

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка веб-системи управління готельним комплексом

Виконав студент групи К-41
спеціальності 122 Комп'ютерні науки
Майнов Олександр Олегович

Керівник к.геогр.н., доцент
Кузніченко Світлана Дмитрівна

Консультант _____

Рецензент к.техн.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2021

ЗМІСТ

Скорочення та умовні позначки	5
Вступ.....	6
1 Аналіз предметної області та програмних систем аналогів	8
1.1 Збір інформації та підготовка вихідних даних	8
1.2 Постановка завдання	10
1.3 Аналіз програмних систем аналогів	10
2 Вибір та обґрунтування засобів розробки	16
2.1 Функціональні вимоги	16
2.2 Вибір системи керування базами даних.....	17
2.3 Вибір фреймворків та мови програмування	22
2.4 Вибір стильових інструментів.....	28
2.5 Обґрунтування вибору web-сервера.....	30
3 Проектування та програмна реалізація веб-системи.....	34
3.1 Діаграма процесів	34
3.2 Діаграма прецедентів	34
3.3 Діаграми станів, діяльності та послідовності	35
3.4 Побудова моделі потоків даних	39
3.5 Структура і схема бази даних.....	45
3.5 Тестування системи.....	48
Висновки	57
Перелік джерел посилання	59
Додаток А Контекстна діаграма управління готельним комплексом	62
Додаток Б Моделі потоків даних DFD.....	63
Додаток В Діаграма дерева вузлів.....	65
Додаток Г Фрагмент програмного коду.....	66

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД	– база даних
ЕОМ	– електронно-обчислювальна машина
ІС	– інформаційна система
ПК	– персональний комп'ютер
СКБД	– система керування базами даних
MVC	– Model View Controller – модель-вигляд-контролер
DFD	– Data Flow Diagrams – діаграми потоків даних
JVM	– Java Virtual Machine – віртуальна машина Java
PMS	– Property Management System – система управління готелем
UI	– User Interface – інтерфейс користувача
UML	– Unified Modeling Language – уніфікована мова моделювання

ВСТУП

Проблема управління роботою готеля є актуальною для кожного власника готельного бізнесу. Web-система, що об'єднує в єдиний цикл обліку всі життєво важливі елементи готельного бізнесу, дозволить мінімізувати втрати прибутку, ефективно контролювати роботу персоналу, підвищити якість обслуговування клієнтів і продуктивність роботи. Підвищення продуктивності досягається завдяки оптимізації використання ресурсів готелю, економії витрат на обробку даних і більш ефективного здійснення основних виробничих операцій.

Створення інтегрованої готельної інформаційної мережі, в центрі якої знаходиться сучасна система управління, дозволить звести до мінімуму необхідність дублювання інформації і відповідно уникнути помилок, що виникають при ручному введенні даних. Економія праці навіть одного працівника в день дозволить за рік заощадити істотну суму грошей, не кажучи вже про уникнення помилок.

Важливо, щоб система була створена з урахуванням вимог замовника, була адаптована під потреби підприємства і мала зручний інтерфейс, який дозволить персоналу максимально швидко і непомітно для гостя провести всі стандартні операції з бронювання та заселення. Основним ресурсом готелю є його номерний фонд, і правильне управління процесом бронювання дозволить його використовувати найбільш ефективно.

Метою кваліфікаційної роботи бакалавра є розробка Web-системи для управління роботою готельного комплексу.

Передбачається, що Web-застосунок дозволить адміністраторам підвищити якість управління процесом бронювання і розміщення гостей, а порт'є скоротити час обробки оперативних даних. Програма повинна підвищити ступінь достовірності обробки інформації, оперативність отримання вихідних даних і поліпшити значення показників якості обробки інформації.

Для досягнення поставленої мети були сформульовані наступні завдання:

- провести аналіз предметної області щодо організації готельного бізнесу;
- провести порівняльний аналіз існуючих програм аналогів;
- обґрунтувати вибір програмних засобів розробки та технологій для вирішення поставленої проблеми;
- провести проектування програмної системи з використанням мови UML;
- розробити структуру бази даних;
- виконати реалізацію клієнтської та серверної частин веб-додатка;
- підготувати інструкцію користувача;
- виконати перевірку ефективності застосування розробленої програми.

Структура кваліфікаційної роботи бакалавра складається з вступу, трьох розділів, висновків, переліку посилань на 21 найменування, додатків. Повний обсяг роботи становить 72 сторінки, містить 21 рисунок і 17 таблиць.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОГРАМНИХ СИСТЕМ АНАЛОГІВ

1.1 Збір інформації та підготовка вихідних даних

На даному етапі була досліджена робота готельного комплексу, роботу і документообіг якого, необхідно автоматизувати. Визначено функції персоналу та особливості їх роботи. Існуючі в готелі інформаційні системи об'єднуються в різні служби. На першому місці знаходиться внутрішня організація, яка покликана керувати працівниками і направляти їх діяльність на досягнення цілей підприємства. У структуру готельного підприємства входять наступні служби:

- відділ обслуговування;
- харчовий комплекс;
- відділ кадрів, бухгалтерія.

Передбачається автоматизувати роботу відділу обслуговування, тому доцільно розглянути його структуру більш детально.

До відділу обслуговування відносяться:

- головний адміністратор;
- служба управління номерним фондом;
- служба прийому і розміщення;
- відділ резервування;
- відділ бронювання;
- адміністративно-господарський підрозділ;
- обслуговуючий персонал (служба порт'є, служба покоївок);
- служба безпеки;
- нічний аудитор.

Основними обов'язками головного адміністратора можна вважати ведення балансових гостьових рахунків, пропозицію гостям послуг готелю (парковка, хімчистка, доставка пошти, послань та ін.), керівництво збутом номерів. Головний адміністратор контролює програмну систему управління. У си-

стемі, слід передбачити чотири важливих процесу, які повинні забезпечувати адміністратора необхідною інформацією в таких напрямках, як:

- загальні питання з управління;
- управління розрахунками з гостями;
- управління службою резервування;
- управління обслуговуванням гостей.

Служба управління номерним фондом бронює номери, веде прийом гостей, їх реєстрацію та розміщення, а також займається наданням гостям побутових послуг.

Завдання служби прийому та розміщення – контроль за процесом розміщення клієнтів та вирішення конфліктів. Служба розміщення використовує не зарезервовані номери для безпосереднього продажу клієнтам за вищою ціною.

Головне завдання служби резервування – продати якомога дорожче всі порожні номери в готелі.

Служба бронювання збирає заявки на бронювання і стежить за підвищенням попиту, ґрунтуючись на якому збільшується вартість розміщення. У службу розміщення передаються номери, що не були заздалегідь заброньовані, з метою їх подальшого продажу за більш високою ціною. В обов'язки даної служби входить спостереження за ситуацією, що складається в готелі, збір заявок на резервування вільних номерів та визначення можливості підвищення попиту, яку готель міг би використовувати для збільшення вартості розміщення з метою отримання підприємством більшого доходу.

Служба порт'є здійснює контроль за номерним фондом готелю, веде картотеку по зайнятості номерів і наявності вільних місць, і виконує функції інформаційного центру. Інформація через службу порт'є рухається в двох напрямках – до гостей та в різні підрозділи готельного підприємства.

