НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»
УДК 004.043

До захисту допущено:
Завідувач кафедри
_______ Ігор ПАРХОМЕЙ
«___»_____2020 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інформаційне забезпечення робототехнічних систем»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Автоматизація фронт-офісу ресторанного закладу»

Виконала:
студентка ІІ курсу, групи ІК-91мп
Кравченко Єлизавета Ігорівна _____

Керівник:
доцент, д.т.н., доцент
Корнага Ярослав Ігорович _____

Консультант з нормоконтролю:
доцент, к.т.н., доц.,
Пасько Віктор Петрович _____

Рецензент:
професор кафедри ММСА, д.т.н., проф.,
Мухін Вадим Євгенович _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студентка _____

Київ – 2020 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Ігор ПАРХОМЕЙ

«___»_____2020 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Кравченко Єлизаветі Ігорівні

1. Тема дисертації «Автоматизація фронт-офісу ресторанного закладу», науковий керівник дисертації доцент, д.т.н, доцент Корнага Ярослав Ігорович, затверджені наказом по університету від « 26 » жовтня 2020р. № 3133-с

2. Термін подання студентом дисертації 14.12.2020.

3. Об’єкт дослідження – автоматизація фронт-офісу гастрономічного комплексу

4. Вихідні дані – технічне завдання на додатку для автоматизації роботи фронт-офісу ресторану.

5. Перелік завдань, які потрібно розробити – аналіз проблеми автоматизації фронт-офісу ресторанів; аналіз і реалізація моделі системи; розробка програмного забезпечення; дослідження ефективності системи.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	01.10.2020-04.10.2020	
2	Постановка задачі	05.10.2020-09.10.2020	
3	Аналіз інформаційного забезпечення	10.10.2020-11.10.2020	
4	Аналіз алгоритмічного забезпечення	12.10.2020-18.10.2020	
5	Розробка алгоритмічного забезпечення	19.10.2020-28.10.2020	
6	Розробка програмного забезпечення	29.10.2020-13.11.2020	
7	Оформлення пояснювальної записки та додатків	14.11.2020-30.11.2020	
8	Попередній захист		
9	Нормоконтроль		
10	Перевірка на співпадіння		
11	Захист		

Студент Єлизавета Кравченко

Науковий керівник Ярослав Корнага

АНОТАЦІЯ

Задача в межах даного дипломного проекту полягає в розробці програми для підвищення ефективності роботи складу ресторанних комплексів, шляхом автоматизації процесу обслуговування клієнта (та всю пов’язану з цим роботу працівників).

Вирішення поставленої задачі досягається шляхом аналізу предметної області та існуючих аналогів, розробки архітектури та математичної моделі, а в кінцевому результаті розробки програмного забезпечення для системи автоматизації фронт-офісу ресторанного комплексу.

В результаті дипломного проекту, була розроблена саме така система, яка готова до використання в сучасних закладах громадського харчування.

Ключові слова: система замовлення, автоматизація ресторану, мінімізація рутинних задач, швидке обслуговування, автоматизована робота співробітників, front-офіс.

Розмір пояснювальної записки – 114 сторінок, містить 21 ілюстрацію, 24 таблиці, 19 джерело та посилань, 8 додатків.

ABSTRACT

The task within this diploma project is to improve the efficiency of the restaurant complex's work, by automating most offront-office tasks.

The solution to this problem is achieved by analyzing the subject area and existing analogues, developing the architecture and mathematical model, and, ultimately, developing a restaurant`s front-office automatization system.

As a result of the diploma project, it was precisely such a system that was ready for use in modern catering establishments.

Keywords: ordering system, restaurant automation, minimization of routine tasks, fast service, automated work of employees, front-office.

Size explanatory note – 114 pages, contain 21 illustrations, 24 tables, 19 sources and references, 8 applications.

Пояснювальна записка
до магістерської дисертації

на тему: ***Автоматизація фронт-офісу ресторанного закладу***

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 9

ВСТУП 10

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ 12

1.1. Постановка задачі 12

1.2. Актуальність роботи 13

1.3. Проблеми неавтоматизованих закладів та їх вирішення 14

1.4. Переваги та недоліки автоматизованих ресторанних комплексів 23

1.5. Особливості бізнес-процесів 29

1.6. Аналіз існуючих програм для автоматизації фронт-офісу 38

1.7. Технічні вимоги 50

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ 53

2.1.ВИБІР АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 53

2.2. Вибір технологій додатку 54

2.3. Вибір основного сховища даних 55

2.4. Вибір архітектури серверного додатку 56

2.5. Вибір технологій для обладнання 57

2.6. Вибір математичної моделі 58

2.7. Мережевий протокол 59

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 61

3.1. Алгоритми роботи додатку 61

3.2. Структурування даних додатку 69

3.3. Опис інтерфейсу додатку 78

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ 81

4.1. Розробка стартап-проекту «RestInst» 81

4.2. Опис ідеї 81

4.3. Технологічний аудит ідеї проекту 84
4.4. Аналіз ринкових можливостей запуску стартап-проекту 85
4.5. Розроблення ринкової стратегії проекту 94
4.6. Розроблення маркетингової програми стартап-проекту 97
ВИСНОВКИ 102
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 104
ДОДАТОК А. Модель додатку 107
ДОДАТОК Б. UML діаграма акторів і їх ролей 108
ДОДАТОК В. Діаграма послідовностей – оплата готівкою 109
ДОДАТОК Г. Діаграма послідовностей – оплата картою 110
ДОДАТОК Д. Діаграма послідовностей – перегляд меню 111
ДОДАТОК Е. Діаграма послідовностей – закриття/відкриття зміни 112
ДОДАТОК Є. Діаграма класів 113
ДОДАТОК Ж. Результат перевірки роботи на співпадіння 114

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

POS – Point Of Sale.

FO – Front-офіс.

BO – Back-офіс.

АСУ – Автоматизована Система Управління.

UML – Unified Modeling Language.

UC – Use Case.

SaaS – Software as a service.

RN – React Native.

HTTP – Hypertext Transfer Protocol.

СУБД – Система управління базами даних.

SQL – Structured Query Language.

URL – Uniform Resource Locator.

API – Application Programming Interface.

GPS – Global Positioning System.

ВСТУП

У наш час про автоматизацію говорять всюди – харчова промисловість не стала винятком. З кожним днем компанії вдосконалююсь та розробляють вражаючі технології для автоматизації щоденних процесів. Це, в свою чергу, зможе знизити експлуатаційні витрати та зробити обслуговування споживачів ефективнішим, а також в цілому сприятиме підвищення продуктивності.

Автоматизація робить речі ефективнішими – якщо правильно налаштувати роботу закладу, це допоможе водночас зменшити витрати ресторану і підвищити ефективність його роботи. І хоча деякі з цих технологій мають високу ціну, нові розробки та вдосконалення старих систем роблять їх доступнішими для використання.

Чи немало ресторанних закладів і досі користуються папером і ручкою для запису і передачі замовлень, деякі з них автоматизують лише декілька процесів, інші не мають автоматизації зовсім.

Хост-менеджери ресторанів мають запам’ятовувати вільні місця, офіціанти записують замовлені страви на папері, оголошують замовлення шеф-кухарю, а потім заносять його до POS-терміналу (у випадку, якщо ця система є у ресторані). Також офіціантам потрібно час від часу підходити на кухню, аби перевірити готовність замовлення. Нерідко столи, після того, як відвідувачі залишать заклад, залишаються неприбраними, що може дуже негативно вплинути на ефективну роботу ресторану і комфорт відвідувачів закладу.

Ще одна проблема «паперового» підходу до роботи ресторанного комплексу – ведення обліку. Така «застаріла» система роботи ресторану, звісно, працює, але поступається ефективністю автоматизованим системам, а також потребує надмірного часу роботи.

У даній роботі я пропоную розглянути процес автоматизації фронт-офісу. Фронт офіс ресторанного комплексу є частиною програмно-апаратного комплексу, яка автоматизує функції обслуговування клієнтів закладу.

Переваги автоматизованих ресторанів перед іншими типами таких закладів незаперечні. Така система надає можливість гастрономічним комплексам якісно обслуговувати клієнтів, повністю виключити помилки при оформленні замовлення, оброблювати і передавати замовлення в автоматичному режимі, а, найголовніше – контролювати всі процеси, від замовлення клієнтом страв, до оперативного прибирання столу співробітниками.

Автоматизація ресторанного комплексу є невід'ємною частиною успішного ведення гастрономічного бізнесу – вона допомагає ресторанам успішно вирішувати широкий спектр різноманітних задач, не відволікаючи працівників від їх основних обов'язків та роблячи процес обслуговування клієнтів значно ефективнішим. Це, в свою чергу, збільшує дохід закладу. Ось чому ми можемо розраховувати, що все більше ресторанів продовжуватимуть рухатись на шляху автоматизації.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

В розділі розглядається актуальність обраної теми дипломної роботи – автоматизація фронт-офісу ресторанного закладу.

Розглянуться особливості бізнес-процесів гастрономічної діяльності, з якими щоденно стикаються робітники цієї сфери праці. У загальних положеннях також наведені процеси та принципи роботи, що напряму визначають ефективність роботи ресторанного закладу. Розглянута структура харчових закладів, з'ясовані цілі підприємств цієї сфери.

Приведуться до розгляду проблеми, з якими пораються працівники кожного з етапів фронт-офісу при обслуговуванні клієнтів, з'ясуються можливості вирішення та автоматизації наведених проблем.

Розглянуті класичні, не автоматизовані, методи роботи фронт-офісу гастрономічних закладів. Наведені вимоги до якості та коректності роботи працівників усіх етапів виробництва.

У розділі проведений глибокий аналіз існуючих програмних рішень, які вже зараз допомагають робітникам ресторанної сфери у автоматизації деяких робочих обов'язків.

1.1. Постановка задачі

Мета роботи: підвищення ефективності роботи ресторанного закладу, шляхом автоматизації всіх етапів роботи фронт-офісу.

Завдання: розробити систему для мобільних пристроїв та планшетів, що допоможе персоналу ресторанних закладів скоординувати їх діяльність та покращити якість обслуговування.

Для досягнення поставлених цілей необхідно вирішити такі задачі:

1. Розглянути фактори впливу на якість роботи систем автоматизації фронт-офісу ресторанних закладів.
2. З'ясувати сучасні методи автоматизації фронт-офісу ресторанних закладів.

3. Сформувати технічні вимоги до дипломного проекту.

4. Ґрунтуючись на сформованих технічних вимогах, обрати технології для реалізації системи.

5. Розробити архітектуру системи автоматизації фронт-офісу ресторану та реалізувати її технічне забезпечення.

6. Сформувати загальний алгоритм роботи системи.

7. Реалізувати технічну частину сформованого загального алгоритму роботи фронт-офісу.

Критеріями ефективності роботи системи є коректність її роботи, а саме:

1. Відсутність помилок на всіх етапах роботи фронт-офісу.
2. Безперервність роботи системи.
3. Швидкість роботи системи.

1.2. Актуальність роботи

Зміни в ресторанній індустрії за минулі 20 років вкрай значні. Даний вид бізнесу зробив величезний стрибок вперед.

Підприємства ресторанного бізнесу все швидше наближаються до міжнародного рівня, є спрямованість до постійного вдосконалення якості сервісу, виготовлення страв і т.д.

Ресторанний бізнес на сьогоднішній день, вважається однією з найбільш перспективних сфер в економіці. При цьому позитивна динаміка розвитку галузі супроводжується посиленням конкуренції на ринку.

Оскільки всі ресторани працюють при обмеженому бюджеті, кожна заощаджена гривня буде на рахунку. Один із способів автоматизації ресторанного закладу – встановлення POS-системи.

POS-система (POS – Point of Sale) – це програмно-апаратний комплекс, що працює на базі фіскального реєстратора. За POS-системою закріплений типовий набір касових функцій: облік та відпуск товару, прийом та видача грошей, анулювання фактів покупок, тощо. Зазвичай складні приватні POS-

системи називають POS-комп'ютер, POS-монітор, чековий принтер, фіскальний реєстратор, POS-клавіатуру, дисплей покупця, грошовий ящик, зчитувач магнітних карт.

POS-система може надати ресторану новий рівень контролю за діяльністю гастрономічного комплексу, допомагаючи підвищити ефективність, збільшити прибуток та точно налаштувати управління усіма процесами фронт-офісу та бек-офісу, від замовлення клієнтом страв до контролю та обліку інгредієнтів, допомоги в інвентаризації продуктового складу. [1]

Перехід від традиційного касового апарату та застарілого «паперового» підходу (стосується етапу фронт-офісу – запис офіціантом замовлення клієнта) до комп’ютерної системи ресторану може бути важким, однак рентабельність інвестицій дійсно може коштувати часу та зусиль. POS-сервіс є обов’язковим, ресторан хоче максимально ефективно використовувати свої ресурси та працювати із мінімальними збитками.

1.3. Проблеми неавтоматизованих закладів та їх вирішення

Ресторанні заклади нерідко мають низку проблем та недоліків у бізнес-процесах, однак чи немало з них може вирішити процес автоматизації. Нижче буде наведений перелік деяких з проблем ресторану, а також запропоновані їх можливі вирішення з використанням технологій.

1.3.1. Проблема реєстрації, бронювання столика та підготовка замовлення

У ресторанній галузі час – це гроші. Кількість часу, що потрібно на обслуговування клієнтів, визначається кількість клієнтів за столиками, що обслуговуються щодня (денна завантаженість закладу). А це, в свою чергу, визначатиме дохід від продаж та величину чайових для персоналу.

Для того, щоб скоротити час, існує кілька функцій, які позбавлять необхідності працівникам хостес вручну реєструвати клієнтів у списки бронювання біля парадного входу, а також постійно перевіряти готовність замовлень, приходячи по декілька разів на кухню. Це, в свою чергу, змушує

офіціанта проводити зайвий час у залі ресторану (бігаючи з одного робочого пункту до іншого), а проводячи більше часу у залі ресторану, офіціанта частіше підкликають клієнти.

Такі моменти збільшують час обслуговування клієнтів, зменшують ефективність роботи працівників та надають додаткове навантаження на персонал, якого можна уникнути.

Для того, щоб це вирішити, було визначено декілька особливостей для додатку. Це включає карту посадочних місць (столиків), чергу виконання замовлення та оновлення статусів замовлення від шеф-кухаря.

1. Карта посадочних місць.

   Коли клієнти заходять, їм потрібно звернутись до планшету, на якому їм пропонують вказати своє ім’я та кількість відвідувачів за одним столиком. Потім ця інформація надсилається назад до нашої бази даних, яка використовує алгоритм, який призначає посадочне місце кожному клієнту. Цей алгоритм спочатку враховує кількість гостей, аби визначити пріоритет клієнтів для кожного столицу. Він робить це, перевіряючи наявність столів, які можуть містити вказаний розмір компанії. Якщо підходящих вільних посадочних місць немає, то наступна за чисельністю компанія клієнтів має пріоритет і так далі.

   Коли частина алгоритму пріоритезування розміру компанії закінчиться, наступна частина складається з пріоритетів за часом. Тут алгоритм перевіряє доступні столики, і якщо кілька розмірів компанії відповідають специфікаціям доступного столика, то стороні, яка прибула першою, надається пріоритет сидіти за столом. Потім алгоритм продовжує робити це для решти клієнтів, які чекають на вільне місце, коли таблиці стануть доступними чи недоступними.

В кінці нашого алгоритму на планшеті в передній частині ресторану відображається ім’я клієнта та схема ресторану з виділеним столом, якому вони призначені. Клієнту пропонується підтвердити, що він отримав повідомлення, згоден з ним і перейде до столу. Ця інформація також оновлюється для офіціантів, щоб вони знали, чи нещодавно був зайнятий один із їхніх столів, або клієнт протягом певного часу не отримав підтвердження, аби підійти до входу ресторану та шукати клієнта особисто.

Цей алгоритм також повинен бути зроблений таким чином, щоб він міг ефективно розподіляти посадочні місця для клієнтів на території всього ресторанного комплексу. Наприклад, середній за розміром ресторан буде поділений на чотири умовні зони, за якими будуть слідкувати чотири відповідні офіціанти, де кожен відповідний за свою зону.

Таким чином, першому клієнтові (який прийшов раніше) назначається до посадки столик у першій умовній посадочній зоні, а для наступного – у другій зоні. Таким чином алгоритм дає можливість найбільш ефективно (згідно завантаженості офіціантів) розподілити клієнтів по ресторанному комплексу, максимально забезпечивши офіціантів і прискоривши всі процеси на піку відвідування клієнтами ресторанного комплексу.

Коли клієнт/клієнти закінчать вечерю, їм буде доступний планшет для оплати свого замовлення. Цей процес буде сигналом для персоналу, що цей столик потрібно прибрати. Це повідомлення автоматично висвітиться у додатку для персоналу, щоб офіціанти знали, які столи потрібно прибрати для наступних клієнтів. Коли стіл прибраний, офіціант повинен буде вручну поставити столику статус «прибрано», простим натисканням кнопки, повідомляючи систему, що стіл готовий обслуговувати нових клієнтів.

2. Порядок виконання замовлень.

   Порядок виконання замовлення стане для офіціантів та шеф-кухарів способом синхронізованої роботи у вигляді електронного списку зі статусами замовлення (виконується, готове для подачі, тощо).

   Він буде складатися з бази даних, яка зберігає страви. Коли офіціант оформить замовлення для клієнтів, усі страви будуть розміщені в кінці черги в базі даних. І шеф-кухар, і офіціант зможуть бачити порядок виконання. Шеф-кухар зможе натиснути на їжу та мати можливість вибрати один із трьох статусів: зупинений, триває чи завершений. Порядок замовлень також покаже, скільки часу минуло з початку приготування страви, що замовили клієнти, так що офіціант зможе оцінити час, коли вона буде завершена. Коли шеф-кухар встановить статус для страви як завершений, система надішле сповіщення офіціанту, який розмістив замовлення страви, щоб вони знали, що він готовий до подачі замовникам.

   Цей порядок виконання замовлень також враховує замовлення, які були оформлені для самовивозу. У такому випадку додаток повідомить офіціантів, що такі замовлення необхідно подати до столику обслуговування клієнтів, які прийшли їх забрати. Пріоритет приготування їжі оцінюється за порядком їх замовлення, незалежно від того, була їжа замовлена безпосередньо у ресторані для трапези на місці, чи для куштування за межами ресторанного закладу.

1.3.2. Марно витрачений час

Багато відвідувачів ресторану приходять до закладу з порожнім шлунком, готові сісти і поглинути все, що трапиться їм у меню. Зайнявши свій столик, клієнти починають засікати час, витрачений на їх обслуговування (поки

офіціант принесе меню, приготує стіл до трапези, поки клієнти визначяться із замовленням і так далі, засікається час, за який страву приготують та принесуть до столу).

Ці всі етапи можна автоматизувати, зменшивши час очікування на усіх етапах, необхідних для того, аби замовлення дісталось клієнтам до столиків.

Проблема, яка порушується, – це час: офіціантам потрібен час, щоб прийняти замовлення, передати отримане замовлення кухарам і потім подати замовлені страви. Через стільки марно витраченого часу клієнти сидітимуть в очікуванні довше, а гостям, що прибувають, доведеться довше чекати, щоб сісти на бажане місце.

Щоб максимізувати прибуток та зберегти клієнтів, які бажають повернутися, було вигадані рішення, які прискорять находження клієнтів у ресторані.

1. Функціональне меню для замовлення страв.

   У таких багатьох ресторанних комплексах етап замовлення і подачі страв стандартний. Хостес чи офіціанти розсаджують гостей і подають кожному гостю меню. Після того, як клієнти визначяться зі стравами, офіціанти приймають їх замовлення і йдуть до кухні, щоб передати це замовлення кухарам. Після того, як страви будуть приготовані, офіціанти приходять у зону видачі і шукають потрібне їм замовлення, щоб принести його до відповідного клієнта.

   Хоча це надзвичайно трудомістко, додаток для автоматизації ресторанів може значно пришвидшити весь процес і максимізувати задоволеність клієнтів, а також збільшити прибуток ресторану.

