МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УМАНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет економіки і підприємництва

Кафедра інформаційних технологій

Допущено до захисту

"_____" _____ 2025 р.

Зав. кафедри, к.т.н., доцент

_____ **Ліщук Р.І.**

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Розробка додатку для обслуговування клієнтів закладів громадського харчування

Виконав - студент 4 курсу, групи 21к-кн
освітньої програми «Комп'ютерні науки»
спеціальності 122 Комп'ютерні науки
Мамалига Анна Сергіївна
Науковий керівник - д.т.н., професор
Кулаков Павло Ігорович

Умань - 2025

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ……..... 7
ВСТУП…… 8
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ…… 10
1.1. Об’єкт та предмет дослідження…… 10
1.2 Огляд та аналіз аналогів розробки…… 14
1.3. Постановка задачі..... 20
РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ РОБОТИ, СТРУКТУРИ ТА UML-ДІАГРАМ ПРОГРАМНОГО ДОДАТКУ…….... 22
2.1. Розробка структури програмного додатку ……... 22
2.2. Розробка загального алгоритму роботи додатку…… 27
2.3. Розробка UML-діаграм..... 28
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОДАТКУ … 33
3.1 Обґрунтування вибору мови та середовища розробки…… 33
3.2. Розробка додатку..... 34
3.3. Розробка інтерфейсу та тестування додатку …… 39
ВИСНОВОК…… 45
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ…… 46
ДОДАТОК А. Лістинг програми……... 48

Вступ

У сучасному глобальному середовищі, що характеризується стрімким розвитком технологій, мобільні додатки набули статусу невід'ємного інструменту для бізнесу, зокрема в секторі громадського харчування. Вони не лише спрощують процес замовлення, але й істотно покращують загальний рівень сервісу, роблячи його більш ефективним та зручним для споживачів.

Однією з ключових переваг мобільних додатків є забезпечення швидкого доступу до меню, зручного оформлення замовлення та безконтактної оплати. Це не тільки заощаджує час відвідувачів, а й сприяє зниженню навантаження на персонал, тим самим оптимізуючи операційні процеси. Окрім того, мобільні додатки надають функціонал для реалізації програм лояльності, пропонування персоналізованих знижок та ексклюзивних пропозицій, що позитивно впливає на підвищення лояльності клієнтів. Важливу роль відіграє можливість збору та аналізу даних про клієнтів. Заклади громадського харчування можуть ефективно використовувати цю інформацію для вдосконалення сервісу, оптимізації асортименту меню і більш точного прогнозування попиту. Це дозволяє адаптувати операційну стратегію відповідно до вподобань клієнтів, створюючи більш привабливий досвід відвідування. Інноваційні підходи до обслуговування, такі як інтеграція чат-ботів, опції бронювання, попереднє замовлення та можливості інтерактивної взаємодії з клієнтами, додатково підвищують якість послуг та забезпечують максимальний комфорт у взаємодії із закладом. Традиційні процеси, такі як прийом замовлень з використанням паперових меню та блокнотів, можуть призводити до затримок, помилок та зниження рівня задоволеності клієнтів. Ці недоліки потенційно можуть не тільки відштовхнути існуючих клієнтів, але й перешкоджати залученню нових.

Тому розробка спеціалізованого додатку для обслуговування клієнтів у ресторанах набуває все більшої актуальності. Такий додаток має бути обов'язково оснащений сучасними технологіями та інтуїтивно зрозумілим інтерфейсом, що може сприяти оптимізації ресторанного обслуговування.

Актуальність

Таким чином, мобільні додатки та їх розвиток є важливим елементом сучасної ресторанної індустрії, забезпечуючи високоякісний сервіс, зручність для операторів та суттєві конкурентні переваги для закладів громадського харчування. Їх застосування не лише підвищує рівень задоволеності клієнтів, а й сприяє ефективному розвитку бізнесу в цифровому середовищі.

Об’єкт дослідження - процес обслуговування клієнтів у закладах громадського харчування.

Предмет дослідження - програмний додаток для обслуговування клієнтів у закладах громадського харчування.

Мета дослідження

Метою дослідження є розширення функціональних можливостей мобільних застосунків для обслуговування у закладах громадського харчування. На відміну від існуючих аналогів, які не надають можливості візуалізації місця у закладі в режимі онлайн, у даній роботі пропонується інтегрувати таку функцію - можливість вибору та перегляду розташування свого столика в залі закладу до прибуття. Ця функція дозволить клієнтам візуалізувати своє майбутнє місце та обрати оптимальний варіант відповідно до їхніх уподобань та доступності місць.

Для досягнення поставленої мети необхідно:

- провести аналіз ринку програмного забезпечення для обслуговування у закладах громадського харчування.

- провести аналіз функціональних можливостей існуючого аналогічного програмного забезпечення, його переваг та недоліків.

- розробити структурну схему програмного додатку, що розробляється.

- розробити алгоритм роботи програмного додатку.

- обґрунтувати вибір мови програмування та середовища розробки, а також фреймворків та бібліотек для розробки додатка.

- реалізувати програмний додаток.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Об’єкт та предмет дослідження

У сучасному світі, що характеризується стрімкими темпами технологічного розвитку, мобільні додатки стали незамінним інструментом для бізнесу, особливо у сфері громадського харчування. Вони не лише спрощують процес замовлення їжі, а й істотно покращують загальний рівень обслуговування, роблячи його більш ефективним, оперативним та зручним для клієнтів.

Наразі ресторанний бізнес функціонує у висококонкурентному середовищі, що вимагає від закладів постійного вдосконалення послуг та пропозицій з метою задоволення зростаючих потреб споживачів. Ця тенденція зумовлена кількома ключовими факторами. Споживачі стали більш вибагливими та інформованими щодо якості обслуговування та продуктів. Завдяки широкій доступності інформації в мережі Інтернет та соціальних медіа, клієнти мають змогу легко порівнювати різні ресторани, обмінюватися відгуками та обирати оптимальні варіанти.

Така прозорість змушує заклади харчування підвищувати свої стандарти та пропонувати винятковий досвід відвідування. Зростаюча урбанізація та підвищення рівня життя в багатьох країнах спричинили збільшення кількості осіб, які віддають перевагу харчуванню поза межами дому. Це, у свою чергу, призвело до збільшення чисельності ресторанів та закладів харчування, що додатково загострило конкуренцію в галузі. У цьому конкурентному середовищі одним з основних чинників успіху є ефективне обслуговування клієнтів. Приємний та незабутній досвід відвідування ресторану може стати вирішальним фактором, який спонукатиме клієнтів до повторних візитів. Однак, зі зростанням потоку відвідувачів та ускладненням операційних процесів, традиційні методи управління замовленнями та обслуговування клієнтів можуть виявитися недостатньо ефективними.

Аналіз предметної області є найважливішим етапом у процесі розробки програмного забезпечення. Він дозволяє поглиблено зрозуміти особливості бізнес-процесів, потреби користувачів та системні вимоги. У цьому підрозділі буде розглянуто основні аспекти функціонування закладів громадського харчування, які необхідно врахувати при розробці додатку для обслуговування клієнтів.

Ресторани активно впроваджують автоматичні системи для скорочення часу обробки замовлень, оптимізації роботи кухні та мінімізації людських помилок. Ці системи включають електронні POS-системи (point of sale), які дозволяють працівникам швидко та достовірно передавати замовлення на кухню [1]. Орім того, використання планшетів для отримання замовлень безпосередньо за столиком клієнта забезпечує уникнення помилок, пов'язаних з ручним введенням даних.

Автоматизація процесів на кухні, зокрема через використання систем управління замовленнями та приготуванням їжі, дозволяє ефективніше розподіляти завдання серед кухарів та скорочувати час очікування. Також автоматичні системи інвентаризації допомагають ресторану контролювати запаси продуктів та своєчасно поповнювати їх, що зменшує ризик нестачі складових страв. Таким чином, впровадження автоматизації в ресторанному бізнесі значно підвищує ефективність, точність та якість обслуговування клієнтів.

Завдяки автоматизації бізнесу в галузі громадського харчування можливо контролювати діяльність всього закладу [2], поліпшити якість обслуговування відвідувачів, запобігти розкраданням та зловживанням з боку найманих працівників, збільшити продуктивність праці співробітників, сформувати систему лояльності постійних клієнтів, планувати майбутні замовлення.

Збільшується популярність використання мобільних додатків та веб-платформ для онлайн-замовлень їжі та її доставки. Клієнти все частіше віддають перевагу зручності та швидкості, які пропонують такі сервіси. Мобільні додатки

дозволяють користувачам переглядати меню, здійснювати замовлення та оплачувати їх безпосередньо зі смартфонів. Веб-платформи надають можливість оформити замовлення через інтернет-браузер, що є зручним для користувачів, які віддають перевагу роботі на персональному комп'ютері.

Додатки для доставки їжі інтегруються з різноманітними платіжними системами, що забезпечує безпечні і швидкі перекази. Завдяки програмам лояльності та акційним пропозиціям, клієнти мають стимули до регулярного використання цих сервісів. Такі технології не тільки покращують досвід клієнтів, але і дозволяють закладам розширювати свою клієнтську базу, збільшувати обсяги продажів та підвищувати загальний прибуток.

Однією з найважливіших переваг сучасного програмного забезпечення для закладів громадського харчування є автоматизація, яка дозволяє суттєво скоротити час виконання завдань та підвищити продуктивність працівників. Інтеграція автоматизованих рішень забезпечує швидке оформлення клієнтських замовлень, їх передавання на кухню та оплату, що значно покращує досвід клієнтів та зменшує ризик помилок. Також автоматизовані системи надають можливість персоналу ефективніше взаємодіяти з клієнтами, пропонуючи персональні послуги та акційні пропозиції.

