

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ І ПРИКЛАДНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

БОНДАРЧУК ПАВЛО ВОЛОДИМИРОВИЧ

Допускається до захисту:
в.о. завідувача кафедри
інформаційних технологій,
д-р. техн. наук, професор
_____ Наталія ВЕСЕЛОВСЬКА
« ____ » _____ 2026 р.

**ВЕБСЕРВІС З СИСТЕМОЮ ОНЛАЙН-ЗАМОВЛЕНЬ ДЛЯ КАФЕ-
БАРІВ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (бакалаврська) робота

Керівник:

Наталія ВЕСЕЛОВСЬКА, професор кафедри
інформаційних технологій
д.т.н., професор

Оцінка: ____ / ____ / ____
(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця - 2026

АНОТАЦІЯ

Бондарчук П.В.

Вебсервіс з системою онлайн-замовлень для кафе-барів.

Спеціальність 122 «Комп'ютерні науки», освітня програма «Комп'ютерні науки». Донецький національний університет імені Василя Стуса, Вінниця, 2026.

Робота спрямована на розробку інформаційної технології сервісу онлайн-бронювання місць у закладах. Основна мета дослідження полягає у вдосконаленні функціонування програмного додатку шляхом інтеграції сучасних технологічних рішень, що забезпечують підвищення швидкості його завантаження та надійності роботи для кінцевих користувачів.

У процесі дослідження здійснено аналіз існуючих технологій, визначено їхні переваги та обмеження, а також сформовано систематизований перелік вимог до розроблюваної системи. Це дозволило обґрунтувати вибір оптимальних інструментів і методів для створення ефективного та зручного сервісу бронювання.

Ключові слова: інформаційна технологія, бронювання, заклад громадського харчування, SILOS, React, React Native, Java Script.

ABSTRACT

Bondarchuk P.V.

Information technology for booking seats in catering establishments.

Specialty 122 “Computer Science”, educational program “Computer Technologies (Data Science)”. Vasyl' Stus Donetsk National University, Vinnytsia, 2026.

The work is devoted to the creation of information technology for the development of the client part of the restaurant reservation service. The purpose of the master's thesis is to improve the application by using modern technologies to develop the client part, which will increase the download speed of the application and its reliability for customers. The advantages and disadvantages of each of them are analyzed, the list of requirements is created. The client part itself was developed and integrated with the server part.

Keywords: interface, client part, React, React Native, Java Script

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. Аналітичний огляд літературних та інших джерел	8
1.1. Поняття закладів громадського харчування та їх автоматизація ..	8
1.2. Переваги та недоліки бронювання місць	12
1.3. Аналіз існуючих систем бронювання.....	20
РОЗДІЛ 2. Системний аналіз об'єкта дослідження	25
2.1. Дерево цілей.....	25
2.2. Конкретизація функціонування системи	31
2.3. Побудова ієрархії процесів.....	35
РОЗДІЛ 3. Програмні засоби розв'язання задачі.....	37
3.1. Вибір парадигми та мови програмування.....	37
3.2. Вибір програмних засобів.....	38
РОЗДІЛ 4. Практична реалізація	44
4.1. Опис створеного програмного засобу	44
4.2. Інструкція користувача	45
4.3. Аналіз контрольного прикладу	46
ВИСНОВКИ.....	49
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	50
ДОДАТКИ.	52

ВСТУП

Актуальність роботи. Інформаційні технології настільки глибоко інтегрувалися у сучасне життя, що вже майже неможливо знайти сферу діяльності, де вони не застосовуються. Використання комп'ютерів та мережі Інтернет дозволяє значно прискорити роботу з документацією, зменшити навантаження на персонал і водночас підвищити продуктивність праці. У закладах громадського харчування це проявляється особливо яскраво: у ресторанах чи кафе офіціанти вводять замовлення безпосередньо у комп'ютер, після чого інформація миттєво передається на кухню, що скорочує час очікування та оптимізує роботу кухарів. Такі системи не лише економлять час, а й роблять процес обслуговування більш зручним і ефективним. Крім того, інформаційні технології внесли елемент інноваційності та креативності у сферу харчування: відвідувачі дедалі частіше обирають заклади, які пропонують безкоштовний Wi-Fi, адже це дає змогу працювати, перевіряти пошту, читати новини чи спілкуватися з близькими під час очікування замовлення. Водночас підприємства громадського харчування стикаються з проблемами, пов'язаними з «людським фактором», зокрема нераціональним розподілом місць для бронювання чи розсаджування гостей, що призводить до втрат прибутку. Автоматизація процесів бронювання здатна вирішити цю проблему: вона забезпечує обслуговування більшої кількості клієнтів, підвищує ефективність зберігання та передачі даних, скорочує час їхньої обробки й тим самим сприяє прийняттю швидших та якісніших управлінських рішень.

Мета роботи: проектування та розробка інформаційної системи для бронювання місць у закладах громадського харчування.

Об'єкт дослідження: дослідження роботи систем бронювання місць у закладах громадського харчування.

Предмет дослідження: методи та засоби для розроблення інформаційної системи бронювання місць у закладах громадського харчування.

Практичне значення отриманих результатів. Створена інформаційна система дозволить підвищити ефективність та продуктивність роботи закладів громадського харчування та допоможе привабити більше відвідувачів.

РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ТА ІНШИХ ДЖЕРЕЛ

1.1. Поняття закладів громадського харчування та їх автоматизація

Підприємства громадського харчування охоплюють цілу систему організацій, що забезпечують населення кулінарною продукцією та послугами харчування, і ця сфера надзвичайно різноманітна. Бар може запропонувати обмежений набір напоїв та закусок, не займаючись складним приготуванням страв, тоді як снєк-бар орієнтований на швидке обслуговування та прості страви на кшталт вареників чи млинців. Кафе забезпечує відпочинок і харчування з менш широким асортиментом, ніж ресторан, а кав'ярня спеціалізується на гарячих напоях, випічці та кондитерських виробах. Заклади швидкого обслуговування працюють у місцях великого потоку людей, пропонуючи прості страви з напівфабрикатів високого ступеня готовності, часто без власного залу, а ресторани вирізняються багатим меню, високим рівнем сервісу та організацією дозвілля. Вони можуть бути відкритими для всіх або закритими, обслуговуючи лише певні групи, а окремі підприємства утворюють мережі під єдиним брендом чи функціональною назвою, як-от «Макдональдс» чи «Мережа шкільних їдальнь». У плануванні їх оцінюють за місткістю та продуктивністю, тобто кількістю місць і страв за зміну.

Сфера харчування дедалі більше інтегрує сучасні технології: веб-сайти, мобільні додатки, електронні меню, системи бронювання та моніторингу, а також бездротовий Інтернет, що забезпечує конкурентні переваги компаніям, які активно їх застосовують. Автоматизація закладів громадського харчування стала стійким терміном у професійному середовищі країн СНД і означає впровадження апаратних та програмних рішень для управління бізнес-процесами, не торкаючись безпосередньо приготування їжі, а охоплюючи бухгалтерію, документообіг, продажі та обробку даних. Попит на такі системи стабільний, адже ресторанний бізнес у цьому регіоні є однією з найдинамічніших сфер приватного підприємництва, і кількість автоматизованих закладів уже вимірюється десятками тисяч.

Основні завдання автоматизації зводяться до підвищення прибутковості, зменшення витрат, контролю та оптимізації роботи, покращення якості обслуговування, запобігання зловживанням персоналу, зростання продуктивності, підтримки маркетингових дій, створення систем лояльності та аналізу діяльності для подальшого розвитку. Вирішення цих завдань є прикладом застосування спеціалізованих систем управління, які відрізняються від універсальних ERP тим, що враховують специфіку ресторанного бізнесу.

Діяльність закладів харчування складніша за роботу роздрібних магазинів, адже тут не лише продають готовий продукт, а й виготовляють його та забезпечують споживання на місці. Автоматизація охоплює продажі, складську діяльність, логістику, бухгалтерію, управління персоналом, бронювання, доставку та інтеграцію з іншими системами, наприклад готельними чи корпоративними. У продажах це означає прийом замовлення, його передачу на кухню, формування рахунку, розрахунки з клієнтами, видачу чека та підтримку маркетингових акцій. У складській діяльності автоматизація забезпечує облік інгредієнтів, напівфабрикатів і товарних запасів, у бухгалтерії — прозорість фінансових операцій. Управління персоналом включає контроль робочого часу, кількості виконаних операцій, систему мотивації, премій і штрафів, а також відеоспостереження. Програми лояльності дозволяють утримувати клієнтів, бронювання забезпечує ефективне планування столиків і банкетів, а доставка охоплює ідентифікацію замовника, прийом замовлення, вибір місця приготування, складання маршруту кур'єра та контроль використання транспорту.

1.2. Переваги та недоліки бронювання місць

Замовлення столика в ресторані фактично є попередньою домовленістю між закладом та клієнтом про резервування місця, і хоча у більшості країн світу ресторани не вимагають обов'язкового бронювання, у закладах високого класу, особливо в переповнених мегаполісах, воно часто є необхідним. У таких випадках столики можуть бути зарезервовані за кілька тижнів наперед, а в

особливо престижних ресторанах неможливо отримати бронювання на той самий день. Сучасна практика бронювання сформувалася як еволюція традиційних методів організації харчування, коли адміністратори користувалися лише блокнотом і ручкою. Сьогодні ж дедалі більше закладів переходять на онлайн-системи, що дозволяють клієнтам самостійно бронювати місця через інтернет-платформи, такі як OpenTable, Resy, Tock, Bookatable, Yelp Reservations, Zomato чи інші, а деякі навіть пропонують можливість перепродажу бронювань.

Разом із тим бронювання на пізніший час може створювати труднощі: накопичення невирішених питань у ресторані призводить до затримок, а популярні страви можуть бути недоступними через брак інгредієнтів. Хоча більшість закладів не стягують плату за невиконане бронювання, етикет вимагає повідомити про відмову від зарезервованого столика. Сучасні системи бронювання не лише спрощують процес резервування, а й інтегрують інструменти для управління даними клієнтів, оптимізації роботи персоналу та маркетингових кампаній.

Онлайн-рішення дозволяють налаштовувати графік роботи, контролювати доступність столиків, додавати чи змінювати замовлення, вести базу клієнтів, надсилати підтвердження та нагадування електронною поштою, а також приймати оплату різними способами — від PayPal і кредитних карток до банківських переказів чи готівки. Система може працювати цілодобово, мінімізуючи кількість телефонних дзвінків і ризик подвійного бронювання, а адміністратори отримують можливість встановлювати власні правила, наприклад максимальну тривалість бронювання чи вимогу попередньої оплати.

Крім того, такі рішення підтримують багатомовність, дозволяють розмежовувати ролі користувачів і надавати доступ працівникам із різними рівнями прав. У результаті бронювання перетворюється на комплексний інструмент управління, що не лише забезпечує комфорт клієнтам, а й допомагає закладу підвищувати ефективність роботи, уникати організаційних проблем, розширювати маркетингові можливості та формувати лояльність відвідувачів.

Резервне копіювання даних.