   У розроблюваному додатку для автоматизації ресторанів у нас є функціональне меню, яким будуть користуватись офіціанти, і ми зможемо розміщувати замовлення, які надсилатимуться безпосередньо кухарам без необхідності йти на кухню, щоб повідомити їм замовлення клієнтів. Наявність цього

функціонального меню значно пришвидшить час, необхідний для надходження замовлення від клієнта до офіціанта і далі до шеф-кухаря. Щойно кухарі виконають замовлення, офіціанти отримають сповіщення через наш додаток для автоматизації ресторанів. Отримавши повідомлення від шеф-кухаря про те, що замовлення виконані, офіціанти не будуть витрачати час на перевірку замовлень і можуть виділити більше часу, спрямованого на замовлення клієнтів; прагнення до задоволення потреб клієнта замість постійної перевірки, чи замовлення виконане, підвищить задоволеність клієнтів. Офіціанти будуть тримати планшети для прийому замовлень, щоб вони завжди могли висловити свою особисту думку чи рекомендацію відносно тих чи інших страв ресторанного закладу, клієнтам це зазвичай подобається. Також клієнти, які мають цей додаток на власному смартфоні, можуть зробити замовлення самостійно, це відобразиться у системі (що цей клієнт сам зробив замовлення), щоб офіціант не мусив переходити до них. Загалом, прискорення процесу замовлення та отримання їжі споживачами істотно скоротить час, який споживачі займають за столом; таким чином, дозволяючи посадити більше гостей і максимізуючи прибуток.

1.3.3. Проблема очікування розрахунку

Уявіть, що ви і група друзів є клієнтами ресторану, і ви всі готові сплатити рахунок. Ви кличете офіціанта, але він обслуговує інші столи свої робочої зони. Минає п’ять хвилин. Нарешті приходить офіціант, і ви говорите йому, що готові платити. Проходить ще п’ять хвилин, коли офіціант повинен розрахувати загальну вартість вашого харчування та видати замовникам чек. Він передає вам чек, але ви хотіли, щоб рахунок розділився між вами та рештою вашої групи. Проходить ще п’ять хвилин, коли офіціант повинен перерахувати рахунок і розділити його згідно вашим побажанням, на вас всіх. Нарешті, ви вирішили

заплатити кредитною карткою, і пройде ще декілька хвилин, доки йому доведеться бігти до машини для проведення оплаті та обробляти рахунок.

Загальний процес оплати рахунку зайняв двадцять хвилин. За ці двадцять хвилин інша група клієнтів могла сісти, зробити замовлення і, можливо, почати куштувати замовлені страви. Незалежно від того, чи розділяється оплата рахунку на декілька людей, чи вам неприємна така затримка в обслуговуванні від офіціанту, оплата рахунку завжди була довгим та виснажливим процесом у ресторанах.

За допомогою додатку для автоматизації ресторанних комплексів ми можемо значно пришвидшити процес оплати рахунку.

1. Розширена функція оплати.

Після того, як офіціант зробив замовлення, воно автоматично записується до рахунку відповідного столику. Крім того, програма буде відстежувати, які замовлення відповідають якому гостю на кожному з столиків. Маючи замовлення, яке зберігається в нашій базі даних, ми виключимо використання олівця та паперу та необхідність прорахунку загальної суми рахунку (оскільки наша заявка буде проводити всю арифметику з рахунку).

Якщо офіціанта просять розділити рахунок, наша програма буде задавати офіціанту питання "на скількох людей ви хотіли б розділити рахунок?".

Додаток повинен мати можливість особисто прорахувати вартість замовлення кожного з клієнтів за столиком, якщо вони цього бажають. Усі ці функції в інтерфейсі оплати рахунків зекономлять час, необхідний офіціантам для підсумку рахунку, і значно скоротять час, необхідний клієнтам для оплати свого замовлення.

З меншим часом, необхідним для обробки транзакцій, більше часу можна виділити для відпочинку та надання допомоги новим гостям ресторанного закладу. Ця функціє допоможе не лише

максимізувати прибуток, але й пришвидшити процес оплати, а це означає, що клієнти залишатимуть ресторан із посмішкою та бажанням повернутися.

1.3.4. Проблема ефективного управління звітами

У наш час знайти цілісну систему, яка б займалася усіма процесами фронт та бек-офісу, доволі важко. Менеджери хочуть бачити свої прибутки та збитки щоразу, коли їм це потрібно, щомісяця, щоб переконатися, що все йде добре.

Реалізування обчислюваної та аналітичної сторони системи для програми являється важливим для Менеджера, оскільки він легко зможе відстежити прогрес у тому, що відбувається, з точки зору фінансів, у його ресторанному закладі, де б він не знаходився.

У менеджерів є багато рутинних та повторюваних задач, на які вони повинні звертати увагу, тому розміщення всіх фінансових питань, пов’язаних безпосередньо з чеками клієнтів, нарахуванням заробітної плати та витратами на зберігання на місці, дозволить менеджеру відстежувати динаміку бюджету і контролювати його використання, оскільки йому щодня необхідно вживати відповідні заходи для корекції бюджетних коштів або вносити зміни.

Ця функція також дозволить менеджерам бачити різні норми прибутку для кожного окремого ресторанного комплексу, щоб мати змогу зрозуміти загальне уявлення про ефективність кожного закладу. Ця частина додатку також зможе обробляти чайові кошти окремо від прибутку та проставляти їх належність для кожного офіціанта, щоб менеджер міг точно знати, скільки чайових надходить кожному співробітнику.

1.3.5. Проблема редагування меню

Цей функціонал додатку є дуже корисним. Він буде дозволяти менеджеру просто додавати та видаляти продукти з меню, налаштовувати його так, як йому потрібно. Менеджерам не потрібно буде витрачати додаткові кошти на купівлю нових меню (кожен раз, коли воно змінюється чи оновлюється), або оновлювати

меню на виніс для клієнтів, оскільки все буде доступно онлайн та готово до роботи.

Менеджер має мати можливість додати страву/напій або видалити його з легкістю, а також відстежувати наявність необхідних страв на кухні та відстежувати, скільки замовлень містить ту чи іншу позицію меню.

Можливість отримати доступ до нього з будь-якого місця, включаючи його власний будинок чи машину, дозволяє менеджеру набагато легше виконувати рутинні обов’язки, за які він відповідає.

Такі як ревізії чи аналітика стану інших ресторанів тощо. Це також дозволяє майбутнім менеджерам зв’язуватися з різними ресторанними закладами мережі та бачити відмінності форматів меню.

1.3.6. Портал для співробітників

Менеджери мають деякі складнощі з моніторингом графіку роботи співробітників. Працівники приходять на роботу, та лишають її у конкретні часи, переглядають свої графіки роботи (зміни), бачать, коли і де вони мають працювати, скільки вихідних днів вони мають, скільки чайових заробили і у які дні. Співробітникам може знадобитись доступ до цієї інформації у будь-який час, незалежно від того, де знаходиться менеджер. Таким чином, нам потрібен портал для співробітників.

Портал для працівників працює як універсальний менеджер планування та інструмент для співробітників ресторану. До нього можна буде увійти двома способами. Співробітники можуть увійти до програми на планшеті та переглянути календар, який відображає всі доступні зміни, а також ті, до яких вони причетні. Подібний інтерфейс буде доступний у додатку.

Ця функціональність надає кілька варіантів для персоналу. Це дозволяє їм записатися на майбутні зміни, відкрити раніше взяту зміну для зміни графіку та пропустити чергування із зазначеною причиною. Цей варіант використовувати ефективніше, на відміну від ручного прийняття до відома всієї

такої інформації; це автоматизує документацію та управління змінами у системі з мінімальним вкладом від працівників.

Крім того, існує можливість розширення, завдяки чому співробітники можуть відслідковувати свої робочі години (скільки триває їх зміна). Мета цієї функції – допомогти в автоматизації розподілу заробітної плати за тиждень, оскільки вона зберігає деталі, що стосуються цих розрахунків (пропрацьованих годин), тобто відпрацьовані години, лікарняні та ін. Ця інформація може використовуватися менеджерами для мінімізації кількості часу, витраченого на відповідні завдання за цикл оплати праці.

Співробітники також знайдуть цей спосіб відстеження їх робочого графіку комфортним. Це допомагає їм переконатися, що всі години роботи ввійшли правильно, а також побачити свій чіткий графік роботи, а також інших співробітників.

Портал з календарем та планер працює, щоб вести організований журнал робочого часу, на відміну від відстеження змінності вручну. Мета автоматизації процесів приходу/виходу на роботу, контроль змін та інших функцій, описаних вище, зменшити рутинні завдання менеджера та автоматизувати роботу, яка є необхідною.

1.4. Переваги та недоліки автоматизованих ресторанних комплексів

Ресторан, як і будь-який вид бізнесу повинен приносити прибуток. Для цього вся робота закладу повинна бути грамотною і чітко організованою. Автоматизація більшості процесів може істотно полегшити роботу як співробітників ресторану, так і самого підприємства громадського харчування.

Для чого потрібні автоматичні системи управління бізнесом? Крім руху грошових коштів, в ресторані потрібно постійно і ретельно враховувати переміщення продуктів харчування, замовлення і потік клієнтів і інші нюанси. Контролювати всі поточні процеси одній людині дуже складно, тому використовують різні системи автоматизації.

Однак, потрібно розуміти, що лише автоматизувавши ресторанний бізнес, миттєвого прибутку не одержати, система – це тільки зручний інструмент в руках грамотного керівника.

За підсумком її використання, автоматизація повинна мінімізувати витрати ресторану, зменшити ручну працю і підвищити прибутковість закладу.

Що стосується перших двох пунктів – економії та продуктивності праці, то це кількісні аспекти. В даному випадку показники ефективності системи можна виміряти, порівняти і оцінити. Ще одна важлива функція комплексу автоматизації з розділу «економія», це контроль над персоналом. Як відомо, це хворе місце підприємств громадського харчування.

Досвід впровадження систем автоматизації на підприємствах харчування показує, що контроль дозволяє заощадити до 20% виручки за рахунок виключення крадіжок та інших зловживань. Очевидно, що навіть для маленького закладу настільки актуальна в умовах кризи економія буде куди більше не при відмові від автоматизації, а, навпаки – при її застосуванні.

Зрозуміло, що слід мати на увазі і період окупності системи, але самі ресторатори, в тому числі власники маленьких закладів кажуть, що автоматизація потрібними вкладеннями. Більш того, якщо мова йде про відкриття нового закладу, відкладати на потім вкрай не рекомендується. Автоматизація сильно впливає на стиль роботи і психологію персоналу, яку потім складно буде перекувати з «папірці» на POS-термінал.

Мова тут йде не про навчання, яке складнощів не викликає, а саме про психологічні аспекти. Багато компаній зі сфери громадського харчування вважають, що автоматизація це серйозні витрати, але автоматизація це не просто якась програма з обладнанням.

Автоматизація це конкретне рішення проблем будь-якої компанії. На питання: скільки коштує автоматизація можна задати зустрічні питання: а скільки коштують помилки при прийманні товарів на склад і інвентаризації?

Скільки коштують «трюки недобросовісного персоналу? Скільки коштують незадоволені якістю і часом обслуговування клієнти?

Зрозуміло, зараз складний час для невеликих ресторанів і кафе. І якщо раніше керівник міг допустити, що від 7 до 15% виручки потрапляє в кишені офіціантів і касирів, минаючи касу, то зараз кожен клієнт і кожна копійка на рахунку. А боротьба з крадіжками неможлива без автоматизованого обліку. Автоматизація ж вирішує всі перераховані проблеми.

Автоматизація це не витрати, це економія і перемога над постійно зростаючими витратами. Важливо, що з введенням системи автоматизації з'являється можливість впроваджувати різні дисконтні програми та кредитування для постійних клієнтів і місцевих жителів, вважати знижки і бонуси буде комп'ютер на підставі запропонованих варіантів, а офіціант не буде забивати голову складними розрахунками.

В середньому термін окупності системи автоматизації для невеликого ресторану становить 3-4 місяці при мінімальному наборі модулів і компонентів.

1.4.1. Критерії вибору інформаційної системи і устаткування

Для того, аби краще зрозуміти переваги та недоліки різних автоматизованих систем для ресторанних закладів, необхідно спочатку з’ясувати по яким критеріям ресторани обирають для свого бізнесу засоби автоматизації (це зможе вказати на проблеми, з якими зможе чи не зможе впоратись система).

Отож, для невеликих ресторанів і кафе, тим більше в нинішній час, дуже важливим критерієм при виборі системи автоматизації є ціна. На нашому ринку існує маса технологічних функціональних рішень для автоматизації невеликого громадського харчування.

Найголовніше – це автоматизувати весь цикл роботи від приймання інгредієнтів, приготування страв до безпосередньої реалізації. Дані бізнес-процеси є спільними для будь-яких ресторанів і кафе. Важлива аналітика для керівника.

Що стосується систем лояльності це важливий момент, але для неавтоматизованих підприємств треба починати з настройки загального обліку, а після цього задуматися над маркетингом, оцінками роботи персоналу і т.д.

Для невеликого ресторану найбільш важливим критерієм при виборі обладнання для автоматизації є його надійність, так як в великих ресторанах, де використовується кілька точок введення замовлень, завжди можна обійтися меншою кількістю терміналів, щоб не закривати ресторан на час ремонту обладнань.

За функціональністю програмне забезпечення для невеликих ресторанів мало чим відрізняється від програмного забезпечення для великих ресторанів і клубів. І там, і там використовуються основні модулі введення і розрахунку замовлень, модулі побудови звітів, програм складського обліку і розрахунку калькуляційних карт. [2]

Функціонал основних модулів може розширюватися додатковими модулями, такими як дисконтні системи, системи тарифікації більярду та караоке, авторизації кредитних карт і відеоспостереження. При виборі системи автоматизації рекомендуємо звернути увагу на масштабується програмне забезпечення для невеликих ресторанів, тобто починати можна з основних програм введення замовлення і розрахунку, поступово розширюючи функціонал калькуляційними програмами, дисконтними та бонусними системами, модулями доставки і відеоспостереження за касовими операціями.

Тенденція зниження загальної кількості відвідувачів в ресторанах змушує їх власників набагато уважніше ставитися до задачі утримання старих і залучення нових гостей.

Ресторатори винаходять цікаві антикризові пропозиції, вводять бонуси для постійних гостей, але одним з головних конкурентних переваг як і раніше є професійний і вмотивований персонал. Всі антикризові заходи в частині управління бізнесом, а також найкращі ідеї по залученню гостей будуть марні,

якщо співробітники ресторану не зацікавлені в успіху особисто і відносяться до роботи формально.

Сучасна інформаційна система дозволяє забезпечити гостинність в залі і безперебійну роботу на кухні, автоматизуючи не тільки касу і склад, а й мотивацію персоналу, оптимізацію фінансів, програми лояльності і навіть управління музикою, яка звучить в ресторані. За результатами досліджень, при впровадженні такої системи шляхом оптимізації бізнес-процесів і скорочення витрат можна заощадити до 25% операційних витрат.

Проект окупається дуже швидко за рахунок підвищення ефективності бізнесу, зниження втрат від зловживань, оптимізації продуктових запасів і закупівельних цін, продуманої фінансової політики, а також збільшення прибутку за рахунок зростання кількості гостей.

Зараз ринок систем автоматизації для ресторанів досить насичений. Є і так звані економічні і навіть зовсім безкоштовні рішення (open source). Можливість заощадити на автоматизації для малого підприємця все-таки є, і закладена вона не в функціоналі системи, а в гнучкості конфігурацій.

Наприклад, невеликий ресторан або кафе може обійтися всього однією інсталяційною ліцензією, яка дозволяє вести роботу фронт і бек-офісу. При цьому програмне забезпечення може бути встановлено відразу на декількох комп'ютерах, правда, з обмеженням можливості одночасної роботи з сервером.

Кілька скоротити витрати невеликим закладам можна і на брендах обладнання, яких зараз досить багато. Однак тут слід пам'ятати про різні умови гарантійного обслуговування та технічної підтримки, яка, безсумнівно, потрібна всім, а маленьким закладам особливо, оскільки вони не можуть собі дозволити утримувати штатного системного адміністратора.

Якщо говорити про програмне забезпечення, то все частіше власники підприємств громадського харчування, вибираючи систему автоматизації, починають розуміти, що кожна додаткова функція в програмному забезпеченні

додає вартості до загальної по всьому рішенням. При цьому не потрібно забувати, що технічна підтримка необхідна підприємству будь-якого формату.

Але, однозначно, чим простіше програмне рішення, тим рідше ви будете звертатися в службу підтримки. Крім того, хороше початкове навчання роботи з програмним продуктом також є запорукою мінімального числа звернень до служби технічної підтримки».

Підводячи підсумки, оформимо переваги та недоліки автоматизації ресторанних комплексів.

Переваги автоматизації:

- Контроль всіх процесів, що відбуваються в ресторані.
- Скорочується час очікування гостем свого замовлення.
- Збільшується швидкість, з якою готується і подаються страви.
- У офіціантів є можливість контролювати замовлення на обслуговуваних ними столах.
- Зменшення ймовірності помилкових дій персоналу.
- Збільшення доходів компанії.
- Виключення крадіжок і недобросовісного використання ресурсів підприємства.
- Підвищення якості і швидкості обслуговування відвідувачів.
- Відбувається автоматичне формування багатьох фінансових звітів.
- Збирається інформація маркетингового характеру.
- Високоефективне управління, а також здійснення аналізу діяльності підприємства.
- Своєчасний облік товарів, а також швидке проведення інвентаризації.
- Найефективніше відбувається контроль (в тому числі і дистанційно) за роботою персоналу та закладу.

Недоліки автоматизації наведені на наступній сторінці.

- При несвоєчасному занесенні даних, неточності інформації і помилок при роботі з системою, вона спотворює підсумкову інформацію.
- Деякі програмні забезпечення працюють тільки при наявності підключення до інтернету, відповідно, при неполадках мережі вони перестають коректно функціонувати.
- Є досить складні в управлінні програми, з великою кількістю незатребуваних функцій. Вони можуть тільки ускладнити процес роботи ресторану.
- При неполадках системи налагодити її роботу зможе лише фахівець.

У висновку хотілося б ще раз відзначити, що система автоматизації незамінна в справі підвищення якості обслуговування і продуктивності праці, а також зниження витрат і зловживань персоналу.

1.5. Особливості бізнес-процесів

Бізнес-процеси – це технологічні, ділові та адміністративні процедури і етапи функціонування підприємства, які включають в себе діловодство, управління персоналом, фінансовими і матеріальними потоками, господарськими і технологічними ділянками. В результаті бізнес-процесів проводиться певна група продуктів (послуг) [3].

Зараз розглянемо персонал гастрономічного закладу, учасників усіх процесів (враховуючи клієнта) та задачі, які вони щоденно виконують.

З’ясуємо, як розроблюваний додаток зможе автоматизувати роботу ресторану, а також опишемо основні бізнес-процеси, які відбуваються у ресторанних комплексах.

Опис бізнес-процесів для персоналу буде стосуватись наступних учасників роботи ресторанного закладу: шеф-кухар; хостес-менеджер; клієнт; офіціант; адміністратор; ранер.

1.5.1. Персонал та бізнес-процеси

1. Шеф-кухар.

   Робота на кухні може бути напруженою і буває важко відстежувати усі замовлення, що потрібно приготувати. Замовлення потрібно заповнювати протягом чіткого періоду часу, одночасно забезпечуючи подачу декількох страв на різні столики. Це потрібно робити з великою увагою та точністю, оскільки будь-яка помилка може бути катастрофою – замовлення може бути приготовано неправильно або, навіть гірше, людина з харчовою алергією може постраждати внаслідок поганого почерку офіціанта. Кухарам потрібно отримувати їх замовлення з високою точністю та швидкістю, щоб переконатися, що клієнт буде задоволений сервісом, аби повернутись до закладу знову. Коли кухари закінчують приготування страв та віддають їх на подачу, а офіціанти не забирають страви із столу видачі протягом декількох хвилин, це може дуже нашкодити, оскільки їжа охолоджується, і вона займає місце на столі видачі, яке можна було б використати для видачі інших блюд.

   Було б дуже зручно, якби кухари мали додаток, що чітко відображає всі вхідні замовлення. Це може заощадити їм значно більше часу на кухні та зробити процес приготування їжі більш ефективним. Нарешті, було б чудово подавати сигнал до офіціантів, що страва приготована і готова до того, щоб її подали до столу, щоб скоротити час, коли їжа простоює на столі видачі.

   Додаток створить чергу замовлень для шеф-кухаря, щоб він знав, які страви їм слід готувати, та надав кухарам загальне уявлення про те, як давно було зроблено замовлення, аби замовлення, які надійшли до приготування раніше, виконувались першими. Додаток також дозволить кухарам викликати офіціантів, коли буде

виконано певне замовлення, щоб вони знали, коли прийти і забрати страви, щоб їх могли подати до столиків клієнтам ресторанного закладу.

2. Хостес-менеджер.

