назва товару	Кількість	Собівартість товару	Ціна продажу	Коментар
Авокадо ваговий	5	270.00	0.00	кількість в кг 
Лосось ваговий	25	195.00	0.00	кількість в кг 
Рис	20	50.00	45.00	кількість в кг, вартість продажу за порцію 200 гр 
Соевий соус	10	50.00	0.00	кількість в літрах 
Томат український	10	40.00	0.00	кількість в кг 
Цибуля ріпчаста	30	20.00	0.00	кількість в кг 

Рис. 2.7.3 Список товарів на складі

Товари на складі

Вигрузка товару

Продати товар

Видати/Списати товар

Повернення товару

Переглянути список

## Вигрузка товару на склад

Оберіть товар

Рис

Введіть собівартість

50

Введіть вартість продажу

45

Введіть кількість, яку потрібно додати (шт)

20

Введіть коментар

кількість в кг, вартість продажу за порцію 200 гр

Зберегти

Рис. 2.7.4 вигрузка товарів на склад

**Повернення товару**

Замовник: Посточальник №1

Назва товару	Кількість
Рис	10
Авокадо ваговий	20
<b>Всього:</b>	<b>30</b>

Дата оформлення: 16.06.2020 1:37:30

Видав: \_\_\_\_\_

Отримав: \_\_\_\_\_

**Квітанція на повернення**

Рис. 2.7.4. Квітанція на повернення

**Повернення товару**

Товари на складі
Вигрузка товару
Продати товар
Видати/Списати товар
Повернення товару

Переглянути повернення

ФОП:

Оберіть товар

Введіть кількість (шт)

Видалити

---

Додати товар
Зберегти

Рис. 2.7.5 Повернення товару

### Списання товару

Рис. 2.7.7 Списання товару

## 2.8 Інструкція користувача

1. Перш за все потрібно автенавторизуватись та автентифікуватись у системі.
2. Потім менеджер, що відповідає за закупівлю асортименту продуктів повинен заповнити існуючі, які планується закуповувати у постачальників
3. Далі при надходженні товару на склад потрібно заповнювати відповідну форму на вкладці "Вигрузка товару" і вигружений товар з'явиться на вкладці "Склад"
4. Після того можливо редагувати вміст товару товару на складі у разі здійснення помилки при заповненні форми вигрузки на склад (попередній пункт) за допомогою кнопки "Edit", що знаходиться напроти кожного запису на вкладці "Склад".
5. Також є функціонал списання товару (наприклад у разі закінчення терміну придатності) за допомогою вкладки "Повернення товару".

6. Існує можливість реєстрації повернення товару та створення накладної у вигляді pdf -файлу на відповідній вкладці.

## РОЗДІЛ 3. ОХОРОНА ПРАЦІ

### 3.1 Вступ

Охорона праці — це система правил і заходів, які забезпечують безпечну роботу на даному виробництві. При роботі на металорізальному верстаті необхідно передбачити ряд вимог, які б дозволили працівнику виконувати поставлене перед ним завдання в умовах, які передбаченні конструкторськими документами.

Основними заходами, які зменшують або попереджають травматизм при роботі на дільниці є автоматизація або механізація технологічного процесу

### 3.2 Керівник відділу постачання повинен

- виконувати правила внутрішнього трудового розпорядку
- не допускати сторонніх осіб на своє робоче місце;
- утримувати робоче місце в чистоті, не захаращувати його ящиками, деталями та інше;
- не торкатися оголеного дроту, арматури загального освітлення, не відкривати електрошафи, не вмикати і не вимикати (крім аварійних ситуацій) обладнання, машини, механізми;
- не стояти під вантажем
- знати і вміти надавати першу медичну допомогу потерпілим від нещасних випадків..

### 3.3 Ряд небезпечних і шкідливих виробничих факторів при роботі з ПК

Найбільш значимі з яких наступні:

- підвищений рівень напруги в електричних ланцюгах харчування і управління ПК, який може привести до електротравми оператора при відсутності заземлення або занулення обладнання (джерело - змінний струм промислової частоти 50 Гц напругою 220 В, службовець для харчування ПК, а також струми високої частоти напругою до 12 000 У систем харчування окремих схем і вузлів дисплея);
- підвищений рівень напруженості електричного і магнітного полів в широкому діапазоні частот (в тому числі від струмів промислової частоти 50 Гц від ПК, допоміжних приладів, інших електроустановок, силових кабелів, освітлювальних установок і т.п. - особливо при відсутності заземлення або занулення обладнання);