Служба адміністрації управляє і несе відповідальність за всі служби готелю.

1.2 Постановка завдання

Проведемо аналіз умов функціонування готельного комплексу, що надає можливість проживання відвідувачів. У готелі зберігається наступна інформація про клієнтів: прізвище, ім'я, по батькові, паспортні дані, адреса, телефон, замітка, стать, поштова скринька, національність. Адміністратор реєструє клієнта, і вказує дату перебування, суму оплати, обрані кімнати, обрані послуги, а також відмітки про виконання оплати страхувальника. При реєструванні відбувається автоматичний підрахунок коштів і генерація рахунку для клієнта.

Мета проектування ІС – автоматизація діяльності та документообігу готельного комплексу.

Основні функції, що вимагають автоматизації:

- генерація календаря заселення;
- генерація звіту поселення.

Використовувані документи та їх опис:

Клієнти – внутрішній документ, що містить інформацію про клієнтів.

Функція: облік клієнтів.

Тарифи – внутрішній документ, тариф проживання на проживання.

Функція: розрахунок вартості проживання.

Реєстрації – внутрішній документ, містить інформацію про всі реєстрації в готелі. Функція: облік реєстрації.

Звіт – внутрішній документ, складається на основі запиту по коду послуг, що надаються клієнту. Функція: облік послуг, що надаються клієнту.

1.3 Аналіз програмних систем аналогів

В даний час існує досить велика кількість PMS, призначених для управління. Більшість з них має схожий функціонал і є платними програмними продуктами. Далі розглянемо деякі з них.

Вново [1] – модульна система управління і контролю для готельного бізнесу, призначена для управління продажами, підвищення ефективності управлінських і збутових процесів в готелях, в хостелах, апартаментах і інших об'єктах розміщення (рис.1.1).

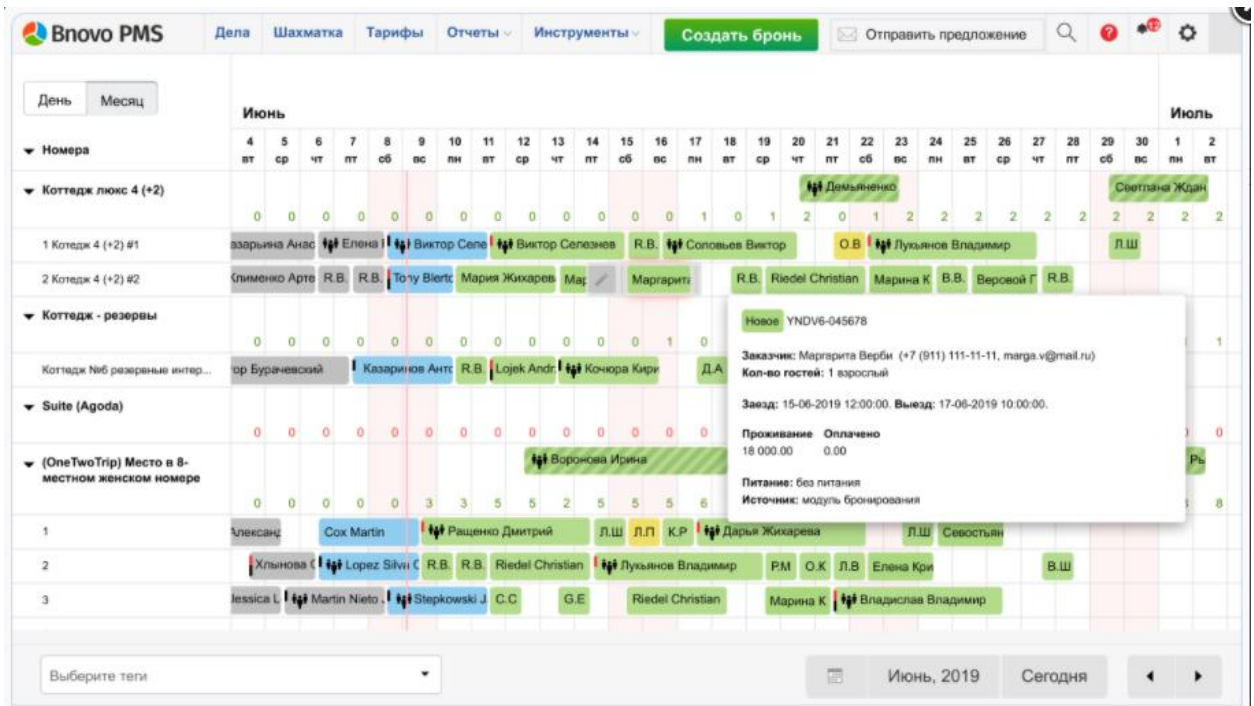


Рисунок 1.1 – Інтерфейс системи Вново

Система має всі функції управління та контролю: номерним фондом за допомогою зручної інтерактивної шахматки, бронюваннями, тарифами, продажами у всієї оффлайн і онлайн каналах, базою даних гостей. Система також дозволяє отримати звіти про ефективність діяльності готелю.

Вново дозволяє підключати більше різних каналів для збільшення продажів. Менеджер управління каналами дозволяє одночасно продавати номери на різних онлайн платформах, таких як Booking.com, AirBnB та інших. Дані про бронювання оновлюються автоматично у всіх каналах, з якими працює користувач. Крім того, система містить додаткові модулі: «Фінанси та звітність», «Прибирання» та ін. Система є платною. Самий дешевий варіант Вново коштує близько 10 у.о. на місяць.

Система Travelline є платним програмним продуктом управління роботою як міні-готелів і хостелів так і великих готельних комплексів [2].

