

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра Автоматизації технологічних процесів і робототехнічних систем



КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА
на тему: Автоматизація управління ресторанним бізнесом
підприємства «Хмари»
(назва кваліфікаційної роботи згідно наказу ОНТУ)

Здобувача М.С. Тесленко

(прізвище, ініціали)

2ск курсу АТ-20 групи

Керівник: доц. О.М. Жигайло

(посада, прізвище та ініціали)

Консультанти:

Жигайло О.М., доц. каф. АТПіРС

(посада, прізвище та ініціали)

Дец Д.В., ст. викл. каф. АТПіРС

(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від «___» _____ 20__р., протокол №_____.

Завідувач кафедри АТПіРС _____

(назва кафедри)

(підп.)

І.М. Світій

(ПБ)

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Інститут	Навчально-науковий інститут комп'ютерної інженерії, автоматизації, робототехніки та програмування ім. П.М. Платонова
Кафедра	Автоматизації технологічних процесів і робототехнічних систем
Ступінь вищої освіти	Бакалавр
Спеціальність	174 Автоматизація, комп'ютерно-інтегровані технології та робототехніка
Освітньо-професійна програма	Комп'ютерні системи та програмна інженерія в автоматизації та робототехніці

ЗАТВЕРДЖУЮ

Зав. кафедри АТПіРС

_____ І.М. Світий

«27» січня 2026 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Здобувач **Тесленко Максим Сергійович**

1. Тема роботи **«Автоматизація процесу управління ресторанним бізнесом»**

2. Керівник кваліфікаційної роботи Жигайло Олексій Михайлович, к.т.н., доцент.

Пункти 1 і 2 затверджені наказом ОНТУ № 39-03 від 24.01.2026 р.

3. Строк подання здобувачем випускної кваліфікаційної роботи на кафедру **«12» червня** 2026 р.

4. Вихідні дані до кваліфікаційної роботи: матеріали індивідуальних завдань виробничої і переддипломної практик, курсових і самостійних робіт, комплексне кваліфікаційне завдання (для заочників).

5. Зміст кваліфікаційної роботи (розділи, які потрібно розробити):

Розділ 1. Загальна характеристика підприємства та бізнес-процесу (записати конкретну назву процесу з теми роботи), обґрунтування доцільності та шляхів підвищення ефективності його управління.

Розділ 2. Розробка комплексу моделей бізнес-процесів підприємства, створення системи показників та концептуальної схеми управління. Імітаційне моделювання обраного бізнес-процесу із застосуванням нотації BPMN для його оптимізації.

Розділ 3. Затвердження меж об'єкта автоматизації, обґрунтування можливих функцій для його системи управління, опис існуючих аналогів автоматизованої системи управління підприємством (АСУП), розробка технічного завдання на створення нового веб-додатку, як основи АСУП.

Розділ 4. Розробка та створення концептуальної моделі даних та структури бази даних.

Розділ 5. Розробка макетів для інтерфейсів користувачів та їх дизайну, верстка дизайн-макетів. Створення функціональних модулів програмного продукту.

Розділ 6. Розробка SCADA для технологічного об'єкту моніторингу і(або) управління, забезпечення її зв'язку із створеним веб-додатком.

Розділ 7. Вибір технічних засобів, мета використання і реалізація їх зв'язку із створеним веб-додатком.

Розділ 8. Вибір методів інтелектуального аналізу даних, представлення результатів їх застосування, розробка алгоритмів для автоматизації підтримки прийняття рішень. Опис реалізованої програмної розробки (інструкція користувача).

Розділ 9. Попереднє обґрунтування економічної доцільності реалізації проекту.

6. Консультанти розділів кваліфікаційної роботи

Номер розділу	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Розділ 1	Жигайло О.М., доц. каф. АТПіРС		
Розділи 2, 3	Жигайло О.М., доц. каф. АТПіРС		
Розділ 4	Жигайло О.М., доц. каф. АТПіРС		
Розділ 5	Дец Д.В., ст. викладач. каф. АТПіРС		
Розділ 6	Дец Д.В., ст. викладач. каф. АТПіРС		
Розділ 7	Дец Д.В., ст. викладач. каф. АТПіРС		
Розділ 8	Жигайло О.М., доц. каф. АТПіРС		
Розділ 9	Савенко І.І., д.е.н., проф. каф. МіЛ		

7. Дата видачі завдання « 18 » березня 2026 р.

КАЛЕНДАРНИЙ ПЛАН

Номер розділу	Строк виконання етапів роботи	Примітка
Розділ 1	«24» березня 2025 р.	
Розділ 2	«28» березня 2025 р.	
Розділ 3	«04» квітня 2025 р.	
Розділ 4	«11» квітня 2025 р.	
Розділ 5	«23» квітня 2025 р.	
Розділ 6	«30» квітня 2025 р.	
Розділ 7	«14» травня 2025 р.	
Розділ 8	«20» травня 2025 р.	
Розділ 9	«23» травня 2025 р.	

Здобувач Тесленко М.С.

Керівник роботи Жигайло О.М.

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач Тесленко М.С. _____

Прізвище, ініціали

Підпис

ВСТУП.....	10
РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПІДПРИЄМСТВА ТА БІЗНЕС-ПРОЦЕСУ «РЕСТОРАННИЙ КЕЙТЕРИНГ», ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ШЛЯХІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЙОГО УПРАВЛІННЯ	12
1.1. Характеристика підприємства ресторану «Хмари» (м. Одеса) та його основних видів діяльності	12
1.2. Опис бізнес-процесу ресторанного кейтерингу та поточної схеми обробки заявок на бронювання місць	13
1.3. Опис існуючого програмного забезпечення для автоматизації обліку та підтримки інформаційних потоків закладу.	13
1.4. Виявлення і формалізація основних проблем життєдіяльності підприємства та недоліків телефонного резервування столиків.	14
1.5. Обґрунтування доцільності впровадження автоматизованої системи онлайн-бронювання на сайті як шляху підвищення ефективності управління	15
РОЗДІЛ 2. ОБСТЕЖЕННЯ ПІДПРИЄМСТВА ЯК ОБ'ЄКТА АВТОМАТИЗАЦІЇ ТА СТВОРЕННЯ КОМПЛЕКСУ МОДЕЛЕЙ ЙОГО БІЗНЕС-ПРОЦЕСІВ.	16
2.1. Аналіз організаційної структури підприємства та створення її моделі.	16
2.2. Виділення бізнес-процесів підприємства та створення його процесної моделі.	17
2.3. Моделювання бізнес-процесів підприємства в нотації eEPC.	20
2.4. Розробка системи показників і концептуальної схеми управління обраного бізнес-процесу.	25
2.5. Обґрунтування необхідності створення (розвитку) автоматизованої системи (АС) управління бізнес-процесами підприємства (постановка завдання).	27
РОЗДІЛ 3. СТВОРЕННЯ КОНЦЕПТУ ПРОГРАМИ ЯК ОСНОВИ РОЗРОБЛЮВАНОЇ АС УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ ПІДПРИЄМСТВА	31
3.1. Розробка і створення концептуальної моделі даних	31
3.2. Розробка ескізів для інтерфейсів користувачів	32
3.3 Вибір програмних засобів.....	33

3.4 Розробка технічного завдання на створення АС	33
3.4.1. Вимоги до системи в цілому.....	34
3.4.2. Вимоги до чисельності та кваліфікації персоналу на АС:	34
3.4.3. Вимоги до експлуатації, технічного обслуговування, ремонту і збереження:	35
3.3.4. Вимоги до функцій (завдань), що виконуються системою.....	35
3.4.5. Вимоги до видів забезпечення.....	35
3.4.6. Організаційне забезпечення:	35
3.4.7. Технічне забезпечення:	36
3.4.8. Програмне забезпечення:.....	36
3.4.9. Математичне забезпечення:.....	36
3.4.10. Метрологічне забезпечення:	36
3.5. Розробка програмного забезпечення для створюваної АС.	37
3.5.1. Створення структури бази даних	37
3.5.2. Планування: створення карти ПО, макетів і шаблонів	40
3.5.3. Створення функціоналу програмного продукту.....	40
3.5.4. Опис реалізованої програмної розробки (інструкція користувача)	41
РОЗДІЛ 4. ОБСТЕЖЕННЯ СИСТЕМИ УПРАВЛІННЯ РЕСТОРАНОМ ДЛЯ АВТОМАТИЗАЦІЇ ЙОГО БІЗНЕС-ПРОЦЕСІВ	43
Аналіз організаційної структури ресторану та створення її моделі.....	43
Виділення бізнес-процесів ресторану та створення його процесної моделі	44
Розробка системи показників і концептуальної схеми управління процесу бронювання столиків	45
Обґрунтування необхідності створення автоматизованої системи управління бронюванням столиків.....	46
4.1. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ІЗ ЗАСТОСУВАННЯМ НОТАЦІЇ VRMN ДЛЯ ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСУ БРОНЮВАННЯ СТОЛИКІВ	48
4.1.1 Створення моделі "як є" для бізнес-процесу бронювання столиків.....	48
4.1.2. Підготовка VRMN-моделі ресторанного бізнес-процесу для виконання у веб- додатку.....	50

4.1.3. Симуляція виконання бізнес-процесу бронювання та аналіз її результатів .	51
4.1.4. Реалізація альтернативних варіантів виконання бізнес-процесу та їх порівняльний аналіз	52
4.2.1 Підготовка обраної моделі для виконання у веб-додатку.....	54
РОЗДІЛ 5. РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ РЕСТОРАНУ.....	56
Технології та інструменти розробки системи	56
Проектування структури бази даних веб-додатку	57
Розробка макетів користувацьких інтерфейсів та дизайну системи.....	62
Створення функціональних модулів програмного продукту	66
РОЗДІЛ 6. РОЗРОБКА SCADA ДЛЯ ТЕХНОЛОГІЧНОГО ОБ'ЄКТУ МОНІТОРИНГУ І УПРАВЛІННЯ.....	71
6.1. Створення моделі системи автоматичного регулювання в програмному середовищі Step7.	71
6.1.1 Створення програми імітації роботи SAP.	72
6.1.2 Виконаємо зв'язок SCADA-системи і проекту SIMATIC Manager.	74
6.3. Реалізація обміну даними між SCADA та БД MySQL.	79
6.3.1. Організація обміну даними між SCADA-системою WINCCflexible і БД MySQL/MariaDB.	79
6.3.2. Створимо в нашому веб-додатку сторінку для відображення параметрів ТП.	82
РОЗДІЛ 7. ВИБІР ТЕХНІЧНИХ ЗАСОБІВ, МЕТА ВИКОРИСТАННЯ І РЕАЛІЗАЦІЯ ЇХ ЗВ'ЯЗКУ ІЗ СТВОРЕНИМ ВЕБ-ДОДАТКОМ.....	84
7.1 Вибір технічних засобів.....	84
7.2 Мета використання кожного засобу.....	84
7.3 Реалізація зв'язку технічних засобів.....	85
7.4 Результати і переваги інтеграції.....	86
РОЗДІЛ 8. ВИБІР І ЗАСТОСУВАННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ (ІАД) ДЛЯ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ	87
8.1. Значення ІАД у сфері ресторанного бізнесу.....	87

8.2. Джерела даних для ІАД у ресторані.....	87
8.3. Вибір методів інтелектуального аналізу даних	88
8.4. Приклад використання: аналіз продажів і оптимізація меню.....	89
8.5. Інструменти для реалізації ІАД.....	89
8.6. Висновки.....	90
РОЗДІЛ 9. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	91
9.1 Загальні положення.....	91
9.2 Очікувані вигоди від впровадження	91
9.3 Вихідні дані для розрахунків.....	92
9.4 Розрахунок витрат на створення ПЗ.....	92
9.5 Розрахунок економічного ефекту.....	93
9.6 Розрахунок показників ефективності.....	94
ВИСНОВКИ	95
СПИСОК ЛІТЕРАТУРИ.....	97

ВСТУП

У сучасних умовах ведення бізнесу сфера громадського харчування зазнає стрімких змін, обумовлених підвищенням вимог споживачів до якості обслуговування, комфорту та загального рівня сервісу. Ресторани вже давно перестали бути лише місцем для прийому їжі — вони трансформувалися у багатофункціональні простори для відпочинку, ділових зустрічей та святкувань. У такому середовищі успіх підприємства визначається не лише кухнею, а й ефективністю внутрішніх процесів, затишною атмосферою та здатністю швидко адаптуватися до потреб клієнтів.

Ресторан «Хмари», розташований в Одесі, є сучасним закладом із унікальним підходом до обслуговування, дизайну інтер'єру та організації простору. Його концепція поєднує естетичну привабливість, високий рівень клієнтського сервісу та комфортні умови для перебування. Однією з ключових складових комфортної атмосфери в приміщенні є правильно організований мікроклімат — температурні умови, вологість, вентиляція та кондиціювання повітря.

Забезпечення належного мікроклімату у закладах громадського харчування має суттєве значення для здоров'я персоналу та відвідувачів, збереження інтер'єру, кухонного обладнання, а також для створення позитивного іміджу ресторану. Раціональне використання кліматичної техніки дає можливість оптимізувати витрати на енергоресурси, зменшити зношення обладнання та забезпечити стабільні умови для приготування і подачі страв.

У межах цієї кваліфікаційної роботи розглядається діяльність ресторану «Хмари» як об'єкта дослідження з точки зору організації бізнес-процесів, управління інфраструктурою приміщення, зокрема питання підтримки мікроклімату. Метою роботи є аналіз діючих механізмів у ресторані, виявлення недоліків та обґрунтування шляхів покращення управління внутрішніми процесами задля підвищення ефективності функціонування підприємства.

Загальний підхід у роботі передбачає використання елементів системного аналізу, оцінку економічної доцільності окремих рішень, а також формування

рекомендацій для подальшого розвитку інфраструктури ресторану відповідно до сучасних вимог та стандартів галузі HoReCa.

РОЗДІЛ 1. ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПІДПРИЄМСТВА ТА БІЗНЕС-ПРОЦЕСУ «РЕСТОРАННИЙ КЕЙТЕРИНГ», ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ШЛЯХІВ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЙОГО УПРАВЛІННЯ

1.1. Характеристика підприємства ресторану «Хмари» (м. Одеса) та його основних видів діяльності

Об'єктом дослідження у даній роботі виступає панорамний ресторан преміум-класу «Хмари», який розташований у центрі міста Одеса (вул. Катерининська, 27/1, 9-й поверх ТЦ «Kadort»). Заклад позиціонується як багатофункціональний ресторанний комплекс, що поєднує три концептуальні гастрономічні напрямки (українська, італійська та паназійська кухні), а також має відкриту панорамну терасу.

Основними видами економічної діяльності підприємства відповідно до КВЕД є:

- **56.10** — Діяльність ресторанів, надання послуг мобільного харчування;
- **56.21** — Постачання готових страв для подій (кейтеринг);
- **56.30** — Обслуговування напоями.

Організаційно-правова структура підприємства побудована за лінійно-функціональним принципом. Найвищою ланкою управління є директор (керівник), якому підпорядковуються шеф-кухар, шеф-бармен, фінансовий відділ, а також менеджер з маркетингу та шеф-сомельє. Безпосереднє оперативне управління залами та персоналом (офіціанти, хостес) здійснюють старші адміністратори.

1.2. Опис бізнес-процесу ресторанного кейтерингу та поточної схеми обробки заявок на бронювання місць

Важливим високодохідним напрямком діяльності ресторану є **ресторанний кейтеринг** — виїзне обслуговування офіційних заходів, банкетів, фуршетів та приватних свят на локаціях замовника в межах Одеси та Одеської області.

На сьогоднішній день інформаційна схема та бізнес-процес обробки заявок (як на бронювання столиків у панорамній залі, так і на виїзне кейтерингове обслуговування) базується на класичній **телефонній комунікації**.

Поточний алгоритм «як є» виглядає так:

1. Клієнт телефонує за контактним номером закладу.
2. Хостес або адміністратор приймає дзвінок, усно з'ясовує побажання (дата, час, кількість гостей, зона посадки або формат кейтерингу).
3. Адміністратор вручну перевіряє наявність вільних місць або доступність персоналу для виїзду на кейтеринг на вказану дату.
4. Дані фіксуються в журналі або локальній системі обліку.
5. Якщо потрібне узгодження меню, адміністратор передає інформацію менеджеру, який повторно зв'язується з клієнтом по телефону чи в месенджерах.

1.3. Опис існуючого програмного забезпечення для автоматизації обліку та підтримки інформаційних потоків закладу.

Для забезпечення поточної життєдіяльності та ведення фінансово-господарського обліку в ресторані «Хмари» використовується спеціалізоване програмне забезпечення автоматизації ресторанного бізнесу (наприклад, система *Poster POS* або *r-keeper*).

Зазначене програмне забезпечення успішно вирішує внутрішні локальні задачі, серед яких:

- Касовий облік та фіксація продажів у реальному часі (фронт-офіс);
- Складський облік, калькуляція страв, списання інгредієнтів та інвентаризація (бек-офіс);

- Моніторинг фінансових показників та формування звітів для керівництва;
- Контроль робочого часу персоналу кухні та залу.

Проте, наявний комплекс програмного забезпечення орієнтований виключно на **внутрішній облік**. Він є закритим для кінцевого споживача і не має інтегрованого зовнішнього веб-інтерфейсу для самостійної взаємодії клієнтів із закладом через мережу Інтернет.

1.4. Виявлення і формалізація основних проблем життєдіяльності підприємства та недоліків телефонного резервування столиків.

Проведений аналіз поточної схеми взаємодії з клієнтами дозволив виявити ряд критичних недоліків та «вузьких місць» в управлінні бізнес-процесом резервування:

1. **Високий рівень впливу «людського фактору»:** під час пікових навантажень (п'ятниця-неділя, вечірні години) адміністратор може припуститися помилки під час ручного внесення даних (переплутати номер столика, дату, або неправильно записати контакти клієнта).

2. **Втрата потенційних замовлень:** якщо телефонна лінія зайнята або дзвінок надходить у неробочий/нічний час, заклад втрачає клієнта, який хотів забронювати стіл або залишити заявку на кейтеринг.

3. **Відсутність візуалізації для клієнта:** оскільки ресторан «Хмари» є панорамним закладом, для гостей критично важливо обрати конкретне місце (наприклад, біля вікна з видом на місто). При телефонній розмові клієнт змушений вірити адміністратору «на слово», не бачачи інтерактивної схеми розташування столів у залі.

4. **Низька ефективність обробки кейтеринг-заявок:** обговорення меню та локації по телефону займає багато часу та вимагає численних повторних дзвінків.

1.5. Обґрунтування доцільності впровадження автоматизованої системи онлайн-бронювання на сайті як шляху підвищення ефективності управління

Для усунення виявлених проблем постає гостра науково-практична доцільність розробки та впровадження інформаційної підсистеми автоматизованого онлайн-бронювання місць та заявок на кейтеринг безпосередньо на веб-сайті ресторану.

Впровадження даної системи «як має бути» дозволить:

- Перевести процес збирання заявок у режим 24/7, повністю виключивши проблему «зайнятої лінії» та мінімізувавши навантаження на хостес.
- Надати клієнту інтерактивний інтерфейс для самостійного вибору зали, конкретного столика на карті та зручного часу візиту.
- Автоматизувати первинний збір параметрів для виїзного кейтерингу (дата, кількість персон, бажаний тип кухні) через зручну веб-форму.
- Забезпечити прозорість процесу та автоматичне підтвердження броні за допомогою SMS або Email-сповіщень, що значно підвищить лояльність клієнтів та економічну ефективність управління бізнес-процесами ресторану.

РОЗДІЛ 2. ОБСТЕЖЕННЯ ПІДПРИЄМСТВА ЯК ОБ'ЄКТА АВТОМАТИЗАЦІЇ ТА СТВОРЕННЯ КОМПЛЕКСУ МОДЕЛЕЙ ЙОГО БІЗНЕС-ПРОЦЕСІВ.

2.1. Аналіз організаційної структури підприємства та створення її моделі.



Рис. 2.1 – зображення ресторану «Хмари» зі супутнику на карті

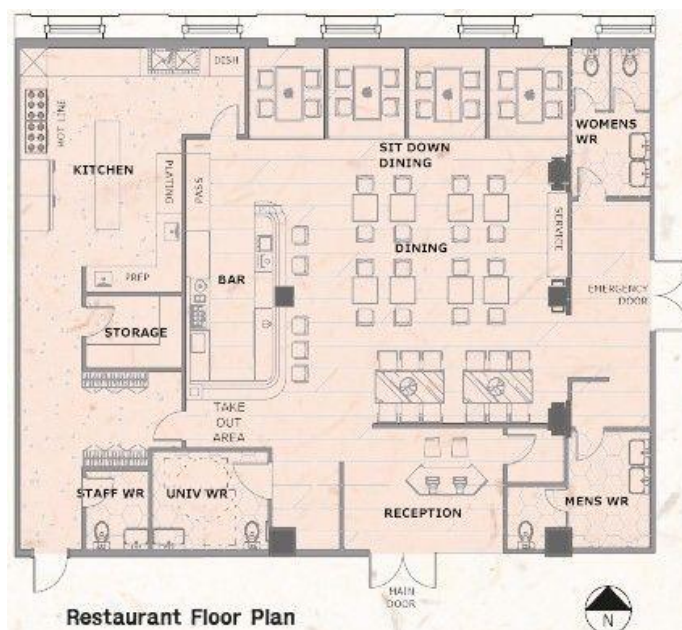


Рис. 2.2 Ескізний план ресторану «Хмари».

Результатом вивчення організаційної структури підприємства має стати обов'язкове її представлення у вигляді схеми (структурної моделі діяльності підприємства). Цю задачу доцільно виконати з використанням будь-якого програмного інструменту. В якості такого інструменту може бути використана програма ARIS Express

Основним результатом виконання лабораторної роботи має стати організаційна структура досліджуваного підприємства.

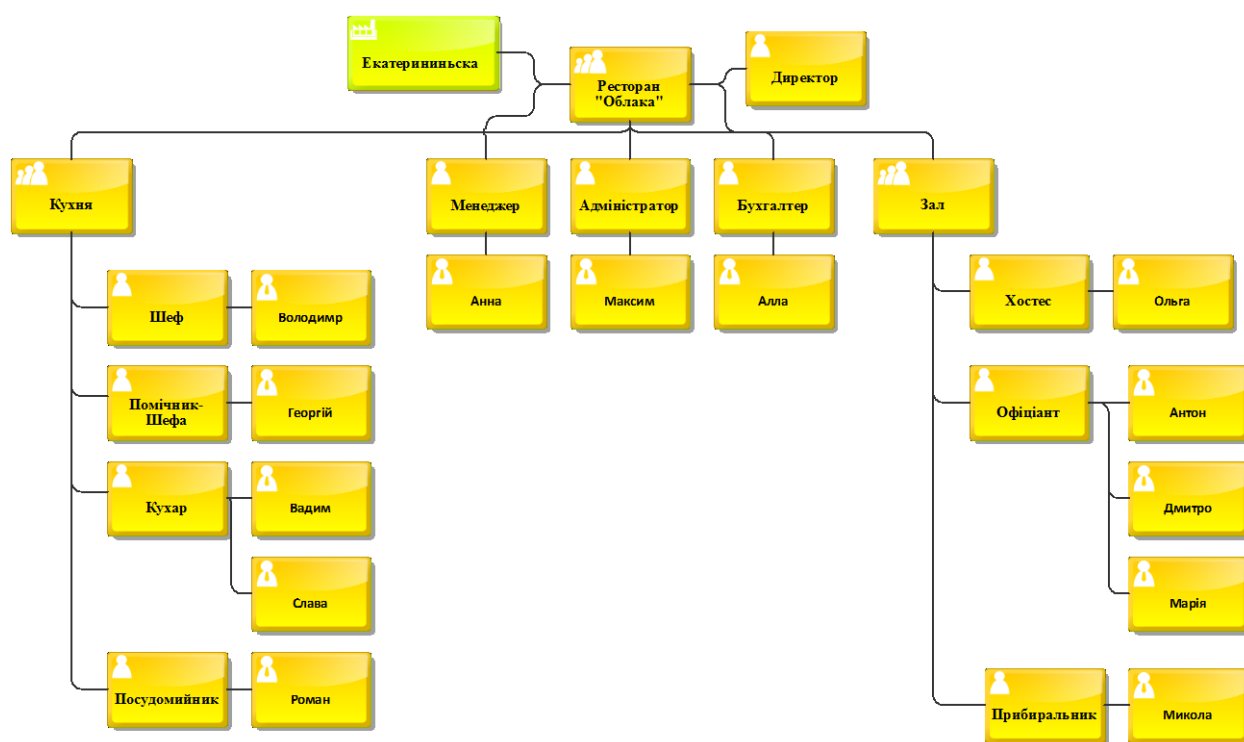


Рис. 2.3 – організаційна структура досліджуваного підприємства.