У Сполучених Штатах практика попереднього резервування столиків у ресторанах стала звичною: приблизно шість із десяти відвідувачів роблять бронювання заздалегідь, причому половина користується онлайн-сервісами. Попит на такі послуги постійно зростає, вони вдосконалюються й знаходять нових прихильників. У країнах СНД ситуація дещо інша — клієнти ще не настільки активно користуються бронюванням, однак тенденція до його поширення вже помітна. Це підтверджується появою та розвитком численних стартапів, які прагнуть змінити традиційний підхід до резервування місць у закладах харчування, пропонуючи сучасні цифрові рішення та нові формати взаємодії між рестораном і відвідувачем. Таким чином, навіть у регіонах, де бронювання не є усталеною практикою, воно поступово набуває популярності й стає важливим елементом організації ресторанного бізнесу.

Види послуг бронювання.

Найбільш відомою у світі платформою для бронювання столиків у ресторанах є американський сервіс OpenTable, який охоплює заклади не лише в США, а й у дев'ятнадцяти інших країнах. Його функціонал загалом подібний до більшості конкурентів, однак на ринку з'явилися й інші моделі, що пропонують нестандартні підходи до резервування.

Одним із таких рішень стали так звані «спекулянти» — сервіси Resy та Table8, які заздалегідь бронюють місця у престижних ресторанах, де столик отримати надзвичайно складно, а потім продають ці бронювання клієнтам. Вартість може становити близько двадцяти доларів, але змінюється залежно від попиту. Інший формат — «портье», як-от сервіс Reserve, що не закріплює конкретний стіл, але забезпечує відвідувачеві доступ до популярного закладу у зручний час. При цьому система надсилає дані клієнта ресторатору та автоматично списує оплату з картки.

Цікавою інновацією стала модель «театру», коли бронювання нагадує купівлю квитка на виставу чи концерт. Клієнт заздалегідь сплачує за стіл і меню, що буде подано, а сервіс Tock, створений шеф-кухарями Чикаго Грантом Акацем

та Ніком Коконасом, поширює цю практику й серед інших закладів. Такий підхід мінімізує ризик, що клієнт не з'явиться, залишивши ресторан у збитках.

Ще один формат — «клубні» сервіси, які допомагають ресторанам залучати нових клієнтів, отримуючи натомість комісію та пропонуючи користувачам бонуси чи знижки. Для закладів це може стати додатковим маркетинговим каналом, хоча вибір таких партнерів потребує обережності.

Усі ці моделі демонструють, що ринок бронювання столиків активно розвивається, а конкуренція між сервісами зростає. Для ресторанів це означає розширення можливостей, адже кожна нова система прагне вдосконалити свої програми та залучити більше закладів, що у підсумку сприяє підвищенню якості обслуговування та зручності для клієнтів.

1.2.1. Переваги для закладу

У сучасних умовах бронювання столиків у ресторанах стало настільки поширеним явищем, що його вже сприймають як невід'ємну складову ресторанного сервісу. Заклади високої кухні активно пропонують клієнтам можливість резервування, адже це забезпечує низку переваг як для відвідувачів, так і для самих рестораторів. Хоча й досі існують ресторани, що працюють за принципом «першим прийшов – першим обслужений», більшість закладів тонкої та навіть повсякденної кухні організовують свою діяльність через систему бронювання. Для багатьох клієнтів це вже стало обов'язковим етапом перед виходом у ресторан, адже послуга гарантує місце, знімає ризик очікування в черзі та створює відчуття впевненості у комфортному відпочинку.

З точки зору закладу, бронювання має як витрати, так і вигоди, проте переваги значно переважають. Резервування столиків дозволяє підтримувати високий попит у години пікового навантаження та стимулювати відвідуваність у менш активні дні, коли клієнти бронюють місця наперед, не знаючи, наскільки переповненим буде заклад. Це також ефективний інструмент у конкурентному середовищі: якщо відвідувач не може отримати бронювання у своєму «першому

виборі», він звертається до іншого ресторану, де така можливість є, і таким чином заклад отримує додаткових клієнтів.

Завдяки бронюванню ресторани можуть працювати більш гнучко — починати обслуговування раніше, завершувати пізніше й охоплювати більшу кількість гостей протягом дня, що безпосередньо впливає на щоденний дохід. Крім того, система резервування дозволяє точніше прогнозувати попит, оптимізувати роботу персоналу, контролювати витрати та підвищувати якість сервісу.

Для рестораторів бронювання є не лише зручністю для клієнтів, а й потужним управлінським інструментом. Сучасні електронні системи зберігають дані про гостей — їхні імена, контакти, кількість і статус бронювань, формують звіти про завантаженість закладу та ефективність роботи. Вони можуть автоматично надсилати повідомлення чи нагадування клієнтам, підтримувати маркетингові кампанії, пропонувати знижки й купони. Доступ до таких систем здійснюється через особистий кабінет, де адміністратори можуть переглядати нові замовлення, обробляти їх онлайн, вести базу даних гостей та аналізувати ефективність послуг.

Таким чином, бронювання столиків у ресторанах перетворилося на комплексну послугу, яка одночасно підвищує комфорт відвідувачів і забезпечує закладам стабільність, ефективність та конкурентні переваги. Це не просто додатковий сервіс, а стратегічний інструмент управління, що дозволяє ресторанам точніше оцінювати попит, краще організовувати роботу персоналу, підвищувати якість обслуговування та збільшувати прибутковість.

1.2.2. Переваги для відвідувача

У сучасному ресторанному бізнесі бронювання столиків стало настільки звичним явищем, що клієнти дедалі рідше ризикують приходити без попереднього резервування. Люди прагнуть мати впевненість у тому, що бажане місце буде доступним саме в той час, коли вони планують відвідати заклад. Це створює відчуття безпеки та комфорту: клієнт знає, що його столик гарантовано

чекатиме, і йому не доведеться витратити час на очікування чи шукати інший ресторан у випадку переповненості.

Попереднє бронювання знімає всі неприємності, пов'язані з чергами чи відмовою у посадці, і водночас забезпечує більш якісне обслуговування. Оскільки заклад заздалегідь знає, коли і з якою кількістю гостей прибуде клієнт, персонал має змогу підготувати відповідний столик, розрахувати кількість місць та організувати роботу так, щоб обслуговування було максимально швидким і комфортним. Це означає, що бронювання не лише гарантує клієнтові спокій та впевненість, а й дозволяє ресторану працювати більш організовано, забезпечуючи відвідувачам вищий рівень сервісу.

Таким чином, резервування столика перетворюється на взаємовигідний інструмент: клієнт отримує гарантію та комфорт, а ресторан — можливість краще планувати роботу і забезпечувати якісне обслуговування, що підвищує його репутацію та конкурентоспроможність.

1.2.3 *Переваги системи онлайн-бронювання*

Тривалий час ресторани здійснювали управління бронюванням у традиційний спосіб: клієнти телефонували адміністратору, а той заносив замовлення до спеціальної книги. Така практика була обмежена робочим часом закладу й вимагала постійної присутності персоналу. З поширенням інтернету та цифрових технологій виникла можливість створення електронних систем резервування, які швидко довели свою ефективність і поступово витіснили паперові журнали.

Онлайн-бронювання забезпечує принципово новий рівень гнучкості: клієнт може здійснити резервування у будь-який час доби та з будь-якого місця, маючи доступ до повної інформації про заклад і його пропозиції. Це перетворює процес на зручну, швидку й безперервно доступну послугу, що підвищує комфорт відвідувачів і формує позитивний досвід взаємодії з рестораном.

Для закладів переваги також очевидні. Використання електронних систем зменшує навантаження на телефонні лінії, дозволяє точніше контролювати заповненість залу та кількість прийнятих замовлень, а також відкриває доступ до розширеної статистики й аналітики. Це сприяє оптимізації бізнес-процесів і прийняттю обґрунтованих управлінських рішень.

Ключовим чинником успіху є багатофункціональність таких систем. Вони включають операційні звіти, історію бронювань, клієнтські бази даних із зазначенням індивідуальних уподобань, що постійно оновлюються з кожним новим замовленням. Ресторани отримують можливість відстежувати скасування, управляти списками очікування, коригувати резерви та здійснювати цільові електронні чи поштові розсилки. У багатьох випадках системи інтегрують інструменти електронного маркетингу, що посилює комунікацію з клієнтами та сприяє формуванню лояльності.

1.2.4. Недоліки

Процес бронювання у закладах громадського харчування та місцях проведення заходів нерідко супроводжується логістичними труднощами. Використання паперових журналів і ручних записів створює ризик помилок, що може призвести до подвійних або некоректних резервувань. Навіть сучасні системи POS та онлайн-платформи не повністю усувають проблему: надмірне бронювання може виникати через неузгодженість дій кількох співробітників.

Постійна повна завантаженість ресторану також має негативний ефект: потенційні клієнти, зіткнувшись із відсутністю вільних місць, можуть відмовитися від спроб бронювання у майбутньому. Додатковим ризиком є так звані «no-show» — випадки, коли гості резервують столик, але не приходять і не скасовують замовлення. Це створює фінансові втрати, призводить до неефективного використання персоналу та зменшує можливості отримання доходу від реальних відвідувачів.

Окрему небезпеку становлять сторонні сервіси бронювання. Хоча деякі з них співпрацюють із ресторанами на партнерських умовах і забезпечують додатковий прибуток, існують моделі, що паразитують на популярності закладів. Наприклад, платформи на кшталт **Killer Rezzy** здійснюють попереднє резервування столиків у затребувані дати, а потім перепродають їх із націнкою. У такій ситуації ресторан не отримує жодної вигоди й навіть може не знати, що клієнт заплатив завищену ціну за бронювання. Це підриває довіру до закладу та створює репутаційні ризики.

1.3. Аналіз існуючих систем бронювання

Оскільки ринок насичений великою кількістю однотипних рішень, постає завдання створення продукту, здатного витримати конкуренцію. Для цього необхідно здійснити систематичний пошук та провести глибокий аналіз наявних платформ бронювання, щоб виявити їхні сильні й слабкі сторони та визначити напрями для розробки унікальних функціональних можливостей.

1.3.1. *SmartReserve*

SmartReserve являє собою комплексне програмне рішення, що функціонує як автоматизована станція управління гостьовим потоком. Система інтегрує всі канали бронювання — онлайн-платформи, телефонні замовлення та локальні точки доступу — в єдину інформаційну мережу.

Завдяки високочастотній синхронізації даних між терміналами користувачі бачать лише реально доступні столики. Якщо на певний час, наприклад о 20:00, місць немає, система автоматично пропонує найближчий вільний інтервал. Такий підхід усуває ризик подвійних бронювань і забезпечує точний контроль заповненості залу, чим принципово відрізняється від традиційних call-центрів.

SmartReserve не лише координує процес резервування, а й автоматизує збір та використання інформації про гостей. Функція персональних позначок у

клієнтських профілях дозволяє ідентифікувати відвідувачів ще до початку розмови, аналізувати їхні дзвінки та враховувати індивідуальні вподобання. Під час бронювання адміністратор отримує доступ до історії взаємодії з клієнтом, його коментарів та особливих побажань, що підвищує якість сервісу та формує лояльність.



Рисунок 1.1 - Приклад системи SmartReserve

Унікальною рисою SmartReserve є інтеграція телефонної ідентифікації з клієнтською базою ресторану: система миттєво отримує номер абонента з мережі та автоматично передає його до CRM, що забезпечує точне розпізнавання гостей і персоналізовану взаємодію.

Ще однією ключовою функцією є автоматичне накопичення статистики бронювань із можливістю багатофакторного онлайн-аналізу за будь-який обраний період. Це дозволяє менеджменту отримувати повну картину завантаженості та ефективності роботи закладу.