Програмне забезпечення для закладів харчування містить модулі управління запасами, які дозволяють контролювати кількість продуктів на складі, прогнозувати потреби у продуктах та оптимізувати закупівлі. Автоматизовані системи забезпечують відстеження руху товарів, аналізують та обліковують витрати, забезпечують контроль залишків, що сприяє зниженню витрат та мінімізації відходів.

Сучасні технології автоматизації дозволяють закладам громадського харчування отримувати аналітику про поведінку клієнтів, популярність товарів, обсяг продажів та ефективність маркетингових дій. Аналітичне програмне забезпечення допомагає власникам закладів приймати обґрунтовані рішення,

оновлювати меню відповідно до уподобань клієнтів та впроваджувати ефективні рекламні акції.

Наприклад, системи лояльності до клієнтів, які інтегровані до мобільних додатків, дозволяють аналізувати кількість відвідувань клієнтів, їх улюблені страви та вподобання. Ця обставина надає можливість створювати персоналізовані пропозиції, індивідуальні знижки та бонусні рахунки, що стимулює повернення гостей до закладу.

Завдяки використанню хмарних технологій, власники закладів можуть керувати бізнесом дистанційно, аналізувати фінансові показники, керувати співробітниками та контролювати робочі процеси в режимі реального часу. Це дуже актуально для мережевих закладів, де управління потребує єдиної централізованої автоматичної системи управління.

Наявність інтеграції з мобільними додатками надає можливість клієнтам оформлювати замовлення в онлайн режимі, бронювати місця та проводити безконтактні платежі. Все це суттєво покращує рівень сервісу, робить процеси більш гнучкими та зручними як для гостей, так і для співробітників.

В умовах використання цифрових технологій критично важливим аспектом є забезпечення вимог кібербезпеки. Програмне забезпечення, призначене для закладів громадського харчування, повинно мати механізми захисту персональних даних клієнтів, фінансової інформації та внутрішніх даних.

Впровадження надійних та сучасних систем автентифікації, шифрування транзакцій та захисту баз даних є обов'язковим для уникнення потенційних кіберзагроз і гарантування безпечного використання інформаційних технологій у ресторанному бізнесі. Отже, програмне забезпечення для закладів громадського харчування є ефективним інструментом для управління бізнесом, підвищення рівня якості обслуговування клієнтів та забезпечення конкурентних переваг.

Завдяки впровадженню автоматизації процесів, управлінню залишками, аналітиці, інтеграції з іншими мобільними додатками та забезпеченню кібербезпеки, сучасні технології сприяють вдосконаленню операційної діяльності ресторанів та пропонуванню якісного сервісу у динамічному цифровому середовищі.

Саме тому розробка додатку в рамках даної кваліфікаційної роботи є актуальною. Цей додаток поєднає деякі існуючі функції з додатковими, наразі відсутніми можливостями, що зробить його більш затребуваним у порівнянні з існуючими аналогами.

Таким чином, створюваний додаток не лише відповідатиме сучасним потребам ринку в умовах конкуренції, але й пропонуватиме нові можливості, які дозволять йому виділитися серед конкурентів.

1.2 Огляд та аналіз аналогів розробки

Сучасний ринок програмного забезпечення пропонує широкий спектр рішень, спрямованих на оптимізацію процесу обслуговування клієнтів у ресторанному секторі. Ці програмні продукти можуть бути категоризовані за їх функціональним призначенням. Зокрема, системи управління ресторанами (RMS) є інтегрованими програмними комплексами, розробленими для всебічного менеджменту різноманітних операційних аспектів ресторанного бізнесу. До їхнього функціоналу можуть входити модулі для управління замовленнями, контролю запасів, кадрового менеджменту, фінансового обліку та аналітичної обробки даних [3]. Серед ключових RMS-систем слід виділити Toast. Toast — це програмне рішення, призначене для ефективного управління процесами бронювання столів, розміщення відвідувачів та моніторингу поточного завантаження залу в ресторанному середовищі [4]. Інтерфейс цієї програми містить кілька основних робочих панелей (рис. 1.1):

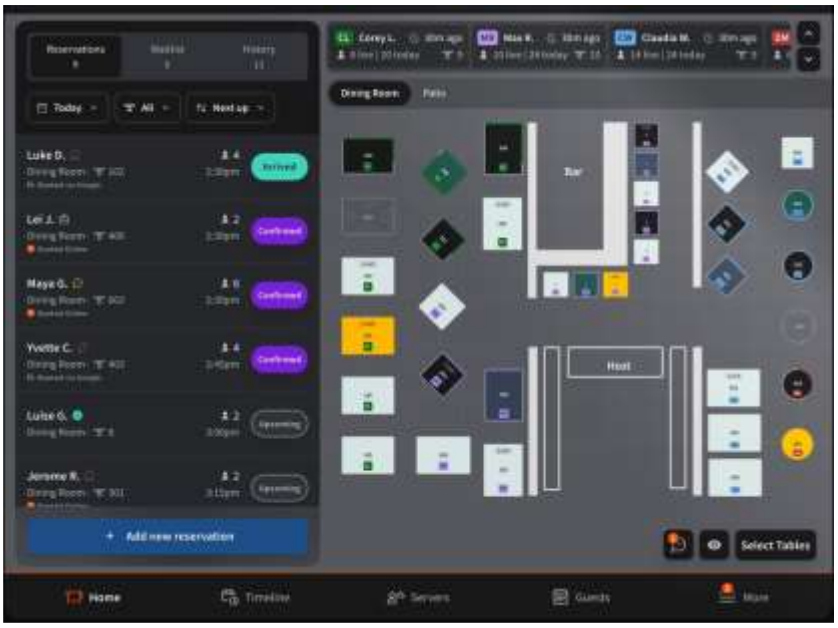


Рисунок 1.1 - Додаток «Toast»

Інтерфейс системи дозволяє візуалізувати актуальні резервування за заданий період або дату, надаючи інформацію про ім'я відвідувача, кількість осіб у групі, час бронювання та статус (наприклад, "прибув" або "очікується"). Панель "Очікування" відображає список гостей, які перебувають у черзі на вільний столик, що сприяє ефективному управлінню потоком відвідувачів. Функціонал "Історія" забезпечує доступ до архівних даних про минулі бронювання та відвідування, що є цінним для аналітичної роботи та формування звітів. Центральна частина інтерфейсу представлена "Картою розміщення" - інтерактивною візуалізацією залу з позначенням розташування столиків, їх поточного статусу (зайнятий, вільний, зарезервований) та відомостями про розміщених гостей. "Інструменти управління" надають персоналу можливість виконувати операції з об'єднання та розділення столиків, переміщення гостей, призначення офіціантів та зміни статусів столиків.

Toast забезпечує інтуїтивно зрозумілий та ефективний механізм для моніторингу та управління резервуваннями, розміщенням гостей та оптимізацією використання столиків у реальному часі. Це сприяє підвищенню ефективності використання простору залу, уникненню конфліктів при розміщенні відвідувачів та забезпеченню високоякісного сервісу. Крім того,

інтеграція Toast з іншими функціональними модулями, такими як управління замовленнями та оплатами, формує комплексне рішення для автоматизації різноманітних операційних процесів у ресторанному бізнесі.

Незважаючи на переваги програми Toast, що проявляються у зручному інтерфейсі та розширених функціях для ресторанного бізнесу, існують певні обмеження. Вартість системи може бути значною для невеликих підприємств, особливо тих, що знаходяться на початковому етапі розвитку, що створює бар'єр для ресторанів з обмеженим бюджетом. Обмежені можливості налаштування: хоча Toast пропонує широкий спектр функціоналу, у деяких випадках можуть виникати ситуації, коли певні можливості або параметри не можуть бути адаптовані до унікальних потреб конкретного закладу. Функціонування сервісів Toast залежить від стабільного інтернет-з'єднання. Відсутність доступу до мережі може призвести до неефективності програмного забезпечення, що спричинить затримки в обслуговуванні та втрату продажів. Крім того, додаток Toast має обмежені інтеграції, що може обмежувати взаємодію з програмними продуктами або сервісами. Також слід враховувати системні вимоги Toast: для оптимальної роботи може знадобитися відповідний рівень технічних ресурсів, включаючи підтримку певних пристроїв або операційних систем.

Upserve (рис. 1.2) є комплексною системою управління ресторанами, яка пропонує потужний інструментарій для оптимізації різних аспектів діяльності закладів харчування [5]. Основні функціональні особливості даного рішення включають комплексне управління замовленнями та меню, що забезпечує гнучке створення й оновлення цифрових меню, доповнених детальною інформацією про страви, візуальними матеріалами та опціями персоналізації. Персонал може ефективно приймати замовлення за допомогою мобільних пристроїв або планшетів, що сприяє значному прискоренню процесу та мінімізації помилок. Система обробки платежів від Upserve гарантує безпечні та зручні транзакції для клієнтів, підтримуючи різні методи оплати, включаючи кредитні картки, мобільні гаманці та готівкові розрахунки.



Рисунок 1.2 - Загальний вигляд «Upserve»

Вона також автоматизує розподіл рахунків та застосування податків і зборів. Функціонал управління столиками та резервуваннями дозволяє в режимі реального часу відстежувати статус столиків, ефективно керувати черговими клієнтами та приймати онлайн-резервування. Це оптимізує використання площі закладу та підвищує якість обслуговування. В рамках аналітики та звітності система збирає та обробляє дані про продажі, популярність страв, продуктивність персоналу, час обслуговування та інші ключові показники. На основі цих даних генеруються деталізовані звіти, що сприяє прийняттю обґрунтованих управлінських рішень. Управління персоналом передбачає відстеження робочого часу співробітників, розподіл завдань та інтеграцію з системами нарахування заробітної плати, що підвищує ефективність кадрового менеджменту та продуктивність праці. Інтеграція та масштабованість системи забезпечують легку взаємодію з іншими програмними рішеннями, зокрема бухгалтерськими програмами, системами лояльності клієнтів та інструментами бізнес-аналітики.

