

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

**Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки**

«До захисту в ЕК»
Директор інституту(декан факультету)
Андрій ФОРСЮК
(підпис) (ім'я та прізвище)

«10» червня 2024р.

«До захисту допущено»
Завідувач кафедри
Сергій ГРИБКОВ
(підпис) (ім'я та прізвище)

«10» червня 2024р.

**КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА**

зі спеціальності 122 «Комп'ютерні науки»
(код та назва спеціальності)

освітньо-професійної програми Комп'ютерні науки

на тему: Розроблення додатку для он-лайн замовлення продукції кафетерію

Виконав: здобувач 4 курсу, групи КН-4-2

Степанець Кирило Романович
(прізвище, ім'я, по батькові повністю) (підпис)

Керівник Харкянен Олена Валеріївна
(прізвище, ім'я та по батькові повністю) (підпис)

Консультанти _____
(ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

_____ (ім'я та прізвище) (підпис)

Рецензент _____
(ім'я та прізвище) (підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) недозволеної допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач _____
(підпис)

Київ - 2024р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
Освітній ступінь бакалавр
Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інформаційних технологій, штучного
інтелекту і кібербезпеки
Сергій ГРИБКОВ

“ 15 ” квітня 2024 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Степанця Кирила Романовича

(прізвище, ім'я, по батькові)

1. Тема роботи Розроблення додатку для он-лайн замовлення продукції
кафетерію
керівник роботи доц. Харкянен Олена Валеріївна
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від 15 квітня 2024 року № 279-кв

2. Строк подання здобувачем роботи 03.06.2024 р.

3. Вихідні дані до роботи
аналітичні та статистичні матеріали стосовно теми проекту, технічна література
з мов програмування, методичні вказівки, дані про роботу кафетерію
“SimpleBite”

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)
системний аналіз об'єкта досліджень та виявлення задач автоматизації,
технічне завдання, опис комплексу задач автоматизації, охорона
праці.

5. Перелік графічного матеріалу

1. Логічна модель бази даних

2. Фізична модель бази даних

3. Фрагменти програмного коду

4. Можливі помилки у програми

5. Відео кадри інтерфейсу

1. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	доц. Харкянен О.В.	15.01.24	24.02.24
2	доц. Харкянен О.В.	15.01.24	14.05.24
3	доц. Харкянен О.В.	15.01.24	18.05.24
4	доц. Харкянен О.В.	15.01.24	20.05.24

2. Дата видачі завдання: 15 лютого 2024 р**КАЛЕНДАРНИЙ ПЛАН**

№ п/п	Найменування робіт	Строки виконання робіт	Примітка
1	Передпроектне дослідження об'єкта автоматизації	15.01.2024 -25.02.2024	Виконано
2	Технічне завдання	18.04.2024 – 30.04.2024	Виконано
3	Розробка додатку	1.05.2024 - 14.05.2024	Виконано
4	Оформлення документації	15.05.2024 - 28.05.2024	Виконано
5	Створення презентації	29.05.2024	Виконано

Здобувач

_____ (підпис)

Степанець К.Р.

(прізвище та ініціали)

Керівник роботи

_____ (підпис)

Харкянен О.В.

(прізвище та ініціали)

АНОТАЦІЯ

Метою кваліфікаційної роботи є створення додатка для самообслуговування клієнтів кафетерію при складанні замовлень, відповідно до прийнятих міжнародних стандартів, відповідно до завдання.

Дослідження стосується діяльності кафетерію.

Предметом дослідження є – додатки які працюють на задоволення потреб клієнта в закладі харчування, зменшення очікування клієнта в черзі, прискорення прийняття замовлення та отримання прибутку.

Методи дослідження – базуються на методах розробки та імплементації до процесів діяльності кафетерію (SimpleBite) (для проведення оптимізації вже існуючого процесу), методах системного аналізу (для вивчення предметної середовища досліджень).

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, КАФЕТЕРІЙ, ОН-ЛАЙН ЗАМОВЛЕННЯ, БАЗА ДАНИХ

ABSTRACT

The aim of the qualified thesis is to create an application for self-service of cafeteria customers when placing orders, in accordance with accepted international standards, according to the task.

The study concerns the activities of the cafeteria.

The subject of the study is applications that work to meet the needs of the client in the catering establishment, reduce the client's waiting in the queue, speed up order acceptance and profit.

Research methods are based on methods of development and implementation of cafeteria (SimpleBite) activity processes (to optimize an already existing process), methods of system analysis (to study the subject environment of research).

KEYWORDS: INFORMATION SYSTEM, CAFETERIA, ONLINE ORDERING, DATABASE

ЗМІСТ Загальна

ВСТУП.....	8
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ КАФЕТЕРІЮ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ.....	9
1.1. Загальна характеристика	9
1.2. Організаційна структура кафетерію, роль і взаємодія підрозділів	9
1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів	14
1.5. Удосконалення процесу прийому та обслуговування клієнтів.....	16
1.7. Обґрунтування доцільності проектування і розроблення додатку SimpleBite	20
1.8. Розрахунок економічного ефекту від впровадження системи	22
РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ	30
2.1. Загальні положення	30
2.2. Призначення та цілі створення системи	30
2.3. Характеристика об'єкта автоматизації	31
2.4. Вимоги до системи	31
2.5. Склад і зміст робіт по створенню системи.....	36
2.6. Порядок контролю і приймання системи	37
2.7. Вимоги до складу і змісту робіт із підготовки до введення додатку в дію	37
2.8. Вимоги до документуванню	38
РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧІ АВТОМАТИЗАЦІЇ.....	39
3.1. Перелік задач.....	39
3.2. Інформаційне забезпечення системи	39
3.3. Алгоритмізація та реалізація комплексу задач автоматизації.	41
3.4. Технічне та системне забезпечення розробки.....	51
3.5. Інструкція користувача.....	54
РОЗДІЛ 4. ОХОРОНА ПРАЦІ.....	62

	7
4.1. Загальні положення	62
4.2. Вимоги безпеки перед початком роботи.....	65
4.3. Вимоги безпеки під час виконання роботи	65
4.4. Вимоги безпеки після закінчення роботи.....	66
4.5. Вимоги безпеки в аварійній ситуації	66
ВИСНОВОК	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	70
ДОДАТКИ.....	72
Додаток А. Логічна Модель Бази Даних.....	72
Додаток Б. Фізична Модель Бази Даних.....	73
Додаток В. Фрагменти Програмного Коду	74
Додаток Г. Можливі Помилки У Програмі.....	78
Додаток Д. Відеокадри Інтерфейсу	81

ВСТУП

На сьогоднішній день швидке та високоякісне обслуговування клієнтів є життєво важливим для успіху будь-якої компанії. Оскільки головним завданням будь-якого бізнесу є отримання прибутку, їм потрібно надавати високоякісні послуги, щоб привернути більше клієнтів і зробити їх більш конкурентоспроможними. З розвитком інформаційних технологій автоматизація процесів роботи закладів харчування, таких як ресторани, кафетерії, бари, кав'ярні та інші, стала можливою та актуальною.

Основні труднощі, які з'являються у закладах харчування, є наступні:

- недостатня ефективність обслуговування клієнтів;
- незадовільна якість продукції;
- конкуренція;
- часта зміна персоналу;
- недостатня діджиталізація закладу.

Таким чином, необхідно створити Додаток для онлайн замовлення, який допоможе вирішити проблеми, пов'язані з роботою даних підприємств. Використання Додатка прискорити обслуговування клієнта.

Для прикладу:

- дозволяє створювати замовлення на екрані у залі. У ресторані клієнти можуть переглядати меню та швидко вибирати потрібні страви, фільтруючи за типами: напої, закуски, десерти та основні страви, а потім робити замовлення. При цьому потрібно менше людей, що дозволить їм краще обслуговувати клієнтів закладу;
- встановити програму бонусів, яка дозволяє нараховувати бонуси постійним клієнтам та надати скидки клієнтам закладу.

Використання цього додатку дозволить підвищити якість обслуговування клієнтів і швидкість.

РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ДІЯЛЬНОСТІ КАФЕТЕРІЮ ТА ВИЯВЛЕННЯ ЗАДАЧ АВТОМАТИЗАЦІЇ

1.1. Загальна характеристика

Кафе — це заклад громадського харчування, який пропонує своїм відвідувачам широкий вибір напоїв і страв. Він може бути окремим об'єктом або частиною більшої організації, наприклад, навчального закладу, офісної будівлі або готелю [1].

Виконання замовлень клієнтів, виготовлення високоякісної продукції та отримання прибутку є головними цілями кафетерію.

Послуги кафе включають обслуговування гостей, приготування чаю, кави, безалкогольних напоїв, десертів, закусок і легких страв. Кафе також можуть використовувати спеціальні пристрої для виготовлення конкретної страви чи напою або його інгредієнтів.

1.2. Організаційна структура кафетерію, роль і взаємодія підрозділів

Організаційна структура кафетерію складається з таких підрозділів:

- Відділ керівництва. Підрозділ займається вирішенням організаційних питань, а також керівництвом та управлінням кафетерієм.
- Відділ виробництва. Цей відділ відповідає за приготування їжі та напоїв відповідно до стандартів якості та специфікацій меню.
- Відділ закупівель продуктів та сировини відповідальний за закупівлю та контроль якості різних продуктів, таких як інгредієнти та напої.
- Управління фінансами. Відповідає за фінансові операції кафетерію, включаючи бухгалтерію, бюджетування, розрахунок заробітної плати та облік доходів та витрат.

- Служба обслуговування клієнтів. Приймання замовлень, обслуговування клієнтів та забезпечення високої якості обслуговування входять до компетенції цього відділу.

Для того, щоб кафетерія працювала ефективно, усі відділи повинні працювати разом. Наприклад, відділи виробництва та обслуговування співпрацюють, щоб забезпечити якісну та швидку обробку замовлень. Відділ закупівлі взаємодіє з виробничим, щоб забезпечити наявність необхідних продуктів і матеріалів. Для планування бюджету та розподілу фінансових ресурсів фінансовий відділ співпрацює з підрозділом управління [2].

1.2.1. Загальна схема організаційної структури

Загальне керівництво, розробка стратегії розвитку та прийняття стратегічних рішень – це обов'язки директора кафедри. Він керує всіма підрозділами, забезпечує вирішення проблем і координує діяльність. Схема організаційної структури подана на рисунку 1.1.

Фінансовий сектор кафетерію контролює бухгалтер. Бюджетні плани та управління, фінансові аналізи, оподаткування та звітність є його обов'язками.

Адміністратор відповідає за керування персоналом, вирішення проблем, запитів і скарг клієнтів.

Розробка меню, контроль якості страв, розподіл обов'язків і навантаження між кухарями та кондитерами — усе це під контролем головного кухаря.

Вони співпрацюють з відділом закупівлі, щоб переконатися, що в них є все, що потрібно.

Касири обслуговують гостей кафетерії. Вони спілкуються з клієнтами, приймають платежі, приймають замовлення, видають замовлення та надають інформацію про меню та послуги.

Забезпечення своєчасної доставки страв для клієнтів залежить від виробництва та кухні.

Менеджер з закупівлі та контролер якості відповідають за закупівлю і контроль якості продуктів, необхідних для приготування страв. Вони співпрацюють з постачальниками, проводять перевірку якості товарів та доглядають за запасами і передають їх до кухарів.

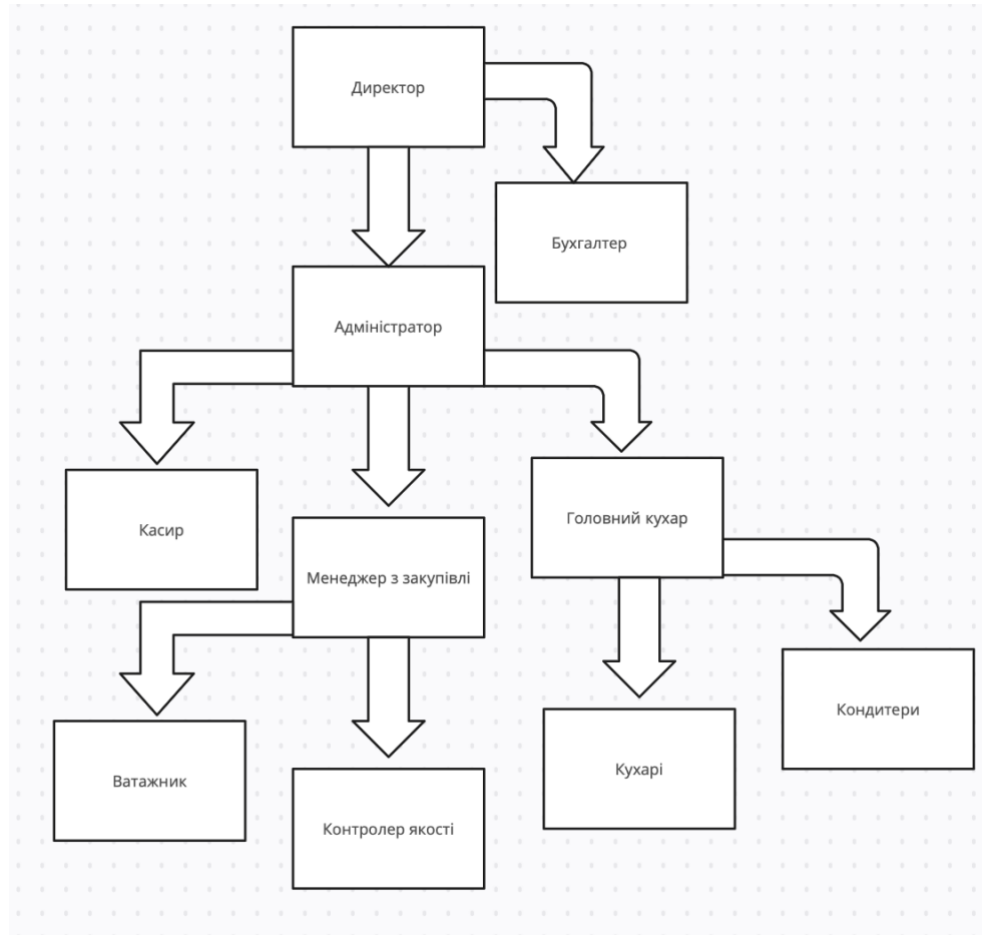


Рисунок 1.1 — Схема організаційної структури

1.2.2. Структура відділу обслуговування

Приймання замовлень, обслуговування клієнтів і забезпечення високої якості обслуговування належать до компетенції цього відділу.

Особливостями роботи та функціями відділу є:

- Адміністратор відповідає за керівництво та координацію роботи відділу обслуговування. Він забезпечує ефективне функціонування відділу, планує графіки роботи, розподіляє обов'язки та навантаження між співробітниками. Менеджер також надає директиви та інструкції персоналу,

контролює якість обслуговування, відповідає на запити та питання клієнтів та розв'язує можливі проблеми.[3]

- Касири відповідають за обслуговування клієнтів у кафетерії. Вони спілкуються з клієнтами, приймають замовлення, приймають оплату, видають замовлення, надають інформацію про меню. Обслуговування клієнтів залежить від якості та швидкості виробництва страв для задоволення клієнта.



Рисунок 1.2 — Структурна схема відділу обслуговування

Процес отримання та передачі інформації до відділу обслуговування наведено у таблиці 1.1.

Таблиця 1.1 Отримання та передача інформації відділу обслуговування

Підрозділ обслуговування	Отримання інформації	Передача інформації
Адміністратор	Директор/Головний кухар/Клієнт	Касири
Касир	Клієнти/Адміністратор	Клієнти/Адміністратор

1.3. Аналіз нинішнього стану комп'ютеризації кафетерію «SimpleBite»

В закладі для кожного підрозділу встановлено програмне забезпечення обліку даних у середовищах Microsoft Excel, 1С та СУБД Access. Це дозволяє виконувати основні завдання та функції підрозділів.

Загалом на сьогоднішній день використовуються обмежені технологічні рішення та програмне забезпечення, тому можливість самообслуговування клієнтів є обмеженою. Впровадження підсистеми самообслуговування у існуючу систему призведе до покращення сервісу.

- Використання технологій, які мають обмеження. Кафетерії не можуть використовувати більш сучасні технології, такі як касові апарати, для покращення своєї роботи.

- Відсутність процесів, які можна автоматизувати. Кафетерій не автоматизує багато завдань, таких як управління фінансами та спілкування з клієнтами. Це може призвести до більшої ручної роботи, помилок, незадоволених клієнтів та втрати популярності.

- Відсутність аналітичних інструментів. Низький рівень комп'ютеризації обмежує можливості аналізу даних та отримання цінної інформації для прийняття управлінських рішень. Відсутність систем аналітики може ускладнювати виявлення тенденцій, розуміння потреб клієнтів та покращення стратегії розвитку.