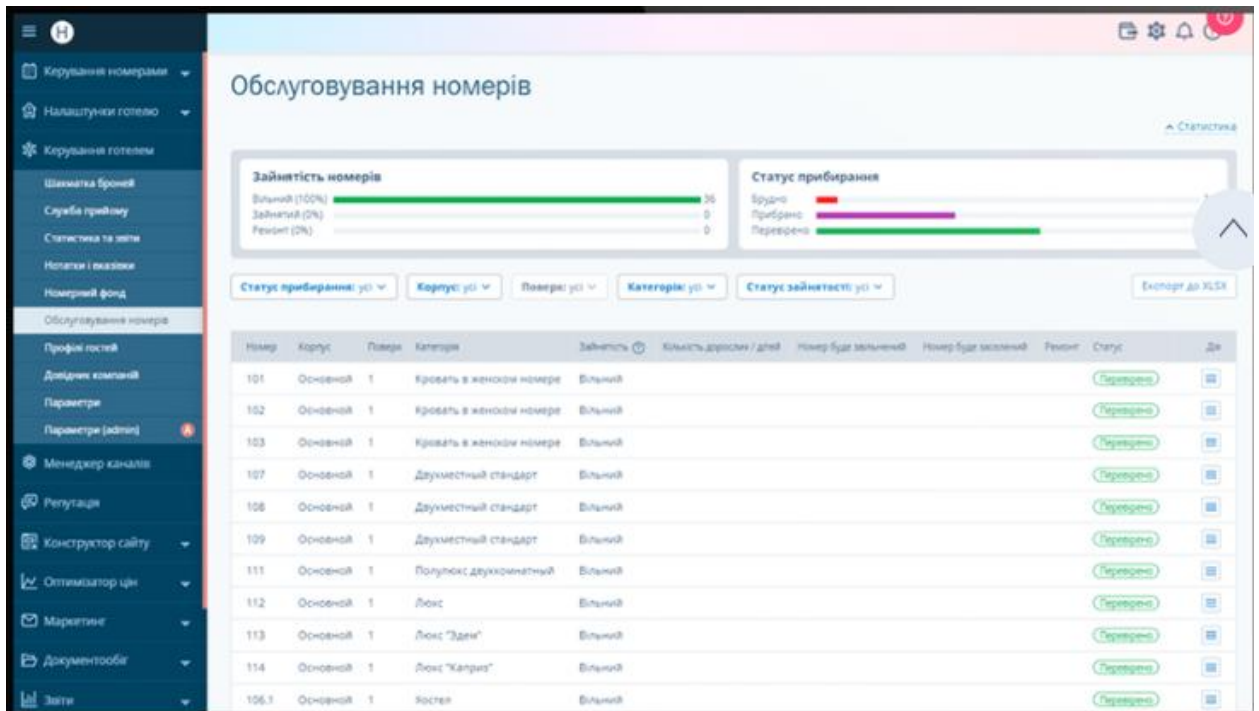


Рисунок 1.2 – Інтерфейс системи Travelline

Travelline є системою, що складається з декількох модулів. Серед модулів, що безпосередньо затребувані при управлінні роботою готелю, найбільш важливими є модуль TL Готель (350 грн./міс.), призначений для он-лайн-бронювання; модуль TL Channel Manager (950 грн./міс.) для управління каналами продажів; модуль TL WebPMS (350 грн./міс.) – автоматизована система управління міні-готелів та хостелів.

Система має зручну форму пошуку вільних номерів, маркетингові інструменти, що дозволяють отримати інформацію про знижки, кількість залишившихся номерів та час недавніх бронювань, календар доступності.

jSolutions – це система управління готелем, яка дозволяє вести комплексний контроль всіх бізнес-процесів закладу [3]. Система є платною. Вартість системи 359 грн/міс при оплаті на рік. Функціональні особливості хмарної системи jSolutions для ведення готельного бізнесу:

Управління номерним фондом. Створення номерного фонду готелю за типами: хостел (ліжко) або готельний (номер повністю). Формування структури за поверхами, приміщеннями, графічне розташування номерів готелю на схемі. Можливість додавання характеристик для кожного номера в вигляді переліку меблів і обладнання із зазначенням вартості. Для кожного номера можливо додавання різних опцій щодо графіка прибирання або міні-барі.

Налаштування тарифів. У розділі тарифи можливо налаштовувати ціни відповідно до сезону, кількістю гостей, кількістю додаткових місць.

Управління завантаженням номерів (рис.1.3). Наявність місць на вибраний період і календар завантаження з графічним відображенням в розрізі місяця.

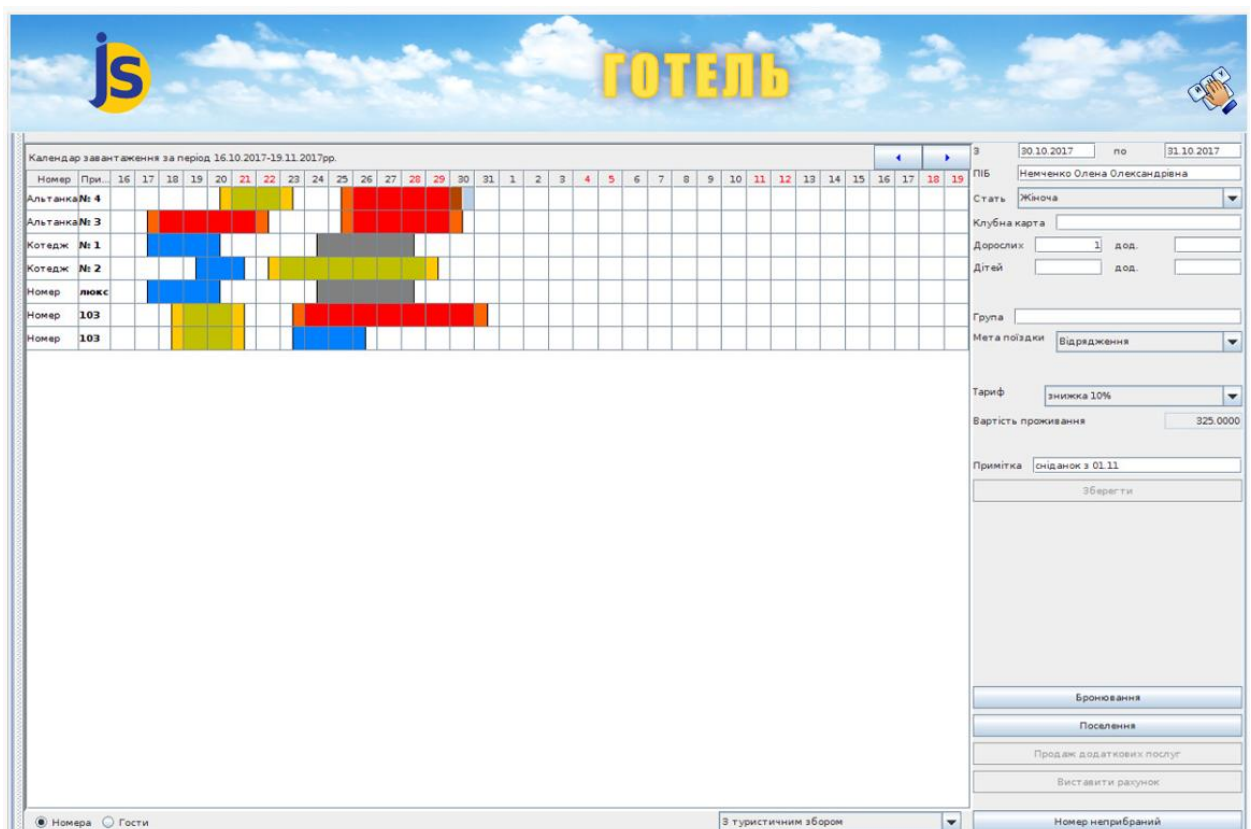


Рисунок 1.3 – Інтерфейс системи jSolutions

Інтеграція з Channel Manager. Channel Manager YieldPlanet – це агрегатор, понад 400 міжнародних каналів продажів таких як: booking, hotels24, dobovo, hotel360, airbnb для готелів, пансіонатів і баз відпочинку.

Управління комунікаціями з клієнтом. В системі є можливість роботи з побажаннями, запереченнями, скаргами клієнта через механізм рекламацій.

Планування обслуговування. Для своєчасного обслуговування номерів та оптимальної завантаженості персоналу в системі є можливість встановлення планового і фактичного часу прибирання із зазначенням переліку товарів для міні-бару.

Звіти та аналітика. Формування звітів, облік безготівкових платежів по рахунках клієнтів, аналіз взаєморозрахунків, виставлення рахунків, контроль оплати/передоплати, формування звітів про фінансову діяльність готелю, доступ до архіву рахунків, зведення по наданим послугам.

Порівняння перерахованих вище програм і системи, яка буде розроблена в кваліфікаційній роботі наведено у табл.1.1.