- не відповідають санітарним нормам візуальні параметри дисплеїв, особливо мають величину зерна (піксель) 0,3 мм і більше, частоту кадрової розгортки - 50-75 Гц, а також порушення візуальних параметрів у сертифікованих ПК (виникнення нестабільного зображення) через вплив на дисплей підвищених значень напруженості магнітного поля від джерел струму промислової частоти 50 Гц (так зване опосередковане вплив магнітних полів); підвищений рівень прямої і відбитої блескості; підвищена яскравість світлового зображення; підвищений рівень пульсації світлового потоку; нерівномірність розподілу яскравості в полі зору і ін .;
- надлишкові енергетичні потоки синьо-фіолетового світла від екрану дисплея у видимому діапазоні довжин електромагнітних хвиль, що знижують чіткість сприйняття зображення оком;
- знижений або підвищений рівень освітленості;
- підвищений рівень напруженості статичної електрики;
- підвищений рівень загазованості повітря (в першу чергу - з вуглекислого газу і аміаку, які утворюються при видихання), особливо в погано вентиляваних приміщеннях;
- підвищений рівень запиленості повітря робочої зони від зовнішніх джерел;
- не відповідають нормам параметри мікроклімату: підвищена температура через постійне нагрівання деталей ПК, знижена вологість, знижена або підвищена швидкість руху (рухливість) повітря робочої зони;
- порушення норм по аероіонного складу повітря, особливо в приміщеннях з розвиненою системою припливно-витяжної вентиляції і (або) з кондиціонерами; при цьому концентрація корисних для організму негативно заряджених легких іонів кисню повітря (аероіонів) може бути в 10-50 разів нижче норми, а концентрація шкідливих позитивних іонів значно перевищувати норму;
- підвищений вміст в повітрі патогенної (що викликає захворювання) мікрофлори (перш за все - стафілокока), особливо взимку при підвищеній температурі в приміщенні, поганому провітрюванні, зниженій вологості і порушення аероіонного складу повітря;
- підвищений рівень шуму від працюючих вентилятора охолодження ПК і принтера, від неврегульованою джерел люмінесцентного освітлення та ін .;
- підвищений рівень м'якого рентгенівського випромінювання від електронно-променевої трубки дисплея (цей фактор має місце тільки у старих ВДТ випуску раніше 1992 г.);
- підвищені зорові навантаження і адинамія очних м'язів, тобто їх мала рухливість при високому статичному зоровому напруженні протягом тривалого часу, що може стати причиною різних очних захворювань, особливо таких, як спазм акомодатії (втрата

можливості м'язів скорочуватися), зниження гостроти зору, зменшення запасу відносної акомодації, а потім і короткозорість;

- монотонність праці;
- підвищене розумове навантаження через великий обсяг переробляється і засвоюється інформації;
- фізичне перенапруження через нераціональну організацію робочого місця (незручні крісла, столи, відсутність підставок для тексту, для ніг і кистей рук і ін.), що в значній мірі посилює напругу м'язів хребта, ніг, рук, шиї, очей;
- підвищене нервово-емоційне напруження (додаткове шкідливе прояв роботи на ПК - при цьому прискорюється висновок з організму багатьох життєво необхідних вітамінів і макроелементів);
- зовнішні постійно діючі екологічні фактори: наявність у повітрі робочої зони шкідливих речовин (окису вуглецю, озону, аміаку, оксидів азоту, сірки і т.п.), солей важких металів і органічних сполук (фенолу, бенз (а) пірену, формальдегіду, поліхлорованих бензенів, вільних радикалів і ін.);
- різке погіршення якості повітря по аероіонного складу, збільшення в ньому вмісту різних алергенів, грибків,
- вірусів, бактерій, мікроорганізмів;
- зростання інформаційних навантажень ззовні (причому не тільки під час роботи на ПК) викликає додаткові психічні перевантаження, стреси, що також підвищує ймовірність захворювань органів зору та інших найбільш навантажених і ослаблених органів.

Крім захворювань очей, можливі захворювання серця, нирок, нервової системи, шлунково-кишкового тракту, імунної та бронхолегеневої систем.