   Щоразу, коли хостес вітає клієнтів у ресторані, вони прагнуть обслугувати клієнтів настільки ефективно та бездоганно, наскільки це можливо. Прибуваючі групи із декількох клієнтів, як правило, мають різні запити та вимоги залежно від кількості людей та місця, де вони хочуть сидіти. Іноді у вас можуть попросити влаштувати вечірку на вісім осіб, яким потрібно влаштуватись за круглим столом у кутку, в інші рази ви можете обслуговувати сім'ю з чотирьох осіб, яка бажає сидіти на веранді ресторанного закладу, під навісом. Хостес-менеджери знаходять для них місце, яке задовольняє усі їх умови, але іноді місця просто немає, і хостес насправді не зможуть передбачити скільки часу пройде перед тим, як таке бажане місце звільниться від попередніх клієнтів.

   Пошук доступного столу для розміщення гостей іноді може бути складним завданням, особливо коли менеджери не можуть передбачити, які столи будуть готові до використання. Коли вони приймають клієнтів, було б чудово, якби існував спосіб легко відстежувати, які столи в даний час чисті та порожні, а які зайняті, без необхідності документування усіх нюансів на папері та запам'ятовуванні вільних чи зайнятих місць у ресторанному закладі. Водночас було б непогано відстежувати, скільки часу кожен столик вже зайнятий. Це буде великою підмогою для надання гостям точних оцінок того, коли може стати доступним дійсне розташування місць.

   За допомогою зручного додатку, хостес-менеджери можуть буду освідомленими які столики зайняті, вільні, чисті, потребують

прибирання чи сервірування, тощо, та керувати їх статусами. Кожного разу, коли хостес саджають нових гостей, вони можуть позначити стіл як зайнятий. Коли гості підуть, цей стіл можна позначити як вільний, повідомивши ранера про те, що стіл готовий до прибирання. Після того, як ранер закінчить прибирання щойно звільненого столика, хостес можуть позначити його як очищений, що дає їм зрозуміти, що стіл готовий до наступного використання та посадки нових клієнтів.

3. Клієнт.

   Клієнти можуть любити певний ресторанний заклад, але нерідкі випадки, коли вони можуть його залишити, через надто довге та повільне обслуговування. Коли клієнти сидять та чекають, було б дуже гарно, якби вони могли замовити напої чи закуски, не чекаючи, поки офіціант дійде до них із меню, особливо якщо ресторан неймовірно завантажений.

   Це справді може скоротити час, витрачений на очікування. Якщо у клієнтів буде особливе доповнення до вже зробленого та відпущеного замовлення у офіціанта, або клієнту потрібно повідомити офіціанта про що-небудь, це може бути грубим приверненням уваги, на яке персонал не завжди в змозі моментально відреагувати. Очікування офіціанта може стати основною незручністю під час обіду. Зазвичай дуже важко привернути їх увагу, якщо вони не проходять повз клієнтів, або якщо ніхто з них не звертає увагу на те, що клієнтам потрібна допомога (не через неввічливість чи ігнорування, а через велику завантаженість та перебування офіціанту сконцентрованого на поточному замовленні чи задачі), так, клієнти можуть чекати тривалі періоди часу, перш ніж хтось помітить їх.

Також може бути важко розділити рахунок на декількох осіб, перебуваючи у закладі з друзями, часто про це згадують лише в той момент, коли компанія збереться залишити заклад та попросить розрахувати столик. Такі випадки стають незручним досвідом – визначитись який найкращий спосіб розділити рахунок, т повідомити його офіціантові, з урахуванням можливостей касового апарату ресторанного закладу. Хтось вважає за найкраще розділити його за допомогою деталізації (кожен клієнт платить лише за своє замовлення), а хтось – просто порівну розділивши загальний рахунок в кінці, між усіх людей у компанії. Коли це не було належним чином сплановано, може бути важко зрозуміти, як це найкраще реалізувати.

Додаток у клієнта з меню, за допомогою якого компанія відвідувачів могла б робити замовлення відразу після того, як відбулась посадка, була б чудовою ідеєю. Це також був би чудовий спосіб – мати більш інтерактивні меню, які могли б дати клієнтам більше інформації про кожну страву, або підкликати офіціанта, оскільки повторно кликати офіціанта до свого столику, привертаючи його увагу жестами, може зайняти забагато часу. Можливість просто натиснути кнопку в програмі, щоб викликати офіціанта до столику – просте рішення цієї проблеми. Також чудова можливість для відвідувачів – замовлення страв прямо із додатку (тобто меню, вбудоване в нього).

4. Офіціант.

   Робота офіціантом – надзвичайно складна робота. Офіціанти повинні бігати по ресторану, приймаючи замовлення клієнтів, відстежуючи, який стіл і що замовляв, подавати відповідне замовлення. Вони також повинні взаємодіяти з кухарями, шефом та негайно доставляти їжу людям, допоки вона не встигла хоч трохи

охолонути. Вони також повинні вводити чеки, коли клієнт чи група готова платити. Оскільки в ресторанних закладах може бути багато столиків, дещо важко визначити певний порядок роботи відповідно до певного столику, а також негайно реагувати та визначати, чи потрібно стіл прибирати та підготовлювати, сервірувати його для наступного клієнта. На жаль, іноді офіціанту може знадобитися відправити блюдо назад, якщо клієнту воно не подобається або з ним виявляє щось не так.

Наявність додатку, який може допомогти подбати про замовлення клієнтів, щоб вони могли не поспішати може поліпшити роботу офіціантам та зробити її ефективнішою. Програмне забезпечення має надсилати замовлення безпосередньо на кухню і пам’ятати, на якій столик надійшло замовлення і хто замовив кожну страву. Іноді клієнтам потрібна допомога, однак офіціанти можуть буду не у своїй робочій зоні через свої обов'язки, доволі часто причиною тому – завантаженість закладу.

Офіціантам потрібен додаток, у якому клієнти зможуть підкликати його до свого столику, у якій би частині ресторану вони не знаходились. Офіціантам також потрібна можливість отримувати повідомлення від шеф-кухаря, коли замовлення будуть готові, щоб їжа не простоювала на стойці видачі занадто довго.

5. Адміністратори.

   Управління рестораном і штат співробітників – це непроста робота, вони мають робити багато різних речей протягом дня, і часом це може бути занадто завантажено для їх робочого графіку. Буде важко відстежувати, які співробітники чергують у які зміни, а також визначати суму оплати праці за кожен період роботи працівників ресторанного закладу. Інше питання, пов’язане з працівниками – це відстеження їх робочого часу, наприклад, вчасно вони доїхали до

своєї зміни чи перебувають у ресторані, і обчислення розміру компенсації/штрафів працівникам за будь-які зміни у робочому графіку. Будь-який спосіб зменшити кількість зусиль, які адміністратори маю докласти до будь-якого з цих завдань – надзвичайно корисні.

На додаток до цих питань, ресторанний додаток повинен бути легко редагованим та зручним для користувача. Адміністратор мав би змогу легко оновлювати меню, щоб переконатися, що воно відповідає позиціям, які ми насправді подаємо у ресторані. Плани поверхів також постійно змінюються, наприклад, коли приходить велика група клієнтів, тому адміни повинні мати можливість редагувати план поверху одразу, щоб гарантувати, що клієнти та офіціанти зможуть використовувати план поверху, який актуальний на даний момент та відповідає плануванню ресторану.

Додаток може служити порталом для співробітників, дозволяючи співробітникам заходити та виходити, а також розраховуватиме належну суму для надання їм компенсації/штрафів/заробітних плат. Наявність цієї функції значно полегшить життя адміністратора, оскільки їм не доведеться робити жоден журнал співробітників або розраховувати оплату кожному співробітнику – натомість програма зробить це за нього. Додаток також перевірить місцезнаходження співробітників на основі IP-адреси або місцезнаходження GPS, так що співробітники можуть лише затримати час перебування на місці. Таким чином, адміністратори будуть знати, чи намагається працівник почати свою зміну ще до того, як він навіть прибуде до робочої зони ресторану.

У доповнення до цього, є дуже прості інтерфейси, які дозволять адміністраторам редагувати параметри меню та швидко регулювати план поверху.

6. Ранер (помічник офіціанта).

   Робота в якості помічника офіціанта може бути доволі виснажливою та тривалою. Іноді важко відстежувати всі столи, які зараз використовуються клієнтами у ресторані, і чи потрібно їх прибирати, оскільки клієнти вже залишили заклад.

   Наявність простого способу розглянути поточний стан столиків у додатку, зробить роботу ранера значно ефективнішою. Можливість побачити, які столи готові до прибирання та сервірування в будь-який момент, є величезною підмогою для ефективного підготовки столиків у ресторанному закладі. Закінчивши прибирання столу, ранери будуть мати можливість оновити статус столику до «вільний», щоб його можна було використовувати для новоприбувших клієнтів.

1.5.2. Актори та їх задачі

З’ясувавши бізнес-процеси та співробітників, тепер розглянемо їх як «акторів» UML-діаграми – учасників фронт-офісу, їх ролі у робочому процесі, та обов’язки.

Таблиця 1.1 – «Актори» та їх задачі

Актор	**Роль**	**Обов’язки**
Клієнт	Клієнт – відвідувач ресторану, який може замовити їжу на місці або на виніс, переглянути меню, замовити страви, поїсти та оплатити послуги.	Мета – зробити усе можливе, аби клієнт залишився задоволений послугами ресторану, з мінімальним часом очікування та якісним сервісом.
Гість	Гість – клієнт, який не має облікового запису у розроблюваній системі і вирішує відмовитись від функцій, які з’являються з завантаженим додатком. В усьому іншому клієнт і гість мають однакову роль.	Мета – зробити усе можливе, аби гість залишився задоволений послугами ресторану, з мінімальним часом очікування та якісним сервісом.

Закінчення таблиці 1.1

Співро-бітник	Співробітник – робітник ресторанного закладу, окрім менеджеру (кухарі, бармени, офіціанти, хостес, ранери).	Мета – надати гостю якісне обслуговування, з мінімальним часом очікування та якісним сервісом.
Менеджер	Менеджер – співробітник з додатковою відповідальністю, який керує роботою ресторану.	Мета – контролювати роботу співробітників, створювати та їх робочий графік, вести облік товарно-матеріальних цінностей, контролювати доходи, збитки. Стежити, аби клієнт був задоволений.

1.5.3. Задачі проекту

Після того, як для проекту були з’ясовані учасники, обов’язки та їх ролі у робочому процесі, розробимо задачі, які необхідно буде вирішувати системі. Для зручності проставимо для кожної задачі пріоритет виконання (де 5 – найвищий пріоритет, 1 – найнижчий).

Таблиця 1.2 – Задачі проекту

Пріоритет	**Задачі**
4	Додаток дозволить клієнтам вибрати незайнятий стіл (посадочне місце) з допомогою інтерактивної схеми посадочних місць після того, як користувач увійде в систему. Після вибору стіл буде позначений як “Зайнято”.
5	Додаток надасть інтерактивне графічне меню, коли клієнт сяде за стіл.
5	Додаток надасть працівникам можливість роботи на окремій вкладці «для працівників».
5	Додаток буде відстежувати робочі години співробітників для оплати праці на основі звітів про час приходу / час виходу.
3	Додаток повідомить ранера, коли клієнт оплатить рахунок, вказавши, що він покинув стіл, аби співробітник міг прибирати стіл.
3	Додаток дозволить менеджерам та офіціантам швидко надавати статус столам (вільно, прибрати, зайнято) для миттєвого інформування задля швидкої роботи.

Закінчення таблиці 1.2

4	Додаток дозволить клієнтам розділити рахунок.
3	Додаток надасть можливість зробити бронювання в ресторані заздалегідь.
4	Додаток повідомляє та надсилає вхідні замовлення шеф-кухару, додаючи замовлення в кінець черги (порядок розміщення залежить від часу замовлення).
3	Додаток надасть можливість замовити їжу на виніс (місце видачі замовлення та вивіз).
4	Додаток дозволить клієнту зареєструватися, аби брати участь у программі винагород та акціях.
4	Додаток повинен надвати бонуси кожен раз, коли клієнт робить покупку.
4	Програма повинна враховувати бонуси та застосовувати їх до загальної суми рахунку, коли користувач вирішить їх використовувати для оплати.
4	Додаток повідомляє офіціанта / офіціантку, коли їжа клієнта готова, аби вони могли подати замовлення до відповідного столу.
3	Додаток дозволить перекласти меню на англійську.
5	Додаток дозволить клієнтам відфільтрувати пункти меню відповідно до дієтичних обмежень/вподобань та повідомити кухаря про зміни у замовленні.
2	Додаток дозволить клієнту оцінювати та залишати коментарі до пунктів меню в кінці трапези.
3	Додаток дозволить офіціантам / офіціанткам позначати столи як "Зайнято".
4	Додаток дозволить клієнту викликати офіціанта / офіціантку за допомогою кнопки у додатку.
3	Додаток дозволить позначити стіл як “Вільно”, як тільки ранер його прибере.
5	Додаток дозволить клієнту оплатити на місці за допомогою терміналу, якщо це потрібно.

1.6. Аналіз існуючих програм для автоматизації фронт-офісу

POS-система (POS – Point of Sale) – це програмно-апаратний комплекс, що функціонує на базі фіскального реєстратора; за POS-системою закріплений типовий набір касових функцій: облік і відпуск товару, прийом і видача грошей,

анулювання факту покупки і ін. Зазвичай складовими частинами POS-системи називають POS-комп’ютер, POS-монітор, чековий принтер, фіскальний реєстратор, POS-клавіатуру, дисплей покупця, грошовий ящик, зчитувач магнітних карт [4].

Front-office (Фронт-офіс) відповідає за обслуговування відвідувачів. У ресторанному залі знаходяться POS-термінали з сенсорним екраном і принтером чеків, за допомогою яких офіціант вводить замовлення, враховуючи необхідні знижки і примітки гостей. [5]

Замовлення відправляється на барну стійку і кухню. Завдяки чековому принтеру кухар і бармен отримує необхідну інформацію для приготування замовлення. На робочому місці касира встановлений POS-термінал с фіскальним реєстратором, принтером чеків, де проводиться оплата замовлення і видача чека.

Back-office (Бек-офіс) – стосується оптимізації роботи керівника, менеджера зі закупівель, товарознавця і співробітників складу. Термінал збору даних і принтер етикеток збільшує швидкість і точність роботи співробітників складу. Товар при необхідності маркується за допомогою принтера етикеток. Оператор завдяки терміналу збору даних швидко зчитує інформацію зі штрих-кодів товарів і направляє дані в ПЗ товарного програму.

Автоматизація ресторану покликана проводити оперативну інвентаризацію, контролювати мінімальні залишки товарів на складі, замовлення необхідних інгредієнтів, аналізувати продажі, вести облік товарів від закупівлі до списання, враховувати грошові запаси і ще безліч функцій. Товарознавець має можливість проводити калькуляцію собівартості готової продукції.

Для ресторанів питання калькуляції є особливо актуальним. Калькуляція собівартості вирішує завдання ціноутворення і забезпечує контроль руху основних позицій товарно-матеріальних запасів підприємства.[6]

Тепер проведемо аналіз існуючих систем для автоматизації ресторанних комлексів.

1. R-Keeper

Найстаріший гравець на ринку автоматизації ресторанного бізнесу. Перша версія програми R-Keeper була створена ще в 1992 році, в даний момент система встановлена в 37 тисячах ресторанах в різних країнах світу.

Система включає різні модулі (фронт-офіс для безпосередньої роботи в ресторані і бек-офіс для контролю процесів і витрат) і додатки. Незважаючи на популярність, деякі користувачі скаржаться на досить складний і неінтуітівнимі інтерфейс.

Що стосується служби підтримки, то можливості підтримки через чат (що було б зручно) не передбачено.

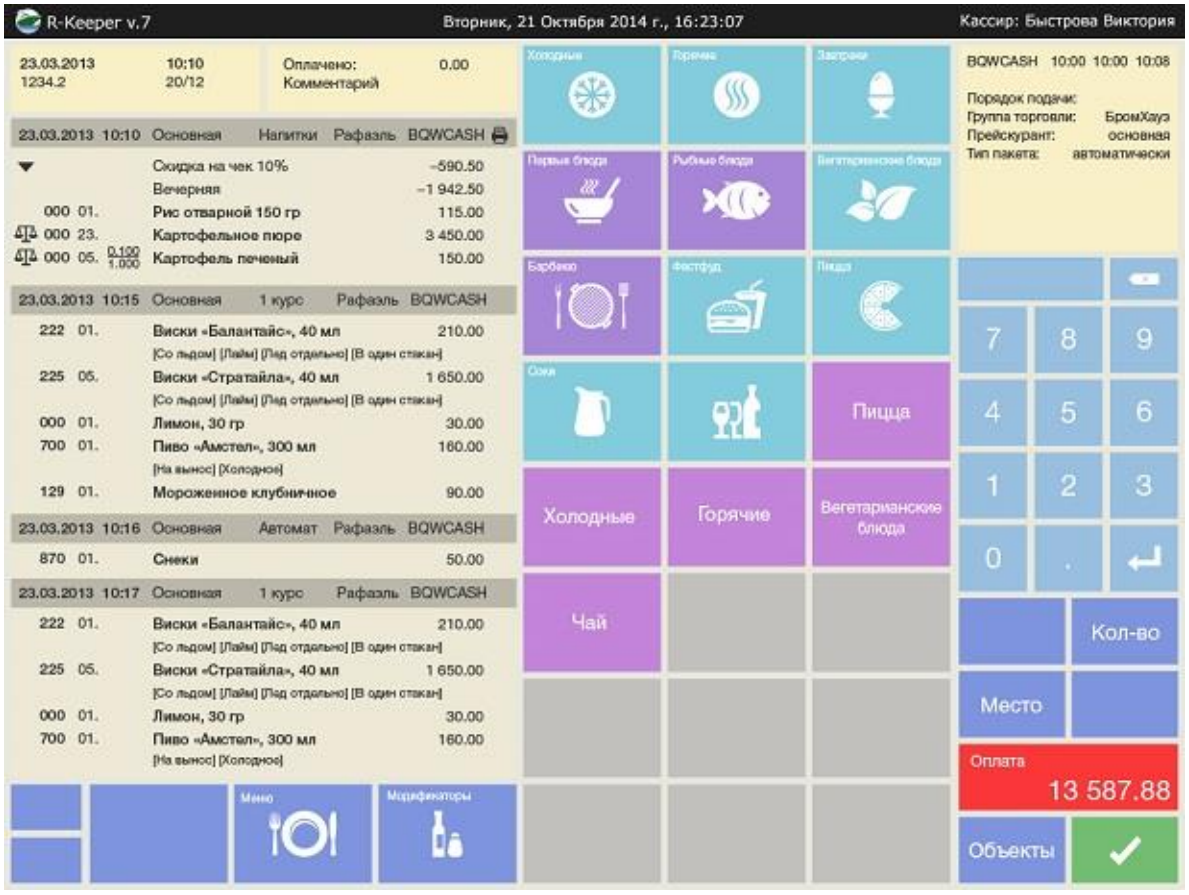

Рисунок 1.1 – Інтерфейс R-Keeper-у

Тип установки: система встановлюється локально.

Вартість ліцензії: від 30000 до 70000 гривень.

2. iiko

Проект, створений співзасновником ABBYY Давидом Яном, займає друге місце за популярністю в Росії. Система характеризується наявністю великої кількості різних модулів, які підключаються до спеціального серверу і працюють через нього. Є функціональність фронт і бек-офісу.

Велике число модулів, які підключаються за окрему плату, досить непросто адмініструвати, що призводить до підвищення ймовірності виникнення збоїв.

Підтримки з допомогою чату немає, зв'язок по телефону в режимі 24/7 досить оперативна, але з'єднують з операторами, які не в курсі всіх технічних тонкощів системи.