Через особистий кабінет SmartReserve адміністратор має постійний доступ до інформації про стан залу й може контролювати всі процеси незалежно від місця перебування та типу пристрою. Такий підхід забезпечує мобільність управління, оперативність прийняття рішень і підвищує рівень сервісу.

1.3.2. R-Keeper

R-Keeper пропонує інтуїтивні інструменти для організації та контролю банкетних замовлень, що значно спрощують роботу адміністратора. Система дозволяє швидко формувати замовлення із зазначенням дати та часу проведення заходу, кількості учасників, а також персональних даних клієнта.

Функціонал включає можливість прийому авансових платежів із видачею податкових чеків, створення детального меню та налаштування часу друку сервісних квитанцій для кухні. Крім того, R-Keeper забезпечує формування договорів та попередніх перевірок у стандартному форматі А4, що робить процес підготовки до банкету максимально прозорим і контрольованим.

Таким чином, система не лише оптимізує управління замовленнями, а й інтегрує фінансові та організаційні аспекти в єдиний робочий процес, підвищуючи ефективність ресторанного бізнесу.



Рисунок 1.2. - Система R-Keeper

Управління банкетами в системі побудоване максимально гнучко: порядок проведення заходу можна змінювати будь-коли, навіть у процесі самого банкету. У день події замовлення автоматично переходить у статус активного, що запускає низку внутрішніх повідомлень. Кухарі отримують сигнал про початок підготовки, а офіціанти — нагадування про ключові організаційні деталі. Система контролює навіть дрібниці, наприклад інформує персонал про необхідність прикрасити стіл квітами, забезпечуючи повну координацію дій та

бездоганний сервіс.

1.3.3. *Iiko*

Від самого запуску **iiko** пропонує сучасний та постійно вдосконалюваний інтерфейс для бронювання столиків і організації банкетів. Процес резервування максимально спрощений: адміністратор обирає потрібний стіл у схемі залу та фіксує побажання гостя, а система автоматично відображає наступний доступний час прямо на макеті. Якщо клієнти прибувають, iiko інформує офіціанта про те, що місце зарезервоване, й блокує можливість відкриття нового замовлення на цей стіл.

Крім того, система оперативно формує перелік продуктів, необхідних для конкретного банкету, що дозволяє кухні отримати рівно ту кількість інгредієнтів, яка потрібна для заходу. Для масштабних подій передбачена функція об'єднання кількох столів у макеті залу одним натисканням, що забезпечує гнучкість у плануванні простору.

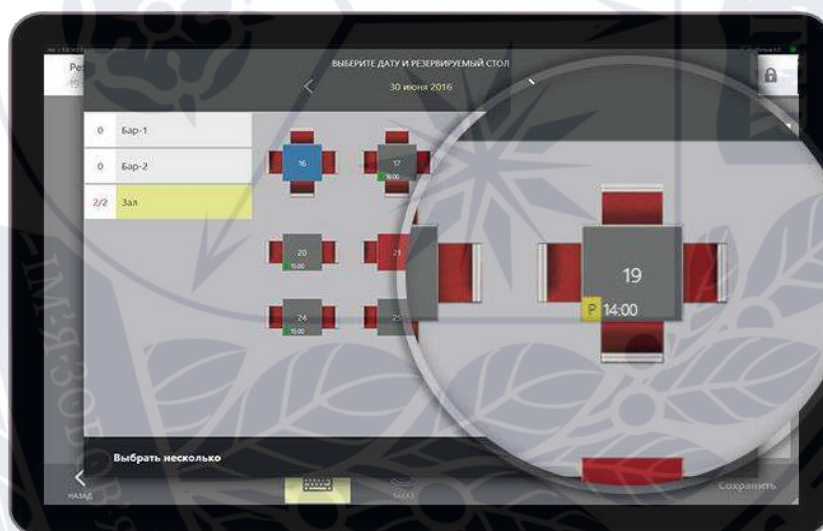


Рисунок 1.3. Система *iiko*

iiko також автоматично надсилає нагадування про майбутні події, допомагаючи персоналу не пропустити жодного етапу підготовки та врахувати всі побажання гостей. Це робить систему потужним інструментом для управління бронюваннями та банкетами, який поєднує простоту використання з високим рівнем організаційної точності.

1.3.4. LeClick

Використання «Електронної книги бронювання LeClick» забезпечує автоматичне формування актуальної клієнтської бази ресторану. Для кожного відвідувача система зберігає історію бронювань, частоту відвідувань та індивідуальні вподобання. Завдяки інтеграції з R-Keerer та телефонною мережею закладу адміністратор може ідентифікувати гостя ще під час дзвінка й звернутися до нього на ім'я, якщо він уже був клієнтом ресторану.

Кожному відвідувачу можна присвоїти статус (наприклад, «важливий» чи «небажаний») та додати коментарі, що робить інформацію доступною для всіх співробітників. База даних також використовується як основа для цільових SMS-та e-mail-розсилок, що сприяє підвищенню лояльності клієнтів, зменшенню відтоку та покращенню якості обслуговування.

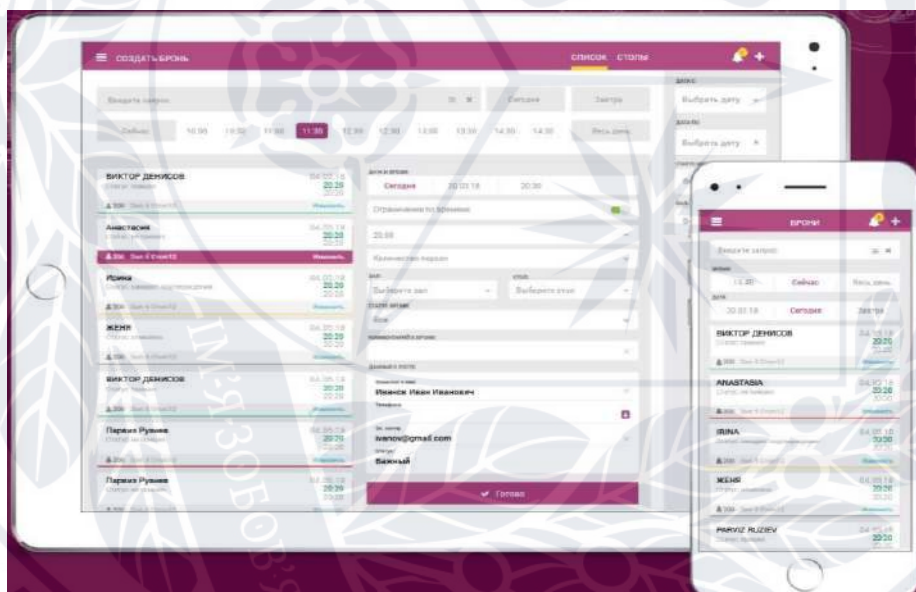


Рисунок 1.4 - Система LeClick

Функціонал LeClick дозволяє генерувати аналітичні звіти про завантаженість закладу у різні дні та години, а також оцінювати ефективність роботи партнерів. Усі бронювання надходять безпосередньо до системи, що знижує навантаження на телефонну лінію. Менеджер може приймати або відхиляти онлайн-запити, а клієнт отримує підтвердження у вигляді SMS. При цьому адміністратор має можливість надсилати не лише стандартні

повідомлення, а й додавати власні коментарі до тексту, що робить комунікацію більш персоналізованою.

Висновок до першого розділу

На основі вищесказаного можна зробити висновок, що система бронювання місць у закладах громадського харчування є дуже необхідною, тому що вони допомагають людям зарезервувати місце у ресторані або барі, та мати впевненість, що у вибраному ними закладі, у той час на який вони замовляли бронювання, місце буде вільне і вони зможуть гарно провести час, а також полегшують для робітників закладу роботу, тому що вони будуть завжди знати скільки відвідувачів буде у конкретний день та конкретну годину. У зв'язку з тим, що закладів громадського харчування стає тільки більше і попит на них теж збільшується, система бронювання стане невід'ємною частиною для ведення справи, й для того, аби збільшити потік клієнтів, а, відповідно, збільшити доходи закладу.

РОЗДІЛ 2. СИСТЕМНИЙ АНАЛІЗ ОБ'ЄКТА ДОСЛІДЖЕННЯ

2.1. Дерево цілей

Системний аналіз являє собою комплекс методів упорядкованого дослідження, спрямованих на пошук оптимальних шляхів вирішення складних і недостатньо структурованих проблем [12]. Його основна ідея полягає у розгляді проблеми як цілісної системи взаємопов'язаних завдань, для яких застосовуються спеціальні методи вирішення. Після цього здійснюється відбір можливих рішень та їх оцінювання за визначеними критеріями з метою досягнення поставленої цілі.

Для аналізу системи використовується дерево цілей — інструмент, що дозволяє визначити ключові елементи, необхідні для досягнення результату. Воно починається з головної мети, яка розташована на вершині, і поступово деталізується через критичні фактори успіху та необхідні умови. Усі рівні дерева мають логічний взаємозв'язок, що забезпечує цілісність структури [13].

У випадку інформаційної системи бронювання місць головною метою є створення ефективної системи управління резервуванням. Перший рівень декомпозиції включає такі підцілі:

1. Збір та аналіз даних — проведення аналітичного дослідження предметної області.
2. Планування та проектування — формування системних вимог і визначення архітектури на основі проведеного аналізу.
3. Розробка системи — реалізація програмного продукту, що забезпечує виконання поставлених завдань.

Таким чином, системний аналіз виступає методологічною основою для створення інформаційної системи бронювання, дозволяючи структурувати процес розробки та забезпечити досягнення кінцевої мети.



Рис. 2.1. Дерево цілей

Кожна підціль першого рівня також має власну декомпозицію і окремо має свої аспекти, такі як:

- 1) пошук подібних систем (служить для пізнання подібних систем, які вже створені, їх функціональності, переваг та недоліків)
- 2) виявлення потреб споживачів (дослідження та вивчення потреб споживачів, для розуміння важливих аспектів, які слід враховувати при створенні системи);
- 3) аналіз даних (аналіз даних дозволить вам повністю зрозуміти переваги та недоліки майбутньої системи та врахувати їх при розробці системи);
- 4) опис системних вимог (розробка детального опису системи, її компонентів та взаємодії різних об'єктів, що її складають);
- 5) побудова діаграми потоків даних (створення DFD-діаграм допоможе повністю побудувати та врахувати всі системні вимоги для подальшого впровадження);
- 6) організація взаємовідносин користувачів із сервером

(побудувати схему взаємодії різних типів користувачів із системою та сервером, а також сервера з базою даних);

7) вибір методів та інструментів для програмного забезпечення (необхідно проаналізувати всі методи та всі можливі інструменти для впровадження системи. Беручи до уваги їх переваги та недоліки, а також системні вимоги, виберіть той, що більше підходить);

8) розробка графічного інтерфейсу (для створення графічного інтерфейсу для користувачів);

9) розробка програмного забезпечення (для реалізації програмного забезпечення для сайту, його особливостей та різних алгоритмів).