### **3.4 Приміщення, в якому розташовані ПК, відповідно до санітарних та інших норм безпеки має бути обладнений**

- системами освітлення (природного або штучного, загального або комбінованого - за потребою);
- системою вентиляції, кондиціонування та опалення;
- пристроями захисту від ураження електричним струмом (захисним заземленням, зануленням), захисним відключенням -УЗО;
- пристроями забезпечення нормального аероіонного режиму, обспиліватель і пристроями, що забезпечують утримання в повітрі патогенної мікрофлори не вище норми;
- засобами автоматичного пожежогасіння та зв'язку у великих приміщеннях і (або) первинними засобами пожежогасіння

### **3.5 Умови, яким повинна задовольняти робоча зона**

Просторові і розмірні співвідношення між елементами робочого місця

повинні бути достатніми для:

1. розміщення працюючої людини з обліком її робочих рухів і переміщень відповідно до технологічного процесу;
2. розташування засобів керування в межах максимальної і мінімальної границь моторного простору;
3. оптимального огляду візуальної інформації;
4. зміни робочої пози і робочого положення;
5. вільного доступу до устаткування при ремонті і налагодженні;
6. раціонального розміщення основних і допоміжних засобів праці;
7. ведення записів, роботи з документами і приладами.

### **3.6 Обов'язки оператора**

Оператор не має права приступати до роботи:

- при відсутності гігієнічного сертифіката (висновку) на ПК;
- при відсутності інформації про результати атестації умов праці на даному робочому місці;
- при наявності інформації про невідповідність параметрів технологічного обладнання вимогам санітарних норм;
- при відсутності захисного заземлення (занулення) пристроїв ПК;
- при невідповідності параметрів мікроклімату (особливо по температурі і вологості) на робочому місці вимогам санітарних норм;
- при підвищеній запиленості повітря;
- при підвищеній загазованості повітря агресивними і шкідливими речовинами;
- при відсутності захисного приєкранного фільтра, що має гігієнічний сертифікат (висновок), якщо він повинен бути встановлений за результатами атестації робочого місця;
- при відключеному заземляючем провіднику захисного приєкранного фільтра;
- при виявленні несправності обладнання;
- при відсутності первинних засобів пожежогасіння (згідно з нормами, наведеними в додатку 8 до цієї Інструкції);
- при відсутності аптечки першої допомоги;
- при порушенні гігієнічних норм розміщення обладнання.

### **3.7 До початку роботи оператор зобов'язаний:**

- вимити руки з милом.
- відрегулювати освітленість на робочому місці, переконатися у відсутності відображень (відблисків) на екрані і клавіатурі.
- оглянути робоче місце і перевірити правильність підключення обладнання до електромережі;
- переконатися в наявності захисного заземлення і підключення екранного провідника до корпусу системного блоку (процесора);
- протерти спеціальною серветкою поверхню екрану і захисного фільтра;

- переконатися в наявності спеціального килимка для пристрою "миша";
- переконатися у відсутності дискет у дисководах ПК;
- перевірити правильність установки столу, крісла. При необхідності відрегулювати положення крісла, а також розташувати обладнання так, щоб виключити незручні пози і тривала напруга тіла..

### **3.8 Вимоги до робочого місця**

При роботі з персональним комп'ютером (ПК) згідно чинного законодавства мають дотримуватись наступні основні вимоги [11].

Відстань між бічними поверхнями ПК має становити не менше 1,2 м.

Відстань від екрана одного ПК до тильної поверхні другого ПК – 2,5 м.

Висота поверхні робочого столу: межах 680-800 мм.

Висота поверхні сидіння : 400-500мм.

Ширина і глибина поверхні сидіння - не менше ніж 400мм.

Кут нахилу сидіння – до 15 градусів вперед і до 5 градусів назад.

Висота спинки стільця має становити (300+-20) мм, ширина – не менше ніж 380 мм.

Кут нахилу спинки сидіння - в межах 1-30 градусів від вертикалі.

Відстань від спинки до переднього краю сидіння – від 260 до 400мм.

Інші деталі організації робочого місця наведені за наданим вище посиланням.

## ВИСНОВКИ

В даній роботі було розроблено інформаційну систему обліку поставчань ресторану «Суші Манія»

Яка дозволяє:

- З легкістю підрахувати вартість закупленого товару.
- Пришвидшити облік продуктів на складі.
- Полегшити процедуру замовлення.
- Швидко списувати товари.
- Повертати товари .
- Коментувати товари.
- Переглядати ціну товару як за одну одиницю так и за декілька .
- Захистити дані від тих хто не має доступу .

Створена система дозволяє пришвидшити роботу складу та ввіділу поставки товарів та пришвидшити роботу складу, формування і звітність замовлень.

