

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ

ВІДДІЛЕННЯ  
«ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА КОМП'ЮТЕРНІ НАУКИ»

ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ  
«КОМП'ЮТЕРНІ НАУКИ»

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

до кваліфікаційної роботи освітнього ступеня фаховий молодший бакалавр  
за спеціальністю 122 Комп'ютерні науки

на тему:

**«Інформаційна система автоматизації роботи  
закладу громадського харчування»**

Виконав здобувач освіти групи П-42  
Коцюба Сергій Олександрович

Керівник роботи:  
Устименко Людмила Миколаївна

Рецензент:  
\_\_\_\_\_

## Зміст

|                                                                  |    |
|------------------------------------------------------------------|----|
| Вступ                                                            | 5  |
| Розділ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ        | 8  |
| 1.1 Основні задачі автоматизації закладу громадського харчування | 8  |
| 1.2 Огляд існуючих рішень                                        | 9  |
| 1.3 Функціональні вимоги до програми                             | 11 |
| 1.4 Вибір технології                                             | 12 |
| 1.5 Постановка задачі                                            | 14 |
| Висновки до розділу 1                                            | 15 |
| Розділ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ                        | 16 |
| 2.1 Модель бази даних                                            | 16 |
| 2.2 Список основних форм та їх призначення                       | 27 |
| 2.3 Архітектура програмного засобу                               | 34 |
| 2.4 Програмна реалізація                                         | 38 |
| 2.5 Тестування програми                                          | 42 |
| Висновки до розділу 2                                            | 46 |
| Розділ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ                     | 49 |
| 3.1 Мінімальні вимоги до системи                                 | 49 |
| 3.2 Інструкція для роботи з програмою                            | 51 |
| Висновки до розділу 3                                            | 58 |
| ВИСНОВКИ                                                         | 59 |
| Перелік літературних джерел                                      | 61 |
| Додаток А. Код для роботи з базою даних                          | 64 |
| Додаток Б. Програмний код модулів                                | 66 |

|             |             |                 |               |             |                                                                                              |              |             |               |
|-------------|-------------|-----------------|---------------|-------------|----------------------------------------------------------------------------------------------|--------------|-------------|---------------|
|             |             |                 |               |             | <b>КРФМБ.122.042.039.2026.ПЗ</b>                                                             |              |             |               |
|             |             |                 |               |             |                                                                                              |              |             |               |
| <b>Змн.</b> | <b>Арк.</b> | <b>№ докум.</b> | <b>Підпис</b> | <b>Дата</b> | <b>Інформаційна система<br/>автоматизації роботи<br/>закладу громадського<br/>харчування</b> | <b>Літ.</b>  | <b>Арк.</b> | <b>Аркуші</b> |
| Розробив    |             | Коцюба С О      |               |             |                                                                                              |              |             |               |
| Перевірів   |             | Устименко Л.М.  |               |             |                                                                                              |              | 3           | 69            |
| Рецензент   |             | .               |               |             |                                                                                              | <b>ЖАТФК</b> |             |               |
| Н. Контр.   |             |                 |               |             |                                                                                              |              |             |               |
| Затвердив.  |             | Гнатюк О.Ф.     |               |             |                                                                                              |              |             |               |

## РЕФЕРАТ

Кваліфікаційна робота містить: 69 сторінок, 11 рисунків, 11 таблиць, 2 додатки, 24 джерела.

Метою роботи є розробка інформаційної системи для автоматизації основних бізнес-процесів закладу громадського харчування, що охоплює управління меню, обробку замовлень, резервування столів, складський облік, управління персоналом та формування звітності.

Об'єктом дослідження виступають процеси діяльності закладу громадського харчування, які потребують автоматизації.

Предметом дослідження обрано методи та засоби автоматизації роботи сучасного ресторанного господарства із застосуванням новітніх інформаційних технологій.

Під час виконання роботи застосовано такі методи дослідження, як системний аналіз, об'єктно-орієнтоване проектування, моделювання баз даних та тестування програмного забезпечення.

Результати роботи включають проектування реляційної бази даних із дев'яти таблиць та розробку клієнт-серверного застосунку на платформі .NET 9 за допомогою мови C#, технологій WPF, Entity Framework Core та SQLite. Також було реалізовано механізми автентифікації з розмежуванням прав доступу для чотирьох ролей та функціонал для створення аналітичних звітів із візуалізацією даних.

Практичне значення отриманих результатів полягає у можливості підвищити швидкість обслуговування, мінімізувати помилки персоналу, автоматизувати контроль залишків продуктів та зменшити витрати на адміністрування.

**Ключові слова:** ІНФОРМАЦІЙНА СИСТЕМА, АВТОМАТИЗАЦІЯ, ЗАКЛАД ГРОМАДСЬКОГО ХАРЧУВАННЯ, .NET, WPF, C#, SQLite, БАЗА ДАНИХ, MVVM, ЗАМОВЛЕННЯ, СКЛАДСЬКИЙ ОБЛІК, ЗВІТНІСТЬ.

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

|             |                                                                                                                                |
|-------------|--------------------------------------------------------------------------------------------------------------------------------|
| API         | Application Programming Interface – програмний інтерфейс застосунку                                                            |
| CSV         | Comma-Separated Values – текстовий формат для зберігання табличних даних, де значення розділяються комами або крапками з комою |
| DBMS        | Database Management System – система керування базами даних                                                                    |
| DI          | Dependency Injection – механізм впровадження залежностей                                                                       |
| DTO         | Data Transfer Object – об’єкт передавання даних                                                                                |
| EF Core     | Entity Framework Core – об’єктно-реляційний маппер для платформи .NET                                                          |
| ER-діаграма | Entity-Relationship Diagram – діаграма «сутність-зв’язок»                                                                      |
| IDE         | Integrated Development Environment – інтегроване середовище розробки                                                           |
| JSON        | JavaScript Object Notation – текстовий формат обміну даними                                                                    |
| LINQ        | Language Integrated Query – засіб для формування запитів безпосередньо в мові C#                                               |
| MVVM        | Model-View-ViewModel – архітектурний патерн, який розділяє логіку програми, дані та інтерфейс користувача                      |
| ORM         | Object-Relational Mapping – технологія зіставлення об’єктів програми з таблицями реляційної бази даних                         |
| OOP         | Object-Oriented Programming – об’єктно-орієнтоване програмування                                                               |
| PC          | Personal Computer – персональний комп’ютер                                                                                     |
| RAM         | Random Access Memory – оперативний запам’ятовувальний пристрій                                                                 |
| SDK         | Software Development Kit – набір інструментів для розробки програмного забезпечення                                            |
| SQL         | Structured Query Language – структурована мова запитів                                                                         |
| UI          | User Interface – інтерфейс користувача                                                                                         |
| UML         | Unified Modeling Language – уніфікована мова моделювання                                                                       |
| WPF         | Windows Presentation Foundation – технологія для створення клієнтських застосунків з багатим графічним інтерфейсом             |
| XAML        | eXtensible Application Markup Language – мова розмітки для декларативного опису інтерфейсу користувача                         |

## ВСТУП

У сучасних умовах розвитку ринкової економіки та зростаючої конкуренції в сфері громадського харчування ефективність діяльності закладів (ресторанів, кафе, їдалень) значною мірою залежить від якості управлінських рішень та оперативності обробки інформації. Традиційні методи ведення обліку, засновані на ручному введенні даних або використанні розрізнених електронних таблиць, призводять до дублювання інформації, помилок персоналу, затримок у приготуванні страв, нестачі продуктів на складі та, як наслідок, до зниження рівня сервісу. У зв'язку з цим виникає об'єктивна необхідність розробки спеціалізованого програмного забезпечення, яке б комплексно автоматизувало основні бізнес-процеси закладу громадського харчування: від прийняття замовлення й контролю залишків інгредієнтів до генерації аналітичних звітів. Таким чином, тема кваліфікаційної роботи є актуальною та має практичну значущість для підвищення ефективності діяльності сучасного закладу громадського харчування.

Метою кваліфікаційної роботи є розробка інформаційної системи автоматизації роботи закладу громадського харчування, яка забезпечує централізоване управління меню, обробку замовлень, резервування столів, контроль залишків продуктів на складі, ведення обліку співробітників та формування аналітичних звітів.

Для досягнення поставленої мети необхідно вирішити наступні **завдання**:

1. Провести аналіз предметної області та виявити основні бізнес-процеси, що потребують автоматизації.
2. Проаналізувати існуючі програмні рішення для автоматизації закладів громадського харчування, визначити їх переваги та недоліки.
3. Сформулювати вимоги до інформаційної системи, що розробляється.
4. Розробити архітектуру системи, обрати технологічний стек для реалізації (WPF, .NET 9, SQLite, Entity Framework Core).
5. Спроектувати логічну та фізичну модель бази даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 5    |

6. Реалізувати програмну систему з урахуванням принципів MVC/MVVM, забезпечити зручний інтерфейс користувача українською мовою.
7. Розробити підсистему розмежування прав доступу на основі ролей (адміністратор, менеджер, офіціант, кухар).
8. Виконати тестування розробленої системи, оцінити її ефективність порівняно з неавтоматизованим обліком.

**Об'єктом дослідження** є процеси управління закладом громадського харчування, зокрема: обробка замовлень, управління меню, резервування столів, контроль складських запасів, облік персоналу та формування звітності.

**Предметом дослідження** виступають методи та засоби автоматизації зазначених процесів за допомогою сучасних інформаційних технологій, зокрема платформи .NET 9 та системи керування базами даних SQLite.

У процесі виконання роботи застосовано методи системного аналізу, об'єктно-орієнтованого проектування, моделювання бізнес-процесів, а також технології розробки клієнт-серверних додатків з використанням мови C#, фреймворку WPF (Windows Presentation Foundation) та Entity Framework Core.

Наукова новизна роботи полягає в розробці комплексної архітектури інформаційної системи для закладів громадського харчування, яка на відміну від існуючих аналогів:

- інтегрує управління замовленнями та контроль складських залишків у режимі реального часу з автоматичним списанням інгредієнтів;
- забезпечує гнучку систему ролей, адаптовану до українського ринку;
- підтримує формування деталізованих фінансових та аналітичних звітів на основі актуальних даних.

Розроблену інформаційну систему може бути впроваджено в реальних закладах громадського харчування для оптимізації роботи персоналу, зменшення втрат через нестачу продуктів, пришвидшення обслуговування клієнтів та підвищення прозорості фінансових потоків. Система відрізняється низьким порогом входження, не потребує спеціального адміністрування

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 6    |

серверів (локальна база даних SQLite) та може використовуватися на звичайних персональних комп'ютерах.

Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків. Перший розділ присвячено аналізу предметної області та постановці вимог до системи. У другому розділі описано проєктування системи, вибір технологій та моделювання бази даних. Третій розділ містить опис реалізації інформаційної системи, її тестування та аналіз ефективності.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
|      |      |          |        |      |                           | 7    |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |      |

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАВДАННЯ

### 1.1 Основні задачі автоматизації закладу громадського харчування

Діяльність сучасного закладу громадського харчування (ресторану, кафе, їдальні) передбачає виконання низки взаємопов'язаних бізнес-процесів, серед яких можна виділити:

- управління меню (створення, редагування, архівування страв та напоїв);
- обробка замовлень (прийом замовлення, передача на кухню, розрахунок вартості з урахуванням знижок, фіксація оплати);
- резервування столів і контроль їхнього статусу;
- ведення складського обліку (контроль залишків інгредієнтів, автоматичне списання при приготуванні, сигналізація про низький рівень запасів);
- управління персоналом (реєстрація співробітників, призначення ролей, автентифікація);
- формування звітності (продажі, популярні страви, фінансові показники, ефективність працівників).

Неавтоматизоване або частково автоматизоване виконання цих процесів (наприклад, за допомогою паперових журналів, електронних таблиць чи застарілих систем) призводить до таких негативних наслідків:

- **дублювання даних** – одна й та сама інформація вводиться багаторазово в різні форми;
- **помилки персоналу** – помилки в підрахунку вартості замовлення, невірне списання інгредієнтів;
- **нестача продуктів** – відсутність оперативного контролю залишків, неможливість спрогнозувати поповнення;
- **низька швидкість обслуговування** – час очікування замовлення зростає через необхідність уточнювати наявність інгредієнтів;
- **ускладнення аналізу** – формування підсумкових звітів вимагає ручного збору даних з різних джерел.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 8    |

Отже, **основною задачею автоматизації** є створення єдиної централізованої інформаційної системи, яка забезпечує:

1. **Цілісність та актуальність даних** – усі підрозділи працюють з єдиною базою даних, зміни в одному місці автоматично відображаються всюди.
2. **Зменшення впливу людського фактора** – автоматичний розрахунок сум, списання інгредієнтів, перевірка наявності продуктів.
3. **Розмежування прав доступу** – залежно від посади (адміністратор, менеджер, офіціант, кухар) співробітник отримує доступ лише до тих функцій, які необхідні для роботи.
4. **Прозорість бізнес-процесів** – фіксація всіх критичних подій (час створення замовлення, зміна статусу, оплата) дає змогу контролювати роботу персоналу.
5. **Оперативне отримання звітності** – формування фінансових, складських та аналітичних звітів в один клік.

## 1.2 Огляд існуючих рішень

Для автоматизації роботи закладів громадського харчування на ринку представлено декілька типів програмних продуктів, які умовно можна розділити на три категорії.

### 1.2.1 Зарубіжні хмарні системи

До цієї категорії належать такі рішення, як *Toast*, *Lightspeed Restaurant*, *Square for Restaurants*, *iiko Cloud* (хоча *iiko* активно використовується й в Україні, проте останнім часом акцент зміщується в бік хмарної моделі). Основні характеристики:

- передплатна модель (щомісячна плата);
- доступ через веб-інтерфейс або мобільний додаток;
- постійна підтримка та оновлення з боку розробника;
- необхідний стабільний доступ до Інтернету;
- дані зберігаються на серверах компанії-провайдера.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 9    |

Переваги: масштабованість, відсутність витрат на адміністрування серверного обладнання, мультиплатформність.

Недоліки: вартість (до \$69–100/міс для середнього ресторану), ризик прив'язки до одного постачальника, можливі проблеми з відповідністю українському законодавству щодо зберігання персональних даних.

### 1.2.2 Локальні стаціонарні системи

Найбільш поширені рішення в Україні: *Poster*, *R-keeper*, *iiko Prem*, *Shtrih-M: Кафе*. Такі системи встановлюються на сервері закладу, працюють без постійного доступу до Інтернету (може використовуватися ліцензійний ключ або USB-ключ). Зазвичай включають окремі робочі місця для касира, кухаря, адміністратора.

Переваги: повний контроль над даними, відсутність абонентської плати (зазвичай одноразова ліцензія), можливість глибокої кастомізації під специфіку конкретного закладу.

Недоліки: висока вартість ліцензій і супроводу, складність налаштування, потреба у виділеному сервері або потужному комп'ютері, технічні вимоги до адміністратора (знання мереж, ОС Windows Server).

### 1.2.3 Open-source та саморобні рішення

Існують проекти з відкритим кодом (наприклад, *openpos*, *UniCentaPOS*), але вони часто мають проблеми з документацією, підтримкою української мови, рідко адаптовані до реалій українського ринку (наприклад, використання фіскальних реєстраторів). Невеликі кафе нерідко використовують саморобні системи на основі Microsoft Excel або Google Sheets, однак такі рішення не забезпечують необхідної безпеки, надійності та продуктивності.

Жодне з розглянутих рішень повністю не задовольняє одночасно всім вимогам: доступна ціна (без щомісячних платежів), підтримка української мови інтерфейсу, простота встановлення (на один комп'ютер без виділеного сервера), гнучкість у налаштуванні функціональності. Тому доцільно

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 10   |

розробити власну інформаційну систему, орієнтовану на роботу в невеликому закладі громадського харчування, з урахуванням специфіки українського законодавства та бізнес-процесів.

### 1.3 Функціональні вимоги до програми

На основі аналізу предметної області сформульовано наступні **функціональні вимоги** до системи, що розробляється:

#### 1. Керування меню

- додавання / редагування / видалення позицій меню (назва, опис, ціна, категорія, ознака доступності);
- архівування застарілих страв;
- пошук та фільтрація страв за назвою та категорією.