2.2. Виділення бізнес-процесів підприємства та створення його процесної моделі.

До **основних** бізнес-процесів можна віднести ті, в ході яких створюється основний потік доходів.

Основні (операційні) процеси:

- Приготування та подача страв.
- Обслуговування клієнтів у залі.
- Управління запасами і закупівлі продуктів.

- Контроль якості приготування страв.

До **підтримуючих** бізнес-процесів можна віднести такі, які забезпечують безперебійне протікання операційних процесів.

Підтримуючі (допоміжні) процеси:

- Бухгалтерія та фінансове управління.
- Управління персоналом та навчання.
- Технічне обслуговування кухонного обладнання.
- Забезпечення санітарно-гігієнічних стандартів.

До **розвиваючих** бізнес-процесів можна віднести такі, які керують функціонуванням системи.

Розвиваючі (процеси управління):

- Розробка та вдосконалення меню.
- Маркетинг та реклама ресторану.
- Планування та аналіз фінансових показників.
- Управління якістю обслуговування.

Враховуючи ці розділення, ресторан може ефективно управляти своєю діяльністю, забезпечуючи високу якість страв, задоволення клієнтів та стабільність у фінансових показниках.

Бізнес-процес «Ресторанний Кейтерінг»

Бізнес-процес може складатися з наступних підпроцесів:

- **Взаємодія з клієнтом;**
- **Приготування їжі;**
- **Закупівля сировини.**

Опис входів та виходів для бізнес-процесу **«Ресторанний Кейтерінг»** наведемо в таблиці 2.1.

Для підпроцесу “Взаємодія з клієнтом”

Назва підпроцесу	Вхід	Вихід
<i>Бронювання</i>	Заявка від клієнта	Заброньоване місце
<i>Розміщення та вибір страв</i>	Заброньоване місце	Прийнятий заказ від клієнта
<i>Обслуговування</i>	Прийнятий заказ від клієнта	Прибираний за клієнтом стіл
<i>Розрахунок</i>	Бажання клієнта	Видача чека

Для підпроцесу “Приготування їжі ”

Назва підпроцесу	Вхід	Вихід
<i>Отримання заказ</i>	Заказ від клієнта	Завдання для кухні
<i>Підготовка інгредієнтів</i>	Завдання для кухні	Інгредієнти для приготування
<i>Приготування страв</i>	Інгредієнти для приготування	Готова страва

Для підпроцесу “Закупівля сировини ”

Назва підпроцесу	Вхід	Вихід
<i>Взаємодія з постачальником</i>	Замовлення сировини	Документи від постачальника
<i>Приймання сировини на склад</i>	Документи від постачальника	Сировина на складі
<i>Облік сировини на складі</i>	Сировина на складі	Відомість остатків сировини

Таблиця 2.1.

Процесна модель — це модель, що описує процес послідовного перетворення матеріальних, сировинних та інформаційних потоків підприємства. Процесна модель може бути отримана за допомогою нотації **Process landscape** програмного інструменту **Aris Express**.

Створив та навів модель процесної структури з використанням програмного інструменту **ARIS**.



Рис. 2.4. Бізнес-процес «Ресторанний кейтерінг»

Висновок: на цій лабораторній роботі я вивчив аналіз ієрархічних процесів.

2.3. Моделювання бізнес-процесів підприємства в нотації eEPC.

Як приклади розглянемо процеси «Взаємодія з клієнтами», «Приготування їжі», «Закупівля сировини». При визначенні їх функцій слід пам'ятати, що вони насправді реалізуються в конкретних підрозділах розглянутих підприємств і повинні мати точно таку послідовність, як в реальності. Для

кожного процесу існує свій набір функцій.

Функції, які виконуються в процесі **«Взаємодія з клієнтами»**:

- Прийом та вітання клієнтів;
- Бронювання столиків;
- Обслуговування;
- Прийом та обробка замовлень;
- Подача страв і напоїв;
- Збір відгуків і скарг;
- Обробка платежів;
- Підтримка програм лояльності;
- Прощання та подяки;

Функції, які виконуються в процесі **«Приготування їжі»**:

- Замовлення і закупівля інгредієнтів;
- Приготування страв;
- Ведення кулінарної документації;
- Контролювання якості;
- Обробка замовлень;
- Координація роботи кухні;
- Співпраця з обслуговуючим персоналом;
- Контролювання запасів;
- Впорядкування робочого місця;

Функції, які виконуються в процесі **«Закупівля сировини»**:

- Планування закупівель;
- Взаємодія з постачальником;
- Замовлення і поставки;
- Інвентаризація та зберігання;
- Оптимізація витрат;
- Управління обліковою інформацією;

Подія - це деякий стан, який є необхідною умовою для початку і закінчення виконання функції. При визначенні подій важливо пам'ятати, що подія миттєва у часі. Для кожного процесу існує свій набір подій

Подіями процесу «**Взаємодія з клієнтами**», що відображають виконання функцій, є:

- Отримання заявки в месенджері або по телефону
- Перевірка бронювання
- Пропозиція про інше місце
- Місце заброньоване
- Реєстрація заявки

Подіями процесу «**Закупівля сировини**», що відображають виконання функцій, є:

- Відправлена заява на сировину;
- прийшли документи на плановану доставку сировини;
- сировину доставлено;
- сировину розвантажено;
- сировину розміщено;
- облік сировини оновлен.

Подіями процесу «**Приготування їжі**», що відображають виконання функцій, є:

- замовлення страв;
- підготовка інгредієнтів;
- готування страв;
- падача страви до віконця офіціанта

Рекомендації розташування графічних елементів на діаграмі eEPC:

- графічні елементи процесу (послідовність подій і функцій) розташовуються зверху вниз;
- графічні елементи, що позначають виконавців функцій або осіб, зацікавлених у виконанні функцій (співробітників або підрозділи), розташовуються вгорі і праворуч від функцій;
- документи, прикладна система, що використовуються при виконанні функцій, розташовуються зліва від функцій;
- документи, що формуються в результаті виконання функцій, розташовуються праворуч від функцій або зліва під документами, що використовуються при виконанні функцій.

Основним результатом виконання лабораторної роботи повинні стати моделі обраних процесів в нотації ARIS eEPC.

Отже наведемо спочатку уявлення (модель) процесу **«Взаємодія з клієнтами»**.



2.4. Розробка системи показників і концептуальної схеми управління обраного бізнес-процесу.

Система ключових показників ефективності - сукупність показників для вимірювання та аналізу рівня і динаміки ефективності виробництва.

Система показників ефективності процесів та організації складається з трьох основних потоків інформації:

- інформація про якість процесу, його ефективність і ресурсоемність;
- інформація про якість продукції або послуг;
- інформація про ступінь задоволеності покупця.

Уявімо більш докладно **виробничі показники** діяльності підприємства. Для прикладу розглянемо підпроцес «**Закупівля сировини**» і опишемо для нього найголовніші **показники** процесу і продукту по кожному з підпроцесів в загальному вигляді

• Взаємодія з постачальником (1):

- Термін поставки (показник продукту);
- Ціни від постачальників (показник процесу);
- Обсяг поставок (показник продукту);
- Стабільність постачальника (показник продукту);
- Кількість проблем з постачальником у місяць (показник продукту).

• Приймання сировини на склад (2):

- Оцінка якості прийнятої сировини (показник процесу);
- Кількість прийнятої сировини (показник процесу);
- Час приймання сировини (показник процесу);
- Термінами придатності сировини (показник продукту).

• Облік сировини на складі (3):

- Кількість запасів (показник процесу);
- Термін зберігання сировини (показник продукту);

- Час проведення обліку (показник процесу);
- Кількість виданої сировини (показник продукту).

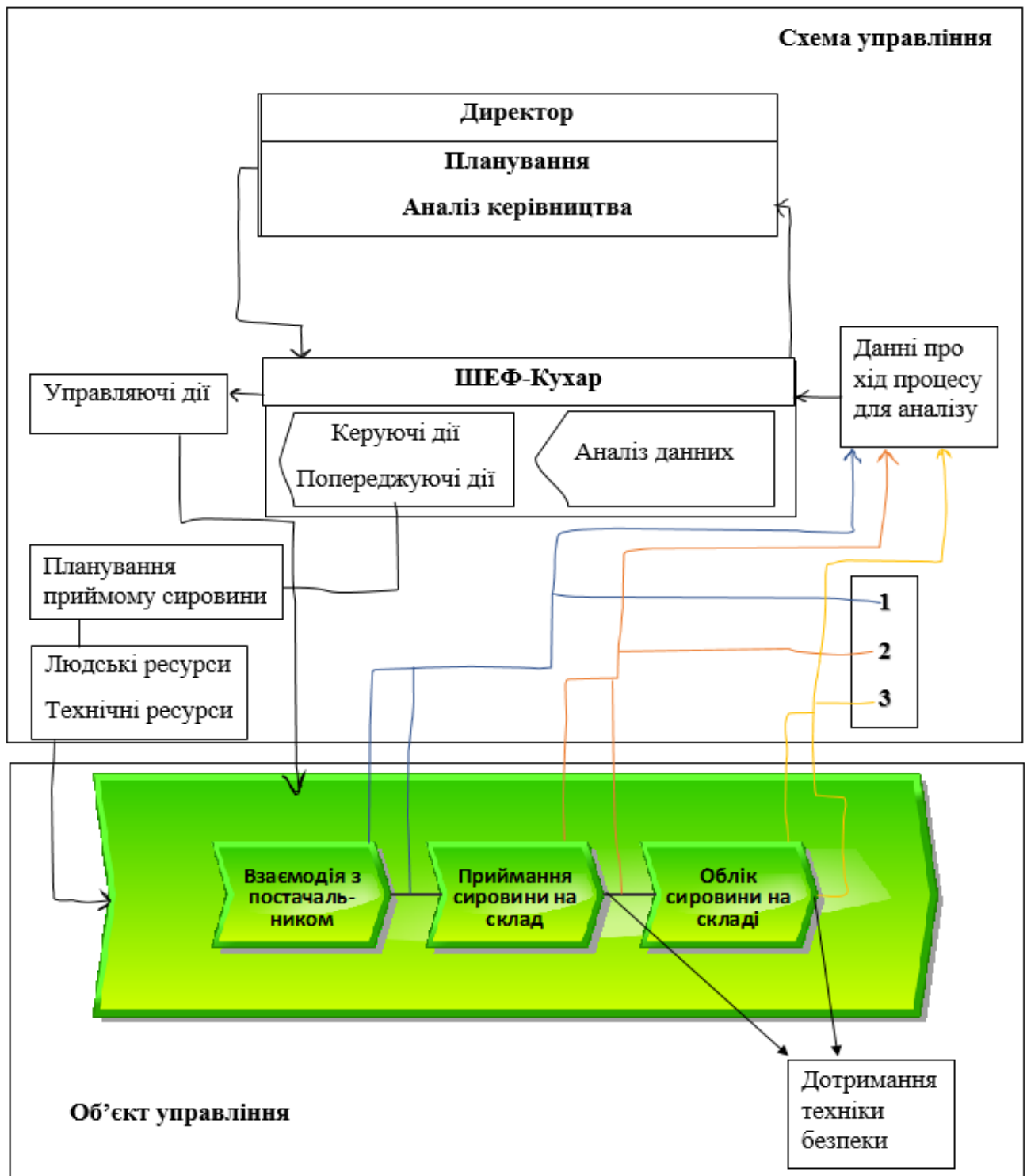


Рис. 2.6- Концептуальна схема системи управління бізнес-процесом «Закупівля сировини».

Висновок: на цій лабораторній роботі я визначив показники бізнес-процесу та визначив методики управління організацією на основі процесного підходу.

2.5. Обґрунтування необхідності створення (розвитку) автоматизованої системи (АС) управління бізнес-процесами підприємства (постановка завдання).

Успішне функціонування ресторану в умовах сучасного ринку потребує високої ефективності управління бізнес-процесами. Створення автоматизованої системи (АС) управління дозволяє оптимізувати роботу підприємства, знижуючи витрати та підвищуючи якість обслуговування клієнтів. Автоматизація бізнес-процесів у ресторані забезпечує такі переваги:

Підвищення якості обслуговування клієнтів: Автоматизовані системи дозволяють швидко обробляти замовлення, зменшувати час очікування та мінімізувати помилки в обслуговуванні. Це безпосередньо впливає на задоволеність клієнтів та їх бажання повертатися до ресторану.

Ефективне управління запасами: Автоматизація процесів дозволяє точно відстежувати наявність сировини, попереджувати її нестачу та оптимізувати закупівлі. Це допомагає знижувати витрати та уникати простоїв у роботі.

Поліпшення умов праці персоналу: Системи автоматизації зменшують навантаження на персонал, автоматизуючи рутинні та повторювані завдання. Це дозволяє співробітникам зосередитися на більш важливих аспектах роботи та підвищує їх задоволеність працею.

Підвищення експлуатаційних показників обладнання: За допомогою автоматизованих систем можна здійснювати моніторинг та управління роботою обладнання, забезпечуючи його оптимальну експлуатацію та своєчасне технічне обслуговування. Це сприяє зниженню витрат на ремонт та збільшує термін служби обладнання.

Таким чином, впровадження автоматизованої системи управління бізнес-процесами є необхідним кроком для забезпечення конкурентоспроможності та сталого розвитку ресторану.

МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСУ ПІДПРИЄМСТВА В НОТАЦІЇ

При визначенні учасників слід пам'ятати, що його дії безпосередньо впливають на хід процесу і є невід'ємною частиною. У нашому прикладі виберемо наступних учасників: відділ замовлення сировини, начальник складу, бригадир, вантажники

1. Визначити завдання кожного учасника, а так само його взаємозв'язок з іншими учасниками даного бізнес-процесу.

Для того, щоб визначити завдання кожного учасника необхідно чітко створити картину виконання даного процесу від початку до кінця, з урахуванням всіх виникаючих нюансів і умов. Тобто знати, що відбувається при виконанні або невиконанні будь-якої умови, які подальші дії виконуються, яким чином вони пов'язані з іншими учасниками.

2. Використовуючи нотацію BPMN, промодельуйте розглянутий бізнес-процес.

BPMN - це графічна нотація для моделювання бізнес-процесів.

BPMN була розроблена Business Process Management Initiative (BPMI) і підтримується Object Management Group. Основна мета BPMN- створення стандартного набору умовних позначень, зрозумілих усім бізнес-користувачам. Бізнескористувачі включають в себе бізнес-аналітиків, які створюють і поліпшують процеси, технічних розробників відповідальних за реалізацію процесів та менеджерів, що стежать за процесами і керуючих ними. Для опису подій в методології BPMN можна використовувати об'єкт "ко- 37 ло", різні форми

якого показують початкові, кінцеві та проміжні події. Об'єктом "ромб" в методології BPMN описуються логічні умови, а прямокутником із закругленими кутами описуються функції бізнес-процесів. У доповненні до цього, для опису інформаційних потоків використовується об'єкт, що позначає документ.

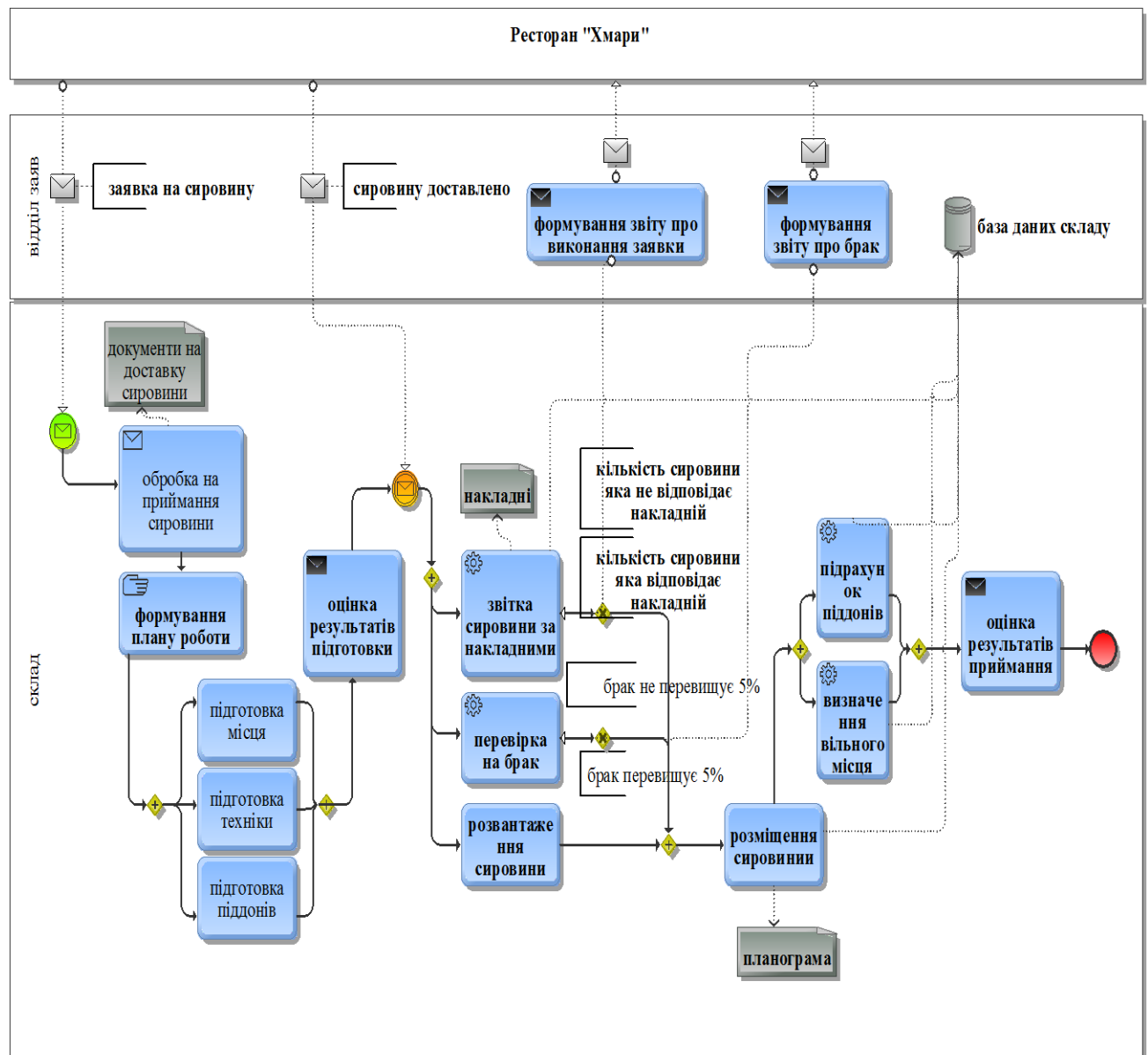


Рис. 2.7.- Модель (діаграма) підпроцесу "Приймання сировини на склад" в нотації ARIS BPMN.

РОЗРОБКА МОДЕЛІ ДАНИХ В ARIS EXPRESS

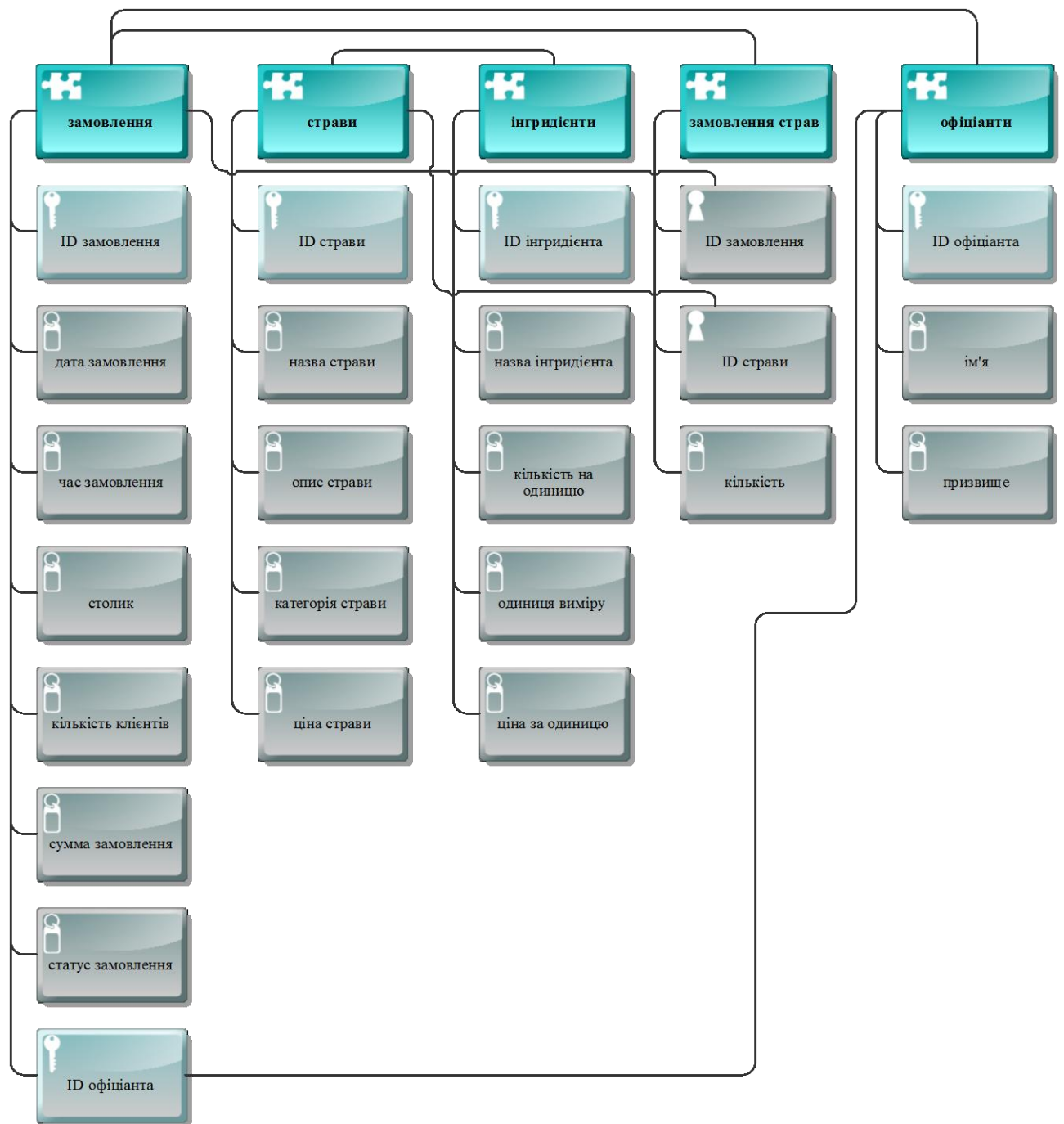


Рис. 2.8.- Повна модель даних, що відображає взаємозв'язок між таблицями.