Для цього було використано засоби Visual Studio. Angular. Web API. .Net Core. Система контролю версій Git. За допомогою AllFusion Erwin Process Modeler Ver. 7.3. була створена функціональна модель , а за допомогою AllFusion Erwin Data Modeler Ver. 7.3. – логічна і фізична моделі бази даних.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Управління ІТ проектами: Методичні рекомендації до виконання курсової роботи для студентів напряму підготовки 6.050101 «Комп'ютерні науки» денної та заочної форм навч. / Уклад. М. В. Гладка, О. А. Хлобистова. — К. : Нац. ун-т харч. технологій, 2014. — CD-R. — кафедра Інформаційних систем
2. Маклаков С.В. CASE-средства разработки информационных систем/ С.В. Маклаков. – М.: Диалог-МИФИ, 2005. – 427с.
3. М'якшило О. М. Моделювання баз даних засобами CASE-технології ERWin: Конспект лекцій з дисципліни «Структурне моделювання систем» для студ. спец. 6.080400 «Інформаційні управляючі системи та технології» напряму 0804 «Комп'ютерні науки» всіх форм навчання. – К.: НУХТ, 2008. – 60с
4. М'якшило О.М. CASE-технології у проектуванні інформаційних систем: електронний навчальний посібник для студентів вищих навчальних закладів / О.М. М'якшило, Л.Г. Загоровська,– К.: НУХТ, 2017. – 190 с.
5. Проектування інформаційних систем. [Електронний ресурс]: лабораторний практикум для студ. освітнього ступеню "бакалавр" спец. 122 “Комп'ютерні науки ” денної і заочної форм навчання. Частина 2 "Проектування клієнтського додатку" / Уклад.: О.М. М'якшило, О.В. Харкянен – К.: НУХТ, 2017 – 33 с.
  6. <https://metanit.com/sharp/aspnet5/23.1.php>
  7. [https://pidru4niki.com/12000213/marketing/vibir\\_postachalnikov\\_tovariv](https://pidru4niki.com/12000213/marketing/vibir_postachalnikov_tovariv)
  8. <https://git-scm.com/>
  9. [http://ena.lp.edu.ua/bitstream/ntb/29919/1/38\\_395-409.pdf](http://ena.lp.edu.ua/bitstream/ntb/29919/1/38_395-409.pdf)<https://getbootstrap.com/>
  10. <https://www.adobe.com/>
  11. <http://oppb.com.ua/news/gigiyenichni-vymogy-do-organizaciyi-i-obladnannya-robochyh-misc>

## ДОДАТКИ

### Додаток А

#### При підключенні БД було використано ORM entity framework core:

```
using ATLAS.Controllers.Models;
using ATLAS.Services.Interface;
using ATLAS.Services.Models;
using Microsoft.AspNetCore.Authorization;
using Microsoft.AspNetCore.Mvc;
using System;
using System.Threading;
using System.Threading.Tasks;

namespace ATLAS.Controllers
{
    [Authorize]
    [Route("api/product")]
    [ApiController]
    public class ProductController : Controller
    {
        private IProductService _productService;

        public ProductController(IProductService productDetailsService)
        {
            _productService = productDetailsService ?? throw new
ArgumentNullException(nameof(productDetailsService));
        }

        [HttpPost]
        public async Task<IActionResult> Create([FromBody]
ProductDetailParams input, CancellationToken cancellationToken)
        {
            if(input.FullPrice < 0)
            {
                ModelState.AddModelError(input.FullPrice.ToString(), "full
pricemust be > 0");
            }
            if (input.CostPrice < 0)
            {
                ModelState.AddModelError(input.CostPrice.ToString(), "cost
pricemust be > 0");
            }
            if (!ModelState.IsValid)
            {
                return BadRequest(ModelState);
            }

            var res = await _productService.CreateAsync(input,
cancellationToken);

            return Ok(res);
        }

        [HttpGet]
        public async Task<IActionResult> GetAll(CancellationToken
cancellationToken)
        {
            var res = await _productService.FindAllAsync(cancellationToken);
```

```

        return Ok(res);
    }

    [HttpGet]
    [Route("{id}")]
    public async Task<IActionResult> GetAsync(int id, CancellationToken
cancellationToken)
    {
        ProductModel res = await _productService.FindByIdAsync(id,
cancellationToken);
        if (res == null)
        {
            return NotFound();
        }
        return Ok(res);
    }

    [HttpPut]
    [Route("{id}")]
    public async Task<IActionResult> Update(int id, [FromBody]
UpdateProductDetailParams input, CancellationToken cancellationToken)
    {
        ProductModel res = await _productService.FindByIdAsync(id,
cancellationToken);
        if (res == null)
        {
            return NotFound();
        }

        if (input.FullPrice < 0)
        {
            ModelState.AddModelError(input.FullPrice.ToString(), "full
price must be > 0");
        }
        if (input.CostPrice < 0)
        {
            ModelState.AddModelError(input.CostPrice.ToString(), "cost
price must be > 0");
        }
        if (!ModelState.IsValid)
        {
            return BadRequest(ModelState);
        }

        res = await _productService.UpdateAsync(id, input,
cancellationToken);
        return Ok(res);
    }
}

using ATLAS.Controllers.Models;
using ATLAS.Data;
using ATLAS.Services.Interface;
using ATLAS.Services.Models;
using AutoMapper;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using System.Threading.Tasks;

namespace ATLAS.Services
{
    public class ProductService : IProductService
    {
        private IProductRepository _producRepository;
        private IStockService _stockService;
    }
}