#### 2. Обробка замовлень

- створення нового замовлення з вибором столу та переліку страв;
- розрахунок вартості замовлення (сума, податок);
- підтримка знижки на замовлення (у відсотках);
- зміна статусу замовлення (нове, готується, готове, подано, оплачено, скасовано);
- перегляд активних замовлень;
- автоматичне списання інгредієнтів зі складу під час додавання страви.

#### 3. Резервування столів

- перегляд схеми столів та їхнього статусу (вільний, зайнятий, заброньований, прибирання);
- створення / зміна / скасування резервування на конкретну дату та час;
- фільтрація резервувань за датою.

#### 4. Складський облік

- ведення реєстру інгредієнтів (назва, кількість, одиниця виміру, мінімальний рівень, постачальник, ціна);
- автоматичне зменшення кількості при приготуванні страв згідно з рецептами;

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 11   |

- інформування про досягнення мінімального рівня запасів;
- ручне поповнення / коригування кількості.

## 5. Управління персоналом

- реєстрація співробітників (ПІБ, логін, пароль, роль, контактні дані);
- автентифікація за логіном і паролем;
- зміна пароля (адміністратором або сам співробітник через скидання);
- деактивація облікового запису.

## 6. Розмежування прав доступу (ролі):

- *Адміністратор* – повний доступ до всіх модулів;
- *Менеджер* – доступ до звітів, управління меню, складом, персоналом (без права видалення критичних даних);
- *Офіціант* – робота із замовленнями та резервуванням столів;
- *Кухар* – перегляд активних замовлень, зміна статусу на «готове».

## 7. Звітність

- звіт про продажі за період (суми, податки, кількість замовлень, середній чек);
- звіт про популярні страви (кількість проданих одиниць, виручка);
- фінансовий звіт (доходи, витрати, прибуток, маржа);
- інвентаризаційний звіт (список позицій, загальна вартість запасу);
- продуктивність співробітників (кількість оброблених замовлень, сума продажів).

### 1.4 Вибір технології

На етапі проектування необхідно обрати технологічний стек, який забезпечує надійність, зручність розробки, можливість розгортання на локальному комп'ютері без додаткового серверного програмного забезпечення та мінімальної вартості володіння.

#### 1.4.1 Вибір платформи та мови програмування

- **Мова програмування** – C#. Сучасна об'єктно-орієнтована мова, що підтримує асинхронне програмування, має велику екосистему бібліотек

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 12   |

і активно використовується для розробки десктопних рішень під управлінням Windows.

- **Платформа** – .NET 9 (остання стабільна версія). Забезпечує високу продуктивність, сучасні засоби роботи з базами даних (Entity Framework Core), можливість використання патерну MVVM.

#### 1.4.2 Вибір фреймворку для користувацького інтерфейсу

Оскільки система орієнтована на роботу в середовищі Windows (робочі місця адміністратора, касира, кухаря), обрано **WPF (Windows Presentation Foundation)**. Переваги:

- роздільна архітектура (XAML для розмітки, C# – для логіки);
- підтримка патерну MVVM, що спрощує тестування та супровід;
- гнучка система стилізації та шаблонів, що дозволяє створити сучасний інтерфейс.

#### 1.4.3 Вибір СУБД

Система має бути самодостатньою і не вимагати встановлення окремого сервера баз даних. Обрано **SQLite** – вбудовувану реляційну СУБД, яка:

- зберігає базу даних в одному файлі;
- не потребує сервера та додаткового адміністрування;
- підтримує транзакції (ACID);
- інтегрується з Entity Framework Core.

Обмеження SQLite (значні обсяги даних, високе конкурентне навантаження) не є критичними для невеликого закладу громадського харчування (до 10 робочих місць, десятки тисяч транзакцій на місяць).

#### 1.4.4 Вибір ORM

Для роботи з базою даних обрано **Entity Framework Core** – об'єктно-реляційний маппер, який:

- дозволяє працювати з даними в термінах об'єктів C#, без написання SQL-запитів;
- автоматично генерує схему бази даних на основі моделей;

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 13   |

- забезпечує міграції для оновлення схеми.

#### 1.4.5 Додаткові бібліотеки

- **CommunityToolkit.Mvvm** – набір засобів для реалізації патерну MVVM (атрибути [ObservableProperty], [RelayCommand] тощо).
- **LiveChartsCore** – графічні компоненти для побудови діаграм у звітах.
- **Serilog** – бібліотека ведення журналу подій (логів).
- **Microsoft.Extensions.DependencyInjection** – вбудований контейнер DI для керування залежностями.

### 1.5 Постановка задачі

На основі проведеного аналізу предметної галузі, огляду існуючих рішень та сформульованих функціональних вимог необхідно розробити інформаційну систему, яка повинна вирішити такі **задачі**:

1. **Створити локальну базу даних** для зберігання інформації про страви, замовлення, столи, резервування, інгредієнти, співробітників та логів продажів.
2. **Реалізувати модуль автентифікації** з розмежуванням прав доступу відповідно до ролей (адміністратор, менеджер, офіціант, кухар).
3. **Розробити зручний інтерфейс користувача** українською мовою на базі WPF, що надає доступ до всіх модулів.
4. **Забезпечити основні бізнес-процеси**:
  - ведення меню (CRUD операції);
  - створення та обробка замовлень (з автоматичним контролем залишків);
  - резервування столів;
  - складський облік;
  - управління співробітниками;
  - формування аналітичних звітів.

5. **Реалізувати функцію резервного копіювання бази даних та її відновлення.**
6. **Провести тестування системи** (компонентне, інтеграційне), виявити та усунути критичні помилки.
7. **Підготувати інсталяційний пакет** (копію готового додатку) та коротку інструкцію для користувача.

### **Висновки до розділу 1**

У першому розділі кваліфікаційної роботи виконано аналіз предметної області автоматизації закладу громадського харчування. Основні результати:

1. Визначено ключові бізнес-процеси: управління меню, замовленнями, столами, складом, персоналом, формування звітів.
2. Проведено огляд існуючих програмних рішень (хмарні, локальні, open-source), виявлено їхні сильні та слабкі сторони. Зроблено висновок про доцільність розробки власної системи для невеликих закладів, орієнтованої на локальне встановлення без абонентської плати.
3. Сформульовано функціональні вимоги до системи, а також вимоги щодо розмежування прав доступу на основі чотирьох ролей.
4. Обґрунтовано вибір технологій: C#, .NET 9, WPF, SQLite, Entity Framework Core, CommunityToolkit.Mvvm, LiveChartsCore, Serilog.
5. Поставлено детальне завдання на розробку, яке охоплює всі етапи – від створення бази даних до інсталяції та тестування.

## РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАСОБУ

### 2.1 Модель бази даних

#### 2.1.1 Опис предметної галузі та основних сутностей

Предметна галузь охоплює автоматизацію ключових бізнес-процесів закладу громадського харчування. Основними сутностями, що підлягають обліку, є:

- **Страви (MenuItem)**, які пропонуються відвідувачам.
- **Інгредієнти (InventoryItem)**, що зберігаються на складі.
- **Рецепти (RecipeIngredient)**, які визначають, які інгредієнти та в якій кількості входять до кожної страви.
- **Столи (Table)**, що розташовані в залі закладу.
- **Замовлення (Order)**, які створюються для обслуговування клієнтів.
- **Позиції замовлення (OrderItem)**, що фіксують перелік страв у складі замовлення.
- **Бронювання (Reservation)**, які здійснюються клієнтами на певний час.
- **Співробітники (Employee)**, які працюють у закладі.
- **Логи продажів (SalesLog)**, що використовуються для формування звітності.

Між цими сутностями існують логічні зв'язки, які забезпечують цілісність даних та описують реальні процеси роботи ресторану.

## 2.1.2 ER-діаграма

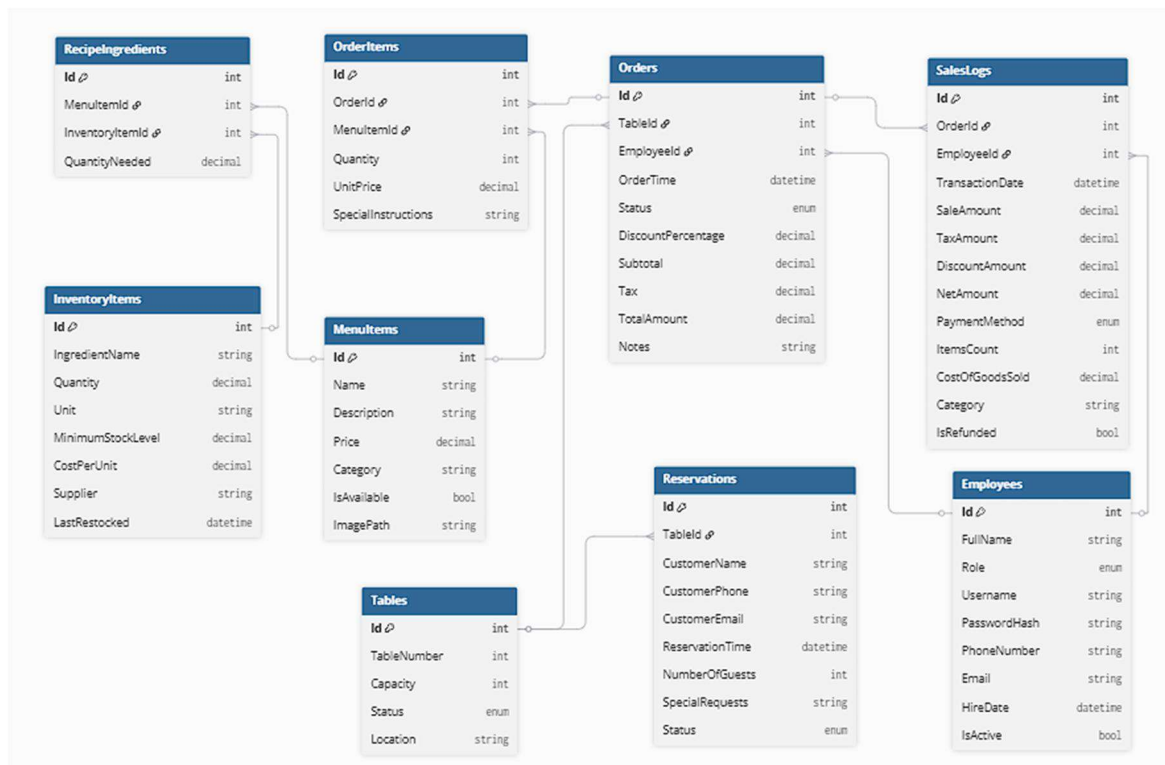


Рисунок 2.1.1. – ER-діаграма

### Сутностей та їх атрибути:

**1. Сутність MenuItem (Страви)** – зберігає список всіх страв, що пропонуються в меню.

#### Атрибути:

- Id (первинний ключ) – простий, однозначний, автоінкремент.
- Name – простий, обов’язковий.
- Description – простий, текстовий (необов’язковий).
- Price – простий, числовий.
- Category – простий, рядковий (наприклад, “Супи”, “Напої”).
- IsAvailable – простий, логічний (доступна страва чи ні).

**2. Сутність InventoryItem (Інгредієнти)** – облік продуктів на складі.

#### Атрибути:

- Id (первинний ключ) – простий.
- IngredientName – простий.
- Quantity – простий, числовий (поточна кількість на складі).
- Unit – простий (одиниця виміру: кг, л, шт).

- MinimumStockLevel – простий (мінімально допустимий запас для сигналізації).

**3. Сутність RecipeIngredient (Рецепти)** – реалізує зв’язок “багато-до-багатьох” між MenuItem та InventoryItem з додатковим атрибутом.

**Атрибути:**

- Id (первинний ключ).
- MenuItemId (зовнішній ключ до MenuItem.Id).
- InventoryItemId (зовнішній ключ до InventoryItem.Id).
- QuantityNeeded – простий атрибут зв’язку (скільки одиниць інгредієнта потрібно на одну страву).

**4. Сутність Table (Столи)** – управління столами в залі.

**Атрибути:**

- Id (первинний ключ).
- TableNumber – простий.
- Capacity – простий (кількість місць).
- Status – простий (перелік: вільний, зайнятий, заброньований, прибирання).

**5. Сутність Order (Замовлення)** – фіксація факту обслуговування клієнта.

**Атрибути:**

- Id (первинний ключ).
- TableId, EmployeeId (зовнішні ключі).
- OrderTime – простий.
- Status – простий (нове, готується, готове, подано, оплачено, скасовано).
- Subtotal, Tax, TotalAmount, DiscountPercentage – прості числові.

**6. Сутність OrderItem (Позиція замовлення)** – фіксує які страви та в якій кількості були замовлені.

**Атрибути:**

- Id, OrderId, MenuItemId (зовнішні ключі).

- Quantity – простий.
- UnitPrice – простий (ціна страви на момент замовлення, береться з MenuItem.Price).

**7. Сутність Reservation (Бронювання)** – попереднє замовлення столу клієнтом.

**Атрибути:**

- Id, TableId (зовнішній ключ).
- CustomerName, CustomerPhone, CustomerEmail – прості текстові.
- ReservationTime (дата та час).
- NumberOfGuests, SpecialRequests.
- Status – простий (підтверджено, розміщені, завершено, скасовано).

**8. Сутність Employee (Співробітник)** – автентифікація та розмежування прав.

**Атрибути:**

- Id (первинний ключ).
- FullName, Username, PasswordHash, Role, PhoneNumber, Email, HireDate, IsActive.

**9. Сутність SalesLog (Лог продажів)** – зберігає фінансову інформацію для звітності, не залежну від подальших змін у замовленнях.

**Атрибути:** деталізована сума, податки, спосіб оплати, собівартість тощо.

**Зв'язки:**

1. MenuItem (1) : (N) RecipeIngredient

- Тип: 1:N.
- Одна страва може містити багато інгредієнтів у рецепті.

2. InventoryItem (1) : (N) RecipeIngredient

- Тип: 1:N.
- Один інгредієнт може входити до багатьох рецептів.

3. Table (1) : (N) Order

- Тип: 1:N.
- За одним столом можна зробити багато замовлень (у різний час).

4. Employee (1) : (N) Order

- Тип: 1:N.
- Один співробітник може прийняти багато замовлень.

5. Order (1) : (N) OrderItem

- Тип: 1:N.
- Одне замовлення може складатися з багатьох позицій (страв).

6. MenuItem (1) : (N) OrderItem

- Тип: 1:N.
- Одна страва може зустрічатися в багатьох замовленнях.

7. Table (1) : (N) Reservation

- Тип: 1:N.
- Один стіл може резервуватися багато разів у різний час.

8. Order (1) : (1) SalesLog (не показано на діаграмі як прямий зв'язок, реалізовано через OrderId)

- Тип: 1:1.
- На одне оплачене замовлення створюється один запис у логах продажів.

Модель забезпечує цілісність даних та уникнення надмірності.

**1. Нормалізація:**

- Інгредієнти зберігаються окремо від страв та зв'язуються через таблицю RecipeIngredient. Це унеможливило повторення назв і характеристик інгредієнтів (наприклад, “Картопля”) для кожної страви.
- Позиції замовлення (OrderItem) винесені в окрему таблицю, що дозволяє уникнути зберігання списку страв у вигляді текстового рядка в таблиці Order.

**2. Первинні ключі** гарантують унікальність кожного запису.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
|      |      |          |        |      |                           | 20   |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |      |

### 3. Зовнішні ключі (Foreign Keys) забезпечують **референтну цілісність**:

- Не можна додати позицію замовлення для неіснуючої страви або замовлення.
- При видаленні страви, всі пов'язані записи в RecipeIngredient та OrderItem автоматично видаляються або обмежуються за допомогою каскадних ефектів (ON DELETE CASCADE або RESTRICT).
- При спробі видалити інгредієнт, який використовується в будь-якому рецепті, СУБД блокує операцію, що запобігає появі “вісячих” посилань.

### 4. Використання типів enum для полів Status (в Order, Table, Reservation) забезпечує **цілісність домену** – неможливо записати некоректне значення статусу.