Таблиця 1.1 – Порівняння систем управління готелем

Критерії порівняння	Вново	Travelline	jSolutions	Власна розробка
Вартість, грн/міс	280	350	359	–
Простий інтерфейс	–	–	–	+
Наявність управління номерним фондом	+	+	+	+
Налаштування тарифів	+	+	+	+
Управління завантаженням номерів	+	+	+	+
Інтеграція з Channel Manager	+	+	+	–
Управління комунікаціями з клієнтом	+	+	+	+
Планування обслуговування	+	+	+	+
Звіти та аналітика	+	+	+	+

Висновки до розділу: під час виконання першого розділу кваліфікаційної роботи проведено аналіз предметної області, в рамках якої розглянуті аналоги і їх особливості. Переваги та недоліки розглянутих систем управління готельним бізнесом були взяті до уваги при розробці програмного продукту.

2 ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ

2.1 Функціональні вимоги

Автоматизована система готельного комплексу повинна виконувати наступні функції:

- дані, що вносяться користувачем повинні автоматично додаватися до бази даних;
- необхідний контроль правильності внесених даних, у разі помилки, дані не повинні додаватися до бази даних;
- повинна бути можливість додавати до бази даних нові записи, видаляти застарілі або непотрібні записи, редагувати існуючі;
- виконувати необхідні запити на отримання потрібних даних;
- програма повинна мати зручний і досить простий інтерфейс, який буде зрозумілий і не кваліфікованому користувачеві.
- отримання переліку та загальної кількості фірм, що забронювали місця в обсязі, не менше вказаного, за весь період співпраці, або за деякий період;
- отримання переліку та загального числа постояльців, які заселяються в номери із зазначеними характеристиками за деякий період;
- отримання кількості вільних або зайнятих або заброньованих номерів на даний момент;
- отримання відомостей про конкретний вільний номер: протягом, якого часу він буде пустувати і про його характеристики;
- отримання відомостей про кількість вільних номерів із зазначеними характеристиками;
- отримання списку зайнятих зараз номерів, які звільняються до даного терміну;
- отримання даних про обсяг бронювання номерів даною фірмою за вказаний період;
- отримання списку незадоволених клієнтів та їх скарги;

- отримання даних про прибуток готелю;
- отримання відомостей про постояльця із заданого номера: його рахунок готелю за додаткові послуги, які надходили від нього скарги, види додаткових послуг, якими він користувався;
- отримання відомостей про фірми, з якими укладені договори про бронь на зазначений період;
- отримання відомостей про постояльців, що найбільш часто відвідують готель, по всіх корпусах готелів, за певними будівлями;
- отримання відомостей про нових клієнтів за вказаний період;
- отримання відомостей про конкретну людину, скільки разів вона відвідувала готель, в яких номерах і в який період зупинялася, які рахунки оплачувала;
- отримання відомостей про конкретний номер: ким він був зайнятий в певний період;

Вхідною інформацією є вихідні дані про готель, особливості номерів, кількості вільних і зайнятих номерів, що вносяться до поля введення з клавіатури українською та англійською мовою порт'є або адміністратором.

Вихідні дані організовуються у вигляді відповіді на запит, який виводиться на головну сторінку. Після перегляду результату є можливість виконати наступний запит.

2.2 Вибір системи керування базами даних

Розглянемо найбільш відомі типи баз даних [4]:

- 1) Ієрархічні бази даних. З'явилися першими в 1968 році. Дані являють собою вузли, у яких визначено відношення «нащадок і предок», тобто кожен вузол може бути нащадком тільки одного елемента, що може викликати незручність.

- 2) Мережеві бази даних відрізняються від попередника тим, що у кожного вузла може бути кілька нащадків, проте, реалізація такої бази даних може бути досить трудомістка.
- 3) Реляційні бази даних (БД) являють собою сукупність пов'язаних таблиць, на перетині стовпців яких може зберігатися одна одиниця даних.
- 4) NoSql бази даних є новим вектором розвитку, відмінним від реляційних баз даних. Її відмітними ознаками є відсутність жорсткої типізації даних, тобто «колонки таблиць» таких баз даних можна міняти на льоту в ході виконання програми. Недоліком такого підходу є неможливість контролю типів даних з боку СКБД, а перевагою – збільшення гнучкості і масштабованості програми. Дані в таких базах даних представлено у вигляді агрегатів, тобто в вигляді об'єктів, це призводить до складнощів оптимізації під різні види запитів, але дані представлені в тому вигляді, в якому з ними буде працювати програма. Також варто відзначити, що NoSql гірше працюють з транзакційними запитами.

На сьогоднішній день в більшості випадків використовуються реляційні БД завдяки тому, що сама по собі модель даних інтуїтивно зрозуміла користувачеві, також вона має розвинений математичний апарат. Так що саме цю модель бази організації бази даних необхідно вибрати для реалізації цільового продукту. Все ж варто відзначити, що дана модель має відносно низький показник доступу до даних, але даний недолік повністю компенсується позитивними сторонами даної моделі.

Одними з найпопулярніших систем керування базами даних, що підтримуються більшістю хостинг-провайдерів, є MySQL і PostgreSQL, які займають 2 і 4 місце за кількістю запитів в пошуковій системі Google за останній рік (табл. 2.1).

Таблиця 2.1 – Рейтинг СКБД у квітні 2021 року за версією TOPDB [5]

Позиція	СКБД
1	Oracle
2	MySQL
3	SQL Server
4	Microsoft Access
5	PostgreSQL
6	MongoDB
7	Firebase
8	Splunk
9	Elasticsearch
10	Redis
	інші

MySQL – реляційна СКБД від корпорації Oracle, яка спочатку призначалася для роботи з великими масивами, але зараз використовується з різними веб застосунками як сервер, до якого звертаються віддалені клієнти [6]. Дані зберігаються у вигляді логічно пов'язаних таблиць, доступ до яких здійснюється за допомогою мови запитів SQL. Робота з БД ведеться трьома способами: через графічний інтерфейс phpmyadmin, через MySQL Workbench, або через консоль.

Переваги MySQL:

- швидкість і надійність;
- простота у використанні;
- відкритість коду;
- може працювати з великими обсягами даних;
- переносимість.

Недоліки MySQL:

- підтримує не всі SQL функції;

- до 5-ї версії не підтримувала функціонал тригерів, курсорів, що стало причиною відмови від MySQL [6];
- слабкий захист у порівнянні з іншими конкурентами.

PostgreSQL – об'єктно-реляційна система управління базами даних, основою якої є некомерційна СКБД Postgres – проект з відкритим вихідним кодом, що розробляється з 1986 року [7]. Її особливістю є використання об'єктно-орієнтованих принципів, всі таблиці представлені у вигляді успадкованих об'єктів, робота з даними відбувається за допомогою об'єктно-орієнтованих функцій. Максимально відповідає стандартам SQL.

Переваги PostgreSQL:

- повна підтримка SQL;
- об'єктно-орієнтований принцип.

Недоліки PostgreSQL:

- відносна складність розробки;
- мала кількість хостинг-провайдерів, які підтримують PostgreSQL.