Підцілі декомпозиції другого рівня повинні відповідати конкретним критеріям, які складають основу для оцінки та визначення підцілей дерева. На основі цих підцілей були встановлені наступні критерії:

1) достовірність (при дослідженні теми інформаційної системи необхідно працювати з якомога більшої кількості літературних джерел та дослідити всі існуючі аналогові системи);

2) актуальність (визначення потреб покупців та аналіз усіх даних, отриманих в ході дослідження, повинні бути визначені точні та надійні висновки для правильної розробки обраної в даний час інформаційної системи);

3) точність, чіткість (ці критерії важливі для реалізації детального опису системних вимог та проектування системних діаграм);

4) швидкість (при організації взаємовідносин користувачів із сервером необхідно враховувати швидкість їх взаємодії);

5) надійність (при створенні програмного забезпечення необхідно вибирати надійні методи та засоби їх реалізації);

6) зручність використання, простота використання, якість та функціональність (графічний інтерфейс та програмне забезпечення повинні бути практичними, простими та зрозумілими для використання, а отже, продукт повинен бути створений з якістю та великою функціональністю).

Наступним кроком є вибір типу системи. Створенню інформаційних

систем у всьому світі приділяється багато уваги. За масштабами, темпами зростання, витратами матеріальних, фінансових і трудових ресурсів, а також за ступенем впливу на процеси управління проблему створення ІС слід розглядати як велике народногосподарське завдання. Інформаційні системи можуть значно різнитися за типами об'єктів управління, характером і обсягом розв'язуваних задач та низкою інших ознак. Загальноприйнятої класифікації ІС досі немає, тому їх можна класифікувати за різними ознаками. Найбільш поширеними протягом тривалого часу були такі класифікаційні угруповання систем:

- За рівнем або сферою діяльності — державні, територіальні (регіональні), галузеві, об'єднань, підприємств або установ, технологічних процесів.
- За рівнем автоматизації процесів управління — інформаційно-пошукові, інформаційно-довідкові, інформаційно-керівні, системи підтримки прийняття рішень, інтелектуальні ІС тощо.
- За ступенем централізації обробки інформації — централізовані ІС, децентралізовані ІС, інформаційні системи колективного використання.
- За ступенем інтеграції функцій - багаторівневі ІС з інтеграцією за рівнями управління (підприємство — об'єднання - галузь та ін.), багаторівневі ІС з інтеграцією за рівнями планування тощо.

Обрано метод аналізу ієрархій для визначення типу інформаційної системи бронювання столиків у закладах громадського харчування.

Обрано чотири типи систем у якості альтернатив: інформаційно-довідкову (A1), інформаційно-пошукову (A2), інформаційно-управляючу (A3), інтелектуальна інформаційно-пошукова система (A4).

Для вибору типу системи необхідно обрати критерії якості: актуальність (K1), повнота дослідження (K2), якість (K3), продуктивність (K4), легкість у використанні (K5) та надійність (K6).

Наступним етапом є побудова матриці попарних порівнянь критеріїв якості, а також матрицю попарних порівнянь альтернатив і розрахувати власні

числа (ВЧ), вектори (ВВ). Використано шкалу відносної важливості пріоритетів для визначення експертної оцінки. Шкалу подано в таблиці 2.1.

Таблиця 2.1

Шкала відносної важливості пріоритетів

Значення	Якісна характеристика
1-3	Несуттєвий пріоритет
4-6	Помірний пріоритет
7-9	Істотний пріоритет

На наступній матриці (матриця критеріїв) показано відносну важливість критеріїв в порівнянні з іншими.

$$A = \begin{matrix} & \begin{matrix} 1 \\ 2 \\ 5 \\ 2 \end{matrix} & \begin{matrix} K1 \\ 1 \\ K6 \end{matrix} & \begin{matrix} 1/2 \\ 1 \\ 6 \end{matrix} & \begin{matrix} 1/5 \\ 1/6 \\ 1 \end{matrix} & \begin{matrix} 1/5 \\ 1/3 \\ 2 \end{matrix} & \begin{matrix} 1/5 \\ 1/6 \\ 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 5 \\ 2 \end{matrix} & & & & & & \\ \begin{matrix} K4 \\ K5 \\ K6 \end{matrix} & \begin{matrix} 5 \\ 4 \\ 5 \end{matrix} & \begin{matrix} 5 \\ 3 \\ 6 \end{matrix} & \begin{matrix} 2 \\ 1/2 \\ 1/4 \end{matrix} & \begin{matrix} 1 \\ 1/3 \\ 1/2 \end{matrix} & \begin{matrix} 3 \\ 1 \\ 3 \end{matrix} & \begin{matrix} 2 \\ 1/3 \\ 1 \end{matrix} \end{matrix}$$

Для розрахунку власних чисел застосовують наступну формулу:

$$\overline{BЧ} = \sqrt[n]{\prod_{j=1}^n a_{ij}} \quad (2.1)$$

$$BЧ = (0,31; 0,38; 2,22; 2,58; 0,93; 1,49)$$

Для розрахунку власних векторів – формулу 2.2:

$$\underline{BВ} = \frac{w_i}{\sum_{i=1}^n w_i}, \quad (2.2)$$

$$BВ = (0,03; 0,04; 0,3; 0,28; 0,11; 0,18)$$

На наступному етапі необхідно побудувати матриці парних порівнянь альтернатив для кожного критерія та відносно головної цілі.

Критерій “актуальність”

$$A = A_2 \begin{pmatrix} A_1 & 1 & 1/3 & 2 & 1/3 \\ 3 & 1 & 3 & 2 & \end{pmatrix}$$

$$\begin{pmatrix} 1/2 & A_1/3 & 1 & 1/2 \\ 3 & 3/2 & 2 & 1 \end{pmatrix}$$

A

4

$$B_1 = (0,68; 2,05; 0,53; 1,31)$$

$$B_2 = (0,14; 0,44; 0,11; 0,28)$$

Критерій “повнота дослідження”:

$$A = A_2 \begin{pmatrix} A_1 & 1/4 & 3 & 1/3 \\ 2 & 3 & & \\ 1/3 & 1/2 & 1 & 1/3 \\ 3 & 4/3 & 3 & 1 \end{pmatrix}$$

$$B_1 = (0,7; 2,21; 0,63; 1,31)$$

$$B_2 = (0,14; 0,45; 0,12; 0,28)$$

Критерій “якість”:

$$\begin{pmatrix} 1 & 1/3 & 1/2 & 1/3 \\ 3 & 1 & 2 & 2 \end{pmatrix}$$

A₁

A = A₂

A

3

A₄

$$B_1 = (0,48; 2,34; 0,84; 1,31)$$

$$BV = (0,09; 0,47; 0,16; 0,26)$$

2	1/2	1	1/2
3	1/2	2	1

Критерій
“продуктивність”:

$$A = \begin{matrix} & A_1 & A_2 & A_3 & A_4 \\ \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{matrix} & \begin{pmatrix} 1 & 1/2 & 1/3 & 1/3 \\ 2 & 1 & 1/2 & 1/2 \\ 3 & 2 & 1 & 1/2 \\ 4 & 3 & 2 & 1 \end{pmatrix} \end{matrix}$$

$$BЧ = (0,48; 1,00; 1,31; 1,86)$$

$$BV = (0,1; 0,21; 0,28; 0,4)$$

Критерій “легкість у використанні”:

$$A = \begin{matrix} & A_1 & A_2 & A_3 & A_4 \\ \begin{matrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{matrix} & \begin{pmatrix} 1 & 1/5 & 1/2 & 1/3 \\ 5 & 1 & 1/3 & 1/3 \\ & & & \end{pmatrix} \end{matrix}$$

$$BЧ = (0,42; 1,49; 1,18; 1,18)$$

$$BV = (0,09; 0,34; 0,22; 0,27)$$

2	2	1	1/3
3	2	1/3	1

Критерій “
надійність”:

$$A = \begin{matrix} & \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \end{matrix} & \begin{pmatrix} 1 & 1/3 & 1/2 & 1/5 \\ 3 & 1 & 1/3 & 2 \\ 2 & 3 & 1 & 2 \\ 5 & 1/2 & 1/2 & 1 \end{pmatrix} \end{matrix}$$

$$BC = (0,42; 1,18; 1,86; 1,05)$$

$$BB = (0,09; 0,26; 0,41; 0,23)$$

Відносно головної
ціль:

$$A = \begin{matrix} & \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \end{matrix} & \begin{pmatrix} 1 & 1/5 & 1/5 & 1/3 \\ 5 & 1 & 1/2 & 1/2 \\ 5 & 2 & 1 & 2 \\ 3 & 2 & 1/2 & 1 \end{pmatrix} \end{matrix}$$

$$BC = (0,33; 1,05; 2,11; 1,31)$$

$$BB = (0,06; 0,21; 0,43; 0,27)$$

Наступним етапом є побудова матриці порівнянь альтернатив. Також необхідно визначити тип інформаційної системи для реалізації популяризації Instagram акаунтів. Наступним кроком є зображення матриці порівнянь альтернатив.

$$A = \begin{matrix} & \begin{matrix} A1 \\ A2 \\ A3 \\ A4 \end{matrix} & \begin{pmatrix} 0,14 & 0,14 & 0,09 & 0,1 & 0,09 & 0,09 \\ 0,44 & 0,45 & 0,47 & 0,21 & 0,34 & 0,26 \\ 0,11 & 0,12 & 0,16 & 0,28 & 0,27 & 0,41 \\ 0,28 & 0,28 & 0,26 & 0,4 & 0,27 & 0,23 \end{pmatrix} \end{matrix}$$

Узагальнені пріоритети =

$$\begin{matrix} 0,1 \\ 0,36 \\ 0,22 \\ 0,28 \end{matrix}$$

В результаті проведення аналізу ієрархій, отримано наступні результати: пріоритет 0.1 – інформаційно-довідкова система, пріоритет 0,36 – інформаційно- пошукова система, 0,22 – це інформаційно управляюча, 0,28 – інтелектуальна інформаційно-пошукова система. Найбільший пріоритет має інформаційно-пошукова система.

2.2. Конкретизація функціонування системи

Діаграми потоків даних (DFD) застосовуються для графічного відображення руху інформації в бізнес-інформаційних системах. Вони демонструють процеси, що забезпечують передачу даних від моменту введення до їхнього збереження у файлах та формування звітів.

DFD поділяються на два основні типи — логічні та фізичні. Логічні діаграми описують рух даних у системі з точки зору виконання бізнес-функцій, тоді як фізичні діаграми показують, як ці логічні потоки реалізуються на практиці.

Графічна структура DFD відображає функції та процеси, що відповідають за збір, обробку, збереження та розподіл даних між системою та її зовнішнім середовищем, а також між внутрішніми компонентами. Завдяки візуальній формі подання DFD є ефективним інструментом комунікації між користувачами та розробниками. Їхня ієрархічна структура дозволяє почати з узагальненого огляду та поступово деталізувати модель до рівня окремих процесів.

Контекстна діаграма є найвищим рівнем DFD і містить лише один процес, що представляє всю систему. Вона демонструє всі зовнішні об'єкти та основні потоки даних між ними і системою, але не включає сховища даних. Цей єдиний процес може бути деталізований у діаграмі наступного рівня — так званий діаграмі 0, яка розкриває основні процеси системи. Кожен із них може бути далі розбитий на підпроцеси для більшої деталізації.

Логічні діаграми потоків даних зосереджуються на бізнес-процесах і показують, як вони функціонують, не враховуючи технічні аспекти реалізації. У цьому випадку ігноруються такі фактори, як конфігурація обладнання, технології збереження даних чи методи комунікації, а увага приділяється лише функціям — збору даних, їх трансформації в інформацію та подальшому представленню.