### 5. Відсутність дублювання: Такі дані, як назва страви, ціна, опис, зберігаються в одному місці (MenuItem), а не повторюються в кожному OrderItem. Це зменшує надмірність та спрощує внесення змін (ціну страви достатньо змінити в одному місці).

Така структура забезпечує несуперечливість, ефективне зберігання та швидкий пошук даних у розробленій інформаційній системі.

#### 2.1.3 Фізична модель бази даних

Фізичну модель бази даних реалізовано з використанням системи керування реляційними базами даних **SQLite**. Ця СУБД обрана як вбудовувана, що не потребує окремого сервера, ідеально підходить для локального розгортання в невеликому закладі громадського харчування. Усі дані зберігаються в одному файлі RestaurantDB.sqlite.

Нижче наведено детальний опис кожної таблиці, її полів, типів даних, первинних та зовнішніх ключів.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 21   |

Таблиця 2.1.3.1 – Структура таблиці MenuItem (Страви)

| Поле        | Тип даних     | Обмеження                  | Опис                                 |
|-------------|---------------|----------------------------|--------------------------------------|
| Id          | INTEGER       | PRIMARY KEY, AUTOINCREMENT | Унікальний ідентифікатор страви      |
| Name        | TEXT          | NOT NULL                   | Назва страви                         |
| Description | TEXT          |                            | Опис страви                          |
| Price       | DECIMAL(18,2) | NOT NULL                   | Ціна                                 |
| Category    | TEXT          | NOT NULL                   | Категорія (Супи, Салати, Напої тощо) |
| IsAvailable | INTEGER       | NOT NULL, DEFAULT 1        | Доступність (1 – так, 0 – ні)        |
| ImagePath   | TEXT          |                            | Шлях до зображення страви            |

Таблиця 2.1.3.2 – Структура таблиці InventoryItems (Інгредієнти складу)

| Поле              | Тип даних     | Обмеження                  | Опис                                 |
|-------------------|---------------|----------------------------|--------------------------------------|
| Id                | INTEGER       | PRIMARY KEY, AUTOINCREMENT | Унікальний ідентифікатор інгредієнта |
| IngredientName    | TEXT          | NOT NULL                   | Назва інгредієнта                    |
| Quantity          | DECIMAL(18,2) | NOT NULL, DEFAULT 0        | Поточна кількість на складі          |
| Unit              | TEXT          | NOT NULL                   | Одиниця виміру (кг, л, шт)           |
| MinimumStockLevel | DECIMAL(18,2) | NOT NULL                   | Мінімальний рівень запасу            |
| CostPerUnit       | DECIMAL(18,2) |                            | Вартість за одиницю                  |
| Supplier          | TEXT          |                            | Назва постачальника                  |
| LastRestocked     | DATETIME      |                            | Дата останнього поповнення           |

Таблиця 2.1.3.3 – Структура таблиці RecipeIngredients (Рецепти)

| Поле            | Тип даних     | Обмеження                                                   | Опис                                           |
|-----------------|---------------|-------------------------------------------------------------|------------------------------------------------|
| Id              | INTEGER       | PRIMARY KEY, AUTOINCREMENT                                  | Унікальний ідентифікатор запису                |
| MenuItemId      | INTEGER       | FOREIGN KEY REFERENCES MenuItem(Id) ON DELETE CASCADE       | Ідентифікатор страви                           |
| InventoryItemId | INTEGER       | FOREIGN KEY REFERENCES InventoryItem(Id) ON DELETE RESTRICT | Ідентифікатор інгредієнта                      |
| QuantityNeeded  | DECIMAL(18,2) | NOT NULL                                                    | Необхідна кількість інгредієнта на одну порцію |

Таблиця 2.1.3.4 – Структура таблиці Tables (Столи)

| Поле        | Тип даних | Обмеження                  | Опис                                                                  |
|-------------|-----------|----------------------------|-----------------------------------------------------------------------|
| Id          | INTEGER   | PRIMARY KEY, AUTOINCREMENT | Унікальний ідентифікатор столу                                        |
| TableNumber | INTEGER   | NOT NULL, UNIQUE           | Номер столу                                                           |
| Capacity    | INTEGER   | NOT NULL                   | Кількість місць                                                       |
| Status      | INTEGER   | NOT NULL, DEFAULT 0        | Статус (0 – вільний, 1 – зайнятий, 2 – заброньований, 3 – прибирання) |
| Location    | TEXT      |                            | Розташування (Зал, Тераса, VIP, Бар)                                  |

Таблиця 2.1.3.5 – Структура таблиці Employees (Співробітники)

| Поле         | Тип даних | Обмеження                  | Опис                                                |
|--------------|-----------|----------------------------|-----------------------------------------------------|
| Id           | INTEGER   | PRIMARY KEY, AUTOINCREMENT | Унікальний ідентифікатор співробітника              |
| FullName     | TEXT      | NOT NULL                   | ПІБ                                                 |
| Role         | INTEGER   | NOT NULL                   | Роль (0 – Admin, 1 – Manager, 2 – Waiter, 3 – Cook) |
| Username     | TEXT      | NOT NULL, UNIQUE           | Логін для входу                                     |
| PasswordHash | TEXT      | NOT NULL                   | Хеш пароля (SHA256)                                 |
| PhoneNumber  | TEXT      |                            | Номер телефону                                      |
| Email        | TEXT      |                            | Електронна пошта                                    |
| HireDate     | DATETIME  | NOT NULL                   | Дата прийняття на роботу                            |
| IsActive     | INTEGER   | NOT NULL, DEFAULT 1        | Активний (1 – так, 0 – ні)                          |

Таблиця 2.1.3.6 – Структура таблиці Orders (Замовлення)

| Поле               | Тип даних    | Обмеження                                               | Опис                                                                                  |
|--------------------|--------------|---------------------------------------------------------|---------------------------------------------------------------------------------------|
| Id                 | INTEGER      | PRIMARY KEY, AUTOINCREMENT                              | Унікальний ідентифікатор замовлення                                                   |
| TableId            | INTEGER      | FOREIGN KEY REFERENCES Tables(Id) ON DELETE RESTRICT    | Ідентифікатор столу                                                                   |
| EmployeeId         | INTEGER      | FOREIGN KEY REFERENCES Employees(Id) ON DELETE RESTRICT | Ідентифікатор співробітника                                                           |
| OrderTime          | DATETIME     | NOT NULL                                                | Дата та час створення                                                                 |
| Status             | INTEGER      | NOT NULL, DEFAULT 0                                     | Статус (0 – нове, 1 – готується, 2 – готове, 3 – подано, 4 – оплачено, 5 – скасовано) |
| DiscountPercentage | DECIMAL(5,2) | NOT NULL, DEFAULT 0                                     | Відсоток знижки                                                                       |

|             |               |          |                         |
|-------------|---------------|----------|-------------------------|
| Subtotal    | DECIMAL(18,2) | NOT NULL | Сума без податку        |
| Tax         | DECIMAL(18,2) | NOT NULL | Сума податку            |
| TotalAmount | DECIMAL(18,2) | NOT NULL | Загальна сума до сплати |
| Notes       | TEXT          |          | Додаткові примітки      |

**Таблиця 2.1.3.7 – Структура таблиці OrderItems (Позиції замовлення)**

| Поле                | Тип даних     | Обмеження                                               | Опис                             |
|---------------------|---------------|---------------------------------------------------------|----------------------------------|
| Id                  | INTEGER       | PRIMARY KEY, AUTOINCREMENT                              | Унікальний ідентифікатор позиції |
| OrderId             | INTEGER       | FOREIGN KEY REFERENCES Orders(Id) ON DELETE CASCADE     | Ідентифікатор замовлення         |
| MenuItemId          | INTEGER       | FOREIGN KEY REFERENCES MenuItems(Id) ON DELETE RESTRICT | Ідентифікатор страви             |
| Quantity            | INTEGER       | NOT NULL, DEFAULT 1                                     | Кількість                        |
| UnitPrice           | DECIMAL(18,2) | NOT NULL                                                | Ціна на момент замовлення        |
| SpecialInstructions | TEXT          |                                                         | Особливі побажання клієнта       |

**Таблиця 2.1.3.8 – Структура таблиці Reservations (Бронювання)**

| Поле            | Тип даних | Обмеження                                            | Опис                                                                                     |
|-----------------|-----------|------------------------------------------------------|------------------------------------------------------------------------------------------|
| Id              | INTEGER   | PRIMARY KEY, AUTOINCREMENT                           | Унікальний ідентифікатор бронювання                                                      |
| TableId         | INTEGER   | FOREIGN KEY REFERENCES Tables(Id) ON DELETE RESTRICT | Ідентифікатор столу                                                                      |
| CustomerName    | TEXT      | NOT NULL                                             | Ім'я клієнта                                                                             |
| CustomerPhone   | TEXT      |                                                      | Контактний телефон                                                                       |
| CustomerEmail   | TEXT      |                                                      | Електронна пошта                                                                         |
| ReservationTime | DATETIME  | NOT NULL                                             | Дата та час бронювання                                                                   |
| NumberOfGuests  | INTEGER   | NOT NULL                                             | Кількість гостей                                                                         |
| SpecialRequests | TEXT      |                                                      | Особливі побажання                                                                       |
| Status          | INTEGER   | NOT NULL, DEFAULT 0                                  | Статус (0 – підтверджено, 1 – розміщені, 2 – завершено, 3 – скасовано, 4 – не з'явилися) |

**Таблиця 2.1.3.9 – Структура таблиці SalesLogs (Логи продажів)**

| Поле            | Тип даних     | Обмеження                                               | Опис                                                                              |
|-----------------|---------------|---------------------------------------------------------|-----------------------------------------------------------------------------------|
| Id              | INTEGER       | PRIMARY KEY, AUTOINCREMENT                              | Унікальний ідентифікатор логу                                                     |
| OrderId         | INTEGER       | FOREIGN KEY REFERENCES Orders(Id) ON DELETE CASCADE     | Ідентифікатор замовлення                                                          |
| EmployeeId      | INTEGER       | FOREIGN KEY REFERENCES Employees(Id) ON DELETE RESTRICT | Ідентифікатор співробітника                                                       |
| TransactionDate | DATETIME      | NOT NULL                                                | Дата та час транзакції                                                            |
| SaleAmount      | DECIMAL(18,2) | NOT NULL                                                | Сума продажу                                                                      |
| TaxAmount       | DECIMAL(18,2) | NOT NULL                                                | Сума податку                                                                      |
| DiscountAmount  | DECIMAL(18,2) | NOT NULL                                                | Сума знижки                                                                       |
| NetAmount       | DECIMAL(18,2) | NOT NULL                                                | Чиста сума                                                                        |
| PaymentMethod   | INTEGER       | NOT NULL                                                | Спосіб оплати (0 – готівка, 1 – картка, 2 – онлайн, 3 – криптовалюта, 4 – ваучер) |
| ItemsCount      | INTEGER       | NOT NULL                                                | Кількість проданих одиниць                                                        |
| CostOfGoodsSold | DECIMAL(18,2) | NOT NULL                                                | Собівартість проданих товарів                                                     |
| Category        | TEXT          |                                                         | Категорія продажу                                                                 |
| IsRefunded      | INTEGER       | NOT NULL, DEFAULT 0                                     | Чи було повернення                                                                |

**Таблиця 2.1.3.10 – Структура таблиць бази даних (зведена)**

| Таблиця           | Первинний ключ | Зовнішні ключі              | Призначення                              |
|-------------------|----------------|-----------------------------|------------------------------------------|
| MenuItems         | Id             | –                           | Зберігає інформацію про страви           |
| InventoryItems    | Id             | –                           | Зберігає інформацію про інгредієнти      |
| RecipeIngredients | Id             | MenuItemId, InventoryItemId | Зв'язує страви з інгредієнтами (рецепти) |
| Tables            | Id             | –                           | Зберігає інформацію про столи            |
| Employees         | Id             | –                           | Зберігає інформацію про співробітників   |
| Orders            | Id             | TableId, EmployeeId         | Зберігає інформацію про замовлення       |
| OrderItems        | Id             | OrderId, MenuItemId         | Зберігає позиції замовлень               |
| Reservations      | Id             | TableId                     | Зберігає інформацію про бронювання       |
| SalesLogs         | Id             | OrderId, EmployeeId         | Зберігає дані для фінансової звітності   |

На рисунку 2.1.2 наведено повну схему бази даних, яка візуалізує всі таблиці та зв'язки між ними.

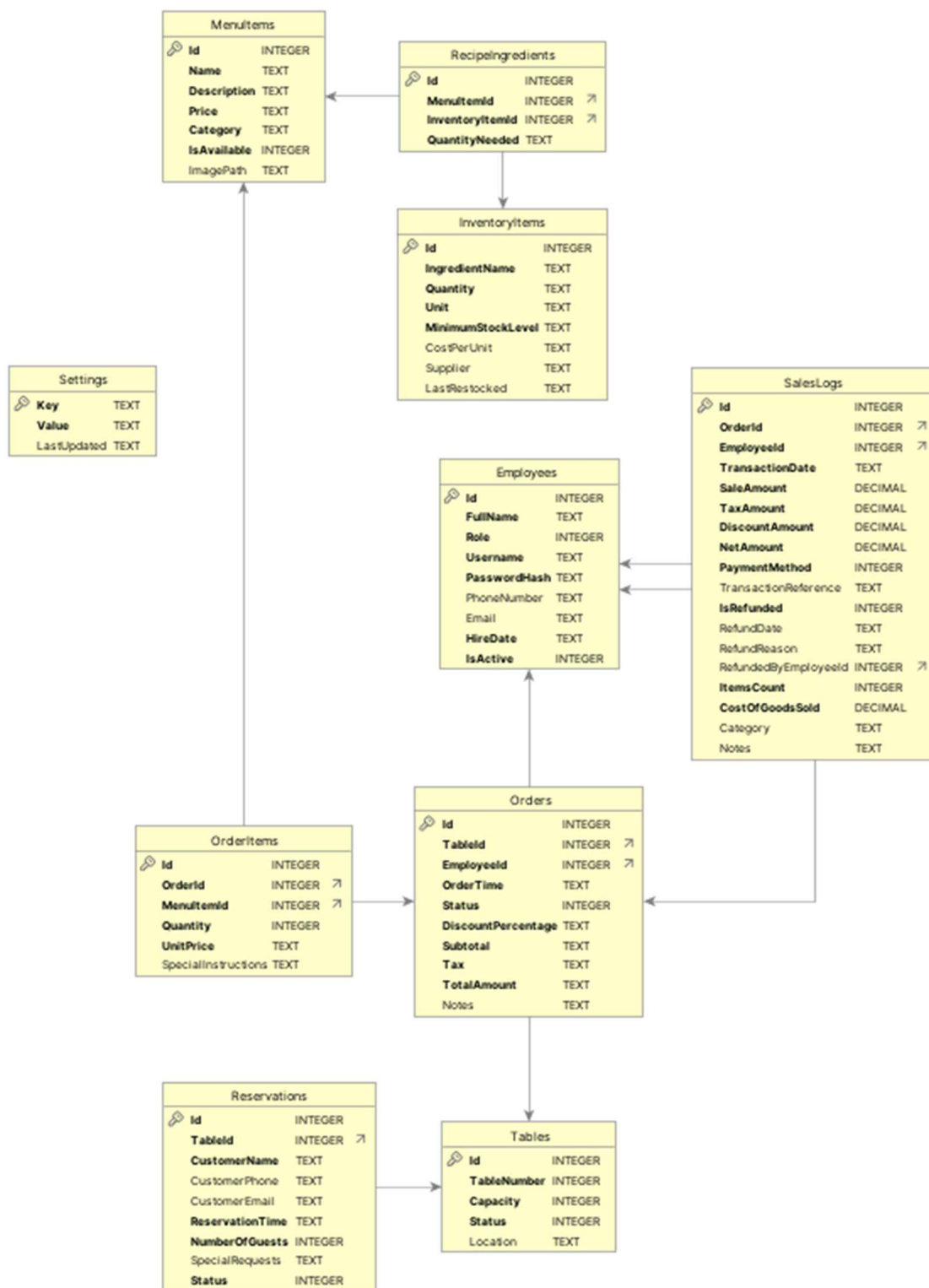


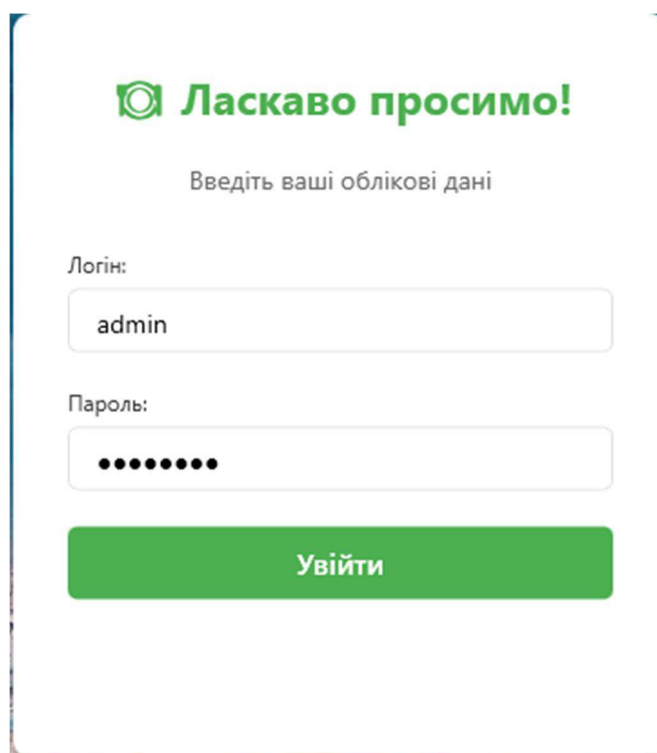
Рисунок 2.1.2. – Схема бази даних

Наведена фізична модель повністю відповідає логічній ER-моделі, забезпечує цілісність даних та є основою для реалізації програмного забезпечення.