- Відсутність автоматизації в процесі обслуговування клієнтів. Клієнти можуть бути незадоволені через довгі черги та відсутність автоматизованих систем обслуговування клієнтів, таких як електронні меню, системи замовлення та оплати.

- Обмежений доступ до інформації про клієнтів. Недостатня комп'ютеризація може ускладнювати збір та аналіз даних про клієнтів. Відсутність цінної інформації про їхні замовлення, уподобання та зворотний зв'язок може перешкоджати розробці персональних пропозицій.

1.4. Розроблення функціональної моделі та аналіз існуючих бізнес-процесів

Розглянемо процес обслуговування клієнта та прийом замовлень на основі функціональної моделі, наведеної на Рисунку 1.3.

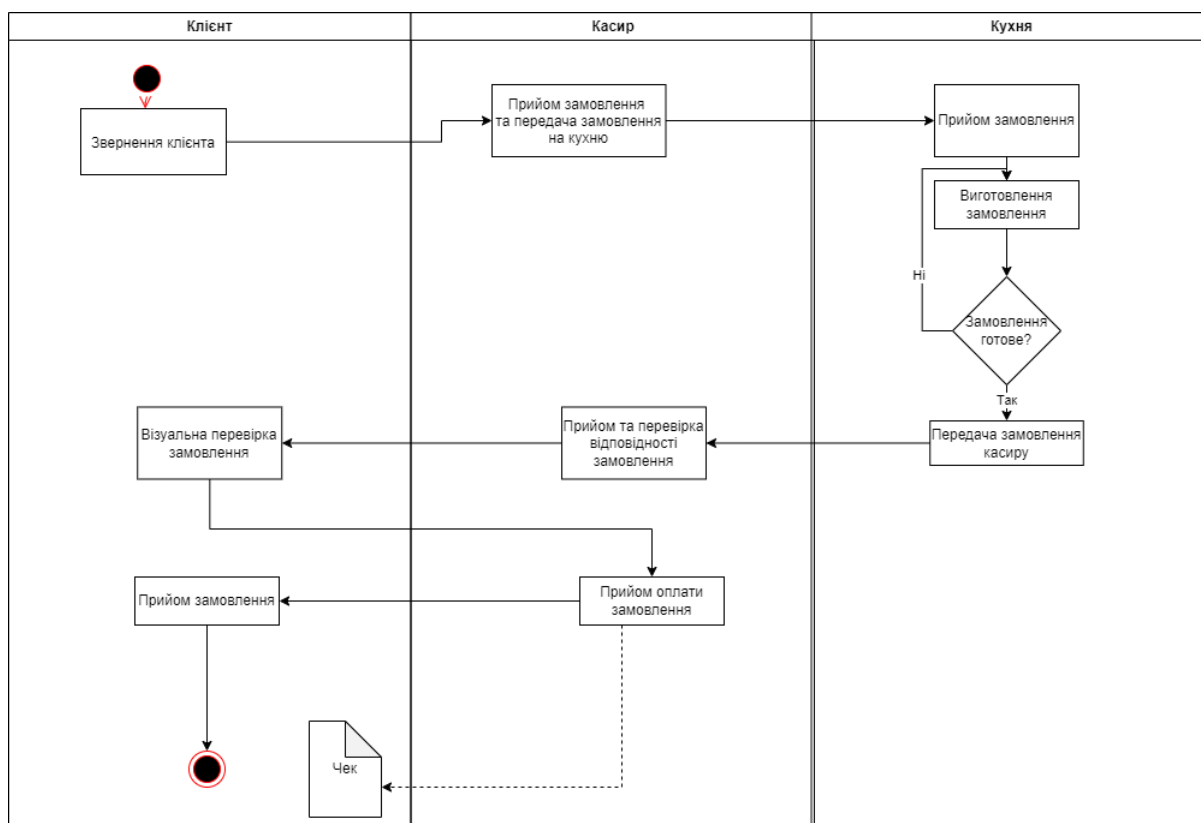


Рисунок 1.3 — Функціональна модель обслуговування клієнта та прийому замовлення

На зображеній вище функціональній моделі (Рис. 1.3.) ми можемо бачити незручний та довгий процес обслуговування. В кафетерії це може стати суттєвим обмеженням для клієнтів і негативно впливати на їхнє задоволення від візиту до закладу.

Якщо не впровадити систему самообслуговування, це може спричинити такі проблеми в організації:

1. Довгі черги: якщо кафетерій не організовує процес обслуговування ефективно, це може призводити до утворення довгих черг. Клієнти, які мають

обмежений час або поспішають, можуть бути розчаровані або просто піти, так як будуть вважати це втратою часу.

2. Неспроможність швидко замовити: якщо клієнти мають обмежений час або поспішають, довгий процес оформлення та очікування замовлення може бути незручним. Нестача персоналу або складний процес замовлення можуть затримувати клієнтів і призводити до втрати їхнього інтересу.

3. Складність оплати: якщо процес оплати не є зручним або недоступним, це може стати перешкодою для клієнтів. Відсутність різноманітних способів оплати, довгий процес оплати або проблеми з платіжними системами можуть призвести до незадоволення та втрати клієнтів.

Автоматизація спростить процес замовлення та зменшить навантаження на персонал, що призведе до покращення швидкості та якості обслуговування клієнтів.

Для вирішення проблеми з навантаженням на персонал необхідно впровадити підсистеми самообслуговування, такі як спеціальні екрани або планшети, а також спеціальні POS-термінали для замовлень, які дозволяють сплатити за продукт і отримати квиток або номер замовлення після його завершення.

Даний спосіб має такі переваги:

1. Швидкість обслуговування: клієнти можуть самостійно зробити замовлення, вибрати продукти, які їм потрібні, та сплатити їх прямо на автоматизованому терміналі. Це скорочує годину, яка зазвичай витрачається на очікування черги та обробку замовлень.

2. Зниження помилок: система самообслуговування може бути налаштована таким чином, щоб забезпечити точність замовлення. Клієнти самостійно вибирають потрібні пункти та можуть бачити своє замовлення перед підтвердженням. Це допомагає уникнути помилок, пов'язаних з неправильним сприйняттям або передачею замовлення персоналом.

3. Зниження навантаження на персонал: завдяки системі самообслуговування персонал може бути звільнений від певних завдань,

пов'язаних з прийомом замовлень та оплатою. Це дозволяє зосередити зусилля на інших важливих аспектах роботи, таких як приготування страв, видачею замовлень та забезпечення клієнтів високо якісним сервісом.

4. Покращення клієнтського досвіду: самообслуговування може бути зручним та привабливим для клієнтів, оскільки вони мають можливість самостійно контролювати процес замовлення та оплати. Це дозволяє їм вибирати продукти з меню, додавати чи видаляти інгредієнти за власним бажанням та контролювати свої витрати.

1.5. Удосконалення процесу прийому та обслуговування клієнтів

Розглянемо функціональну модель прийому та обслуговування клієнтів з використанням системи самообслуговування, наведену на рисунку 1.4.

Відмінність від функціональної моделі, наведеної на рисунку 1.3 полягає в тому, що користувач може скоротити годину очікування в черзі, мати можливість переглядати меню та дізнатися про ціни, а також авторизуватися.

Автоматизовані системи стають дедалі більш поширеними у сфері обслуговування, оскільки технології не стоять на місці, особливо щодо закладів харчування.

Впровадження системи самообслуговування матиме наступні переваги:

Збільшення ефективності: оскільки клієнти можуть робити замовлення та оплачувати самостійно, система самообслуговування може скоротити годину обслуговування. Це зменшує навантаження на персонал і потребує черг.

Покращення задоволення клієнтів: велика кількість клієнтів вимагають швидкого та зручного обслуговування. Їм подобається система самообслуговування, оскільки вона дозволяє самостійно вибирати та замовляти страви, контролювати час та процес обслуговування, а також створює кращі враження.

Зниження витрат: впровадження системи самообслуговування може допомогти зменшити витрати на персонал, оскільки менше працівників потрібно для прийому замовлень та обслуговування клієнтів.

Збільшення точності та уникнення помилок: помилки людей у процесі обробки замовлень можуть викликати незадоволення клієнтів та фінансові втрати. Система самообслуговування забезпечує більш точну та швидку обробку замовлень, а також зменшує ймовірність помилок.

Розширення маркетингових можливостей: система самообслуговування може збирати аналітичні дані про уподобання, замовлення та звички клієнтів. Рекомендації, пропозиції та програми лояльності можна адаптувати за допомогою цих даних, щоб покращити маркетингові стратегії та збільшити продажі.

Зважаючи на ці переваги, розробка додатку самообслуговування для кафетерії може значно покращити якість обслуговування, збільшити кількість задоволених клієнтів та знизити витрати. Тим не менш, перед розробкою та впровадженням системи самообслуговування необхідно враховувати специфіку кафетерію, цільову аудиторію та технологічні можливості кафе.

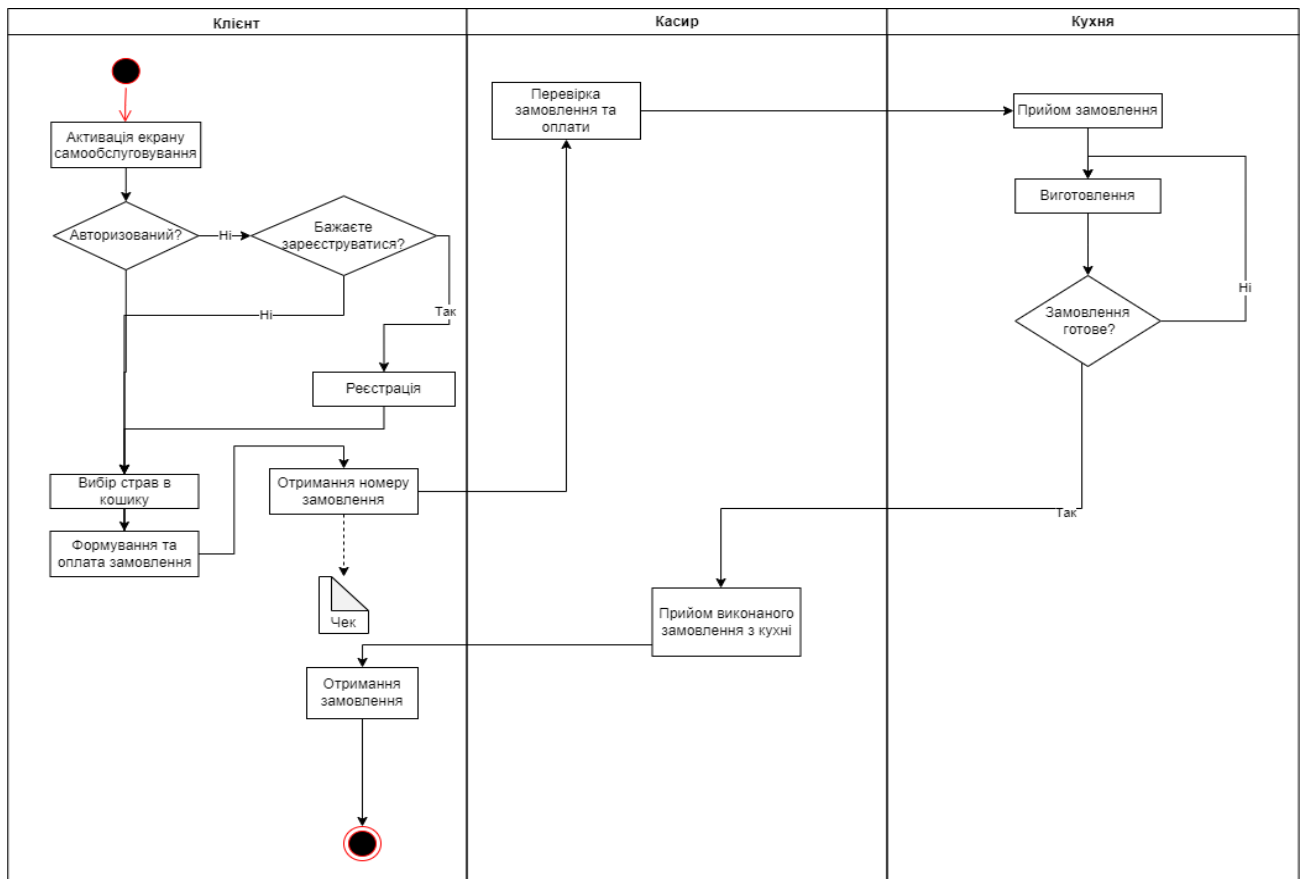


Рисунок 1.4 — Функціональна модель прийому та обслуговування клієнтів з використанням системи самообслуговування

1.6. Огляд існуючих рішень для розв’язання виявлених проблем

В закладах “MacDonalds” використовується додаток “MacDonalds” (Рисунок. 1.5)



Рисунок 1.5 — Екран самообслуговування MacDonaldis

В цього застосунку є такі характеристики :

1. Зрозумілий інтерфейс (Інтерфейс який дозволяє легко у всьому розібратися);
2. Можливість самостійно оформлювати замовлення (Клієнт може обрати певні позиції з меню, а програма перемикає на декілька сторінок з розділами їжі та оплатити й отримати чек замовлення);
3. Акції (можливість використовувати мобільний додаток, для отримання спеціальних купонів);
4. Замовлення страв (Користувач може вибрати бажані страви з електронного меню, яке відображає різноманітні опції їжі та напоїв).

В закладах “Mr.Grill” використовується додаток “СмартКафе” (Рисунок. 1.6)

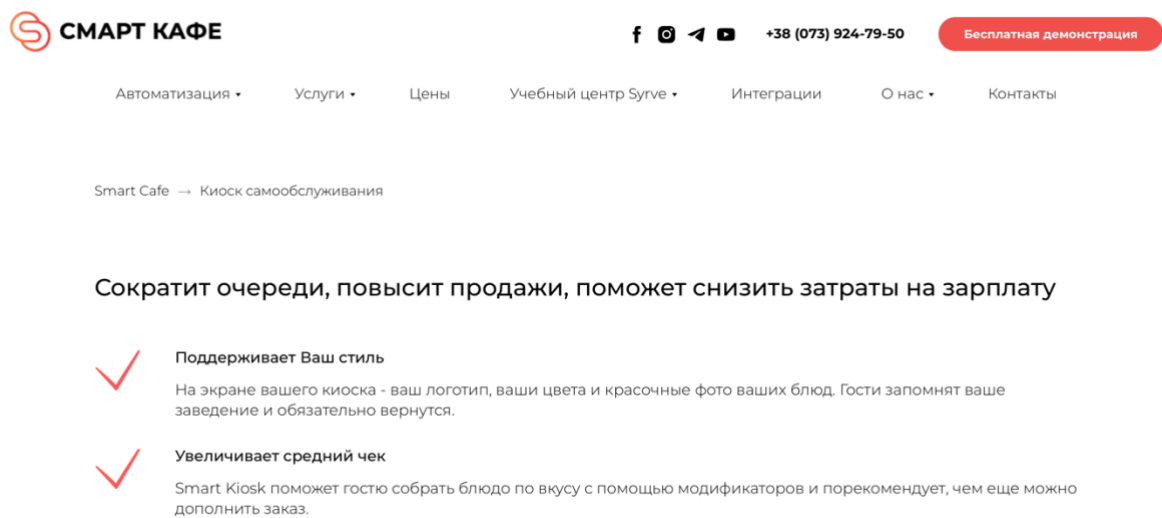


Рисунок 1.6 — Додаток “Smart Kiosk” від «Смарт Кафе»

Система має такі можливості:

1. Кастомізація (Дозволяє встановити логотип закладу, та фотографії страв);
2. Збільшує середній чек (Smart Kiosk допомагає та рекомендує замовити блюдо згідно до того що замовив користувач);
3. Вміє працювати з ііко (Гість може зробити все сам, а програма прийме замовлення, відправить його на кухню, прийме оплату та роздрукує гостю чек з номером замовлення);

4. Замовлення страв (Користувач може вибрати бажані страви з електронного меню, яке відображає різноманітні опції їжі та напоїв).

В таблиці 1.2 наведено порівняння систем аналогів.

Таблиця 1.2 Порівняння систем аналогів

Назва засобу	Додаток MacDonalds	Smart Kiosk
Вартість	--	900 грн/міс
Швидкодія	Середня швидкість	Висока швидкість
Функції	Прийом оплати; Можливість самостійно замовити продукцію; Перегляд замовлень; Редагування замовлень	Прийом оплати; Базова система лояльності; Робота зі смарт касосою; Можливість самостійно замовити продукцію
Сумісність	Android	Android
Мова інтерфейсу	Українська	Українська
Можливість використання на телефоні	Немає	Немає

1.7. Обґрунтування доцільності проектування і розроблення додатку SimpleBite

Після оцінки стану комп'ютеризації у кафетеріях та вивчення проблем і можливості виникнення тривалих черг у процесі обслуговування можна зробити наступні висновки:

- Кафетерій потребує поліпшення у сфері комп'ютеризації та автоматизації процесів. Багато кафетеріїв мають обмежені системи ПЗ, обмежені лише касовими апаратами.