SQLite – вільно розповсюджувана БД, що випущена в 2005 р, не використовує парадигму клієнт-сервер і являє собою бібліотеку, що легко вбудується в будь-яке застосування, в зв'язку з чим має високу швидкість і продуктивність [8].

Переваги SQLite:

- підтримує набір команд SQL;
- переносима, вся база зберігається в одному файлі;
- швидкодія.

Недоліки SQLite:

- немає можливості створення декількох користувачів з різними правами доступу;
- допускається лише одна операція запису одноразово;
- дуже мало типів даних, наприклад, щоб зберігати значення дати і часу в unix-форматі, доводиться використовувати строковий тип даних і в коді перетворювати в long.

Виходячи з того, що в SQLite відсутні множинні транзакції, її не слід вибирати в якості СКБД для інформаційної системи, також в даній системі передбачається багато працювати з датами, робота з якими не підтримується в SQLite.

Таблиця 2.2 – Порівняльний аналіз СКБД

	MySQL	PostgreSQL	SQLite
Архітектура	Клієнт-сервер	Клієнт-сервер	Вбудована БД
Підтримка веб-хостингами (веб-серверами)	Дуже гарна. Переважна більшість хостингів PHP мають підтримку MySQL	Задовільна. Підтримка PostgreSQL часто зустрічається	Задовільна. Підтримка SQLite часто зустрічається, можна направити запит в підтримку про установку необхідного модуля
Документація	Англомова на офіційному сайті		
Простота розробки та адміністрування	Адміністрування через консоль, безліч додатків для різних платформ для локального адміністрування, адміністрування через веб-сервер		
Надійність збереження даних	Надійність забезпечується сервером БД. Є механізми резервного копіювання	Вся БД зберігається в одному файлі	

Так, в якості основної СКБД для зберігання даних на сервері, була обрана MySQL, тому що дана СКБД підтримується більшістю хостинг-провайдерів, а значить при зміні хостингу не повинно виникнути ніяких проблем.

2.3 Вибір фреймворків та мови програмування

Найбільш популярні мови програмування для розробки web застосунків представлені в табл. 2.3

Таблиця 2.3 – Популярність мов програмування для web-розробки по числу запитів [9]

№	Мови програмування
1	PHP
2	Python
3	JavaScript
4	Java
5	Ruby

Java Script – мова програмування, яка створена фірмою Netscape Communication Corporation [10,11]. мова об'єктно-орієнтована. Розробнику доступні численні об'єкти, такі, як документи, гіперпосилання, форми, фрейми тощо. Об'єкти характеризуються властивостями і методами.

Для розробки серверних застосунків не можна використовувати чистий Java Script. Тому і були розроблені такі JavaScript фреймворки, як: React Native, Apache Cordova і інші. Проте створення великих проектів використовуючи тільки цю мову, може привести до збільшення складності розробки, проте цей інструмент підходить як засіб для управління об'єктною моделлю документа.

Переваги JavaScript:

- кросплатформеність;
- безліч фреймворків;
- велика кількість документації.

Недоліки JavaScript:

- застосунок може бути тільки гібридним;

- не дивлячись на кроссплатформенність, у кожній платформі свої особливості, і написати єдиний для всіх платформ код не вдасться.

Основні фреймворки мови JavaScript для роботи на клієнтській частині: JQuery, React та Vue Js.

PHP – скриптова мова загального призначення, інтенсивно застосовується для розробки веб-застосунків, розроблена в 1995 році групою розробників в рамках проекту з вихідним кодом [12]. В даний час підтримується переважною більшістю хостинг-провайдерів і є одним з лідерів серед мов, що застосовуються для створення динамічних веб-сайтів.

Переваги PHP [13]:

- простота, багато конструкцій мови запозичені з C, Perl;
- велика кількість доступних бібліотек;
- гнучкі засоби безпеки;
- немає залежності від браузерів;
- PHP безкоштовно поширюється;
- кроссплатформеність, PHP може бути запущений в будь-якій операційній системі.

Недоліки PHP:

- більша гнучкість мови, яка є надлишковою, так як при її використанні ускладнюється підтримка проекту і його супровід;
- дана мова узконаправлена, тобто крім побудови серверної частини веб-застосунків, вона ніде не використовується.
- непослідовність синтаксису.

Розробляти систему на чистому PHP, з урахуванням всіх вимог до системі достоточно складно. Як вже було описано раніше, для розробки системи однієї мови мало, до мови потрібно вибрати інструмент. Одним з самих популярних Інструменти для php є Laravel [14] - фреймворк з відкритим вихідним кодом, створений для спрощення розробки web сайтів. Laravel підтримує структуру MVC, як ORM виступає Eloquent. В якості шаблонізатора виступає Blade.

Java є одним з найбільш затребуваною і популярною мовою програмування у всьому світі. Зараз мова підтримується компанією Oracle [15,16]. Ключовою особливістю мови є її транслювання в спеціальний байт-код, який не залежить від платформи. Потім цей байт код виконується спеціальною Java-машиною. Подібна архітектура служить доказом крос-платформності і апаратної переносимості програм, написаних на Java.

Переваги Java:

- офіційна мова розробки під ОС Android, саме ця мова потрібно розглядати в першу чергу як інструмент Android модулів для системи;
- повна і доступна документація;
- великий відсоток розробників, які використовують цю мову;
- простота інтеграції сторонніх модулів – за допомогою засобів управління залежностей, таких як Maven або Grable можна швидко збирати будь-які проекти будь-якої складності;
- статична типізація – у кожній змінній жорстко визначено тип. Мови з динамічною типізацією здаються більш простими і гнучкими, але код на даній мові складніше розширювати. Так само систему зі статичної типізацією набагато легше оптимізувати.

Недоліки Java:

- програми на мові Java займають набагато більше оперативної пам'яті, ніж програми на конкурентних мовах програмування. Це не дивно, так як дані програми працюють під керуванням JVM, тому коли запускається java-програма, то разом з нею запуситься і JVM, яка резервує пам'ять із запасом;
- мова може використовуватися тільки для серверної частини системи. Клієнтська частина може бути реалізована на технології Applet або технології Java Net Start. Але вони вже застаріли.

Java включає в себе JavaSE і JavaEE [17]. Перший стандарт включає в себе основу мови: колекції, узагальнення, класи для роботи з потоками (thread), потоки даних (stream API), класи для роботи зі звуком, побудовою

застосунків з графічним інтерфейсом. Цей стандарт є ядром для інших напрямків Java.

JavaEE-стандарт для побудови enterprise рішень і для побудови web застосунків. Основна структурна одиниця це сервелет [17]. Даний клас є обробником http-запитів. Як параметри він приймає класи http-запиту і http-відповіді, тобто необхідно самому формувати html, передаючи потрібні рядки потоку виведення http-відповіді. Також не можна забувати про формування заголовків.

Для побудови системи даний підхід є сильно низькорівневим, але крім сервелету даний клас включає інші технології, які перетворюються в сервелет, наприклад JSP-сторінки.

Проте, у даного засобу є велика перевага, а саме безліч класів і технологій, найпоширеніші – це Enterprise Java Bean і Dependency injection (створення зв'язку агрегації і композиції між класами). EJB – це класи для роботи з даними. Вони можуть бути різні, можуть класифікуватися по збереженню стану: Statfull, Stateless і Singleton. Так ж варто згадати, що JavaEE дозволяє організувати роутинг своїми базовими засобами, так само плюсом є підтримка паттерна проектування MVC.