## 2.2 Список основних форм та їх призначення

Розроблена інформаційна система автоматизації роботи закладу громадського харчування налічує вісім основних форм, кожна з яких реалізує конкретний функціональний модуль і забезпечує виконання визначених бізнес-процесів.

Першою формою, з якою взаємодіє користувач, є форма авторизації. Вона призначена для ідентифікації та аутентифікації співробітника закладу шляхом введення логіну та пароля. Формат введення логіну – текстовий рядок без обмежень, поле пароля приховує символи для забезпечення конфіденційності. Після успішної перевірки облікових даних система визначає роль користувача (адміністратор, менеджер, офіціант або кухар) та надає доступ до відповідних функціональних можливостей. На рис. 2.2.1 зображено форму авторизації.



Ласкаво просимо!

Введіть ваші облікові дані

Логін:

admin

Пароль:

••••••••

Увійти

Рисунок 2.2.1. – Форма авторизації

Головне вікно програми містить бокове меню навігації, яке забезпечує перемикання між основними модулями, та область відображення вмісту вибраного модуля. Залежно від ролі користувача, деякі пункти меню можуть бути приховані або недоступні для виконання. Це вікно є контейнером для всіх інших форм-модулів.

The screenshot shows a web application window titled "Інформаційна система автоматизації роботи закладу громадського харчування". On the left is a green sidebar for the "Адміністратор" (Administrator) role, with menu items: "Замовлення" (Orders), "Столики" (Tables), "Меню" (Menu), "Склад" (Inventory), "Звіти" (Reports), "Налаштування" (Settings), and "Вихід" (Exit). The main area is titled "Активні замовлення" (Active orders). On the right is the "Нове замовлення" (New order) form. It includes a "Стіл:" (Table) dropdown menu, a warning "Виберіть стіл" (Select a table), a "Додати страву:" (Add dish) section with a dropdown, a quantity input (1), and an add button (+). Below is a list of "Страви в замовленні:" (Dishes in order) with a large empty box. At the bottom right, it shows "Загалом: 20.00" (Total: 20.00) and three buttons: "+ Нове" (Add new), "Скасувати" (Cancel), and "Оформити" (Confirm).

Рисунок 2.2.2. – Форма «Замовлення»

Форма «Замовлення» є центральним модулем для роботи офіціантів та менеджерів. Вона призначена для створення нових замовлень та управління вже існуючими. Ліва частина форми відображає список активних замовлень із зазначенням номера столу, часу створення, загальної суми та поточного статусу. Права частина містить поля для вибору столу зі списку доступних, вибору страв з меню (з можливістю пошуку за назвою або категорією), введення кількості порцій у числовому форматі (лише цілі додатні значення) та спеціальних інструкцій у текстовому вигляді. Для кожної страви відображається її ціна, а загальна сума замовлення розраховується автоматично з урахуванням податку та можливої знижки. Форма дозволяє

змінювати статус замовлення за допомогою вбудованого перемикача, а після завершення обслуговування – фіксувати оплату.

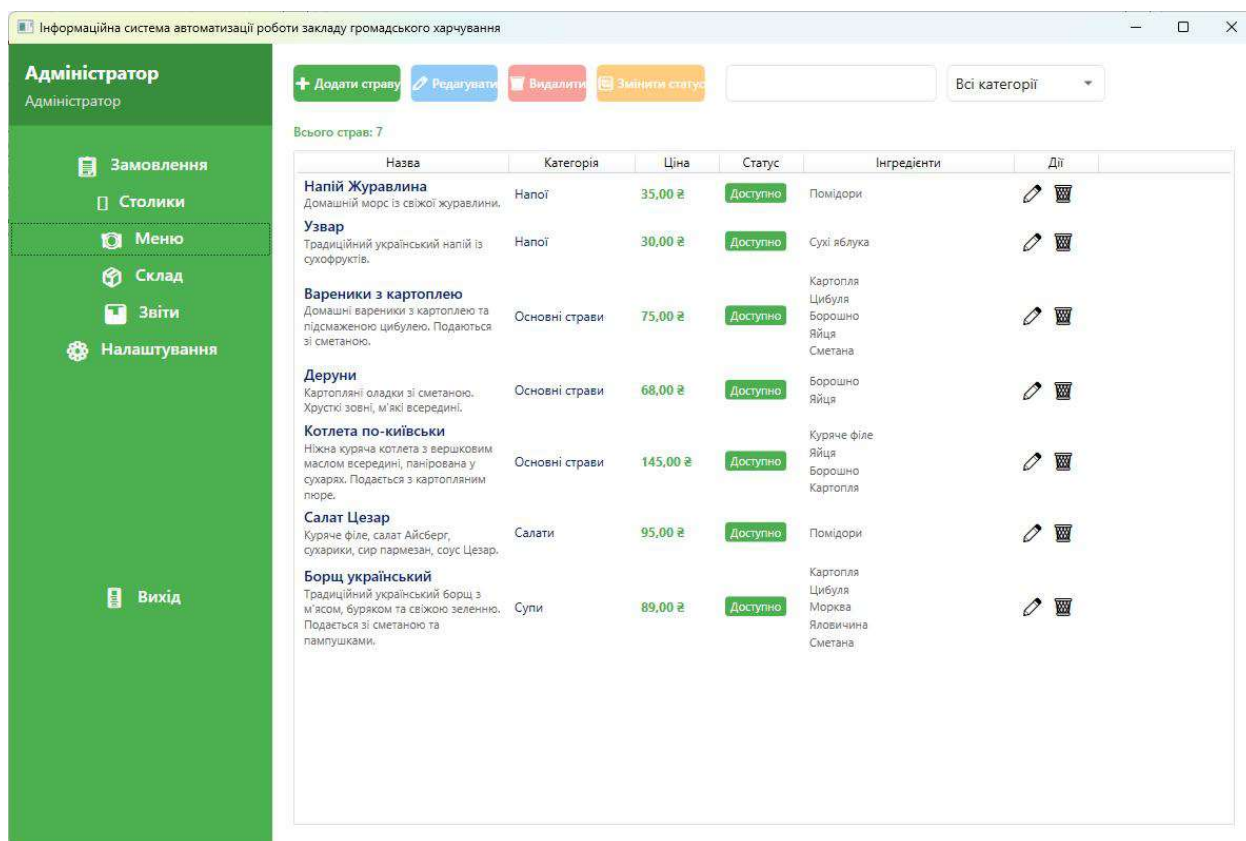


Рисунок 2.2.3. – Модуль «Меню»

Модуль «Меню» призначений для адміністраторів та менеджерів з метою управління складом страв та напоїв. Він відображає список усіх позицій меню у вигляді таблиці з колонками: назва, категорія, ціна, статус доступності та перелік інгредієнтів. Через контекстне меню або кнопки керування можна додати нову страву, відредагувати наявну або видалити її. При додаванні або редагуванні відкривається діалогове вікно, в якому потрібно ввести назву (текстовий рядок, обов'язкове поле), опис (текст, необов'язково), категорію (вибір зі списку або введення нової), ціну (додатне число з двома десятковими знаками) та позначити, чи доступна страва наразі. Крім того, в цьому ж діалозі можна додавати інгредієнти до рецепту страви, вибираючи їх зі списку наявних на складі та вказуючи необхідну кількість у числовому форматі з відповідною одиницею виміру.

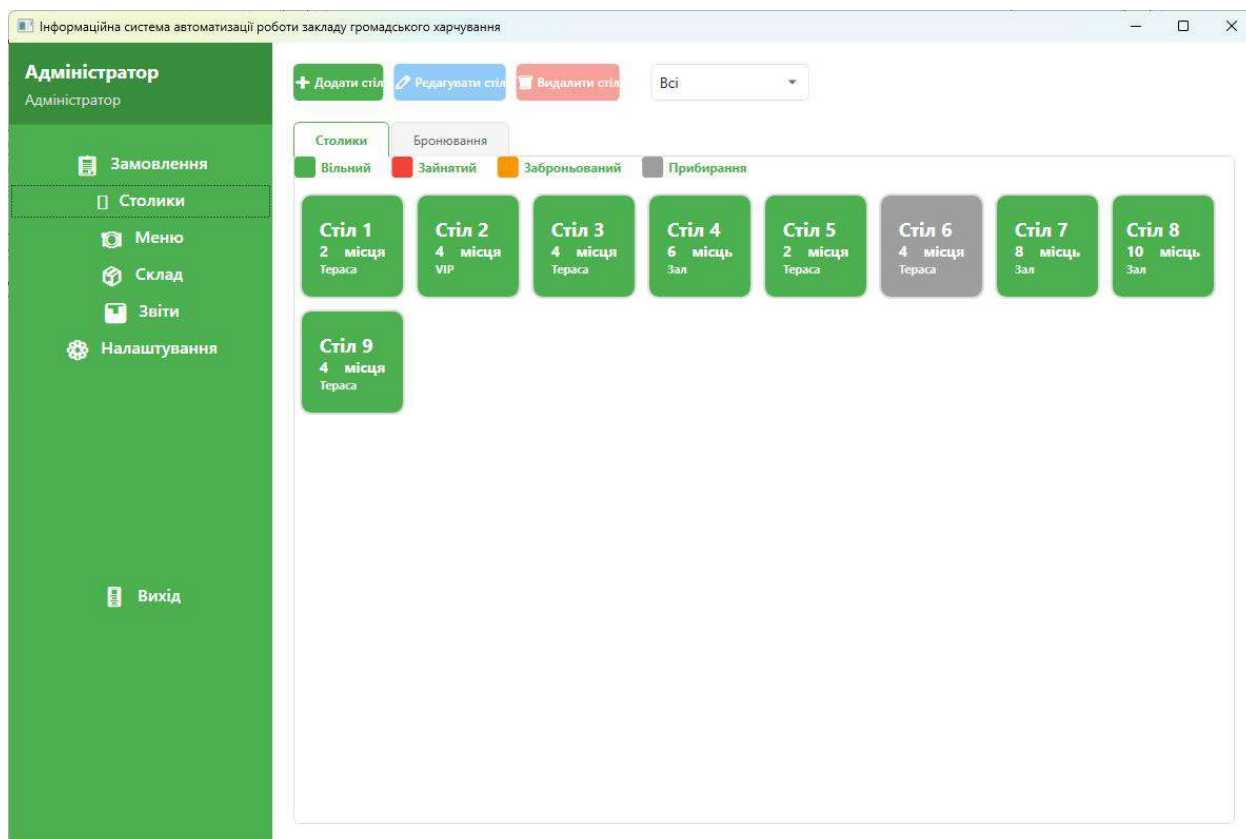


Рисунок 2.2.4. – Форма «Столики»

Форма «Столики» призначена для візуального контролю статусу всіх столів закладу. Вона представлена у вигляді сітки кольорових кнопок, де кожна кнопка відповідає одному столу. Колір кнопки інформує про поточний статус: зелений – вільний, червоний – зайнятий, помаранчевий – заброньований, сірий – прибирання. При натисканні на стіл правою кнопкою миші відкривається контекстне меню для зміни статусу або видалення столу. Окрема вкладка цього модуля призначена для управління резервуваннями. Тут можна вибрати дату за допомогою календаря, після чого система відображає список бронювань на обраний день. Додавання нового бронювання потребує введення імені клієнта (обов'язковий текст), номера телефону та електронної пошти (необов'язкові поля), вибору столу зі списку, встановлення дати та часу (через календар і поле часу) та кількості гостей (ціле додатне число). Також можна додати текстові особливі побажання.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
|      |      |          |        |      |                           | 30   |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |      |

Інформаційна система автоматизації роботи закладу громадського харчування

Адміністратор

Адміністратор

Замовлення

Столики

Меню

Склад

Звіти

Налаштування

Вихід

Всього позицій

15

Загальна вартість

€18,211.60

Низький запас

2

Експорт звіту

+ Додати інгредієнт

Редагувати

Видалити

Поповнити

Коригувати

Всі

Тільки

| Назва                   | Кількість | Мін. рівень | Ціна/од | Загальна вартість | Постачальник           | Останнє поповнення |
|-------------------------|-----------|-------------|---------|-------------------|------------------------|--------------------|
| Борошно                 | 22.80     | 5.00        | €12.00  | €273.60           | Млин                   |                    |
| Млин                    | кг        |             |         |                   |                        |                    |
| Гриби печериці          | 5.00      | 2.00        | €55.00  | €275.00           | Грибна ферма           |                    |
| Грибна ферма            | кг        |             |         |                   |                        |                    |
| Картопля                | 47.75     | 10.00       | €15.00  | €716.25           | Фермерське господарств |                    |
| Фермерське господарство | кг        |             |         |                   |                        |                    |
| Кураче філе             | 23.00     | 5.00        | €95.00  | €2,185.00         | Птахофабрика           |                    |
| Птахофабрика            | кг        |             |         |                   |                        |                    |
| Майонез                 | 8.00      | 2.00        | €38.00  | €304.00           | Харчовий комбінат      |                    |
| Харчовий комбінат       | л         |             |         |                   |                        |                    |
| Молоко                  | 15.00     | 5.00        | €28.00  | €420.00           | Молочна ферма          |                    |
| Молочна ферма           | л         |             |         |                   |                        |                    |
| Морква                  | 14.80     | 3.00        | €10.00  | €148.00           | Овочева база           |                    |
| Овочева база            | кг        |             |         |                   |                        |                    |
| Огірки                  | 10.00     | 3.00        | €32.00  | €320.00           | Овочева база           |                    |
| Овочева база            | кг        |             |         |                   |                        |                    |
| Помідори                | 1.00      | 3.00        | €45.00  | €45.00            | Овочева база           |                    |
| Овочева база            | кг        |             |         |                   |                        |                    |
| Сир твердий             | 8.00      | 2.00        | €210.00 | €1,680.00         | Сиророб                |                    |
| Сиророб                 | кг        |             |         |                   |                        |                    |
| Сметана                 | 9.85      | 3.00        | €45.00  | €443.25           | Молочна ферма          |                    |
| Молочна ферма           | л         |             |         |                   |                        |                    |
| Сухі яблука             | 41.00     | 50.00       | €127.00 | €5,207.00         | Петро                  | 21.02.2026         |
| Петро                   | кг        |             |         |                   |                        |                    |

Увага! Низький запас інгредієнтів:

Помідори - залишилось 1.00 кг

Сухі яблука - залишилось 41.00 кг

Рисунок 2.2.5. – Модуль «Склад»

Модуль «Склад» реалізує облік інгредієнтів, необхідних для приготування страв. Головна таблиця містить перелік усіх інгредієнтів з полями: назва, поточна кількість, одиниця виміру, мінімальний рівень, ціна за одиницю, постачальник та статус. Рядки з низьким рівнем запасів підсвічуються червоним кольором. Для додавання нового інгредієнта необхідно ввести назву (текст), початкову кількість (додатне число), вибрати одиницю виміру зі списку (кг, л, шт, г, мл) або ввести власну, вказати мінімальний рівень (додатне число) та ціну за одиницю. При редагуванні інгредієнта можна скоригувати кількість або змінити інші параметри. Функції поповнення та коригування запасу викликають діалогові вікна для введення зміни кількості (додатне або від’ємне число) та текстового пояснення причини.