```

```

        private IMapper _mapper;

        public ProductService(IProductRepository productRepository,
            IStockService stockService, IMapper mapper)
        {
            _productRepository = productRepository ?? throw new
            ArgumentNullException(nameof(productRepository));
            _stockService = stockService ?? throw new
            ArgumentNullException(nameof(stockService));
            _mapper = mapper ?? throw new
            ArgumentNullException(nameof(mapper));
        }

        public async Task<ProductModel> CreateAsync(ProductDetailParams
            input, CancellationToken cancellationToken)
        {
            ProductDto productDetailDto = _mapper.Map<ProductDto>(input);
            Product productDetail = await
            _productRepository.CreateAsync(productDetailDto, cancellationToken);
            return MapToProductDetailModel(productDetail);
        }

        public async Task<int> DeleteAsync(int id, CancellationToken
            cancellationToken)
        {
            return await _productRepository.DeleteAsync(id,
            cancellationToken);
        }

        public async Task<ICollection<ProductModel>>
            FindAllAsync(CancellationToken cancellationToken)
        {
            ICollection<Product> product = await
            _productRepository.FindAllAsync(cancellationToken);
            IDictionary<int, int> productAmount =
            _productRepository.GetProductAmounts(product.Select(p => p.Id).ToArray());

            return product.Select(p => new ProductModel
            {
                Id = p.Id,
                ProductName = p.ProductName?.Name,
                ProductNameId = p.ProductName.Id,
                FullPrice = p.FullPrice,
                CostPrice = p.CostPrice,
                Amount = productAmount[p.Id],
                Comment = p.Comment
            }).ToArray();
        }

        public async Task<ProductModel> FindByIdAsync(int id,
            CancellationToken cancellationToken)
        {
            Product p = await _productRepository.FindByIdAsync(id,
            cancellationToken);
            IDictionary<int, int> productAmount =
            _productRepository.GetProductAmounts(new[] { p.Id });
            return new ProductModel
            {
                Id = p.Id,
                ProductName = p.ProductName?.Name,
                ProductNameId = p.ProductName.Id,
                FullPrice = p.FullPrice,
                CostPrice = p.CostPrice,
                Amount = productAmount[p.Id]
            };
        }
    }

```

```

        public async Task<ICollection<ProductModel>>
FindByProductNameIdsAsync(int[] ids, CancellationToken cancellationToken)
    {
        if(ids.Length == 0)
        {
            return Array.Empty<ProductModel>();
        }

        ICollection<Product> product = await
        _producRepository.FindByProductNameIdsAsync(ids, cancellationToken);
        IDictionary<int, int> productAmount =
        _producRepository.GetProductAmounts(product.Select(p => p.Id).ToArray());

        return product.Select(p => new ProductModel
        {
            Id = p.Id,
            ProductName = p.ProductName?.Name,
            ProductNameId = p.ProductName.Id,
            FullPrice = p.FullPrice,
            CostPrice = p.CostPrice,
            Amount = productAmount[p.Id]
        }).ToArray();
    }

    public async Task<IDictionary<int, int>> GetProductAmounts(int[]
productIds, CancellationToken cancellationToken)
    {
        return _producRepository.GetProductAmounts(productIds);
    }

    public async Task<ProductModel> UpdateAsync(int id, UpdateProductDetailParams
input, CancellationToken cancellationToken)
    {
        Product productDetail = await _producRepository.UpdateAsync(id,
input.CostPrice, input.FullPrice, input.Comment, cancellationToken);
        return MapToProductDetailModel(productDetail);
    }

    private ProductModel MapToProductDetailModel(Product productDetail)
    {
        return new ProductModel
        {
            Id = productDetail.Id,
            ProductName = productDetail.ProductName?.Name,
            ProductNameId = productDetail.ProductName.Id,
            FullPrice = productDetail.FullPrice,
            CostPrice = productDetail.CostPrice
        };
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading;
using System.Threading.Tasks;
using ATLAS.Data;
using ATLAS.Services.Interface;
using ATLAS.Services.Models;
using Microsoft.EntityFrameworkCore;

namespace ATLAS.Repositories
{
    public class ProductRepository : IProductRepository
    {
        private DbContext _db;
    }
}