Разом з тим, як і будь-який програмний продукт, система Upserve має певні обмеження. Зокрема, вартість використання може бути значною, особливо для

новостворених або невеликих ресторанних закладів. Деякі користувачі відзначають складнощі в установці та налаштуванні, особливо у випадках інтеграції з іншими програмними продуктами або сервісами. Надходять повідомлення про нестабільність програмного забезпечення, що проявляється у збоях або непередбачуваній поведінці системи. У порівнянні з аналогічними POS-рішеннями, Upserve може мати менші можливості інтеграції з іншими програмами або сервісами. Окремі користувачі також вказують на обмежені можливості звітності та аналітики, що ускладнює поглиблений аналіз ефективності бізнесу. Зазначені недоліки можуть впливати на ефективність та продуктивність впровадження програми Upserve у ресторанній галузі.

TouchBistro є системою управління ресторанами, що пропонує комплексне рішення для автоматизації та оптимізації різноманітних аспектів функціонування закладів харчування (рис. 1.3) [6].

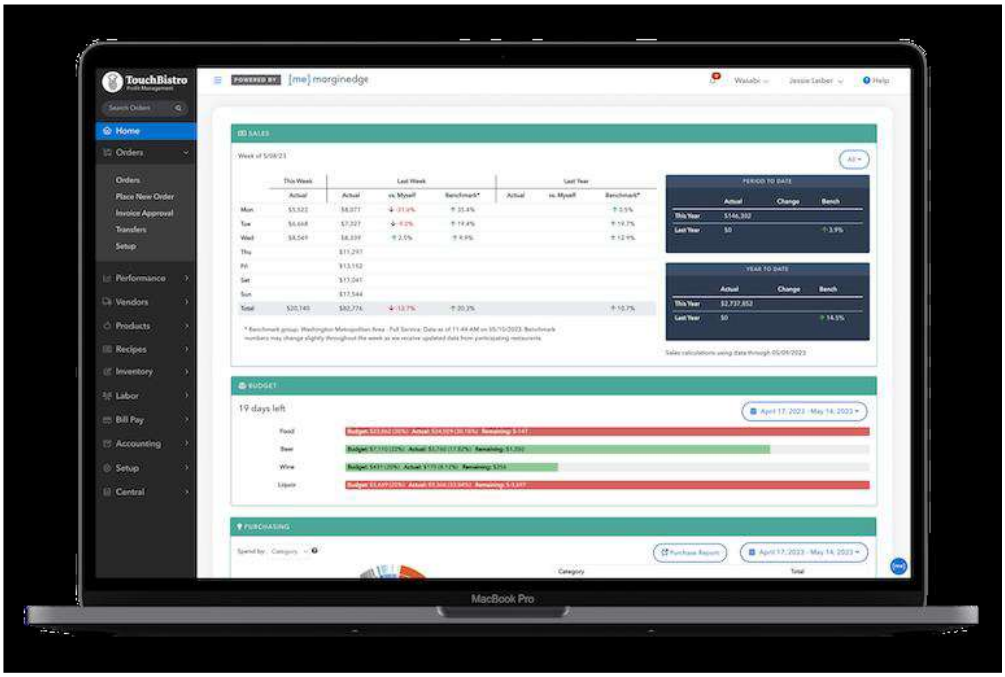


Рисунок 1.3 - Додаток «TouchBistro»

Система TouchBistro пропонує низку ключових можливостей. У сфері управління замовленнями реалізовано цифрове меню з деталізованою інформацією про страви, фотографіями та можливістю внесення оперативних

змін. Офіціанти можуть приймати замовлення безпосередньо біля столиків за допомогою планшетів або мобільних пристроїв. Замовлення автоматично надсилаються на кухню, а їх статус відстежується в режимі реального часу. Також передбачена функція розділення та об'єднання рахунків для груп.

Управління столиками та резервуваннями включає візуальне відображення схеми залу та актуального статусу кожного столика. Система підтримує онлайн-бронювання столиків клієнтами, забезпечуючи ефективне управління чергами та оптимізацію використання посадкових місць.

Обробка платежів здійснюється безпечно та зручно з використанням кредитних карток, мобільних гаманців або готівки. Система автоматично застосовує податки, чайові та інші збори, а також інтегрується з програмами лояльності та знижок.

Управління персоналом охоплює відстеження робочого часу співробітників, розподіл завдань, аналіз продуктивності та генерацію звітів. Також передбачена інтеграція з системами нарахування заробітної плати.

Інвентаризація та управління запасами дозволяє відстежувати рівень запасів інгредієнтів та напоїв у режимі реального часу, автоматично генерувати замовлення для постачальників, оптимізувати витрати та зменшувати відходи.

У розділі звітності та аналітики надаються детальні звіти про продажі, популярність страв, час обслуговування та інші показники. Візуалізація даних сприяє виявленню трендів та прийняттю обґрунтованих управлінських рішень, а також інтеграції з інструментами бізнес-аналітики.

Хмарна інфраструктура та масштабованість характеризують додаток як хмарне рішення, що забезпечує легкий обмін даними. Система масштабується для обробки значних обсягів даних та обслуговування великої кількості клієнтів, а також надає можливість централізованого управління декількома ресторанами в мережі.

Однак, дана програма має низку недоліків, подібних до тих, що притаманні попередньо розглянутим рішенням. Вартість використання TouchBistro може

бути значною для невеликих ресторанів або стартапів.. Відзначаються обмежені можливості налаштування: користувачі повідомляють про складнощі в адаптації системи до конкретних потреб ресторану. Для нових користувачів інтерфейс TouchBistro може здаватися неінтуїтивним, що вимагає часу на освоєння. Оскільки додаток використовує хмарні сервіси, він може стати неефективним у випадку відсутності інтернет-з'єднання, що потенційно призводить до затримок в обслуговуванні та втрати продажів. Ці недоліки можуть вплинути на загальну ефективність впровадження TouchBistro у ресторанному бізнесі.

Аналіз показує, що додатки для офіціантів та персоналу спеціально розроблені для полегшення їхньої роботи. Вони пропонують такі функції, як прийом замовлень за допомогою планшетів, оперативне переміщення столиків, розподіл рахунків, управління чергами тощо. Їхня основна мета - підвищити ефективність роботи персоналу та скоротити час обслуговування.

Системи автоматизованого замовлення та оплати надають клієнтам можливість самостійно розміщувати замовлення та сплачувати рахунки через цифрові кіоски, планшети на столиках або мобільні додатки, мінімізуючи необхідність взаємодії з офіціантами. Вони особливо популярні у закладах швидкого харчування, де значно підвищують пропускну здатність ресторану. Деякі компанії пропонують комплексні інтегровані рішення, що поєднують різноманітні модулі та функції в єдиній системі. Окрім того, існує безліч додаткових інструментів та послуг, таких як системи лояльності клієнтів, управління онлайн-репутацією, аналітика соціальних медіа тощо, які можуть інтегруватися з основними програмними рішеннями для закладів громадського харчування. Деякі недоліки можуть бути загальними для багатьох POS-систем, оскільки вони часто стикаються з подібними викликами та обмеженнями.

1.3 Постановка задачі

Метою кваліфікаційної роботи бакалавра є створення нового додатку, призначеного для обслуговування клієнтів у закладах громадського харчування.

Ця розробка спрямована на розширення функціональних можливостей існуючих додатків для обслуговування клієнтів у даній галузі. На відміну від аналогічних програмних рішень, які не передбачають можливості візуалізації свого місця розташування у закладі в режимі реального часу, в межах даної роботи пропонується інтегрувати таку функцію. Вона полягає у наданні користувачам можливості обрання та перегляду свого столика в залі ще до моменту прибуття. Впровадження цієї функції дозволить клієнтам візуалізувати майбутнє місце розміщення та здійснити оптимальний вибір, ґрунтуючись на власних уподобаннях та актуальній наявності вільних місць.

Для досягнення поставленої мети необхідно:

- провести аналіз ринку програмного забезпечення для обслуговування у закладах громадського харчування.

- провести аналіз функціональних можливостей існуючого аналогічного програмного забезпечення, його переваг та недоліків.

- розробити структурну схему програмного додатку, що розробляється.

- розробити алгоритм роботи програмного додатку.

- обґрунтувати вибір мови програмування та середовища розробки, а також фреймворків та бібліотек для розробки додатка.

- реалізувати програмний додаток.

РОЗДІЛ 2. РОЗРОБКА АЛГОРИТМУ РОБОТИ, СТРУКТУРИ ТА UML-ДІАГРАМ ПРОГРАМНОГО ДОДАТКУ

2.1 Розробка структури програмного додатку

При розробці програмного забезпечення для автоматизації обслуговування клієнтів у сфері громадського харчування, критично важливо провести всебічний аналіз потенційних архітектурних рішень для ключових функціональних компонентів. Цей аналіз є запорукою вибору оптимальних стратегій, що відповідають визначеним цілям проекту та очікуванням кінцевих користувачів. У даній частині ми сфокусуємося на розгляді декількох варіантів імплементації центральних модулів додатку, акцентуючи увагу на їхніх специфічних характеристиках.