JavaEE все ж не є оптимальним рішенням, але існує Java Spring фреймворк [17], який базується на даному стандарті і при цьому розширює його. Він включає Spring MVC (для розробки сайтів), Spring Boot (для легкого підключення інших модулів). Spring підтримує свої Beans і теж підтримує Dependency injection, JPA Hibernate. Hibernate це ORM від Spring, що дозволяє працювати з СКБД за допомогою об'єктної моделі. Працювати можна як в реляційними базами даних, так і з NoSql, при цьому робота з відображеними об'єктами проводиться за допомогою анотацій. Так само не можна не згадати про нову мову запитів JPQL, яка працює вже з Java-об'єктами.

Це робить застосунок масштабованим. Так само варто згадати про можливість роботи з REST сервісами, а саме організувати взаємодією java script і Spring за допомогою повідомлень в форматі JSON.

Звичайно, у даного фреймворка є великі функціональні можливості.

До мінусів цього фреймворка можна віднести підвищений (в порівнянні з Laravel (php)) вимоги до апаратної частини ЕОМ. Так само, як і JavaEE для розгортання Spring потрібно контейнер сервлетів.

Незважаючи на все це, Spring володіє високою швидкістю і надійністю. Його застосовують в таких сферах як банківський enterprise. Не дивлячись на підвищені вимоги до апаратури та системи, Spring має перевагу перед Laravel і Django зростанням навантаження на сервер і в швидкодії. Так само варто відзначити, що на відміну від своїх конкурентів, Spring це щось більше ніж фреймворк для веб розробки (за це відповідає тільки один модуль це Spring MVC), так на Spring можна працювати з хмарними технологіями, або підключити кластер серверів до проекту.

Python – це універсальна мова високого рівня, яка з'явилася у 1991 року [18]. Широко застосовується як інтерпретована мова для скриптів різного призначення. Елегантний дизайн і ефективний, дисциплінуючий синтаксис цієї мови полегшують програмістам спільну роботу над кодом. Python дозволяє поєднувати процедурний підхід до написання коду з об'єктно-орієнтованим і функціональним.

Переваги Python:

- відкрита розробка;
- простий у вивченні, особливо на початковому етапі;
- особливості синтаксису стимулюють програміста писати код. Що добре читається;
- надає засоби швидкого прототипування і динамічної семантики;
- має велике співтовариство, позитивно налаштоване по відношенню до новачків;
- безліч корисних бібліотек і розширень мови можна легко використовувати в своїх проектах завдяки гранично уніфікованому механізму імпорту та програмним інтерфейсами;

- механізми модульності добре продумані і можуть бути легко використані;
- абсолютно все в Python є об'єктами в сенсі ООП, але при цьому об'єктний підхід не нав'язується програмісту, на відміну від мови Java.

Недоліки Python:

- не надто вдала підтримка багатопоточності;
- створено не так багато якісних програмних проектів в порівнянні з іншими універсальними мовами програмування, наприклад, з Java;
- відсутність комерційної підтримки засобів розробки (хоча ця ситуація з часом змінюється);
- початкова обмеженість засобів для роботи з базами даних.

Найпопулярнішим Python-фреймворком є Django. розберемо його сильні і слабкі сторони.

Переваги Django [19]:

- відносно швидка швидкість розробки, як формування структури, так і реалізація проекту;
- наявність Python дає перевагу щодо PHP (але не Java) в довгострокових проектах, так як Python володіє більшими, ніж PHP можливостями.

Недоліки Django:

- маршрутизація проводиться із зазначенням URL;
- збільшення складності роботи з фреймворком в міру збільшення релізів;
- відносна складність адаптації ORM Django для потреб проекту;
- Django – монолітний фреймворк.

Як висновок можна сказати, що Django (Python) підходить як інструмент для розроблюваної системи. Він включає в себе простоту і швидкість розробки як в Laravel і при цьому на ньому можна розробляти складніші системи, причому не обмежені web технологіями.

На основі проведеного обґрунтування можна зробити висновки, що по швидкості розробки Laravel і Django займають лідируючі місця і при цьому

вимагають менших апаратних витрат, ніж Spring. Spring ж будучи «важким» інструментом володіє великими можливостями. Для вибору правильного інструменту необхідно оцінити час і ресурси, виділені на розробку, а також оцінити майбутнє системи. Планується, що функціонал системи повинен повільно, але монотонно доопрацьовуватися цільовою компанією з плином тривалого часу. Так як дана система повинна бути максимально надійною, то вибрати найкраще Spring. В якості клієнтського інструменту підходить JQuery або Vue Js, За рахунок їх простоти і високої швидкості розробки.

2.4 Вибір стильових інструментів

У системах провідних обмін за допомогою протоколу http крім самого каркаса сторінки важливий і її стиль. Стиль можна задати за допомогою атрибутів html, правда змішування стилів і елементів сторінки призводить до зменшення читання розмітки даної сторінки і до ускладнення змін і модифікацій даного документа. Для вирішення цієї проблеми використовуються css стилі, які можуть працювати з властивостями об'єктів і можуть розміщуватися в окремому файлі.

Для побудови адаптивного дизайну використовують CSS-фреймворки. Звичайно, можна обійтися і без них використавши стандартний CSS, але такий підхід не є оптимальним, так як це призведе до витрат часу, так само використовуючи стандартні класи CSS-фреймворка (там, де це необхідно) можна створити легко модифікуються сторінку.

До недоліків даний фреймворків можна віднести залежність проекту від CSS-класів, а також необхідність підключення додаткових файлів. Так само варто відзначити, що створення власних дизайнів може дати більше можливостей при правильному підході, так що найсильнішим підходом є комбінування класів фреймворка і власних.

Розглянемо докладніше деякі CSS-фреймворки.

Bootstrap [20] – CSS-фреймворк, призначений для адаптивної верстки сайтів як для мобільних пристроїв так і стаціонарних ПК.

Переваги Bootstrap:

- адаптивний дизайн для будь-яких типів дисплеїв, від самих маленьких до гігантських;
- найбільш популярний фреймворк;
- підтримка ґратчастої верстки і flex-контейнерів, як фіксованих, так і гумових;
- величезна різноманітність UI-елементів.

Недоліки Bootstrap:

- великий розмір файлів;
- вимога наявності JQuery.

Skeleton – CSS-фреймворк для швидкої верстки максимально простих фіксованих дизайнів.

Переваги Skeleton:

- мінімалізм;
- простота.

Недоліки Skeleton:

- можливість працювати тільки з фіксованими сітками;
- даний фреймворк поступається в популярності;
- підтримують не всі браузерами.

Foundation V3 – CSS-фреймворк для ґратчастої верстки з максимальною надійністю.

Переваги Foundation:

- підтримка надійних «гумових» сіток;
- підтримка різноманітних UI-елементів, правда, ця різноманітність поступається Bootstrap;

Недоліки Foundation:

- відсутність підтримки фіксованих контейнерів;
- деякі плагіни вимагають наявності JQuery.

На підставі вищезгаданого, можна зробити висновок, що найбільш оптимальним рішенням для даної системи є Bootstrap, так як він може запропонувати велику гнучкість в побудові каркаса документа.