- Низький рівень комп'ютеризації обмежує ефективність та продуктивність кафетеріїв, що може призводити до незадоволення клієнтів та втрати потенційного бізнесу.

- Застосування автоматизованих систем самообслуговування може бути ефективним рішенням для зменшення довгих черг та поліпшення процесу обслуговування.

- Автоматизація замовлення та оплати, управління запасами, управління персоналом, аналітика та звітність, а також забезпечення безпеки та контролю є ключовими завданнями, які можна вирішити за допомогою автоматизації.

Таким чином, можна сформулювати наступні рекомендації щодо створення програми самообслуговування для кафетерій:

1. Розробити і впровадити додаток для смартфонів або комп'ютерів, який дозволяє клієнтам самостійно замовляти та оплачувати страви.

2. Встановлення терміналів самообслуговування в кафетерії, де клієнти можуть самостійно замовляти і оплачувати страви без необхідності стояти у черзі.

3. Впровадити систему аналітики та звітності, яка надає детальну інформацію про продажі, популярність страв, клієнтські вподобання тощо.

4. Забезпечити надійну безпеку даних та контроль над фінансовими операціями.

5. Навчати персонал використовувати нові автоматизовані системи та додатки, щоб вони могли надати ефективну підтримку клієнтам.

6. Створення бонусних програм для заохочення клієнта. Щоб клієнт повертався та залишав заклад задоволеним.

Впровадження автоматизованих систем самообслуговування в кафетерії може значно поліпшити якість обслуговування, знизити час очікування клієнтів та покращити загальний досвід відвідування кафетерію. Крім того, це може допомогти збільшити ефективність роботи кафетерію, знизити помилки та забезпечити більш точне управління процесами.

На нашу думку, можливість допомогти кафетерію удосконалити його процеси та зробити обслуговування більш ефективним та зручним для клієнтів.

Згідно з нашим аналізом, введення додатків для замовлення та оплати через смартфони або терміналів самообслуговування може значно зменшити годину очікування та покращити загальний досвід клієнтів. Це також дозволить ефективніше керувати запасами, а також забезпечити точний аналіз продажів та попиту через системи аналітики та звітності.

Одним із ключових переваг є можливість впровадження бонусних програм, що стимулює лояльність клієнтів та підвищує їхню залученість до кафетерію.

Згідно з нашим аналізом, введення додатків для замовлення та оплати через смартфони або терміналів самообслуговування може значно зменшити годину очікування та покращити загальний досвід клієнтів. Це також дозволить ефективніше керувати запасами, а також забезпечити точний аналіз продажів та попиту через системи аналітики та звітності.

Одним із ключових переваг є можливість впровадження бонусних програм, що стимулює лояльність клієнтів та підвищує їхню залученість до кафетерію. Забезпечить надійну захист конфіденційності даних та фінансових операцій.

1.8. Розрахунок економічного ефекту від впровадження системи

Фактори, які можуть бути джерелами прибутку від впровадження комп'ютерної системи, включають:

1. зменшення кількості працюючих у відповідних підрозділах;
2. можливість виконувати більшу кількість замовлень;
3. зменшення збитків від аварій внаслідок несвоєчасного проведення технічного обслуговування обладнання і навчання персоналу;
4. зменшення збитків внаслідок запобігання розкраданню готової продукції, сировини та напівфабрикатів тощо;
5. зменшення браку та підвищення сортності продукції;
6. зменшення витрат на транспортування і складування продукції;

Ступінь новизни завдань, які розробляються: "Використання типових проектних рішень за умови їх змін, розробка проектів, які мають подібні рішення." Однією з категорій є складність алгоритму розробки.

Таким чином, підсистема буде розповсюджувати інформацію, яка охоплює управління персоналом, виробництвом, технічним обслуговуванням, безпекою, контролем якості, логістикою та фінансами. Це забезпечить комплексний підхід до управління виробничими процесами та оптимізацію витрат.

В таблиці 1.3 наведено види інформації проекту.

Таблиця 1.3. Види інформації проекту

Вид інформації	Позначення	Кількість наборів даних
Кількість видів змінної інформації	ЗІ	m= 7
Кількість видів нормативно-довідкової інформації	НДІ	n= 2
Кількість банків (баз) даних	БД	P= 1
Обробка в режимі реального часу	РЧ	Так
Забезпечення телекомунікаційної обробки даних і управління віддаленими об'єктами	ТОУ	Ні

Витрати часу на розробку ескізного проекту технічного завдання будуть складати:

Таблиця 1.4. Витрати часу для стадії «Технічний проект»

Вид системи	Стадія розробки системи	
Ескізний проект	Технічне завдання	
В	В	
Управління матеріально-технічним постачанням, управління збутом продукції, управління комплектацією, управління експортними та імпорними поставками	T1=53	T2=42

Визначимо витрати години на кожному з етапів технологічного проекту, робочого проекту та впровадження.

Вхідними даними для визначення є:

- кількість форм вхідної інформації 7;
- кількість форм вихідної інформації 2;
- базове значення витрат часу для стадії «Технічний проект» $T_{Б3}=90$
- базове значення витрат часу для стадії «Робочий проект» $T_{Б4}=144$
- базове значення витрат часу для стадії «Впровадження» $T_{Б5}=58$

Базове значення витрат часу ТБ коригується за допомогою поправочних коефіцієнтів для всіх стадій розробки автоматизованої системи.

Визначення витрат часу для стадії «Технічний проект» (T_3).

Таблиця 1.5. Коефіцієнт новизни

Вид використаної інформації	Ступінь новизни
	В
k_1	1.0
k_2	0.72
k_3	2.08

Таблиця 1.6. Коефіцієнти для стадій розробки системи

Стадія розробки системи	Вид обробки	Ступінь новизни
		В
Технічний проект	РЧ	1.26
Робочий проект	РЧ	1.32
Впровадження	РЧ	1.21

Розрахуємо витрати часу на проект скориставшись наступною формулою (1.1.):

$$k_{\Pi} = \frac{(1 * 7 + 0.72 * 2 + 2.08 * 1)}{(7 + 2 + 1)} = 10.52/10 = 1.052 \quad (1.1)$$

Пронумеруйте формули

Витрата часу на технічний проект буде $T_3 = 90 * 1.052 * 1.26 = 116$

Визначення витрат часу для стадії «Робочий проект» (T_4). За формулою (1.2):

$$k_{\Pi} = \frac{k_1 * m + k_2 * n + k_3 * p}{m + n + p} \quad (1.2)$$

де:

k_{Π} - зважене середнє значення,

k_1, k_2, k_3 - значення, які усереднюються,

m, n, p - ваги відповідних значень.

Таблиця 1.7. Коефіцієнти k_1, k_2, k_3 для стадії «Робочий проект»

Вид використаної інформації	Ступінь новизни
	В
$k_1(ЗІ)$	1.0
$k_2(НДІ)$	0.48
$k_3(БД)$	0.4

$$k_{\Pi} = \frac{1 \cdot 7 + 0.48 \cdot 2 + 0.4 \cdot 1}{7 + 2 + 1} = 8,36/10 = 0,836$$

Складність контролю вхідної та вихідної інформації $k_c = 1.00$

$$T_4 = 144 \cdot 0.836 \cdot 1.32 \cdot 1.00 = 158,9$$

Визначення витрат часу для стадії «Впровадження» (T_5) використовуючи формулу (1.1)

$$T_5 = 58 \cdot 0,836 \cdot 1.21 = 58.67$$

Отже, загальні людські витрати праці складають. За формулою (1.3) :

$$T_{\Sigma} = T_1 + T_2 + T_3 + T_3 + T_4 + T_5 \quad (1.3)$$

де:

T_{Σ} - загальна сума величин,

T_1, T_2, T_3, T_4, T_5 - індивідуальні величини, які додаються.

$$T_{\Sigma} = 53 + 42 + 116 + 159 + 59 = 429$$

Визначимо чисельність виконавців Чол. За формулою (1.4):

$$\Psi = \frac{T_{\Sigma}}{\Phi} \quad (1.4)$$

де:

T_{Σ} - загальний час виконання завдань,

Φ - кількість завдань.

Для кваліфікаційної роботи бакалавра кількість місяців роботи береться 3 місяці. Отже для виконання такого проекту потрібна така чисельність виконавців:

$$Ч = 429/91 = 5 \text{ Чол.}$$

Розмір заробітної плати програміста 10000 грн, за формулою (1.5) сума заробітних плат програміста складає:

$$V' = Ч * М * ЗП = 5 * 3 * 10000 = 150000 \text{ грн.} \quad (1.5)$$

де:

V' - обчислювана величина,

$Ч$ - середнє значення,

$М$ - кількість,

$ЗП$ - зарплата.

Витрати, пов'язані з розробкою програми на ПК.

Розрахунок річного фонду часу роботи ПК.

Дійсний річний фонд часу ПК у годинах дорівнює числу робочих годин у році для оператора, за винятком часу на технічне обслуговування і ремонт ПК (в середньому 5 год/міс + 6 роб.днів/рік). Використовуючи формулу (1.6). [8]

$$T_{ПК} = 2000 - (6 * 8 + 5 * 12) = 1892 \text{ год.} \quad (1.6)$$

Оскільки під час виконання бакалаврської роботи здобувач в середньому витрачає 450 год. машинного часу [8], то величина фонду часу ПК дорівнює:

$$T_{ПК} = 1892 * (450/2000) = 425.7 \text{ год}$$

Поточні витрати на експлуатацію V

Балансова вартість ПК, де $Ц_r$ - ринкова вартість ПК, орієнтовно складає 40000 грн, $k_{ун}$ – коефіцієнт, що враховує витрати на установку ПК . $k_{ун}=0,12$, вираховуємо за формулою (1.7).

$$Ц_{ПК} = Ц_r * (1 + k_{ун}) = 40000 * (1 + 0,12) = 44800 \text{ грн} \quad (1.7)$$

де:

$Ц_{ПК}$ - нова ціна продукції,

$Ц_r$ - базова (поточна) ціна продукції,

$k_{ун}$ – коефіцієнт збільшення ціни.

Амортизаційні відрахування використання ПК, Z_{AM} , обчислюються за формулою (1.8).

$$Z_{AM} = \frac{Ц_{ПК}}{Н_d} = 44800/5 = 8960 \text{ грн} \quad (1.8)$$

де:

Z_{AM} - середнє значення,

$Н_d$ - показник.

Витрати на електроенергію ($Z_{ел}$), споживану ПК, обчислюються

де потужність ПК, $P_{ПК} = 0.5$ кВт; фонд корисного часу роботи ПК.

$T_{ПК} = 435.16$ год, вартість 1 кВт електроенергії для підприємств, $Ц_{ел} = 1,86$ грн/кВт, коефіцієнт інтенсивного використання ПК, $A = 0.9$, за формулою (1.9).

$$Z_{ел} = 0,5 * 435,16 * 1,68 * 0,9 = 329 \text{ грн.} \quad (1.9)$$

Витрати на поточний ремонт і технічне обслуговування ПК (визначаються як 6% від балансової вартості ПК, $Ц_{ПК}$, за формулою (1.10).

$$Z_p = Ц_{ПК} * 0,06 \quad (1.10)$$

$$Z_p = 44800 * 0,06 = 2688$$

Непрямі витрати, пов'язані з експлуатацією ПК, визначаються як 5% від балансової вартості ПК $Ц_{ПК}$, за формулою (1.11).

$$Z_p = Ц_{ПК} * 0,05 \quad (1.11)$$

$$Z_p = 44800 * 0,05 = 22400$$

Поточні витрати на експлуатацію V'' , за формулою (1.12).

$$V''_1 = Z_{OP} + Z_{AM} + Z_{ел} + Z_p + Z_{MAT} \quad (1.12)$$

де:

Z_{OP} - витрати на основну продукцію,

Z_{AM} - середнє значення,

$Z_{ел}$ - витрати на електроенергію,

Z_p - витрати на робочу силу (заробітна плата),

Z_{MAT} - витрати на матеріали.

Заробітна плата обслуговуючого персоналу складає в середньому - 10000

Тож, поточні витрати на експлуатацію, V_1 , грн, складають:

$$V''_1 = 10000 + 8960 + 329 + 2688 + 2240 = 24217 \text{ грн.}$$

А, загальні витрати на розробку програмного забезпечення комп'ютерної системи складають, за формулою (1.13):

$$V_1 = V'_1 + V''_1 = 150000 + 24217 = 174217 \text{ грн.} \quad (1.13)$$

Розрахунок витрат на придбання і встановлення комп'ютерів, за формулою(1.14):

$$Ц_{ПК} = Ц_p * (1 + k_{ун}) \quad (1.14)$$

$$Ц_{ПК} = 20000 * (1 + 0,12) = 22400 \text{ грн.}$$

Так як для комфортної роботи нашого закладу потрібно мінімум 2-3 пристрої, то ціна на придбання та встановлення комп'ютерів буде:

$$V_2 = Ц_{ПК} = 67200 \text{ грн.}$$

Розрахунок витрат на підготовку приміщення і навчання персоналу

Витрати на підготовку приміщення $V_3 = 0$, так як приміщення є в наявності.

В середньому навчання персоналу триватиме 1 місяць, тому можна вважати, що $V_4 = 4500$ грн;

Загальна вартість розробки і впровадження системи вираховується за формулою (1.15):

$$V_{\Sigma} = V_1 + V_2 + V_3 + V_4 \quad (1.15)$$

$$V_{\Sigma} = 174217 + 67200 + 0 + 4500 = 245919 \text{ грн.}$$

Оскільки норма амортизаційних втрат для комп'ютерних систем $H_A = 5$, то для обрахування річного економічного ефекту слід брати до розгляду величину, за формулою (1.16):

$$V_p = \frac{V_{\Sigma}}{H_A} \quad (1.16)$$

де:

V_p - показник,

V_{Σ} - загальна величина,

H_A - кількість одиниць.

$$V_p = \frac{245919}{5} = 49183 \text{ грн}$$

Термін окупності розробки визначається, за формулою (1.17):

$$T_{OK} = \frac{1}{K_{EF}} \quad (1.17)$$

де коефіцієнт економічної ефективності за формулою, (1.18)

$$K_{EF} = \frac{P_p}{V_p} \quad (1.18)$$

Де річний прибуток P_p

Зменшення витрат на працю: від впровадження системи буде досягнуто за рахунок зменшення кількості працюючих і як результат - можливість виконувати більше замовлень, нехай поточні витрати на заробітну плату складають, наприклад, 60,000 гривень на місяць для 2-х працівників.

Впровадження системи дозволяє зменшити кількість працівників на 1, скоротивши витрати на заробітну плату на 30,000 гривень на рік.

$$K_{EF} = \frac{30000}{49183} = 0,6$$

Отже, термін окупності ІС складатиме: $T_{OK} = \frac{1}{0,6} = 0,6$ років.

РОЗДІЛ 2. ТЕХНІЧНЕ ЗАВДАННЯ НА ПРОЕКТУВАННЯ

2.1. Загальні положення

2.1.1. Дане технічне завдання розроблено згідно з вимогами ДСТУ 34.602-89 і є головним документом, у якому зазначено всі вимоги та порядок проектування, розробки та впровадження додатку для он-лайн замовлення продукції кафетерію “SimpleBite”.

2.1.2. Повне найменування: Додаток для он-лайн замовлення продукції кафетерію “SimpleBite”.

2.1.3. Замовник: Кафетерій “SimpleBite” за згодою з координатором проекту (Генеральним користувачем) кафетерію “SimpleBite”

2.1.4. Виконавець: Степанець К.Р.

2.1.5. Перелік документів, згідно яких виконуються роботи:

Технічні вимоги до додатку для он-лайн замовлення продукції кафетерію “SimpleBite”

Меморандум про співробітництво між ФОП Степанець К.Р. та кафетерієм “SimpleBite”

2.1.6. Строк виконання робіт: 1-й етап – IV квартал 2024 року. Терміни наступних етапів узгоджуються сторонами після завершення попереднього етапу.

2.1.7. Джерело фінансування: власні кошти

2.1.8. Допускається внесення змін до цього ТЗ за взаємною згодою сторін у відповідності до ДСТУ 34.602-8

2.2. Призначення та цілі створення системи

“SimpleBite” — це додаток для онлайн-замовлення продукції кафетерію. Він містить комплекс програмно-апаратних засобів, які автоматизують обробку, зберігання, відображення та передачу інформації, включаючи інформацію з обмеженим доступом, що не містить державну таємницю, згідно з діючим законодавством.