РОЗДІЛ 3. СТВОРЕННЯ КОНЦЕПТУ ПРОГРАМИ ЯК ОСНОВИ РОЗРОБЛЮВАНОЇ АС УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ ПІДПРИЄМСТВА

3.1. Розробка і створення концептуальної моделі даних

3.1.1. Сутність

- **Тип сутності:** Стрижнева
- **Екземпляр сутності:** Страва, Столик, Замовлення, Замовлення-

Страва, Персонал

- **Класифікація:**
 - **Стрижневі:** Страва, Столик, Замовлення
 - **Асоціативні:** Замовлення-Страва
 - **Характеристичні:** Персонал

3.1.2. Атрибут

- **Класифікація:**
 - **Описові:** Назва страви, Ціна страви, Місце за столиком, Дата замовлення, Час замовлення, Ім'я працівника, Посада працівника
 - **Вказівні:** Ідентифікатор страви, Ідентифікатор столика, Ідентифікатор замовлення, Ідентифікатор працівника
 - **Допоміжні:** Фото страви, Опис страви, Статус замовлення (прийнято, готується, подано, оплачено)

3.1.3. Зв'язок

- **Характеристика:**
 - **Бінарність:** Бінарний
 - **Кардинальність:**
 - **Один до багатьох (1:N):** Замовлення-Страва (Замовлення може мати багато страв, Страва може бути включена до багатьох замовлень)
 - **Один до одного (1:1):** Столик-Замовлення (Столик може мати лише одне замовлення, Замовлення може бути розміщене лише за одним столиком)

- Багато до багатьох (M:N): Персонал-Замовлення (Один працівник може обслуговувати багато замовлень, одне замовлення може обслуговуватися багатьма працівниками)

- **Модальність:**

- **Обов'язковий:** Ідентифікатор страви, Ідентифікатор столика, Ідентифікатор замовлення, Ідентифікатор працівника

- **Необов'язковий:** Фото страви, Опис страви, Статус замовлення

3.2. Розробка ескізів для інтерфейсів користувачів

Можу описати можливі інтерфейси користувачів для вашої АС управління рестораном:

Головний інтерфейс:

- Показує план залу ресторану з вільними та зайнятими столиками.
- Дозволяє користувачеві вибрати столик, щоб створити нове замовлення.
- Містить кнопки для доступу до інших функцій, таких як меню, каси, налаштування.

Меню:

- Показує список доступних страв з категоріями (наприклад, закуски, салати, основні страви, десерти).
- Кожна страва має зображення, опис, ціну.
- Користувач може додати страви до замовлення, вибравши їх з меню.

Замовлення:

- Показує список страв, включених до замовлення, з кількістю кожної страви, ціною та загальною сумою.
- Користувач може змінити кількість страв, видалити їх з замовлення.
- Містить кнопку для оформлення замовлення.

Каса:

- Показує загальну суму замовлення.
- Користувач може вибрати спосіб оплати (готівка, картка).
- Після оплати замовлення друкується чек.

Налаштування:

- Дозволяє адміністратору ресторану додавати та видаляти страви з меню, змінювати ціни, налаштовувати параметри системи.

3.3 Вибір програмних засобів

Готові рішення:

- існують готові програмні продукти, розроблені спеціально для ресторанів.
- Ці продукти зазвичай прості у використанні та не потребують глибоких знань програмування.
- Деякі популярні готові рішення: Poster, iiko, r_keeper, Тилли, 1С:Ресторан.

Розробка на замовлення:

- Якщо вам потрібна система з унікальними функціями або інтеграцією з іншими системами, ви можете замовити розробку АС управління рестораном у професійної компанії.
- Цей варіант дає вам більше гнучкості та контролю над процесом розробки, але може бути дорожчим.

Відкриті програмні рішення:

- Існують також безкоштовні та відкриті програмні продукти, які можна використовувати для управління рестораном.
- Ці продукти зазвичай потребують більше знань програмування для налаштування та використання.
- Деякі популярні відкриті програмні рішення: OpenResty, SambaPOS, FreedomPOS.

3.4 Розробка технічного завдання на створення АС

1. Повне найменування системи та її умовне позначення:

- Система управління рестораном "Restoran"

2. Найменування підприємств (об'єднань) розробника і замовника (користувача) системи та їх реквізити:

- Розробник: [Назва вашої компанії/організації], [Контактна

інформація]

- Замовник: [Назва ресторану], [Контактна інформація]

3. Перелік документів, на підставі яких створюється система, ким і коли затверджені ці документи:

- Технічне завдання на створення АС управління рестораном "Restoran" (затверджено [дата] директором ресторану [ПІБ])

4. Призначення системи:

- Автоматизація процесів управління рестораном, включаючи:
 - Прийняття замовлень
 - Ведення обліку страв та запасів
 - Розрахунок замовлень
 - Управління персоналом
 - Збір статистичних даних

5. Цілі створення системи:

- Збільшення ефективності роботи ресторану
- Покращення якості обслуговування клієнтів
- Зниження витрат
- Збільшення прибутку

3.4.1. Вимоги до системи в цілому

Вимоги до структури та функціонування системи:

- Система повинна бути модульною, що дозволяє легко додавати нові функції та масштабувати систему.
- Система повинна бути гнучкою та налаштовуваною, щоб відповідати потребам конкретного ресторану.
- Система повинна бути інтегрованою з іншими системами ресторану, такими як касова система, система управління запасами, система лояльності.

3.4.2. Вимоги до чисельності та кваліфікації персоналу на АС:

- Для роботи з системою буде потрібно 2-3 адміністратора системи, які мають знання комп'ютерних технологій та досвід роботи з ресторанными системами.

- Персонал ресторану (офіціанти, кухарі, бармени) буде використовувати систему для прийняття замовлень, обліку страв та запасів, розрахунку замовлень.

3.4.3. Вимоги до експлуатації, технічного обслуговування, ремонту і збереження:

- Система повинна бути простою у використанні та не потребувати глибоких знань комп'ютерних технологій від персоналу ресторану.

- Система повинна бути надійною та стійкою до збоїв.
- Система повинна легко оновлюватися та модернізуватися.

3.3.4. Вимоги до функцій (завдань), що виконуються системою

- **Прийняття замовлень:**

- Система повинна дозволяти офіціантам приймати замовлення від клієнтів.

- Система повинна відображати план залу ресторану з вільними та зайнятими столиками.

- Система повинна дозволяти офіціантам додавати страви до замовлення, змінювати їх кількість, видаляти страви з замовлення.

- Система повинна розраховувати загальну суму замовлення.

- Система повинна дозволяти друкувати чеки.

3.4.5. Вимоги до видів забезпечення

Інформаційне забезпечення:

- Система повинна використовувати базу даних для зберігання інформації про страви, замовлення, клієнтів, персонал.

- Система повинна забезпечувати безпеку даних.
- Система повинна бути сумісною з іншими системами ресторану.

3.4.6. Організаційне забезпечення:

- Для роботи з системою буде створено новий підрозділ - відділ інформаційних технологій.

- У відділі інформаційних технологій буде працювати 2-3 адміністратора системи.

- Персонал ресторану буде пройде навчання з роботи з системою.

3.4.7. Технічне забезпечення:

- Система повинна працювати на персональних комп'ютерах або планшетах.
- Для роботи системи буде потрібно сервер для зберігання бази даних.
- Система повинна мати принтер для друку чеків.

3.4.8. Програмне забезпечення:

- Система повинна бути розроблена на мові програмування [назва мови програмування].
- Система повинна використовувати операційну систему [назва операційної системи].
- Система повинна використовувати базу даних [назва бази даних].

3.4.9. Математичне забезпечення:

- Система повинна використовувати математичні методи для:
 - **Розрахунку загальної суми замовлення:**
 - $\text{Ціна страви} * \text{Кількість страви} = \text{Сума страви}$
 - $\text{Сума страви 1} + \text{Сума страви 2} + \dots + \text{Сума страви N} = \text{Загальна сума замовлення}$
 - **Надання знижок:**
 - Фіксована знижка: $\text{Загальна сума замовлення} * (1 - \text{Відсоток знижки}) = \text{Сума з знижкою}$
 - Знижка на певну суму: $\text{Загальна сума замовлення} - \text{Сума знижки} = \text{Сума з знижкою}$
 - **Ведення статистики:**
 - Кількість замовлень за день/тиждень/місяць
 - Середня сума замовлення
 - Найпопулярніші страви
 - Найменш популярні страви

3.4.10. Метрологічне забезпечення:

- Система повинна відповідати вимогам державних стандартів до

вимірювальних приладів, які використовуються в ресторані (наприклад, ваги, касові апарати).

- Система повинна регулярно проходити метрологічну перевірку цих приладів.

3.5. Розробка програмного забезпечення для створюваної АС.

3.5.1. Створення структури бази даних

1. Таблиця "Користувачі"

Поле	Тип даних	Опис
ID_користувача	INT	Первинний ключ, унікальний ідентифікатор користувача
Ім'я	VARCHAR(255)	Ім'я користувача
Прізвище	VARCHAR(255)	Прізвище користувача
Логін	VARCHAR(255)	Логін для входу в систему
Пароль	VARCHAR(255)	Зашифрований пароль користувача
Роль	ENUM('гість', 'адміністратор')	Роль користувача: 'гість' або 'адміністратор'

2. Таблиця "Столики"

Поле	Тип даних	Опис
ID_столика	INT	Первинний ключ, унікальний ідентифікатор столика
Кількість місць	INT	Кількість місць за столиком
Розташування	VARCHAR(255)	Розташування столика (наприклад, "зал 1", "тераса")
Статус	ENUM('вільний', 'зайнятий')	Статус столика: 'вільний' або 'зайнятий'

3. Таблиця "Бронювання"

Поле	Тип даних	Опис
ID_бронювання	INT	Первинний ключ, унікальний ідентифікатор бронювання
ID_користувача	INT	Зовнішній ключ, що посилається на таблицю "Користувачі"
ID_столика	INT	Зовнішній ключ, що посилається на таблицю "Столики"
Дата_час_бронювання	DATETIME	Дата і час бронювання
Кількість_людей	INT	Кількість людей, які збираються забронювати столик

4. Таблиця "Складники"

Поле	Тип даних	Опис
ID_складової	INT	Первинний ключ, унікальний ідентифікатор інгредієнта
Назва	VARCHAR(255)	Назва інгредієнта
Кількість на складі	INT	Кількість інгредієнта на складі
Ціна за одиницю	DECIMAL(10,2)	Ціна за одиницю інгредієнта

5. Таблиця "Замовлення_інгредієнтів"

Поле	Тип даних	Опис
ID_замовлення	INT	Первинний ключ, унікальний ідентифікатор замовлення
ID_складової	INT	Зовнішній ключ, що посилається на таблицю "Складники"
Кількість	INT	Кількість замовлених інгредієнтів
Дата_замовлення	DATETIME	Дата замовлення інгредієнтів

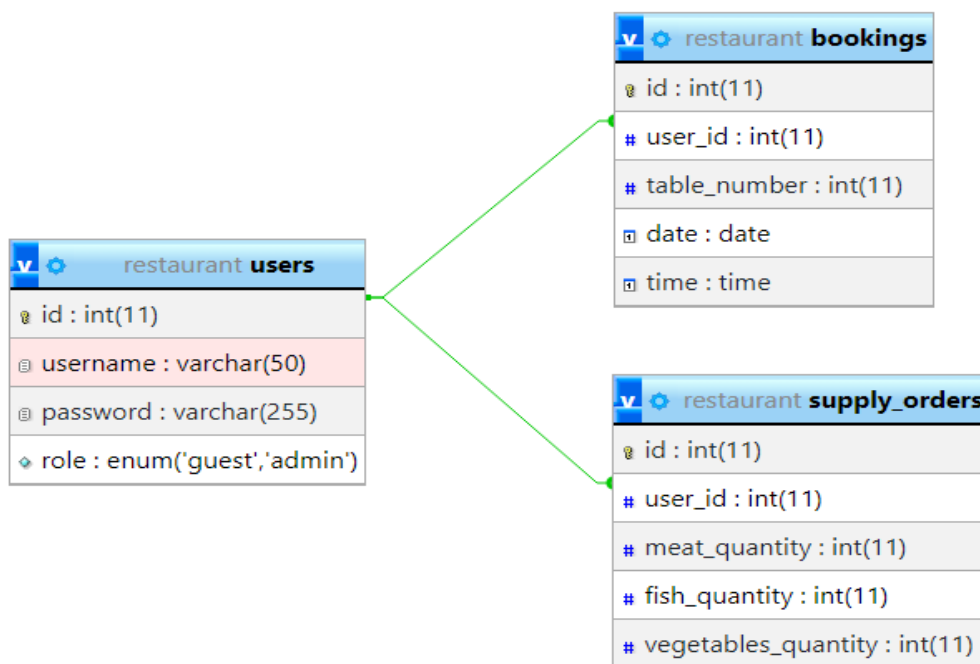


Рис. 3.5.1. Структура бази даних «restaurant».

Таблиця	Действие	Строки	Тип	Сравнение	Размер	Фрагментировано
☐ bookings	☆ [иконки]	0	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
☐ supply_orders	☆ [иконки]	0	InnoDB	utf8mb4_unicode_ci	32.0 КиБ	-
☐ users	☆ [иконки]	0	InnoDB	utf8mb4_unicode_ci	16.0 КиБ	-
3 таблицы	Всего	0	InnoDB	utf8mb4_unicode_ci	80.0 КиБ	0 Байт

Рис. 3.5.2. Структура таблиці «RESTAURANT»

3.5.2. Планування: створення карти ПО, макетів і шаблонів

- Гость:

- Сторінка входу: форма для введення ім'я користувача та паролю.
- Сторінка бронювання столів: форма для вибору стола, дати та часу бронювання.

- Адміністратор:

- Сторінка входу: форма для введення ім'я користувача та паролю.
- Сторінка замовлення сировини: форма для вказання кількості м'яса, риби та овочів.

3.5.3. Створення функціоналу програмного продукту

Гость (guest.php)

- Форма для вибору стола, дати та часу бронювання.
- Обробник форми для бронювання столів (book_table.php), який додає

запис до таблиці bookings у базі даних.

Адміністратор (admin.php)

- Форма для вказання кількості м'яса, риби та овочів для замовлення.
- Обробник форми для замовлення сировини (order_supplies.php), який

додає запис до таблиці supply_orders у базі даних.

3.5.4. Опис реалізованої програмної розробки (інструкція користувача)

Гость

1. Відкрийте сторінку входу.
2. Введіть своє ім'я користувача та пароль.
3. Після успішного входу перейдіть на сторінку бронювання столів.
4. Оберіть бажаний стіл, дату та час бронювання та натисніть кнопку

"Забронювати".

5. Ви будете перенаправлені на сторінку підтвердження бронювання.

Адміністратор

1. Відкрийте сторінку входу.
2. Введіть своє ім'я користувача та пароль.
3. Після успішного входу перейдіть на сторінку замовлення сировини.
4. Введіть кількість м'яса, риби та овочів, яку ви хочете замовити.
5. Натисніть кнопку "Замовити".
6. Ви будете перенаправлені на сторінку підтвердження замовлення.

Ласкаво просимо до системи бронювання ресторану

[Увійти](#)

Рис. 3.5.3. Головна сторінка

Увійти

Ім'я користувача:

Пароль:

Рис. 3.5.4. Вхід під ім'ям «ГІСТЬ»

Ласкаво просимо, Гостю!

Оберіть стіл для бронювання

Оберіть стіл:

Дата:

Час:

[Вийти](#)

Рис. 3.5.5. Що бачить та що може робити «ГІСТЬ»

Увійти

Ім'я користувача:

Пароль:

Рис. 3.5.6. Вхід під ім'ям «АДМІН»

РОЗДІЛ 4. ОБСТЕЖЕННЯ СИСТЕМИ УПРАВЛІННЯ РЕСТОРАНОМ ДЛЯ АВТОМАТИЗАЦІЇ ЙОГО БІЗНЕС-ПРОЦЕСІВ

Аналіз організаційної структури ресторану та створення її моделі

Сучасний ресторан являє собою складну організаційну структуру, що включає декілька ключових підрозділів, кожен з яких виконує специфічні функції для забезпечення якісного обслуговування гостей. Основу організаційної структури складають три основні напрямки діяльності: операційний (кухня та зал), адміністративний (управління та координація) та сервісний (обслуговування клієнтів).

На вершині організаційної піраміди знаходиться керівництво ресторану, яке включає директора або власника закладу, що відповідає за загальну стратегію розвитку, фінансову політику та прийняття ключових рішень. Безпосередньо під керівництвом працює адміністратор залу, який координує роботу всього персоналу, контролює якість обслуговування та вирішує оперативні питання, що виникають під час роботи закладу [1].

Операційний блок представлений кухнею на чолі з шеф-кухарем, який керує процесом приготування страв, складанням меню та контролем якості продукції. До цього підрозділу також входять кухарі різних спеціалізацій, помічники кухаря та персонал, відповідальний за підготовку інгредієнтів. Паралельно функціонує сервісна служба, яка включає офіціантів, барменів та хостес, що забезпечують безпосередній контакт з гостями.

Особливе місце в структурі займає підрозділ бронювання та планування, який традиційно входить до обов'язків адміністратора або спеціально призначеного співробітника. Цей підрозділ відповідає за прийом заявок на бронювання столиків, координацію завантаженості залу, планування розсадки гостей та ведення обліку резервацій. Саме цей напрямок діяльності потребує особливої уваги через свою критичну важливість для забезпечення ефективного використання місць у залі.

Допоміжні служби включають персонал з прибирання, службу безпеки,

бухгалтерію та постачання. Всі ці підрозділи працюють у тісній взаємодії, забезпечуючи комплексне функціонування ресторану [2]. Ефективність роботи всієї організаційної структури значною мірою залежить від якості координації між підрозділами та швидкості обміну інформацією, особливо в частині управління бронюванням столиків.

Виділення бізнес-процесів ресторану та створення його процесної моделі

Діяльність ресторану можна розділити на декілька ключових бізнес-процесів, кожен з яких має свою специфіку та вимагає окремого розгляду. Основними процесами є: управління бронюванням столиків, обслуговування гостей, приготування страв, управління запасами, фінансовий облік та маркетингова діяльність. Центральне місце серед цих процесів займає управління бронюванням, оскільки воно безпосередньо впливає на завантаженість закладу та рівень доходів.

Процес управління бронюванням столиків включає декілька етапів: прийом заявки від клієнта, перевірку доступності столиків на потрібну дату та час, резервацію місця, підтвердження бронювання та контроль його виконання. Кожен з цих етапів має свої особливості та потребує чіткої регламентації. На етапі прийому заявки важливо зібрати всю необхідну інформацію від клієнта: бажану дату та час, кількість осіб, контактні дані та особливі побажання.

Процес обслуговування гостей тісно пов'язаний з бронюванням і включає зустріч гостей, розсадку за столиками, прийом замовлення, подачу страв та напоїв, розрахунок з клієнтами. Ефективність цього процесу значною мірою залежить від якості попереднього планування та розподілу столиків. Неякісне управління бронюванням може призвести до конфліктних ситуацій, перебоїв в обслуговуванні та втрати клієнтів.

Операційні процеси ресторану включають управління кухнею, планування меню, закупівлю продуктів, контроль якості страв та дотримання санітарних норм. Ці процеси повинні бути синхронізовані з процесом бронювання для забезпечення оптимального завантаження кухні та своєчасної подачі замовлень.

Планування завантаженості залу дозволяє кухні підготуватися до пікових періодів та забезпечити високу якість обслуговування.

Підтримуючі бізнес-процеси включають управління персоналом, фінансовий облік, маркетинг та рекламу, технічне обслуговування обладнання. Всі ці процеси створюють необхідну інфраструктуру для ефективного функціонування основних операційних процесів [3]. Процесна модель ресторану демонструє тісну взаємозалежність всіх елементів системи, де управління бронюванням виступає як ключовий процес, що визначає ритм роботи всього закладу.

Розробка системи показників і концептуальної схеми управління процесу бронювання столиків

Для ефективного управління процесом бронювання столиків необхідно визначити систему ключових показників ефективності (KPI), які дозволять оцінювати якість роботи та виявляти напрямки для покращення. Основними показниками є: коефіцієнт завантаженості залу, кількість успішних бронювань за період, відсоток скасованих бронювань, середній час обробки заявки на бронювання та рівень задоволеності клієнтів процесом резервації.

Коефіцієнт завантаженості залу розраховується як відношення зайнятих столиків до загальної кількості доступних місць за певний період. Цей показник дозволяє оцінити ефективність використання ресторанного простору та планувати заходи щодо збільшення відвідуваності. Оптимальний рівень завантаженості становить 75-85%, що забезпечує високий дохід при збереженні якості обслуговування. Кількість та якість бронювань характеризуються декількома метриками: загальна кількість заявок на бронювання за період, кількість підтверджених резервацій, кількість скасованих клієнтами бронювань та кількість "no-show" (неявки клієнтів). Аналіз цих показників дозволяє виявити закономірності в поведінці клієнтів та оптимізувати стратегію управління бронюванням, наприклад, встановити правила передбронювання або систему штрафів за скасування.

Операційні показники включають середній час обробки заявки на бронювання, кількість помилок при прийомі замовлень, час відповіді на запити клієнтів та ефективність роботи персоналу. Ці метрики особливо важливі для оцінки якості ручної обробки бронювань та обґрунтування необхідності автоматизації процесу. Скорочення часу обробки заявки з 10-15 хвилин до 2-3 хвилин може значно покращити клієнтський досвід.

Концептуальна схема управління процесом бронювання включає три основні блоки: блок прийому та обробки заявок, блок планування та координації, блок контролю та аналізу. Кожен блок має свої функції, відповідальних співробітників та інструменти для реалізації [4]. Центральним елементом схеми є база даних бронювань, яка забезпечує централізоване зберігання інформації та можливість швидкого доступу до актуальних даних про завантаженість столиків.

Обґрунтування необхідності створення автоматизованої системи управління бронюванням столиків

Аналіз поточного стану управління бронюванням столиків в ресторані виявляє ряд критичних проблем, які значно знижують ефективність роботи та негативно впливають на якість обслуговування клієнтів. Основними проблемами є: висока трудомісткість ручної обробки заявок, висока ймовірність помилок при веденні записів, складність контролю завантаженості залу, неможливість оперативного аналізу статистики бронювань та обмежені можливості для клієнтів щодо самостійного оформлення резервацій.