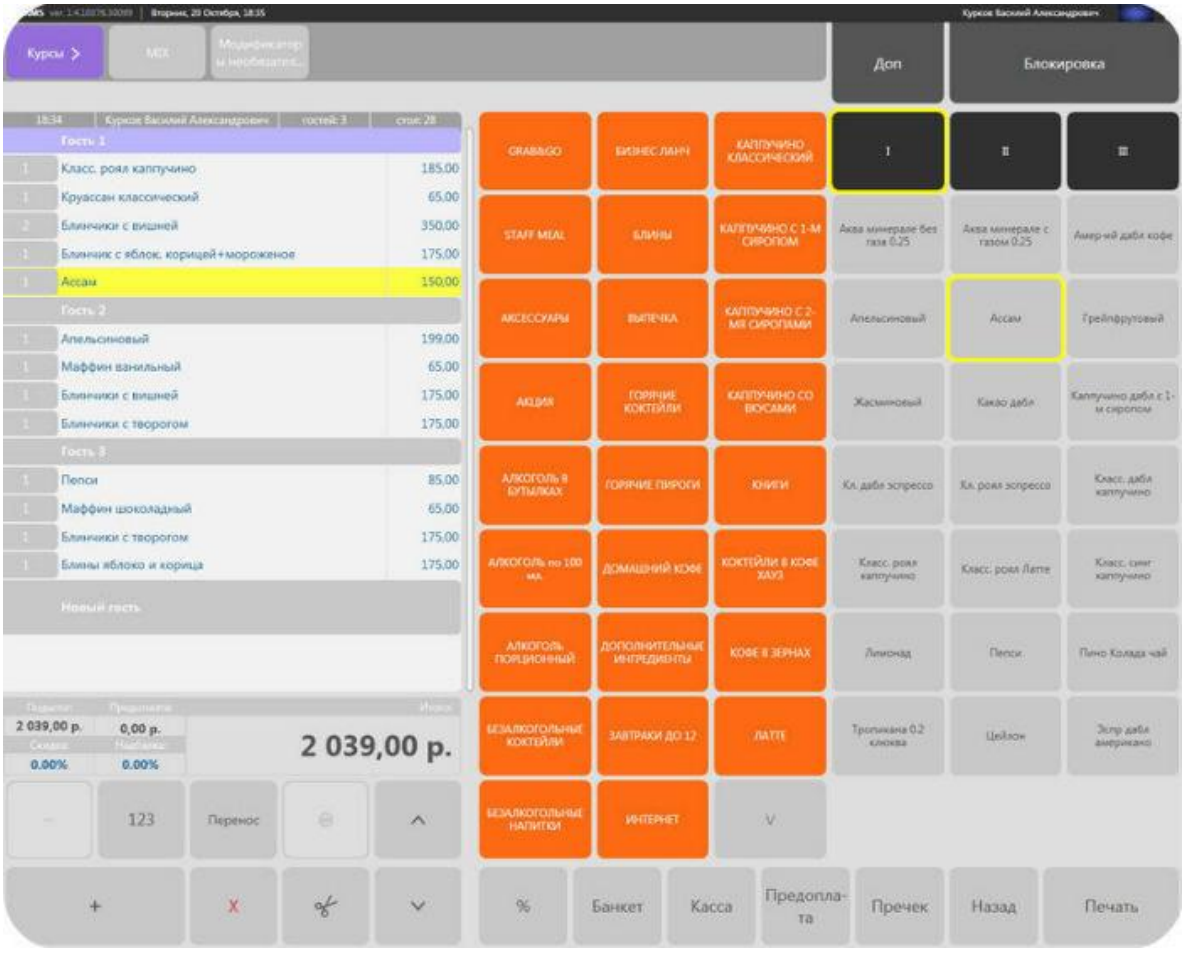

Рисунок 1.2 – Інтерфейс iiko

Тип установки: локальний.

Вартість: базова функціональність програми варто 10000 гривень за сервер, 7600 гривень за iikoFront (автоматизація касової станції) і 4500 грн. за iikoOffice (автоматизація управління складом). [7]

3. СБИС Presto

Програма включає фронт-офіс, бек-офіс і мобільний додаток для офіціанта і кухаря. Все, як у кращих представників галузі, але додані приємні фішки. Наприклад, замість грубої схеми залу промальована детальна, використовуються наочні аватарки для гостей, екран кухаря вміє промовляти замовлення, є можливість завантажити каталог продуктів і готових техкартою однією кнопкою, прив'язати відеофіксацію до кожного чеку і т.д.

Розробник – компанія Тензор, відома, головним чином, своїми рішеннями для електронної звітності в держоргани і документообігу. У Presto цей функціонал теж реалізований. Можна не купувати додаткове ПЗ, а прямо в системі відзвітувати по алкоголю, підписувати бухгалтерські та кадрові документи.

Рисунок 1.3 – Інтерфейс Presto

По роботі з системою зібрана база знань з докладними інструкціями і кейсами. Підтримка відповідає 24/7, є допомога через чат і віддалене підключення.

Тип установки: хмарне рішення, є офлайн-додаток на випадок проблем з інтернетом.

Вартість: від 300 грн./міс. до 760 гривень на місяць.

4. Jowi

Гібридний сервіс автоматизації, який поєднує хмарний і локальні підходи – модуль Jowi встановлюється в ресторані, а потім дані синхронізуються на віддалені сервери. Це дозволяє зберегти працездатність системи навіть при «падінні» інтернету.

Система також модульна – є додатки для менеджерів залу, офіціантів, які синхронізують дані між собою (офіціант прийняв замовлення, кухар побачив, яке блюдо потрібно готувати, менеджер отримує інформацію про час приготування і т.п.).

Рисунок 1.4 – Інтерфейс Jowi

За статистикою засновника сервісу програмою користуються 3500 закладів в країнах СНД. Досить гнучка система, наприклад, вона дозволяє змінювати різні дані навіть заднім числом – в «реальному світі» така необхідність повинна виникати досить часто.

Можливий зв'язок різними способами, відповіді досить швидкі.

Тип установки: локальний + SaaS.

Вартість: $150 в місяць.

5. Poster POS

Система автоматизації з хмарним зберіганням даних, що працює на планшетах iPad і Android. Робоче місце касира або офіціанта працює на планшеті, чеки друкуються на термальному принтері Epson TM-T20. [8]

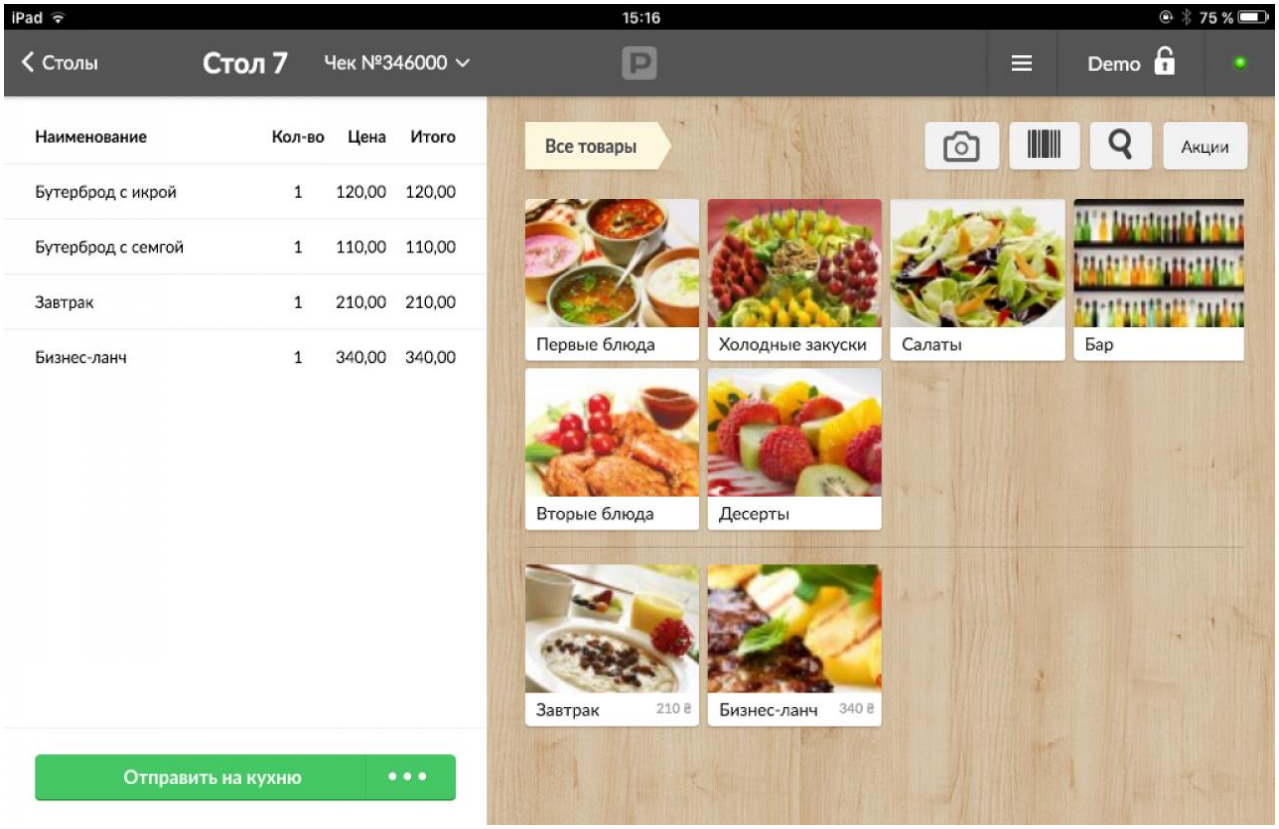

Рисунок 1.5 – Інтерфейс Poster POS

На сайті представлений відмінний FAQ по роботі з продуктом, є підтримка в чаті.

Тип установки: локальна + SaaS.

Вартість: $ 24- $ 79 в місяць.

6. Quick Resto

Хмарна система автоматизації ресторанного бізнесу, яка дозволяє управляти меню, розсаджуванням гостей, приймати оплату і налаштовувати принтери для друку чеків.

З невеликих недоліків – немає підтримки Windows, тільки iOS і Android. Для підключення периферійних пристроїв потрібно придбати QR Box.

У додатку меню страви йдуть з картинками – красиво, але в ході роботи може відволікати офіціантів.

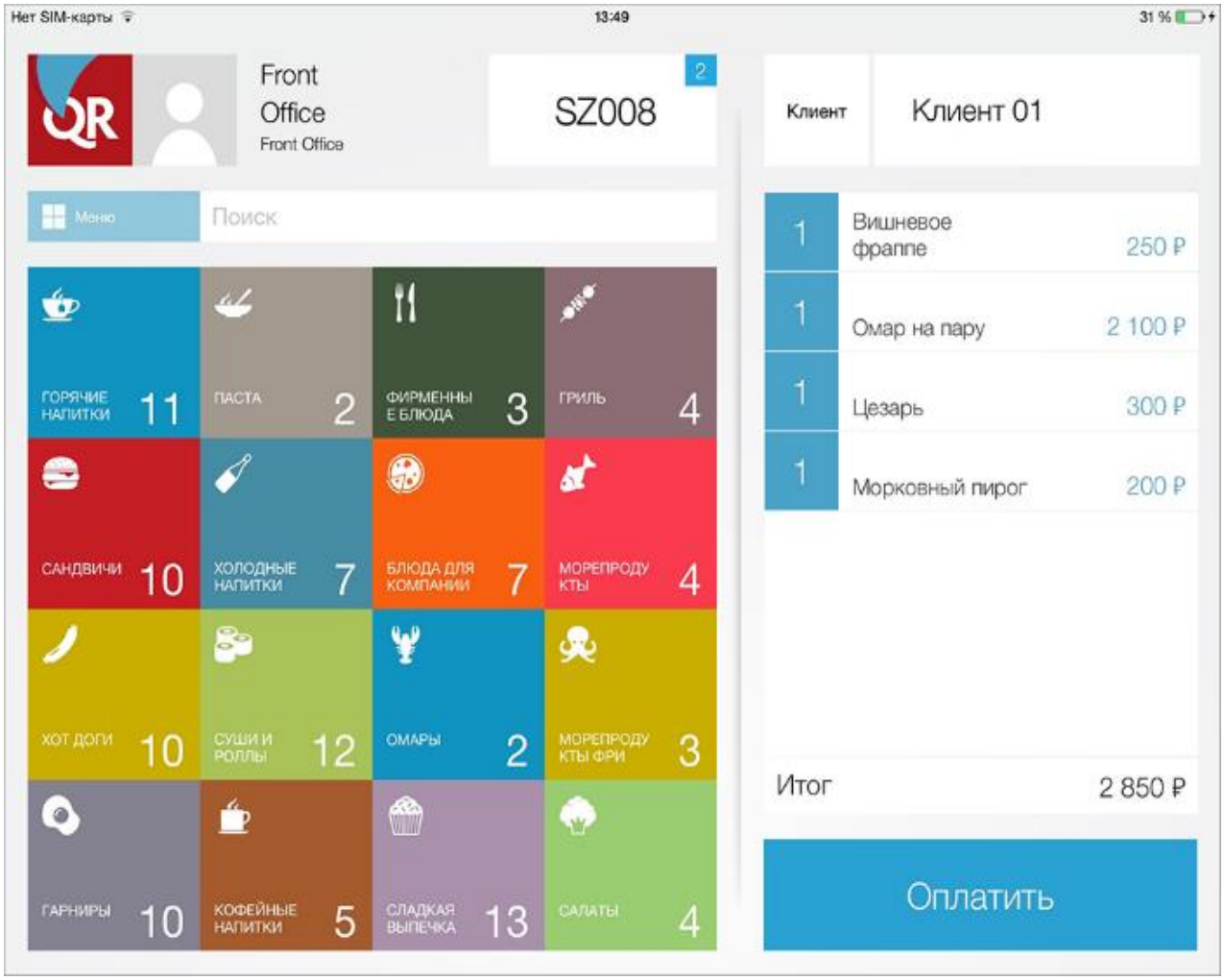

Рисунок 1.6 – Quick Resto

Підтримка працює в режимі 24/7, відповідає якісно і швидко.

Тип установки: Saas + Mobile.

Вартість: від 1000 гривень / місяць (вартість враховує обслуговування одного терміналу).

7. АСТОР

Компанія, що займається автоматизацією підприємств роздрібної торгівлі, логістики та індустрії харчування в Росії і СНД.

За даними сайту, рішенням для ресторанів користуються 12000 клієнтів.

Позиціонується як системи для невеликих і середніх ресторанів, існують модулі бек і фронт-офісу, є додатки для касирів, менеджерів залу, керуючих, комірників, офіціантів і так далі.

Сервісу миттєвих повідомлень немає, у підтримці ніяких відповідей про роботу системи не дають, пропонуючи ознайомитися з презентацією.

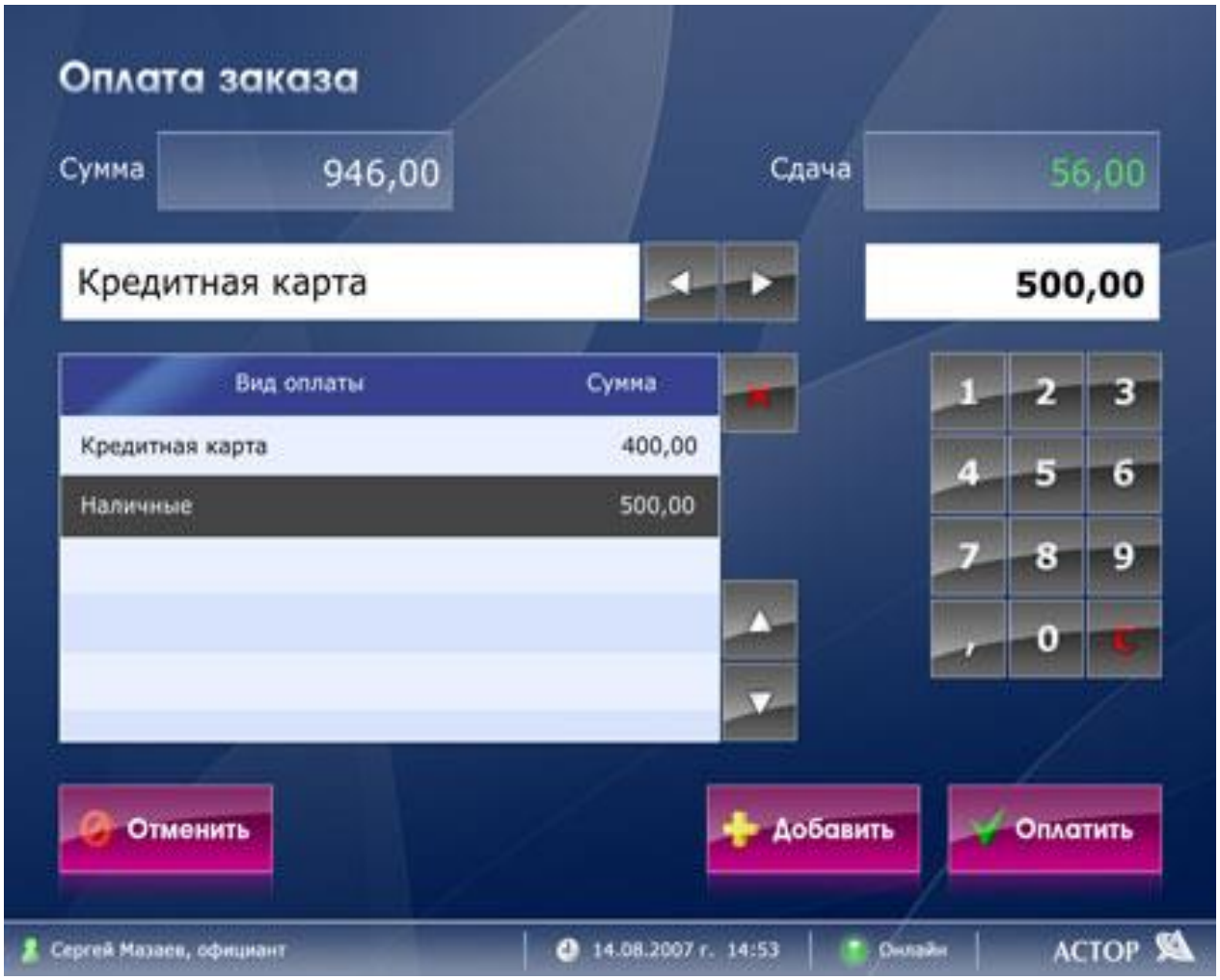

Рисунок 1.7 – Інтерфейс АСТОР

Тип установки: локальна.

Вартість: від $3100.

8. GBS.Market

Система для автоматизації магазинів і кафе – досить проста програма для невеликих закладів, що володіє відповідним скромним набором функцій. Звітність з терміналів можна відправляти на пошту.

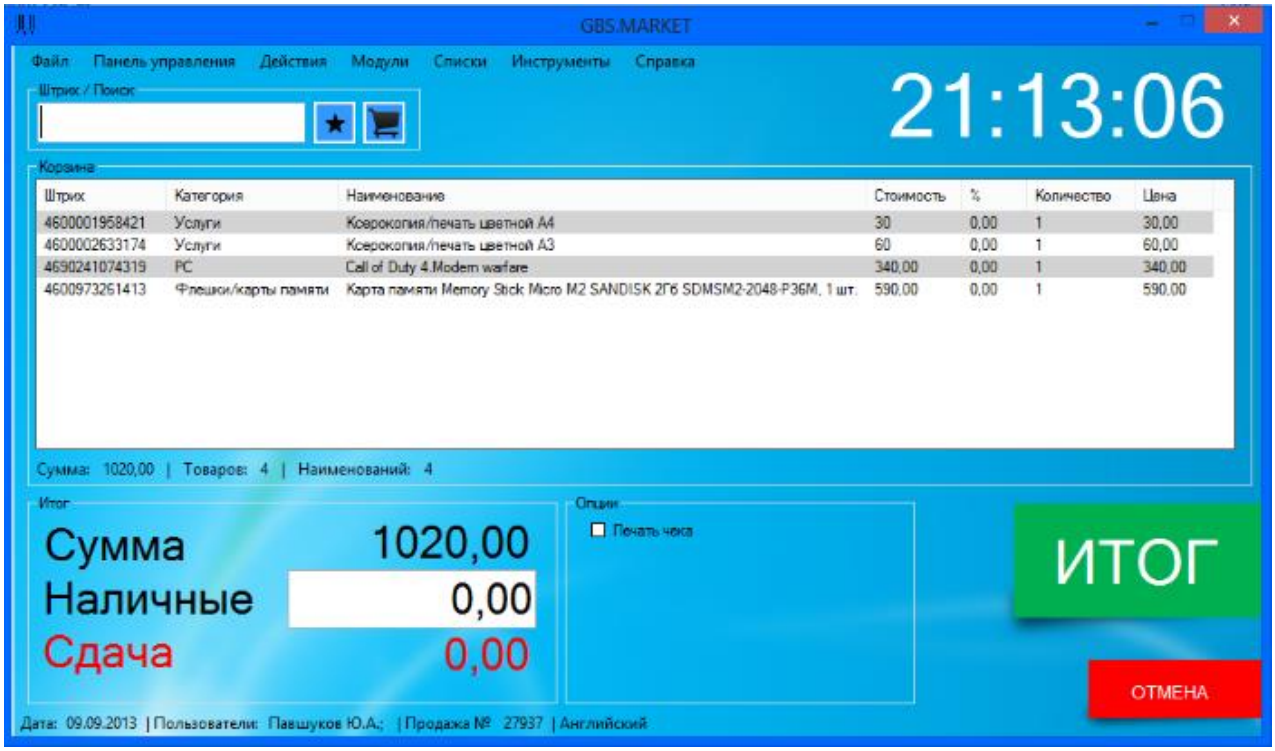

Рисунок 1.8 – Інтерфейс GBS.Market

Тип установки: локальна + SaaS.