Система кафетерію «SimpleBite» інтегрує всіх учасників інформаційного простору, дозволяючи їм надавати та переглядати інформацію, яка використовується в технологічних процесах кафетерії, за допомогою засобів технічного захисту, схвалених державою.

Мета додатку полягає в тому, щоб скоротити кількість працівників, які працюють на касі, зменшувати кількість часу, який клієнти проводять у черзі, пришвидшити процес виконання замовлень і надавати клієнтам точну, достовірну інформацію про меню, ціни та страви.

2.3. Характеристика об'єкта автоматизації

В рамках автоматизації охоплено діяльність, пов'язану з обслуговуванням кафетерію "SimpleBite", з метою покращення продуктивності закладу та уникнення помилок під час роботи.

2.4. Вимоги до системи

2.4.1. Вимоги до системи у цілому

Система повинна передбачати додавання, змінення, видалення страв чи продуктів з замовлення в режимі реального часу (при роботі безпосередньо з системою), дозволяти реєструватися новим клієнтам, або авторизуватися вже існуючим клієнтам закладу.

Робочі місця клієнтів системи – користувачів можуть знаходитись:

- біля екрану самообслуговування;
- на касі у продавця.

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

2.4.2.1. Загальна функціональність

Таблиця 2.1 Перелік функцій програмного засобу

п/п	Найменування функції	Вхідна інформація	Вихідна інформація
1	Авторизація клієнта	Таблиці БД «Користувач»	Форма з полями для авторизації
2	Реєстрація клієнта	Запит на додавання клієнта в таблицю «Користувач»	Форма з полями для реєстрації
3	Додавання товарів у кошик	Запит користувача, дані про товари	Форма з доданими у кошик товарами
4	Редагування товарів у кошику	Запит користувача, дані про товари у кошику	Форма з переліком товарів для редагування
5	Впорядкування товарів за типом	Запит на фільтрацію даних БД	Форма з переліком товарів обраного типу
6	Оформлення замовлення	Запит користувача, дані з кошика	Підтвердження замовлення, інформація про оплату
7	Оплата замовлення	Дані про замовлення, платіжні реквізити	Підтвердження оплати
8	Друк замовлення	Ідентифікатор замовлення	Друкований документ - замовлення

По функціональному складу система повинна складатися із:

- a. Підсистема «Головне меню»
- b. Підсистема «Замовлення»
- c. Підсистема «Реєстрація»
- d. Підсистема «Кошик»

По структурі додатку система повинна складатися із:

- a. клієнтської частини
- b. серверної частини
- c. модуля обміну інформацією з базою даних та іншими клієнтами системи

Функціональність підсистем повинна відповідати постановам Кабінету міністрів № 451 від 21.05.2012 р. № 320 від 08.04.2009 р.

Технологічна схема функціонування додатку:

- Клієнт натискає початок та переглядає список страв.
- Клієнт обирає страви, які бажає замовити.

- Клієнт натискає кнопку «Замовити».

Функціональність підсистеми «Замовлення» повинна відповідати технологічній схемі.

- Технологічна схема функціонування системи:
- Клієнт може бачити всі замовлені страви та їх ціну.
- Клієнт може видалити або обрати кількість страв або товарів.
- Клієнт формує замовлення.
- Клієнт оплачує замовлення.

2.4.2.2. Вимоги до клієнтської частини

Клієнтська частина повинна реалізувати наступну функціональність:

1. Забезпечити доступ до системи користувачів у режимі реального часу.
2. Забезпечувати завдання формування інтерфейсу користувача до системи.
3. Забезпечувати формування замовлень.

2.4.2.3. Вимоги до серверної частини

Серверна частина повинна реалізовувати наступну функціональність:

- реалізовувати бізнес-логіку відповідних підсистем.
- забезпечувати формування замовлень та редагування його зі збереженням на сервері.

Модуль повинен складатися з:

Програмного інтерфейсу до системи з можливістю отримання інформації в режимі реального часу і вивантаження її із системи у вигляді електронного документу в узгодженому форматі.

2.4.3. Вимоги до видів забезпечення

2.4.3.1. У вимогах до математичного забезпечення (МЗ) додаток не вимагає спеціального математичного забезпечення для реалізації покладених на неї функцій. Достатньо можливостей обраної СУБД.

2.4.3.2. Вимоги до інформаційного забезпечення (ІЗ).

2.4.3.2.1. Інформаційне забезпечення системи повинно містити дані, достатні для виконання всіх покладених на систему функцій. ІЗ повинно гарантувати раціональну організацію зберігання інформації та доступу до неї.

За заповнення БД інформацією відповідальність переходить до замовника за методиками і формами, створеними розробниками системи.

Склад, структура і спосіб організації інформації представляються у логічній моделі БД і можуть уточнюватись на етапі технічного проектування.

2.4.3.2.2. Слід передбачити захист даних від руйнування при аваріях і порушеннях у енергоживленні системи — використання резервних копій БД.

2.4.3.3. Вимоги до лінгвістичного забезпечення (ЛЗ).

2.4.3.3.1. Для розробки програмних засобів, які будуть реалізовувати виконання функцій і забезпечують сервіс користувачів повинні використовуватися мови високого рівня програмування, які забезпечують створення програм, а також мова обраної СУБД для здійснення доступу, редагування, додавання та маніпулювання даними.

2.4.3.3.2. Діалог користувача до системи повинен будуватися на наборах меню і підказок, сфокусованих на виконанні вказаних користувачем функцій. Запити користувача до системи повинні подаватись переважно природною мовою.

2.4.3.4. Вимоги до програмного забезпечення (ПЗ).

2.4.3.4.1. Системне ПЗ має забезпечувати надійність та якість виконання функціональних завдань системи. До системного ПЗ належить:

- операційна система (далі ОС) — Windows/Android/IOS
- система управління БД (далі СУБД) — MySQL/SQLite .

2.4.3.4.2. Вимоги до системного ПЗ мають бути такі:

- мінімальні вимоги до ресурсів технічних засобів (ТЗ);
- максимальна швидкодія;
- повне задоволення потреб функціональних завдань системи.

2.4.3.4.3. Вимоги до ОС:

- мінімальне використання ресурсів комп'ютера для власних потреб.

Пропорційне навантаження оперативної і дискової пам'яті, якщо вона присутня в пристрої;

- максимальна швидкодія при управлінні зовнішніми пристроями;
- ОС сервера — Windows, ОС клієнта — Android

2.4.3.4.4. Вимоги до СУБД:

- максимальне задоволення потреб функціональних задач;
- надійність;
- ефективне управління потрібного обсягу і структури;
- швидкість виконання запитів користувачів;
- мінімальні вимоги до ТЗ.

2.4.3.4.5. Програмні засоби введення та виведення даних і ведення діалогу повинні забезпечувати:

- Виведення необхідних даних на екран у вигляді відповідних позицій або тексту;
- Індикація введення даних контролем і спеціальним вікном користувачу про наявність помилок з можливістю їх виправлення під час введення даних;
- Керований комп'ютером діалог при введенні даних;
- Виведення даних у відповідному вигляді (Формі на екрані) за запитом при натисканні відповідної кнопки користувачем.

2.4.3.4.6. При розробленні спеціального ПЗ слід виконати наступні вимоги:

- Програми які будуть використані мають бути сумісні та із загальносистемним ПЗ;
- ПЗ має розроблятися засобами об'єктно-орієнтованого програмування;
- Забезпечити відповідність інтерфейсу користувача стандартам систем;
- Повинна бути передбачена можливість розширення автоматизованих задач у відповідності з новими функціональними потребами замовника;

- ПЗ не повинно залежати від типу зовнішніх пристроїв (касових апаратів, терміналів оплати, принтерів, дисків, сканерів тощо);
- Діалог програми з користувачем має бути за допомогою клавіатури з поясненням виконання дій і можливістю отримати підказки.

2.4.3.5. Вимоги до технічного забезпечення.

2.4.3.5.1. Технічні засоби системи (таблиця 2.2) повинні забезпечувати виконання функцій, перерахованих в таблиці 2.1.

2.4.3.5.2. Засоби обчислювальної техніки повинні забезпечувати обмін інформації в об'ємах, приведених в п. 4.3.2.

Таблиця 2.2 Вимоги до технічного забезпечення системи

№ п/п	Основні характеристики комп'ютера
Технічне забезпечення для сервера	
1	ACER Aspire 3 5GHz\64Gb\1 TB\ LAN 1 Gbit
Технічне забезпечення для клієнта	
1	PosTerminal Intel Celeron 2GHz/64GB

2.5. Склад і зміст робіт по створенню системи

Таблиця 2.3 Вимоги до технічного забезпечення систем

№ п/п	Найменування робіт	Строки виконання робіт
1	Передпроектне дослідження об'єкта автоматизації	12.04.2024
2	Технічне завдання	09.05.2024
3	Технічний проект	14.05.2024
4	Оформлення документації	22.05.2024

2.6. Порядок контролю і приймання системи

2.6.1. Види, склад, об'єми і методи випробувань додатку та її складових частин

У процесі розробки додатку виконуються наступні види випробувань:

- тестування елементів системи
- тестування системи у цілому
- досвідчена експлуатація

Крім того, при впровадженні комплексної системи захисту інформації, проводяться додаткові випробування, передбачені інструкціями ДСТЗІ.

2.6.2. Тестування для визначення працездатності і рішення про можливість прийняття системи в експлуатацію проводять розробники разом із замовником. Програму тестувань складає розробник і затверджує замовник.

2.6.3. Задача тестової експлуатації здійснюється на основі технічного завдання та інструкції користувача. За результатами тестування формується перелік доробок і рекомендовані строки їх виконання.

2.6.4. Введення в дію системи оформлюється актом здачі-прийому. Завершення етапу фіксується відповідним протоколом між Виконавцем і Замовником.

2.7. Вимоги до складу і змісту робіт із підготовки до введення додатку в дію.

Для введення в дію замовник виконує ряд робіт із підготовки об'єкта:

- Проводить укомплектування технічних засобів;
- Організовує навчання персоналу системи роботі на засобі та вивченню інструкції з її експлуатації;
- Проводить тестову експлуатацію і вводить додаток в роботу.

2.8. Вимоги до документуванню

У процесі розробки додатку створюється повний комплекс документації, що включає технічне завдання та технічний проект. Ця документація розробляється відповідно до вимог Державних стандартів серії 19 “Єдина система програмної документації” і серії 24 “Єдина система стандартів автоматизованих систем управління”. Усю документацію, пов'язану з додатком, потребується створювати з урахуванням вимог ДСТУ-34 201. Спеціалізоване обладнання, яке постачається партнерськими організаціями, мусить супроводжуватися вичерпною набором документаційних матеріалів та (за необхідності) сертифікатами відповідності.

РОЗДІЛ 3. ОПИС КОМПЛЕКСУ ЗАДАЧІ АВТОМАТИЗАЦІЇ

Основні задачі, які необхідно реалізувати в процесі розробки програми:

- створення нового користувача;
- формування меню;
- заповнення кошику стравами та товарами;
- додавання або вилучення обраних позицій меню;
- оплата замовлення бонусами;
- оплата замовлення.

3.1. Інформаційне забезпечення системи

3.2.1. Логічна модель БД

Логічна модель бази даних (БД), наведена на рисунку 3.1 призначена для управління інформацією про клієнтів, замовлення їжі та напоїв у кафе.

Нижче наведено описи сутностей БД.

Сутність "Customers" (Клієнти): містить інформацію про клієнтів, таку як ідентифікатор, ім'я, номер телефону, кількість бонусів та пароль для доступу.

Сутність "mainFood" (Основні страви), "desserts" (Десерти) та "drinks" (Напої): містять інформацію про доступні страви, десерти та напої. Кожна з цих таблиць містить унікальний ідентифікатор, назву та ціну.

Сутність "order" (Замовлення): зберігає інформацію про конкретні замовлення. Кожне замовлення містить унікальний ідентифікатор, посилання на клієнта, основну страву, десерт, напій, а також загальну суму замовлення.

Взаємозв'язки між цими сутностями полягають у тому, що інформація про замовлення (сутність "order") посилається на інформацію про клієнтів (сутність "Customers") та на сутності інгредієнтів (основні страви, десерти та напої) через зовнішні ключі (FK), які вказують на відповідні ідентифікатори у відповідних сутностях. Це дозволяє зберігати і відстежувати інформацію про замовлення конкретних клієнтів та їх вибір страв.

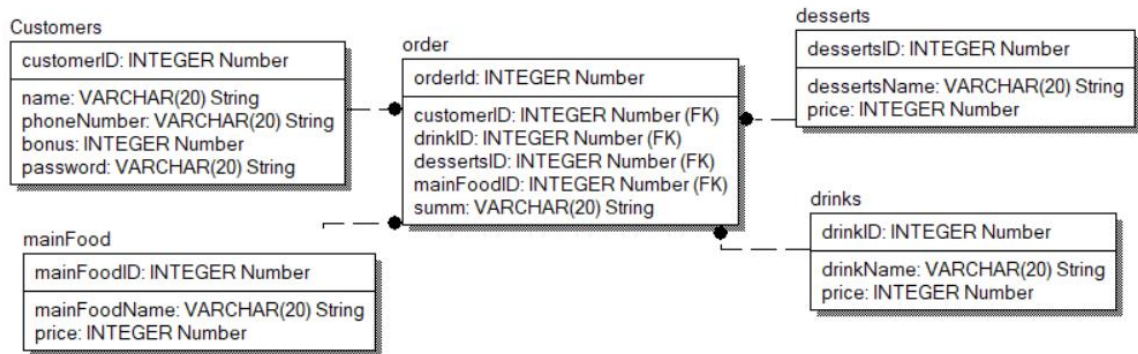


Рисунок 3.1 — Логічна модель БД

3.2.2. Фізична модель БД

Фізична модель бази даних (БД) див. рисунок 3.2 описує, як саме дані зберігаються на рівні файлів або таблиць в базі даних.

В даному випадку, фізична модель БД містить наступні складові.

Таблиця "Customers" (Клієнти):

customerID: INTEGER - Унікальний ідентифікатор клієнта.

name: VARCHAR(20) - Ім'я клієнта.

phoneNumber: VARCHAR(20) - Номер телефону клієнта.

bonus: INTEGER - Кількість бонусів у клієнта.

password: VARCHAR(20) - Пароль для доступу клієнта.

Таблиця "mainFood" (Основні страви):

mainFoodID: INTEGER - Унікальний ідентифікатор основної страви.

mainFoodName: VARCHAR(20) - Назва основної страви.

price: INTEGER - Ціна основної страви.

Таблиця "desserts" (Десерти):

dessertsID: INTEGER - Унікальний ідентифікатор десерту.

dessertsName: VARCHAR(20) - Назва десерту.

price: INTEGER - Ціна десерту.

Таблиця "drinks" (Напої):

drinkID: INTEGER - Унікальний ідентифікатор напою.

drinkName: VARCHAR(20) - Назва напою.

price: INTEGER - Ціна напою.

Таблиця "order" (Замовлення):

orderID: INTEGER - Унікальний ідентифікатор замовлення.

customerID: INTEGER - Ідентифікатор клієнта (зовнішній ключ, FK, посилання на таблицю "Customers").

drinkID: INTEGER - Ідентифікатор напою (зовнішній ключ, FK, посилання на таблицю "drinks").

dessertsID: INTEGER - Ідентифікатор десерту (зовнішній ключ, FK, посилання на таблицю "desserts").

mainFoodID: INTEGER - Ідентифікатор основної страви (зовнішній ключ, FK, посилання на таблицю "mainFood").

sum: VARCHAR(20) - Загальна сума замовлення.

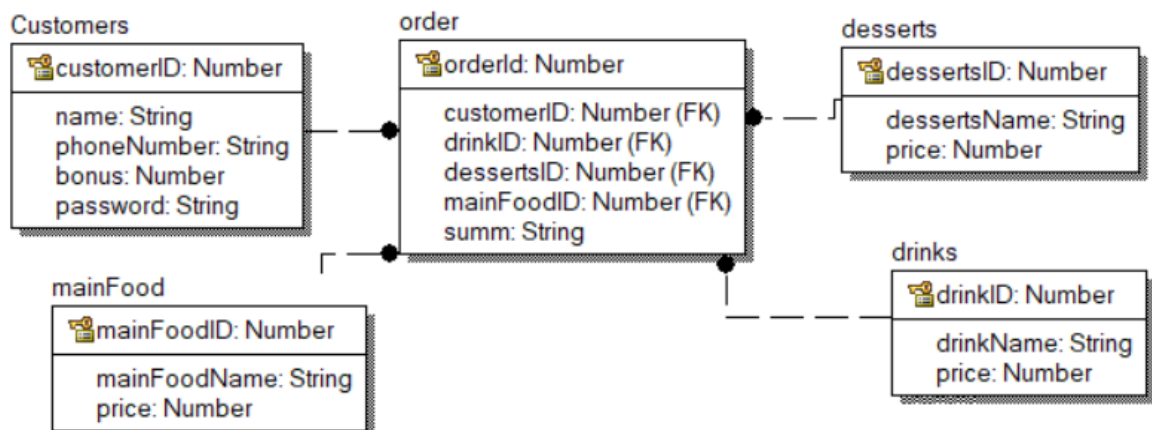


Рисунок 3.2 — Фізична модель БД

3.3. Алгоритмізація та реалізація комплексу задач автоматизації

Під'єднання до БД

Для під'єднання до БД розроблено наведений у Додатк В. Рисунок В.1 код визначає методи для отримання шляху до бази даних і відкриття з'єднання з нею в залежності від платформи, на якій працює додаток.