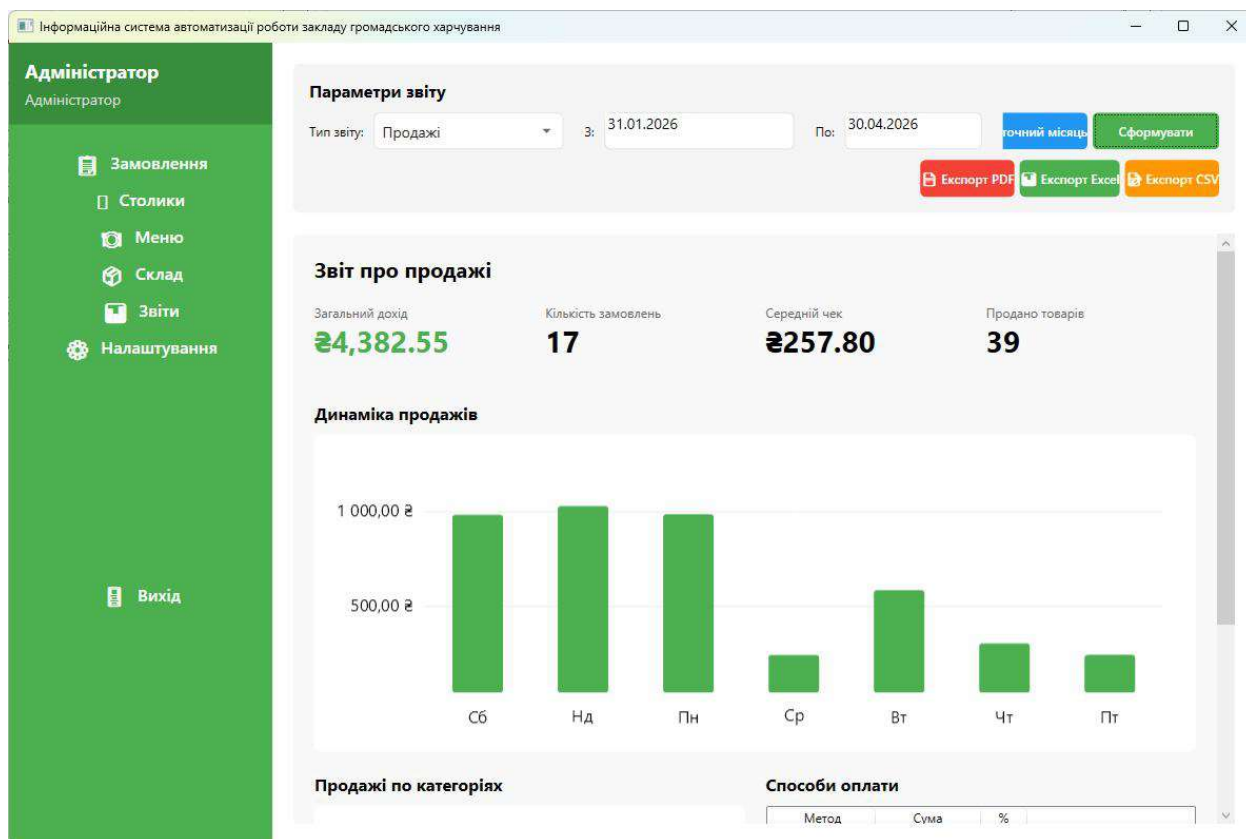


Рисунок 2.2.6. – Форма «Звіти»

Форма «Звіти» генерує аналітичну інформацію на основі накопичених даних про продажі, замовлення та роботу співробітників. Користувач обирає тип звіту з випадаючого списку: «Продажі», «Популярні страви», «Фінанси», «Склад» або «Співробітники». Потім встановлює період аналізу за допомогою двох календарів (початкова та кінцева дата), після чого система формує звіт у вигляді таблиць, числових показників та діаграм. Наприклад, звіт про продажі містить загальний дохід, кількість замовлень, середній чек та графік продажів по днях. Звіт про фінанси розраховує валовий та чистий прибуток, маржинальність та сукупні витрати. Усі звіти можна експортувати у формат CSV для подальшого аналізу в інших програмах.

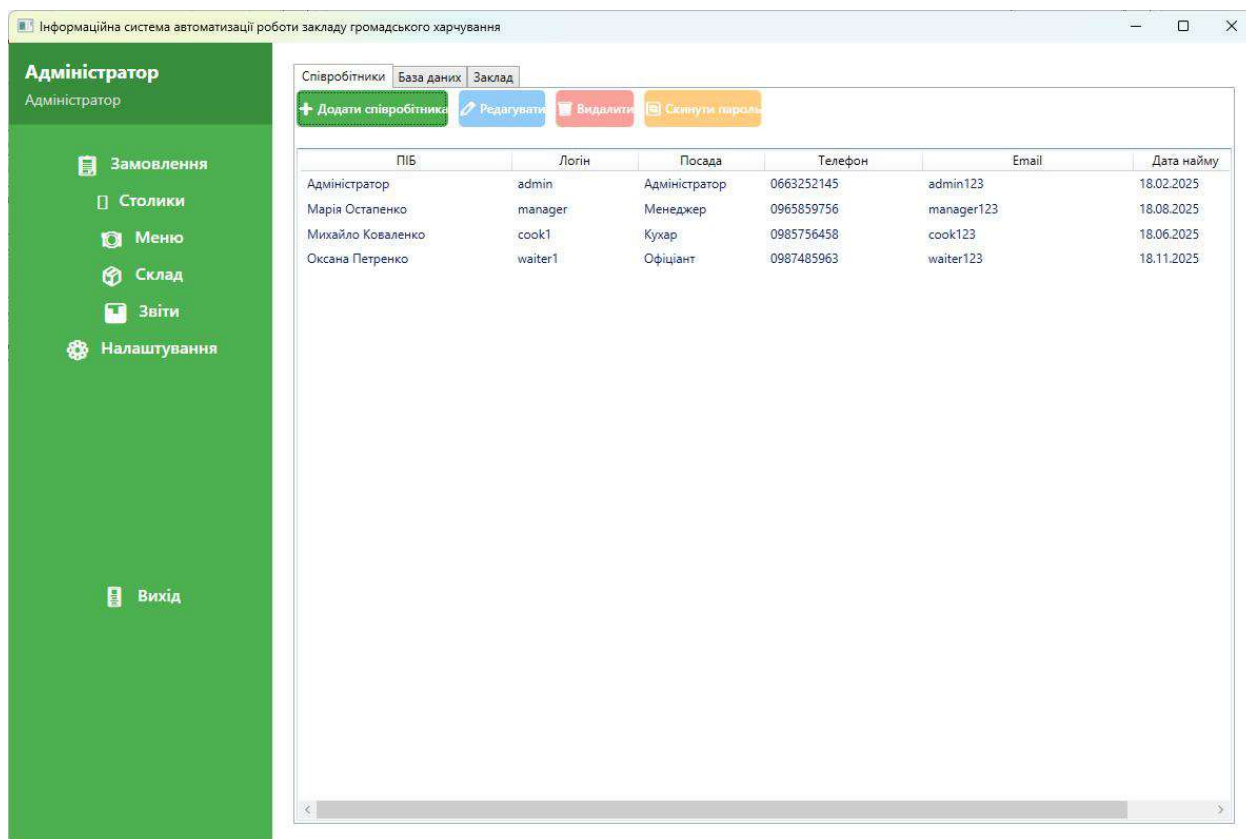


Рисунок 2.2.7. – Модуль «Налаштування»

Модуль «Налаштування» доступний виключно адміністраторам та менеджерам. Він складається з трьох вкладок: перша призначена для управління співробітниками (створення облікових записів, зміна ролей, скидання паролів, активація або деактивація), друга – для роботи з базою даних (створення резервної копії, відновлення з резервної копії, оптимізація бази даних, перегляд інформації про поточний стан), третя – для заповнення реквізитів самого закладу (назва, адреса, телефон, електронна пошта, податковий номер). Усі текстові поля у вкладці «Заклад» є редагованими, а внесені зміни зберігаються в окремій таблиці бази даних.

Кожна форма розроблена з урахуванням принципів зручності та інтуїтивної зрозумілості інтерфейсу, всі текстові написи виконано українською мовою, а формати введення даних контролюються як на стороні клієнта (перевірка безпосередньо перед відправкою), так і на рівні бази даних (обмеження цілісності). Такий підхід забезпечує мінімізацію помилок персоналу та підвищує загальну ефективність роботи закладу.

## 2.3 Архітектура програмного засобу

Розроблена інформаційна система ґрунтується на сучасних принципах проєктування програмного забезпечення, що забезпечують її надійність, масштабованість та зручність супроводження. Основним архітектурним патерном обрано **Model-View-ViewModel (MVVM)**, який є стандартом для побудови клієнтських додатків на платформі .NET з використанням Windows Presentation Foundation (WPF). Цей патерн забезпечує чітке розділення відповідальності між різними компонентами системи, що спрощує розробку, тестування та подальшу модернізацію.

**Модель (Model)** є найнижчим рівнем архітектури та відповідає за зберігання та управління даними. У розробленій системі модель представлена класами-сутностями, які відображають структуру бази даних (MenuItem, Order, Table, InventoryItem, Employee, Reservation тощо), а також контекстом бази даних (RestaurantDbContext), що забезпечує зв'язок між об'єктами та реляційною базою даних SQLite через технологію Entity Framework Core. Модель не містить жодної логіки відображення або взаємодії з користувачем, вона лише надає дані та методи для їх отримання, збереження або видалення.

**Подання (View)** відповідає за візуальне відображення інтерфейсу користувача та обробку вхідних подій (кліки миші, натискання клавіш, введення тексту тощо). У системі подання реалізовані у вигляді XAML-файлів, кожен з яких описує розташування елементів керування (кнопок, текстових полів, списків, таблиць) та їхній зовнішній вигляд. Наприклад, форма авторизації містить поля для введення логіну та пароля, а форма роботи із замовленнями – комбіновані списки для вибору страв та таблиці для відображення активних замовлень. Подання не містить бізнес-логіки, воно лише передає дії користувача до ViewModel за допомогою команд та механізму прив'язки даних (data binding).

**Модель подання (ViewModel)** є сполучною ланкою між моделлю та поданням. Вона містить усю бізнес-логіку програми: перевірку коректності введених даних, обчислення вартості замовлення, розрахунок знижок,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 34   |

контроль наявності інгредієнтів на складі, формування звітів. Кожна форма має відповідну ViewModel (наприклад, LoginViewModel, OrdersViewModel, MenuViewModel, TablesViewModel), яка реалізує інтерфейс INotifyPropertyChanged для автоматичного сповіщення подання про зміну стану. ViewModel отримує необхідні залежності (сервіси для роботи з базою даних, сервіси логування, сервіси обробки повідомлень) через механізм Dependency Injection, що забезпечує слабе зв'язування компонентів та полегшує тестування.

На рисунку 2.3.1 наведено діаграму компонентів розробленої інформаційної системи.

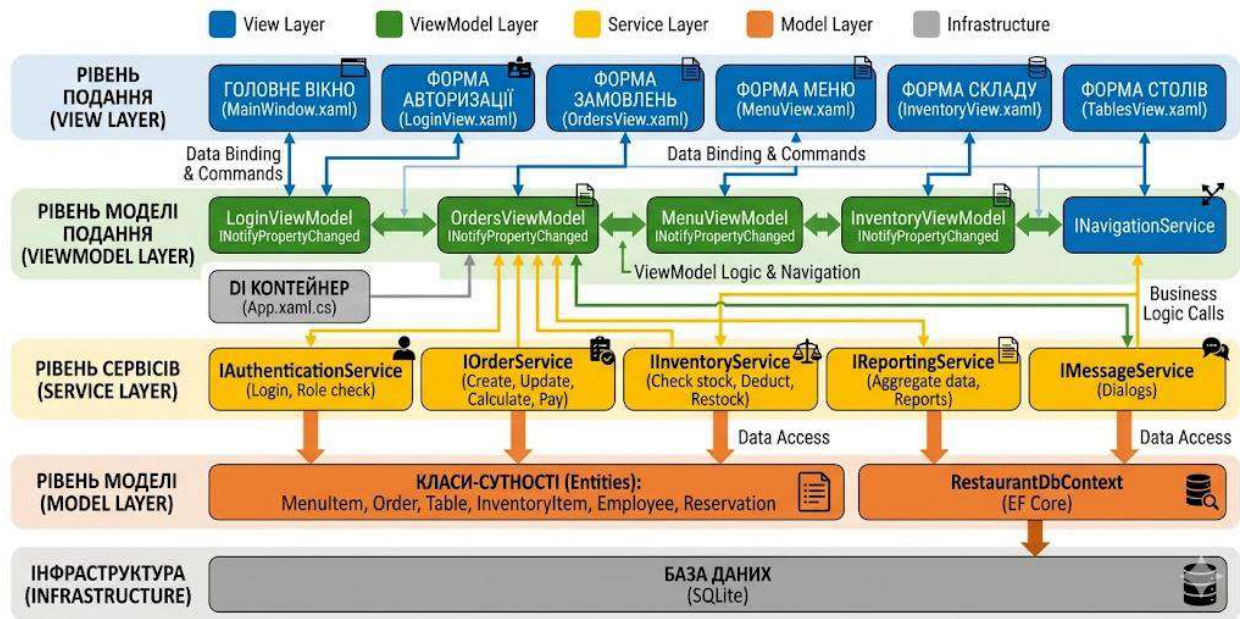


Рисунок 2.3.1. – Діаграма компонентів інформаційної системи

Окрім патерну MVVM, система використовує архітектурний патерн «Сервіс» (Service). Усі операції, які виходять за межі простої взаємодії з базою даних, винесено в окремі сервісні класи. До них належать:

- IAuthenticationService – сервіс автентифікації, що перевіряє логін та пароль користувача, зберігає інформацію про поточного користувача та надає методи для перевірки його ролі.
- IOrderService – сервіс управління замовленнями, який реалізує створення нових замовлень, оновлення їхнього статусу, додавання позицій до замовлення, розрахунок вартості та обробку платежів.

- `IInventoryService` – сервіс складського обліку, що забезпечує перевірку наявності інгредієнтів для приготування страви, автоматичне списання інгредієнтів при створенні замовлення, поповнення запасів та отримання списку позицій з низьким рівнем.
- `IReportingService` – сервіс формування звітів, який агрегує дані з різних таблиць бази даних (замовлення, продажі, склад, співробітники) та повертає структуровані об’єкти-звіти, придатні для відображення у вигляді таблиць та діаграм.
- `IMessageService` – сервіс візуального сповіщення, який інкапсулює виклики діалогових вікон (інформаційних, попереджувальних, помилок, підтвердження). Це дозволяє легко змінити тип сповіщень (наприклад, замінити стандартне `MessageBox` на кастомне вікно) без внесення змін до `ViewModel`.

Взаємодія між компонентами відбувається через механізм **Dependency Injection (DI)**. Контейнер DI, налаштований у файлі `App.xaml.cs`, відповідає за створення екземплярів усіх необхідних сервісів та `ViewModel`. Кожна `ViewModel` отримує залежності через свій конструктор, що робить код прозорим та легко тестованим. Наприклад, конструктор `OrdersViewModel` приймає інтерфейси `IOrderService`, `IInventoryService`, `IMessageService` та `IAuthenticationService`, які підставляються автоматично DI-контейнером.