Фізичні діаграми потоків даних, навпаки, демонструють практичну реалізацію системи, включаючи апаратне та програмне забезпечення, файли й персонал, що бере участь у процесах. Вони створюються для того, щоб

забезпечити правильне втілення процесів, описаних у логічних діаграмах, і досягнення бізнес-цілей.

Методологія DFD є способом моделювання процесів обробки інформації. Вона була розроблена Гейном і Сарсоном незалежно від методології IDEF0. На відміну від IDEF0, DFD не має стандартизованої форми. Якщо стрілки IDEF0 відображають жорсткі взаємозв'язки, то стрілки DFD показують реальний рух даних між функціями. Це дозволяє моделі відображати фізичні характеристики системи — переміщення, збереження та розповсюдження об'єктів.

DFD широко застосовуються для опису інформаційних потоків як усередині системи, так і між системою та зовнішнім середовищем. Їх використовують для моделювання документообігу та побудови бізнес-інформаційних систем. Вони показують, як кожен процес перетворює вхідні дані на вихідні та як ці процеси взаємопов'язані, формуючи мережу діяльності, що відображає роботу всієї системи.

На рис.2.2 зображена контекстна діаграма

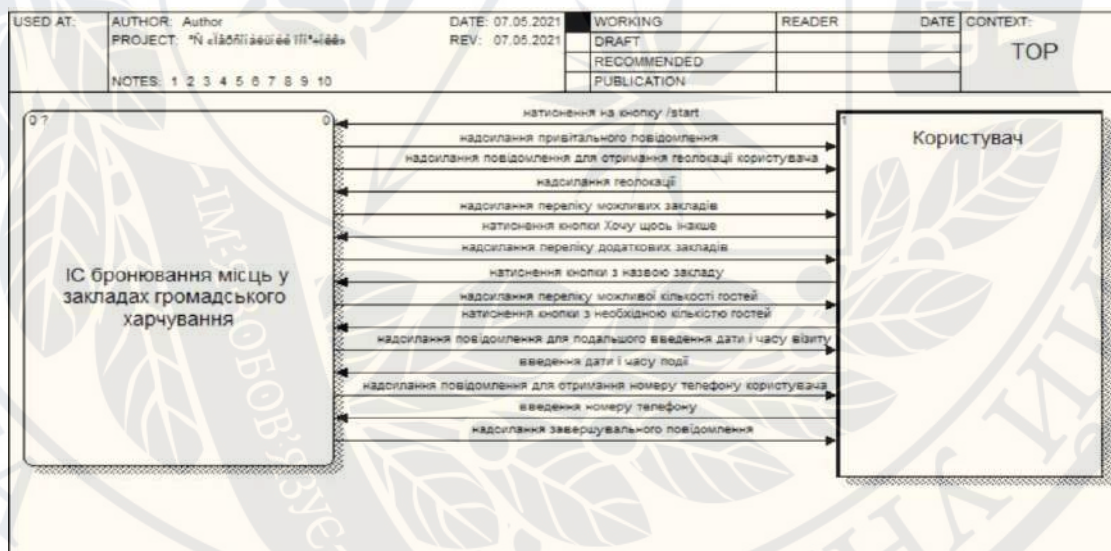


Рис 2.2 Контекстна діаграма

Наступним етапом моделювання є побудова діаграми декомпозиції першого рівня. Вона формується шляхом деталізації головного процесу, представленого на контекстній діаграмі, та його розбиття на окремі підпроцеси.

Для досягнення мети, визначеної контекстною моделлю, необхідно реалізувати всі завдання та функції, що відображені на діаграмі декомпозиції.

Таким чином, діаграма першого рівня демонструє структуру системи у вигляді взаємопов'язаних підпроцесів, які разом забезпечують виконання основної функції. Вона дозволяє більш детально описати логіку роботи системи та визначити ключові етапи її функціонування.

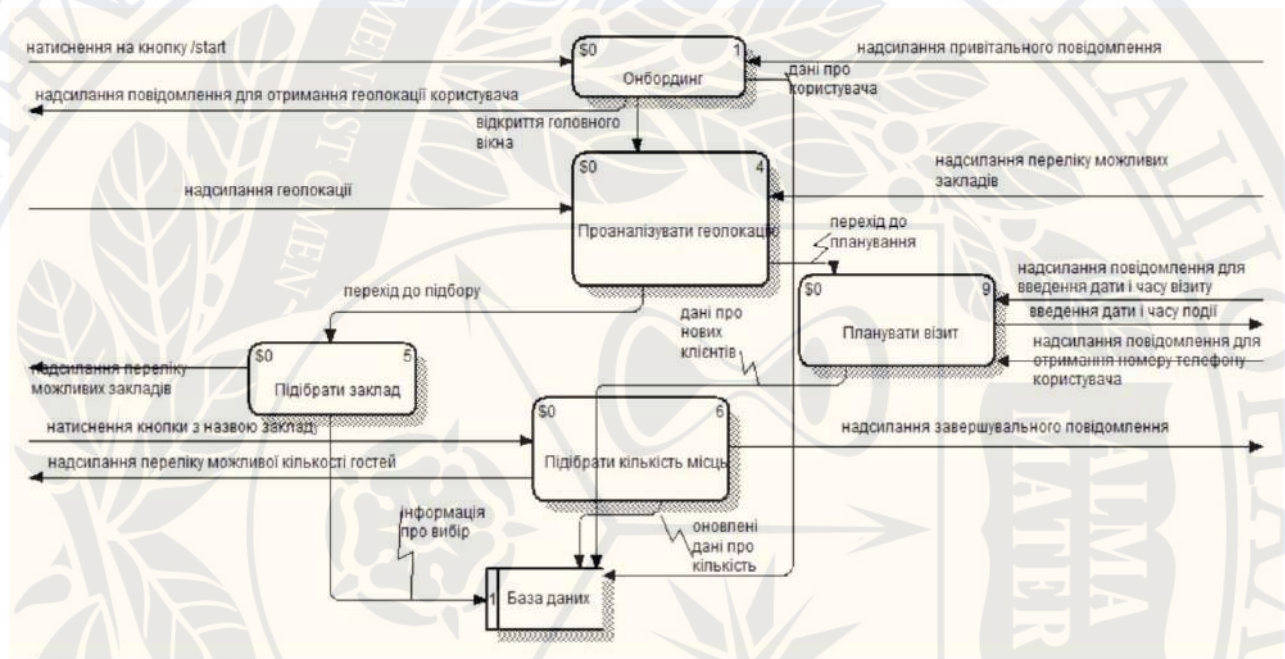


Рис 2.3 DFD першого рівня

Послідовно всі операції такі:

- натиснення на кнопку /start;
- надсилання привітального повідомлення;
- надсилання повідомлення для подальшого отримання геолокації користувача;
- надсилання геолокації;
- надсилання переліку можливих закладів;
- натиснення кнопки «Хочу щось інакше»;
- надсилання переліку додаткових закладів;
- натиснення кнопки з назвою закладу;
- надсилання переліку можливої кількості гостей;
- натиснення кнопки з необхідною кількістю гостей;

- надсилання повідомлення для подальшого введення дати і часу візиту;
- введення дати і часу події;
- надсилання повідомлення для подальшого отримання номеру телефону користувача;
- введення номеру телефону;
- надсилання завершального повідомлення;

2.3. Побудова ієрархії процесів

Дерево вузлів відповідає представленням зв'язків в моделі між дочірніми і батьківськими вузлами. Node Tree Diagram – демонструє ієрархічну залежність функцій. Дана діаграма використовує дерево ієрархій, де верхній блок є контекстною діаграмою, а нижні рівні – декомпозиція. Процес створення є ітеративним, тому функція може декілька разів змінити своє розташування в дереві. Щоб уникнути плутанини, потрібно створювати схему дерева вузлів після кожної зміни.

На рисунку 2.4 зображена ієрархія процесів для системи управління сервісних центрів. Головною метою є «Здійснити бронювання у закладах громадського харчування», мета ділиться на 5 дочірніх процесів. Кожен процес поділяється на функції.

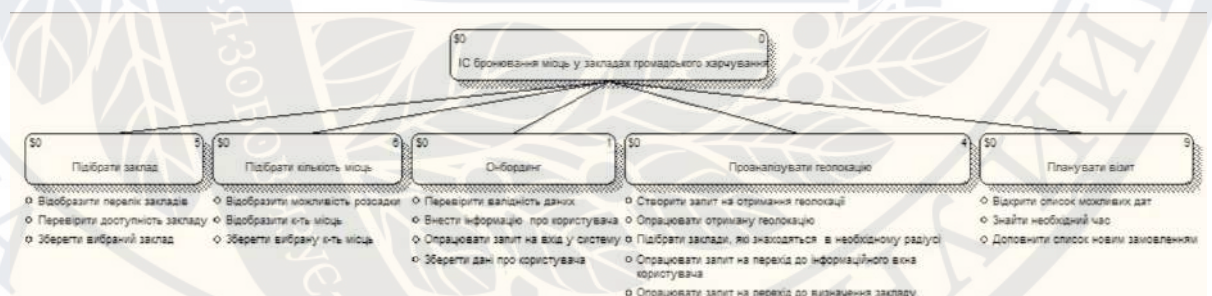


Рис. 2.4. Ієрархія процесів

Перевагою даної моделі є демонстрація зв'язку батьківських діаграм і нащадків в ієрархічному вигляді, що дозволяє оглянути повністю всю модель.

Висновок до другого розділу

Результатом виконання другого розділу роботи є проведений системний аналіз, а саме виконано побудову дерева цілей, діаграми контекстів та її декомпозиції. В процесі побудови дерева цілей визначено основну мету роботи. Розроблено діаграму процесів DFD. Діаграма містить всю інформацію, необхідну для реалізації розробленої системи. Також було здійснено декомпозицію 1 рівня та побудовано ієрархію процесів.

РОЗДІЛ 3. ПРОГРАМНІ ЗАСОБИ РОЗВ'ЯЗАННЯ ЗАДАЧІ

3.1. Вибір парадигми та мови програмування

Від моменту появи першої мови програмування високого рівня **Plankalkül**, а згодом таких мов, як **FORTRAN** та **Common Lisp**, програмування почало розширювати свої межі, формуючи нові підходи до роботи з даними та алгоритмами. Ці підходи отримали назву **парадигм програмування**, які визначають методологію та концептуальну основу мови.

Парадигма програмування — це набір ідей та принципів, що дозволяють описувати завдання на більш високому рівні, ніж машинний код. Кожна парадигма має свої сильні сторони та обмеження: наприклад, створення ігор на функціональних мовах може бути складнішим, ніж на об'єктно-орієнтованих.

Загалом виділяють дві основні категорії:

- **Декларативне програмування**, де програміст описує бажаний результат без уточнення способу його досягнення. Сюди належать функціональна, логічна, математична та модульна парадигми.
- **Імперативне програмування**, що базується на прямому управлінні станом комп'ютера. Прикладами є процедурне та об'єктно-орієнтоване програмування.

Імперативні підходи розглядають програму як послідовність команд, що змінюють стан системи. У процедурному програмуванні повторювані дії групуються в підпрограми, а в структурованому — відмовляються від оператора **goto**, використовуючи блоки та цикли. Обчислювальною моделлю виступає **машина Тьюрінга**.

В об'єктно-орієнтованій парадигмі програма складається з об'єктів, які мають стан (поля) та поведінку (методи). Більшість мов реалізують це через класи, проте існує й прототипний підхід, де нові об'єкти створюються копіюванням існуючих. Прикладами мов є **C++**, **Java**, **JavaScript** та **Common Lisp**.