Ручна обробка заявок на бронювання потребує значних трудових ресурсів та часу. Співробітник повинен прийняти дзвінок або особисто спілкуватися з клієнтом, записати всі необхідні дані, перевірити доступність столиків, зафіксувати бронювання в журналі та підтвердити резервацію. Цей процес займає 10-15 хвилин на одне бронювання, що при високому потоці заявок створює черги та знижує якість обслуговування. Крім того, персонал не може одночасно обробляти декілька заявок, що обмежує пропускну здатність системи.

Ведення записів про бронювання в паперовому або електронному журналі

створює високий ризик помилок: можливі подвійні бронювання одного столика, втрата інформації про клієнтів, неточності в датах та часі резервацій. Такі помилки призводять до конфліктних ситуацій з клієнтами, погіршення репутації ресторану та втрати доходів. Статистика показує, що ймовірність помилки при ручному веденні записів становить 5-10%, що є неприйнятним рівнем для професійного закладу. Контроль завантаженості залу та планування розсадки гостей при ручному управлінні є складним завданням, особливо в періоди високого попиту. Персонал не має можливості швидко оцінити доступність столиків на певну дату, проаналізувати тенденції завантаженості або спрогнозувати потребу в додаткових місцях. Це призводить до неефективного використання ресторанного простору та втрати потенційних клієнтів [5].

Впровадження автоматизованої системи управління бронюванням столиків дозволить вирішити всі зазначені проблеми та забезпечить значні переваги як для ресторану, так і для клієнтів. Система забезпечить автоматизацію процесу прийому та обробки заявок, мінімізує ймовірність помилок, надасть можливості для аналітики та планування, покращить клієнтський досвід завдяки можливості онлайн-бронювання. Економічний ефект від впровадження системи включає зниження трудових витрат, збільшення кількості бронювань, покращення завантаженості залу та підвищення загального рівня задоволеності клієнтів.

4.1. ІМІТАЦІЙНЕ МОДЕЛЮВАННЯ ІЗ ЗАСТОСУВАННЯМ НОТАЦІЇ VRMN ДЛЯ ОПТИМІЗАЦІЇ БІЗНЕС-ПРОЦЕСУ БРОНЮВАННЯ СТОЛИКІВ

4.1.1 Створення моделі "як є" для бізнес-процесу бронювання столиків

Модель «як є» показує, як процес обслуговування клієнтів відбувається до повної автоматизації або без зручної єдиної системи керування. У такому випадку клієнт звертається до ресторану для бронювання столика або приходить без попереднього запису, після чого працівник закладу перевіряє наявність вільних місць. На практиці ця перевірка часто виконується вручну: адміністратор або менеджер переглядає записи, уточнює час, кількість гостей і намагається визначити, чи не зайнятий потрібний столик. Якщо інформація ведеться не в одній системі, а частково в записнику, таблиці або усно, можуть виникати помилки. Наприклад, одне бронювання можуть записати двічі або не врахувати, що після клієнтів потрібен час на підготовку столика.

Після підтвердження бронювання дані клієнта передаються персоналу, який уже безпосередньо готує столик і приймає відвідувачів. Варто зазначити, що в такому процесі багато залежить від уважності працівника. Якщо менеджер забув оновити статус бронювання або не передав інформацію іншому працівнику, виникає плутанина. Клієнт може прийти у визначений час, а столик ще буде зайнятий або не підготовлений. Такі ситуації не завжди критичні, але вони впливають на якість обслуговування і створюють зайве навантаження для персоналу.

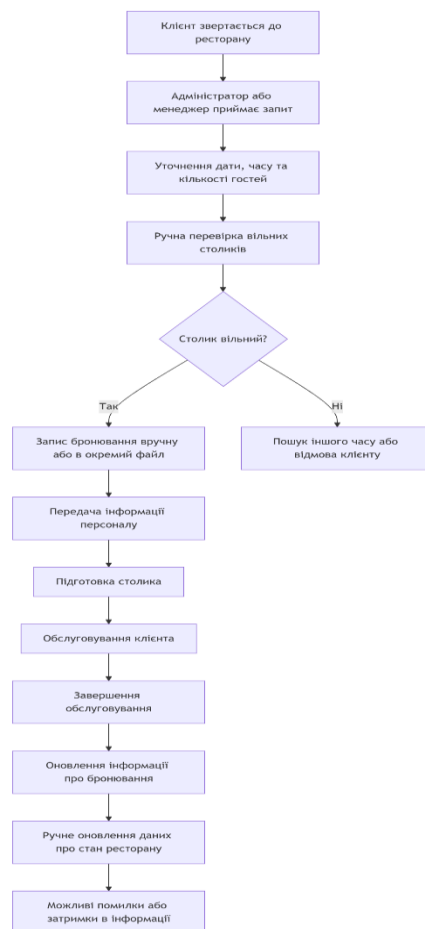


Рисунок 4.1.1 – Блок-схема бізнес-процесу обслуговування замовлень «як є»

Окремою проблемою у моделі «як є» є те, що процес обслуговування клієнта майже не пов'язаний зі складським обліком. Менеджер може знати, що певні товари або продукти закінчуються, але ця інформація не завжди швидко потрапляє в систему обліку. Як показує досвід, коли складські залишки ведуться вручну, списання продукції або прихід товару можуть виконуватися із запізненням. Через це керівництву складніше оцінити реальний стан ресторану в конкретний момент часу.

Таким чином, модель «як є» відображає процес, у якому основні дії виконуються послідовно, але між ними є багато ручної роботи. Клієнт звертається до ресторану, менеджер або адміністратор перевіряє можливість обслуговування, оформлює бронювання, передає інформацію персоналу, а після завершення обслуговування дані можуть оновлюватися не одразу. Саме через це виникає потреба у web-системі, яка дозволить централізовано зберігати інформацію про столики, бронювання, користувачів і складські операції.

4.1.2. Підготовка BPMN-моделі ресторанного бізнес-процесу для виконання у веб-додатку

Після аналізу поточного процесу обслуговування клієнтів було підготовлено BPMN-модель ресторанного бізнес-процесу, яка використовується як основа для реалізації web-додатка. Така модель дозволяє не просто описати послідовність дій працівників ресторану, а й зрозуміти, як саме ці процеси будуть виконуватися всередині інформаційної системи. На практиці BPMN-моделі часто використовують для того, щоб ще до написання програмного коду визначити логіку роботи системи та взаємодію між користувачами.

У розробленому проєкті основна увага приділялась процесам бронювання столиків і складського обліку, оскільки саме вони найбільше пов'язані з щоденною роботою адміністратора та менеджера ресторану. Коли клієнт звертається для бронювання столика, менеджер вносить інформацію у web-додаток, після чого система автоматично перевіряє доступність місць і наявність конфліктів по часу. Варто зазначити, що ця перевірка виконується вже не вручну, а безпосередньо через базу даних, де зберігаються всі активні бронювання. Якщо столик зайнятий, система повідомляє про це користувача та не дозволяє створити конфліктне бронювання.

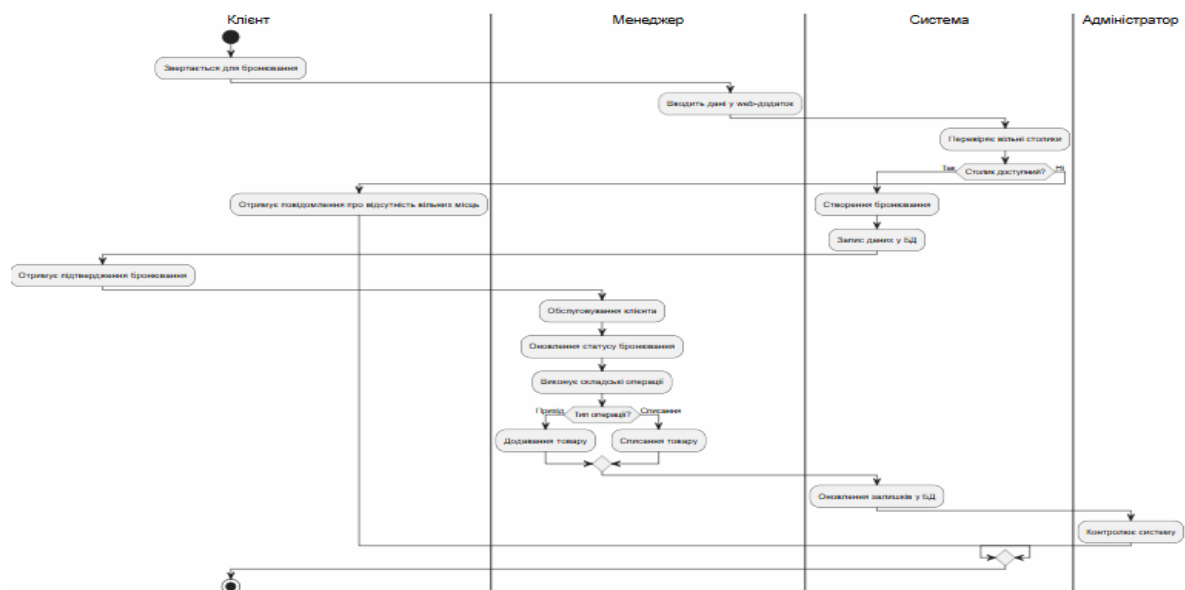


Рисунок 4.1.2 – Схема динамічної моделі процесів автоматизації роботи ресторану

Під час побудови BPMN-моделі також було враховано роботу зі складом ресторану. Менеджер може виконувати прихід товару за штрихкодом або оформлювати списання продукції. Усі дії автоматично фіксуються в базі даних, а інформація про залишки оновлюється без додаткових ручних операцій. Як показує досвід, саме автоматичне оновлення даних дозволяє уникнути помилок, які часто виникають під час паперового обліку або використання окремих таблиць.

Окремо в моделі враховано адміністратора системи. Його функції пов'язані з керуванням користувачами, зміною ролей та контролем доступу до окремих функцій web-додатка. Можна побачити, що BPMN-модель у цьому випадку показує не лише послідовність дій, а й розподіл відповідальності між користувачами системи. Це важливо для подальшої реалізації ролей у Flask-додатку та обмеження доступу до окремих сторінок.

Підготовлена модель стала основою для реалізації логіки web-системи, оскільки дозволила описати рух інформації між користувачем, інтерфейсом системи та базою даних. Завдяки цьому процеси ресторану були перенесені у цифровий формат із мінімальною кількістю ручних операцій.

4.1.3. Симуляція виконання бізнес-процесу бронювання та аналіз її результатів

Для оцінки ефективності поточного бізнес-процесу було проведено симуляцію його виконання протягом типового тижня роботи ресторану з урахуванням реальних показників завантаженості та інтенсивності звернень клієнтів. Симуляція базувалася на даних про 50 столиків різної місткості та середньому потоці 120-150 звернень на тиждень, що відповідає показникам середньостатистичного ресторану класу "casual dining".

Результати симуляції показали, що середній час обробки одного звернення клієнта становить 12 хвилин, включаючи прийом дзвінка, з'ясування деталей бронювання, перевірку доступності столиків та фіксацію резервації в журналі. У пікові години (п'ятниця-субота вечором) цей час може збільшуватися до 18-20 хвилин через необхідність детального аналізу завантаженості та пошуку

альтернативних варіантів розсадки.

Аналіз виявив, що приблизно 15% звернень призводить до конфліктних ситуацій або незадоволення клієнтів через помилки в розкладі, подвійні бронювання або неточності в інформації про доступність столиків. Додатково, близько 8% потенційних клієнтів відмовляються від бронювання через тривале очікування відповіді або неможливість оперативно отримати інформацію про вільні столики в зручний для них час.

Пропускна здатність системи обмежується можливостями адміністратора, який може ефективно обслуговувати не більше 5-6 звернень на годину. Це створює вузьке місце в пікові періоди, коли інтенсивність звернень може досягати 12-15 на годину. Внаслідок цього формуються черги, клієнти змушені очікувати або намагатися зателефонувати пізніше, що призводить до втрати потенційних бронювань.

Економічний аналіз симуляції показав, що неефективність поточного процесу призводить до втрати приблизно 20% потенційних резервацій, що становить суттєву частку можливих доходів ресторану. Крім того, трудовитрати на обробку бронювань становлять близько 25-30 годин на тиждень, що еквівалентно повній зайнятості одного співробітника [8].

Симуляція також виявила сезонні коливання навантаження: в періоди свят та вихідних кількість звернень збільшується в 2-2.5 рази, що створює критичне перевантаження для ручної системи обробки. У такі періоди відсоток втрачених бронювань може досягати 35-40%, що суттєво впливає на прибутковість закладу.

4.1.4. Реалізація альтернативних варіантів виконання бізнес-процесу та їх порівняльний аналіз

Для оптимізації бізнес-процесу бронювання столиків було розроблено три альтернативні варіанти реалізації, кожен з яких передбачає різний рівень автоматизації та залучення інформаційних технологій. Перший варіант - часткова автоматизація з використанням електронної таблиці та email-повідомлень, другий - створення простого веб-сайту з формою бронювання, третій - повнофункціональна веб-система з інтерактивним інтерфейсом та

автоматизованим управлінням.

Варіант часткової автоматизації передбачає заміну паперового журналу на електронну таблицю (Excel або Google Sheets) з можливістю спільного доступу для персоналу ресторану. Клієнти продовжують звертатися телефоном, але адміністратор отримує інструменти для швидшого пошуку вільних столиків та автоматичного розрахунку завантаженості. Додатково впроваджується система email-підтверджень бронювань та нагадувань для клієнтів.

Другий варіант включає створення простого веб-сайту з формою онлайн-бронювання, де клієнти можуть самостійно обирати дату, час та кількість гостей. Заявки автоматично надходять адміністратору на email для подальшого підтвердження або відхилення. Цей варіант зберігає контроль персоналу над процесом, але надає клієнтам можливість подавати заявки цілодобово без необхідності телефонувати [9].

Третій варіант представляє повнофункціональну автоматизовану систему, яка реалізована в рамках даного проекту. Система включає інтерактивний інтерфейс для клієнтів з візуалізацією доступних столиків, автоматичну перевірку конфліктів бронювань, адміністративну панель для управління резерваціями та аналітичні інструменти для оцінки ефективності роботи.

Порівняльний аналіз показав значні відмінності в ефективності запропонованих варіантів. Перший варіант дозволяє скоротити час обробки заявки на 30-40% завдяки швидшому пошуку інформації, але не вирішує проблему обмеженої пропускної здатності та залежності від адміністратора. Другий варіант збільшує кількість заявок на 25-30% через можливість цілодобового прийому, але створює додаткове навантаження на персонал через необхідність ручного підтвердження кожного бронювання.

Третій варіант демонструє найвищу ефективність: скорочення часу обробки заявки до 2-3 хвилин, збільшення пропускної здатності системи в 5-6 разів, практично повне виключення помилок подвійного бронювання та можливість цілодобового прийому заявок без участі персоналу. Економічний ефект включає зниження трудовитрат на 70-80% та збільшення кількості успішних бронювань на

40-50%.

Вибір оптимального варіанта здійснювався на основі комплексної оцінки критеріїв: економічної ефективності, технічної складності впровадження, зручності для клієнтів та персоналу, можливостей масштабування та довгострокової перспективи розвитку. За результатами аналізу для реалізації було обрано третій варіант як найбільш перспективний та відповідний сучасним вимогам ресторанного бізнесу [10].

4.2.1 Підготовка обраної моделі для виконання у веб-додатку

Після вибору оптимального варіанта автоматизації було розпочато детальну підготовку моделі бізнес-процесу для реалізації у веб-додатку. Цей етап включав декомпозицію процесу на окремі функціональні модулі, визначення інтерфейсів взаємодії, специфікацію вимог до системи та створення архітектурного дизайну програмного рішення.

Функціональна декомпозиція виділила п'ять основних модулів системи: модуль візуалізації столиків для клієнтів, модуль прийому та валідації заявок на бронювання, модуль управління конфліктами та перевірки доступності, адміністративний модуль для персоналу ресторану та модуль звітності і аналітики. Кожен модуль отримав чітко визначені функції, інтерфейси входу-виходу та критерії якості роботи.

Модуль візуалізації столиків забезпечує інтуїтивний інтерфейс для клієнтів з можливістю перегляду доступних столиків на обрану дату та час. Система візуально відображає статус кожного столика (вільний, зайнятий, забронований) та дозволяє клієнту обрати найбільш зручний варіант. Інтерфейс оптимізовано для використання як на десктопних комп'ютерах, так і на мобільних пристроях, що важливо з огляду на сучасні тенденції користування інтернетом.

Модуль прийому заявок реалізує автоматизовану обробку форм бронювання з валідацією введених даних на стороні клієнта та сервера. Система перевіряє коректність дати та часу, відповідність кількості гостей місткості обраного столика, правильність заповнення контактних даних та додаткових вимог. У

випадку виявлення помилок користувач отримує зрозумілі повідомлення з інструкціями щодо виправлення.

Модуль управління конфліктами є ключовим елементом системи, що забезпечує запобігання подвійним бронюванням та оптимальне розподілення столиків. Система автоматично перевіряє доступність обраного столика на потрібну дату та час, враховуючи встановлені інтервали між бронюваннями (мінімум 3 години) та можливість дострокового звільнення столиків. Алгоритм також пропонує альтернативні варіанти у випадку недоступності першочергового вибору клієнта.

Адміністративний модуль надає персоналу ресторану потужні інструменти для управління всіма аспектами бронювання: перегляд поточних резервацій, редагування існуючих бронювань, скасування резервацій, пошук клієнтів за різними критеріями та генерацію звітів. Інтерфейс розроблено з урахуванням потреб користувачів різного рівня технічної підготовки, з акцентом на простоту та ефективність роботи [11].

Технічна архітектура системи базується на технологіях HTML5, CSS3, JavaScript для frontend-частини та Python Flask для backend-логіки з використанням MySQL як системи управління базою даних. Така архітектура забезпечує стабільність, масштабованість та можливість подальшого розвитку функціональності системи відповідно до зростаючих потреб ресторану.

РОЗДІЛ 5. РОЗРОБКА ВЕБ-ДОДАТКУ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ РЕСТОРАНУ

У даному розділі буде розглянуто процес розробки веб-додатка для автоматизації бізнес-процесів ресторану. Буде описано структуру системи, реалізацію функцій бронювання столиків, керування користувачами, складського обліку та роботи з товарами. Також буде розглянуто використання Flask і MySQL для створення серверної частини додатка, організацію ролей адміністратора та менеджера, а також взаємодію веб-системи з базою даних ресторану.

Технології та інструменти розробки системи

Під час розробки системи автоматизації ресторану було використано декілька технологій, які забезпечують роботу веб-додатка, взаємодію з базами даних та реалізацію бізнес-процесів. Основою серверної частини став Flask – фреймворк для мови програмування Python, який дозволяє створювати web-додатки з відносно простою структурою та гнучкою організацією коду. Варто зазначити, що Flask добре підходить саме для невеликих і середніх систем, де важливо швидко реалізувати логіку роботи сторінок, авторизацію користувачів та взаємодію з базою даних без зайвого ускладнення проєкту.

Для зберігання інформації використовується система керування базами даних MySQL. У базі даних зберігаються користувачі системи, бронювання столиків, інформація про товари, операції приходу та списання продукції. На практиці використання окремої СУБД дозволяє централізовано працювати з даними та спрощує доступ до інформації з різних модулів системи. Крім основної бази ресторану, у проєкті також використовується окрема база даних WinCC, яка містить технологічні параметри вентиляції приміщення. Web-додаток підключається до цієї бази окремою функцією та відображає інформацію у відповідному модулі системи.

Клієнтська частина реалізована за допомогою HTML, CSS та Bootstrap. HTML використовується для створення структури сторінок, форм бронювання, таблиць та елементів керування. CSS відповідає за оформлення інтерфейсу та

зовнішній вигляд системи, а Bootstrap дозволив швидше реалізувати адаптивний дизайн і готові компоненти інтерфейсу. Можна побачити, що завдяки цьому сторінки системи мають однаковий стиль і коректно відображаються на різних розмірах екрана.

Для роботи з базою даних у Python використовується бібліотека `mysql.connector`, через яку виконуються SQL-запити до MySQL. Як показує досвід, такий підхід є достатньо зручним для невеликих web-систем, оскільки дозволяє напряду працювати із таблицями та швидко реалізовувати необхідну логіку без використання складних ORM-систем. У системі через SQL-запити виконується створення бронювань, оновлення статусів, керування товарами та зміна ролей користувачів.

Під час розробки також використовувались допоміжні інструменти. Для локального запуску сервера та бази даних застосовувався пакет Open Server Panel, який містить Apache, MySQL і phpMyAdmin для адміністрування баз даних через браузер. Редагування програмного коду виконувалось у Visual Studio Code, оскільки це середовище підтримує Python, HTML, CSS і дозволяє зручно працювати з усією структурою проєкту.

Проектування структури бази даних веб-додатку

Під час розробки веб-додатку було спроектовано структуру бази даних, яка забезпечує зберігання інформації про користувачів системи, бронювання столиків, склад товарів та операції із продукцією ресторану. Варто зазначити, що структура бази даних створювалась таким чином, щоб усі основні бізнес-процеси могли виконуватись через веб-додаток без дублювання інформації та зайвих ручних операцій. На практиці саме правильна організація таблиць значно спрощує реалізацію програмної логіки та подальше розширення системи.

Для реалізації системи використовується база даних MySQL. Основною таблицею є таблиця `users`, яка містить інформацію про користувачів веб-додатку. У ній зберігаються логіни, паролі, ролі користувачів, а також статус активності

облікового запису. Це дозволяє адміністратору керувати доступом до функцій системи та змінювати права користувачів без внесення змін у програмний код.

Таблиця 5.1 – Структура таблиці users

Поле	Тип даних	Опис
id	INT	Унікальний ідентифікатор користувача
full_name	VARCHAR(255)	Повне ім'я користувача
username	VARCHAR(100)	Логін для входу в систему
password	VARCHAR(255)	Пароль користувача
role	VARCHAR(50)	Роль користувача в системі
is_active	BOOLEAN	Статус активності користувача
created_at	TIMESTAMP	Дата створення облікового запису

Для роботи із бронюваннями створено таблицю tables, у якій міститься інформація про столики ресторану. У таблиці зберігається номер столика, кількість місць, його розташування та поточний статус. Як показує досвід, така структура дозволяє швидко визначати доступність столиків під час створення бронювання.

Таблиця 5.2 – Структура таблиці tables

Поле	Тип даних	Опис
id	INT	Ідентифікатор столика
table_number	INT	Номер столика
seats	INT	Кількість місць
location	VARCHAR(100)	Розташування столика
status	VARCHAR(50)	Поточний статус столика

Однією з основних таблиць системи є таблиця bookings, яка використовується для зберігання інформації про бронювання клієнтів. У ній містяться дані клієнта, дата і час бронювання, кількість гостей, коментар та статус замовлення. Варто зазначити, що саме ця таблиця використовується системою для перевірки конфліктів бронювання і визначення зайнятості столиків.

Таблиця 5.3 – Структура таблиці bookings

Поле	Тип даних	Опис
id	INT	Ідентифікатор бронювання
table_id	INT	Ідентифікатор столика
client_name	VARCHAR(255)	Ім'я клієнта
phone	VARCHAR(50)	Контактний номер телефону
email	VARCHAR(100)	Електронна

		пошта клієнта
guests_count	INT	Кількість гостей
booking_date	DATE	Дата бронювання
booking_time	TIME	Час бронювання
comment	TEXT	Додатковий коментар
status	VARCHAR(50)	Статус бронювання
created_by	INT	Користувач, який створив бронювання

Для ведення складського обліку була створена таблиця products. У ній зберігаються товари ресторану, їх штрихкоди, кількість та одиниці виміру. На практиці використання штрихкодів значно спрощує процес приходу товару та пошук продукції на складі.