Вартість: 100 грн. / міс за 1 ПК, 1200 грн. безлімітний тариф.

9. iSOK POS STORE

POS-система, спочатку створена для магазинів, але може бути використана і в невеликому ресторані або кафе.

Може працювати як в одиночному режимі (1 пристрій – тільки iPad), так і в здвоєному, коли один мобільний пристрій є головним, а друге – клієнтським. Синхронізація через інтернет, тобто моделі SaaS немає.

Тип установки: локальна + SaaS

Вартість: 119 руб / міс за 1 ПК, 3475 руб. безлімітний тариф

Сам по собі додаток безкоштовне, проте платити доведеться за деякі необхідні функції, на зразок настройки принтера та каси. Не сама гнучка

система – наприклад, немає можливості введення даних «заднім числом». Період зберігання даних на торгових пристроях становить від 2 до 40 днів. [9]

Ніяких варіантів зв'язку з розробниками знайти не вдалося, зате на сайті є непоганий FAQ.

Тип установки: Mobile.

Вартість: безкоштовно з оплатою за додаткових функцій.

10. Трактиръ

Сервіс, заточений під роботу з мобільними пристроями на iOS і Android. Є, додаток для офіціантів і електронне меню, за допомогою якого можна оформляти замовлення без офіціантів. Засмучує, що кожен модуль коштує окремих грошей. За різними оцінками системою користуються від 2000 до 3000 закладів.

Рисунок 1.10 – Інтерфейс Трактиръ

Тип установки: локальна + Saas + Mobile.

Вартість: фронт-офіс за 13000 руб, бек-офіс за 11500 руб.

11. Intellect Style

Проста, встановлюється локально, програма для автоматизації готелів і невеликих ресторанних закладів. Інтерфейс модуля офіціанта явно перевантажений – наприклад, є маса функцій, які можна активувати при роботі з рахунком, які можуть відволікати офіціанта.

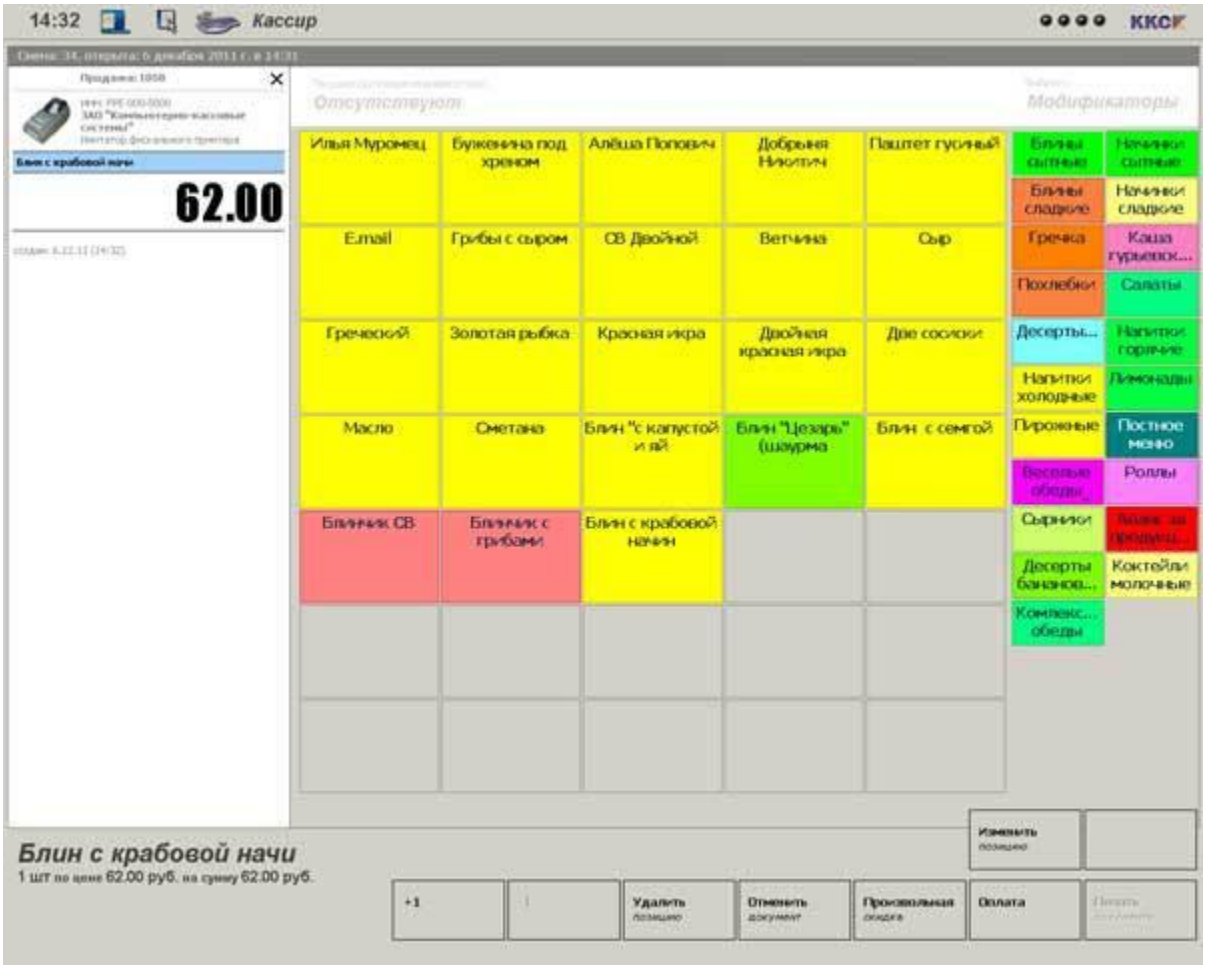

Рисунок 1.11 – Інтерфейс Intellect Style

Є онлайн-чат технічної підтримки.

Тип установки: локальна.

Вартість: від 7000 гривень.

12. Guscom.POS

Багатоплатформовий комплекс автоматизації, що включає кілька модулів – серед них модуль підключення фіскальних принтерів, додаток для офіціантів і сервіс адміністрування. [10]

Тип установки: локальний.

Вартість: за запитом.

13. Pos Sector

На сайті відразу пишуть, що користувачів системи вже більше 4500 тисяч. Локальний модуль з простим і інтуїтивно зрозумілим інтерфейсом – в ньому створюються страви, відділи, склади і подібні сутності. Є також браузерні додаток для офіціантів.

Рисунок 1.12 – Інтерфейс Pos Sector

Є кол-центр, підримка по телефону працює цілодобово.

Тип установки: локальна.

Вартість: базова ліцензія $590 (на 1 комп'ютер).

1.7. Технічні вимоги

Для роботи з додатком, який розроблюється для дипломного проекту, нижче наведений ряд технічних вимог, а також деякі рекомендації.

- Для роботи додатку необхідний смартфон або планшет, який працює на базі операційної системи iOS чи Android.

- Також для роботи додатку необхідне підключення до мережі інтернет. Додаток може працювати без доступу до інтернету з обмеженим функціоналом. Дані, у випадку відсутності інтернету, згодом збережуться та відобразяться необхідним чином у додатку.
- Програму потрібно попередньо завантажити на необхідний пристрій та створити для кожного користувача обліковий запас з відповідною для нього роллю.

Наш програмний додаток зможе працювати на мобільних пристроях, таких як смартфони та планшети, що використовують мережеву підтримку (WiFi або мобільні дані). Компонування для обох типів пристроїв (смартфонів та планшетів) буде однаковою, оскільки обидва вони використовують одну версію програми. Користувацька програма не призначена для використання на комп'ютері, тому не існує розробленого макета для робочого столу.

Це програмне забезпечення буде сумісним з операційними системами iOS та Android. Для iOS ми будемо підтримувати iOS 12 та новіші версії, які охоплюють понад 80% iPhone в обігу. Що стосується операційної системи Android, прагнемо підтримати API 23-Marshmallow, підтримуючи цю операційну систему, будемо підтримувати 71% поточних телефонів Android.

Це програмне забезпечення буде взаємодіяти через сенсорний інтерфейс. Наше програмне забезпечення буде мати кольоровий дисплей з мінімальною роздільною здатністю від 1330 x 700 пікселів до 2960 x 1440 пікселів. Для завантаження та встановлення програми користувач повинен мати мінімальний обсяг пам’яті 50 мегабайт. Додаток буде здійснювати зв’язок через ресторан Wi-Fi і буде надсилати та витягувати запити на зв’язок із сервером, а також вимагатиме доступу до GPS.

Висновок до розділу

У рамках дипломного проекту поставлена задача для написання додатку, який допоможе ресторанним комплексам підвищити ефективність їх роботи.

Розглянута актуальність обраної теми. З’ясовані переваги та недоліки неавтоматизованих гастрономічних закладів, знайдені проблеми, що виникають під час обслуговування гостей, та шляхи їх вирішення за допомогою розроблюваного додатку.

Розглянуті основні бізнес-процеси ресторанного закладу, визначені ролі та задачі співробітників, поставлені задачі, які має виконувати додаток, згідно з робочими обов’язками співробітників ресторану. Проведений аналіз існуючих програм для автоматизації фронт-офісу та означені технічні вимоги до додатку.

РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Вибір архітектури програмного забезпечення

У нашому додатку також використовується модель архітектури Клієнт-Сервер, яка є взаємовідносинами один до багатьох. Сервер надає / зберігає дані. У ресторані серверами є безліч різних інтерфейсів для клієнтів та персоналу ресторану.

Користувачами є клієнти, яким доводиться вводити інформацію в три інтерфейси; замовлення їжі з меню, бронювання, вхід в систему винагород та здійснення платежів.

Співробітники – це клієнти, які використовують сервер синхронізації входу / виходу для реєстрації своїх змін, за які вони отримують оплату.

Менеджери – це клієнти, які використовують свій сервер опцій для редагування інформації про співробітників та планування, редагування меню та редагування інформації щодо столиків.

Вибрана архітектура клієнт-сервер – доволі популярна, також вона є обчислювальною моделлю, коли сервер розміщує, розподіляє та контролює більшість ресурсів, а також послуг, що використовуються клієнтом.

Такі структурні конструкції складаються з однієї або декількох клієнтських систем, підключених до центральних або основних серверів через мережу, яку ми зазвичай називаємо підключенням до Інтернету. Всі подібні системи, пов'язані з нею, діляться обчислювальними ресурсами.

Архітектура клієнт-сервер в деяких інтерпретаціях називається структурою, що допомагає проводити обчислення для мереж. Її друга назва походить через те, що запити та пов'язані з ним інші, попутні операції, розподіляються по мережі. Нижче пояснення, як вона працює. [11]

В архітектурі клієнт-сервер, коли клієнтський комп'ютер надсилає запит на передачу даних на сервер через Інтернет, сервер приймає запитувані дані, обробляє їх і доставляє запитувані пакети даних назад клієнту.

Особливістю є те, що серверний комп’ютер має можливість керувати численними клієнтами одночасно. Крім того, один клієнт може підключатися до численних серверів за одну позначку часу, де кожен сервер надає різний набір послуг для цього конкретного клієнта.

Іншим архітектурним стилем, що використовується, є модель Peer-to-Peer. Ця модель є розподіленою архітектурою додатків, яка розділяє завдання між задачами одного рангу. Вона буде використана для оплати замовлення клієнтом у ресторані, а модель децентралізована, тому кожен платіж обробляється окремо. [12]

У нашому додатку ця модель використовується, коли якщо клієнт платить карткою, тоді модель однорангової мережі стає платіжною послугою для переказу коштів з банківського рахунку клієнтів на рахунок ресторану.

2.2. Вибір технологій додатку

Додаток розроблений з допомогою фреймворку React Native, відповідно, мова програмування проекту – Java Script.

React Native (також відомий як RN) – це популярний фреймворк мобільних додатків на основі JavaScript, який дозволяє створювати мобільні додатки, що відображаються власним чином, для iOS та Android.

Фреймворк дозволяє створювати програми для різних платформ за допомогою однієї і тієї ж кодової бази.

Вперше React Native був випущений Facebook як проект з відкритим кодом у 2015 році. Всього за пару років він став одним з найкращих рішень, що використовуються для мобільної розробки.

Розробка React Native використовується для роботи з деякими провідними мобільними додатками у світі, зокрема Instagram, Facebook та Skype. [13]

Глобальним успіхом React Native користується за рядом причин.

1. Використовуючи React Native, компанії можуть створити код лише один раз і використовувати його для живлення своїх додатків для iOS та Android. Це означає величезну економію часу та ресурсів.
2. React Native був побудований на основі React – бібліотеки JavaScript, яка вже була надзвичайно популярною, коли вийшов мобільний фреймворк.
3. Фреймворк надав можливість розробникам інтерфейсів, які раніше могли працювати лише з веб-технологіями, створювати надійні, готові до виробництва програми для мобільних платформ.

Цікаво, що, як і у багатьох революційних винаходах, React Native був розроблений як відповідь на велику технологічну помилку.

2.3. Вибір основного сховища даних

У якості бази даних, для додатку була обрана MySQL.

MySQL – це система управління реляційними базами даних (RDBMS) із підтримкою Oracle з відкритим кодом, заснована на мові структурованих запитів (SQL). MySQL працює практично на всіх платформах, включаючи Linux, UNIX та Windows. Хоча його можна використовувати в широкому діапазоні програм, MySQL найчастіше асоціюється з веб-додатками та публікацією в Інтернеті.

MySQL є важливим компонентом корпоративного стеку з відкритим кодом, який називається LAMP. LAMP – це платформа веб-розробки, яка використовує Linux як операційну систему, Apache як веб-сервер, MySQL як реляційну систему управління базами даних та PHP як об'єктно-орієнтовану мову сценаріїв (іноді замість PHP використовується Perl або Python). [14]

Спочатку задуманий шведською компанією MySQL AB, MySQL був придбаний Sun Microsystems у 2008 році, а потім Oracle, коли він придбав Sun у 2010 році. Розробники можуть використовувати MySQL під загальною

публічною ліцензією GNU (GPL), але підприємства повинні отримати комерційну ліцензію від Oracle.

Сьогодні MySQL є СУБД, що стоїть за багатьма провідними веб-сайтами у світі та незліченними корпоративними та споживчими веб-додатками, включаючи Facebook, Twitter та YouTube.

2.4. Вибір архітектури серверного додатку

Для додатку вирішено використати Layered Pattern mode, який зазвичай застосовується для загальних десктопних програм.

Три шари – це рівні презентації, програми та даних.

1) Презентаційний рівень бачать користувачі на фронт-енду, як клієнти, так і персонал ресторану.
2) Шар додатків і даних знаходиться на бек-енді та відстежує введену користувачем інформацію, а також відповідає на команди з рівня презентації.

Ця модель підходить для нашого застосування, оскільки шари абстрагуються один від одного, щоб, таким чином, працювати паралельно.

Підсистема, зображена на рисунку 2.1, відображає нашу тришарову архітектуру, яка складається з рівня презентації, рівня додатків та рівня даних. Кожен рівень несе конкретну відповідальність за всю систему.

Шар презентації обробляє рівень інтерфейсу користувача. Цей рівень містить інформацію про шеф-кухаря, менеджера, ранера/ офіціанта / офіціантку та інтерфейси клієнта.

Прикладний рівень є середнім рівнем цієї архітектури, який обробляє зв'язок між верхнім і нижнім шарами, по суті керуючи загальною роботою. Він буде керувати бізнес-логікою, тобто набором правил, необхідних для запуску програми для настанов, наданих організацією. У цьому шарі міститься Комунікатор і Контролер. [15]

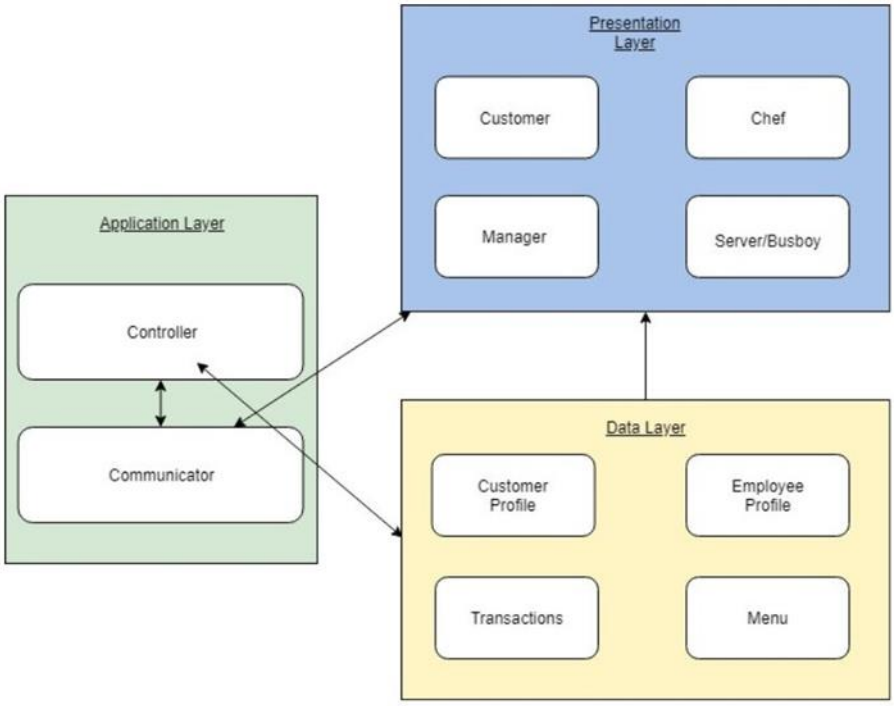

Рисунок 2.1 – Три шари архітектури додатку

Контролер використовується для полегшення завдань між шарами, а Комунікатор використовується для передачі інформації між шарами. Найнижчий рівень – це рівень даних, який займається збереженням та вилученням даних. Цей рівень буде містити інформацію, що стосується профілів працівника та клієнтів, а також доступні транзакції меню та ресторану.

Кожному шару не потрібно турбуватися про інші шари, через розділення проблем. Крім того, через шари ізоляції зміни, внесені в один шар, як правило, не впливають на інші шари. Завдяки цьому архітектура робить тестування та діагностування проблем більш керованими.

2.5. Вибір технологій для обладнання

Для нашого програмного забезпечення в ресторані буде клієнт, який працює на планшетах та мобільних телефонах. Також буде бек-сервер, який буде містити REST API та базу даних. Додаток можна запускати на багатьох різних пристроях.

Вимоги до обладнання. Наша система матиме RESTful API, який використовується для зв'язку з нашими серверами. Запити POST і GET ініціюються від клієнта і дають відповідь від сервера. Запити POST надають додаткові дані від клієнта на сервер. Запити GET отримують корисну інформацію для клієнта.

Також існує фізичний сервер, який ми будемо контролювати за допомогою Node.js. Для сервера Node.js містить пакети Express і Sequelize, що сприяє зв'язку між сервером і базою даних.

Express – це мінімальна та гнучка структура веб-додатків Node.js, яка забезпечує надійний набір функцій для розробки веб- і мобільних додатків. Express використовується для створення дескриптора маршрутизації та обробки запитів від клієнта.

Sequelize – це заснований ORM для Node.js, який дозволить системі оперувати об’єктами, які відображаються в реляційній базі даних.

Він підтримує MySQL і має надійну підтримку транзакцій, реплікацію читання тощо.

Для нашої бази даних ми будемо використовувати MySQL. MySQL – це реляційна система управління базами даних, яка використовує структуровану мову запитів (SQL).

SQL допомагає додавати, отримувати доступ та керувати вмістом у базі даних. [16]

2.6. Вибір математичної моделі

Нашому програмному забезпеченню потрібно зберігати постійні дані, які повинні пережити єдине виконання системи. З одного боку, інформацію про користувача потрібно зберігати в таблиці користувачів. Наше програмне забезпечення має запам'ятати ім'я та прізвище клієнта, ім'я користувача та пароль, а також отримані бали винагороди та інформацію про оплату.

Для працівника програмне забезпечення для вас також має запам'ятати ім'я та прізвище працівника, ім'я користувача та пароль, зарплату, відпрацьований час, графік роботи тощо. Нашому програмному забезпеченню також потрібно буде утримувати транзакції ресторану, щоб менеджер міг отримати доступ до нього, коли їм заманеться, і мати можливість відстежувати прибуток (загальний дохід і прибуток).

Оскільки менеджер також повинен мати можливість коригувати макет столу, наше програмне забезпечення має мати можливість запам’ятовувати поточний макет ресторану. Іншою важливою інформацією, яку потрібно зберегти, є доступне меню разом із підсумковими рейтингами, розраховані для кожного товару, ціни на товари, назви товарів, рецепти та номер товару. Вся ця інформація буде зберігатися в базі даних MySQL, що працює на нашому веб-сервері. Вся описана вище інформація схематично наведена у Додатку А.