Модуль авторизації - розглядаються два основні підходи до реалізації функціоналу автентифікації. Традиційна форма авторизації - метод базується на використанні стандартних полів для введення облікових даних (логіну та пароля), які в подальшому валідуються додатком на відповідність записам у базі даних. Переваги - простота в розробці та інтуїтивно зрозумілий інтерфейс для користувачів. Недоліки - Може вимагати впровадження додаткових механізмів безпеки, таких як двофакторна автентифікація, для підвищення рівня захисту даних. Авторизація через соціальні мережі - альтернативний підхід передбачає інтеграцію з платформами соціальних мереж (наприклад, Facebook, Google), дозволяючи користувачам використовувати вже існуючі облікові записи для доступу. Переваги - забезпечує швидкий та зручний процес входу для користувачів, а також підвищує загальний рівень безпеки за рахунок використання перевірених систем автентифікації. Недоліки - характеризується підвищеною складністю реалізації, а також потребою в постійному налаштуванні та підтримці інтеграції з кожною вибраною соціальною мережею.

Модуль замовлень - при розробці функціоналу для оформлення замовлень розглядаються два ключові підходи. Створення замовлень за допомогою

текстового інтерфейсу - цей метод передбачає, що користувачі вводять замовлення через текстові команди або заповнюють відповідні текстові поля. Переваги - спрощена реалізація та зменшене навантаження на графічний інтерфейс системи. Недоліки - знижена зручність для користувача та підвищена ймовірність помилок при введенні інформації. Створення замовлень за допомогою графічного інтерфейсу - цей підхід передбачає застосування візуальних елементів, таких як кнопки, спадні списки та картки страв, для ініціювання та редагування замовлень. Переваги - забезпечує вищий рівень зручності та інтуїтивності для користувачів, а також мінімізує ризик помилок при введенні даних. Недоліки- характеризується складнішою реалізацією та підвищеним навантаженням на графічний інтерфейс.

Модуль резервацій - для реалізації функціоналу резервацій розглядаються варіанти статичного та інтерактивного календаря. Використання статичного календаря - у цьому випадку користувачі здійснюють вибір дати та часу резервації з фіксованого календаря. Переваги - простота реалізації та легкість розуміння для користувачів. Недоліки - обмежена гнучкість, що ускладнює відображення змін у режимі реального часу. Використання інтерактивного календаря - цей варіант передбачає застосування динамічного календаря, який постійно оновлюється та надає актуальну інформацію про доступність столиків. Переваги - високий рівень гнучкості та точності, а також зручність для користувачів. Недоліки - складніша реалізація та потреба у значних обчислювальних ресурсах.

Модуль меню - при реалізації функціоналу меню розглядаються статичне та динамічне меню. Статичне меню - цей метод передбачає відображення меню у вигляді статичних списків або таблиць, оновлення яких здійснюється адміністраторами вручну. Переваги - характеризується простотою реалізації та мінімальними вимогами до системних ресурсів. Недоліки - обмежена гнучкість у відображенні інформації. Динамічне меню - даний підхід дозволяє автоматичне оновлення меню на основі даних з бази даних, а також надає користувачам

можливість фільтрувати та сортувати страви. Переваги - забезпечує високу гнучкість, зручність у використанні та актуальність даних у режимі реального часу. Недоліки - вимагає складнішої реалізації та створює значніше навантаження на базу даних та інтерфейс.

Модуль вибору місця (специфічне рішення для проєкту) - для поточного проєкту було обрано інноваційне рішення щодо реалізації модуля замовлень, яке включає функціонал вибору місця в закладі за фотографіями та візуалізацію розташування столиків. Ця функція надає клієнтам можливість заздалегідь візуалізувати своє місце, що є унікальною перевагою на ринку. Переваги - забезпечує інноваційний та унікальний досвід для користувачів, значно підвищуючи їхню зручність та задоволеність. Недоліки - вимагає складнішої реалізації та більших ресурсних витрат на підтримку інтерактивного інтерфейсу та актуальності даних.

Початкові концепції, зокрема автентифікація через традиційні форми та формування замовлень за допомогою текстового інтерфейсу, попри свої переваги, не надають необхідного рівня гнучкості та інноваційності, що є критично важливим для сучасного сектору ресторанного бізнесу. На противагу цьому, обрана стратегія, яка передбачає можливість візуального вибору місця на основі фотографій та інтерактивного перегляду плану розміщення столиків, пропонує унікальний користувацький досвід. Це дозволяє диференціювати пропозицію на ринку та суттєво підвищити рівень задоволеності клієнтів.

Розроблюваний програмний додаток буде забезпечувати такі функціональні можливості:

- авторизація користувачів;
- облік замовлень;
- управління резервaціями;
- робота з меню закладу;
- зберігання даних у відповідній базі даних.

Структура програмного забезпечення передбачає наявність наступних основних компонентів (модулів):

Модуль авторизації (класи LoginRegisterApp, User) Цей модуль є початковою точкою взаємодії користувача із системою, забезпечуючи функції входу та реєстрації. Його реалізація здійснюється класом LoginRegisterApp, який відповідає за відображення інтерфейсу входу/реєстрації та керування відповідними процесами. Інтерфейс включає дві вкладки: для існуючих користувачів та для створення нових облікових записів. Процес входу передбачає валідацію облікових даних (логін та пароль) користувача в базі даних. У разі реєстрації нового користувача, до таблиці `users` додається новий запис з унікальним іменем користувача та хешем пароля. Клас User призначений для моделювання даних користувача та реалізації методів взаємодії з його об'єктом, включаючи отримання інформації з бази даних.

Модуль головного меню (клас MainMenuTab) Даний модуль є центральною навігаційною точкою, забезпечуючи перехід між іншими функціональними блоками системи. У фреймі MainMenuTab розташовані віджети для роботи з активними та завершеними замовленнями, майбутніми резерваціями, а також кнопки навігації. Модуль дозволяє здійснювати операції з подачі та оплати страв, оновлення даних та перехід до інших модулів. Важливою особливістю є оптимізоване представлення замовлень, згрупованих за столами, у деревоподібних віджетах, що супроводжується регулярним оновленням, сортуванням та функціоналом пошуку. Передбачено механізм запитів до бази даних для отримання інформації про статус замовлень.

Модуль замовлень (клас OrdersTab) - призначений для функцій створення, редагування та відображення замовлень. Інтерфейс OrdersTab містить динамічне меню, представлене у вигляді карток для кожної позиції. Окрім назви та зображення, картка включає ціну та кнопку додавання до замовлення. При виборі позиції з'являється діалогове вікно для введення кількості та номера столика. Вся інформація про замовлення зберігається у таблиці orders бази даних.

Також реалізовані механізми пошуку страв за назвою та сортування меню. Передбачається можливість розширення функціоналу для редагування чеків або кошиків. Інтуїтивно зрозумілий інтерфейс та можливості керування замовленнями дозволяють підвищити ефективність обслуговування клієнтів.

Модуль резервацій (клас ReservationsTab) Модуль керування столиками та бронюванням. На фреймі ReservationsTab відображається таблиця, що містить поточні та майбутні резервації. Користувачі мають можливість додавати нові бронювання, а також редагувати чи видаляти існуючі. Зберігаються такі дані, як повне ім'я, номер телефону, дата, час, кількість гостей та номер столика. Завдяки цьому модулю реалізовано процес планування та оптимізації роботи закладу харчування, що позитивно впливає на якість обслуговування та фінансові показники. Додаткові діалогові вікна сприяють зручності використання функціоналу резервацій.

Модуль меню (клас MenuTab) - відповідає за управління каталогом страв та їх ціноутворенням, а також є ключовим елементом представлення закладу для клієнтів. Він забезпечує можливість перегляду меню у форматі карток, кожна з яких містить фотографію, назву та ціну страви. Користувачі можуть здійснювати пошук страв, а також активувати або деактивувати їх у меню. Всі дані зберігаються у таблиці Menu в базі даних.

Модуль БД (клас DB, модель data) Цей модуль є центральним для збереження та отримання даних з бази даних MySQL. Його реалізація забезпечується класом DB, який керує виконанням запитів до бази даних (вибірка, вставка, оновлення, видалення) та отриманням відповідних даних. Моделі даних (такі як User, Dish, Order тощо) інкапсулюють поля таблиць та валідатори. Це забезпечує цілісність записів, високу швидкодію та потенціал для майбутнього масштабування системи. Зв'язок з базою даних через ORM (SQLAlchemy) значно спрощує взаємодію з нею для інших модулів.

Крім того, була розроблена структурна схема програмного додатку для обслуговування клієнтів у закладах громадського харчування (рис. 2.1). Ця схема

є важливим етапом у розробці програмного забезпечення, оскільки візуалізує архітектуру системи та демонструє взаємозв'язки між її компонентами.