```

```

public ProductRepository(DatabaseContext db)
{
    _db = db;
}

public async Task<Product> CreateAsync(ProductDto productDetailDto,
CancellationToken cancellationToken)
{
    var productDetail = new Product
    {
        ProductNameId = productDetailDto.ProductNameId,
        FullPrice = productDetailDto.FullPrice,
        CostPrice = productDetailDto.CostPrice,
        Comment = productDetailDto.Comment,
        LastModified = DateTime.UtcNow
    };

    var entry = _db.Products.Add(productDetail);
    await _db.SaveChangesAsync(cancellationToken);
    return await _db.Products.Include(p =>
p.ProductName).FirstOrDefaultAsync(p => p.Id == entry.Entity.Id,
cancellationToken);
}

public async Task<ICollection<Product>>
FindAllAsync(CancellationToken cancellationToken)
{
    return await _db.Products.Include(pd =>
pd.ProductName).OrderBy(p=>p.ProductName.Name).ToArrayAsync(cancellationToken
);
}

public async Task<ICollection<Product>> FindByIdsAsync(int[] ids,
CancellationToken cancellationToken)
{
    return await _db.Products.Include(pd => pd.ProductName).Where(p
=> ids.Contains(p.Id)).ToArrayAsync(cancellationToken);
}

public async Task<Product> FindByIdAsync(int id, CancellationToken
cancellationToken)
{
    return await _db.Products.Include(pd =>
pd.ProductName).FirstOrDefaultAsync(p => p.Id == id, cancellationToken);
}

public async Task<int> DeleteAsync(int id, CancellationToken
cancellationToken)
{
    Product pd = _db.Products.FirstOrDefault(n => n.Id == id);
    if (pd != null)
    {
        _db.Products.Remove(pd);
    }

    return await _db.SaveChangesAsync(cancellationToken);
}

public async Task<Product> UpdateAsync(int id, decimal costPrice,
decimal fullPrice, string comment, CancellationToken cancellationToken)
{
    Product pd = await _db.Products.FirstOrDefaultAsync(n => n.Id ==
id);
    pd.FullPrice = fullPrice;
    pd.CostPrice = costPrice;
    pd.Comment = comment;
    pd.LastModified = DateTime.UtcNow;
}

```

```

        await _db.SaveChangesAsync(cancellationToken);
        return pd;
    }

    public async Task<ICollection<Product>>
    FindByProductNameIdsAsync(int[] ids, CancellationToken cancellationToken)
    {
        return await _db.Products
            .Include(p => p.ProductName)
            .Where(p => ids.Contains(p.ProductNameId))
            .ToArrayAsync(cancellationToken);
    }

    public IDictionary<int, int> GetProductAmounts(int[] productIds)
    {
        var products = _db.Products
            .Include(p => p.Recieveds)
            .Include(p => p.OrderDetails)
            .Include(p => p.ReturnedDetails)
            .Where(p => productIds.Contains(p.Id))
            .AsEnumerable();

        return products.Select(p => new
        {
            p.Id,
            Amount = p.Recieveds.Sum(r => r.ProductAmount)
                + p.ReturnedDetails.Sum(r => r.ProductAmount)
                - p.OrderDetails.Sum(o => o.ProductAmount)
        })
            .ToDictionary(p => p.Id, v => v.Amount);
    }
}
}