На рисунку 2.3.2 зображено схему взаємодії компонентів відповідно до патерну MVVM.

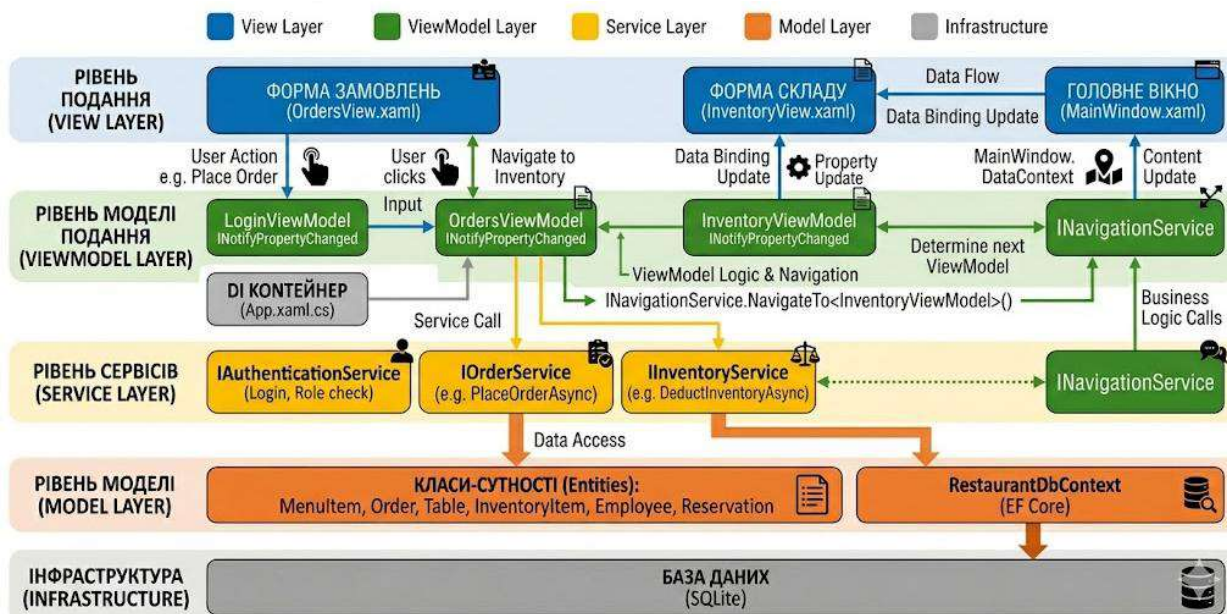


Рисунок 2.3.2. – Схема взаємодії компонентів у патерні MVVM

Управління навігацією між різними поданнями реалізовано через окремий сервіс `INavigationService`. Його відповідальність полягає у зміні поточного вмісту головного вікна програми. Коли користувач обирає пункт меню (наприклад, «Склад»), `ViewModel` викликає метод `NavigateTo<InventoryViewModel>`, сервіс створює або отримує з кешу відповідну `ViewModel`, після чого головне вікно оновлює область відображення, застосовуючи визначений у ресурсах програми `DataTemplate`, який автоматично зв'язує `ViewModel` з відповідним поданням.

Такий підхід забезпечує високу модульність системи, дозволяє легко додавати нові функціональні модулі без зміни існуючого коду, спрощує виявлення та усунення помилок (завдяки чіткому розподілу відповідальності) та робить систему готовою до можливого розширення в майбутньому (наприклад, до впровадження веб-інтерфейсу або мобільного додатку).

## 2.4 Програмна реалізація

### 2.4.1 Загальна структура проєкту

Програмна реалізація інформаційної системи виконана у вигляді єдиного проєкту типу Windows Presentation Foundation (WPF) на мові C# з використанням платформи .NET 9. Проєкт організовано за принципом розділення відповідальності відповідно до архітектурного патерну MVVM, що знайшло своє відображення у логічній структурі каталогів. Коренева тека проєкту містить файли App.xaml (точка входу програми, налаштування ресурсів та стилів), App.xaml.cs (конфігурація Dependency Injection, ініціалізація бази даних) та AssemblyInfo.cs. Каталог Models вміщує класи-сутності, що описують структуру бази даних. Каталог Data містить контекст бази даних RestaurantDbContext, успадкований від DbContext Entity Framework Core, та клас DatabaseInitializer для початкового наповнення бази даних тестовими даними. Каталог ViewModels реалізує всі моделі подання, кожна з яких успадкована від базового класу ViewModelBase, що реалізує інтерфейс INotifyPropertyChanged. Каталог Views містить XAML-файли опису інтерфейсу користувача та відповідні код-бекайнди (xaml.cs) для обробки подій, що не можуть бути реалізовані через механізм команд. Каталог Services вміщує реалізації сервісів (автентифікація, навігація, робота із замовленнями, складом, звітами, повідомленнями) та їхні інтерфейси, що забезпечує слабе зв'язування компонентів. Каталог Converters містить класи-конвертери для перетворення типів даних у прив'язках (наприклад, перетворення статусу замовлення у кольорове тло). Каталог Helpers включає допоміжні класи для валідації даних та логування подій за допомогою бібліотеки Serilog.

### 2.4.2 Реалізація бази даних та доступу до даних

Взаємодія програми з базою даних реалізована за допомогою об'єктно-реляційного мапера Entity Framework Core. Контекст бази даних RestaurantDbContext містить набори сутностей (DbSet<T>) для кожної таблиці, що дозволяє виконувати операції створення, читання, оновлення та видалення

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 38   |

даних у вигляді LINQ-запитів без написання SQL-коду. Налаштування зв'язків між сутностями, обмежень та каскадних дій виконано у методі OnModelCreating із застосуванням Fluent API. Файл бази даних RestaurantDB.sqlite створюється автоматично в підкаталозі Base поруч із виконуваним файлом програми при першому запуску завдяки виклику методу EnsureCreatedAsync. Клас DatabaseInitializer містить статичний метод SeedDataAsync, який перевіряє наявність записів у таблицях, і за їх відсутності наповнює базу даних початковими даними: співробітниками з різними ролями (адміністратор, менеджер, офіціант, кухар), прикладами столів, інгредієнтами складу, позиціями меню та рецептами, а також тестовими замовленнями для демонстрації роботи модуля звітності. Вилучення даних з бази виконується асинхронно з використанням методів ToListAsync, FirstOrDefaultAsync та AnyAsync, що дозволяє не блокувати потік інтерфейсу користувача під час виконання довгих запитів.

### 2.4.3 Реалізація бізнес-логіки у ViewModel

Кожна ViewModel містить набір властивостей, анотованих атрибутом [ObservableProperty] з бібліотеки CommunityToolkit.Mvvm, що автоматично генерує код сповіщення про зміну значень. Команди, які реагують на дії користувача, реалізовано за допомогою атрибутів [RelayCommand], що також генерує необхідну обгортку. Наприклад, у LoginViewModel наявна команда LoginCommand, яка викликає асинхронний метод LoginAsync, що отримує збережені в полях значення логіну та пароля, викликає метод AuthenticateAsync сервісу IAuthenticationService та в разі успіху відкриває головне вікно програми. У OrdersViewModel реалізовано команди AddItemCommand, SubmitOrderCommand, UpdateOrderCommand, кожна з яких має власний метод CanExecute для динамічного керування доступністю кнопок залежно від стану форми. Наприклад, кнопка додавання страви стає активною лише після того, як користувач вибрав страву зі списку та обрав стіл або замовлення для редагування. У InventoryViewModel реалізовано команди RestockCommand та AdjustStockCommand, які викликають діалогові вікна для

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 39   |

введення коригувань кількості інгредієнта та зберігають зміни в базі даних після підтвердження.

#### 2.4.4 Реалізація сервісів

Сервіси є ключовими компонентами, що містять основну бізнес-логіку системи. Клас `OrderService` реалізує інтерфейс `IOrderService` та виконує операції створення та оновлення замовлень. Його метод `CreateOrderAsync` додає нове замовлення до контексту, змінює статус відповідного столу на «Зайнятий» та зберігає зміни. Метод `AddItemToOrderAsync` перевіряє наявність необхідної кількості інгредієнтів через виклик `CheckIngredientAvailabilityAsync` сервісу `IInventoryService`, після чого додає позицію до замовлення. Метод `ProcessPaymentAsync` змінює статус замовлення на «Оплачено», звільняє стіл та створює запис у логах продажів (`SalesLog`). Клас `InventoryService` містить метод `ConsumeIngredientsAsync`, який проходить за рецептом обраної страви та зменшує відповідні інгредієнти на складі на кількість, кратну замовленим порціям. Якщо після списання кількість інгредієнта стає меншою або дорівнює мінімальному рівню, сервіс ініціює показ попереджувального повідомлення через `IMessageService`. Клас `ReportingService` агрегує дані з таблиць `SalesLogs`, `Orders`, `OrderItems` та `InventoryItems` для формування звітних об'єктів. Для підвищення продуктивності складні запити виконуються з використанням методів `GroupBy`, `Sum`, `Average` та `Count`, а результати повертаються у вигляді спроектованих об'єктів, які містять лише необхідну для відображення інформацію.

#### 2.4.5 Реалізація інтерфейсу користувача

Інтерфейс користувача реалізовано мовою XAML, що дозволяє декларативно описувати структуру вікон та елементів керування. Головне вікно `MainWindow` має сітку з двох колонок: ліва колонка містить панель навігації у вигляді набору кнопок, права – контентний елемент `ContentControl`, який відображає поточне подання залежно від значення властивості

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 40   |

CurrentView у MainViewModel. Для автоматичного зв'язування типів ViewModel з відповідними поданнями у ресурсах App.xaml визначено набір шаблонів DataTemplate, який зіставляє, наприклад, OrdersViewModel з OrdersView. Такий підхід унеможлиблює необхідність написати додатковий код для створення екземплярів подань. Візуальне оформлення всіх елементів керування (кнопок, текстових полів, списків) винесено у стилі, збережені в ресурсах застосунку, що забезпечує однаковий зовнішній вигляд усіх вікон. Ресурси включають набір кольорів (зокрема, базовий зелений для кнопок дії, червоний для помилок, жовтий для попереджень), стиль PrimaryButtonStyle для основних кнопок, стиль TextBoxStyle для полів введення, стиль ComboBoxStyle для випадаючих списків та стиль ListViewStyle для табличного подання даних.

#### 2.4.6 Реалізація конвертерів та допоміжних класів

Для перетворення даних у прив'язках використано набір конвертерів, розташованих у каталозі Converters. Конвертер OrderStatusToColorConverter перетворює значення переліку OrderStatus у відповідний об'єкт SolidColorBrush, що забезпечує кольорове підсвічування статусів замовлень у списку. Конвертер BooleanToVisibilityConverter перетворює логічне значення у значення переліку Visibility (Visible або Collapsed), що дозволяє динамічно показувати або приховувати елементи керування залежно від стану програми. Конвертер LowStockToStringConverter повертає текст «Низький запас» або «Достатньо» залежно від порівняння поточної кількості інгредієнта з мінімальним рівнем. Конвертер TotalValueConverter обчислює загальну вартість позиції складу як добуток кількості та ціни за одиницю. Утилітарний клас Logger на основі бібліотеки Serilog забезпечує запис діагностичних повідомлень, попереджень та помилок у текстовий файл щоденно з ротацією, що значно спрощує пошук і виправлення помилок під час експлуатації програми.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 41   |

#### 2.4.7 Реалізація механізму резервного копіювання

Модуль резервного копіювання реалізовано у SettingsViewModel шляхом простого копіювання файлу бази даних RestaurantDB.sqlite до окремого каталогу RestaurantBackups із додаванням часової мітки до імені файлу. Користувач має змогу ініціювати створення резервної копії вручну, а також налаштувати автоматичне копіювання з заданим інтервалом (у годинах). Відновлення з резервної копії виконується шляхом заміни поточного файлу бази даних на вибраний файл з каталогу бекапів, після чого програма автоматично перезапускається для коректного завантаження відновлених даних. У разі помилки під час копіювання (наприклад, відсутність прав доступу або пошкодження файлу) сервіс повідомлень відображає деталізоване сповіщення про характер помилки, що дозволяє адміністратору вжити відповідних заходів.

#### 2.5 Тестування програми

Тестування розробленої інформаційної системи проводилося з метою перевірки відповідності реалізованого функціоналу вимогам, виявлення та усунення помилок, а також оцінювання зручності використання програмного засобу кінцевими користувачами. Процес тестування охоплював перевірку коректності роботи окремих модулів, взаємодії компонентів системи між собою та з базою даних, а також оцінювання продуктивності програми за типових сценаріїв роботи закладу громадського харчування.

Тестування виконувалося поетапно із застосуванням різних стратегій, а саме модульного тестування, інтеграційного тестування, системного тестування та приймального тестування. Кожен з цих етапів мав власну мету та передбачав використання специфічних тестових даних і сценаріїв. Усі результати тестування фіксувалися у формі протоколів із зазначенням виявлених дефектів, їхньої критичності та способів усунення.

**Модульне тестування** спрямовувалося на перевірку працездатності окремих методів класів, зокрема сервісів автентифікації, обробки замовлень,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 42   |

складського обліку та формування звітів. Для цього було створено набір тестових сценаріїв, які охоплювали як штатні, так і граничні умови. Наприклад, метод `AuthenticateAsync` сервісу `IAuthenticationService` перевірявся на коректність розпізнавання правильних облікових даних (логін та пароль збігаються з записами у базі даних) та на здатність відхиляти неправильні комбінації. Метод `CheckIngredientAvailabilityAsync` сервісу `IInventoryService` тестувався з різними значеннями замовленої кількості страви, щоб переконатися, що система коректно визначає недостатню кількість інгредієнтів на складі. Також перевірялася правильність розрахунку загальної суми замовлення методом `CalculateOrderTotalAsync` сервісу `IOrderService` при різних значеннях кількості позицій та відсотків знижки. Модульне тестування дозволило виявити та усунути помилки на ранніх стадіях розробки, зокрема, неправильне округлення сум під час розрахунку податку та некоректне оновлення статусу столу після скасування замовлення.

**Інтеграційне тестування** мало на меті перевірку взаємодії між різними компонентами системи. Основна увага приділялася взаємодії між `ViewModel` та сервісами, а також між сервісами та базою даних. Тестувалися сценарії створення нового замовлення, які передбачають послідовне виконання низки операцій: вибір столу, вибір страви з меню, додавання позиції до замовлення, автоматичне списання інгредієнтів із таблиці `InventoryItems` згідно з рецептом, зміна статусу столу на «Зайнятий», розрахунок загальної суми та фіксація замовлення в таблиці `Orders`. Перевірялося, чи у разі нестачі інгредієнта система не створює замовлення та не змінює статус столу, а також видає відповідне попередження. Іншим важливим сценарієм було оновлення існуючого замовлення після додавання або видалення страв: перевірялося, чи коректно перераховується сума (субтотал, податок, загальна сума) та чи правильно оновлюються відповідні записи в таблиці `OrderItems`. Інтеграційне тестування виявило проблеми з каскадним видаленням пов'язаних записів (зокрема, при видаленні страви з меню видалялися не лише рецепти, але й

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 43   |

позиції в історичних замовленнях, що було виправлено налаштуванням зовнішніх ключів з обмеженням ON DELETE RESTRICT).

**Системне тестування** виконувалося на повністю зібраній програмі з метою перевірки відповідності реалізованого функціоналу затвердженим вимогам. Було перевірено всі основні функціональні модулі: автентифікація (вхід з різними ролями та відповідне обмеження доступу до пунктів меню), управління меню (додавання, редагування, видалення страв та інгредієнтів рецептів), обробка замовлень, резервування столів, складський облік (поповнення, коригування, контроль низького запасу), формування звітів, налаштування системи. Додатково перевірялися нефункціональні вимоги, зокрема швидкодія програми при роботі з великою кількістю записів (понад 1000 позицій у таблицях замовлень та продажів) та коректність обробки помилок при втраті зв'язку з файлом бази даних. У ході системного тестування було виявлено, що при тривалій роботі програми без перезавантаження деякі фільтри на формі звітів перестають оновлюватися, що було усунуто додаванням примусового виклику `CommandManager.InvalidateRequerySuggested` після зміни параметрів звіту.