Метод `GetDatabasePath()` використовує директиви препроцесора `#if`, `#elif` та `#endif`, щоб визначити шлях до БД залежно від платформи. Якщо додаток працює в редакторі Unity, він повертає шлях до БД в папці `StreamingAssets`. Для самостійних платформ (наприклад, Windows, MacOS) використовується шлях до `Application.dataPath`, а для платформи Android - `Application.persistentDataPath`. Якщо БД відсутня за вказаним шляхом, викликається метод `UnpackDatabase()`, щоб скопіювати її з вихідного файлу.

Метод `OpenConnection()` відповідає за відкриття з'єднання з БД. Він створює новий об'єкт `SqlConnection`, використовуючи шлях до БД, отриманий за допомогою методу `GetDatabasePath()`, і відкриває це з'єднання.

Створення нового користувача

Розглянемо реєстрацію нового користувача (див. Рисунок 3.3).

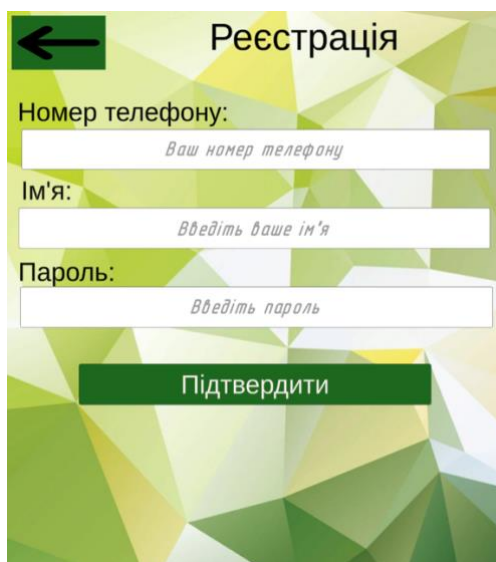


Рисунок 3.3 — Форма реєстрації користувача

В Додатку В наведено метод, `ApplyRegistration`, що виконує реєстрацію нового користувача у системі.

Після успішного проходження всіх перевірок, виконується ряд подальших дій:

- Починається перевірка наявності введених даних. Якщо хоча б одне з полів для введення (номер телефону, ім'я чи пароль) порожнє, відображається вікно помилки та виконання методу переривається.

- Перевіряється правильність формату номера телефону. Якщо введений номер телефону коротший за встановлену мінімальну довжину (`minPhoneNumberCharLenght`), також відображається вікно помилки та виконання методу переривається.
- Здійснюється SQL-запит до бази даних для перевірки наявності користувача з таким номером телефону. Якщо користувач з таким номером вже існує, виводиться відповідне повідомлення про помилку та виконання методу переривається.
- Якщо усі перевірки успішні, виконується SQL-запит на додавання нового користувача до таблиці бази даних "Customers". Після цього виводиться повідомлення про успішну реєстрацію та викликається метод `BackToMainMenu()`, який повертає користувача до головного меню.

Заповнення меню створюється за допомогою конфігурацій (див. Рисунок 3.4)

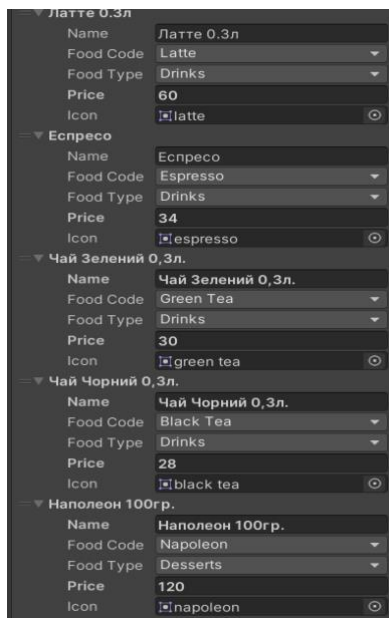


Рисунок 3.4 — Конфігурація страв в меню

Заповнення кошику товарами (див. Рисунок 3.5)



Рисунок 3.5 — Заповнений кошик

Для заповнення кошику використано метод, `HandleBasketItemSpawn`, що відповідає за додавання обраних товарів з меню в кошик.

Під час виклику методу `Instantiate`, створюється новий об'єкт інтерфейсу `FoodItemUIController`, який відображатиме інформацію про товар у кошику. Цей об'єкт створюється на основі префабу `foodPositionPrefab` та розміщується в `holder`.

За допомогою `SettingsManager.Instance.foodSettings.GetConfig(foodCode)` отримується конфігурація товару (`FoodConfig`) за кодом `foodCode`. Ця конфігурація містить необхідну інформацію про товар, таку як його назва, ціна, зображення тощо.

Метод `InitializeFoodItem` об'єкта `foodItem` ініціалізується з отриманою конфігурацією `config` та кількістю `amount`. Це означає, що об'єкт `foodItem` буде відображати відповідні дані про товар (назва, ціна тощо), а також кількість одиниць товару у замовленні.

Нарешті, об'єкт `foodItem` додається до списку `basketItems`. Цей список відслідковує всі обрані товари, які додані в кошик, для подальшого використання, наприклад, при розрахунку загальної вартості замовлення або відображення інформації про замовлення користувачеві.

Програмний код:

```
public void HandleBasketItemSpawn(FoodCode foodCode, int amount)
{
    // Контролер отримує кількість обраних товарів з меню, та додає в кошик.
    FoodItemUIController foodItem = Instantiate(foodPositionPrefab,
holder).GetComponent<FoodItemUIController>());
    // Отримати конфігурацію для заданого коду їжі
    FoodConfig config =
SettingsManager.Instance.foodSettings.GetConfig(foodCode);
    // Ініціалізувати елемент їжі з заданою конфігурацією та кількістю
    foodItem.InitializeFoodItem(config, amount);
    // Додати ініціалізований елемент їжі до списку елементів кошика
    basketItems.Add(foodItem);
}
```

Додавання або вилучення.

Методи `AddOrderCount()` та `SubstractOrderCount()` — відповідають за збільшення та зменшення кількості обраних позицій в замовленні.

Метод `AddOrderCount()` дозволяє збільшити кількість певної позиції товару в замовленні. Перш ніж збільшити кількість, він перевіряє, чи не досягнута максимальна кількість одиниць товару в замовленні. Якщо максимальна кількість вже досягнута, метод завершується без змін. Потім кількість збільшується на одиницю і обмежується в межах від 0 до 10 (якщо збільшення виходить за межі діапазону). Після цього спрацьовує подія `OnItemAdded`, що інформує про додання одиниці товару в замовлення, та викликається метод оновлення інтерфейсу користувача `UpdateCounterUI()`.

Метод `SubstractOrderCount()` зменшує кількість певної позиції товару в замовленні. Перед зменшенням кількості, він перевіряє, чи кількість не дорівнює нулю (тобто, чи є товари в замовленні). Якщо кількість вже нульова, метод завершується без змін. Потім кількість зменшується на одиницю і обмежується в

межах від 0 до 10 (якщо зменшення виходить за межі діапазону). Після цього спрацьовує подія `OnItemRemoved`, що інформує про видалення одиниці товару з замовлення, та викликається метод оновлення інтерфейсу користувача `UpdateCounterUI()`.

Метод збільшує кількість позиції замовленого товару в обсязі від 0 до 10.

```
private void AddOrderCount()
{
    // Перевірка, чи досягнута максимальна кількість товару
    if (orderCount == maxItemAmount)
        return;

    // Збільшення кількості товару
    orderCount++;

    // Обмеження кількості товару від 0 до 10
    orderCount = Mathf.Clamp(orderCount, 0, 10);
    // Виклик події додавання товару з оновленою кількістю
    OnItemAdded?.Invoke(foodConfig.foodCode, orderCount);
    // Оновлення інтерфейсу користувача для відображення нової кількості
    UpdateCounterUI();
}
```

Метод зменшує кількість позиції замовленого товару в обсязі від 0 до 10.

```
private void SubstractOrderCount()
{
    // Перевірка, чи кількість товару дорівнює 0
    if (orderCount == 0)
        return;

    // Зменшення кількості товару
    orderCount--;
```

```

// Обмеження кількості товару від 0 до 10
orderCount = Mathf.Clamp(orderCount, 0, 10);

// Виклик події видалення товару з оновленою кількістю
OnItemRemoved?.Invoke(foodConfig.foodCode, orderCount);

// Оновлення інтерфейсу користувача для відображення нової кількості
UpdateCounterUI();
}

```



Рисунок 3.6 — Заповнений вибраними стравами кошик

Оплата замовлення бонусами.

Метод, `HandlePayWithBonuses`, відповідає за оплату замовлення за допомогою бонусів користувача.

Перевіряється, чи авторизований користувач. Якщо користувач не авторизований, тобто не має облікового запису або не увійшов у систему, відображається відповідне повідомлення про помилку, і виконання методу завершується.

Розраховується вартість замовлення з урахуванням знижки за бонуси. Вартість замовлення віднімається від кількості бонусів користувача.

Якщо результат розрахунку (вартість замовлення зі знижкою) більше або рівний нулю, це означає, що бонусів менше, ніж вартість замовлення. Тоді списуються бонуси, а вартість замовлення оновлюється. Також оновлюється

інтерфейс корзини замовлення. Далі, розраховується кількість бонусів, які користувач отримає за це замовлення (зазвичай це відсоток від загальної вартості замовлення) і ця кількість зберігається у базі даних.

Якщо результат розрахунку менше нуля, це означає, що кількість бонусів перевищує вартість замовлення. Тоді списуються бонуси в повному обсязі, вартість замовлення стає нулем, а залишок бонусів записується у базу даних.

На останньому етапі оновлюється інтерфейс, щоб відобразити зміни після оплати замовлення за допомогою бонусів.

Обробка оплати за допомогою бонусів:

```
private void HandlePayWithBonuses()
{
    // Перевірка, чи авторизований користувач.
    // Якщо користувач не авторизований, то бонуси використовувати він не
    може.
    if (isUserAuthorized == false)
    {
        // Показ повідомлення про помилку, якщо користувач не авторизований.
        MainMenuUIManager.Instance.errorPopupController.ShowNotificationText(notAuthorizedMessage);
        return;
    }

    // Розрахунок суми замовлення з урахуванням знижки за бонуси.
    int orderPriceWithBonusDiscount = totalOrderPrice - bonusCount;
}
```

Якщо бонусів менше ніж ціна замовлення то списуються бонуси, і залишається сума з списаними бонусами і виконується перезапис їх значення до БД.

```
if (orderPriceWithBonusDiscount >= 0)
```

```

{
    // Якщо сума замовлення з урахуванням знижки більша або дорівнює 0,
    виконується наступне:

    // Оновлення загальної вартості замовлення з урахуванням знижки за
    бонуси.
    totalOrderPrice = orderPriceWithBonusDiscount;
    // Оновлення інтерфейсу кошика.
    UpdateBasketUI();
    // Скидання лічильника бонусів до нуля.
    bonusCount = 0;
    // Розрахунок бонусів, які будуть додані користувачу після покупки.
    int addedBonusToUser = Mathf.FloorToInt(orderPriceWithBonusDiscount *
    bonusCoefficientAfterPurchase);
    // Оновлення інформації про кількість бонусів у базі даних користувача.
    MyDataBase.ExecuteQueryWithoutAnswer($"UPDATE Customers SET bonus
    = {addedBonusToUser} WHERE id = {userID}");
}
Else

```

Якщо бонусів більше ніж сума замовлення, то списуються бонуси і виконується перезапис їх значення до 0.

```

{
    Скидання загальної вартості замовлення до 0.
    totalOrderPrice = 0;
    // Встановлення від'ємного значення лічильника бонусів.
    bonusCount = -(orderPriceWithBonusDiscount);
    // Оновлення інтерфейсу кошика.
    UpdateBasketUI();
    // Оновлення інформації про кількість бонусів у базі даних користувача.

```

```
MyDataBase.ExecuteQueryWithoutAnswer($"UPDATE Customers SET bonus
= {bonusCount} WHERE id = {userID}");
```

```
// Додаткові оновлення інтерфейсу кошика, які можуть бути потрібні.
```

```
UpdateBasketUI();
```

```
}
```



Рисунок 3.7 — Заповнений стравами кошик з сумою оплати до використання бонусів



Рисунок 3.8 — Заповнений стравами кошик зі списаними бонусами та зменшеною сумою до сплати

3.4. Технічне та системне забезпечення розробки

3.4.1. Обґрунтування вибору технічних засобів

Вибір Unity, Visual Studio C# та SQLite для створення підсистеми підтримки кафетерію “SimpleBite” ґрунтується на потребах і вимогах, наведених у технічному завданні. Кожен інструмент має свої переваги та можливості, які сприятимуть успішному впровадженню концепції самообслуговування.

Unity є визнаним середовищем розробки ігор та інтерактивних додатків для ряду платформ. Його функціонал у графіці 2D/3D, фізичній моделювання, анімаціях і штучного інтелекту дозволить покращити вигляд додатку самообслуговування. Використання Unity також гарантує сумісність з різноманітною платформами, що полегшить доступ до нього на будь-яких пристроях, що має велике значення для закладу.

Visual Studio є потужною інтегрованим середовищем розробки, що спеціалізується на різних мовах програмування. Використання C# дозволяє ефективно розробити програмну логіку додатку, забезпечуючи його функціональність та надійність. Синтаксис мови програмування C# є простою та зрозумілою, що полегшує розробку та підтримку коду.

Використання SQLite в системі самообслуговування кафетерію дозволить зберігати дані про користувачів, їх ім'я, кількість їх бонусів та іншу важливу інформацію. Це допоможе забезпечити швидкий доступ до даних та управління ними. SQLite також має гарну масштабованість, що дозволить розширити базу даних в майбутньому у випадку зростання потреб до підсистеми.

Отже, використання Unity та Visual Studio C# разом з SQLite як СКБД є доцільним варіантом для розробки та використання системи самообслуговування в кафетерії “ SimpleBite”. Ці перелічені інструменти забезпечать потужність, гнучкість, надійність та ефективність роботи підсистеми.

Враховуючи вищезазначені фактори, вважаю, що використання Unity, SQLite та Visual Studio C# є ефективним і сприятиме успішній розробці та

ефективному функціонуванню підсистеми самообслуговування в кафетерію “SimpleBite”.

3.4.2. Розрахунок та визначення топології комп’ютерної мережі

У зв'язку з тим, що в кафетерії обмежена кількість пристроїв, оптимальним вибором топології буде зіркова. У зірковій мережі всі пристрої підключені до центрального комутатора або роутера. Це означає, що кожен пристрій має своє окреме підключення до центрального вузла. Це спрощує управління мережею та гарантує надійність, оскільки відмова одного пристрою не впливає на іншу частину мережі. Крім того, зіркова топологія полегшує процес розширення мережі, адже новий обладнання легко додаються до центрального комутатора.

3.4.3. Обґрунтування вибору ОС та протоколу обміну даними

Вибір операційної системи обґрунтовується тим, що Android є однією з найпоширеніших ОС для різних пристроїв і користується популярністю серед багатьох людей. Гнучкість та налаштування Android надають розробникам великі можливості для налаштування, адаптації та створення додатків. Це дозволяє пристосовувати додаток під конкретні потреби.

Android підтримує широкий спектр функцій, таких як штучний інтелект, голосове розпізнавання, мультимедійні можливості тощо. Це може забезпечити додатковий функціонал для програми.

Так як в закладу буде локальна система, тобто додаток буде працювати в межах внутрішньої мережі без доступу до зовнішнього Інтернету, то протокол захисту TCP/IP: TCP/IP буде гарним рішенням для мережевого обміну даними в локальній мережі. Він забезпечує надійний транспорт даних між пристроями, дозволяє побудувати з'єднання між додатком самообслуговування та серверною системою кафетерію і передавати дані.

3.4.4. Розробка і обґрунтування стратегії адміністрування системи

Адміністрування системи самообслуговування кафетерію “SimpleBite” є ключовим елементом для забезпечення ефективної та неперервної роботи цієї системи. Основна мета адміністрування полягає в тому, щоб система завжди працювала належним чином, задовольняла потреби користувачів та покращувала продуктивність. Крім того, забезпечення безпеки також є важливим завданням, оскільки система зберігає конфіденційні дані користувачів. Дотримання необхідних заходів безпеки допоможе запобігти несанкціонованому доступу та шкідливим діям.