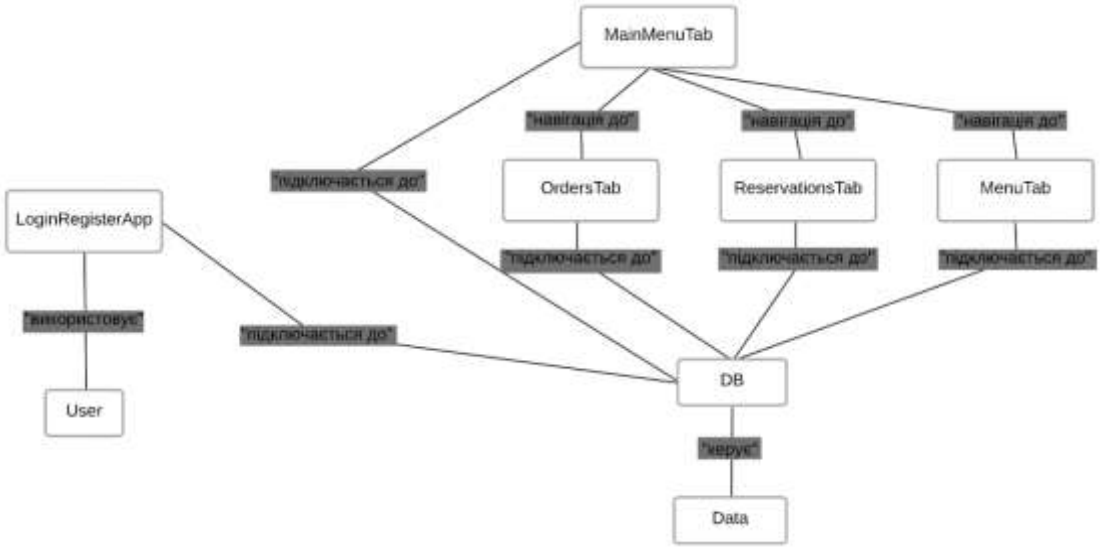


Рисунок 2.1 - Структурна схема програмного додатку для обслуговування клієнтів у закладах громадського харчування

Структурна схема програмного додатку, представлена на рис. 2.1, охоплює шість ключових модулів: модуль авторизації, модуль головного меню, модуль замовлень, модуль резервацій, модуль меню та модуль бази даних. Ця схема наочно демонструє взаємозв'язки та основні потоки даних між зазначеними компонентами. Користувачі взаємодіють із системою через інтерфейси модулів авторизації, головного меню, замовлень та резервацій, які, у свою чергу, звертаються до модуля бази даних для збереження та отримання необхідної інформації. Цей підхід забезпечує високу гнучкість та масштабованість системи, а також спрощує процес подальшого розширення її функціоналу. Кожен модуль виконує чітко визначені завдання, що сприяє ефективності та зручності використання програмного додатка.

2.2. Розробка загального алгоритму роботи додатку

Для повного розуміння функціонування розроблюваного програмного забезпечення було створено алгоритм його роботи, представлений на рис. 2.2.

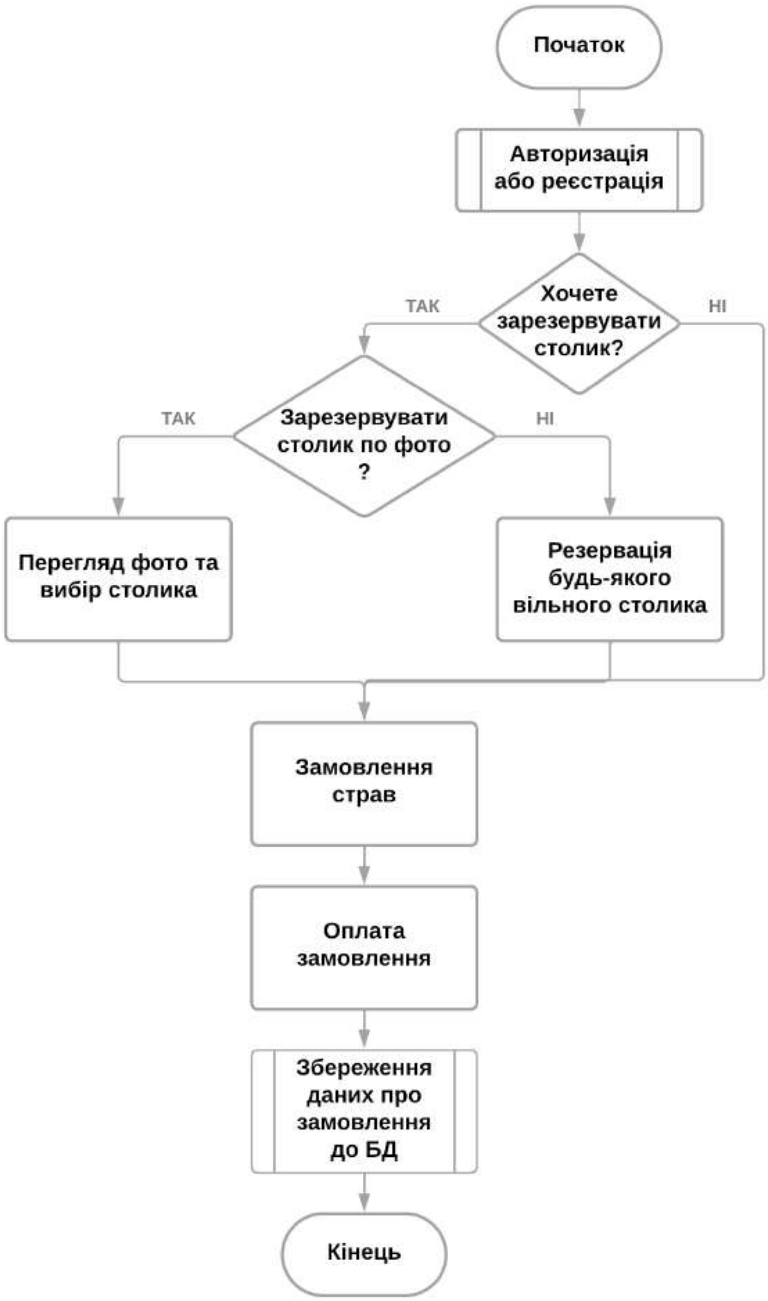


Рисунок 2.2 - Алгоритм роботи програмного додатку для обслуговування клієнтів у закладах громадського харчування

Цей алгоритм детально визначає послідовність операцій, що забезпечують функціональність додатка - від моменту авторизації користувача до завершення процесу обслуговування. Розробка такого алгоритму є критично важливим етапом, оскільки він гарантує ефективну та зручну взаємодію користувачів із системою, що безпосередньо впливає на її загальну продуктивність.

Для ініціювання замовлення користувач зобов'язаний пройти процедуру авторизації або реєстрації нового облікового запису. Після цього йому надається можливість обрати між резервуванням столика в ресторані або безпосереднім оформленням замовлення**.** У випадку вибору опції резервування, клієнт може скористатися розширеним функціоналом додатка, що включає перегляд фотографій інтер'єру ресторану та вибір найбільш прийнятного місця, або ж просто забронювати вільний столик, якщо такий доступний.

Далі слідує стандартна процедура оформлення замовлення, яка передбачає вибір страв та здійснення оплати. Якщо ж користувач не обрав резервування, він одразу переходить до оформлення замовлення, вказуючи адресу доставки та здійснюючи оплату. Після цього замовлення фіксується у базі даних**.**

Представлена логіка функціонування забезпечує ефективність та зручність використання програмного додатка. Кожен модуль чітко виконує свої завдання, гарантуючи плавний та безперебійний процес взаємодії між клієнтом та закладом. Така архітектура дозволяє легко масштабувати додаток та інтегрувати нові функції без суттєвих змін в існуючій логіці. Крім того, централізоване управління даними через модуль бази даних забезпечує надійність та оперативний доступ до необхідної інформації.

2.3. Розробка UML-діаграм

Діаграма варіантів використання (або діаграма прецедентів чи сценарій використання) є інструментом, що візуалізує ролі користувачів (акторів) та їхню взаємодію з системою, демонструючи функціональні вимоги з перспективи користувача, тобто що саме система здатна робити. Важливо зазначити, що ця діаграма не відображає послідовність виконання кроків [7, 8].

На рис. 2.3 представлена діаграма варіантів використання, розроблена для нашого програмного додатка. Вона ілюструє взаємодію між одним актором та шістьма різними сценаріями використання.



Рисунок 2.3 - Діаграма use-cases

Користувач є основним актором у цій системі, що взаємодіє з її різноманітними функціональними можливостями. Нижче наведено деталізацію сценаріїв використання:

- Вхід до системи: Користувач аутентифікується, вводячи свої облікові дані.
- Вихід із системи: Користувач завершує у сесію та виходить із системи.
- Керування резерваціями: Користувач має можливість переглядати, модифікувати та анулювати існуючі резервації.
- Створення замовлення: Користувач додає обрані страви до замовлення.
- Взаємодія з меню: Користувач переглядає перелік страв, їх опис та вибирає позиції для замовлення.
- Перегляд головного меню: Користувач отримує доступ до загального переліку доступних страв у меню.

Ця діаграма ефективно ілюструє ключову функціональність та можливості системи, що значно спрощує взаємодію користувачів із нашим програмним додатком для обслуговування в ресторанах.

Діаграма станів описує всі можливі стани об'єкта та переходи між ними, які спричинені певними подіями [9, 10]. Для програмного додатка, призначеного

для обслуговування клієнтів у ресторанах, нами була розроблена діаграма станів для процесу замовлення страв (рис. 2.4).



Рисунок 2.4 - Діаграма станів для замовлення страв

Приклади станів для сутності "замовлення" включають:

- new: початковий стан, коли замовлення щойно створене користувачем.
- confirmed: стан, що настає після підтвердження користувачем страв.
- served: стан, коли замовлення подано клієнту.
- completed: фінальний стан, що означає повне завершення обслуговування.
- cancelled: стан, що вказує на скасування обслуговування.

Переходи між цими станами визначаються наступним чином:

- Зі стану new до confirmed я перехід після підтвердження замовлення.
- Зі стану in progress до served перехід після подачі страв клієнту.
- Зі стану served до completed перехід після завершення обслуговування.
- З будь-якого стану (за винятком completed) до cancelled можливий перехід у випадку скасування замовлення.

Діаграма класів у UML є статичним представленням структури моделі, що відображає її декларативні елементи, такі як класи, типи даних, їх вміст та відносини [11, 12]. Діаграма класів є ключовим елементом об'єктно-орієнтованого дизайну. Вона дає змогу зрозуміти взаємозв'язки між різними компонентами системи та сприяє ефективному створенню додатка. Завдяки цій

діаграмі можна детально описати структуру об'єктів, які використовуються в додатку, а також визначити їх основні властивості та поведінку. У цьому підрозділі розглянуто UML-діаграму класів для нашого додатка для обслуговування клієнтів у ресторанах (рис. 2.5).