Функціональне програмування ґрунтується на **λ-численні** та представленні програми як набору чистих функцій. Логічне програмування,

започатковане у 1960-х роках, описує проблему через факти та правила, а вирішення здійснюється за допомогою логічного висновку. Прикладом є мова **Prolog**.

У сучасній розробці важливу роль відіграють **JavaScript**, **HTML** та **PHP**.

- **JavaScript** забезпечує динамічність веб-контенту, інтерактивність та роботу як на клієнтській, так і на серверній стороні (завдяки Node.js).
- **HTML** використовується для створення структури веб-сторінок, включаючи гіперпосилання та навігацію.
- **PHP** дозволяє реалізувати серверну логіку, генерувати динамічні сторінки та обробляти дані форм, хоча має певні проблеми з безпекою.

Для створення інтерфейсів застосовуються фреймворки:

- **EJS** — проста мова шаблонів для Node.js, що генерує HTML.
- **ReactJS** — бібліотека для побудови інтерфейсів користувачів, орієнтована на багаторазове використання компонентів.
- **Angular** — потужний фреймворк для створення масштабних веб-додатків.

Таким чином, розвиток мов та парадигм програмування від декларативних і імперативних підходів до сучасних веб-технологій демонструє еволюцію інструментів, які дозволяють ефективно вирішувати завдання різної складності та забезпечувати інтерактивність у цифровому середовищі.

3.2. Вибір програмних засобів

3.2.1. *Xamarin*

Xamarin - це інструмент, який використовується для кросплатформеної розробки мобільних додатків, що дозволяє інженерам ділитися до 90 відсотків коду на основних платформах. Будучи порівняно новим інструментом, він заснований на стику технологій Microsoft і вже налічує понад 1,4 мільйона розробників.

Платформа була побудована розробниками Mono, відкритої платформи

розробки на основі .NET Framework, на чолі з Мігелем де Ікаса і вперше представлена в 2001 році. Компанія Xamarin була заснована 16 травня 2011 року.

Xamarin створювалася як комерційний проект до тих пір, поки компанія не була придбана Microsoft в 2016 році. Xamarin став популярним кросплатформним продуктом для розробки мобільних додатків в рамках екосистеми Microsoft. Це придбання подолало фінансовий бар'єр для використання Xamarin. Оскільки Microsoft зробила Xamarin SDK з відкритим вихідним кодом, він став частиною інтегрованого середовища розробки Xamarin Visual.

Мова C#. Xamarin використовує єдину мову, C#, для створення додатків для всіх мобільних платформ. На відміну від інтерпретованих рішень, таких як Appcelerator Titanium, Xamarin скомпільована на рідній мові, що робить його ідеальним варіантом для створення високопродуктивних додатків з рідним видом і відчуттями.

Засноване на .NET фреймворку. C# - це зрілий мову з сильною безпекою, що запобігає несподіване поведінку коду. Оскільки C# є одним з мов .NET-фреймворка, він може використовуватися з рядом корисних функцій .NET, таких як Lambdas, LINQ і асинхронне програмування (async / await).

Технічно кажучи, Xamarin використовує C# і рідні бібліотеки, загорнуті в шар .Net, для розробки крос-платформних додатків. Такі додатки часто порівнюються з нативними для мобільних платформ розробки як iOS, так і Android з точки зору продуктивності і користувацького досвіду. Крім того, Xamarin може використовувати весь нативний і новітній доступ до API для використання базових платформних можливостей додатків Xamarin, таких як ARKit на iOS або Android Multi-Window.

3.2.2. *Unity*

Unity - це ігровий движок, розроблений компанією Unity Technologies, вперше анонсований і випущений в червні 2005 року на Всесвітній конференції розробників Apple Inc. у вигляді ексклюзивного ігрового движка для Mac OS X.

Станом на 2018 рік, движок був розширений для підтримки більш ніж 25 платформ. Движок може використовуватися для створення тривимірних, двовимірних, віртуальних ігор та ігор доповненої реальності, а також для симуляцій та інших експериментів [4, 5]. Движок був перейнятий в галузі, не пов'язані з відеоіграми, такі як кінематограф, автомобілебудування, архітектура, проектування та будівництво.

З моменту запуску було випущено кілька основних версій Unity. Остання стабільна версія 2021.1.0 [1] була випущена в березні 2021 року.

Unity дає користувачам можливість створювати ігри і досвід як в 2D, так і в 3D, а движок пропонує основний скриптова API на C #, як для редактора Unity у вигляді плагінів, так і для самих ігор, а також функціональність перетягування. До того, як C # стала основною мовою програмування, використовуваним в движку, він раніше підтримував Boo, який був вилючений з виходом Unity 5 [19], і версію JavaScript під назвою UnityScript, яка була застаріла в серпні 2017 року, після виходу Unity 2017.1, на користь C #. [14].

В рамках 2D ігор, Unity дозволяє імпортувати спрайт і просунутий 2D рендер світу. Для 3D ігор Unity дозволяє задавати параметри текстурного стиснення, мип- карт і дозволу для кожної платформи, підтримуваної ігровим движком [21] і забезпечує підтримку бампапінга, рефлексивного мапінга, параллаксного мапінга, екранної оклюзії навколишнього простору (SSAO), динамічних тіней з використанням теней- карт, рендеринга в текстуру і повноекранних пост-обробок. [24].

3.2.3. *Apache Cordova*

Apache Cordova - це фреймворк з відкритим вихідним кодом, що дозволяє веб- розробникам використовувати HTML, CSS і JavaScript-контент для створення власного додатка для різних мобільних платформ.

Cordova бере веб-додаток і рендерить його в рамках рідного WebView.

WebView - це компонент програми (наприклад, кнопка або рядок табуляції), який використовується для відображення веб-контенту всередині «рідного»

додатка. WebView можна уявити як веб-браузер без будь-яких стандартних елементів призначеного для користувача інтерфейсу, таких як поле URL або рядок стану. Веб-додаток, що працює всередині цього контейнера, як і будь-яка інша веб-додаток, що буде працювати всередині мобільного браузера - воно може відкривати додаткові HTML-сторінки, виконувати JavaScript-код, відтворювати медіа-файли і взаємодіяти з віддаленими серверами. Цей тип мобільного додатка часто називають гібридним додатком.

Як правило, веб-додатки виконуються в пісочниці, що означає, що вони не мають прямого доступу до різних апаратних і програмних функцій на пристрої. Хорошим прикладом цього є база даних контактів на мобільному пристрої. Ця база даних імен, телефонних номерів, електронної пошти та інших бітів інформації недоступна для веб-додатків.

Крім базового фреймворка для запуску веб-додатків в рідному додатку, Cordova також надає JavaScript API для доступу до широкого спектру функцій пристрою, наприклад, до бази даних контактів. Ці можливості розкриваються за допомогою колекції плагінів. Модулі забезпечують міст між нашим веб-додатком і рідними функціями пристрою.

Команда Ionic пішла далі і створила Ionic Native, з інтерфейсами TypeScript для більш ніж 200 найбільш поширених плагінів, і версіями, які підтримуються на підприємстві, для команд, які хочуть скористатися перевагами Ionic для управління поточними оновленнями, виправленнями безпеки, сумісністю та іншими аспектами доступу до рідного пристрою.

3.2.4. *React Native*

React Native — це фреймворк на основі JavaScript, призначений для створення повноцінних мобільних застосунків під iOS та Android. Він побудований на базі бібліотеки React, розробленої компанією Facebook для формування інтерфейсів користувача, проте замість браузера орієнтується на мобільні платформи. Це означає, що веб-розробники отримали можливість

створювати мобільні додатки, які виглядають і функціонують як справжні нативні програми, використовуючи знайомі інструменти JavaScript.

Важливою перевагою є те, що значна частина коду може бути спільною для різних платформ, що спрощує одночасну розробку під Android та iOS. Подібно до React для вебу, застосунки у React Native пишуться за допомогою комбінації JavaScript та JSX (JavaScript XML). Під час виконання React Native використовує спеціальний «міст», який звертається до нативних API рендерингу — Objective-C для iOS та Java для Android. Завдяки цьому інтерфейс програми складається з реальних мобільних компонентів, а не веб-переглядів, і виглядає як класичний мобільний застосунок.

Крім того, React Native надає JavaScript-інтерфейси для доступу до системних API, що дозволяє використовувати функції пристрою, такі як камера чи геолокація. На сьогоднішній день фреймворк підтримує дві основні платформи — iOS та Android, але має потенціал для подальшого розширення на інші мобільні середовища.

3.2.5 Node-RED

Node-RED використовує графічні потоки і вузли, які мають окремі компоненти в потоці, щоб по суті створити програму. Що мені дійсно подобається в Node-RED, так це те, що він обидва графічний, тому він дає візуальні можливості для створення програми, але, також, він дозволяє багато функціонального управління через JavaScript. JavaScript - це мова програмування, що лежить в основі Node-RED. Node-RED - це середовище програмування, що працює всередині браузера. Node-RED використовується для створення графічних програм, які називаються потоками. Потоки складаються з вузлів, що представляють собою прямокутні об'єкти, які ви бачите в прикладі нижче.

Для побудови потоку ми використовуємо інтерфейс перетягування. За допомогою цього інтерфейсу легко збирати вузли в конфігураціях, в результаті

чого виходить програма, яка робить щось корисне. Кожен вузол попередньо запрограмований на виконання чогось визначеного (схожого на функцію в рядку текстового мови Python або Ruby). Ви можете створювати користувацькі вузли, які містять обраний вами JavaScript, подібно призначеним для користувача функціям в текстових мовах.

Node-RED поставляється з декількома вбудованими вузлами, але ви також можете встановити сторонні вузли. Так само, як і в Arduino, ви можете встановити сторонні бібліотеки.

Назва Node-RED походить від базової технології, на якій він побудований - Node.js, що представляє собою JavaScript-фреймворк. Це дуже легка середовище розробки і середовища виконання, що робить її відмінним інструментом для створення додатків, які повинні бути дуже моторними і дуже швидкими у виконанні, так що вони можуть працювати на дешевому обладнанні, такому як Raspberry Pi.

Node-RED з відкритим вихідним кодом. І в результаті, є багато людей, які вносять свій внесок в це. Він існує вже довгий час, і дійсно стабільний і використовується як любителями, так і великими корпораціями.

Висновок до третього розділу

Для реалізації описаної програми використовується платформа Node-RED. Саме ця програма була обрана для використання, тому що вона проста, зручна та зрозуміла це й були головні критерії підбору програми. А щоб реалізувати скриптову частину роботи використовується мова програмування Js.

РОЗДІЛ 4. ПРАКТИЧНА РЕАЛІЗАЦІЯ

4.1. Опис створеного програмного засобу

На основі описаних вище розділів було створено систему бронювання даної інформаційної системи бронювання місць у закладах громадського харчування.

Програма була реалізована за допомогою платформи Node-RED. Вона складається з інтерфейсної частини скриптів і сцен. Кожна сцена має закріплений скрипт з набором відповідних функцій.

Розроблена база даних підключена до створеної системи та збирає замовлення, зберігаючи їх у системі.

Структура бази даних

Для даного веб-сайту використовується база даних MongoDB. Вона складається з наступних таблиць: places, tables, join_table, reservation, order_info.

Таблиця Places містить дані про заклади.