Таблиця 5.4 – Структура таблиці products

Поле	Тип даних	Опис
id	INT	Ідентифікатор товару
barcode	VARCHAR(100)	Штрихкод товару
name	VARCHAR(255)	Назва товару
quantity	INT	Кількість товару
unit	VARCHAR(50)	Одиниця виміру
updated_at	TIMESTAMP	Дата оновлення інформації

Для фіксації всіх операцій зі складом використовується таблиця `product_operations`. Вона містить інформацію про прихід і списання товарів, кількість продукції та причину виконання операції. Можна побачити, що така структура дозволяє зберігати історію всіх дій зі складом і контролювати рух товарів у системі.

Таблиця 5.5 – Структура таблиці `product_operations`

Поле	Тип даних	Опис
<code>id</code>	INT	Ідентифікатор операції
<code>product_barcode</code>	VARCHAR(100)	Штрихкод товару
<code>operation_type</code>	VARCHAR(50)	Тип операції
<code>quantity</code>	INT	Кількість товару
<code>reason</code>	TEXT	Причина виконання операції
<code>created_by</code>	INT	Користувач, який виконав операцію
<code>created_at</code>	TIMESTAMP	Дата виконання операції

Після проєктування структури бази даних були визначені зв'язки між таблицями. Таблиця `bookings` пов'язана з таблицею `tables` через поле `table_id`, оскільки кожне бронювання належить певному столику ресторану. Також таблиця `bookings` має зв'язок із таблицею `users` через поле `created_by`, яке показує, хто саме створив бронювання. Таблиця `product_operations` пов'язана з таблицею `users`,

оскільки кожна складська операція виконується конкретним користувачем системи. Крім цього, зв'язок із товарами реалізовано через поле `product_barcode`, що дозволяє визначати продукцію, над якою виконувалась операція приходу або списання.

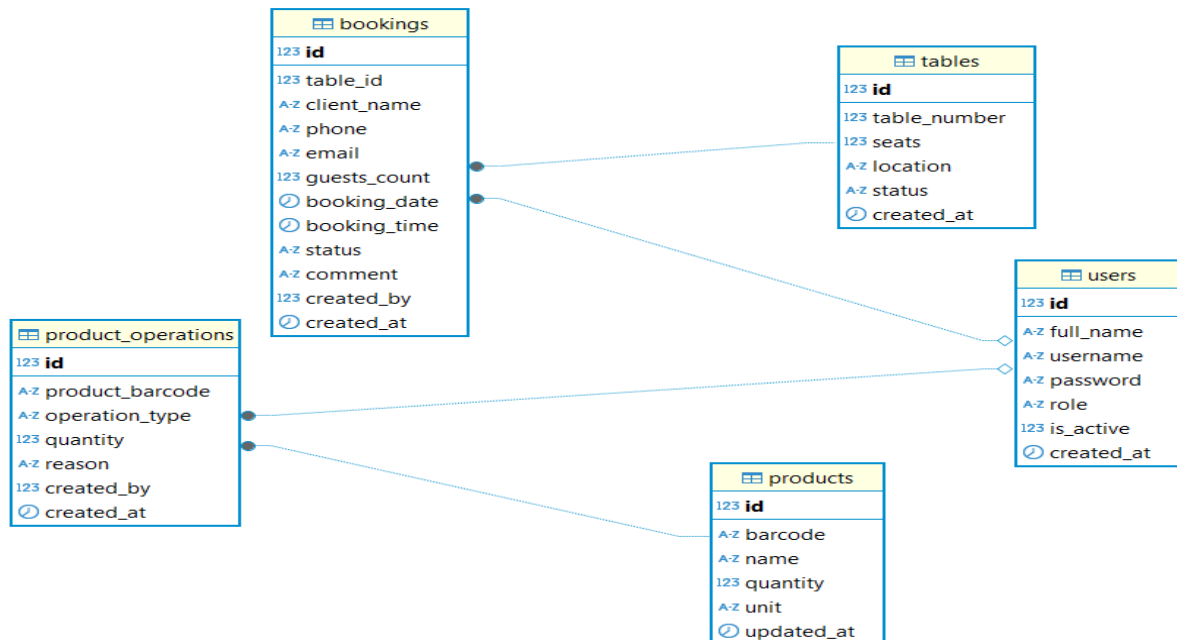


Рисунок 5.1 – Схема зв'язків та структури таблиць бази даних інформаційної системи

Розробка макетів користувацьких інтерфейсів та дизайну системи

Під час розробки веб-додатку значна увага приділялась не лише реалізації функціоналу, а й створенню зручного інтерфейсу для роботи адміністратора та менеджера ресторану. На практиці навіть добре реалізована система може бути незручною у використанні, якщо користувачеві складно швидко знайти потрібну функцію або виконати необхідну операцію. Саме тому під час створення макетів основний акцент був зроблений на простоті навігації, мінімальній кількості зайвих дій та єдиному стилі всіх сторінок системи.

Для оформлення інтерфейсу використовувалась світла кольорова схема з темною боковою панеллю навігації. Ліве меню містить основні модулі системи: панель керування, бронювання, підбір столика, склад товарів, прихід та списання товару, модулі вентиляції WinCC і систему відеоспостереження. Варто зазначити, що таке розташування елементів дозволяє користувачу швидко переходити між

сторінками без необхідності повертатися на головний екран. Крім цього, однакове розміщення меню на всіх сторінках створює відчуття цілісності системи та спрощує роботу з додатком.

Головна сторінка системи реалізована у вигляді інформаційної панелі, де відображається коротка статистика роботи ресторану. На екрані можна побачити кількість столиків, число бронювань, кількість підтверджених замовлень та товарів на складі. Як показує досвід, саме такий формат дозволяє адміністратору швидко оцінити поточний стан роботи ресторану без відкриття додаткових сторінок. Нижче на панелі також розміщено блоки з описом системи та інформацією про дані вентиляції WinCC.

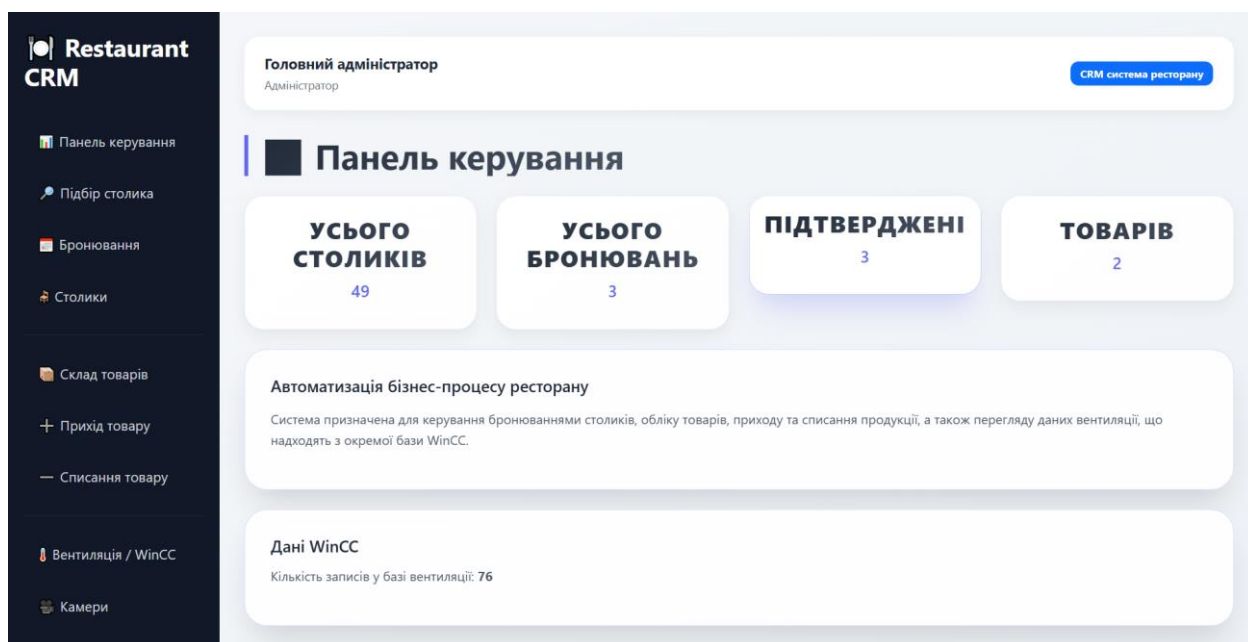


Рисунок 5.2 – Головна сторінка інтерфейсу адміністратора інформаційної системи ресторану

Для роботи з бронюванням була створена окрема сторінка підбору столика. У цьому модулі менеджер вводить дату, час та кількість гостей, після чого система автоматично підбирає доступні столики. Інтерфейс сторінки зроблений максимально спрощеним: основні поля знаходяться в одному блоці, а кнопка виконання пошуку одразу помітна користувачу. Можна побачити, що такий підхід дозволяє виконати підбір столика буквально за декілька дій.

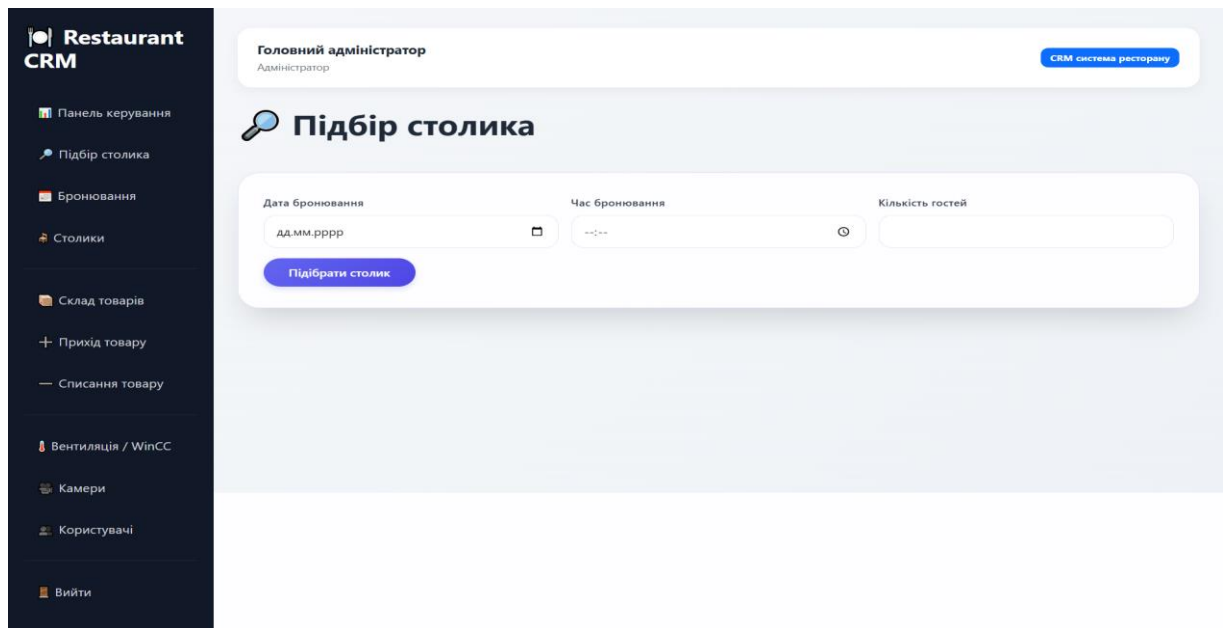


Рисунок 5.3 – Екранна форма модуля «Підбір столика»

Сторінка бронювань містить таблицю із переліком усіх створених замовлень. Для зручності користувача реалізовано фільтрацію за датою та статусом бронювання. Крім цього, менеджер може змінювати статус замовлення через випадаючий список без переходу на окрему сторінку редагування. Варто зазначити, що використання таблиць у такому модулі є найбільш зручним варіантом, оскільки дозволяє одночасно переглядати велику кількість записів.

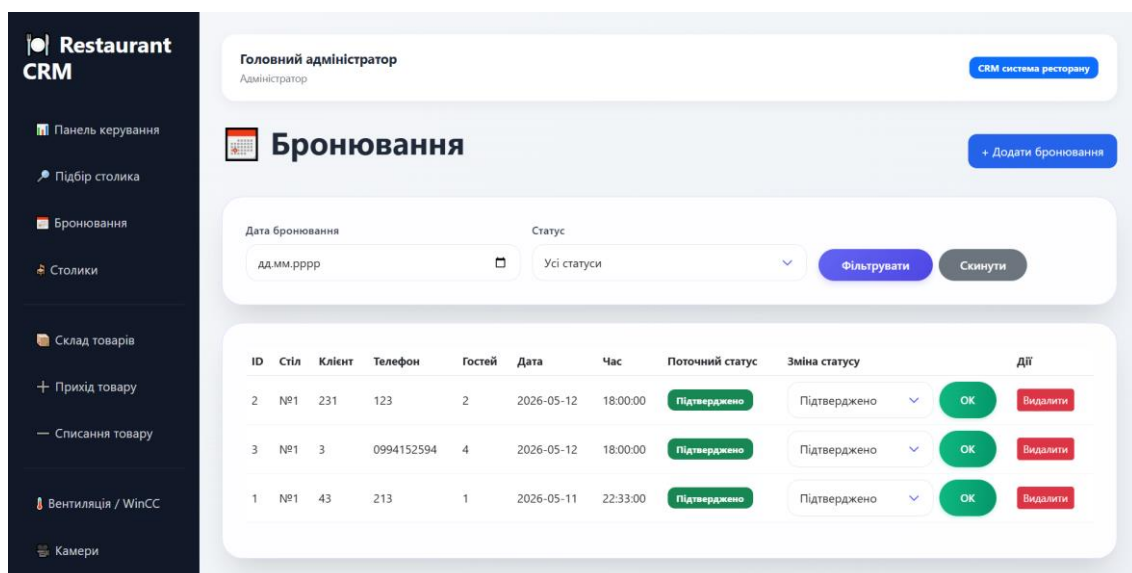


Рисунок 5.4 – Екранна форма модуля перегляду та адміністрування бронювань

Окрему увагу було приділено дизайну складського модуля. Сторінка складу товарів реалізована у вигляді таблиці з підтримкою пошуку продукції за штрихкодом. Інтерфейс приходу товару та списання продукції побудований у формі великих полів введення, що спрощує використання системи під час роботи зі сканером штрихкодів. На практиці це дозволяє менеджеру швидко вносити нові товари або виконувати списання без складних д

Рисунок 5.5 – Екранна форма модуля «Склад товарів»

У модулі вентиляції реалізовано окрему сторінку перегляду даних WinCC. На ній відображаються останні значення температури та журнал параметрів вентиляції, які надходять із окремої бази даних SCADA-системи. Для кращого сприйняття інформації основні показники винесені у великі інформаційні блоки, а історія параметрів представлена у вигляді таблиці.

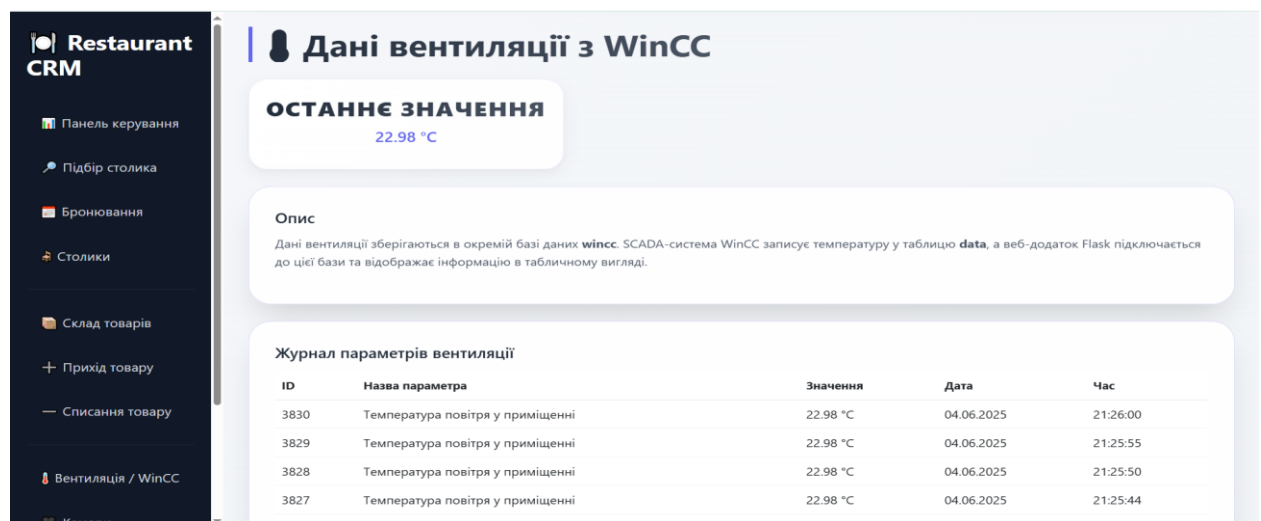


Рисунок 5.6 – Екранна форма модуля моніторингу даних вентиляції зі SCADA-системи WinCC

Також у системі передбачено сторінку відеоспостереження. На макеті реалізовано декілька блоків камер із кнопками керування та областями для відображення відеопотоку. Навіть якщо фактичне підключення камер у проєкті не реалізоване повністю, сторінка створює єдину структуру системи контролю ресторану та демонструє можливість інтеграції із зовнішніми сервісами спостереження.

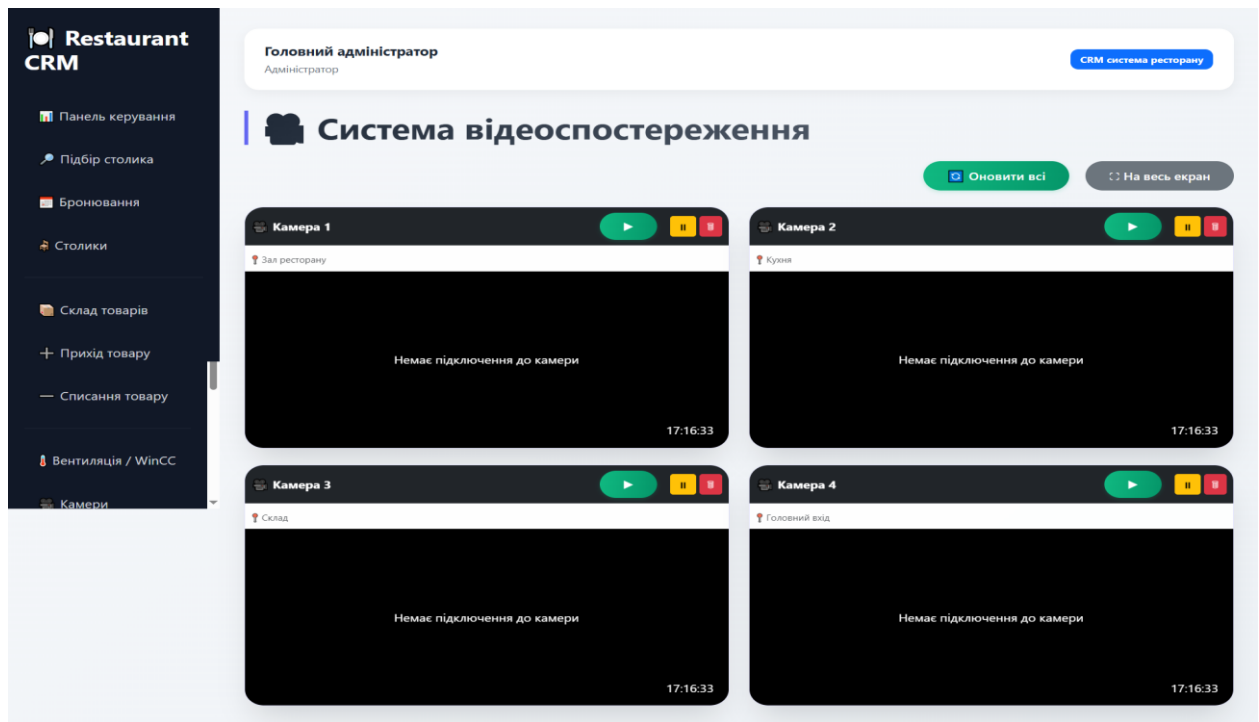


Рисунок 5.7 – Екранна форма модуля «Система відеоспостереження»

Під час створення дизайну використовувались HTML, CSS і Bootstrap. Це дозволило створити адаптивний інтерфейс із єдиним стилем оформлення, однаковими елементами керування та зручним розташуванням інформації на сторінках системи.

Створення функціональних модулів програмного продукту

Після проєктування структури бази даних та створення макетів інтерфейсу було виконано реалізацію основних функціональних модулів системи. Усі модулі розроблялись у межах одного Flask-додатка, де кожна сторінка має власний маршрут, логіку обробки даних і взаємодію з базою даних MySQL. Варто зазначити, що під час розробки основна увага приділялась не тільки реалізації окремих функцій, а й взаємозв'язку між модулями системи. Частина програмного коду веб-додатка наведена у файлі app.py[4].

Одним із перших було реалізовано модуль авторизації користувачів. Система перевіряє логін, пароль та активність облікового запису, після чого створює сесію користувача і визначає його роль у системі. На практиці це дозволяє обмежити доступ до окремих сторінок та функцій web-додатка.

Наприклад, адміністратор має доступ до керування користувачами та столиками, а менеджер працює із бронюваннями та складом товарів.

Для реалізації перевірки авторизації були створені окремі декоратори доступу:

```
def login_required(f):  
    @wraps(f)  
    def decorated_function(*args, **kwargs):  
        if 'user_id' not in session:  
            return redirect('/login')  
        return f(*args, **kwargs)  
    return decorated_function
```

Також у системі реалізовано перевірку ролей користувачів для адміністратора та менеджера.

Окремий функціональний модуль відповідає за роботу з бронюванням столиків. Під час створення нового бронювання система автоматично перевіряє наявність конфліктів за часом. Варто зазначити, що в алгоритмі враховується додатковий інтервал у 75 хвилин, який використовується для обслуговування клієнта та підготовки столика після завершення бронювання.

Для перевірки конфлікту реалізовано функцію:

```
def has_booking_conflict(cursor, table_id, booking_date, booking_time):  
    new_start = datetime.strptime(  
        f'{booking_date} {booking_time}',  
        "%Y-%m-%d %H:%M"  
    )  
  
    new_end = new_start + timedelta(minutes=75)
```

Після цього система отримує всі активні бронювання для конкретного столика і перевіряє перетин часових інтервалів.

Для створення бронювання реалізовано окремий маршрут /bookings/create, який працює з HTML-формою та записує інформацію до таблиці bookings. Під час надсилання форми система перевіряє правильність даних і зберігає інформацію у базі даних:

```
cursor.execute("""
    INSERT INTO bookings (
        table_id,
        client_name,
        phone,
        email,
        guests_count,
        booking_date,
        booking_time,
        comment,
        status,
        created_by
    )
    VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)
""", (
    data['table_id'],
    data['client_name'],
    data['phone'],
    data.get('email'),
    data['guests_count'],
    data['booking_date'],
    data['booking_time'],
    data.get('comment'),
    data['status'],
    session['user_id']
))
```

Для автоматизації складського обліку було реалізовано модулі приходу та списання товару. Менеджер може додавати продукцію на склад за допомогою штрихкоду або виконувати списання товару із зазначенням причини. Як показує досвід, саме автоматизація цих процесів дозволяє значно зменшити кількість помилок під час ведення обліку.