2.7. Мережевий протокол

Для того, щоб заповнити програму автоматизації ресторанів, буде впроваджено REST API. API REST виконуватиме дії на сервері на основі запитів HTTP POST та GET. HTTP REST добре підходить для архітектури клієнтського сервера. [17]

Інтерфейс програми буде побудований за допомогою React Native, який буде обробляти виконання запитів. Користувач взаємодіє з додатком, і коли вони виконують завдання, запит надсилається на сервер через HTTP, де інформація оброблятиметься.

Наприклад, якщо користувач замовляє їжу після натискання кнопки підтвердження, замовлення буде відправлено через HTTP POST-запит на сервер, який відповість назад інформацією, включаючи передбачуваний час очікування, який необхідний для того, щоб клієнт відображав користувачеві.

Інший приклад – коли користувач шукає інформацію про те, скільки очок винагороди користувач заробив, буде видано запит GET від клієнта на сервер із

запитом залишку балів винагороди користувача, сервер обробить цей запит і у своєму тілі відповіді включить баланс.

Висновок до розділу

У розділі «інформаційне забезпечення» вибрана та детально описана трьох шарова архітектура розроблюваного додатку. Описані технології, за допомогою яких додаток написаний – React Native, визначене основне сховище даних – MySQL. Детально описана архітектура серверного додатку, а також описана математична модель проекту, визначений мережевий протокол.

РОЗДІЛ 3. РОЗРОБКА АЛГОРИТМІЧНОГО І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Даний монолітний додаток побудований як єдине ціле. Корпоративні додатки ґрунтуються на трьох складових частинах: базі даних (що складається з багатьох таблиць, як правило, у системі реляційних баз даних), інтерфейсу користувача на стороні клієнта (що складається додатку, написаному за допомогою React Native, мовою JavaScript), а також серверної програми.

3.1. Алгоритми роботи додатку

Для кращого розуміння роботи системи і полегшення розробки була створена use-case діаграма з описом роботи системи за участі акторів.

Опис діаграми:

UC-1: Reservation – Дозволяє клієнтам забронювати стіл онлайн до того, як вони відвідають ресторан для резервування.

UC-2: Payment – Дозволяє клієнтам отримати рахунок з замовлення, вибрати варіант чеку (на пошту/паперовий), а також вибрати спосіб оплати.

UC-3: View Menu – Дозволяє користувачам продивитись усе меню у системі.

UC-4: Meal Prep – Дозволяє кухарам отримувати оновлені статуси замовлення та особливі вподобання до замовлених страв.

UC-5: Rate Food – Дозволяю клієнтам проставити оцінки за отримані послуги та страви у меню.

UC-6: Food Filters – Дозволяє клієнтам відфільтрувати меню за бажанням та змінити інгредієнти страви.

UC-7: Clocking In/Out – Дозволяє співробітникам відмічати початок та кінець робочої зміни.

UC-8: Serving – Дозволяє офіціантам відслідковувати статус замовлення та клієнтів (очікують замовлення, чекають чек і тд).

UC-9: Placing an Order – Дозволяє серверу чи користувачу надіслати своє замовлення до приготування.

UC-10: Table Marking – Дозволяє серверу надавати статус столам «Зайнято», коли клієнт зайняв стіл та ранерам статус «Вільно», коли той прибере зі столу.

UC-11: Earning Rewards – Коли клієнт робить покупку, він отримує бонуси.

UC-12: Redeeming Rewards – Коли клієнт робить покупку, він може розплатитись за допомогою відповідної кількості бонусів.

UC-13: Take-Out – Можливість клієнтам замовити їжу для самовивозу.

UC- 14: Table Selection – Дозволяє клієнтам вибрати вільний стіл у закладі, після того, як вони увійшли в додаток, або продовжити як гість, якщо вони вже у закладі, шляхом перегляду столів усіх статусів да обрання вподобаного.

UC-15: Floor Plan Status – Дозволяє менеджеру і офіціантам змінювати статус столів «Вільно – Зайнято – Прибрати» у потрібному порядку.

UC-16: Login – Дозволяє користувачу увійти у систему (від ролі користувача залежить який інтерфейс він побачить і буде користуватись).

UC-17: Create Account – Дозволяє користувачу створити обліковий запис і отримати відповідний до ролі інтерфейс системи.

UC-18: Translation – Дозволяє користувачу вибрати мову перегляду меню, аби не користуватись перекладачем.

UC-19: Menu Changes – Дозволяє менеджеру вносити швидкі зміни до меню.

Переглянути Use-Case діаграму можна у Додатку Б.

На Use-Case діаграмі відображаються всі взаємозв'язки між Акторами, Акторами-учасниками, Випадками використання та базою даних.

Актори та актори-учасники – замовник / гість, шеф-кухар, ведучий, офіціант / офіціантка, менеджер, ранер. Деякі випадки взаємодії ініціюються

актором. Деякі випадки взаємодії поширюються на інші, які є пов'язаними з ними у роботі. Описуються випадки взаємодії з базою даних.

3.1.1. Оплата

Актор: Клієнт.

Мета актора: Оплатити виконане замовлення.

Актори-учасники: База даних, Офіціант.

Передумови:

- Користувач завантажив систему.
- Користувач увійшов як Клієнт і відкрив меню.

Постумови:

- Користувачу пропонується спосіб оплати.
- Користувачеві пропонується варіант отримання.

Перелік подій для основного сценарію успіху:

- Клієнт натискає кнопку «Оформити замовлення» на екрані меню.
- Система відображає меню оплати.
- Система відображає варіанти способу оплати.
- Клієнт обирає кнопку “Готівка” для оплати готівкою.
- Система пропонує клієнтові варіанти отримання чеку.
- Клієнт вибирає кнопку «Друк квитанції».
- Система повідомляє офіціанту / офіціантці надрукувати квитанцію та повернути її замовнику.

Перелік подій для альтернативного сценарію успіху:

- Клієнт натискає кнопку «Оформити замовлення» на екрані меню.
- Система відображає меню оплати.
- Система відображає варіанти способу оплати.
- Клієнт обирає кнопку “Кредит” для оплати кредитом.
- Система пропонує клієнту варіанти отримання чеку.
- Клієнт вибирає кнопку “Квитанція електронною поштою”.
- Система пропонує користувачеві вказати електронну адресу.

- Клієнт вводить електронну адресу та отримує квитанцію електронною поштою.

Діаграма послідовностей для функції оплати готівкою наведена у Додатку В, для оплати карткою – у Додатку Г.

Опис діаграми.

Клієнт натискає кнопку оплати та вибирає спосіб оплати. База даних записує інформацію, яка їй потрібна. Якщо оплата здійснюється готівкою, офіціант отримує повідомлення та бере готівку. Як варіант, якщо вибраним варіантом була картка, інформація надсилається в Stripe API для обробки, Stripe поверне інформацію про успіх платежу.

3.1.2. Перегляд меню

Актор: Клієнт.

Мета актора: Переглянути меню.

Актори-учасники: База даних.

Передумови:

- Користувач завантажив систему.

Постумови:

- Користувачеві відображається доступне меню.

Перелік подій для основного сценарію успіху:

- Система надає параметри для «Вхід», «Продовжити як гість» та «Створити рахунок».
- Клієнт вирішує увійти до свого облікового запису.
- Система пропонує клієнтові увійти зі своїми даними користувача.
- Клієнт вводить інформацію для входу у систему.
- Система надає параметри “Dine-In” або “Take-Out”.
- Клієнт вибирає опцію “Dine-In”.
- Система дозволяє замовнику вибрати вільний столик.
- Клієнт сідає за столик.
- Система надає можливості вибору дієтичних обмежень.

- Клієнт вирішує не фільтрувати меню.
- Система відображає доступне меню.

Перелік подій для альтернативного сценарію успіху:

- Система надає параметри “Вхід”, “Реєстрація” або “Продовжити як гість”.
- Клієнт вибирає “Продовжити як гість”.
- Система надає параметри “Dine-In” або “Take-Out”.
- Клієнт вибирає опцію “Dine-In”.
- Система дозволяє замовнику вибрати вільний столик.
- Клієнт сідає за столик.
- Система надає можливості вибору дієтичних обмежень.
- Клієнт вирішує не фільтрувати меню.
- Система відображає доступне меню.

Діаграма послідовностей для функції перегляду меню наведена у Додатку Д.

Опис діаграми.

Клієнт просить систему відобразити меню, система запитує у бази даних потрібну інформацію з меню, що відповідають будь-якому фільтру, а позиції відображаються користувачеві системою.

3.1.3. Відкриття/закриття зміни

Актор: Співробітник.

Мета актора: Точно фіксувати кількість часу, який вони витрачають на роботу, і переконатися, що працівник фактично знаходиться в ресторані, коли вони відривають та закривають робочу зміну.

Актори-учасники: усі працівники ресторану.

Передумови:

- Користувач завантажив додаток.

Постумови:

- Користувач успішно здійснив вхід / вихід.

- База даних реєструє інформацію для подальшого використання.

Перелік подій для основного сценарію успіху:

- Працівник відкриває програму та обирає вхід як звичайний працівник.
- Система запитує працівника про інформацію для входу в систему.
- Співробітник входить в систему зі своїми даними.
- Система відкриває працівникові портал працівників.
- Співробітник вибирає кнопку входу.
- Система відслідковує дані про час співробітника.
- Система перевіряє, чи працівник насправді знаходиться в ресторані, перевіряючи IP-адресу та координати GPS.
- Коли працівник закінчив зміну протягом дня, працівник вибере кнопку виходу.
- Система відстежує дані часу виходу співробітника.
- Система перевіряє, чи працівник насправді знаходиться в ресторані, перевіряючи IP-адресу та координати GPS.

Перелік подій для альтернативного сценарію успіху:

- Співробітник відкриває програму і вирішує ввійти в систему як менеджер.
- Система пропонує менеджеру ввести дані для входу.
- Менеджер авторизується зі своєю інформацією.
- Система відкриває менеджеру відповідний інтерфейс.
- Менеджер вибирає кнопку входу.
- Система відстежує дані про час роботи менеджера.
- Система перевіряє, що менеджер насправді знаходиться в ресторані, перевіряючи IP-адресу та координати GPS.
- Коли менеджер закінчить зміну за день, менеджер вибере кнопку виходу.
- Система відстежує дані часу виходу менеджера.

- Система перевіряє, що менеджер насправді знаходиться в ресторані, перевіряючи IP-адресу та GPS координати.

Діаграма послідовностей для відкриття/закриття зміни співробітників наведена у Додатку Е.

3.1.4. Бонусний рахунок

Актор: Клієнт.

Мета актора: Отримати купони / знижки на основі суми попередніх покупок у ресторані.

Актори-учасники: База даних.

Передумови:

- Користувач завантажив систему.
- Користувач має намір здійснити покупку.

Постумови:

- Користувач заробив відповідну кількість балів за те, скільки було витрачено.

Перелік подій для основного сценарію успіху:

- Система спочатку просить замовника ввійти / створити обліковий запис.
- База даних отримає поточний баланс бонусів користувача.
- Клієнт вибирає потрібні страви у ресторанному закладі та підтверджує замовлення.
- Клієнт проводить оплату.
- База даних оновлює бонусний рахунок на основі транзакції.

Перелік подій для альтернативного сценарію успіху:

- Система спочатку пропонує споживачу ввійти / створити обліковий запис.
- База даних отримає поточний баланс бонусів користувача.
- Клієнт підтверджує замовлення.
- Клієнт не виконує оплату.

- База даних відповідає, що транзакція є неповною, замовлення не виконано, бали не списуються з бонусного рахунку.

Діаграми послідовності відображають деякі потенційні шляхи, якими можна пройти при спробі оплати за допомогою обробки платежів. Друга діаграма (оплати) показує найкращий сценарій під час спроби оплатити їжу кредитною або дебетовою карткою, після того як клієнт взаємодіє з інтерфейсом і вводить свою платіжну інформацію, інформація буде надіслана контролеру, який потім надішле дані до конкретної бази даних для картки (наприклад, VISA, MasterCard тощо), щоб перевірити, чи надіслана інформація насправді є дійсною.

На діаграмі додатків Г показано, як цей процес працює безперебійно, з першої спроби вводиться дійсна інформація про картку. На третій діаграмі показаний подібний процес, але з введенням недійсної інформації про картку, що призводить до циклу, доки користувач не введе дійсну інформацію, яка може бути використана для оплати.

Принципи високої згуртованості та низького зв’язку активно використовувались у різних місцях під час створення діаграм. Застосування контролера використовує високий принцип згуртованості, оскільки він дозволяє багатьох з інших об'єктів, такі як інтерфейси і бази даних для виконання намічених завдань (взаємодіє з користувачем і перевіркою інформації карти) без необхідності відправляти повідомлення на інші об'єкти, що беруть участь на схемі про стан їхніх дій.

Замість того, щоб інтерфейс надсилав інформацію про карту до бази даних, або інтерфейс клієнта надсилав інформацію на інтерфейс офіціанта про необхідність отримання готівки, інтерфейси надсилають цю інформацію контролеру, щоб контролер тоді міг виконувати відповідні завдання за необхідності. Принцип низького зв’язку значною мірою використовувався в процесі оплати готівкою, щоб забезпечити швидкість і чистоту.

3.2. Структурування даних додатку

1. Клас: Menu.

Атрибути:

- menuItems: список товару – продукти харчування, доступні замовнику.

Методи:

- getAll () – база даних отримує всі пункти меню.
- filterItem () – база даних відфільтровує елементи меню відповідно до переваг.

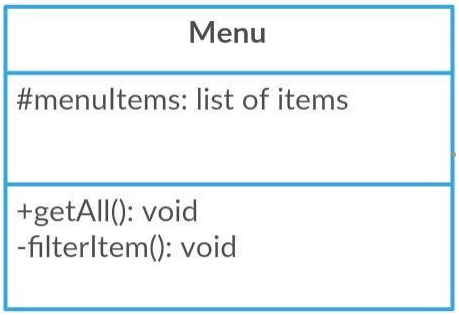

Рисунок 3.1 – Клас Menu

2. Клас: Order.

Атрибути:

- totalPrice: double – загальна ціна всіх предметів замовлення.
- orderItems: Item [] – список продуктів харчування в кошику, вибраних замовником.
- specialRequest: String – місце, де клієнти можуть висловити будь-які запити, недоступні в інтерфейсі користувача.
- Status: String – підтримуйте статус замовлення.

Методи:

- removeItem (foodID #) – видаляє їжу з існуючого замовлення – бере ідентифікатор та кількість їжі.
- addItem (foodID #) – додає їжу / напої, вказані замовником, до поточного замовлення.

- sendOrder (Order) – надсилає замовлення на кухню, яке потрібно зробити.

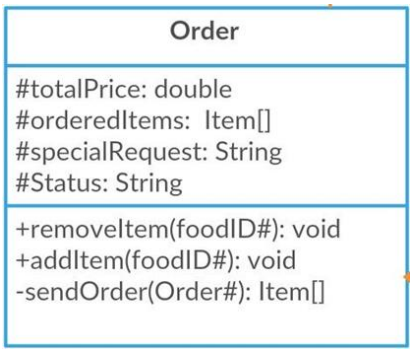

Рисунок 3.2 – Клас Order

3. Клас: Item.

Атрибути:

- itemName: String – назва елемента.
- itemPrice: double – ціна кожного товару.
- ingredients: list of String – список різних інгредієнтів, необхідних для виготовлення предмета.
- description: String – кожен предмет описується кількома короткими реченнями.
- category: String – String для відстеження, до якої категорії може бути включений предмет.

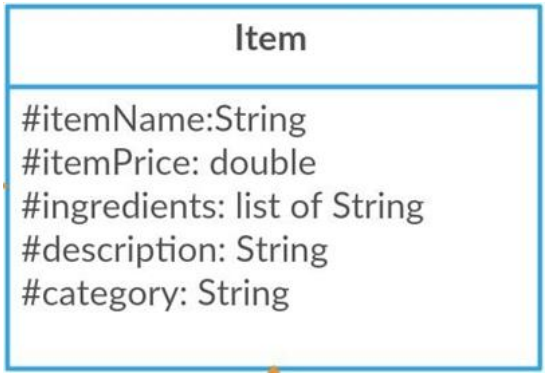

Рисунок 3.3 – Клас Item

4. Клас: TableLayout.

Атрибути:

- id: int – ідентифікатор столику.

- status: int – поточний стан / статус столику – доступний «зелений», зайнятий «червоний», а «кораловий» – брудний.

Методи:

- selectTable () – клієнт вибирає столик.
- changeStatus () – статус столику зміниться після вибору.
- conjoin () – клієнт може вибрати два столики, щоб об’єднати їх.
- getAll () – база даних оновлює всі столики в ресторані.

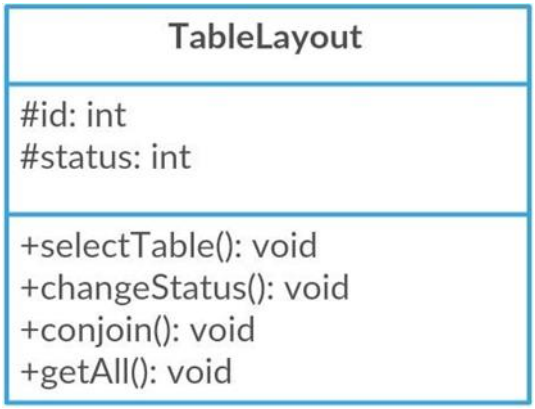

Рисунок 3.4 – Клас TableLayout

5. Клас: User.

Атрибути:

- userName: String – ым'я користувача користувача.
- firstName: String – ім'я користувача.
- lastName: String – прізвище користувача.
- password: String – пароль користувача.
- rewardPoints: int – скільки балів нагород має користувач.
- rewardBalance: double – сума винагород, яку має користувач (1 бал – це не 1 винагорода)
- accountType: int – тип облікового запису користувача.
- Email: String – електронна адреса користувача.
- Status: Int – цілочисельне значення для відстеження того, чи працює співробітник чи ні.

- minutesIn: Double – записує кількість хвилин, проведених користувачем
- address: String – адреса користувача.
- totalHoursWorked: int – загальна кількість годин, відпрацьованих користувачем, якщо користувач має тип облікового запису працівника.

Рисунок 3.5 – Клас User

Методи:

- addPoints () – додайте ціле число балів винагороди на баланс користувача.
- removePoints () – видаліть ціле число бонусних балів на баланс користувача.
- getPoints () – отримайте баланс балів користувача.
- notifyUser () – якщо користувач має активний сеанс, надішліть йому повідомлення.

6. Клас: Employee Portal – цей клас створений, щоб дозволити співробітникам здійснювати моніторинг робочої зміни (відпрацьованих годин за зміну, враховуючи перерви).

Атрибути:

- username: String – це власне ім’я користувача окремого працівника, щоб мати змогу мати обліковий запис.
- password: String – це саморобний пароль окремого працівника для захисту його профілю.
- firstName: String – це ім’я окремого працівника.
- lastName: String – це прізвище працівника.
- hoursWorked: int – це дозволить відстежувати кількість часу, який окремий працівник пропрацював протягом поточної зміни.
- timeStamp: Date – The – дата початку/кінця зміни.
- workerID: String – це ідентифікатор працівника, щоб система та менеджер могли ідентифікувати роль працівника та мати можливість редагувати його права та відстежувати його.
- timeClockedIn: String – працівник має можливість зафіксувати час, коли починається зміна, реєструється час.
- timeClockedOut: String – працівник має можливість закінчити свою зміну, закінчуючи свою зміну, час записується.
- ipClockedIn: String – співробітник вмикається, IP реєструється, тому працівник не може користуватися комп’ютером, що не належить компанії, або входити з дому.
- ipClockedOut: String – працівник закінчується, IP реєструється, тому працівник не може користуватися комп’ютером, що не належить компанії, або входити з дому.
- latitudeClockedIn: String – співробітник вмикається, широта реєструється, тому працівник не може користуватися комп’ютером, що не належить компанії, або входити з дому.

- latitudeClockedOut: String – співробітник закінчує синхронізацію, широта записується, тому працівник не може користуватися комп'ютером, що не належить компанії, або входити з дому.
- longitudeClockedIn: String – співробітник вмикається, довгота записується, тому працівник не може увійти з дому.
- longitudeClockedOut: String – співробітник закінчує синхронізацію, довгота записується, тому працівник не може користуватися комп'ютером, що не належить компанії, або входити з дому.