```

## Додаток Б

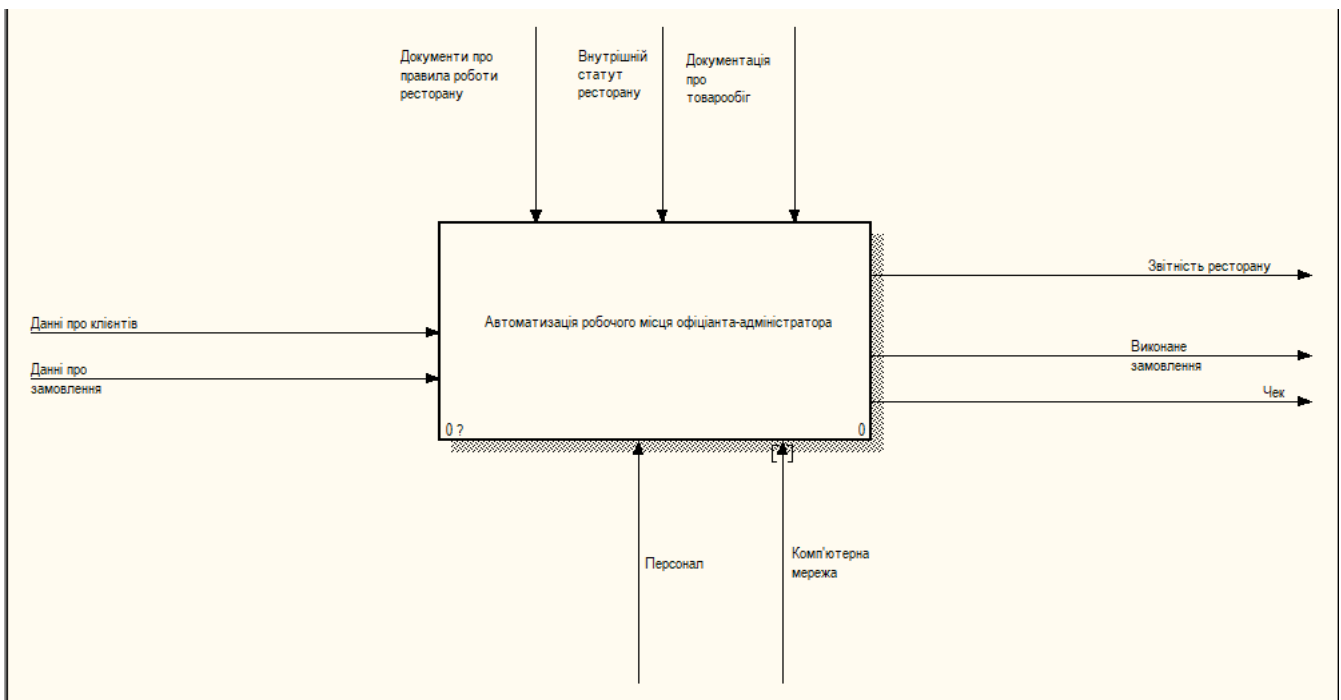


Рис1. Контекстна діаграма функціональної моделі

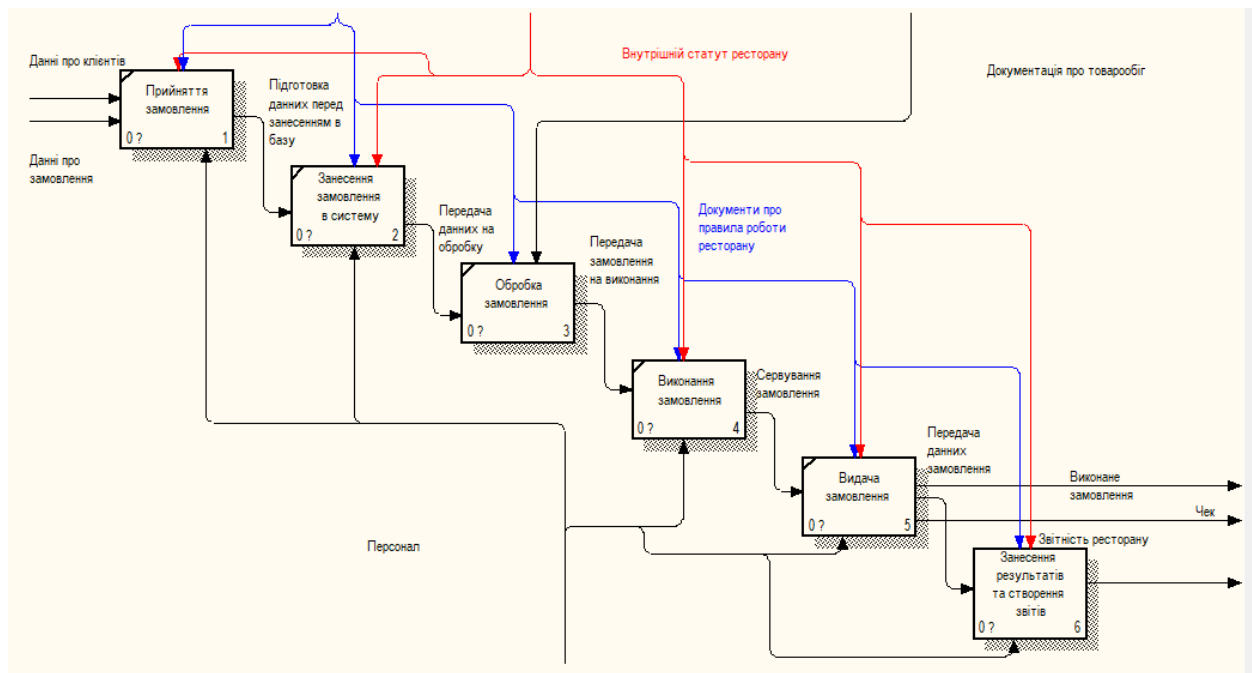