2.5 Обґрунтування вибору web-сервера

Web-сервер виконує роль сховища і менеджера ресурсів. Для вибору web-сервера необхідно, в першу чергу визначити інструменти розробки, так як, не всі сервера підтримують роботу з певними мовами програмування. Наприклад, для написання системи за участю Java Enterprise Edition необхідний контейнер сервлетів. Так само не всі сервери підтримують динамічне формування даних, наприклад сервер Nginx. Правда, для його використання можна підключити додатковий сервер.

Розглянемо деякі найбільш популярні web-сервери згідно рейтингом популярності, представленому в табл.2.4.

Таблиця 2.4 – Рейтинг популярності web-серверів

№	Назва	Частка ринку, %
1	Apache	40,92
2	Nginx	23,19
3	IIS	16,99
4	ApacheTomkat	0,16
5	LightHttpd	0,13

Веб-сервер Apache розробляється і підтримується відкритою спільнотою розробників під егідою Apache Software Foundation. До теперішнього часу Apache є найпоширенішим web сервером. Проте його популярність повільно падає. Ядро Apache написано на мові C, включає в себе основні функціональні можливості, такі як обробка конфігураційних файлів, протокол HTTP

і система завантаження модулів. Apache підтримує наступні мови програмування: PHP, Python, Ruby, Perl, ASP, Tcl.

Переваги Apache:

- безкоштовний навіть для використання в комерційних цілях;
- надійний, стабільний програмне забезпечення;
- часто оновлюваний, регулярні патчі безпеки;
- гнучкий, завдяки своїй модульній структурі;
- простий в експлуатації, навіть для початківців;
- крос-платформний (працює однаково добре на Unix і на Windows-серверах);
- працює для сайтів на WordPress відразу після установки без додаткових налаштувань;
- велике співтовариство і легко доступна підтримка в разі будь-якої проблеми.

Недоліки Apache:

- проблеми продуктивності на сайтах з великим трафіком;
- занадто багато параметрів конфігурації можуть привести до вразливостей в безпеці.

Web-сервер Nginx працює на Unix-подібних операційних системах, Nginx позиціонується виробником як простий, швидкий і надійний сервер, не переобтяжений функціями. Застосування Nginx доцільно перш за все для статичних веб-сайтів і як зворотного проксі-сервера перед динамічними сайтами. У Nginx робочі процеси обслуговують одночасно безліч з'єднань, мультиплекуючи їх викликами операційної системи. Розібраний запит послідовно обробляється ланцюжком модулів, що задається конфігурацією.

На відміну від Apache, Nginx працює в рамках одного потоку, але використовує неблоковані сокети. Дана модель дуже ефективно використовує процесор і пам'ять, але досить складна в реалізації.

Переваги Nginx:

- шифрування, стиснення, підтримка багатьох сайтів за однією IP-адресу і інші можливості, доступні в більшості веб-серверів;
- міжсистемних, малий розмір, простота конфігурації, масштабованість;
- використання переваги від асинхронної системи введення-виведення, що на практиці означає економію ресурсів системи і вигідно позначається в разі великих навантажень;
- вбудована стійкість до багатьох видів DoS і DDos-атак.

Недоліки Nginx:

- не обробляє динамічний вміст. Для обробки PHP або інших динамічних запитів потрібно використовувати зовнішній інтерпретатор і чекати поки він поверне результат обробки.

Web-сервер Lighttpd розробляється з розрахунком на швидкість і захищеність, а також відповідність стандартам. Це вільне програмне забезпечення, яке розповсюджується за ліцензією BSD. Lighttpd працює в Linux і інших Unix-подібних операційних системах, а також в Microsoft Windows.

Проект Lighttpd почався з прагнення автора реалізувати веб-сервер, який міг би витримати одночасно 10 тисяч з'єднань. Lighttpd використовує так звану асинхронну обробку мережевих з'єднань. Завдяки цьому завантаженість сервера (на відміну від Apache) при доступі до файлів на диску не залежить від кількості поточних з'єднань.

Web-сервер Tomcat – контейнер сервлетів з відкритим вихідним кодом, що розробляється Apache Software Foundation [21].

Даний сервер не займає лідируючих місць, тим не менш, у нього є особливість, а саме підтримка контейнер сервлетів, а це значить, що на ньому може розміститися Spring або Java програма.

Для вибору відповідного web-сервера необхідно, перш за все оцінити середнє число запитів до сервера, наявність динамічного формування HTML документів та цільову платформу. Так, з огляду на той факт, що число підключень буде менше 10000 і формування близько 90% документів буде фор-

муватися динамічно, то виходячи з перерахованих вище достоїнств серверів, можна сказати, що web сервер Apache і Lighttpd будуть більш кращі, ніж web-сервер Nginx, незважаючи на більш якісну систему безпеки (в порівнянні з Apache). Nginx може бути обраний, але тільки разом з іншим сервером, що ускладнить задачу.

У підсумку, якщо вибрати в якості інструменту розробки Laravel (php) або Django (Python), то вибрати варто саме Apache, а якщо розробляти з допомогою Spring, тоді необхідно використовувати Apache Tomcat. Правда, web-сервер Nginx дуже часто використовується в spring системах в якості проксі-сервера.

Провівши аналіз найбільш популярних веб серверів і а також мов програмування, вибір пав на apache Tomcat, мова JavaScript (jQuery) і Spring (Java). Для досягнення поставленої задачі і цілі, цей вибір є найбільш оптимальним з огляду на велику кількість документації та вільно розповсюджуваних бібліотек, платформ даних мов. В якості інструменту завдання стилів був обраний CSS фреймворк Bootstrap.

Таким чином, були обрані всі інструменти, які будуть необхідні для розробки системи, починаючи від платформи закінчуючи аналізом місця розгортання програми.

3 ПРОЕКТУВАННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ВЕБ-СИСТЕМИ

3.1 Діаграма процесів

Діаграма процесів дозволить наочно представити функціонал web-системи. Можна виділити наступні процеси:

- пошук номеру;
- здійснення бронювання;
- перевірка бронювання.

Розроблена діаграма процесів для web-системи, що проектується наведена на рис.3.1.

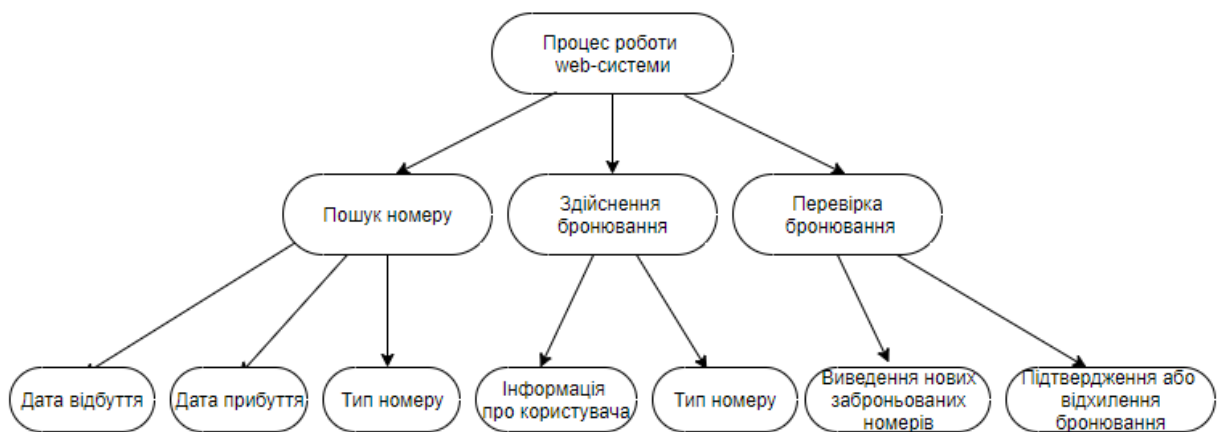


Рисунок 3.1 – Діаграма процесів web-системи

3.2 Діаграма прецедентів

Діаграма прецедентів чи варіантів використання (Use Case Diagram) визначає функціональне призначення модельованої системи або предметної області. Дана діаграма відображає безліч акторів, що взаємодіють з проектованою системою (програмним засобом) за допомогою варіантів використання. Таким чином, основними елементами діаграми варіантів використання є актор і варіант використання.