Таблиця Tables зберігає інформацію про вільні столики. Таблиця Join_table зберігає інформацію про обрані столики

Таблиця Reservation містить інформацію про створені користувачем резервацію. Таблиця Order_info містить усю інформацію про заброньовані столики.

Структуру бази даних зображено на рис.4.1.

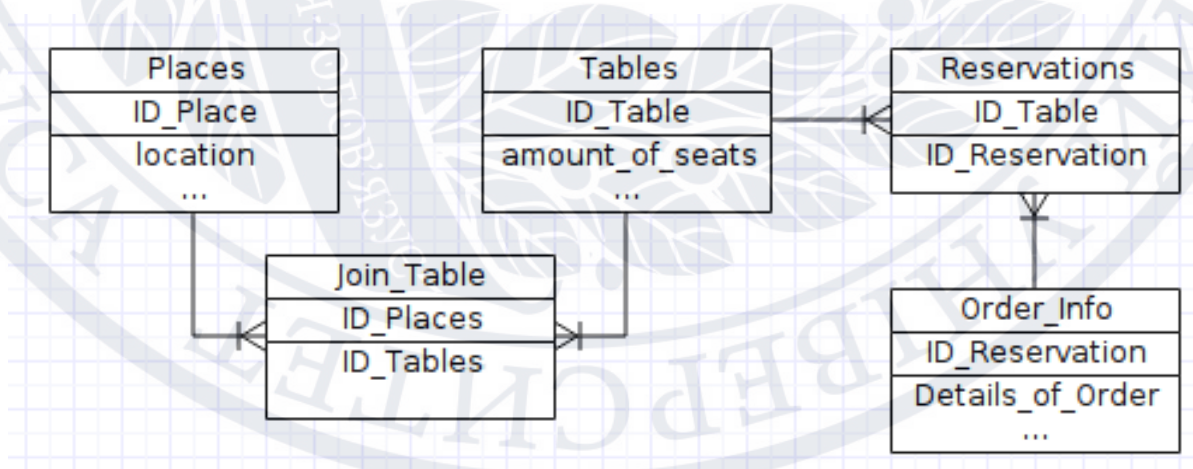


Рис. 4.1. Схема бази даних

4.2. Інструкція користувача

Мета цієї інформаційної системи - дозволити користувачам здійснювати бронювання ресторанів через Інтернет.

Основними компонентами є:

- реєстрація користувача;
- реєстрація користувача;
- надає доступ до вибору закладів;
- запис персональних даних;
- інформувати користувачів про підтвердження бронювання.

Повна назва впровадженої програми - «Інформаційна система FoodSearch для бронювання у закладах харчування».

Скорочена назва - "FoodSearch".

Додаток пишеться за допомогою платформи Node-RED, використовуючи сценарій, написаний мовою програмування JavaScript.

Програма може бути корисною для користувачів, які хочуть забронювати столик, не надто відволікаючись від своїх справ.

Класи завдань, які потрібно вирішити.

Розробка базується на парадигмі програмування потоків. Основні функції програми:

- авторизація зареєстрованого користувача;
- реєстрація нового користувача;
- навігаційні засоби;
- створення нового бронювання;
- підтвердження / скасування бронювання.

Опис основних особливостей та характеристик програми

Ця система створена за принципом програми, яка має один режим роботи - онлайн-режим користувача.

Функціональні обмеження програми.

Додаток обмежений для використання на смартфонах з операційною системою нижче Android 4.4 та 32-розрядним процесором.

Вхідні вихідні дані

Вхідні дані - це дані для реєстрації користувачів, обробки бронювання. Вихідні дані - це оформлення бронювання.

4.3. Аналіз контрольного прикладу

Для перевірки працездатності системи бронювання місць у закладах громадського харчування було здійснено аналіз її функціонування та наведено відповідні скріншоти роботи.

У результаті дослідження створено інформаційно-консультаційну систему «**FoodSearch**», яка дозволяє користувачеві обрати серед найближчих до його поточного розташування заклад, визначити кількість осіб та бажану дату відвідування. Після оформлення бронювання система надає контактний номер телефону закладу для забезпечення зворотного зв'язку.

На рисунку 4.1 показано початковий етап роботи додатку. Після запуску програма пропонує користувачеві надати доступ до його геолокації, щоб автоматично визначити найближчі заклади громадського харчування та сформувати список доступних варіантів для бронювання.

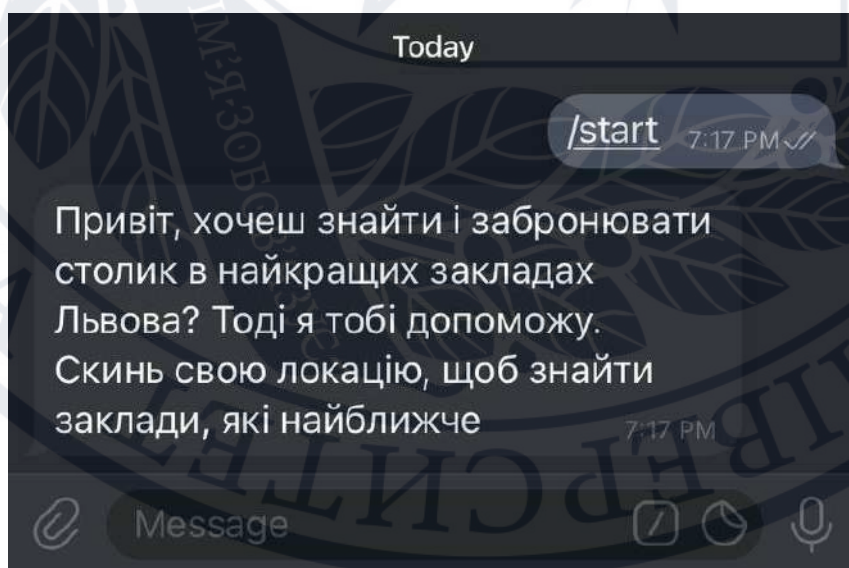


Рис.4.1. Початок роботи системи

Після вводу місця знаходження починається основна діяльність програми. Система пропонує користувачу обрати заклад серед запропонованих (рис. 4.2), або натиснути варіант «Хочу щось інакше» аби подивитися інші заклади (рис. 4.3).

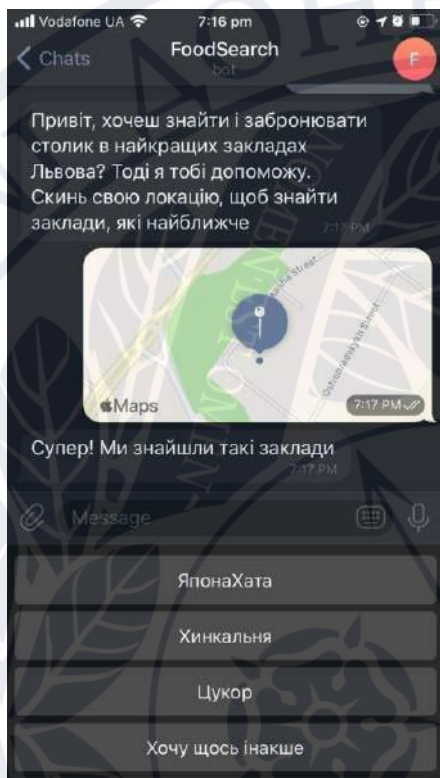


Рис.4.2. Вибір закладу

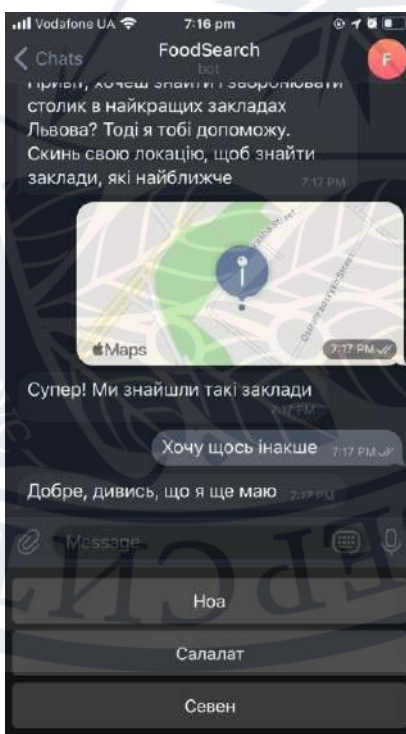


Рис.4.3. Вибір закладу

Після вибору закладу, програма запитує кількість гостей на яких замовляється столик. Якщо користувач передумав і хоче переобрати заклад, є можливість повернутися у попереднє меню, натиснувши кнопку «Назад» (рис. 4.4).

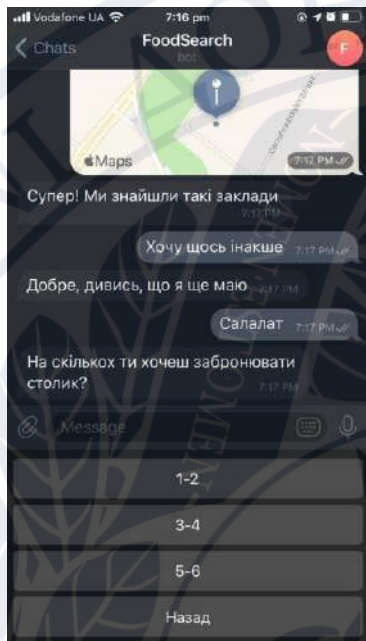


Рис.4.4. Вибір кількості відвідувачів

Наступним кроком програма просить користувача ввести дату на яку потрібно поставити бронювання. Введення відбувається самим користувачем через клавіатуру його пристрою (рис. 4.5).

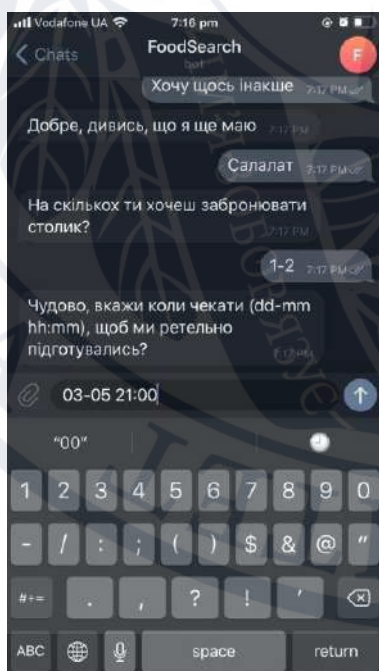


Рис. 4.5. Вибір дати бронювання

Передостаннім кроком є введення свого дійсного номеру телефону для можливості зв'язку у разі непередбачуваних обставин. Програма також дає номер телефону обраного закладу, щоб користувач зміг зателефонувати та обговорити деталі (якщо це потрібно), або з інших питань (рис. 4.6).