Планування розвитку системи є важливою складовою для забезпечення конкурентоспроможності та відповідності змінним потребам бізнесу. Розробник повинен постійно вдосконалювати систему, впроваджувати нові функції та збільшувати її функціональність в залежності від вимог та оновлень в галузі самообслуговування.

3.4.5. Заходи захисту від несанкціонованого доступу до системи

Для захисту системи від несанкціонованого доступу слід прийняти наступні заходи безпеки:

- Вимагати встановлення складних паролів з комбінацією великих і малих літер, цифр і спеціальних символів.
- Захистити фізичний доступ до серверів. Обмежити доступ до цих пристроїв тільки необхідному персоналу.
- Використовувати останні версії програмного забезпечення, включаючи операційні системи. Регулярно оновлювати патчі безпеки для захисту від відомих вразливостей.
- Обмежити підключення зовнішніх пристроїв, таких як USB-флешки або зовнішні жорсткі диски, для запобігання встановлення шкідливого програмного забезпечення.

- Проводити навчання з питань кібербезпеки для всього персоналу, щоб підвищити усвідомлення про потенційні загрози та процедури забезпечення безпеки.

Ці заходи сприятимуть забезпеченню надійного захисту системи самообслуговування від несанкціонованого доступу та збережуть конфіденційність, цілісність та доступність даних.

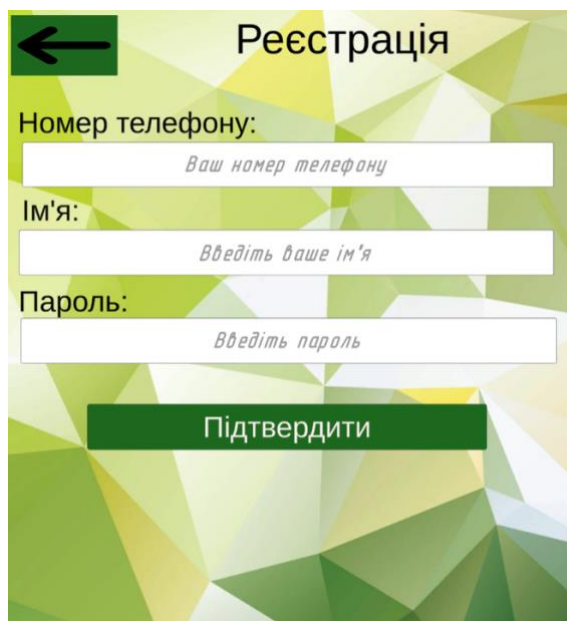
3.5. Інструкція користувача

Після запуску програми “SimpleBite” відображається привітання разом із полями для входу, а також кнопками для продовження авторизації користувача у програмі бонусів. Незареєстрований користувач може перейти до використання програми без авторизації або зареєструватися.



Рисунок 3.9 — Вікно «Авторизація»

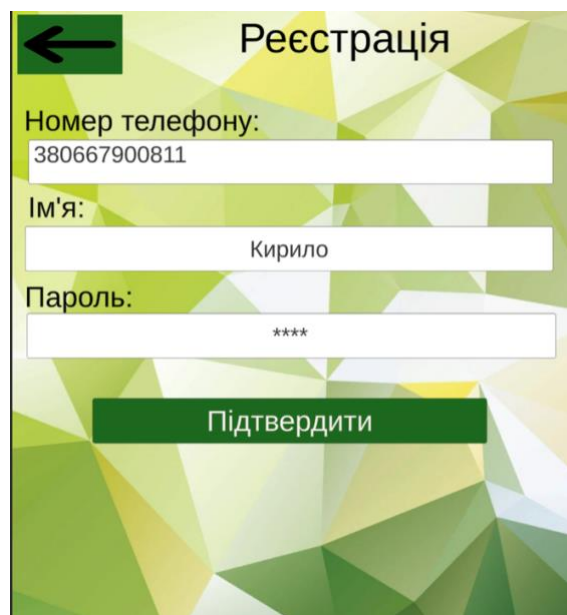
Після натиснення кнопки реєстрації відкривається форма «Реєстрація», для проведення реєстрації в програмі бонусів.



The screenshot shows a mobile application interface for registration. At the top left is a green arrow pointing left. The title 'Регістрація' is centered at the top. Below the title are three input fields: 'Номер телефону:' with the placeholder 'Ваш номер телефону', 'Ім'я:' with the placeholder 'Введіть ваше ім'я', and 'Пароль:' with the placeholder 'Введіть пароль'. At the bottom center is a green button labeled 'Підтвердити'.

Рисунок 3.10 — Вікно «Регістрація»

Після того, як вікно відкрилося, користувачу необхідно заповнити поля з інформацією (номер телефону, ім'я, пароль), щоб програма працювала належним чином.



The screenshot shows the same registration screen as Figure 3.10, but with the input fields filled. The 'Номер телефону:' field contains '380667900811', the 'Ім'я:' field contains 'Кирило', and the 'Пароль:' field contains '****'. The 'Підтвердити' button remains at the bottom.

Рисунок 3.11 — Вікно «Регістрація» з заповненими даними користувача.

Після підтвердження, користувач побачить що реєстрація була успішною і повернеться на головний екран.

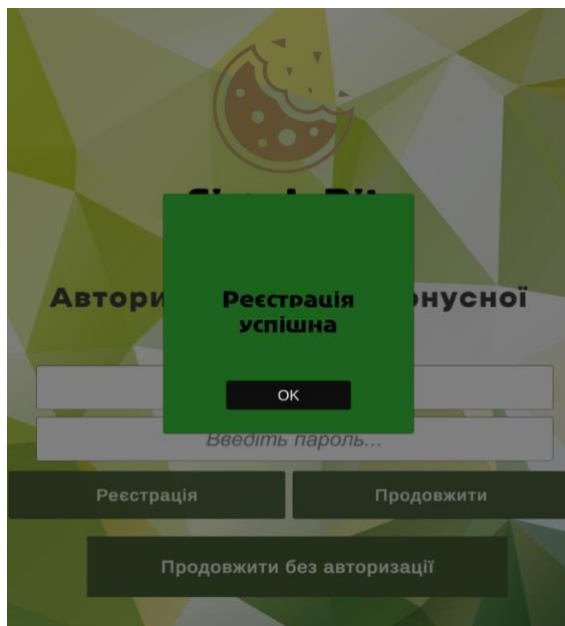


Рисунок 3.12 — Спливаюче вікно про успішну реєстрацію особи.

Після того як користувач успішно зареєструвався, він може авторизуватися. Після авторизації особа переміститься в меню зі списком страв.

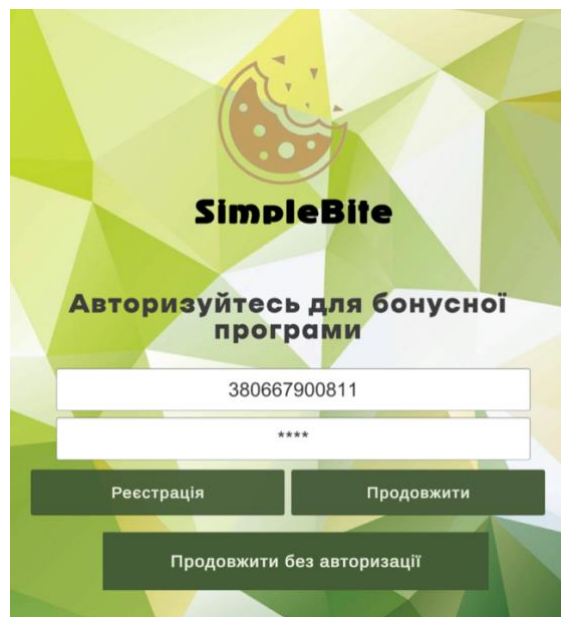


Рисунок 3.13 — Заповнені поля для авторизації

Після того як користувач натиснув кнопку «Продовжити», він бачить меню, в якому є позиції страв в яких є:

- Назва;
- Ціна;
- Кількість;



Рисунок 3.14 — Меню з стравами

Після того як користувач побачив Меню він може його обмежити використовуючи наступні можливості фільтрації:

- Всі страви;
- Напої;
- Солодке;
- Основні страви;

При натисканні на певний фільтр можливо побачити позицію згідно умови фільтрації.



Рисунок 3.15 — Меню з відфільтрованими стравами по критерію «Напої»

Після того як користувач відфільтрував всі позиції які його цікавлять, він може додавати кількість страв за допомогою кнопок. Натиснувши кнопку «+», кількість збільшується, а при натисканні кнопки «-» кількість зменшується.



Рисунок 3.16 — Меню з доданою позицією напою «Мохіто»



Рисунок 3.17 — Меню з зменшеною позицією напою «Мохіто» та іншими позиціями товарів

Після додавання необхідних позицій користувач може використати стрілку «Назад» та кошик, який сформує остаточне замовлення. Біля кошику є індикатор кількості загальної кількості обраних страв.



Рисунок 3.18 — Меню з індикатором замовлень біля кошику

Після натискання на кнопку користувач перейде на сторінку кошика з замовленням, в якому наведені всі позиції товарів та їх кількість.



Рисунок 3.19 — Кошик з сформованим замовленням

При переході в кошик, клієнт може бачити замовлення, кількість доступних йому бонусів (зверху сторінки), сума замовлення, кнопку «Використати бонуси» та кнопку «Оплата».



Рисунок 3.20 — Кошик з використаними бонусами та сумою замовлення

При натисканні кнопки «Використати бонуси» будуть списані бонуси та загальна сума замовлення зменшується за рахунок їх використання.

Користувач може оплатити банківською карткою чи готівкою на касі своє замовлення.

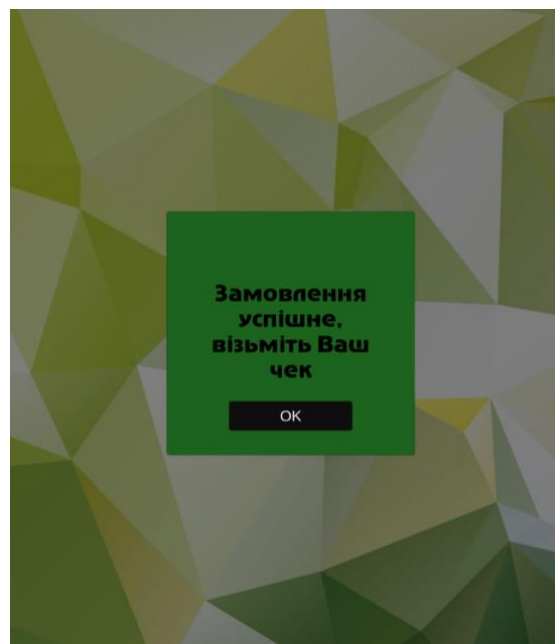


Рисунок 3.21 — Спливаюче вікно успішної оплати замовлення

Після оформлення замовлення чи оплати, клієнт побачить спливаюче вікно про успішне замовлення та отримання чеку із номером замовлення. Після отримання чеку, система автоматично перенаправляє на головну сторінку

авторизації, де наступний особа має авторизуватися або здійснити замовлення без авторизації.

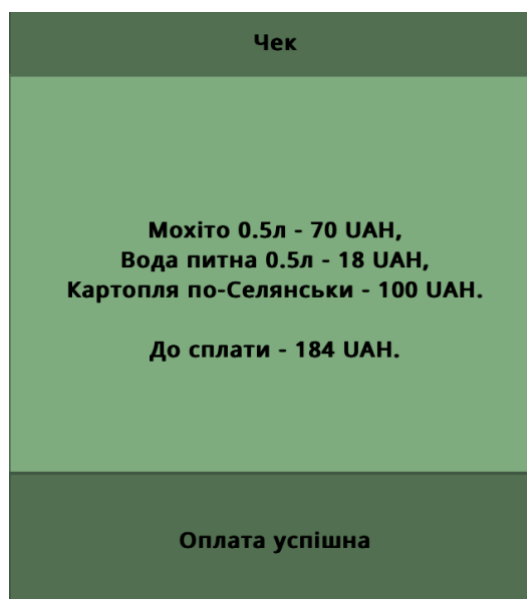


Рисунок 3.22 — Чек

РОЗДІЛ 4. ОХОРОНА ПРАЦІ

4.1. Загальні положення

4.1.1. Інструкція застосовується до всіх відділів підприємства, де виконують завдання з електронним обладнанням.

4.1.2. Інструкція створена відповідно до Положення про розробку інструкцій з безпеки праці, затвердженого наказом Держпраці від 29.01.1998 № 9, Положення про порядок проведення навчання і перевірки знань з питань безпеки праці, затвердженого наказом Держпраці від 26.01.2005 № 15, Вимог щодо безпеки та охорони здоров'я працівників під час роботи з екранними пристроями, затверджених наказом Мінсоцполітики від 14.02.2018 № 207, Державних санітарних правил і норм роботи з дисплейними пристроями електронно-обчислювальних машин ДСанПіН 3.3.2.007-98, затверджених постановою Головного державного санітарного лікаря України від 10.12.1998 № 7, Загальних вимог щодо забезпечення роботодавцем охорони праці працівників, затверджених наказом ДСНС від 25.01.2012 № 67 (НПАОП 0.00-7.11-12).

4.1.3. Згідно з цією інструкцією, працівника, який використовує електронно-обчислювальні пристрої, навчають перед початком роботи (первинний інструктаж), а потім кожні 6 місяців (повторний інструктаж). Результати інструктажу записують у Журнал реєстрації інструктажів з питань охорони праці на робочому місці (у журналі має бути підпис інструктора та працівника).

4.1.4. Працівник повинен піклуватися про свою безпеку і здоров'я, а також про безпеку і здоров'я інших осіб при виконанні будь-яких завдань, а також під час знаходження на території підприємства.

4.1.5. До роботи на пристроях допускають осіб, які пройшли інструктажі з питань охорони праці та пожежної безпеки.

4.1.6. Користувач зобов'язаний:

- Виконувати правила внутрішнього трудового розпорядку;

- не виконувати вказівок, які суперечать правилам охорони праці та пожежної безпеки;
- знати правила надання домедичної допомоги;
- знати розташування та вміти користуватись первинними засобами гасіння пожежі;
- вміти працювати з пристроєм самообслуговування.

4.1.7. Основні небезпечні та шкідливі виробничі фактори, що можуть впливати на користувача [9]:

- Підвищений рівень статичної електрики;
- нерівномірність розподілу освітлення в полі зору;
- підвищена яскравість зображень;
- ураження електричним струмом;
- напруження зору та уваги;

4.1.8. У приміщеннях із пристроєм самообслуговування має бути природне і штучне освітлення.

4.1.9. При облаштуванні робочого місця необхідно уникнути прямого засвічування монітора природним світлом.

4.1.10. При природному освітленні необхідно забезпечити наявність сонцезахисних засобів (плівка, жалюзі, завіси тощо).

4.1.11. Світлові відблиски із клавіатури, екрана та інших частин пристрою у напрямку очей користувача неприпустимі.

4.1.12. Основним обладнанням робочого місця є пристрій самообслуговування, монітор, маніпулятор.

4.1.13. При розміщенні елементів робочого місця слід враховувати:

- Простір для розміщення користувача;
- Можливість огляду елементів робочого місця;
- Можливість огляду простору поза межами робочого місця;

4.1.14. Розміщення елементів робочого місця не має заважати рухам та переміщенню для експлуатування.

4.1.15. Екран встановлюють так, щоб відстань від поверхні екрана до очей користувача була 600-700 мм залежно від розміру екрана.

4.1.16. Сенсорна клавіатуру має бути гарно видна та освітлена, розмір літер та кнопок має бути зручним і легко зчитувати для комфорту очей.

4.1.17. Конструкція зони самообслуговування має бути такою, щоб оптимально розмістити обладнання, що використовують, з урахуванням кількості, розмірів, конструктивних особливостей і характеру його роботи [10].

4.1.18. Пристрій самообслуговування встановлювати на рівній твердій поверхні (підлозі). Не дозволено встановлювати пристрій на хитких підставках чи на похилій поверхні.

4.1.19. Пристрій самообслуговування не встановлювати впритул до стіни, перегородки тощо.

4.1.20. Розетка біля пристрою самообслуговування має бути в доступному місці, щоб в аварійних випадках можна було своєчасно його вимкнути. Не рекомендовано використовувати подовжувачі.

4.1.21. Під час переміщення пристрою самообслуговування, периферійних пристроїв витягти вилку живлення з розетки.