Рис2. Діаграма деталізації на першому рівні

Додаток В

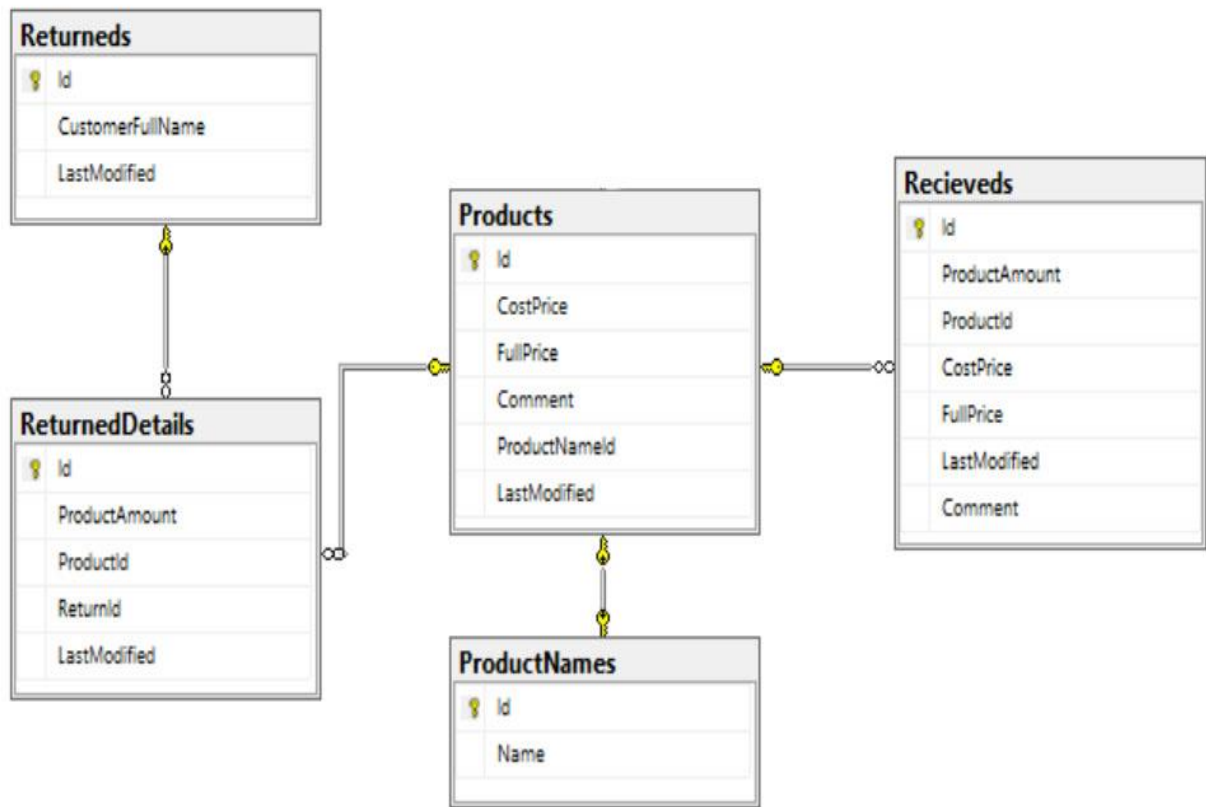


Рис1. Фізичная модель бази даних