**Тестування інтерфейсу користувача** проводилося з метою оцінки зручності та інтуїтивної зрозумілості розташування елементів керування. Перевірялося, чи всі текстові написи виконані українською мовою, чи коректно масштабуються вікна при зміні роздільної здатності екрана, чи працюють спливаючі підказки для кнопок та полів введення. Особлива увага приділялася доступності програми для офіціантів та кухарів, які можуть не мати глибоких знань у сфері інформаційних технологій. З цією метою було проведено тестування за участю трьох добровольців із числа потенційних користувачів, які виконували типові сценарії роботи: створення замовлення, додавання страв, зміна статусу, формування звіту. За результатами тестування було скориговано розташування деяких кнопок (зокрема, кнопку «Оновити замовлення» переміщено ближче до списку страв для зменшення кількості

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 44   |

помилкових натискань) та збільшено шрифт на формі роботи зі столовими резервуваннями.

**Тестування продуктивності** виконувалося шляхом вимірювання часу відгуку програми при виконанні найбільш ресурсоємних операцій. Вимірювався час завантаження форми звітів при періоді аналізу в один місяць (у тестовій базі даних містилося близько 5000 записів у таблиці SalesLogs), час формування звіту про популярні страви, час пошуку страв за неповною назвою у формі меню, час відкриття діалогу редагування страви з великою кількістю інгредієнтів. Усі вимірювання проводилися на персональному комп'ютері з процесором Intel Core i5, 8 ГБ оперативної пам'яті та твердотільним накопичувачем. Результати засвідчили, що жодна з операцій не перевищує порогу в дві секунди, що є прийнятним для роботи в реальному часі.

**Тестування безпеки** включало перевірку коректності розмежування прав доступу на основі ролей. Тестувалося, чи не може офіціант отримати доступ до модуля налаштувань або звітів, чи не може кухар змінювати ціни на страви або видаляти інгредієнти зі складу, чи не може менеджер додавати нових адміністраторів. Для цього було виконано набір спроб доступу до заборонених функцій через інтерфейс програми та через безпосереднє викликання відповідних методів ViewModel за допомогою налагоджувальника. У всіх випадках система або приховувала відповідні кнопки, або блокувала виконання команд методом CanExecute. Крім того, перевірялася безпека зберігання паролів: у базі даних паролі зберігаються виключно у хешованому вигляді, що унеможлиблює їх відновлення навіть у разі компрометації файлу бази даних.

За результатами проведеного тестування можна стверджувати, що розроблена інформаційна система працює стабільно, відповідає встановленим вимогам та може бути рекомендована до впровадження в реальному закладі громадського харчування. Усі виявлені під час тестування дефекти були класифіковані, зареєстровані та усунені, а критичних помилок, що могли б призвести до втрати даних або неможливості виконання ключових бізнес-

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 45   |

процесів, не зафіксовано. У таблиці 2.5.1 наведено зведені результати тестування за типами тестів та кількістю виявлених і виправлених дефектів.

**Таблиця 2.5.1 – Результати тестування програмного засобу**

| Тип тестування            | Кількість тестових сценаріїв | Виявлено дефектів | Виправлено | Залишилося |
|---------------------------|------------------------------|-------------------|------------|------------|
| Модульне тестування       | 35                           | 8                 | 8          | 0          |
| Інтеграційне тестування   | 42                           | 12                | 12         | 0          |
| Системне тестування       | 28                           | 5                 | 5          | 0          |
| Тестування інтерфейсу     | 18                           | 4                 | 4          | 0          |
| Тестування продуктивності | 12                           | 2                 | 2          | 0          |
| Тестування безпеки        | 15                           | 1                 | 1          | 0          |

Проведений комплекс тестувань підтвердив готовність програмного забезпечення до експлуатації у реальних умовах. Подальше вдосконалення системи може стосуватися розширення функціональності, зокрема інтеграції з фіскальними реєстраторами та онлайн-сервісами доставки їжі.

## Висновки до розділу 2

У другому розділі кваліфікаційної роботи виконано проєктування програмного засобу автоматизації роботи закладу громадського харчування, що включає розробку логічної та фізичної моделі бази даних, визначення архітектури системи, реалізацію основних функціональних модулів та проведення комплексу тестувань.

Розроблено логічну модель бази даних у вигляді ER-діаграми, яка містить дев'ять основних сутностей: MenuItem (страви), InventoryItem (інгредієнти складу), RecipeIngredient (рецепти), Table (столи), Order (замовлення), OrderItem (позиції замовлення), Reservation (бронювання), Employee (співробітники) та SalesLog (логи продажів). Визначено первинні та зовнішні ключі, типи зв'язків між сутностями (1:1, 1:N, N:M), а також класифіковано атрибути на прості, складні, однозначні та багатозначні. На основі логічної моделі створено фізичну модель для системи управління базами даних SQLite, яка включає детальний опис кожної таблиці з точним зазначенням типів даних, обмежень та правил цілісності. Така структура

забезпечує несуперечливість даних, унеможлиблює дублювання інформації та гарантує референтну цілісність у межах усієї системи.

Визначено перелік основних форм програмного засобу (авторизації, головне вікно, замовлення, меню, столики та резервування, склад, звіти, налаштування) та детально описано призначення кожної з них, формати введення даних, обмеження та способи взаємодії з користувачем. Зазначено, що всі текстові написи виконано українською мовою, а введення даних контролюється як на стороні клієнта (перевірка через механізми валідації), так і на рівні бази даних (обмеження цілісності).

Обґрунтовано вибір архітектурного патерну MVVM (Model-View-ViewModel), який забезпечує чітке розділення відповідальності між рівнями представлення, логіки та даних. Описано взаємодію компонентів системи, зокрема роль сервісів (IAuthenticationService, IOrderService, IInventoryService, IReportingService, IMessageService), механізм Dependency Injection для слабкого зв'язування залежностей та сервіс навігації INavigationService для керування переходами між різними поданнями. Наведено діаграму компонентів та схему взаємодії відповідно до патерну MVVM, що візуалізує потоки даних у системі.

У підрозділі програмної реалізації детально описано загальну організацію проєкту, структуру каталогів та призначення кожного з них. Розкрито особливості реалізації доступу до даних через Entity Framework Core, зокрема використання асинхронних методів для забезпечення чуйності інтерфейсу користувача. Описано реалізацію бізнес-логіки у ViewModel із застосуванням атрибутів [ObservableProperty] та [RelayCommand] бібліотеки CommunityToolkit.Mvvm, що зменшує обсяг шаблонного коду. Висвітлено реалізацію ключових сервісів (OrderService, InventoryService, ReportingService), їхні методи та роль у виконанні основних бізнес-процесів. Наведено приклади реалізації інтерфейсу користувача за допомогою XAML, стилів та конвертерів, а також описано механізм резервного копіювання бази даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 47   |

Проведено комплексне тестування програмного засобу, яке охоплювало модульне, інтеграційне, системне тестування, тестування інтерфейсу користувача, продуктивності та безпеки. Загалом було виконано 150 тестових сценаріїв, виявлено та усунено 32 дефекти. Результати тестування підтвердили працездатність системи, її відповідність встановленим функціональним вимогам, зручність інтерфейсу для кінцевих користувачів та достатній рівень продуктивності для роботи в реальному закладі громадського харчування.

Виконане проектування та реалізація програмного засобу повністю відповідає вимогам технічного завдання, а отримані результати є підґрунтям для впровадження системи в практичну діяльність та подальшого розвитку (інтеграція з фіскальним обладнанням, онлайн-сервісами доставки, мобільними додатками).

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 48   |

## РОЗДІЛ 3. КЕРІВНИЦТВО КОРИСТУВАННЯ ПРОГРАМОЮ

### 3.1 Мінімальні вимоги до системи

Для коректної інсталяції та стабільної роботи інформаційної системи автоматизації роботи закладу громадського харчування необхідно, щоб апаратне та програмне забезпечення комп'ютера, на якому планується експлуатація програми, відповідало наведеним нижче мінімальним вимогам. Дотримання цих вимог забезпечує належну продуктивність системи під час виконання всіх функціональних операцій, зокрема роботи з базою даних, формування звітів та взаємодії з графічним інтерфейсом користувача.

#### Апаратні вимоги

Процесор (CPU) повинен мати тактову частоту не менше ніж 1,8 ГГц. Рекомендується використовувати процесори з підтримкою набору інструкцій SSE2, що є типовим для всіх сучасних моделей Intel Core, AMD Ryzen та аналогічних. Оперативна пам'ять (RAM) має становити щонайменше 2 ГБ для роботи операційної системи та самої програми. Для забезпечення плавної роботи з великими масивами даних (понад 10 000 записів у таблицях замовлень або продажів) рекомендовано збільшити обсяг пам'яті до 4 ГБ та більше. Вільне місце на жорсткому диску має становити не менше ніж 500 МБ, з яких приблизно 100 МБ займає сама програма, а решта призначена для файлу бази даних, логів подій та резервних копій. Зі зростанням кількості записів у базі даних (протягом тривалого періоду експлуатації) потреба в дисковому просторі може збільшитись до 1-2 ГБ. Дисплей повинен підтримувати роздільну здатність екрана щонайменше 1280×720 пікселів для коректного відображення всіх вікон та елементів керування. Оптимальним є використання моніторів з роздільною здатністю 1920×1080 пікселів або вищою.

#### Програмні вимоги

Операційна система – Microsoft Windows 10 версії 1809 або новішої (64-розрядна), а також Windows 11. Система також може працювати на Windows Server 2016 та новіших версіях, однак для використання як робочого місця офіціанта або адміністратора рекомендовано застосовувати клієнтські редакції

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 49   |

Windows. Для забезпечення роботи платформи .NET 9, на якій розроблено програму, необхідно встановити .NET Runtime 9.0.0 або новішої версії. Інсталяційний пакет програми може включати цей компонент або автоматично завантажувати його з мережі Інтернет під час встановлення за умови наявності з'єднання. Якщо комп'ютер не має доступу до Інтернету, слід заздалегідь завантажити та встановити .NET 9 Desktop Runtime з офіційного веб-сайту Microsoft.

#### **Додаткові вимоги**

Для роботи системи не потрібне постійне з'єднання з мережею Інтернет, оскільки всі дані зберігаються локально у файлі бази даних SQLite. Однак наявність інтернет-з'єднання може знадобитися під час початкової інсталяції для завантаження компонентів .NET Runtime, а також для отримання оновлень програми в майбутньому. Необхідні права користувача – рівень доступу «Читання та запис» до каталогу, в якому розміщено виконуваний файл програми та файл бази даних, оскільки система створює файли логів, резервні копії та може оновлювати базу даних. Для створення резервних копій також потрібні права на запис у довільну теку, обрану користувачем. Антивірусне програмне забезпечення не повинно блокувати роботу програми або доступ до файлу бази даних. Рекомендується додати виконуваний файл програми до списку винятків антивіруса, якщо спостерігаються уповільнення роботи або помилки доступу.

Дотримання описаних мінімальних системних вимог гарантує стабільну та продуктивну роботу інформаційної системи. Використання більш потужного обладнання (процесор із вищою тактовою частотою, збільшений обсяг оперативної пам'яті, твердотільний накопичувач) дозволяє покращити швидкодію системи, особливо під час формування зведених звітів за тривалі періоди або за одночасної роботи декількох користувачів з однією базою даних через мережевий доступ до спільної папки.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
|      |      |          |        |      |                           | 50   |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |      |

## 3.2 Інструкція для роботи з програмою

### 3.2.1 Загальні положення

Інструкція для роботи з програмою призначена для кінцевих користувачів інформаційної системи автоматизації роботи закладу громадського харчування, а саме для адміністраторів, менеджерів, офіціантів та кухарів. У цьому документі наведено послідовність дій для виконання основних операцій, які виникають у процесі щоденної діяльності закладу. Перед початком роботи з програмою необхідно переконатися, що комп'ютер відповідає мінімальним системним вимогам, а файл бази даних RestaurantDB.sqlite розташовано в каталозі Base поруч із виконуваним файлом програми.

### 3.2.2 Запуск програми та вхід у систему

Для запуску програми необхідно двічі клацнути лівою кнопкою миші на файлі RestaurantAutomation.exe. Після завантаження на екрані з'явиться вікно авторизації, яке містить два текстові поля – «Логін» та «Пароль» – а також кнопку «Увійти». Для отримання доступу до системи користувач повинен ввести облікові дані, які були надані адміністратором. Логін вводиться у відповідне текстове поле без будь-яких обмежень щодо формату. Пароль вводиться у поле, призначене для пароля, причому символи не відображаються на екрані з метою забезпечення конфіденційності. Після введення облікових даних слід натиснути кнопку «Увійти». Якщо логін та пароль введено правильно, програма визначить роль користувача (адміністратор, менеджер, офіціант або кухар) та відкриє головне вікно програми. Якщо облікові дані недійсні, система відобразить відповідне повідомлення про помилку, після чого користувач може повторити спробу входу.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 51   |

### 3.2.3 Інтерфейс головного вікна

Головне вікно програми складається з двох основних областей. Ліва частина являє собою бокове меню навігації, яке містить кнопки для переходу між функціональними модулями: «Замовлення», «Столики», «Меню», «Склад», «Звіти», «Налаштування» та «Вихід». Набір доступних пунктів залежить від ролі користувача. Права частина вікна є робочою областю, де відображається вміст вибраного модуля. У верхній частині бокового меню розташовано інформаційний блок із зазначенням прізвища та імені поточного користувача, а також його ролі в системі. Для виходу з програми необхідно натиснути кнопку «Вихід» у нижній частині бокового меню.

### 3.2.4 Робота з модулем «Замовлення»

Модуль «Замовлення» є основним робочим інструментом для офіціантів та менеджерів. Вікно модуля поділено на дві частини. У лівій частині відображається список активних замовлень, кожне з яких представлено у вигляді окремого блоку з інформацією про номер столу, час створення, загальну суму та поточний статус. Для перегляду деталей замовлення необхідно клацнути на відповідному блоці, після чого його вміст завантажиться у праву частину вікна.

Права частина призначена для створення нового замовлення або редагування існуючого. Для створення нового замовлення слід натиснути кнопку « Нове» у верхній частині правої області. Потім необхідно обрати стіл з випадаючого списку «Стіл». Далі у списку «Додати страву» потрібно обрати страву, вказати кількість порцій у полі з числовим значенням (за замовчуванням встановлено одиницю) та натиснути кнопку «» для додавання страви до замовлення. Додані страви відображаються у списку «Страви в замовленні» із зазначенням назви, категорії, кількості та ціни. Для видалення страви з замовлення слід натиснути кнопку з хрестиком () у відповідному рядку списку. Після завершення формування замовлення необхідно натиснути кнопку «Оформити», яка створить нове замовлення,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
|      |      |          |        |      |                           | 52   |
| Змн. | Арк. | № докум. | Підпис | Дата |                           |      |

спише інгредієнти зі складу згідно з рецептами змінених страв та додасть його до списку активних замовлень.

Для редагування існуючого замовлення необхідно вибрати його у лівій частині вікна. Після цього можна додавати нові страви, змінювати кількість вже доданих страв (шляхом повторного додавання тієї самої страви) або видаляти позиції. Після внесення змін слід натиснути кнопку «Оновити», яка збереже зміни в базі даних та оновить інформацію про замовлення. Для зміни статусу замовлення (з «Нове» на «Готується», «Готове», «Подано» або «Оплачено») необхідно вибрати потрібне значення у випадяючому списку «Статус», розташованому під загальною сумою замовлення.

### 3.2.5 Робота з модулем «Столики»

Модуль «Столики» призначений для управління статусами столів та бронюваннями. Він містить дві вкладки: «Столики» та «Бронювання».