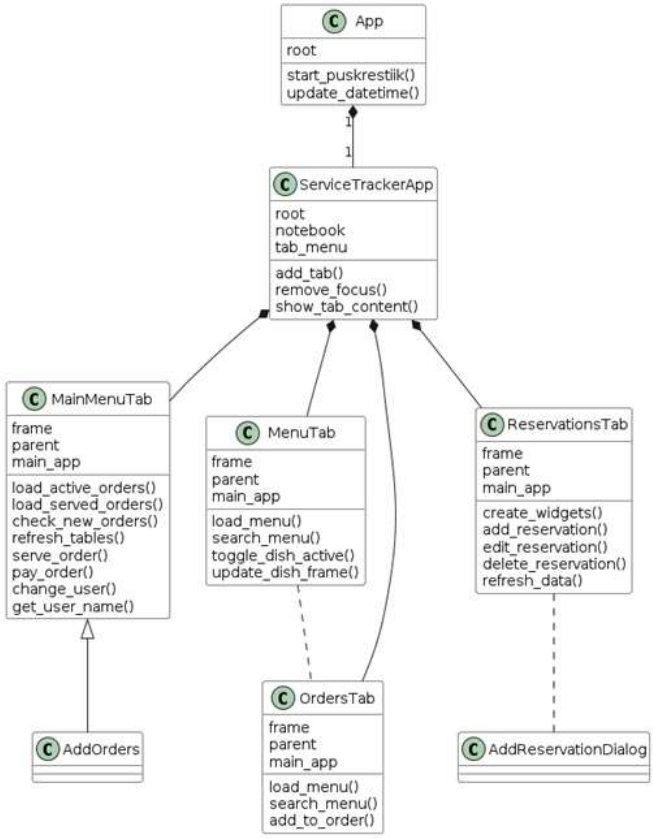


Рисунок 2.5 - Діаграма класів

Отже, розроблено алгоритм роботи програмного додатка для обслуговування клієнтів у закладах громадського харчування. Алгоритм включає покроковий процес авторизації, зручне управління замовленнями та резерваціями, а також можливість вибору столиків за фотографіями. Розробка алгоритму дозволила створити ефективну, надійну та зручну систему, яка підвищує якість обслуговування, підвищує задоволеність клієнтів та оптимізує роботу персоналу. Для розробленого програмного додатка була створена структурна схема, яка відображає архітектуру системи.

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОДАТКУ

3.1. Обґрунтування вибору мови та середовища розробки

Розробка програмного забезпечення для обліку послуг у ресторанній індустрії є критично важливим завданням, що вимагає ретельного підбору технологічного стеку для досягнення оптимальної ефективності, масштабованості та зручності користування. У цьому розділі представлено обґрунтування вибору технологій, які будуть застосовані для реалізації даного проєкту.

Як основну мову програмування обрано Python. Це інтерпретована мова високого рівня, що відзначається лаконічним синтаксисом, високою продуктивністю та легкістю освоєння. Python знаходить широке застосування у веб-розробці, наукових обчисленнях, автоматизації, машинному навчанні та розробці рішень на основі штучного інтелекту [13].

Ключові переваги Python, які зумовили його вибір для цього проєкту, включають: простий та інтуїтивно зрозумілий синтаксис, що сприяє прискореній розробці, спрощує супровід та масштабування застосунку; розгалужена екосистема готових бібліотек та фреймворків, що охоплюють широкий спектр застосувань (веб-розробка, обробка даних, візуалізація, машинне навчання тощо), дозволяючи уникнути розробки надлишкового коду; та кросплатформеність, що забезпечує функціонування розробленого програмного забезпечення на різних операційних системах без значних модифікацій коду.

Для взаємодії з базою даних була обрана бібліотека PyMySQL. Вона є повноцінним клієнтом до системи управління базами даних MySQL, реалізованим виключно на Python [14]. Основними аргументами на користь такого вибору є: повна сумісність з MySQL, що дозволяє використовувати весь функціонал цієї СУБД; простий та зручний API, що спрощує інтеграцію з базою даних; а також підтримка різних типів курсорів, зокрема словникових, що

забезпечує зручну роботу з результатами запитів у вигляді словників, оптимізуючи обробку даних у середовищі Python.

Для операцій з обробки та маніпуляції зображеннями в межах застосунку була використана бібліотека PIL (Python Imaging Library), яка є однією з найбільш поширених та ефективних бібліотек для роботи з зображеннями в екосистемі Python [15].

3.2. Розробка додатку

Під час розробки застосунку було створено шість взаємопов’язаних файлів, кожен з яких виконує специфічну функцію, що забезпечує стабільну роботу та ефективну інтеграцію системних компонентів. Для написання, налагодження та тестування коду використовувалось інтегроване середовище розробки PyCharm, яке надає розширений набір інструментів для якісної та зручної розробки програмного забезпечення. Програмний код забезпечує підключення до бази даних MySQL за допомогою бібліотеки PyMySQL у середовищі Python, при цьому модуль Tkinter відповідає за виведення повідомлень про помилки. У коді визначено параметри з’єднання з базою даних, такі як хост, пароль та назва бази даних. Також реалізовано функцію create_connection, яка ініціює спробу підключення: у разі успішного встановлення з’єднання повертається відповідний об’єкт, а у випадку збою виводиться повідомлення про помилку [16, 17].

Програмний код виконує низку послідовних дій. Він імпортує необхідні бібліотеки: messagebox з tkinter для відображення графічних повідомлень, а також DictCursor з pymysql.cursors, що дозволяє отримувати результати SQL-запитів у форматі словника. Далі задаються параметри підключення до бази даних, включаючи хост, ім’я користувача, пароль та назву бази даних. Визначено функцію create_connection, яка намагається встановити з’єднання з базою даних на основі заданих параметрів. У разі успішного підключення функція повертає об’єкт connection. Якщо підключення не вдається, користувачу відображається діалогове вікно з повідомленням про помилку: "Не вдалося підключитися до бази

даних. Перевірте параметри підключення.", після чого функція повертає значення None.

Файл main.py реалізує механізм автентифікації користувачів у системі ресторанного обліку, забезпечуючи базовий рівень безпеки через хешування паролів та їх зберігання в базі даних. Для досягнення цієї функціональності використовуються різноманітні бібліотеки та модулі, що забезпечують створення графічного інтерфейсу, обробку даних та підтримку багатопоточності.

Код починається з імпорту необхідних бібліотек, кожна з яких виконує специфічні функції:

- tkinter: Використовується для побудови графічного інтерфейсу користувача (GUI).
- datetime: Застосовується для роботи з поточною датою та часом.
- hashlib: Використовується для криптографічного хешування паролів, зокрема за алгоритмом SHA-256.
- subprocess: Призначений для запуску зовнішніх процесів.
- config: З цього модуля імпортується функція create_connection, що відповідає за встановлення з'єднання з базою даних.

Функції та класи, далі визначено ключові функції та класи, що складають ядро застосунку:

- hash_password: Це функція, призначена для генерації хешу пароля за допомогою алгоритму SHA-256, що є важливою частиною механізму безпеки.
- App: Представляє клас головного вікна програми. Він відповідає за створення основного інтерфейсу, що включає привітальне повідомлення, меню для доступу до функцій входу/реєстрації, віджет для відображення поточної дати та часу, а також ініціює виклик вікна автентифікації.
- LoginRegisterApp: Цей клас реалізує вікно входу та реєстрації користувачів. Він містить окремі вкладки для кожної з цих операцій, поля

для введення облікових даних (логіна та пароля) та відповідні кнопки для підтвердження дій.

- run_app: Функція, яка ініціює запуск графічного інтерфейсу в окремому потоці, що дозволяє уникнути блокування виконання основної програми.

Інтерфейс користувача представлений двома основними компонентами:

- Вікно входу/реєстрації: Має дві окремі вкладки — одну для існуючих користувачів (вхід) та іншу для створення нового облікового запису (реєстрація). Передбачено поля для введення імені користувача та пароля, а також кнопки для підтвердження відповідних операцій.

Логіка роботи програми охоплює два ключові аспекти:

- Авторизація: При спробі входу система перевіряє наявність відповідного облікового запису в базі даних та порівнює хеш введеного пароля із збереженим. У разі успішної перевірки відображається повідомлення про вдалий вхід, і запускається окремий процес (наприклад, файл app.py), якому передається ідентифікатор користувача.
- Реєстрація: Система перевіряє, чи не існує вже користувача з таким самим ім'ям. Якщо обліковий запис з таким ім'ям відсутній, створюється новий, а користувача інформують про успішну реєстрацію.

Програмний код app.py реалізує додаток для обслуговування клієнтів у ресторанному бізнесі, використовуючи графічний інтерфейс, розроблений на базі бібліотеки Tkinter. Програмний код включає імпорт таких бібліотек:

- tkinter: Фундамент для створення графічного інтерфейсу користувача.
- ttk: Надає розширені стильові елементи для сучасного оформлення інтерфейсу.
- PIL (Pillow): Застосовується для ефективної роботи з графічними зображеннями.

Центральним елементом архітектури є клас ServiceTrackerApp. Цей клас відповідає за створення головного вікна програми та ініціалізацію всіх вкладок. Він інтегрує функціонал відповідних розділів інтерфейсу через використання

класів MainMenuTab, MenuTab, OrdersTab, і ReservationsTab. Додаток зчитує ідентифікатор користувача з командного рядка та ініціалізує головне вікно, передаючи отриманий ідентифікатор до екземпляра класу ServiceTrackerApp.