Модуль приходу товару працює за таким принципом: якщо товар із певним штрихкодом уже існує у таблиці products, система збільшує його кількість, інакше створює новий запис.

```
cursor.execute("""
    SELECT *
    FROM products
    WHERE barcode = %s
""", (barcode,))
```

Після перевірки система або оновлює кількість товару, або додає новий запис у таблицю продукції.

Для списання товару реалізовано перевірку наявності продукції та достатньої кількості одиниць на складі. Якщо товару недостатньо, система повертає повідомлення про помилку. Можна побачити, що такий підхід дозволяє уникнути від’ємних залишків товарів у базі даних.

```
elif product['quantity'] < quantity:
    error = "Недостатньо товару"
```

Окремий модуль реалізовано для перегляду даних вентиляції з SCADA-системи WinCC. У цьому випадку Flask-додаток підключається вже не до основної бази ресторану, а до окремої бази wincc, де зберігаються параметри вентиляції. Після підключення система отримує останні записи та відображає їх на сторінці веб-додатка.

```
def get_wincc_connection():
    return mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="wincc"
    )
```

Для отримання параметрів вентиляції використовується SQL-запит:

```
cursor.execute("""
    SELECT ID, Name, Parameter, Dates, Times
    FROM data
    ORDER BY ID DESC
    LIMIT 100
    """)
```

Також у системі реалізовано модуль керування користувачами. Адміністратор може створювати нових користувачів, змінювати ролі та активність облікових записів через інтерфейс web-додатка. Для цього використовується окремий маршрут оновлення користувача:

```
@app.route('/users/update/<int:id>', methods=['POST'])
@admin_required
def update_user(id):
```

Після отримання даних із форми система виконує SQL-запит оновлення ролі та статусу активності користувача.

У результаті було створено набір функціональних модулів, які забезпечують автоматизацію основних бізнес-процесів ресторану: бронювання столиків, облік товарів, складські операції, керування користувачами та перегляд даних вентиляції. Усі модулі взаємодіють між собою через єдину систему баз даних і реалізовані в межах одного Flask-додатка.

РОЗДІЛ 6. РОЗРОБКА SCADA ДЛЯ ТЕХНОЛОГІЧНОГО ОБ'ЄКТУ МОНІТОРИНГУ І УПРАВЛІННЯ

6.1. Створення моделі системи автоматичного регулювання в програмному середовищі Step7.

Для реалізації системи автоматичного керування процесом кондиціонування повітря у приміщенні ми у програмі SIMATIC Manager створюємо новий проект для контролера S7-313-2DP, скориставшись підказками помічника Wizard. Потім запускаємо програму конфігурації обладнання HW Config.

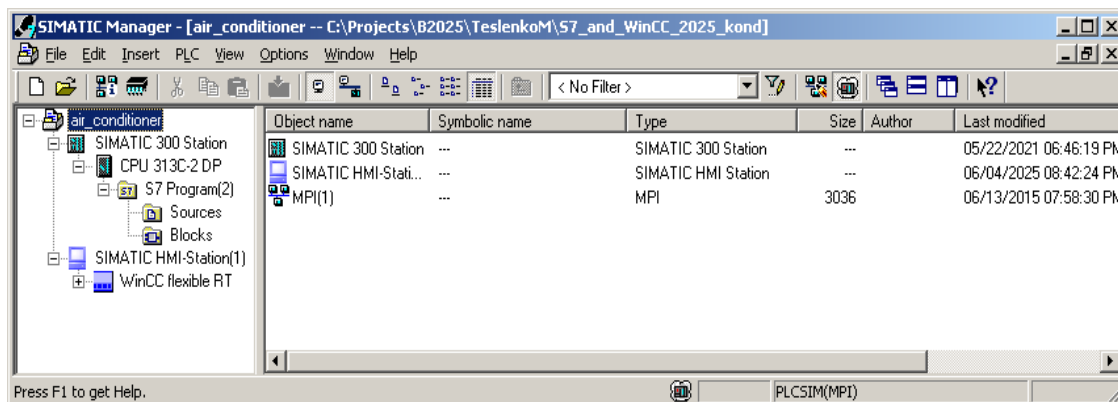


Рис. 6.1 – Створений проект в програмі SIMATIC Manager.

Далі в програмі HW Config додаємо мережу та виконуємо конфігурацію обладнання. А саме контролера та модулів ПЗО.

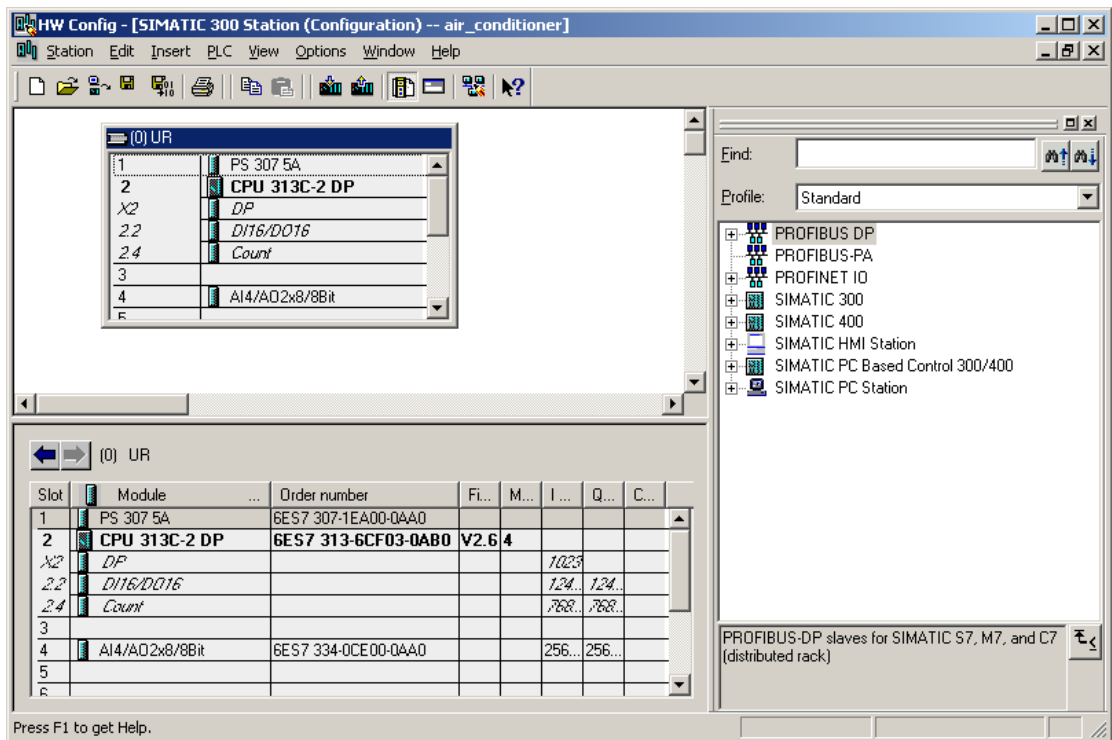


Рис. 6.2 – Додане і зконфігуроване обладнання.

В якості зв'язку між контролером і SCADA-системою будемо використовувати MPI-мережу.

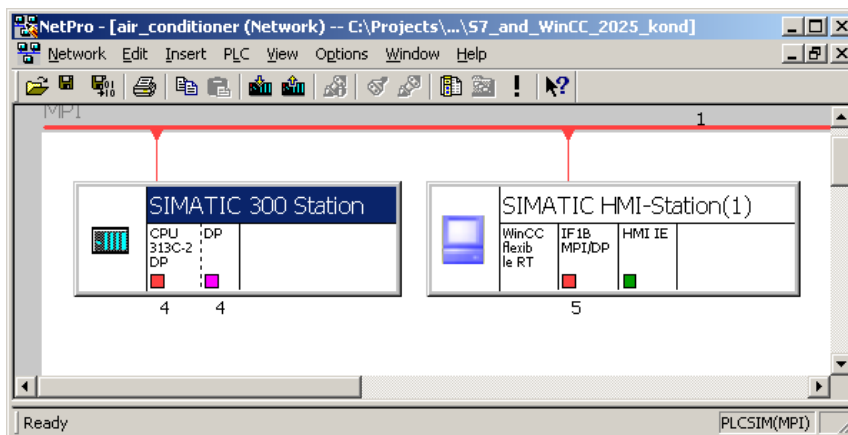


Рис. 6.3 – Зконфігурована мережа MPI.

6.1.1 Створення програми імітації роботи SAP.

В проєкті контролера створимо блок DB2 для HMI-тегів для подальшого зберігання, використання в програмі контролера і зв'язку SCADA-системи з контролером.

Address	Name	Type	Initial value	Actual value	Comment
0.0	PID_Tk.MAN_ON	BOOL	FALSE	FALSE	сигнал перемикавання
2.0	PID_Tk.MAN	REAL	0.000000e+000	0.000000e+000	сигнал керування в
6.0	PID_Tk.SP_INT	REAL	4.000000e+001	2.200000e+001	задане значення
10.0	PID_Tk.PV	REAL	0.000000e+000	0.000000e+000	поточне значення
14.0	PID_Tk.U	REAL	0.000000e+000	0.000000e+000	керуючий вплив
18.0	PID_Tk.GAIN	REAL	7.390000e+000	1.800000e+000	коефіцієнт передач
22.0	PID_Tk.TI	TIME	T#23s340MS	T#20s340MS	час ізодрому
26.0	PID_Tk.SP_INT	TIME	T#18s900MS	T#10s900MS	час предварення
30.0	PID_Tk.TM_LAG	TIME	T#3s780MS	T#3s780MS	час інерційності р
34.0	PID_Mk.MAN_ON	BOOL	FALSE	FALSE	сигнал перемикавання
36.0	PID_Mk.MAN	REAL	0.000000e+000	0.000000e+000	сигнал керування в
40.0	PID_Mk.SP_INT	REAL	7.500000e+001	5.000000e+001	задане значення
44.0	PID_Mk.PV	REAL	0.000000e+000	0.000000e+000	поточне значення
48.0	PID_Mk.U	REAL	0.000000e+000	0.000000e+000	керуючий вплив
52.0	PID_Mk.GAIN	REAL	3.560000e+000	1.200000e+000	коефіцієнт передач
56.0	PID_Mk.TI	TIME	T#11s600MS	T#1s600MS	час ізодрому
60.0	PID_Mk.TD	TIME	T#4s500MS	T#4s500MS	час предварення
64.0	PID_Mk.TM_LAG	TIME	T#900MS	T#1s	час інерційності р
68.0	PUSK	BOOL	FALSE	FALSE	Команда пуск
68.1	STOP	BOOL	FALSE	FALSE	Команда стоп
70.0	Time_2S	S5TIME	S5T#10S	S5T#10S	Час звукової сигна
72.0	V1_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при
74.0	M1_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при
76.0	T_START	BOOL	FALSE	FALSE	Ввімкнення контуру
76.1	TP_START	BOOL	FALSE	FALSE	Ввімкнення контуру
78.0	M2_start	S5TIME	S5T#10S	S5T#10S	Затримка часу при
80.0	M2_stop	S5TIME	S5T#5S	S5T#5S	Затримка часу на в
82.0	M1_stop	S5TIME	S5T#15S	S5T#15S	Затримка часу на в
84.0	V1_stop	S5TIME	S5T#10S	S5T#10S	Затримка часу на в

Рис. 6.4 – Створений блок HMI-тегів.

Створимо модель об'єкта керування за каналами температури та відносної вологості. Фрагмент програми що реалізує моделі ОК наведено на рис. 6.5.

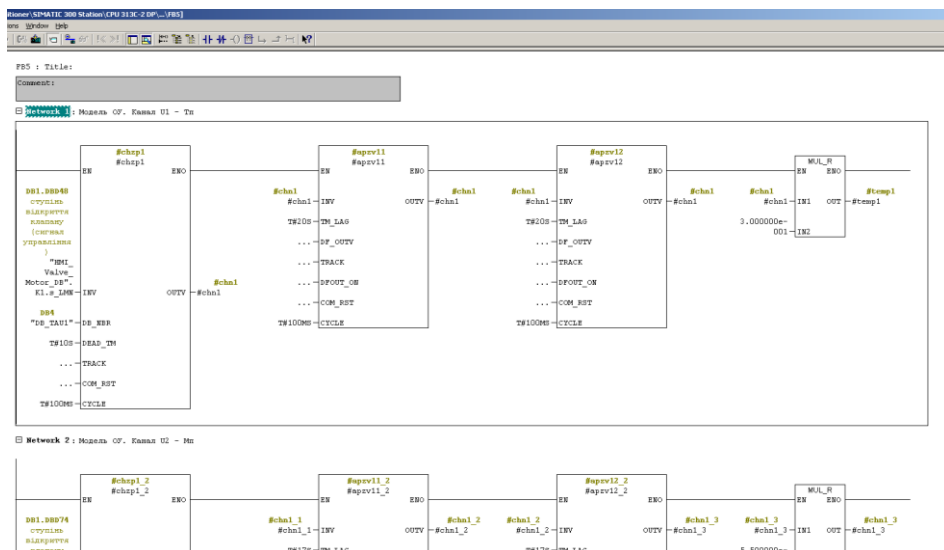


Рис. 6.5 – Фрагмент програми з моделлю ОК за каналами керування температури і вологості у приміщенні ресторану яка створена в Step7.

Створимо програму ПИД регулювання температури та вологості повітря у приміщенні ресторану. Фрагмент програми наведено на рис. 6.6.

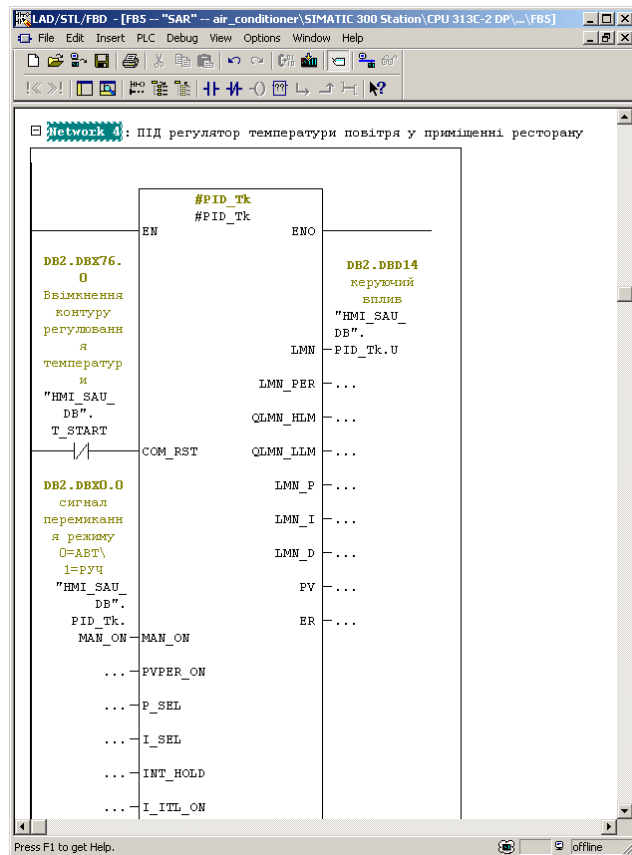


Рис. 6.6 – Фрагмент програми з моделлю ПІД-регулятора температури у приміщенні барбершопу яка створена в Step7.

6.1.2 Виконаємо зв'язок SCADA-системи і проекту SIMATIC Manager.

За допомогою SCADA-системи ми матимемо можливість змінювати параметри регулятора і значення заданої температури і вологості у приміщенні ресторану.

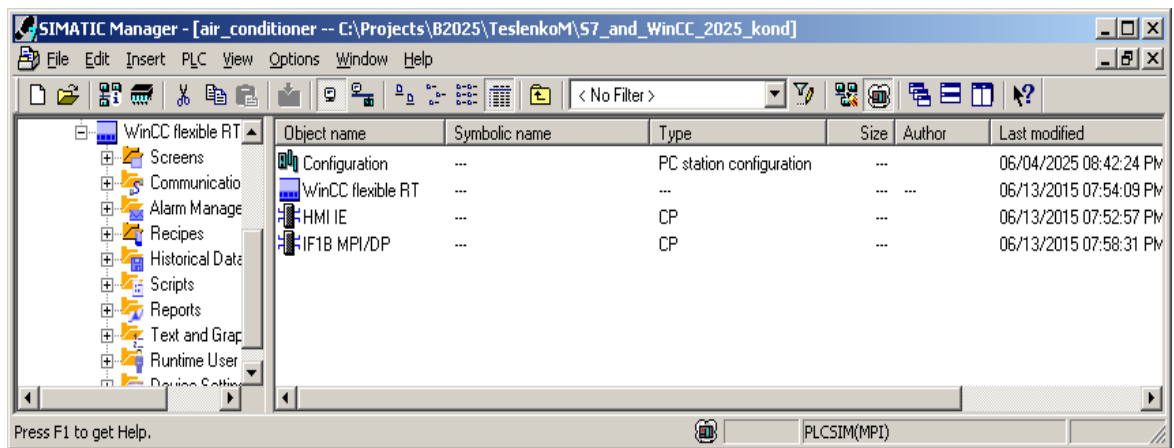


Рис. 6.7 – Загальний вигляд вікна Simatic Manager при інтеграції SCADA-системи і проекту контролера Step7.

6.2. Створення екранної панелі управління в SCADA WinCCflexible.

6.2.1. Створимо екранну панель користувача для керування мікрокліматом у приміщенні.

Наша панель управління передбачатиме:

- відображення поточного стану обладнання установки для кондиціонування;
- можливість перемикання стану обладнання установки;
- можливість ручного дистанційного керування обладнанням;
- відображення регульованих змінних температури і вологості;
- відображення повідомлень про відхилення в технологічному процесі;
- виконання налаштування САР температури і вологості у приміщенні ресторану;
- запис даних технологічного процесу (значень температури повітря у приміщенні) в БД MySQL;
- авторизований доступ користувачів до системи керування.

Параметри мережевої взаємодії між контролером та SCADA налаштовуються у розділі «Connection» (рис. 6.8).

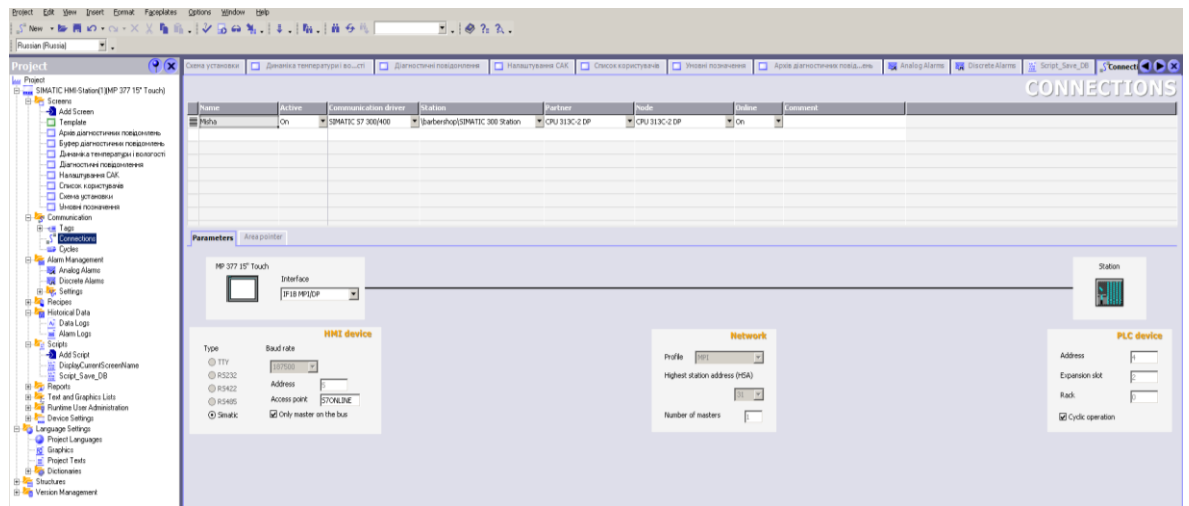


Рис. 6.8 – Вікно налаштування мережевого зв'язку між контролером та SCADA.

Далі створимо список тегів, які будуть використовуватися в ході роботи. Фрагмент вікна з тегами ПІД регулятора температури повітря у приміщенні ресторану наведено на рис. 6.9.

Name	Display name	Connection	Data type	Symbol	Address	Array elements	Acquisition cycle	Comment	Data log	Logging acquisition mode	Logging cycle
HMI_SAU_DB_PID_TL_TD		Maha	Time	TD	DB 2 DBD 26	1	1 s	час передавання	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_TM_LAG		Maha	Time	TM_LAG	DB 2 DBD 30	1	1 s	час непрацездатності	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_PV		Maha	Real	PV	DB 2 DBD 10	1	5 s	поточне значення	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_U		Maha	Real	U	DB 2 DBD 14	1	1 s	кратковільне значення	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_MAN_ON		Maha	Bool	MAN_ON	DB 2 DBD 0.0	1	1 s	опинял перемикач	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_TI		Maha	Time	TI	DB 2 DBD 22	1	1 s	час паузи	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_GAIN		Maha	Real	GAIN	DB 2 DBD 18	1	1 s	коэффициент передат.	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_SP_INT		Maha	Real	SP_INT	DB 2 DBD 6	1	1 s	задане значення	<undefined>	Cyclic continuous	<undefined>
HMI_SAU_DB_PID_TL_MAN		Maha	Real	MAN	DB 2 DBD 2	1	1 s	опинял нажатие	<undefined>	Cyclic continuous	<undefined>

Рис. 6.9 - Список тегів для ПІД регулятора температури повітря у приміщенні ресторану.

Далі створюємо декілька екранних форм для оператора. Створимо екранну форму «Схема установки» (рис. 6.10). Це головна екранна форма з зображенням мнемосхеми установки для кондиціювання повітря, стану обладнання, значень основних параметрів процесу, динаміки зміни температури та вологості у приміщенні. Далі на рисунках наведено декілька вікон.