4.1.22. Не дозволяти пошкодження чи зміни кабелю живлення. Заборонено класти важкі предмети на кабель живлення, тягнути чи надмірно перегинати його, скручувати та зав'язувати кабель у вузол.

4.1.23. Пристрій самообслуговування під'єднувати до електромережі лише за допомогою справних штепсельних з'єднань та електророзеток заводського виробництва.

4.1.24. Штепсельні з'єднання та розетки повинні бути зі спеціальними контактами для підключення захисного нульового провідника. Їхня конструкція повинна забезпечувати підключення захисного нульового провідника перед підключенням фазного та нульового робочого провідників. Порядок відключення при вимкненні має бути зворотним.

4.1.25. Заборонено під'єднати електрообладнання до звичайної двошнурової електромережі.

4.1.26. За невиконання цієї інструкції працівники несуть відповідальність згідно з чинним законодавством.

4.2. Вимоги безпеки перед початком роботи

4.2.1. Оглянути робоче місце і привести його в порядок; переконатися, що на ньому немає сторонніх предметів, все обладнання і блоки пристрою самообслуговування підключені.

4.2.2. Перевірити надійність встановлення апаратури на робочому місці. Екран не має стояти похило.

4.2.3. Перевірити загальний стан апаратури, справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана.

4.2.4. Вставити вилку в розетку і впевнитися, що вона міцно тримається. Заборонено вставляти і виймати вилку мокрими руками.

4.2.6. За необхідності підключити до комп'ютера потрібне обладнання (принтер, платіжний термінал тощо). Усі кабелі, що з'єднують пристрій самообслуговування з іншими приладами, підключати та відключати лише при вимкненому пристрої.

4.2.7. Відрегулювати яскравість світла, контрастність монітора.

4.2.8. Про всі виявлені несправності інформувати керівника робіт і не експлуатувати, доки їх не буде усунено.

4.3. Вимоги безпеки під час виконання роботи

4.3.1. Під час роботи пристрою самообслуговування:

- Стабільно встановити пристрій на робочій поверхні, не допускаючи її хитання, водночас передбачити можливість її поворотів та переміщень;
- Періодично при вимкненому пристрої прибирати пил із поверхонь апаратури спеціальними серветками.

4.3.2. При роботі з пристроєм самообслуговування заборонено:

- Самостійно розбирати та ремонтувати пристрій, термінали оплати, принтери, сканери, тощо;
- Встромляти сторонні предмети до вентиляційних отворів пристрою;
- Ставити на пристрій та периферійні пристрої металеві предмети, ємності з водою (вази, горщики для квітів, склянки), оскільки через потрапляння води у середину апарата може статися пожежа або ураження електрострумом.

4.4. Вимоги безпеки після закінчення роботи

4.4.1. Зберегти інформацію.

4.4.2. Вимкнути пристрій самообслуговування.

4.4.3. Вимкнути стабілізатор, якщо комп'ютер під'єднаний до мережі через нього.

4.4.4. Прибрати робоче місце.

4.5. Вимоги безпеки в аварійній ситуації

4.5.1. Нещасні випадки та ризиковані ситуації під час роботи на пристроях самообслуговування можуть виникнути внаслідок: короткого замикання, перевантаження блоку живлення системного блоку, перегрівання, пожежі, тощо.

4.5.2. У разі виникнення аварії або ситуації, що може призвести до аварії, нещасного випадку, негайно від'єднати пристрій самообслуговування від електромережі, повідомити про інцидент керівникові.

4.5.3. Не допускати в небезпечну зону сторонніх осіб.

4.5.4. У разі нещасного випадку, зберегти ситуацію на робочій зоні та пристроях у такому стані, в якому вони були на момент події (якщо це не загрожує життю і здоров'ю інших працівників та людей і не призведе до більш серйозних наслідків). Повідомити про подію керівника робіт або іншу відповідальну особу

та діяти відповідно до його вказівок. Прийняти заходи, щоб уникнути подібних ситуацій у майбутньому.

4.5.5. У разі виникнення пожежі (ознак горіння), сповістити керівництво, при необхідності, зателефонувати оперативно-рятувальну службу за номером 101 або 112 (назвати адресу та місце виникнення пожежі, кількість людей на місці, повідомити своє прізвище) та вжити можливих заходів безпеки для евакуації людей, гасіння (локалізації) пожежі наявними засобами протипожежного захисту. Пам'ятати, що гасити електротехнічні пристрої, які перебувають під напругою, слід лише після їх відключення від електричної мережі. Гасити полум'я за допомогою вуглекислотних або порошкових вогнегасників, в окремих випадках — сухим піском.

4.5.6. У разі необхідності надати постраждалому першу медичну допомогу відповідно до правил, що діють на підприємстві. Якщо стан потерпілого подальше загострюється, без припинення надання медичної допомоги, потрібно зателефонувати за номером "103" для виклику швидкої медичної допомоги.

4.5.7. Виконувати вказівки керівника робіт для ліквідування небезпеки.

ВИСНОВОК

В процесі написання кваліфікаційної роботи було розроблено додаток для он-лайн замовлення продукції кафетерію.

Даний додаток реалізує наступні функції:

- авторизація користувачів;
- реєстрація користувачів;
- перегляд меню;
- можливість сформувати замовлення;
- можливість додавати, редагувати позиції замовлення;
- можливість оплати;
- зменшення черг та часу обслуговування;
- покращення зручності для клієнтів шляхом самообслуговування.

У майбутньому програму можна буде розширювати, створювати додаткові функції, інтегрувати з іншими системами та створювати версії для інших платформ.

В результаті імплементації додатка в систему кафетерію “SimpleBite” буде забезпечено:

- збільшення задоволених клієнтів через зменшення часу очікування в черзі і простий та зрозумілий процес замовлення;
- збільшення ефективності роботи кафетерію завдяки автоматизованому обліку даних;
- потенційне збільшення прибутку завдяки збільшенню кількості клієнтів та підвищенню їх задоволення.

Практичне використання одержаних результатів можна буде використати для:

- впровадження додатку самообслуговування в кафетерії, забезпечуючи необхідну підготовку персоналу та надання необхідної підтримки;
- моніторинг та оновлення системи для забезпечення її безперебійної роботи та вдосконалення відповідно до змін та потреб ринку;

- аналіз відгуків клієнтів та врахування їх пропозицій для подальшого вдосконалення системи.

Розроблений мобільний додаток для замовлення їжі у кафетерії “SimpleBite” є перспективним, оскільки він може покращити обслуговування та зробити більше клієнтів задоволеними.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. «Організація готельного господарства» / О. М. Головка, Н. С. Кампов, С. С. Махлинець, Г. В. Симочко. – 35 с.
2. Відповідальний ред. Глушков В. Енциклопедія кібернетики [Електронний ресурс] / В. відповідальний ред. Глушков // Wikipedia.org. – 1974. – Режим доступу до ресурсу: «Енциклопедія кібернетики».
3. ДСТУ 3008:2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. [На заміну ДСТУ 3008-95; чинний від 2017-07-01]. Вид. офіц. Київ ДП «УкрНДНЦ», 2016, 31 с. (Інформація та документація).
4. ДСТУ 3973:2000. Система розроблення та поставлення продукції на виробництво. Вид. офіц. Київ : Держстандарт України, 2001, 20 с.
5. ДСТУ Б В.2.5–82:2016. Електробезпека в будівлях і спорудах. Вимоги до захисних заходів від ураження електричним струмом. [На заміну ДБН В.2.5-27-2006; чинний від 2017-04-01]. Вид. офіц. Київ ДП «УкрНДНЦ», 2016, 110 с.
6. Інструкція з охорони праці при роботі на персональному комп'ютері [Електронний ресурс]. – 2024. – Режим доступу до ресурсу: 1. <https://pro-op.com.ua/article/485-nstruktsya-z-ohoroni-prats-pri-robot-na-personalnomu-kompyuter>.
7. Конспект лекцій з дисципліни [електронний ресурс] "Проектування інформаційних систем" для студентів спеціальності 122 "Комп'ютерні науки" /укл. М'якшило О.М., Харкянен О.В.- К.:НУХТ, 2018 – 48 с.
8. М'якшило, О. М. CASE-технології у проектуванні інформаційних систем [Електронний ресурс] [Текст] : навч. посіб. / О. М. М'якшило, Л. Г. Загоровська. — Київ : НУХТ, 2017. — 190 с. — каф. інформаційних систем.
9. Методичні рекомендації до виконання випускної кваліфікаційної роботи на здобуття освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійної програми «Комп'ютерні науки» ден.

форми навчання [Електрон. ресурс] / уклад. О. М. М'якшило, М. П. Костіков. – К.: НУХТ, 2022. – 34 с.

10. Правила техніки безпеки при експлуатації електрообладнання [Електронний ресурс] – Режим доступу до ресурсу: <https://budtehnika.pp.ua/7985-pravila-tehnki-bezpeki-pri-ekspluatacyi-elektroobladnannya.html>.

11. Проектування та розробка програмного забезпечення [Електронний ресурс] : лабораторний практикум для здобувачів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійних програм «Комп'ютерні науки» та «Інформаційні системи та штучний інтелект» денної та заочної форм навчання / укладачі : О. М. М'якшило, О. В. Харкянен ; Національний університет харчових технологій. – Київ : НУХТ, 2022. – 102 с.

12. Сьогодні оновлений McDonald's відчинив для гостей свої двері [Електронний ресурс] – Режим доступу до ресурсу: https://m.gorod.cn.ua/news_98822.html.

13. Управління ІТ проектами [Електронний ресурс]: методичні рекомендації до самостійної роботи для здобувачів освітнього ступеня «Бакалавр» спеціальності 122 «Комп'ютерні науки» освітньо-професійних програм «Комп'ютерні науки» та «Інформаційні системи та штучний інтелект» денної та заочної форм навч./уклад. С. В. Грибков, О. Л. Сєдих – К.: НУХТ, 2022 – 27 с.

14. Харчування та обслуговування [Електронний ресурс] – Режим доступу до ресурсу: <https://buklib.net/books/32587/>.<https://smartcafe.com.ua/kiosk-samoobslugivaniya>.

15. Що таке кафетерій? [Електронний ресурс] – Режим доступу до ресурсу: <https://web.archive.org/web/20170501012157/http://10000menu.ru/termini/16085-shho-take-kafeterij.html>. 1.

ДОДАТКИ

Додаток А. Логічна Модель Баз Даних

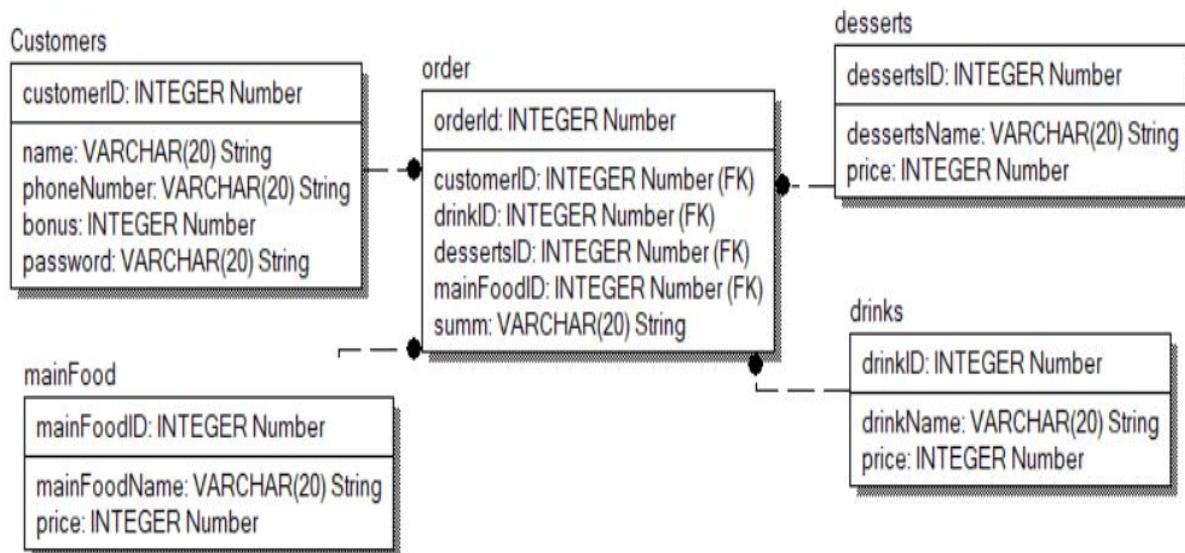


Рисунок А.1 — Логічна модель БД

Додаток Б. Фізична Модель Бази Даних

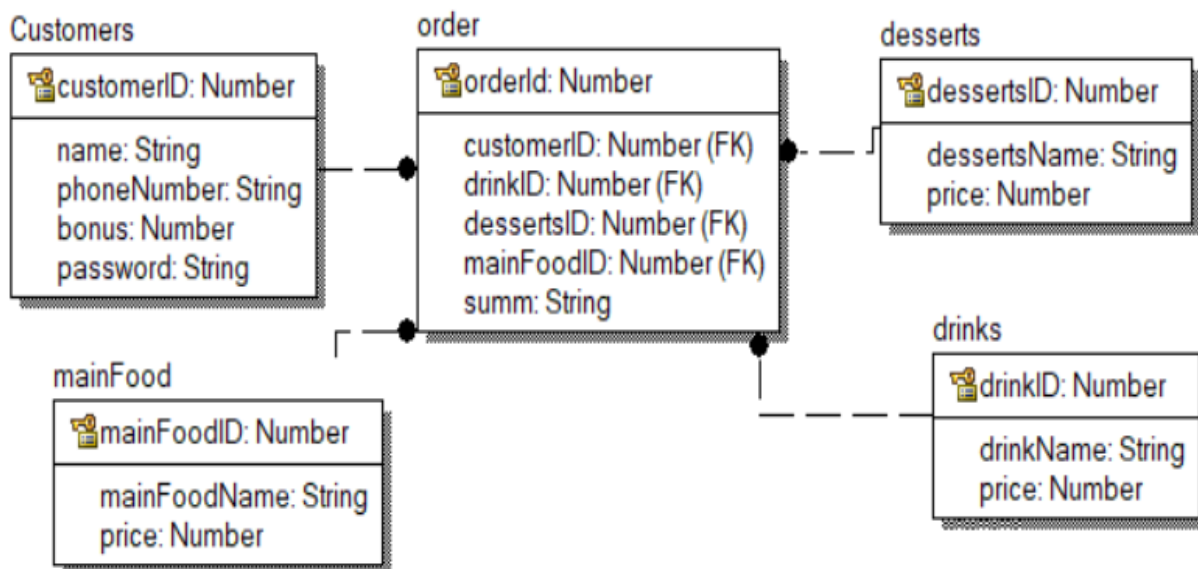


Рисунок Б.1 — Фізична модель БД

Додаток В. Фрагменти Програмного Коду

Програмний код процедури під'єднання до БД

```

private static string GetDatabasePath()
{
    /// <summary> Повертає шлях до БД. Якщо її немає в папці на Android, то
копіює її з вихідного APK файлу. </summary>
    #if UNITY_EDITOR
    // В режимі редактора повертаємо шлях до БД в папці StreamingAssets.
    return Path.Combine(Application.streamingAssetsPath, fileName);
    #elif UNITY_STANDALONE
    // В режимі standalone перевіряємо наявність БД в папці даних.
    string filePath = Path.Combine(Application.dataPath, fileName);
    if (!File.Exists(filePath)) UnpackDatabase(filePath); // Розпаковуємо БД, якщо
вона відсутня.
    return filePath;
    #elif UNITY_ANDROID
    // В режимі Android повертаємо шлях до БД в папці persistentDataPath.
    string filePath = Path.Combine(Application.persistentDataPath, fileName);
    if (!File.Exists(filePath)) UnpackDatabase(filePath); // Розпаковуємо БД, якщо
вона відсутня.
    return filePath;
    #endif
}
/// <summary> Створює підключення до БД. </summary>
private static void OpenConnection()
{
    // Створюємо нове з'єднання з БД за допомогою вказаного шляху.
    connection = new SqlConnection("Data Source=" + DBPath);

```

```
// Створюємо нову команду для взаємодії з БД.  
command = new SqliteCommand(connection);  
// Відкриваємо з'єднання з БД.  
connection.Open();  
}
```