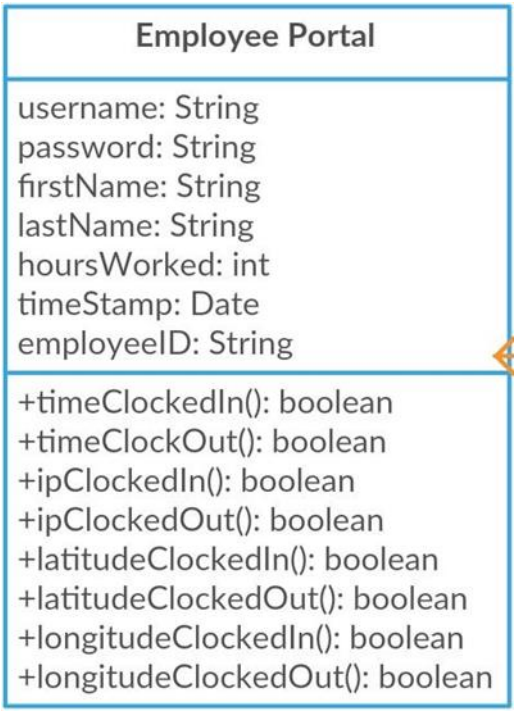

Рисунок 3.6 – Клас Employee Portal

Методи:

- clockedIn (): boolean – працівник має можливість зафіксувати час початку своєї зміни.
- clockedOut (): boolean – працівник має можливість закінчити свою зміну.

7. Клас: Employee Info – клас призначений для зберігання інформації про працівника, якщо працівник бажає отримати до неї доступ.

Атрибути наведені на наступній сторінці.

- firstname: String – ім'я працівника.
- last name: String – прізвище працівника.
- stripe_id: int – унікальний ідентифікатор працівника.
- email: String – електронна пошта працівника.
- totalHoursWorked: int – дозволить відстежувати загальну кількість часу, який окремий працівник пропрацював протягом платіжного циклу.
- hourlyWage: double – буде відстежувати щогодинну заробітну плату окремого працівника.

Методи:

- getEfficiee () – відображатиме інформацію про працівника.

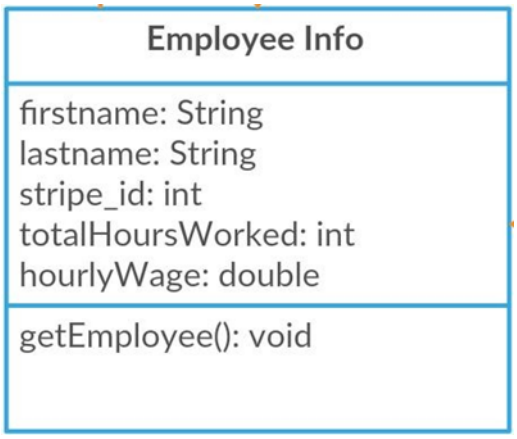

Рисунок 3.7 – Клас Employee Info

8. Клас: Charge – Цей клас створений, щоб допомогти контролювати інформацію, пов’язану з транзакціями, що відбувається.

Атрибути:

- Stripe_id: String – унікальний ідентифікатор оплати.
- amount: int – сума, що нараховується.
- created: date – час і дата, коли була здійснена транзакція.
- currency: String, – тип використовуваної валюти.
- failure_code: String- код, пов'язаний з причиною відмови.
- failure_message: String, – повідомлення, що відображається, коли виникає помилка.

- outcome: String, – результат транзакції.
- paid: Boolean – використовується для визначення, оплачено замовлення чи ні.
- receipt_email: String, – електронний лист, який буде надіслано на пошту.
- receipt_number: String, – унікальний номер квитанції.
- receipt_url: String, – URL-адреса, пов'язана з квитанцією.
- status: String, – пов'язаний зі статусом оплати.

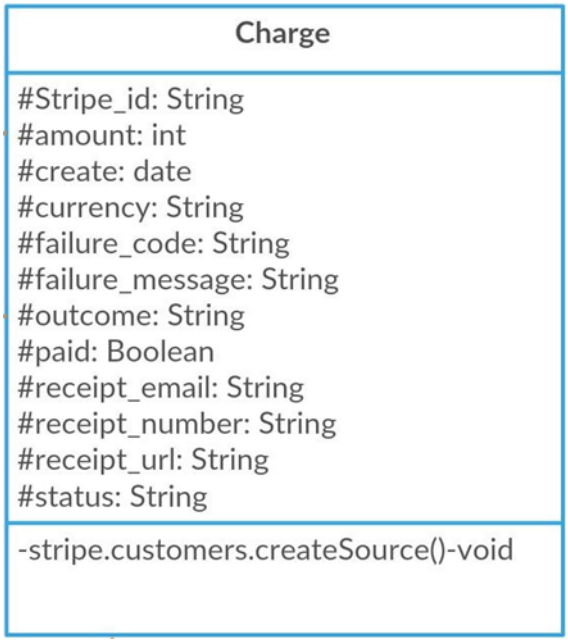

Рисунок 3.8 – Клас Charge

Методи:

- stripe.customers.createSource () – створює новий Source та визначає всі пов’язані змінні.

9. Клас: Source – цей клас буде відстежувати всі деталі щодо способу оплати, якщо це було зроблено готівкою або карткою, а якщо транзакція виконана карткою, інформація про картку буде збережена.

Атрибути:

- firstname: String, – ім'я особи, з якої висунуто обвинувачення.
- lastname: String, – прізвище особи, яка оплачує рахунок.

- stripe_id: {type: Sequelize.STRING, primaryKey: true}, – унікальний ідентифікатор операції оплати.
- address_city: String, – місто адреси виставлення рахунку.
- address_country: String, – країна адреси виставлення рахунку.
- address_line1: String, – адреса виставлення рахунку.
- address_line2: String, – альтернативна адреса адреси виставлення рахунку.
- address_state: String, – стан адреси виставлення рахунку.
- address_zip: String, – поштовий індекс адреси виставлення рахунку.
- cvc_check: String, – перевірте правильність коду безпеки платіжної картки.
- exp_month: String, – місяць закінчення терміну дії платіжної картки.
- exp_year: String, – місяць закінчення терміну дії платіжної картки.
- last4: String, – останні 4 цифри платіжної картки.

Метод:

- Charge.create () – створює нову оплату для столику та визначає всі пов’язані зміни.

Діаграма класів наведена у Додатку Є.

Рисунок 3.9 – Клас Source

3.3. Опис інтерфейсу додатку

3.3.1. Інтерфейс користувача

Після того, як клієнти запускають додаток, їх зустрічає заставковий екран, а потім домашня сторінка. Звідти вони можуть безкоштовно створити новий обліковий запис, увійти до свого наявного облікового запису або продовжити користування додатком як гість.

Увійшовши в систему, вони можуть вибрати «Пообідати у ресторані» або «Замовити на виніс». Замовлення на виніс негайно направляє їх до меню, тоді як обід у ресторані дозволяє клієнту вибрати доступний столик на свій вибір.

Потім клієнт може переглядати меню та додавати до замовлення все, що йому подобається. Увійшовши в систему, вони можуть торкнутися значка дзвоника вгорі праворуч, щоб надіслати сповіщення до офіціанта для того, аби він виконав все, що клієнти можуть запросити.

Зробивши остаточний вибір, вони можуть переглянути своє замовлення та оплатити його обраним способом. Коли транзакція квитанції пройшла успішно, квитанція може бути надіслана на електронну пошту.

3.3.2. Інтерфейс адміністратора (менеджера)

Після того, як менеджер входить в систему через екран входу, про який згадувалося раніше, перед ним з’являється екран із доступом до кількох функцій.

Перший варіант показує їм поточний стан усіх столиків у ресторані. Друга кнопка змушує програму надсилати менеджеру електронний лист із інформацією про заробітну плату. Третя кнопка переводить менеджера до інтерфейсу, де вони можуть переглядати всіх поточних співробітників, де кожного можна вибрати окремо для перегляду інформації про співробітників. Наступна кнопка дозволяє менеджеру переглядати меню як зазвичай.

Черга замовлень дозволяє менеджеру бачити стан усіх поточних замовлень. Функції редагування меню дозволяють менеджеру додавати позиції страв до меню, а також редагувати та видаляти наявні. Додавання позиції

просто потребує, щоб вони заповнили необхідну інформацію. Редагування елемента дозволяє менеджеру вибрати елемент із меню та відредагувати всі властивості вручну.

Видалення елемента запитує назву елемента та видаляє елемент із меню. Заключна кнопка підводить менеджера до інтерфейсу, який дозволяє їм заходити в або з зміни. Поряд з цим, клацнувши на правому верхньому значку дзвоника, менеджери можуть бачити всі поточні пінги як від клієнтів, так і від колег, а також надаючи їм можливість самостійно відправляти пінг.

3.3.3. Інтерфейс персоналу

Інтерфейс співробітника схожий на інтерфейс менеджера, однак функцій менше, оскільки вони не мають управлінського доступу. Кнопка «Перегляд столиків» дозволяє їм переглянути всі доступні столи в ресторані.

Параметр «Переглянути профіль» дозволяє їм переглядати власну інформацію про співробітників. Черга замовлень представляє працівникові чергу, що показує стан усіх існуючих замовлень у ресторані. За допомогою цього інтерфейсу працівники можуть отримати доступ до меню ресторанів, якщо вони хочуть замовити собі їжу.

Фінальна кнопка дозволяє їм розпочати або закінчити свою робочу зміну. Натискання значка дзвоника у верхньому правому куті інтерфейсу дозволяє працівнику переглядати всі існуючі пінг-повідомлення /сповіщення від клієнтів та колег, а також мати можливість самостійно надсилати пінг-повідомлення.

Висновки до розділу

В останньому розділі – технічне забезпечення, розроблені алгоритми, на основі яких працює додаток, для кращого розуміння роботи системи і полегшення розробки була створена use-case діаграма з описом роботи системи за участі акторів.

Розглянуті основні операції, що будуть здійснюватися користувачами системи, для кожної з них була створена діаграма послідовностей. Детально розписана структура додатку, описані усі класи системи.

В кінці розділу описаний інтерфейс для кожного з типу користувачів системи – клієнта, адміністратора, рядового співробітника ресторанного закладу.

РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

4.1. Розробка стартап-проекту «RestInst»

Базовою стратегією стартапу є задоволення спільних інтересів шляхом формування прийнятної пропозиції з обопільною корисністю. Однак, створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка.

У даному розділі наведена розробка проектної пропозиції для проекту, що було створено у рамках магістерської дисертації. Очевидно, що результати даної дисертаційної роботи є важливими з соціальної та економічної точок зору. Дана робота є інноваційною та потрібною суспільству. У загальному, суттю роботи є підвищення ефективності роботи ресторану, шляхом автоматизації фронт-офісу закладу.

4.2. Опис ідеї

Необхідно розробити програмну систему, яка б усунула необхідність традиційного «паперового» підходу у роботі ресторанів. Проект орієнтований на те, аби ресторан був повністю автоматизований для того, щоб було простіше координувати різні робочі процеси, які відбуваються всередині ресторанного закладу.

Проблеми примітивної «паперової» системи:

- Відстеження вільних столиків вимагає постійного контролю адміністратора, а це додатковий час в обслуговуванні – оскільки перевірити вільні чи доступні столики можна лише підійшовши до нього.
- Офіціант записує замовлення на папері і повинен передати отриману інформацію в термінальну систему. Це займає певний час і знижує ефективність роботи закладу, особливо у часи великої завантаженості. Також у ресторані може бути недостатньо

терміналів, оскільки існує певна кількість столиків, офіціантам необхідно чекати, поки інші закінчать вводити свої замовлення до системи.

- Кухари не мають змоги повідомити офіціантів, що страва для столику приготована, окрім як викрику: «Сервіс!». Іноді навіть цього шеф-кухарі не роблять і замовлення залишаються очікувати на стойці видачі.
- Ведення рахунків та іншої статистичної інформації потребує багато часу.

Предметом проекту є розробка додатку, який представляє собою систему автоматизації ресторанного комплексу. Додаток написаний за допомогою фреймворку React Native, для зручного користування за допомогою смартфонів чи планшетів.

Основними задачами проекту є розробка додатку, що дозволить клієнтам ресторану та його співробітникам зменшити час на обслуговування, мінімізувати рутинні задачі та допомогти кожному працівнику ресторану відслідковувати свої робочі зміни, заробітну плату та чайові.

Основною ідеєю проекту є ефективності роботи ресторану, шляхом автоматизації роботи фронт-офісу закладу.

У таблиці 4.1 зображено зміст ідеї та можливі базові потенційні ринки, в межах яких потрібно шукати групи потенційних клієнтів.

Таблиця 4.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Додаток для автоматизації фронт-офісу ресторанного закладу.	Використання для гастрономічних закладів.	Підвищення ефективності роботи ресторанного закладу. Точні данні щодо змін працівників, їх заробітної плати, чайових. Мінімізація рутинних задач співробітників. Зменшення часу обслуговування клієнтів.

Отже, пропонується додаток, що являє собою систему для автоматизації фронт-офісу ресторанного закладу, яка підвищить ефективність його роботи, буде надавати інформацію щодо їх робочих змін, заробітної плати, чайових. Додаток має мінімізувати рутинні задачі та зменшити обслуговування клієнтів.

Проведемо аналіз потенційних техніко-економічних переваг ідеї порівняно із пропозиціями конкурентів. Для цього необхідно визначити перелік техніко-економічних властивостей та характеристик ідеї; попередній перелік конкурентів (проектів-конкурентів) або товарів замінників чи товарів-аналогів, що вже існують на ринку, та провести збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку і провести порівняльний аналіз показників: для власної ідеї визначено показники, що мають гірші значення, аналогічні значення та кращі значення (табл. 4.2).

Таблиця 4.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ н/п	Техніко-Економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона	S (сильна сторона)
		Мій проект	R-Keeper	iiko			
1	Вартість додатку (у грн.)	5000	30000	17600	-	-	+
2	Складність технічної реалізації	Відсутня	Наявна	Наявна	-	+	-
3	Точність роботи алгоритмів системи	Висока	Висока	Висока	-	-	+
4	Додаткова автоматизація бек-офісу	Ні	Так	Так	-	+	-
5	Синхронізація з існуючими автоматизованими системами	Ні	Так	Так	-	+	-

Отже, можна зробити висновок, що головними перевагами для користувача є вартість додатку та точність роботи алгоритмів системи. Нейтральними сторонами є складність технічної реалізації, додаткова автоматизація бек-офісу та синхронізація з існуючими автоматизованими системами у ресторані. Складність технічної реалізації, а інакшими словами майже її відсутність, одна з основних переваг серед аналогів, оскільки чим легша схема пристрою, тим менше буде ціна додатку.

Проте, можна помітити, що у додатку окрім сильних та нейтральних сторін є й слабкі сторони, а саме: додаткова автоматизація бек-офісу та синхронізація з автоматизованими системами, які вже працюють у ресторані. У конкурентів система поділена на блоки автоматизації, кожен з яких можна додатково придбати, в залежності від потреб закладу, в тому числі і автоматизацію бек-офісу.

4.3. Технологічний аудит ідеї проекту

Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (табл. 4.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту?
- чи існують такі технології, чи їх потрібно розробити/доробити?
- чи доступні такі технології авторам проекту?

В таблиці 4.3 проводиться аудит технології, за допомогою якої можна реалізувати ідею проекту.

Таблиця 4.3 – Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Додаток для автоматизації фронт-офісу ресторанного закладу	Розробка алгоритму автоматизації роботи, використовуючи планшети та смартфони	Існуюча технологія	Доступна технологія

Закінчення таблиці 4.3

2	Додаток для автоматизації фронт-офісу ресторанного закладу	Розробка алгоритму автоматизації роботи, шляхом симбіозу паперового методу та додатку	Існуюча технологія	Доступна технологія
3	Додаток для автоматизації фронт-офісу ресторанного закладу	Розробка алгоритму автоматизації роботи, використовуючи спеціальні планшети та власні технологічне обладнання	Існуюча технологія	Доступна технологія
Обрана технологія реалізації ідеї проекту: використовуючи планшети та смартфони				

Проаналізувавши таблицю 4.3. було обрано пункт під номером 1, оскільки використання будь-яких планшетів та смартфонів на базі операційної системи iOS чи Android беззаперечно ефективніше симбіозного методу (пункт 2), а також дешевше закупівлі спеціального обладнання для автоматизація (пункт 3).

4.4. Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, що можуть перешкодити реалізації проекту, дозволяють спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів конкурентів.

Рентабельність – поняття, що характеризує економічну ефективність виробництва, за якої за рахунок грошової виручки від реалізації продукції

(робіт, послуг) повністю відшкодовує витрати на її виробництво й одержується прибуток як головне джерело розширеного відтворення [18].

Для кількісного виміру рентабельності в цілому по аграрних підприємствах використовують такі три традиційні показники: рівень рентабельності, норму прибутку і приведену до земельної площі масу прибутку. Рівень рентабельності (R) визначається за формулою 4.1:

$$R = \frac{П}{Вв} \cdot 100\% \quad (4.1)$$

де П — валовий прибуток від реалізації (робіт, послуг); Вв — виробничі витрати на реалізовану продукцію (її виробнича собівартість).

Для повнішої уяви про реальну ефективність певного виду товарної продукції доцільно цей показник обчислювати з врахуванням витрат на її збут, зменшивши при цьому валовий прибуток на величину цих витрат і водночас збільшивши на них знаменник формули .

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

Проведемо аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 4.4).

Таблиця 4.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	6
2	Загальний обсяг продаж, грн/ум.од	5000-10000
3	Динаміка ринку (якісна оцінка)	Розвиток
4	Наявність обмежень для входу (вказати характер обмежень)	Висока конкуренція

Закінчення таблиці 4.4

5	Специфічні вимоги до стандартизації та сертифікації	Відповідність стандартам ГОСТ 14254-96, ГОСТ 12.1.019-79, ГОСТ 26104-89, ГОСТ 12.1.004-9 та ГОСТ 12.2.007.0-75.
6	Середня норма рентабельності в галузі (або по ринку), %	32%

Провівши аналіз попиту, можна зробити висновок, що коефіцієнт рентабельності достатньо високий. Оскільки, середня норма рентабельності складає більш ніж 30%, що майже вдвічі вигідніше за депозити в банку, а це в свою чергу, найбільше приваблює інвесторів.

Далі визначаємо потенційні групи клієнтів, їх характеристики, та формуємо орієнтовний перелік вимог до товару для кожної групи (табл. 4.5).

Таблиця 4.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Збільшення прибутку ресторану	Власники ресторанів, адміністратори	Перевага надається новим алгоритмам з покращеними характеристиками	Низька ціна та висока точність кінцевого продукту
2	Зменшення часу обслуговування клієнтів	Власники ресторанів, персонал фронт-офісу	Перевага надається локальним системам	Низька ціна та висока точність кінцевого продукту
3	Мінімізація повторюваних задач персоналу ресторанного закладу	Власники ресторанів, адміністратори	Перевага надається системам, які автоматизують широкий спектр задач	Низька ціна та висока точність кінцевого продукту

Потреби, що формують ринок є збільшення прибутку ресторану, зменшення часу обслуговування клієнтів, мінімізація повторюваних задач персоналу ресторанного закладу. При цьому потенційні клієнти потребують нижчої ціни пристрою та високої точності роботи

Ринкові можливості – це сприятливі обставини, які підприємство може використовувати для отримання переваг. Як приклад ринкових можливостей можна привести погіршення позицій конкурентів, різке зростання попиту, появу нових технологій виробництва продукції, зростання рівня доходів населення і т. п. Слід зазначити, що можливостями з погляду SWOT-аналізу є не всі можливості, які існують на ринку, а тільки ті, які можна використовувати [19].

Проведемо аналіз факторів ринкового середовища, що сприяють ринковому впровадженню (табл. 4.6) проекту, та факторів, що йому перешкоджають (табл. 4.7).

Таблиця 4.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Потреба в розробці ефективніших алгоритмів автоматизації	Знаходження або переманювання кваліфікованих кадрів для написання алгоритму	Підвищення робітникам заробітної плати для утримання їх на робочих місцях
2	Потреба в крос-платформленості пристрою	Програма не повинна працювати тільки на одному програмному забезпеченні	Відведення додаткового часу та ресурсів для вирішення цієї проблеми
3	Потреба в сучасних технологіях розробки	Використання сучасних компонентів сприятиме підвищенні ефективності системи в цілому	Співпраця з власниками ресторанного бізнесу, взаємовигода від співпраці
4	Потреба роботи додатку у локальному режимі (за відсутності інтернету)	Програма має працювати у різних режимах для задоволення потреб усіх користувачів додатку для автоматизації ресторанного закладу	Написання програмістами компанії оновлених алгоритмів

Отже, при виведені проект на ринок необхідно враховувати усі загрози та ризики. Основними ризиками даного проекту може бути потреба у різних режимах роботи та точності роботи алгоритмів, використання сучасних алгоритмів роботи дозволить швидко реагувати на можливі загрози.