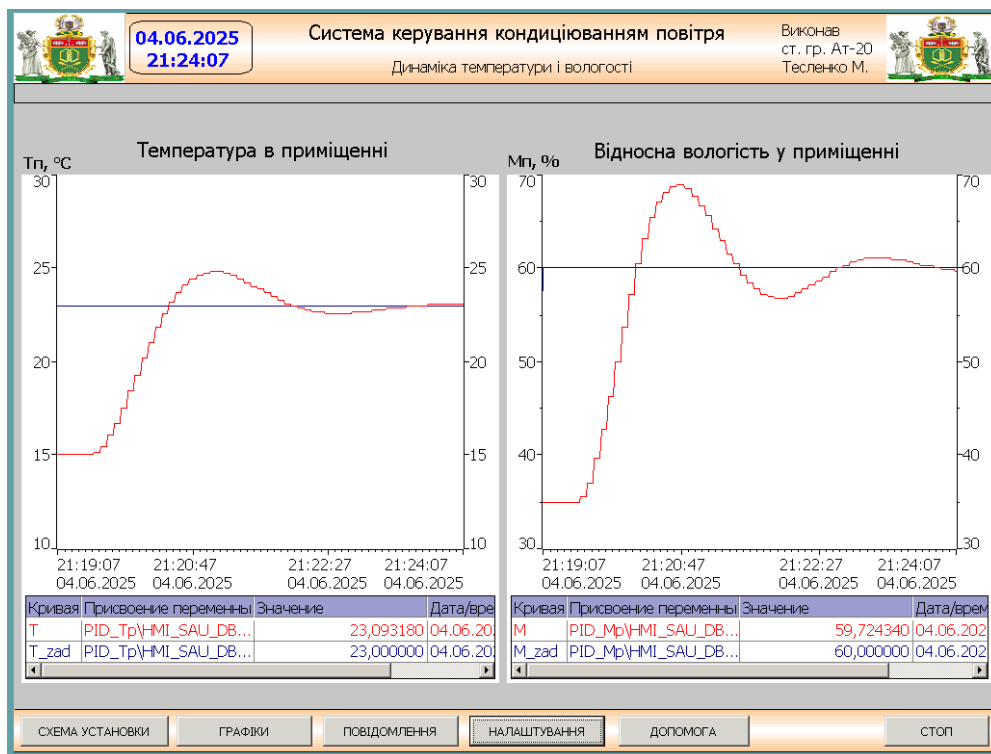
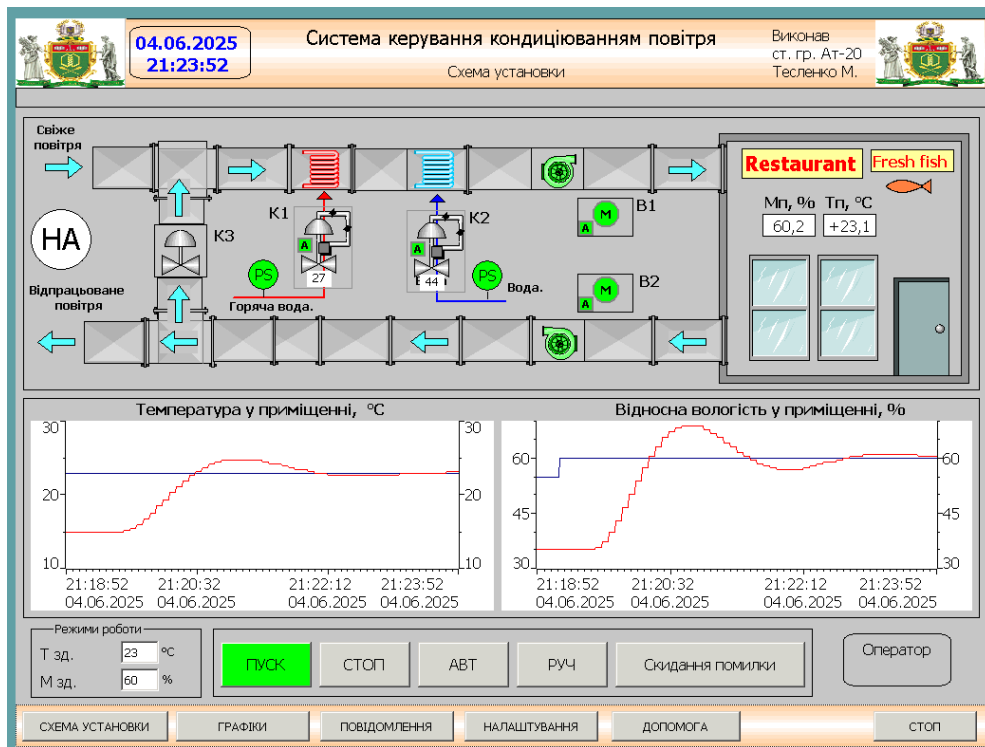


Рис. 6.10 – Екранні форми «Схема установки» та «Графіки».

Нижче на рисунках в якості прикладу наведено зображення екранних форм «Повідомлення» та «Налаштування».



04.06.2025

21:24:30

Система керування кондиціонуванням повітря

Виконав
ст. гр. Ат-20
Тесленко М.



Буфер діагностичних повідомлень

Буфер робочих повідомлень

№	Время	Дата	Состояние	Текст	QGR
2	21:20:18	04.06.2025	(C)D	Температура повітря нижче заданого значення	0
3	21:20:18	04.06.2025	C	Вологість повітря вище заданого значення	0
4	21:20:05	04.06.2025	(C)D	Вологість повітря нижче заданого значення	0
3	21:19:14	04.06.2025	C	Двигун M1 включений	0
4	21:19:14	04.06.2025	(C)D	Двигун M1 виключений	0
4	21:18:28	04.06.2025	C	Двигун M1 виключений	0
2	21:18:28	04.06.2025	C	Температура повітря нижче заданого значення	0
4	21:18:28	04.06.2025	C	Вологість повітря нижче заданого значення	0

Буфер повідомлень про несправності

Очистити буфер

Архів повідомлень

Очистити буфер

СХЕМА УСТАНОВКИ

ГРАФІКИ

ПОВІДОМЛЕННЯ

НАЛАШТУВАННЯ

ДОПОМОГА

СТОП

Рис. 6.11 – Екранна форма «Повідомлення».



04.06.2025

21:24:52

Система керування кондиціонуванням повітря

Виконав
ст. гр. Ат-20
Тесленко М.



Налаштування САК

ПІД-регулятор

Температура у приміщенні

Тк °C

Тк_зад °C

Керуючий вплив % х.р.о.

Вибір режиму

Кер. вплив ручний режим

% х.р.о.

GAIN %х.р.о./°C

TI с

TD с

TM_LAG мс

ПІД-регулятор

Відносна вологість у приміщенні

Мк %

Мк_зад %

Керуючий вплив %х.р.о.

Вибір режиму

Кер. вплив ручний режим

%х.р.о.

GAIN %х.р.о./%

TI с

TD с

TM_LAG мс

КОРИСТУВАЧІ

ЗАВЕРШИТИ СЕАНС КОРИСТУВАЧА

СХЕМА УСТАНОВКИ

ГРАФІКИ

ПОВІДОМЛЕННЯ

НАЛАШТУВАННЯ

ДОПОМОГА

СТОП

Рис. 6.12 – Екранна форма «Налаштування».

6.3. Реалізація обміну даними між SCADA та БД MySQL.

6.3.1. Організація обміну даними між SCADA-системою WINCCflexible і БД MySQL/MariaDB.

Для початку роботи дізнаємося яка БД використовується сервером (рис. 6.13).

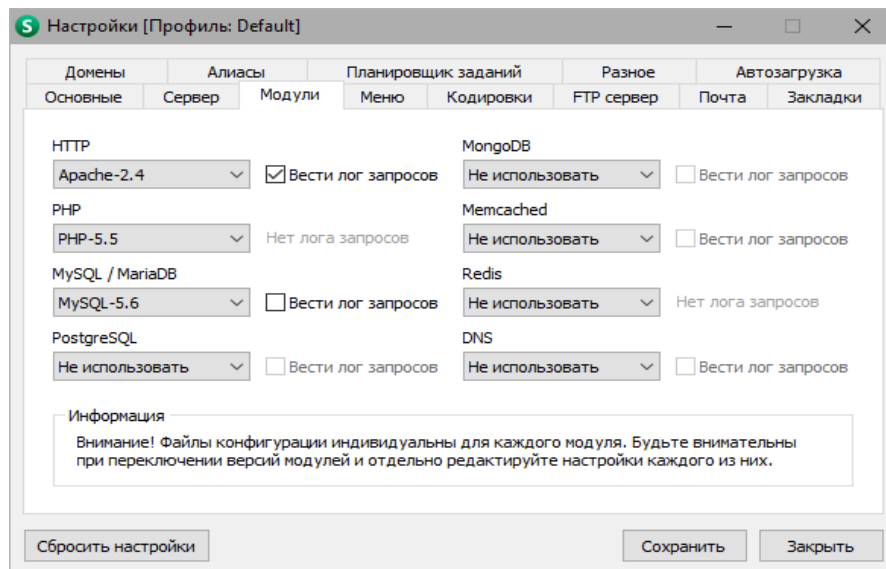


Рис. 6.13 – Налаштування веб-сервера и БД.

Створимо БД для зберігання даних.

В нашому випадку буде використовуватися БД веб-додатку розробленого під час виконання курсової роботи.

Створимо таблицю для зберігання даних технологічного процесу (температури повітря у приміщенні ресторану), які буде надсилати SCADA-система (рис. 6.14).

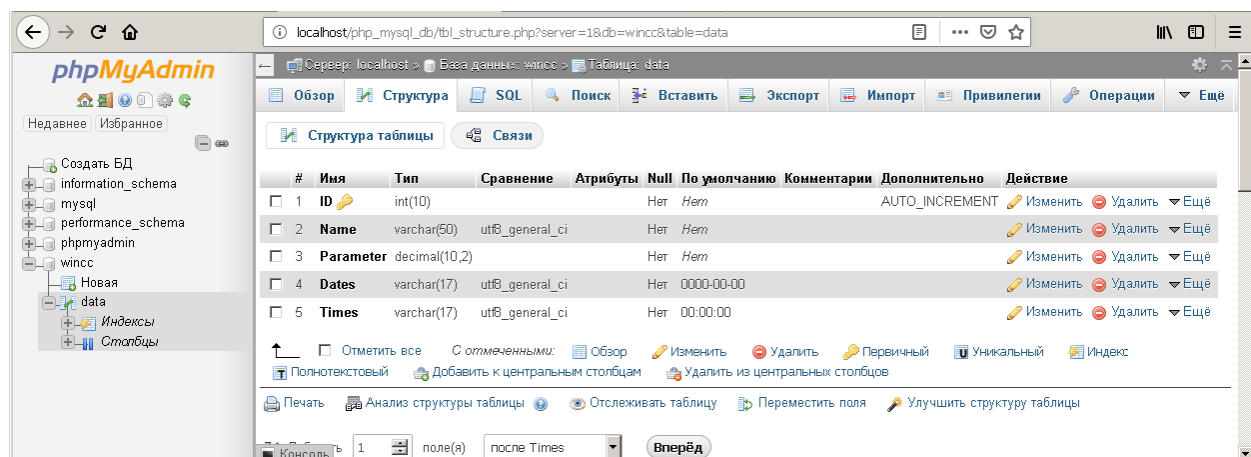


Рис. 6.14 – Створена таблиця “DATA”.

Створимо об’єкт джерела даних odbs для обміну даними SCADA системи та веб-додатком (рис. 6.15, 6.16).

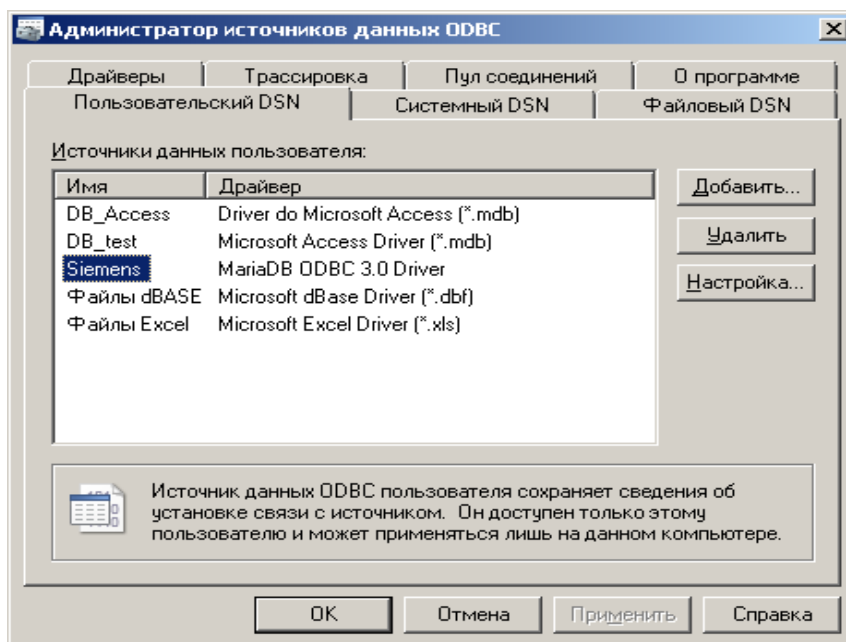


Рис. 6.15 - Створений об'єкт джерела даних ODBC.

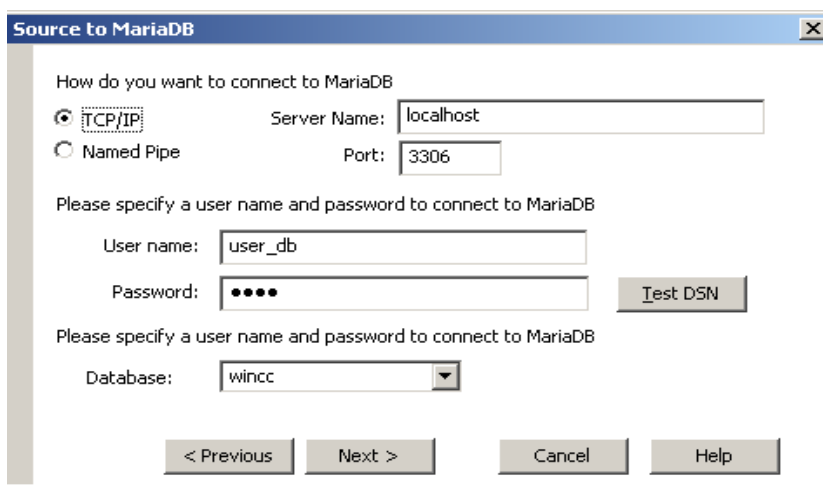


Рис. 6.16 - Створений об'єкт джерела даних ODBC. Підключення до бази даних wincc.

Для запису даних зі SCADA системи в базу даних необхідно створити програмний скрипт. Цей скрипт буде записувати необхідні данні у базу через створене джерело даних ODBC .

Створимо скрипт на запис даних в БД (рис. 6.17, 6.18).

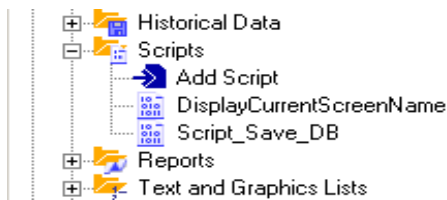


Рис. 6.17 - Список всіх створених скриптів використаних в проекті.

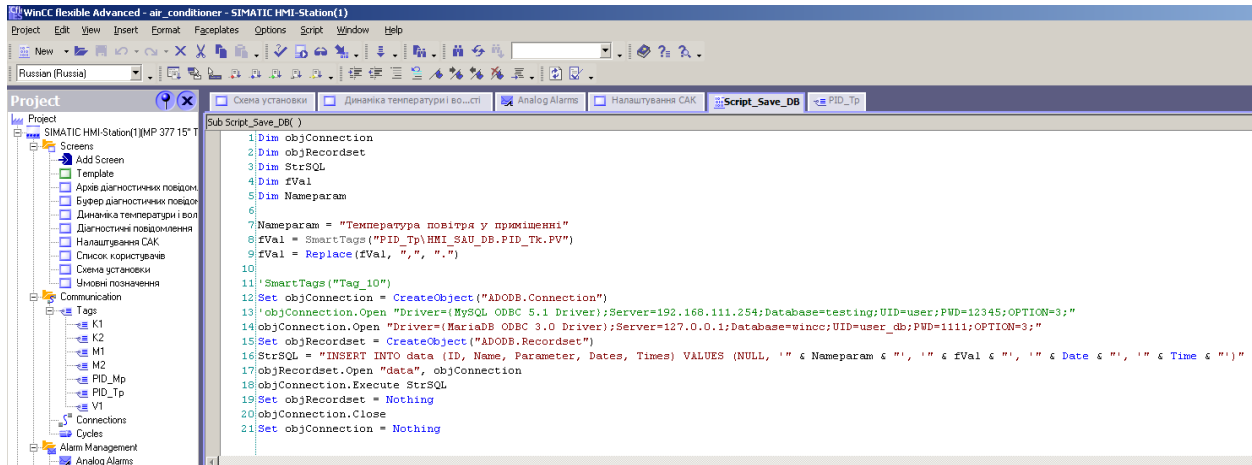


Рис. 6.18 - Скрипт для запису даних по температурі повітря в БД.

Налаштуємо наш проект і зробимо так що запис у базу даних буде проходити при кожній зміні тегу температури повітря. Опитування тегу будемо проводити з кроком 5 секунд щоб зменшити обсяг даних що потраплять у базу.

В моєму випадку SCADA-система буде надсилати данні в БД про вихідну величину об'єкту керування (температура повітря у приміщенні ресторану).

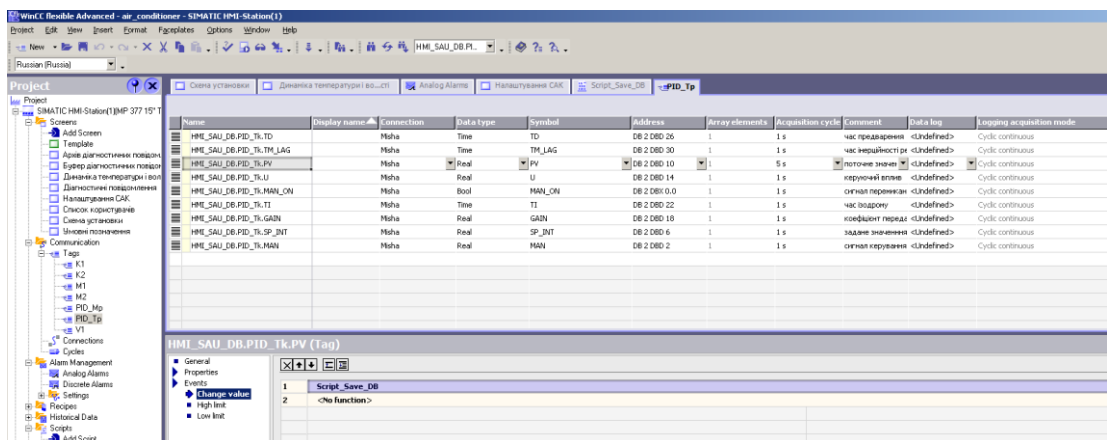
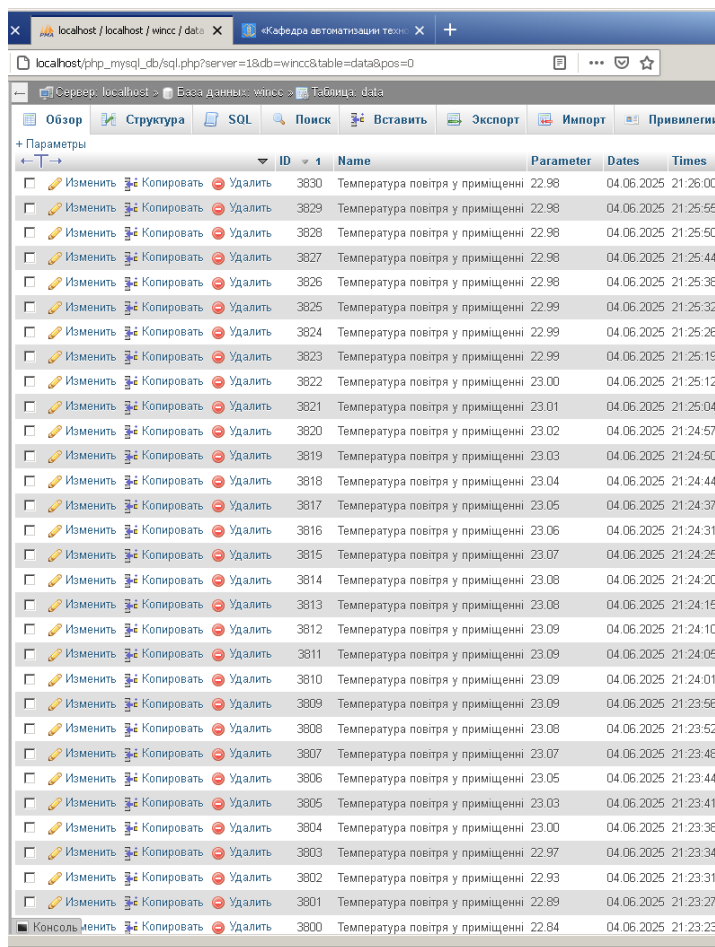


Рис. 6.19 – Фрагмент вікна прив'язки розробленого скрипта до тегу температура повітря у приміщенні ресторану.

Перевіримо роботу нашого скрипта і перевіримо записи у базі даних (рис. 6.20).



ID	Name	Parameter	Dates	Times
3830	Температура повітря у приміщенні	22.98	04.06.2025	21:26:00
3829	Температура повітря у приміщенні	22.98	04.06.2025	21:25:55
3828	Температура повітря у приміщенні	22.98	04.06.2025	21:25:50
3827	Температура повітря у приміщенні	22.98	04.06.2025	21:25:44
3826	Температура повітря у приміщенні	22.98	04.06.2025	21:25:38
3825	Температура повітря у приміщенні	22.99	04.06.2025	21:25:32
3824	Температура повітря у приміщенні	22.99	04.06.2025	21:25:26
3823	Температура повітря у приміщенні	22.99	04.06.2025	21:25:19
3822	Температура повітря у приміщенні	23.00	04.06.2025	21:25:12
3821	Температура повітря у приміщенні	23.01	04.06.2025	21:25:04
3820	Температура повітря у приміщенні	23.02	04.06.2025	21:24:57
3819	Температура повітря у приміщенні	23.03	04.06.2025	21:24:50
3818	Температура повітря у приміщенні	23.04	04.06.2025	21:24:44
3817	Температура повітря у приміщенні	23.05	04.06.2025	21:24:37
3816	Температура повітря у приміщенні	23.06	04.06.2025	21:24:31
3815	Температура повітря у приміщенні	23.07	04.06.2025	21:24:25
3814	Температура повітря у приміщенні	23.08	04.06.2025	21:24:20
3813	Температура повітря у приміщенні	23.08	04.06.2025	21:24:15
3812	Температура повітря у приміщенні	23.09	04.06.2025	21:24:10
3811	Температура повітря у приміщенні	23.09	04.06.2025	21:24:05
3810	Температура повітря у приміщенні	23.09	04.06.2025	21:24:01
3809	Температура повітря у приміщенні	23.09	04.06.2025	21:23:56
3808	Температура повітря у приміщенні	23.08	04.06.2025	21:23:52
3807	Температура повітря у приміщенні	23.07	04.06.2025	21:23:48
3806	Температура повітря у приміщенні	23.05	04.06.2025	21:23:44
3805	Температура повітря у приміщенні	23.03	04.06.2025	21:23:41
3804	Температура повітря у приміщенні	23.00	04.06.2025	21:23:38
3803	Температура повітря у приміщенні	22.97	04.06.2025	21:23:34
3802	Температура повітря у приміщенні	22.93	04.06.2025	21:23:31
3801	Температура повітря у приміщенні	22.89	04.06.2025	21:23:27
3800	Температура повітря у приміщенні	22.84	04.06.2025	21:23:23

Рис. 6.20 - Результат запису даних в БД

6.3.2. Створимо в нашому веб-додатку сторінку для відображення параметрів ТП.

На сторінці з параметрами ТП передбачимо вивід інформації з БД в табличному і в графічному вигляді.

Скрипт сторінки для відображення даних ТП (див. додаток 1, ст.21).

Скрипт модулю побудови графіка (див. додаток 2, ст.23).

Перевіримо роботу нашого веб-додатку та SCADA-системи.