Програмний код створення методу ApplyRegistration

```

private void ApplyRegistration()
{
    // Якщо поля для вводу даних пусті, то викликати вікно помилки вводу
    тексту
    if (string.IsNullOrEmpty(phoneNumberInputField.text) ||
        string.IsNullOrEmpty(nameInputField.text) ||
        string.IsNullOrEmpty(passwordInputField.text))
    {
        // Показ повідомлення про некоректні дані вводу.
        MainMenuUIManager.Instance.errorPopupController.ShowNotificationText(in
correctData);
        return;
    }
    // Перевірка довжини номера телефону
    if (phoneNumberInputField.text.Length < minPhoneNumberCharLenght)
    {
        // Показ повідомлення про неправильний формат номера телефону.
        MainMenuUIManager.Instance.errorPopupController.ShowNotificationText(in
correctPhoneNumberFormat);
        return;
    }
    // Перевірка наявності користувача в базі даних за допомогою SQL-запиту.
    string phonenumber = MyDataBase.ExecuteQueryWithAnswer($"SELECT
phoneNumber FROM Customers WHERE phoneNumber =
'{phoneNumberInputField.text}'");

    // Якщо користувач вже існує, вивести повідомлення про помилку.
    if (!string.IsNullOrEmpty(phonenumber))

```

```
{
    MainMenuUIManager.Instance.errorPopupController.ShowNotificationText(us
erAlreadyExist);
    return;
}
// Запис нового користувача в базу даних за допомогою SQL-запиту, якщо
всі вимоги виконані.
MyDataBase.ExecuteQueryWithoutAnswer($"INSERT INTO Customers
(name, phoneNumber, bonus, password) VALUES ('{nameInputField.text}',
'{phoneNumberInputField.text}', {10}, '{passwordInputField.text}')");
// Показ повідомлення про успішну реєстрацію користувача.
MainMenuUIManager.Instance.errorPopupController.ShowNotificationText(re
gistrationSuccess);
// Повернення до головного меню.
BackToMainMenu();
}
```

Додаток Г. Можливі Помилки У Програмі

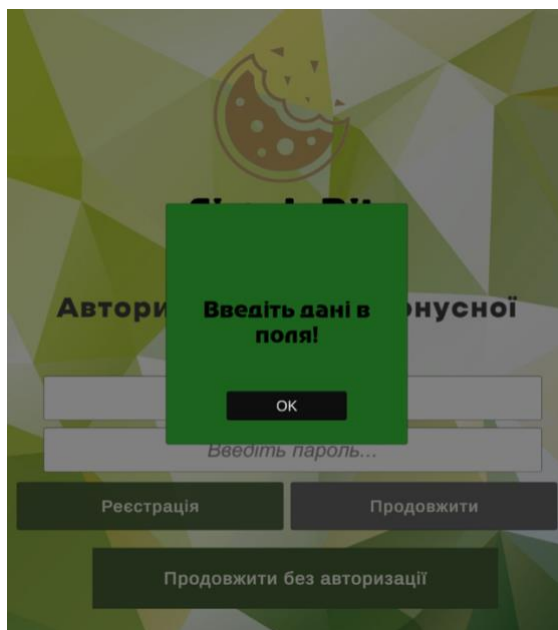


Рисунок Г.1 — Виведення помилки коли користувач вводить пусті поля і намагається авторизуватися

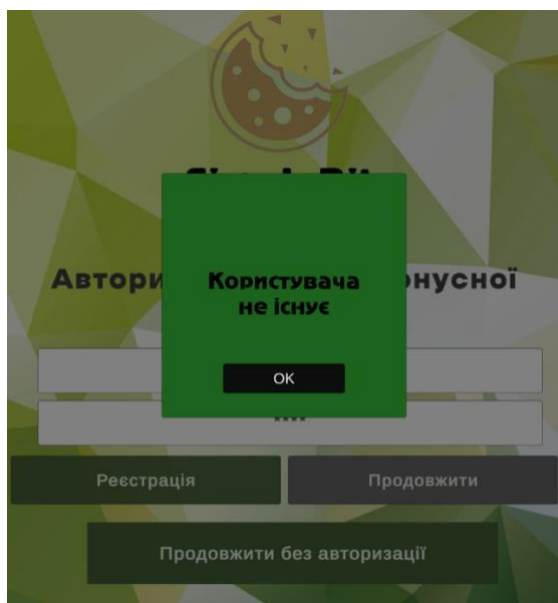


Рисунок Г.2 — Виведення помилки коли користувача не існує при авторизації

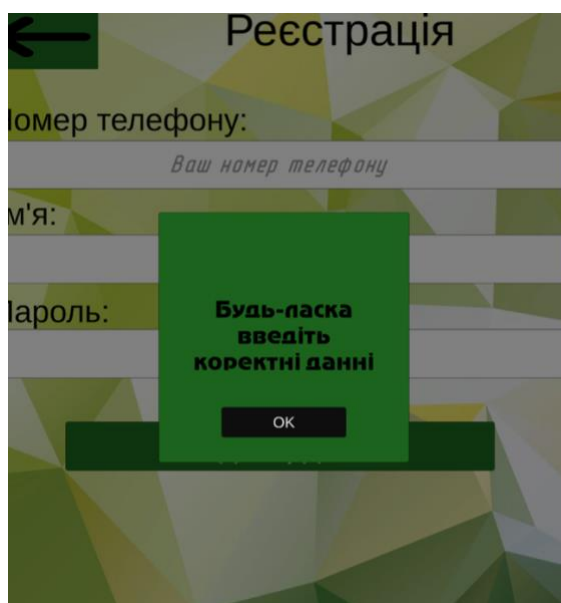


Рисунок Г.3 — Виведення помилки коли користувач при реєстрації залишив пусті поля

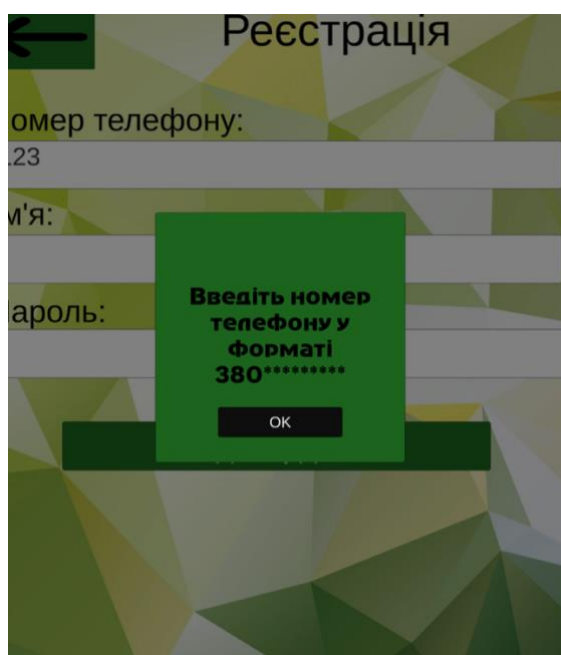


Рисунок Г.4 — Виведення помилки коли користувач вказав номер невірною формату

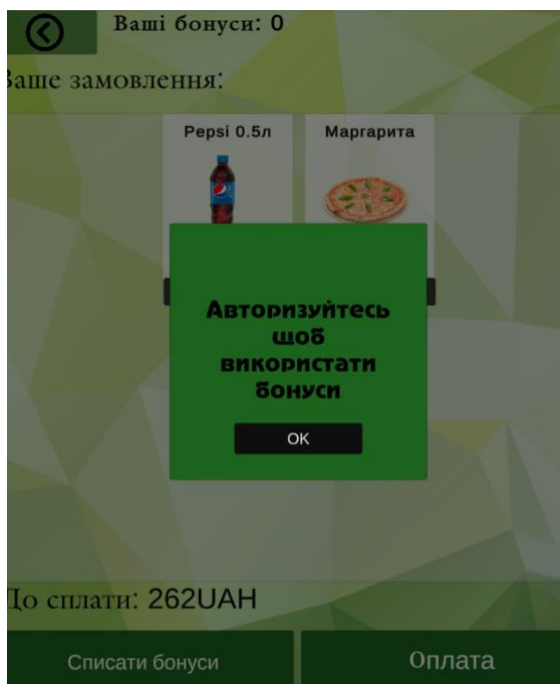


Рисунок Г.5 — Виведення помилки коли не авторизований користувач намагається здійснити оплату бонусами

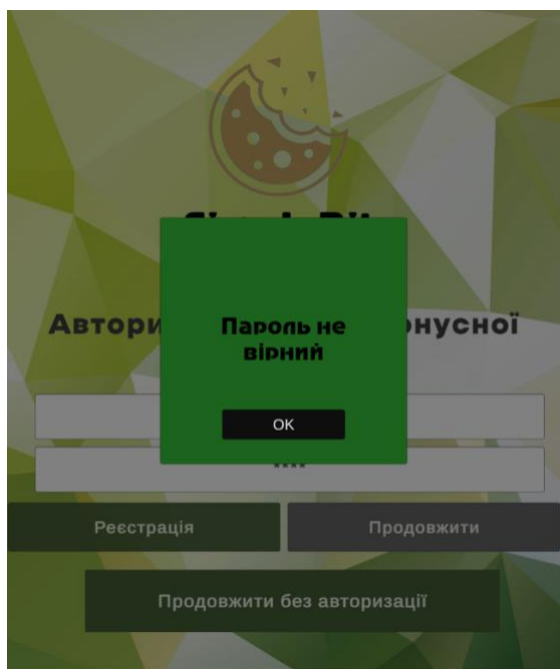


Рисунок Г.6 — Виведення помилки коли користувач ввів невірний пароль

Додаток Д. Відеокадри Інтерфейсу



Рисунок Д.1 — Вікно «Авторизація»

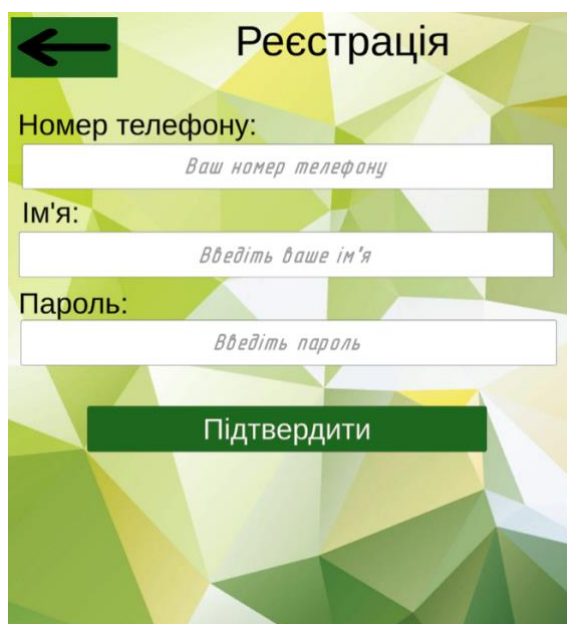
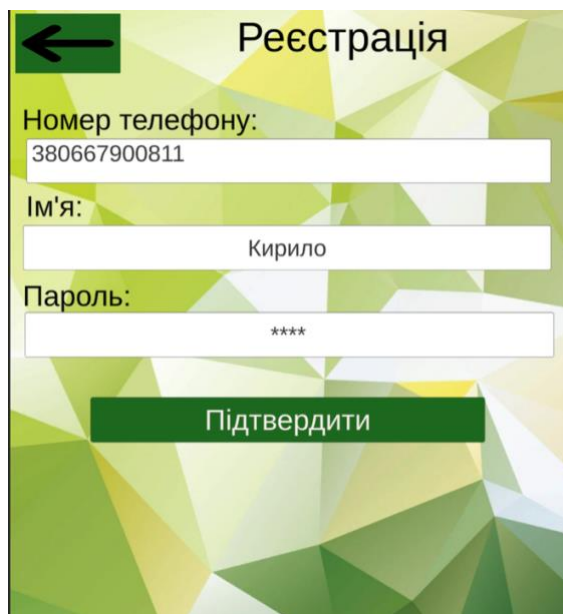


Рисунок Д.2 — Вікно «Реєстрація»



← Реєстрація

Номер телефону:
380667900811

Ім'я:
Кирило

Пароль:

Підтвердити

Рисунок Д.3 — Вікно «Реєстрація» з заповненими даними користувача

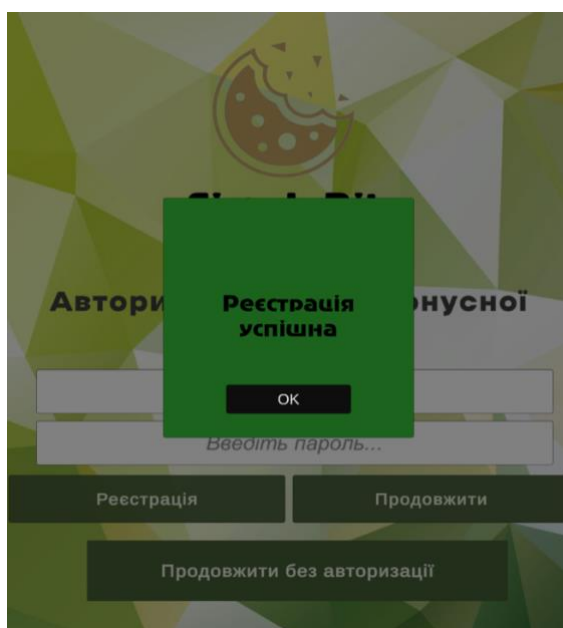


Рисунок Д.4 — Спливаюче вікно про успішну реєстрацію користувача

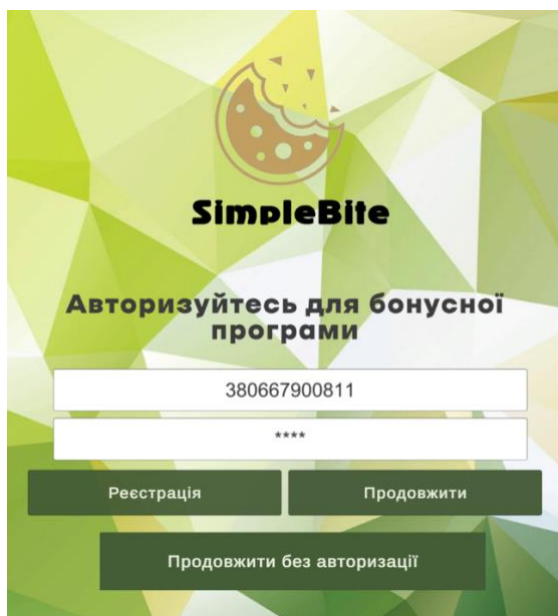


Рисунок Д.5 — Заповнені поля для авторизації



Рисунок Д.6 — Меню з стравами



Рисунок Д.7 — Меню з відфільтрованими стравами по критерію «Напої»



Рисунок Д.8 — Меню з доданою позицією напої «Мохіто»



Рисунок Д.9 — Меню з зменшеною позицією напої «Мохіто» та доданих інших позицій



Рисунок Д.10 — Меню з індикатором замовлень біля кошику



Рисунок Д.11 — Кошик з сформованим замовленням



Рисунок Д.12 — Кошик з використаними бонусами та сумою замовлення

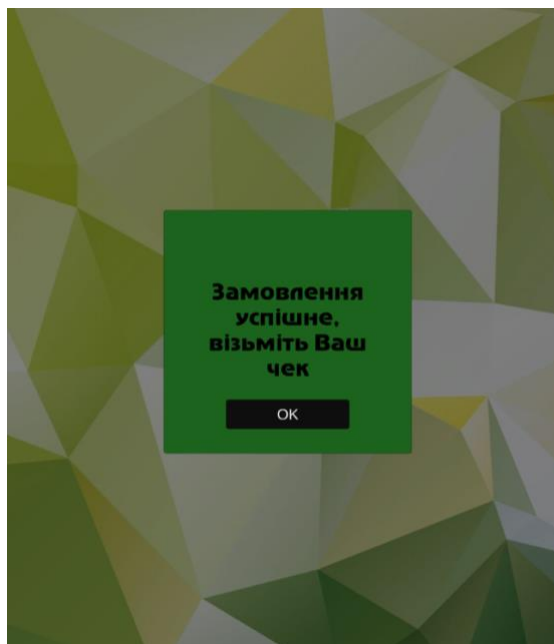


Рисунок Д.13 — Спливаюче вікно успішної оплати замовлення

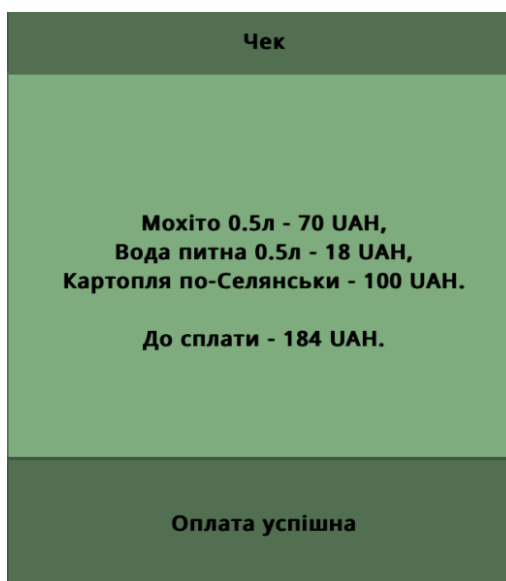


Рисунок Д.14 — Чек