Вкладка «Столики» відображає візуальну схему столів у вигляді сітки кольорових кнопок. Кожна кнопка відповідає одному столу та містить його номер, місткість (у дужках) та розташування. Колір кнопки інформує про поточний статус столу: зелений – вільний, червоний – зайнятий, помаранчевий – заброньований, сірий – прибирання. Для зміни статусу столу необхідно клацнути на ньому правою кнопкою миші, у контекстному меню обрати потрібний статус («Вільний», «Зайнятий», «Заброньований» або «Прибирання»). Для редагування інформації про стіл (наприклад, зміна місткості або розташування) слід вибрати стіл лівою кнопкою миші, після чого на панелі інструментів стануть доступними кнопки «Редагувати стіл» та «Видалити стіл». Для додавання нового столу необхідно натиснути кнопку « Додати стіл» на панелі інструментів.

Вкладка «Бронювання» призначена для управління попередніми замовленнями столів. У верхній частині вкладки розташовано календар для вибору дати, а також кнопка « Нове бронювання». Після вибору дати система відображає список усіх бронювань на цей день. Для створення нового

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 53   |

бронювання необхідно натиснути кнопку « Нове бронювання», у діалоговому вікні заповнити поля: «Стіл» (вибрати зі списку), «Ім'я клієнта» (обов'язкове поле), «Телефон», «Email», «Дата та час», «Кількість гостей» та «Особливі побажання». Після заповнення всіх необхідних полів слід натиснути кнопку «Зберегти». Для редагування або скасування існуючого бронювання необхідно вибрати його у списку, після чого на панелі інструментів стануть доступними кнопки «Редагувати» та «Видалити».

### 3.2.6 Робота з модулем «Меню»

Модуль «Меню» призначений для адміністраторів та менеджерів з метою управління складом страв та напоїв. Головне вікно модуля містить таблицю з переліком усіх страв. У таблиці відображаються такі дані: назва страви, опис, категорія, ціна, статус доступності (доступна/недоступна) та перелік інгредієнтів, що входять до рецепту. Панель інструментів містить кнопки « Додати страву», « Редагувати», « Видалити» та « Змінити статус», а також поле для пошуку за назвою та випадаючий список для фільтрації за категорією.

Для додавання нової страви необхідно натиснути кнопку « Додати страву». У діалоговому вікні, що з'явиться, слід заповнити наступні поля: «Назва страви» (обов'язкове), «Опис» (необов'язкове), «Категорія» (можна обрати зі списку або ввести нову), «Ціна» (додатне число) та позначити прапорець «Доступно», якщо страва готова до замовлення. Для додавання інгредієнтів необхідно натиснути кнопку « Додати інгредієнт», у новому рядку вибрати інгредієнт з випадаючого списку та вказати необхідну кількість. Після заповнення всіх даних слід натиснути кнопку «Зберегти». Для редагування існуючої страви необхідно вибрати її у таблиці (рядок підсвітиться синім кольором) та натиснути кнопку « Редагувати», після чого відкриється аналогічне діалогове вікно з попередньо заповненими даними. Для видалення страви слід вибрати її у таблиці та натиснути кнопку « Видалити», підтвердивши дію у спливаючому вікні. Кнопка « Змінити

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 54   |

статус» дозволяє швидко перемкнути стан доступності страви без відкриття діалогу редагування.

### 3.2.7 Робота з модулем «Склад»

Модуль «Склад» призначений для ведення обліку інгредієнтів, необхідних для приготування страв. Головне вікно містить таблицю з переліком усіх інгредієнтів та наступними колонками: назва, кількість, одиниця виміру, мінімальний рівень, ціна за одиницю, загальна вартість, постачальник, дата останнього поповнення та статус. Рядки з інгредієнтами, кількість яких опустилася нижче мінімального рівня, підсвічуються червоним кольором. У верхній частині вікна відображаються підсумкові показники: всього позицій, загальна вартість всіх запасів та кількість позицій з низьким запасом. Панель інструментів містить кнопки « Додати інгредієнт», « Редагувати», « Видалити», « Поповнити», « Коригувати», а також поле пошуку, фільтр за одиницями виміру та прапорець «Тільки низький запас».

Для додавання нового інгредієнта необхідно натиснути кнопку « Додати інгредієнт». У діалоговому вікні слід заповнити поля: «Назва інгредієнта» (обов'язкове), «Кількість» (додатнє число), «Одиниця виміру» (вибрати зі списку «кг», «л», «шт» тощо), «Мінімальний рівень» (число, при досягненні якого система сигналізуватиме про низький запас), «Ціна за одиницю» (необов'язкове) та «Постачальник» (необов'язкове). Для збільшення кількості інгредієнта на складі необхідно вибрати потрібний рядок у таблиці та натиснути кнопку « Поповнити», після чого в діалоговому вікні ввести кількість одиниць для додавання. Для ручного коригування кількості (наприклад, при списанні через псування або інвентаризації) слід використовувати кнопку « Коригувати», вказавши від'ємне значення для зменшення залишку.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 55   |

### 3.2.8 Робота з модулем «Звіти»

Модуль «Звіти» призначений для формування аналітичної інформації на основі накопичених даних про продажі, замовлення та роботу співробітників. У верхній частині вікна розташовано панель налаштувань з випадającym списком «Тип звіту», полями для вибору початкової та кінцевої дати (календарі) та кнопкою «Сформувати». Доступні наступні типи звітів: «Продажі», «Популярні страви», «Фінанси», «Склад» та «Співробітники». Після вибору типу звіту та встановлення періоду необхідно натиснути кнопку «Сформувати». Система відобразить звіт у вигляді таблиць, числових показників та діаграм.

Звіт «Продажі» містить загальний дохід за обраний період, кількість замовлень, середній чек, розподіл продажів за категоріями страв, способами оплати та годинами доби, а також графік динаміки продажів. Звіт «Популярні страви» показує перелік страв, які найчастіше замовляли, із зазначенням кількості проданих одиниць та отриманої виручки. Звіт «Фінанси» надає інформацію про валовий дохід, податки, знижки, чистий дохід, собівартість проданих товарів, операційні витрати, валовий та чистий прибуток, а також показники маржинальності. Звіт «Склад» відображає загальну кількість позицій, загальну вартість запасів та перелік інгредієнтів з низьким рівнем. Звіт «Співробітники» показує ефективність роботи кожного працівника (кількість оброблених замовлень, загальна сума продажів, середній чек). Усі звіти можна експортувати у формат CSV для подальшого аналізу в інших програмах (наприклад, Microsoft Excel) за допомогою кнопки « Експорт звіту» на панелі налаштувань.

### 3.2.9 Робота з модулем «Налаштування»

Модуль «Налаштування» доступний виключно адміністраторам та менеджерам. Він складається з трьох вкладок: «Співробітники», «База даних» та «Заклад».

Вкладка «Співробітники» призначена для управління обліковими записами персоналу. Таблиця відображає список усіх співробітників із

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 56   |

зазначенням їхніх ПІБ, логінів, посад, контактних даних, дати прийняття на роботу та статусу активності. Панель інструментів містить кнопки « Додати співробітника», « Редагувати», « Видалити» та « Скинути пароль». Для додавання нового співробітника необхідно заповнити діалогове вікно, вказавши ПІБ, логін, пароль (мінімум 6 символів), посаду (адміністратор, менеджер, офіціант або кухар), контактний телефон, електронну пошту та позначити прапорець «Активний». Після збереження новий співробітник отримує доступ до системи з правами, визначеними його роллю.

Вкладка «База даних» містить інформацію про поточний стан бази даних (розташування файлу, розмір), налаштування резервного копіювання (папка для збереження бекапів, інтервал автоматичного копіювання) та кнопки для ручного створення резервної копії, відновлення з бекапу та оптимізації бази даних. Регулярне створення резервних копій рекомендується для запобігання втраті даних у разі збоїв обладнання.

Вкладка «Заклад» призначена для заповнення реквізитів самого закладу громадського харчування: назва закладу, код ЄДРПОУ, адреса, контактний телефон та електронна пошта. Ці дані використовуються у звітах та можуть бути змінені в будь-який час шляхом редагування відповідних текстових полів та натискання кнопки «Зберегти налаштування».

### 3.2.10 Завершення роботи з програмою

Для завершення сеансу роботи необхідно натиснути кнопку «Вихід» у боковому меню головного вікна. Система закриє всі відкриті форми та завершить виконання програми. Перед виходом рекомендується створити резервну копію бази даних, особливо якщо протягом робочого дня було внесено значну кількість змін. У разі аварійного завершення роботи програми (наприклад, через знеструмлення комп'ютера) база даних SQLite зберігає цілісність даних завдяки підтримці транзакцій, однак останні незбережені

зміни можуть бути втрачені. Після перезапуску програми необхідно виконати вхід, використовуючи ті самі облікові дані.

### **Висновки до розділу 3**

У третьому розділі кваліфікаційної роботи розроблено керівництво користувача для інформаційної системи автоматизації роботи закладу громадського харчування. Визначено мінімальні апаратні (процесор від 1,8 ГГц, оперативна пам'ять від 2 ГБ, вільний дисковий простір від 500 МБ) та програмні (операційна система Windows 10/11, .NET 9 Desktop Runtime) вимоги до системи, дотримання яких забезпечує стабільну роботу програми.

Створено детальну інструкцію для кінцевих користувачів, яка охоплює всі функціональні модулі системи: авторизацію, головне вікно, роботу із замовленнями (створення, редагування, зміна статусу, оплата), управління столами та бронюваннями, ведення меню та складського обліку, формування аналітичних звітів, адміністрування персоналу та резервне копіювання бази даних. Інструкція містить чіткі послідовності дій для виконання кожної операції, опис форматів введення даних та призначення кожного елемента керування. Розроблене керівництво дозволяє користувачам із різним рівнем комп'ютерної грамотності ефективно використовувати систему в щоденній роботі, мінімізуючи ймовірність помилок та підвищуючи загальну продуктивність закладу громадського харчування.

## ВИСНОВКИ

У результаті виконання кваліфікаційної роботи розроблено інформаційну систему автоматизації роботи закладу громадського харчування, яка забезпечує комплексне вирішення завдань управління меню, обробки замовлень, резервування столів, складського обліку, управління персоналом та формування аналітичних звітів.

У першому розділі роботи проведено аналіз предметної області, визначено ключові бізнес-процеси, які потребують автоматизації, розглянуто існуючі програмні рішення (хмарні системи, локальні стаціонарні рішення, open-source проєкти), виявлено їхні переваги та недоліки, обґрунтовано доцільність створення власної системи. Сформульовано функціональні вимоги до програмного засобу та обґрунтовано вибір технологічного стеку (C#, .NET 9, WPF, SQLite, Entity Framework Core, CommunityToolkit.Mvvm). Результатом першого розділу стала чітка постановка завдання на розробку, яка визначила напрямок подальшого проєктування.

У другому розділі спроектовано логічну та фізичну модель бази даних, що містить дев'ять основних сутностей. Розроблено ER-діаграму, визначено первинні та зовнішні ключі, типи зв'язків між сутностями. Створено структуру таблиць для SQLite з детальним описом полів, типів даних та обмежень цілісності. Описано архітектуру програмного засобу на основі патерну MVVM, розроблено всі необхідні форми, реалізовано бізнес-логіку у вигляді сервісів та ViewModel. Проведено комплексне тестування (модульне, інтеграційне, системне, інтерфейсу, продуктивності, безпеки), яке підтвердило працездатність системи, її відповідність вимогам та готовність до промислової експлуатації.

У третьому розділі розроблено керівництво користувача, визначено мінімальні системні вимоги до апаратного та програмного забезпечення, створено детальну інструкцію для роботи з усіма модулями системи (авторизація, замовлення, столики та бронювання, меню, склад, звіти, налаштування). Інструкція орієнтована на кінцевих користувачів з різним

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 59   |

рівнем комп'ютерної грамотності та містить чіткі алгоритми виконання основних операцій.

Розроблена інформаційна система має практичну цінність та може бути впроваджена в закладах громадського харчування різного типу (ресторани, кафе, їдальні). Її використання дозволяє підвищити швидкість обслуговування відвідувачів, мінімізувати помилки персоналу, забезпечити оперативний контроль залишків продуктів на складі, автоматизувати формування звітності за будь-який період, зменшити витрати на адміністрування завдяки локальному розгортанню на одному комп'ютері без потреби в окремому сервері. Подальший розвиток системи може бути спрямований на інтеграцію з фіскальними реєстраторами, онлайн-сервісами доставки їжі, створення мобільного додатку для офіціантів, а також розширення функціоналу звітності.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 60   |

## ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. Відомості Верховної Ради України. 2010. № 34. Ст. 481. (з наступними змінами та доповненнями).
2. Про електронну ідентифікацію та електронні довірчі послуги : Закон України від 05.10.2017 № 2155-VIII. Відомості Верховної Ради України. 2017. № 45. Ст. 398.
3. Про затвердження Правил роботи закладів (підприємств) ресторанного господарства : Наказ Міністерства економіки України та Міністерства фінансів України від 24.07.2002 № 219/119/399. Офіційний вісник України. 2002. № 33. Ст. 1573.
4. ДСТУ 2844-94. Розроблення та постачання програмних засобів. Загальні
5. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models. Geneva : International Organization for Standardization, 2011. 34 p.
6. Ситник В. Ф. Інформаційні системи в менеджменті : навчальний посібник. Київ : КНЕУ, 2017. 548 с.
7. Глинський Я. М., Аниськін В. В. Проектування інформаційних систем : підручник. Львів : Магнолія 2006, 2019. 286 с.
8. Пасічник В. В., Резніченко В. А. Організація баз даних та знань : підручник. Київ : Центр учбової літератури, 2020. 384 с.
9. Тарнавський Ю. А., Кузьменко І. М. Інформаційні технології : теорія та практика. Київ : КПІ ім. Ігоря Сікорського, 2021. 312 с.
10. Швачич Г. Г., Толстой В. В., Петречук Л. М. Сучасні інформаційні технології : підручник. Дніпро : НМетАУ, 2019. 336 с.
11. Фаріон В. М., Фаріон Т. В. Інструментальні засоби візуального проектування інформаційних систем : навчальний посібник. Тернопіль : ТНТУ ім. І. Пулюя, 2020. 152 с.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | КРФМБ.122.042.039.2026.ПЗ | Арк. |
| Змн. | Арк. | № докум. | Підпис | Дата |                           | 61   |

12. Troelsen A., Japikse P. Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming. 11th ed. New York : Apress, 2022. 1680 p.
13. Price M. J. C# 12 and .NET 8 – Modern Cross-Platform Development Fundamentals. 8th ed. New York : Apress, 2023. 825 p.
14. Schwichtenberg H. Moderne Datenzugriffslösungen mit Entity Framework Core 9.0. Essen : IT-Visions, 2024. 824 p.
15. Nagel C. Professional C# and .NET – 2025 edition. Hoboken : John Wiley & Sons, 2024. 1100 p.
16. MacDonald M. WPF 4.5 Unleashed. 2nd ed. Indianapolis : Sams Publishing, 2021. 864 p.
17. Prostak L., Zinko T., Shevchuk R. Microsoft .NET and WPF technology as a tool for creating a mobile restaurant. Young Scientist. 2020. No. 3 (79). P. 47–52.
18. Sharp J. Microsoft Visual C# Step by Step. 10th ed. Redmond : Microsoft Press, 2021. 832 p.
19. Petzold C. Code: The Hidden Language of Computer Hardware and Software. 3rd ed. Redmond : Microsoft Press, 2023. 480 p.
20. Meijer E. Applied Software Architecture and Design with .NET. New York : Manning Publications, 2023. 550 p.
21. Chawdhury T. H., Laskar R. R. A survey on the restaurant management system. International Journal of Computer Applications. 2019. Vol. 178, No. 12. P. 28–32.
22. Albahari J., Albahari B. C# 12 in a Nutshell. Sebastopol : O'Reilly Media, 2023. 1100 p.
23. Liberman A., Vokorokos L., Balaz A. The restaurant information system based on the client-server architecture. Journal of Information, Control and Management Systems. 2018. Vol. 6, No. 2. P. 45–52.
24. Griffin M., Jorgensen A. The MVVM pattern: modern approach to desktop application development. Microsoft Developer Network Architecture Journal. 2021. No. 45. P. 12–18.