MainMenu.py: Головна вкладка програми (MainMenuTab). Цей фрагмент коду імплементує головну вкладку клієнтського сервісу ресторанної інформаційної системи. Основні компоненти:

- create_connection з модуля config — для встановлення з'єднання з базою даних.
- tkinter — основна бібліотека для побудови графічного інтерфейсу користувача.
- messagebox — для відображення інформаційних або помилкових повідомлень у діалогових вікнах.
- os та sys — модулі для взаємодії з файловою системою та системними параметрами середовища виконання.

Клас MainMenuTab відповідає за створення головної вкладки інтерфейсу, що включає відображення інформації про поточного користувача та активні замовлення. Він також реалізує методи для завантаження як активних, так і завершених замовлень, забезпечує перевірку наявності нових замовлень та функціонал зміни користувача. Ключові методи цього класу:

- load_active_orders(): Запитує з бази даних список активних замовлень та візуалізує їх у віджеті Treeview.
- load_served_orders(): Аналогічно, завантажує замовлення, які вже були обслужені.
- check_new_orders(): Періодично перевіряє базу даних на предмет надходження нових замовлень та, за потреби, оновлює їхній список в інтерфейсі.
- serve_order(): Змінює статус обраного замовлення на "обслужене".
- pay_order(): Ініціює процес розрахунку для виконаного замовлення.

- change_user(): Дозволяє здійснити зміну користувача, перезапускаючи програму з новими обліковими даними.

Вкладка «Меню» (MenuTab). Цей програмний фрагмент відповідає за імплементацію вкладки «Меню» у клієнтському застосунку для ресторану. Ця вкладка надає користувачам можливість взаємодіяти зі стравами, виконуючи функції пошуку, перегляду детальної інформації та модифікації їхнього статусу.

Реалізація цієї вкладки включає використання наступних імпортованих модулів:

- Бібліотеки для створення інтерфейсу: tkinter та ttk забезпечують функціонал для побудови та стилізації графічного інтерфейсу.
- Бібліотеки для роботи з графікою: PIL (Pillow) інтегрована для обробки та відображення зображень страв.
- Власні класи: Використовуються для встановлення з'єднання з базою даних та ефективної обробки даних, отриманих з неї.

Клас MenuTab. Клас MenuTab відповідає за ключовий функціонал цієї вкладки:

- Реалізує функціонал пошуку страв за їхньою назвою.
- Відображає ключову інформацію про страву, включаючи зображення, назву та ціну.
- Завантажує актуальний список страв безпосередньо з бази даних.

Ключові методи класу MenuTab включають:

- _on_mousewheel(): Забезпечує функціонал прокручування вмісту вкладки за допомогою колеса миші, покращуючи навігацію.
- _on_frame_configure(): Адаптує відображення скролінгу відповідно до розміру та вмісту вкладки, оптимізуючи користувацький досвід.
- search_menu(): Імплементує логіку пошуку страв за заданими ключовими словами.
- toggle_dish_active(): Змінює статус активності страви (робить її доступною або недоступною) та оновлює її візуальне представлення в інтерфейсі.

Програмний фрагмент Orders.py призначений для реалізації процесу додавання замовлень до бази даних через спеціалізований діалоговий інтерфейс.

- Клас AddOrders: Цей клас забезпечує функціонал для додавання нових замовлень до бази даних. Кожне замовлення містить інформацію про номер столика, ідентифікатор страви та кількість порцій.
- Клас AddOrderDialog: Реалізує діалогове вікно для введення параметрів замовлення. Форма цього вікна містить поля для введення номера столика та кількості порцій. Після натискання кнопки "Додати" введені дані проходять перевірку на коректність.

Програмний модуль Reservations.py реалізує систему управління процесами бронювання столиків у ресторанному закладі. Функціонал системи забезпечується такими ключовими компонентами:

- Клас ReservationsTab: Відповідає за створення та відображення вкладки "Резервації". На цій вкладці представлено перелік усіх активних бронювань, причому кожен запис містить такі атрибути, як ID резервування, ім’я клієнта, номер телефону, дата та час бронювання.
- Клас AddReservationDialog: Імплементує діалогове вікно, призначене для створення нового бронювання. Користувач вводить необхідну інформацію про клієнта, бажані дату та час, кількість осіб та номер столика. Після підтвердження введена інформація передається до бази даних і зберігається як новий запис про бронювання.
- Метод refresh_data класу ReservationsTab: Цей метод відповідає за оновлення відображеного списку резервувань шляхом завантаження актуальних даних безпосередньо з бази даних.

3.3. Розробка інтерфейсу та тестування додатку

Як зазначено у підрозділі 3.2, для розробки графічного інтерфейсу користувача було обрано фреймворк Tkinter. Це стандартний інструмент у середовищі Python, який надає широкий набір графічних компонентів [20].

Процес створення інтерфейсу включав такі ключові етапи:

- Розробка дизайну екранних форм та діалогових вікон, базуючись на попередньо підготовлених макетах.
- Створення базових класів-контейнерів, таких як фрейми та панелі, для структурованої організації інтерфейсу.
- Додавання інтерактивних елементів управління: кнопок, текстових полів, списків, таблиць та інших компонентів.
- Застосування стилізації до інтерфейсу за допомогою класів стилів Tkinter.
- Візуальне розміщення компонентів, використовуючи менеджери компоновки grid та pack.
- Програмування обробників подій для забезпечення реакції на взаємодію з користувачем.

У результаті, при введенні правильних облікових даних під час авторизації або реєстрації користувач отримує відповідне повідомлення про успішний вхід (рис. 3.2) або реєстрацію (рис. 3.3). Вхідна форма зображена на рис. 3.1.

Вхідна форма представлена на рис. 3.1. В результаті, при введенні коректних облікових даних протягом авторизації або реєстрації, користувач отримує повідомлення про успішний вхід (рис. 3.2) або реєстрацію (рис. 3.3).

Після успішної реєстрації або входу користувача з'являється початкове вікно (рис. 3.4), яке дозволяє змінити користувача, підтвердити замовлення та здійснити розрахунок.

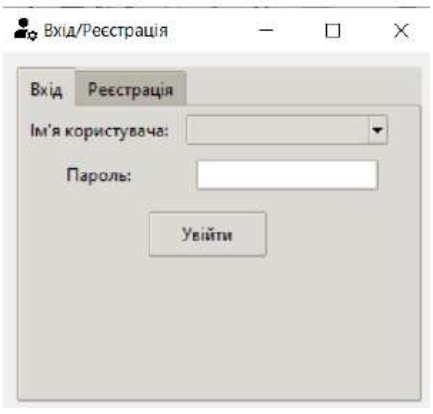


Рисунок 3.1 - Вікно входу/реєстрації

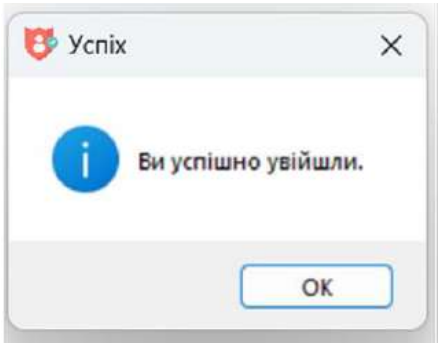


Рисунок 3.2 - Вікно успішного входу

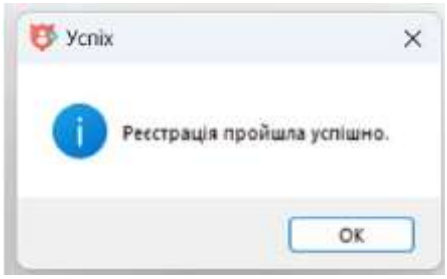


Рисунок 3.3 - Вікно успішної реєстрації у додатку



Рисунок 3.4 - Головне меню додатку для обслуговування клієнтів у ресторанах

Якщо натиснути кнопку “Змінити користувача” ми побачимо вікно підтвердження зміни користувача (рис. 3.5).

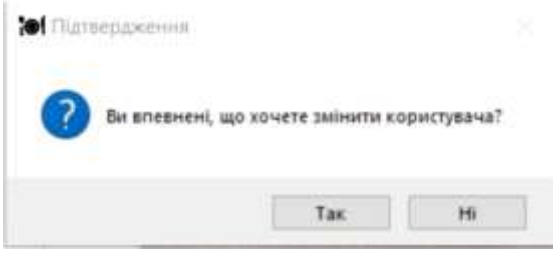


Рисунок 3.5 - Вікно підтвердження зміни користувача

Ця сторінка презентує інтерфейс, що забезпечує функціонал пошуку та додавання замовлень у межах основного меню (рис. 3.6). Після вибору певної позиції зі стравами з'являється окреме діалогове вікно, де користувач може вказати необхідну кількість порцій для замовлення (рис. 3.7).



Рисунок 3.6 – Сторінка з меню для замовлення

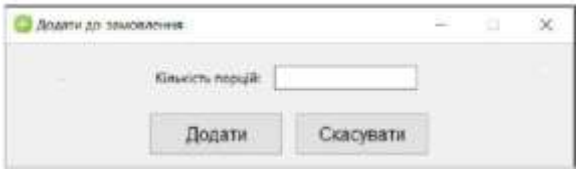


Рисунок 3.7 – Вікно для вибору кількості замовляємих порцій

Сторінка для бронювання столиків. Тут можна додати або редагувати резервування столика, обрати зарезервувати столик за фото (рис. 3.8).



Рисунок 3.8 – Сторінка резервацій

При активації кнопки «Резервація» відкривається діалогове вікно, призначене для введення персональних даних (рис. 3.9). У випадку вибору опції

«Резервація по фото» з'являється аналогічне вікно, проте воно доповнене функціоналом вибору конкретного столика (рис. 3.10).