Розглянемо можливості даного проекту (табл. 4.7)

Таблиця 4.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Зростання конкурентів	Зростання конкурентноспроможних розробок	Покращення характеристик системи та розробка нових функцій
2	Поява схожих алгоритмів автоматизації гастрономічних закладів	Конкуренти можуть розробити аналогічні або дуже схожі алгоритми роботи	Оформлення заявки на патент для власної розробки
3	Стрімкий розвиток технологій	Розвиток нових сучасних технологій	Перехід на сучасні технології

Разом з розширенням ринку та плином часу фактори можливостей будуть збільшуватись, разом з тим будуть збільшуватись і загрози, тому необхідно швидко реагувати за можливості і загрози.

Проведемо аналіз пропозиції (табл. 4.8) та визначимо загальні риси конкуренції на ринку.

Таблиця 4.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1.Тип конкуренції: Олігополія	На ринку налічується не багато фірм, що розроблюють подібні системи	Необхідно слідкувати за точністю роботи алгоритмів та безперебійною роботою

Закінчення таблиці 4.8

2. За рівнем конкурентної боротьби: національний	Користувачами даної системи можуть бути фірми з усього світі. Роль грає лише наявність локалізації продукту	Необхідно розширювати сегмент користувачів
3. За галузевою ознакою: внутрішньогалузева	Основною галуззю є автоматизація фронт-офісу контроль.	Необхідно розширювати функціональні можливості даної системи
4. Конкуренція за видами товарів: товарно-видова	Спостерігається конкуренція між схожими алгоритмами	Підвищення точності та ефективності алгоритмів
5. За характером конкурентних переваг: цінова	Ціни на дані системи дуже варіюються, за допомогою нового алгоритму збільшується ефективність системи	Необхідність використання нових алгоритмів для ефективнішої системи
6. За інтенсивністю: марочна	Велику роль відіграє репутація компанії, що виготовляє систему	Збільшити просування системи, покращити взаємодію з користувачами

Провівши ступеневий аналіз на ринку, можна зробити висновок, що на ринку достатньо конкурентоспроможних систем, проте як вже було зазначено вище, використання будь-яких планшетів та смартфонів на базі операційної системи iOS чи Android беззаперечно ефективніше симбіозного методу, а також дешевше закупівлі спеціального обладнання для автоматизація.

Далі проведемо більш детальний аналіз умов конкуренції в галузі за моделлю п’яти сил М. Портера (табл. 4.9), на наступній сторінці.

Провівши аналіз конкуренції в галузі за М. Портером видно, що є достатня кількість прямих конкурентів, що гарно зарекомендували себе на ринку, тому необхідно постійно слідкувати за якістю та точністю роботи своєї системи. Також, необхідно постійно покращувати алгоритми роботи та технології.

Таблиця 4.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари замінники
	Навести перелік прямих конкурентів	Визначити бар'єри входження в ринок	Визначити фактори сили постачальників	Визначити фактори сили споживачів	Фактори загроз з боку замінників
Висновки: ринок є достатньо заповнений конкурентоспроможними системами, конкуренція висока, оскільки дані програми добре зарекомендували себе в якості автоматизації ресторану. Завадою до входу на ринок також є значна сума капіталовкладень.	R-Keeper, iiko, Трактирь, iSOK POS STORE, Jowi, Poster POS, СБИС Presto, Quick Resto.	Висока репутація конкурентних фірм; необхідний розмір інвестицій та необхідний час для установки системи та специфічних даних для ресторану.	Зазвичай постачальники не диктують умови співпраці.	Користувачам важлива низька ціна та висока точність роботи системи.	Точніші алгоритми для ефективної автоматизації ресторану, які можуть надати системи-замінники.

На основі аналізу конкуренції в галузі, що наведено в таблиці 4.9, а також із урахуванням характеристик ідеї проекту, які були розглянуті в таблиці 4.2, вимог споживачів до товару (табл. 4.5.) та факторів маркетингового середовища (табл. 4.6, 4.7) визначимо та обґрунтуємо перелік факторів конкурентоспроможності. Аналіз конкурентоспроможності представлено в таблиці 4.10.

Таблиця 4.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Швидкодія роботи алгоритму виявлення радіації та його точність	Використання нових алгоритмів та систем, що підвищують ефективність та точність алгоритму

Закінчення таблиці 4.10

2	Цінова політика	Ціна на систему буде значно нижчою ніж у конкурентів
3	Крос-платформленість	Робота системи не тільки на базі ОС Android, а й iOS.
4	Зворотній зв’язок	Користувачі потребуються консультування як в технічних так і в експлуатаційних питаннях.
5	Репутація	У зв’язку з тим, що система буде використовуватись для ресторанних закладів, то репутація є важливим фактором під час вибору системи

Як можна побачити з таблиці було обґрунтовано основні п’ять факторів конкурентоспроможності, основними з яких стали: цінова політика, швидкодія роботи алгоритму та його точність і репутація компанії.

Проведемо порівняльний аналіз сильних та слабих сторін факторів конкурентоспроможності (табл. 4.11).

Таблиця 4.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з власним пристроєм «RestInst»						
			-3	-2	-1	0	+1	+2	+3
1	Швидкодія роботи алгоритму та його точність	20						+	
2	Цінова політика	16			+				
3	Крос-платформленість	15					+		
4	Зворотній зв’язок	17				+			
5	Репутація	13						+	

З таблиць 4.10 та 4.11 бачимо, що фактори конкурентоспроможності суттєві та мають великий позитивний внесок при впроваджені на ринок пристрою «RestInst». Основною перевагою даного пристрою є висока точність алгоритму та низька ціна.

Далі проведемо SWOT-аналіз стартап-проекту, що наведено в таблиці 4.12.

Таблиця 4.12 – SWOT-аналіз стартап-проекту

Сильні сторони: 1. Гнучка цінова політика. 2. Висока точність та ефективність алгоритму. 3. Автоматизація ресторанів будь-якого розміру.	Слабкі сторони: 1.Низька репутація системи на початку впровадження проекту. 2. Необхідність початкового капіталовкладення (кредит в банку або сторонні інвестиції).
Можливості: 1. Стрімке поширення системи на ринку за рахунок доступності технології. 2. Розробка нових функцій системи.	Загрози: 1. Розробка ефективніших систем для автоматизації ресторану. 2. Мінливість ринку.

Для успішності впровадження стартап-проекту на ринку необхідно враховувати появу ризиків та слабких сторін проекту. Наприклад, для покращення репутації системи необхідно враховувати побажання користувачів та постійно оновлювати алгоритми роботи.

На основі SWOT-аналізу розробимо альтернативи ринкової поведінки для виведення стартап-проекту на ринок та орієнтовний оптимальний час їх ринкової реалізації з огляду на потенційні проекти конкурентів, що можуть бути виведені на ринок.

Визначені альтернативи аналізуються з точки зору строків та ймовірності отримання ресурсів (табл. 4.13).

Таблиця 4.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Створити прототип системи автоматизації фронт-офісу ресторану	Прототип системи не потребує отримання значних ресурсів	20-30 днів

Закінчення таблиці 4.13

2	Використання сучасних алгоритмів у вже існуючих системах	Необхідно зробити детальний аналіз конкурентів, поговорити з технічною підтримкою та дізнатись ефективні нові ідеї, які можна реалізувати у власному додатку	30-40 днів

Краще за все використовувати підхід, що наведено у першому пункті, оскільки час його реалізації до місяця. Використання сучасних алгоритмів у вже існуючих системах не найкращий варіант, оскільки такий підхід потребує багато часу, а також не дає ніяких гарантій того, що конкуренти будуть детально розповідати алгоритми роботи системи, якщо її не купити. Із зазначених альтернатив обираємо стратегію компенсації слабких сторін стартапу наявними ринковими можливостями.

4.5. Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Таблиця 4.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Власники та адміністратори великих ресторанів	Висока	Середній	Інтенсивна	Висока складність
2	Власники та адміністратори середніх та малих ресторанів	Висока	Високий	Інтенсивна	Середня складність
Цільова група: власники та адміністратори середніх та малих ресторанів					

За результатами аналізу потенційних груп споживачів було обрано власників та адміністраторів середніх та малих ресторанів. Цей вибір аргументовано тим, що для даних груп користувачів орієнтовний попит вище для обраної групи, а також середня складність входу у сегмент. Оскільки важливим фактором залишається ціна системи автоматизації, тоді при відповідності ціновим очікуванням користувача та якістю роботи алгоритмів потрапити до даного сегменту буде неважко.

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл. 4.15).

Таблиця 4.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Створення прототипу системи автоматизації ресторанного закладу	Швидка та ефективна робота системи, безкоштовна демо-версія	Можливо зробити аналіз конкурентів, поговорити з технічною підтримкою та дізнатись ефективні нові ідеї, які можна реалізувати у власному додатку, але не має гарантії, що конкуренти будуть ділитись детальною інформацією не з покупцем	Стратегія спеціалізації

Визначена базова стратегія розвитку проекту – стратегія спеціалізації, оскільки ця стратегія передбачає концентрацію на потребах одного цільового сегменту, без прагнення охопити увесь ринок. Мета тут полягає в задоволенні потреб вибраного цільового сегменту краще, ніж конкуренти. Така стратегія може спиратися як на диференціацію, так і на лідерство по витратах, або і на те, і на інше, але тільки у рамках цільового сегменту. Тоді, точність роботи системи дає можливість встановлювати вищу ціну на продукцію, так як споживачі готові їх сприйняти.

Наступним кроком є вибір стратегії конкурентної поведінки (табл. 4.16).

Таблиця 4.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Ні	Компанія буде шукати нових споживачів	Ні, плануються власні інноваційні розробки	Наступальна стратегія

На основі проведеного аналізу для вибору стратегії конкурентної поведінки була обрана наступна стратегія — наступальна стратегія. Наступальна стратегія припускає збільшення своєї частки ринку.

При цьому переслідувана мета полягає в подальшому підвищенні прибутковості роботи компанії на ринку за рахунок максимального використання ефекту масштабу.

Наступальна стратегія припускає активну інноваційну політику компанії. Вона постійно атакує власні ж досягнення, збільшуючи розрив між собою і основними конкурентами.

На основі вимог споживачів з обраних сегментів до постачальника та до продукту, а також в залежності від обраної базової стратегії розвитку та стратегії конкурентної поведінки розробимо стратегію позиціонування (табл. 4.17), наведена на наступній сторінці.

Отже, окрім високої точності роботи алгоритму та низької ціни користувач також потребує легкості в обслуговуванні, простоти в використанні та постійного оновлення функцій продукту.

Таблиця 4.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1	Постійне вдосконалення системи враховуючи побажання споживачів	Стратегія спеціалізації	Висока точність роботи системи та формування прихильності користувачів	Зворотній зв'язок із виробником, технічна підтримка, якість та точність
2	Обслуговування	Стратегія спеціалізації	Легке обслуговування	Простота в використанні, точність роботи та швидкодія алгоритму
3	Якість	Стратегія спеціалізації	Швидкість, гнучкість	Ціна, якість

4.6. Розроблення маркетингової програми стартап-проекту

Під час розробки маркетингової програми першим кроком є розробка маркетингової концепції товару, який отримає споживач. У таблиці 4.18 підсумуємо результати аналізу конкурентоспроможності товару.

Очевидними вигодами даного товару є стабільність роботи системи, продуманість системи та ефективні робочі алгоритми, що пришвидшують роботу ресторанного закладу та роблять його роботу точною.

Таблиця 4.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Точність роботи системи	Продуманість системи та ефективні робочі алгоритми, що пришвидшують роботу ресторанного закладу та роблять його роботу точною	Можливість підвищити точність роботи алгоритму
2	Відмовостійкість	Стабільність роботи системи	Висока стабільність системи роботи системи за рахунок використання систем штучного інтелекту

В подальшому, можливі зміни у режимах роботи системи та її функціоналу – можливо автоматизувати на додаток до фронт-офісу також бек-офіс.

Наступним кроком є розробка трирівневої маркетингової моделі товару, де уточнюється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (табл. 4.19).

Таблиця 4.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові	
I. Товар за задумом	Додаток для мобільних пристроїв та планшетів (на базі операційної системи iOS чи Android), написаний за допомогою фреймворку React Native, з використанням GPS функції	
II. Товар у реальному виконанні	Властивості/характеристики	
	1. Новий підхід до алгоритму	Створення продуманої системи для автоматизації фронт-офісу, яка допоможе не тільки співробітникам, а й клієнтам закладу
	2. Простота технічної реалізації	Написання додатку за допомогою фреймворку React Native (кросплатформеність)

Закінчення таблиці 4.19

	3. Додаткове програмне забезпечення	Додавання нових можливостей
	Якість: відповідає нормам ГОСТ 14254-96, ГОСТ 12.1.019-79, ГОСТ 26104-89, ГОСТ 12.1.004-9 та ГОСТ 12.2.007.0-75.	
	Пакування: відсутнє. Система є додатком для планшетів та смартфонів	
ІІІ. Товар із підкріпленням	До продажу з введенням у роботу	
	Після продажу – технічна підтримка, гарантійне обслуговування	
Потенційний товар буде захищено від копіювання за рахунок логотипу та патент на розроблений алгоритм		

Далі визначимо цінові межі, якими необхідно керуватися при встановленні ціни на потенційний товар, це передбачає аналіз цін товарів конкурентів, та доходів споживачів продукту (табл. 4.20).

Таблиця 4.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники (грн)	Рівень цін на товари-аналоги(грн)	Рівень доходів цільової групи споживачів (грн/міс)	Верхня та нижня межі встановлення ціни на товар/послугу (грн)
1	10000-20000	17000-3000	800000	7000-15000

Якщо збільшувати кількість користувачів, то можна трохи понизити ціну товару. В будь-якому випадку ціна буде меншою, ніж у товарів аналогів.

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (табл. 4.21):

Таблиця 4.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має Виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Попереднє замовлення з підписанням контракту	Інтеграція в ресторанну роботу, налагодження коректної роботи на всіх пристроях	Пряма	Пряма
2	Оптові замовлення			

У зв’язку з тим, що обслуговується специфічний сегмент ринку з обов’язковою подальшою технічною підтримкою, доцільніше використовувати прямий канал збуту. Використання прямого каналу збуту дає можливість більш якісно контролювати ціни на ринку та не тратити доходи на оплату роботи посередників.

Останньою складової маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (табл. 4.22).

Таблиця 4.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Огляд продукту, перспективи на майбутнє	Форма зворотного зв’язку в мобільному додатку	Встановлення мети позиціювання. Розробка стратегії позиціювання. Розробка комплексу маркетингу. Оцінка ефективності позиціювання.	Розповсюдження інформації про продукт	Підкреслення переваг продукту

Висновок

Обрано ключові позиції: маркетингові дослідження, встановлення мети позиціювання, розробка стратегії позиціювання, розробка тактики позиціювання, розробка комплексу маркетингу, оцінка ефективності позиціювання. Розглянуто завдання рекламного повідомлення та концепцію рекламного звернення.

ВИСНОВКИ

У рамках дипломного проекту поставлена задача для написання додатку, який допоможе ресторанним комплексам підвищити ефективність їх роботи. Визначені переваги та недоліки неавтоматизованих гастрономічних закладів, знайдені проблеми, що виникають під час обслуговування гостей, та шляхи їх вирішення.

Розглянуті основні бізнес-процеси ресторанного закладу, визначені ролі та задачі співробітників, поставлені задачі, які має виконувати додаток, згідно з робочими обов'язками співробітників ресторану. Проведений аналіз існуючих програм для автоматизації фронт-офісу та означені технічні вимоги до додатку.

Вибрана та детально описана трьох шарова архітектура розроблюваного додатку. Описані технології, за допомогою яких додаток написаний – React Native, визначене основне сховище даних – MySQL. Детально описана архітектура серверного додатку, а також описана математична модель проекту, визначений мережевий протокол.

Розроблені алгоритми, за якими працює створений додаток, детально описана діаграма класів, описаний інтерфейс додатку. Розроблений додаток був протестований для різних задач, які можуть виникнути у користувачів будь-якого рівня. Створена система безпомилково справляється із поставленими задачами, які детально описані у розділах вище. Проведено маркетингові дослідження, додаток оформлений як стартап проект з детальним описом розвитку створеної системи.

Таким чином, можна зробити висновок, що цей додаток для автоматизації фронт-офісу ресторанних комплексів може значно пришвидшити роботу персоналу закладу. Пришвидшення роботи, шляхом нівелювання моментів простою (які були створені через людський фактор та багато зайвих дій – за відсутності актуальної інформації щодо роботи закладу), в свою чергу, підвищить ефективність роботи гастрономічного закладу – допоможе швидше

та якісніше обслуговувати клієнтів. А задоволені клієнти – високий прибуток закладу.

Таким чином, мета дипломного проекту досягнута. Основні переваги такої системи – відсутність надлишкових функцій, а також повністю продумана автоматизація фронт-офісу гастрономічного закладу. Додаток акцентується не лише на задоволенні потреб клієнтів, як способі підвищенні прибутку ресторану, а і на вдосконаленні робочих моментів, с якими працює персонал – шляхом автоматизації рутинних задач.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Плотникова, Н.І. Комплексна автоматизація туристського бізнесу. Ч. II: Інформаційні технології в сфері гостинності: Навчально-методичне посібник.-М .: Радянський спорт, 2001 –208 с.

2. Чудновський А.Д. Жукова М.А. Інформаційні технології управління в туризмі: навчальний посібник.-М.: КНОРУС, 2009 –101 с.

3. Горяєв Ю.А. Інформаційні технології управління. Курс лекцій. – М .: Изд-во «МІЕМП», 2006. – 102 с.

4. А. М. Пирогова, Е. І. Свиридова РЕСТОРАН, КАФЕ закусочних Особливості бухгалтерського обліку та оподаткування

5. Титоренко Г.А. Інформаційні технології управління. Навчальний посібник для вузів. – М .: Изд-во «ЮНИТИ-ДАНА», 2003. – 439 с.

6. Карачаровскій В. ІКТ в ресторанному бізнесі – нагальна потреба або випередження часу [Електронний ресурс]. – Режим доступу: http://tourlib.net/statti_tourism/karacharovskij.htm

7. Чудновський А.Д. Інформаційні технології управління в туризмі. Навчальний посібник. – М .: Изд-во «КНОРУС», 2006. – 104 с.

8. Вендров А.М. Практикум з проектування програмного забезпечення економічних інформаційних систем: Учеб. посібник. – М .: Фінанси і статистика, 2004. – 192 с.

9. Н. Єлманова, С. Трепалін, А. Тенцлер. Delphi і технологія COM – СПБ .: Пітер, 2003 – 698 с.

10. Роберт Вієйра. Програмування баз даних Microsoft SQL Server 2005. Базовий курс = Beginning Microsoft SQL Server 2005 Programming. – М .: «Діалектика», 2007. – С. 832. – ISBN 0-7645-8433-2 Посилання на літру.

11. Роберт Шелдон, Джоффрі Мойї MySQL: базовий курс = Beginning MySQL. – М .: «Діалектика», 2007. – С. 880. – ISBN 0-7645-7950-9.

12. С.В. Маклаков. Створення інформаційних систем з ALLFusion Modelling Suite. М., 2003.

13. С.В. Маклаков. ERwin і Bpwin. CASE-засобу розробки інформаційних систем. М., 1999..

14. Вендров А.М. Один з підходів до вибору засобів проектування баз даних і додатків. «СУБД», 1995, №3.

15. Уенді Боггс, Майкл Боггс. UML і Rational Rose 2002. 2004. – С. 346.

16. Ільїн В.В. Designer Реінжиніринг бізнес-процесів з використанням ARIS Практика реального бізнесу. «Вільямс», 2008. – С. 456.

17. Олексій Ковязин, Сергій Востріков «Світ InterBase. 3-е видання» 2005. С. 496.

18. SWOT Analysis for Your Small Business [Електронний ресурс] – Режим доступу до ресурсу: https://www.wordstream.com/blog/ws/2017/12/20/swot-analysis.

19. How to do market research in 4 steps [Електронний ресурс] – Режим доступу до ресурсу: https://www.hotjar.com/blog/market-research/.

ДОДАТКИ

ДОДАТОК А

Модель додатку

ДОДАТОК Б

UML діаграма акторів і їх ролей

ДОДАТОК В

Діаграма послідовностей – оплата готівкою

ДОДАТОК Г

Діаграма послідовностей – оплата картою

ДОДАТОК Д

Діаграма послідовностей – перегляд меню

ДОДАТОК Е

Діаграма послідовностей – закриття/відкриття зміни

ДОДАТОК Є

Діаграма класів

ДОДАТОК Ж

Результат перевірки роботи на співпадіння