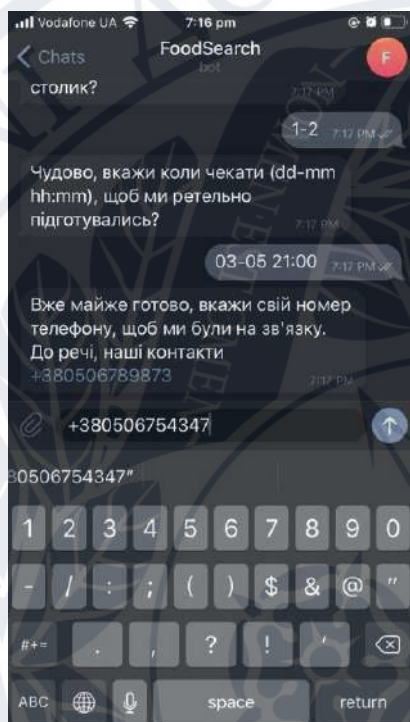


Рис. 4.6. Введення контактного номеру телефону для зв'язку

Останнім кроком йде завершальне повідомлення (рис. 4.7)

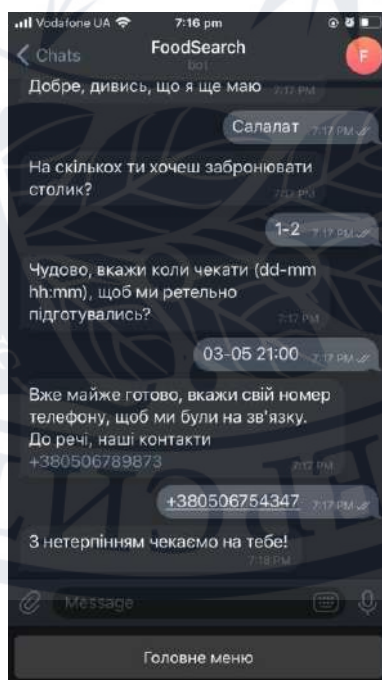


Рис. 4.7. Завершальне повідомлення

Висновок до четвертого розділу

В четвертому розділі було розроблено здійснено програмну реалізацію, виконано загальний опис розробленого програмного забезпечення, продемонстровано та розглянуто контрольний приклад.



ВИСНОВКИ

Популярні в наш час відвідування закладів громадського харчування можуть стати проблемою якщо у той час, коло прийдете до закладу, а там не буде вільних місць. Тому було придумано бронювання. Це дуже зручно, а головне, можна впевнитись, що тоді, коли прийдете, місце буде. У той самий час, бронювання на папері доволі часто приводить до помилок. Всі адміністратори живі люди і вони можуть помилитися, або чогось недогледіти. Це не критично, але доволі неприємно, особливо коти чекаєш цього походу до, наприклад, ресторану, а прийшовши з'ясовуєш, що через помилку, заброньованого місця немає. Тому, в наш час усього технологічного, майже всі заклади перейшли на онлайн-бронювання. Програма завжди знає і пам'ятає які заклади, дати та місця вільні для відвідування. Онлайн- бронювання допомагає клієнту зекономити час та мінімум відволікатися від своїх поточних справ, а закладу уникнути казусів і помилок.

Основною метою програми є надання вибору місць та бронювання. Система також повинна надавати доступ для вибору кількості людей, які прийдуть, та можливість ввести бажану дату.

Шукаючи літературу, було виявлено, що системи бронювання дуже популярні та необхідні, оскільки допомагають економити час клієнтів та уникати помилок. Крім того, були визначені системи-аналоги та визначені технічні характеристики, необхідні для подальшого моделювання програми.

В результаті досліджень та аналізу системи були розроблені діаграми DFD. Діаграми містять всю інформацію, необхідну для реалізації розробленої системи. Проведення системного аналізу дозволило окреслити коло проблем, що підлягають вирішенню, та методи їх вирішення, а саме розробку інтелектуальної інформаційної системи для бронювання місць у закладах громадського харчування.

Серед варіантів реалізації цієї функціональності були проаналізовані програмні можливості ряду платформ. Node-RED - це платформа, яку було обрано основною для реалізації цієї програми. Для реалізації частини сценаріїв була використана мова програмування Js.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Андрущенко, Т. В. Інформаційні системи в сервісних підприємствах: сучасні рішення та тренди. – Київ: КНЕУ, 2024. – 312 с.
2. Бодненко, Д. М. Архітектура програмних систем: навч. посіб. – Київ: Ліра-К, 2025. – 280 с.
3. Гуменюк, Ю. О. Бази даних та інформаційні сховища: моделювання і оптимізація. – Львів: ЛНУ ім. Франка, 2024. – 364 с.
4. ДСТУ ISO/IEC 25010:2023. Інженерія систем та програмного забезпечення. Вимоги до якості. – Київ: УкрНДНЦ, 2024.
5. ДСТУ ISO/IEC 27001:2024. Системи управління інформаційною безпекою. Вимоги. – Київ: Мінекономіки, 2024.
6. Євтушенко, К. О. Проектування інформаційних систем: сучасні методи та практики. – Харків: ХНЕУ, 2025. – 248 с.
7. Захарченко, Л. П. Web-розробка та інтерактивні інтерфейси. – Київ: Видавничий дім «Освіта», 2024. – 290 с.
8. Кравченко, І. В. Методологія створення корпоративних ІС. – Дніпро: ДНУ, 2025. – 308 с.
9. Козак, С. І. Основи тестування програмного забезпечення: навч. посіб. – Тернопіль: ТНТУ, 2024. – 212 с.
10. Литвин, В. М., Шаховська, Н. Б. Інтелектуальні ІС: моделі, алгоритми, застосування. – Львів: ЛП, 2024. – 420 с.
11. Мазаракі, А. А. Цифровізація сервісних послуг: виклики та рішення. – Київ: КНТЕУ, 2025. – 280 с.
12. Мельник, О. В. Програмна інженерія: від вимог до впровадження. – Львів: Сполом, 2024. – 356 с.
13. Олійник, В. Ю. Хмарні технології та веб-сервіси: навч. посіб. – Київ: КНУ ім. Шевченка, 2025. – 260 с.

- 14.Петренко, Д. В. Технології кібербезпеки в сучасних ІС. – Київ: ДУТ, 2024. – 300 с.
- 15.Pichler, M. Modern Web-Based Booking Systems: Architecture and Performance. – Springer, 2024. – 215 p.
- 16.Johnson, A. Designing Reservation Systems: User Experience and Data Flows. – O'Reilly Media, 2025. – 198 p.
- 17.IEEE Standard 29148-2024. Systems and Software Engineering — Requirements Engineering. – IEEE, 2024.
- 18.ISO/IEC 42001:2024. Artificial Intelligence Management Systems — Requirements. – Geneva: ISO, 2024.
- 19.European Union Agency for Cybersecurity (ENISA). Cybersecurity Guidelines for Online Service Platforms. – Brussels, 2025.
- 20.MDN Web Docs. Web Technologies Documentation. – URL: <https://developer.mozilla.org/>
- 21.W3C. Web Application Security and Architecture Guidelines. – URL: <https://www.w3.org/>
- 22.Sharma, K. Load Testing Techniques for High-Traffic Web Systems. – ACM Journal, 2024. – Vol. 19(3), P. 44–58.
- 23.Nguyen, T. Advanced Database Transactions in Multi-User Systems. – IEEE Computing, 2025. – №1, P. 18–33.
- 24.Li, J. Scalable Backend Architecture for Real-Time Booking Platforms. – MIT Press, 2024. – 276 p.
- 25.European Data Protection Board. Guidelines on Personal Data Processing in Online Platforms. – Brussels, 2024.
- 26.Journal of Information Systems Engineering. Trends in Reservation System Automation. – 2025. – Vol. 12(2), P. 5–29.
- 27.Що таке On-line сервіси? [Електронний ресурс] – Режим доступу: <https://avada-media.ua/services/on-line-servisy/>.
- 28.Онлайн сервіси і їх види [Електронний ресурс] – Режим доступу: <http://book-science.ru/applied/it/onlajn-servisy-i-ih-vidy.html>.

29. За і проти: Навіщо ресторанам сервіси бронювання столиків [Електронний ресурс] – Режим доступу: <https://habr.com/ru/company/jowi/blog/369115/>.
30. Клієнтська оптимізація і етапи розробки [Електронний ресурс] – режим доступу: <http://tods-blog.com.ua/ezarabotok/klientskaya-chast/>.
31. Купер А.Б. Про інтерфейс. Основи проектування взаємодії. – Лондон, 2012. – 365 с.
32. Раскин Д.М. Интерфейс: нові напрямки в проектуванні комп'ютерних систем. – Париж, 2009. – 144 с.
33. Поради до розробки архітектури сайту [Електронний ресурс] – Режим доступу: <https://semantica.in/blog/polnoe-rukovodstvo-po-planirovaniyu-arkhitektury-sajta-15-sovetov-dlya-maksimalnogo-seo.html#toc1Eclipse>.
34. React [Електронний ресурс] – режим доступу: <https://ru.wikipedia.org/wiki/React>.
35. React Native [Електронний ресурс] – режим доступу: https://en.wikipedia.org/wiki/React_Native.
36. Переваги використання React Js [Електронний ресурс] – режим доступу: <https://xbsoftware.ru/blog/pochemu-stoit-ispolzovat-react-js-razrabotke-prilozhenij/>
37. Документація Python [Онлайн]. Доступно: <https://docs.python.org/3/>
38. Документація Django. [Онлайн]. Доступно: <https://docs.djangoproject.com/en/3.2/>
39. Документація TextBlob. [Онлайн]. Доступно: <https://textblob.readthedocs.io/en/dev/>
40. HTML. Доступно: <https://developer.mozilla.org/en-US/docs/Web/HTML>
41. CSS. [Онлайн]. Доступно: <https://developer.mozilla.org/en-US/docs/Web/CSS>
42. Deep Learning Book by Ian Goodfellow, Yoshua Bengio, and Aaron Courville [Онлайн]. Доступно: <http://www.deeplearningbook.org/>

43. Downey A. Think Python: How to Think Like a Computer Scientist, 2nd edition. Needham: Green Tea Press, 2015. 36 c
44. James A., Edward R. Talking Nets An Oral History of Neural Networks. Cambridge: The MIT Press, 2020. 10 c
45. Christian B. The Alignment Problem: Machine Learning and Human Values. New York: W. W. Norton & Company, 2020. 25 c.
46. Bradski G., Kaehler A. Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol: O'Reilly Media, Inc, 2011. 53 c.
47. Raschka S., Mirjalili V. Python Machine Learning. Livery Place: Packt Publishing Ltd, 2019. 296 c.
48. Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. Morgan & Claypool Publishers.
49. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing. Pearson. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>



ДОДАТОК А

```

####
# semantic version comparition using semver specification
http://semver.org/ # This bash script compares pre-releases
alphabetically as well
#
# returns 1 when A
# greater than B #
# returns 0 when A
# equals B
# returns -1 when
# A lower than B #
# Usage
# chmod +x semver.sh
# ./semver.sh
1.0.0 v1.0.0-
rc.0 # --> 1
#
# Author
Ariel
Rodriguez
# License
MIT
####
semver_compare() {
    local version_a version_b pr_a pr_b
    # strip word "v" and extract first subset version (x.y.z from x.y.z-foo.n)
    version_a=$(echo "${1//v/}" | awk -F'-' '{print $1}')
    version_b=$(echo "${2//v/}" | awk -F'-' '{print $1}')

    if [ "$version_a" \=
"$version_b" ] then
        # check for pre-release
        # extract pre-release (-foo.n from x.y.z-
foo.n) pr_a=$(echo "$1" | awk -F'-'
'{'print $2}')
        pr_b=$(echo "$2" | awk -F'-' '{print $2}')

        #####
        # Return 0 when A is equal to B
        [ "$pr_a" \= "$pr_b" ] && echo 0 && return 0

        #####
        # Return 1

        # Case when A is

```

```
not pre-release if [  
-z "$pr_a" ]  
then  
  echo  
1 &&  
return  
0 fi
```