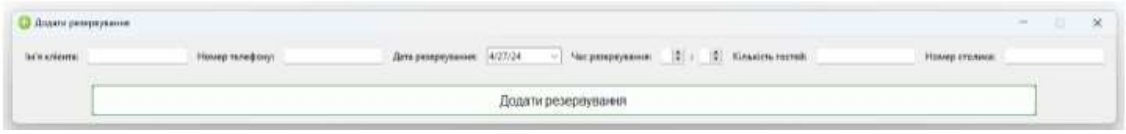


Рисунок 3.9 – Вікно резервацій столика

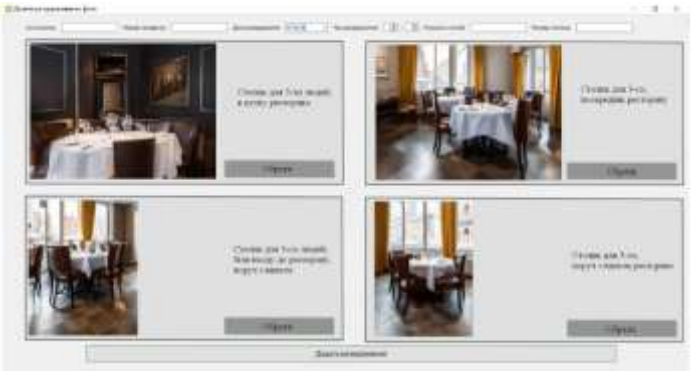


Рисунок 3.10 – Вікно резервацій столика за фото

Якщо столик був зарезервований, але нам потрібно внести певні зміни, потрібно натиснути кнопку “Редагувати”, і відкриється вікно, в якому можна буде внести потрібні зміни (рис. 3.11).

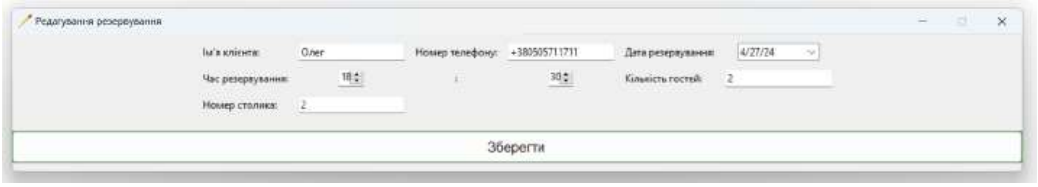


Рисунок 3.11 – Вікно редагування резервацій

У разі некоректного введення даних застосунок генерує відповідне повідомлення про помилку (рис. 3.12). Ця вкладка містить інтерфейс для пошуку, а також функціонал для зміни статусу доступності страви.

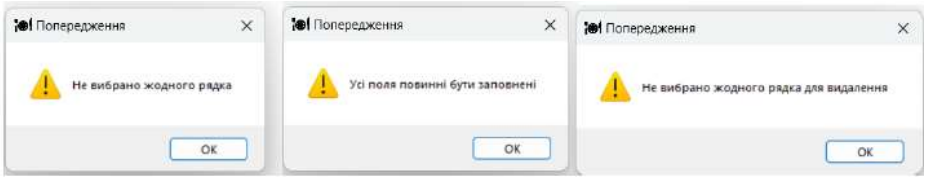


Рисунок 3.12 – Вікна можливих помилок при резервації

Зміна статусу активності страви в цій вкладці безпосередньо корелює з вкладкою замовлень: якщо страву позначено як неактивну, вона не буде відображатися під час формування нових замовлень. Отже, розроблений застосунок повною мірою реалізує ключові функції, необхідні для управління замовленнями, а саме: авторизацію, обслуговування замовлень, розрахунки та фільтрацію даних. Нижче представлено фрагмент програмного коду, що ілюструє реалізацію зазначених функцій.

```
import tkinter as tk
from tkinter import ttk, messagebox
from datetime import datetime
import hashlib
# import subprocess  # Видалено, оскільки не використовується у наданому фрагменті
from config import create_connection
import threading
 class LoginRegisterApp:
#     def __init__(self, master):
#         # ... реалізація LoginRegisterApp ...
#         pass
def hash_password(password):
    return hashlib.sha256(password.encode('utf-8')).hexdigest()
class App:
    def __init__(self, root):
        self.root.title("Обслуговування клієнтів у ресторані") # Більш описова назва
        self.root.geometry("400x300")
        try:
            icon_image = tk.PhotoImage(file='image/test.png')
            self.root.tk.call('wm', 'iconphoto', self.root, icon_image)
        except tk.TclError:
            print("Помилка: не вдалося завантажити image/test.png")
            pass # Продовжуємо без іконки
        self.style = ttk.Style()
        self.style.theme_use("clam")
        self.label_welcome = ttk.Label(
            root,
            text="Вітаємо у додатку для обслуговування клієнтів у ресторані!",
            font=("Helvetica", 16),
            background="white",
            wraplength=300
        )
        self.label_welcome.pack()
```

Висновок

У ході виконання кваліфікаційної роботи бакалавра було проведено аналіз поточного ринку програмних рішень, призначених для обслуговування клієнтів у закладах громадського харчування. В рамках цього аналізу були досліджені їхні функціональні можливості, а також визначені ключові переваги та недоліки. Проведений аналіз виявив типові проблеми, які зустрічаються при впровадженні програмних засобів для управління підприємствами харчової галузі.

Розроблено структурну схему програмного додатку, орієнтованого на обслуговування клієнтів у закладах громадського харчування. Ссхема відображає основні компоненти системи та принципи їх взаємодії. Зазначена схема демонструє низку переваг порівняно з існуючими аналогами, зокрема інтерактивний вибір місця, оновлення даних у режимі реального часу, а також кращу комунікацію з клієнтами. Завдяки реалізації цих особливостей забезпечується вищий рівень зручності та ефективності функціонування системи. Такий підхід гарантує високу гнучкість і масштабованість додатку, що сприяє подальшому розширенню його функціональних можливостей.

Розроблено алгоритм функціонування програмного додатку, який визначає послідовність дій та логіку роботи системи від моменту входу користувача до завершення процесу обслуговування. Для наочної демонстрації функціоналу були створені UML-діаграми класів, варіантів використання (use-case) та станів.

Обґрунтовано вибір мови програмування, середовища розробки, а також фреймворку та бібліотек для реалізації додатку. В якості мови програмування було обрано Python, а для розробки використано інтегроване середовище PyCharm, що визнано одним з найефективніших для роботи з Python. Для створення графічного інтерфейсу користувача було застосовано фреймворк Tkinter.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Що таке POS система та як вона працює? [Електронний ресурс] - Режим доступу: https://magefan.com/ua/blog/shtcho-take-pos-systema

2. Автоматизація ресторану [Електронний ресурс] - Режим доступу: https://www.vostok.dp.ua/ukr/infa1/Avtomatizatsiya/avtomatizatsiya_restorana/

3. Resource Management System (RMS) [Електронний ресурс] - Режим доступу: https://newline.tech/case/resource-management-system-rms-uk/

4. Toast. Built for restaurants. Built for you. [Електронний ресурс] - Режим доступу: https://pos.toasttab.com/

5. Upserve by Lightspeed is now Lightspeed Restaurant (U-Series) [Електронний ресурс] - Режим доступу: https://www.lightspeedhq.com/upserve/

6. TouchBistro All-in-One POS and Restaurant Management System [Електронний ресурс] - Режим доступу: https://www.touchbistro.com/

7. Програми лояльності для ресторанів [Електронний ресурс] - Режим доступу: https://joinposter.com/ua/post/prohramy-loyalnosti-dlya-restoraniv

8. Ресторанний маркетинг і карантин. [Електронний ресурс] - Режим доступу: https://www.restorator.ua/post/ресторанний-маркетинг-і-карантин-що-внесла-пандемія-які-рекламні-інструменти-сьогодні-актуальні

9. Як будувати UML-діаграми. [Електронний ресурс] - Режим доступу: https://dou.ua/forums/topic/40575/

10. Діаграма станів [Електронний ресурс] - Режим доступу: https://studfile.net/preview/9828818/

11. Діаграма класів [Електронний ресурс] - Режим доступу: https://uk.wikipedia.org/wiki/Діаграма_класів

12. Що таке мова програмування Python? [Електронний ресурс] - Режим доступу: https://freehost.com.ua/ukr/faq/wiki/chto-takoe-jazik-programmirovanija-python/

13. Переваги і недоліки мови Python [Електронний ресурс] - Режим доступу: https://blog.ithillel.ua/articles/perevagi-i-nedoliki-movi-python

14. Веб-застосунок [Електронний ресурс]. Режим доступу: https://uk.wikipedia.org/wiki/Вебзастосунок

15. The top 5 software architecture patterns [Електронний ресурс]. Режим доступу: https://techbeacon.com/app-dev-testing/top-5-software-architecturepatterns-how-make-right-choice

16. Types of Software Architecture Patterns [Електронний ресурс]. Режим доступу: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/

17. Топ кращих ide для python розробки у 2024 [Електронний ресурс] - Режим доступу: https://university.sigma.software/best-python-ide-and-code-editors/

18. PyMySQL documentation [Електронний ресурс] - Режим доступу: https://pymysql.readthedocs.io/en/latest/

19. Як працювати з бібліотекою Python Pillow(Pil) [Електронний ресурс] - Режим доступу: https://pythonru.com/biblioteki/osnovnye-vozmozhnosti-biblioteki-python-imaging-library-pillow-pil

20. Знайомимось з Tkinter [Електронний ресурс] - Режим доступу: https://medium.com/@gr.gromko/знайомимось-з-tkinter-c89ee588234c