Завантажуємо в пам'ять написані програми для контролера та запускаємо симуляцію його роботи.

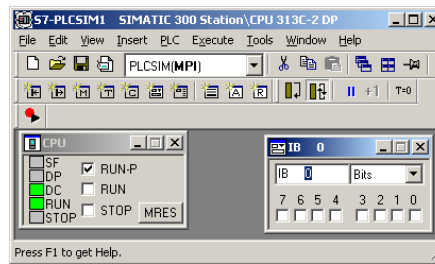


Рис.6.22 – Вікно симулятора контролера.

Запускаємо SCADA-систему. Запускаємо систему керування кондиціонуванням повітря у приміщенні ресторану і бачимо що значення температури записуються у базу даних а потім відображуються у табличному вигляді на сторінці веб додатку.

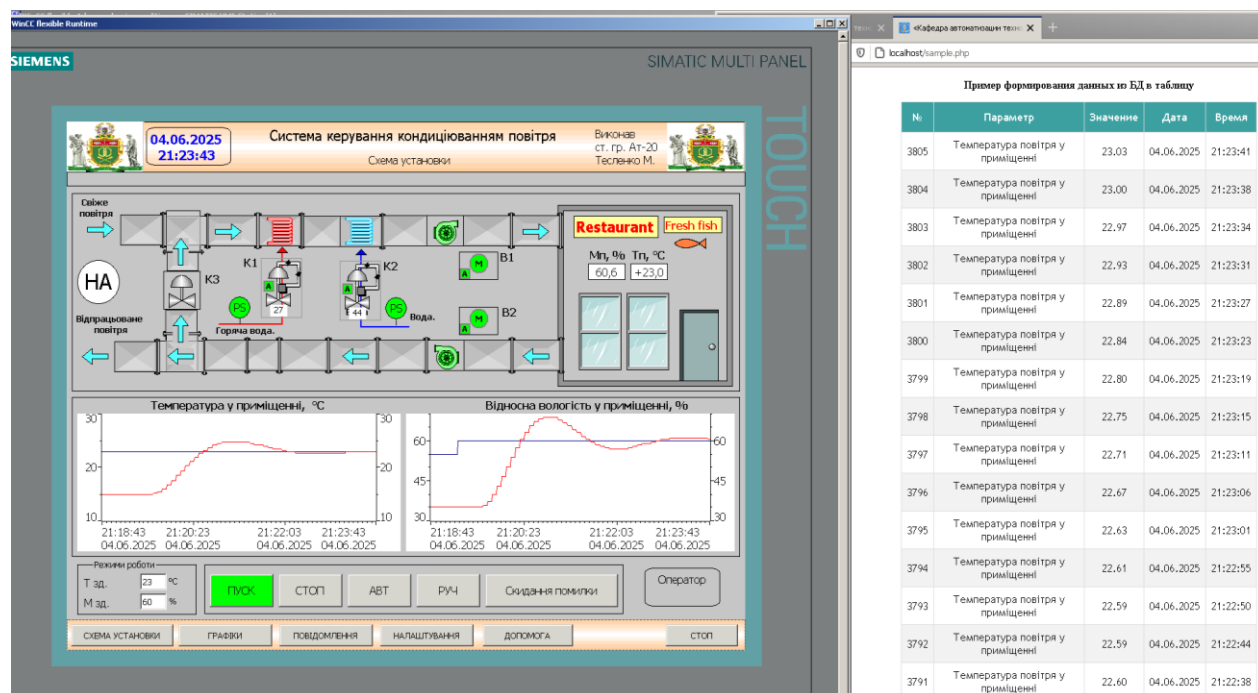


Рис.6.23 – Вікно SCADA-системи та сторінки веб-додатку.

Після порівняння отриманих даних з веб-додатку та SCADA-системи, ми можемо зробити висновок що, вони ідентичні. Тобто запис даних про зміну температури повітря в приміщенні ресторану у базу виконується без помилок. Таким чином данні через СУБД з рівня АСУТП можуть передаватися у програмне забезпечення систем керування загального виробничого рівня для можливості подальшого аналізу.

РОЗДІЛ 7. ВИБІР ТЕХНІЧНИХ ЗАСОБІВ, МЕТА ВИКОРИСТАННЯ І РЕАЛІЗАЦІЯ ЇХ ЗВ'ЯЗКУ ІЗ СТВОРЕНИМ ВЕБ-ДОДАТКОМ.

7.1 Вибір технічних засобів

- Під час створення веб-додатку для автоматизації бронювання столиків були обрані наступні технічні засоби:

- **Мова програмування і фреймворк:** Python + Flask – забезпечують легкість розробки, модульність коду та швидке прототипування. Обрана саме ця комбінація для підтримки RESTful API та підключення до БД.

- **СУБД:** MySQL – реляційна, з підтримкою зв'язків (foreign key), індексації та масштабованості.

- **Бібліотека для роботи з БД:** mysql-connector-python – забезпечує надійне та безпечне підключення, виконання запитів, закриття сесій та обробку помилок.

- **Веб-сервер:** початково — вбудований Flask, у продакшн-середовищі — Nginx + uWSGI або Gunicorn для забезпечення високої продуктивності і масштабованості.

- **Frontend:** HTML5 + CSS3 + Bootstrap 5.3.0 — для кросбраузерності, адаптивності та зручного інтерфейсу.

7.2 Мета використання кожного засобу

Засіб

Мета використання

Python + Flask

Швидкий випуск MVP, висока гнучкість,

Засіб	Мета використання
	модульний підхід
MySQL	Стійке зберігання даних, забезпечення цілісності й зв'язності
mysql-connector-python	Оптимізована взаємодія з БД, ефективне керування з'єднаннями
Nginx + uWSGI/Gunicorn	Надійне обслуговування запитів, масштабованість
Bootstrap	Гнучкий UI, адаптація під різні пристрої

7.3 Реалізація зв'язку технічних засобів

1. Підключення до БД

```
def get_db_connection():
    conn = mysql.connector.connect(host='localhost', user='root',
    password='...', database='restaurant_booking')
    return conn
```

2. Обробка HTTP-запитів (Flask)

- При GET і POST запитах для бронювання створюється/оновлюється запис в БД через `get_db_connection()`.

3. Валідація та логіка

- Перевірка конфліктів бронювання (мін. інтервал 3 години) реалізована на рівні функції.
- Логіка перетворення типів MySQL → Python використовується для перевірки часових діапазонів.

4. Frontend–Backend взаємодія

- API-контролери у Flask повертають HTML-шаблони (через Jinja2) та відповіді JSON/AJAX для інтерактивності.
- Інтерфейс клієнтів адаптивно показує статуси столиків (вільний/заброньований), доступні дати та час.

5. Вирішення масштабу і навантаження

- На продакшн-середовищі використаний Nginx для балансування навантаження та ssl-шифрування.
- uWSGI обробляє запити Flask, налаштування воркерів гарантує підтримку ≥ 100 одночасних користувачів.

7.4 Результати і переваги інтеграції

- **Безшовна взаємодія frontend–backend** – досягається за допомогою чітко визначених маршрутів у Flask та зв'язків у MySQL.
- **Модульність і гнучкість** – архітектура дозволяє легко вводити нові функції (наприклад API для мобільних клієнтів).
- **Захищені підключення до БД** – завдяки закриттю сесій у кожному запиті, знижується ризик зависань.
- **Легке масштабування** – вбудований стік Nginx + uWSGI дозволяє просто збільшувати кількість робочих процесів в залежності від навантаження.
- **Адаптивний інтерфейс** – Bootstrap забезпечує зручність користування на мобільних пристроях та різних браузерах.

РОЗДІЛ 8. ВИБІР І ЗАСТОСУВАННЯ МЕТОДІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ (ІАД) ДЛЯ УПРАВЛІННЯ БІЗНЕС-ПРОЦЕСАМИ

8.1. Значення ІАД у сфері ресторанного бізнесу

Інтелектуальний аналіз даних (ІАД), або Data Mining, відіграє ключову роль у сучасному управлінні рестораном, дозволяючи виявляти приховані закономірності, прогнозувати поведінку клієнтів та оптимізувати внутрішні бізнес-процеси. Завдяки ІАД можливо підвищити ефективність обслуговування, зменшити витрати, поліпшити якість страв, а також сформувати персоналізовані пропозиції для гостей.

До основних задач, які вирішуються за допомогою ІАД у ресторані, належать:

- аналіз попиту на страви та напої;
- прогнозування відвідуваності;
- оцінка ефективності персоналу;
- оптимізація закупівель і складу запасів.

8.2. Джерела даних для ІАД у ресторані

Для ефективного застосування методів ІАД необхідно мати якісні, структуровані та релевантні дані. У ресторані такими джерелами можуть бути:

- системи автоматизованого обліку замовлень (POS-системи);

- дані з CRM-систем (контактна інформація клієнтів, історія замовлень);
- результати онлайн-опитувань і відгуки;
- дані з камер відеоспостереження (у поєднанні з системами розпізнавання обличчя);
- інформація з кулінарної лінії (час приготування страв, залишки на складі);
- погодинна статистика продажів.

8.3. Вибір методів інтелектуального аналізу даних

Для потреб ресторану доцільно обрати кілька основних методів ІАД:

1. **Кластеризація (Cluster Analysis)** – використовується для групування клієнтів за схожими характеристиками поведінки або смаковими вподобаннями. Наприклад, можна виділити групу клієнтів, які часто замовляють вегетаріанські страви або відвідують ресторан у будні дні.
2. **Асоціативний аналіз (Association Rules Mining)** – дозволяє виявити взаємозв'язки між стравами. Наприклад, якщо клієнт замовляє бургер, з великою ймовірністю він також обере картоплю фрі.
3. **Прогнозування (Predictive Modeling)** – застосовується для оцінки майбутньої відвідуваності (наприклад, у святкові дні) або прогнозу обсягу продажів окремих страв.

4. **Аналіз відхилень (Anomaly Detection)** – дозволяє виявити нестандартну поведінку (наприклад, різке зниження кількості замовлень певної страви, що може свідчити про проблеми з якістю).

5. **Регресійний аналіз** – використовується для моделювання впливу факторів (день тижня, погода, акції) на обсяг продажів.

8.4. Приклад використання: аналіз продажів і оптимізація меню

Для демонстрації ефективності ІАД розглянемо приклад аналізу продажів за останні 3 місяці. За допомогою алгоритму Apriori були знайдені часті поєднання страв, які клієнти замовляють разом. Наприклад:

=Комбінація страв	Частота	Достовірність
Бургер + картопля фрі	340	82%
Паста + салат Цезар	210	68%
Піца + морс	185	74%

Ці дані дали змогу сформуванати комбо-пропозиції, що підвищили середній чек на 12%.

Також кластеризація клієнтів на основі часу відвідувань дозволила створити спеціальні обідні меню для офісних працівників, які приходять у будні з 12:00 до 14:00.

8.5. Інструменти для реалізації ІАД

Для реалізації ІАД у ресторані можуть бути використані такі програмні засоби:

- **Power BI, Tableau** – для візуалізації та інтерактивного аналізу даних;
- **Python (бібліотеки pandas, scikit-learn, matplotlib)** – для програмного аналізу та побудови моделей;

- **SQL-сервери** – для зберігання та обробки даних;
- **RapidMiner, Orange** – для візуального створення моделей без програмування.

8.6. Висновки

Застосування методів інтелектуального аналізу даних у ресторанному бізнесі є важливим інструментом підвищення конкурентоспроможності та якості обслуговування. За допомогою ІАД можна більш глибоко розуміти клієнтів, прогнозувати попит, оптимізувати роботу персоналу та приймати обґрунтовані управлінські рішення. У довгостроковій перспективі це забезпечує зростання прибутковості та формує лояльність клієнтів.

РОЗДІЛ 9. ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

9.1 Загальні положення

У сучасних умовах високої конкуренції в галузі ресторанного бізнесу важливим чинником підвищення ефективності роботи є автоматизація основних процесів обслуговування клієнтів. Одним із ключових напрямків цифрової трансформації ресторану є впровадження системи онлайн-бронювання столиків.

Створене програмне забезпечення, яке дозволяє клієнтам самостійно обирати вільні місця у зручний час, а адміністраторам — бачити, редагувати та контролювати поточні бронювання, має на меті не лише покращити якість обслуговування, а й забезпечити економічну вигоду підприємству.

Метою даного розділу є обґрунтування економічної доцільності впровадження розробленої автоматизованої системи бронювання на основі аналізу витрат, вигод і строку окупності інвестицій.

9.2 Очікувані вигоди від впровадження

Автоматизована система бронювання дозволяє досягти наступних покращень у роботі ресторану:

1. Оптимізація навантаження на персонал — зниження кількості телефонних дзвінків, які обробляє адміністратор.
2. Підвищення точності обліку бронювань — усунення помилок при записі замовлень.
3. Зменшення простоїв столиків — ефективніше планування посадки клієнтів.

4. Аналітика та звітність — можливість отримувати інформацію про популярні години, дні, завантаженість тощо.

5. Підвищення лояльності клієнтів — завдяки зручності бронювання через сайт з будь-якого пристрою.

9.3 Вихідні дані для розрахунків

Для оцінки економічної ефективності приймаємо наступні параметри:

- Потужність ПК: 0,4 кВт/год
- Тариф на електроенергію: 1,68 грн/кВт· год
- Оплата праці розробника: 700 грн/день
- Тривалість розробки: 6 тижнів, 5 днів/тиждень, 8 годин/день
- Кількість сторінок друку: 300 аркушів по 1,5 грн
- Вартість серверного хостингу: 200 грн/міс.
- Інші засоби автоматизації: планшет адміністратору (7000 грн), маршрутизатор (2000 грн)

9.4 Розрахунок витрат на створення ПЗ

1. Загальний робочий час:

$$\text{ЧР} = 6 \text{ тижн.} \times 5 \text{ днів} \times 8 \text{ год.} = 240 \text{ годин}$$

2. Витрати на електроенергію:

$$\text{В}_{\text{ел}} = 240 \text{ год.} \times 0,4 \text{ кВт} \times 1,68 \text{ грн/кВт} = 161,28 \text{ грн}$$

3. Заробітна плата розробника:

$$\text{ЗП} = 700 \text{ грн/день} \times 30 \text{ днів} = 21000 \text{ грн}$$

4. Єдиний соціальний внесок (22%):

$$\text{ЄСВ} = 21000 \times 0,22 = 4620 \text{ грн}$$

5. Витрати на друк:

$$\text{Вдрук} = 300 \times 1,5 = 450 \text{ грн}$$

6. Вартість серверного хостингу (1 рік):

$$\text{Хостинг} = 200 \times 12 = 2400 \text{ грн}$$

7. Інвестиції в обладнання:

$$\text{Інв} = 7000 + 2000 = 9000 \text{ грн}$$

8. Загальні інвестиції:

$$\begin{aligned} \text{ІС} &= \text{Вел} + \text{ЗП} + \text{ЄСВ} + \text{Вдрук} + \text{Хостинг} + \text{Інв} = \\ &= 161,28 + 21000 + 4620 + 450 + 2400 + 9000 = 37631,28 \text{ грн} \end{aligned}$$

9.5 Розрахунок економічного ефекту

Очікуване скорочення витрат:

- Зниження витрат на обслуговування бронювань: 30 хв/бронювання $\rightarrow$ 10 хв.

- В середньому 20 бронювань/день $\times$ 20 хв економії = 400 хв = $\sim 6,67$ год/день.

- За ставкою 80 грн/год: щоденна економія = 533,6 грн.

- За 300 робочих днів на рік:

$$E_{\text{річна}} = 533,6 \times 300 = 160080 \text{ грн}$$

9.6 Розрахунок показників ефективності

1. Чистий прибуток після оподаткування (18%):

$$\text{ЧП} = (E_{\text{річна}} - \text{Амортизація}) \times (1 - 0,18)$$

Припустимо, амортизація 60%:

$$A_{\text{м}} = IC \times 0,6 = 37631,28 \times 0,6 = 22578,77$$

$$\text{ЧП} = (160080 - 22578,77) \times 0,82 = 112878,6 \text{ грн}$$

2. Чистий грошовий потік:

$$\text{ЧГП} = \text{ЧП} + A_{\text{м}} = 112878,6 + 22578,77 = 135457,37 \text{ грн}$$

3. Строк окупності інвестицій:

$$T_{\text{окуп}} = IC / \text{ЧГП} = 37631,28 / 135457,37 \approx 0,28 \text{ року} \approx 3,4 \text{ місяці}$$

Висновки

Впровадження автоматизованої системи бронювання столиків для ресторану є економічно доцільним проектом. Отриманий чистий грошовий потік — понад 135 тис. грн за рік, а термін окупності становить лише 3,4 місяці, що свідчить про високу інвестиційну привабливість розробки.

Окрім фінансових вигод, система забезпечує покращення сервісу, зниження ризиків помилок, аналітичну підтримку для прийняття управлінських рішень та посилення конкурентних переваг ресторану. Надалі можливе розширення функціоналу системи — наприклад, додавання онлайн-оплати, нагадувань клієнтам, інтеграції з CRM або мобільним застосунком.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра на основі проведених досліджень, аналізу та інженерного проєктування розв'язано практичну задачу розробки та впровадження інтегрованої системи автоматизованого управління бізнес-процесами та технологічними параметрами ресторану преміумсегмента «Хмари» (м. Одеса). За результатами виконання роботи зроблено такі висновки:

1. За результатами аналітичного огляду та моделювання «As-Is» проаналізовано організаційно-технічну структуру ресторану «Хмари». Виявлено «вузькі місця» в логіці обслуговування клієнтів (ручна обробка телефонних резервувань, затримки у комунікації) та технологічному забезпеченні залів (відсутність автоматизованого контролю мікроклімату). За допомогою методологій бізнес-моделювання у нотаціях eEPC та BPMN побудовано модель поточних процесів, що дозволило чітко визначити зони для автоматизації та цифровізації.

2. Щодо реінжинірингу процесів та моделювання «To-Be» розроблено цільові моделі процесів «To-Be», які передбачають повну інтеграцію клієнтського вебсервісу із внутрішніми алгоритмами закладу. Перехід на автоматизовану обробку заявок дозволяє оптимізувати робочий час персоналу, виключити дублювання замовлень (овербукінг) та мінімізувати вплив людського фактора. На основі цих моделей сформовано вимоги до структури бази даних під керуванням СКБД MySQL.

3. В частині розробки програмного та інформаційного забезпечення спроектовано реляційну базу даних MySQL, яка забезпечує надійне збереження даних про клієнтів, столи, замовлення та меню, з дотриманням правил цілісності та нормалізації. Розроблено архітектуру вебдодатка для онлайн-бронювання, створено клієнтський та адміністративний інтерфейси

користувача, які адаптовані під бізнес-потреби закладу HoReCa та інтегруються з аналітичними модулями.

4. Щодо автоматизації технологічного процесу для забезпечення комфорту відвідувачів та дотримання санітарних норм спроектовано локальну систему автоматичного управління (CAУ) мікрокліматом (вентиляція, кондиціонування, опалення). В середовищі Step7 (SIMATIC Manager) розроблено алгоритми та логіку керування для контролерів Siemens. Створено людино-машинний інтерфейс (HMI/SCADA), що забезпечує оператору візуалізацію параметрів середовища в реальному часі та ведення архівів. Налагоджено механізм передачі ключових технологічних даних у центральну базу сайту ресторану.

5. В частині інтелектуального аналізу даних для підвищення конкурентоспроможності бізнесу застосовано методи інтелектуального аналізу даних. Реалізовано алгоритми для прогнозування попиту на основі ретроспективних даних замовлень та оптимізації структури меню ресторану, що дає змогу менеджменту гнучко реагувати на зміни ринку та оптимізувати закупівлю сировини.

6. За результатами техніко-економічного обґрунтування проведено комплексний розрахунок економічної ефективності проєкту. Обчислено обсяг капітальних інвестицій на закупівлю обладнання, розробку ПЗ та монтаж системи, визначено річні операційні витрати. Розраховані показники ефективності (чистий приведений дохід, індекс рентабельності та термін окупності) підтверджують високу фінансову доцільність впровадження розробленої системи автоматизації на підприємстві «Хмари».

СПИСОК ЛІТЕРАТУРИ

1. Автоматизація технологічних і бізнес-процесів (Automation of technological and business processes) : наук.-виробн. журн. Одеса : ОНТУ, 2023.
1. Том 10, № 1. 120 с.
2. ТОВ «Ресторан ХМАРИ» <https://oblaka.choiceqr.com/menu>
3. <https://oblaka.choiceqr.com/menu/section:menuyu>
4. Бобровський С. В. Програмування на PHP: Вступний курс. Київ : Освіта, 2019. 256 с.
5. Розенталь І. М. Основи веб-розробки: HTML, CSS, JavaScript. Харків : Фоліо, 2020. 384 с.
6. Грін Дж. MySQL: Повний довідник. Львів : Модерн, 2018. 512 с.
7. Шнайдер А. Програмування на PHP та MySQL для початківців. Одеса : Південний, 2021. 352 с.
8. Документація PHP. URL: <https://www.php.net/docs.php>.
9. Документація MySQL. URL: <https://dev.mysql.com/doc/>.
10. Відкриті навчальні ресурси з веб-розробки.
11. URL: <https://www.w3schools.com/>.
12. Статті на тему безпеки веб-додатків. URL: <https://owasp.org/www-project-top-ten/>.
13. [project-top-ten/](https://owasp.org/www-project-top-ten/).
14. Вільямс М. Створення користувацького інтерфейсу для веб-додатків. Вінниця : Поділля, 2022. 200 с.90
15. Трибой В. В., Панібудль В. В., Терепенев В. В. Автоматизація виробничих процесів. Теорія подібності : навч. посіб. / за ред. В. В. Трибоя. Черкаси : ЧДТУ, 2022. 18 с.
16. Автоматизація технологічних і бізнес-процесів = Automation of technological and business processes : наук.-виробн. журн. Одеса : ОНТУ, 2023.

17. Автоматизація технологічних і бізнес-процесів = Automation of technological and business processes : наук.-виробн. журн. Одеса : ОНТУ, 2024.
18. Жигайло О. М., Дец Д. В. Основи керування бізнес-процесами : конспект лекцій / відп. за вип. В. А. Хобія ; Каф. автоматизації технологічних процесів і робототехнічних систем. Одеса : ОНАХТ, 2018. 19 с.
19. Жигайло О. М., Дец Д. В. Методичні вказівки до виконання курсової роботи з освітнього компоненту «Основи керування бізнес-процесами» / відп. за вип. В. А. Хобія. Одеса : ОНТУ, 2023. 59 с.
19. Левінський В. М. Методичні вказівки до виконання лабораторних
20. робіт з курсу «Основи САПР». Одеса : ОНТУ, 2023. 44 с.
21. Левінський В. М. Методичні вказівки до виконання самостійної роботи з курсу «Основи САПР». Одеса : ОНАХТ, 2020. 7 с.
22. Ivanova L., Kavernitskiy V., Kotlyk S. et al. Modern information technologies in scientific research and educational activities : monograph. Odessa : Iowa State University Digital Press, 2024. 273 p.
23. Optimization in Supply Chain Automation. Cambridge University Press, 2020.
24. ARIS Express— Business Process Modelling Tool. URL:
25. <https://www.ariscommunity.com/tools/aris-express>.
26. Bizagi Modeler— інструмент BPMN. URL: <https://www.bizagi.com>.