Актор – це зовнішня по відношенню до модельованої системи сутність, що взаємодіє з системою для вирішення деяких завдань. Як актор може використовуватися людина, інша система, пристрій або програмний засіб.

Варіант використання визначає деякий набір дій (операцій), які повинні бути виконані моделюється системою або програмним засобом при взаємодії з актором.

В роботі була створена діаграма прецедентів для web-системи. На діаграмі зображені два актора: користувач та адміністратор (рис.3.2 та рис.3.3).

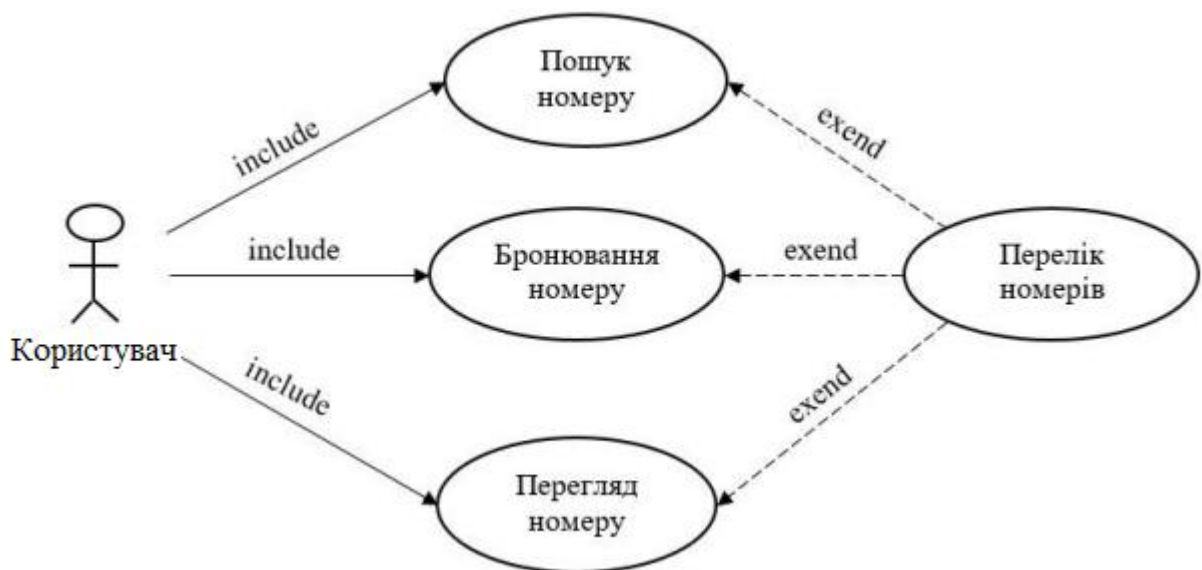


Рисунок 3.2 – Діаграма варіантів використання для користувача

3.3 Діаграми станів, діяльності та послідовності

Далі була створена діаграма діяльності для web-системи (рис. 3.4).

Варіанти використання системи:

- бронювання номеру;
- пошук;
- зворотний зв'язок.

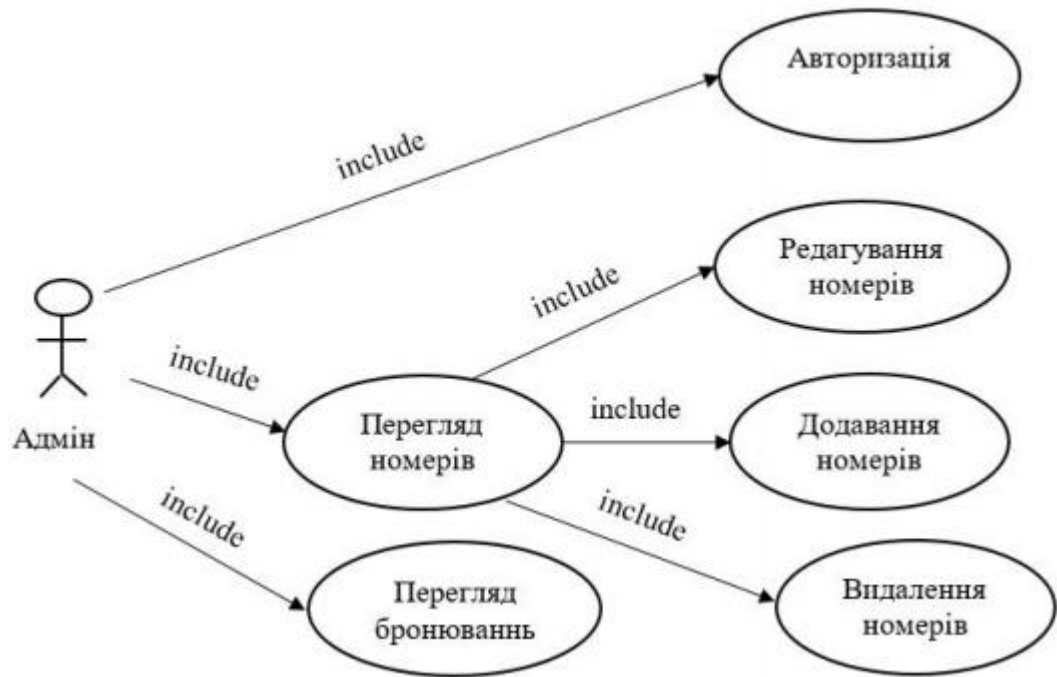


Рисунок 3.3 – Діаграма варіантів використання для адміністратор

Діаграма діяльності UML (activity diagram) показує динамічні аспекти поведінки системи. По суті, ця діаграма являє собою блок-схему, яка наочно показує, як потік управління переходить від однієї діяльності до іншої. Тобто діаграма діяльності – це технологія, що дозволяє описувати логіку процедур, бізнес-процеси і потоки робіт. Діаграму діяльності можна вважати окремим випадком діаграми станів.

На рис.3.4 наведена діаграма діяльності для процесу бронювання номеру. На ній вказані всі дії, які потрібно виконати користувачу системи, щоб забронювати готельний номер.

На рис.3.5 представлено діаграму станів для пошуку.

Діаграма діяльності зворотнього зв'язку представлена на рис.3.6.

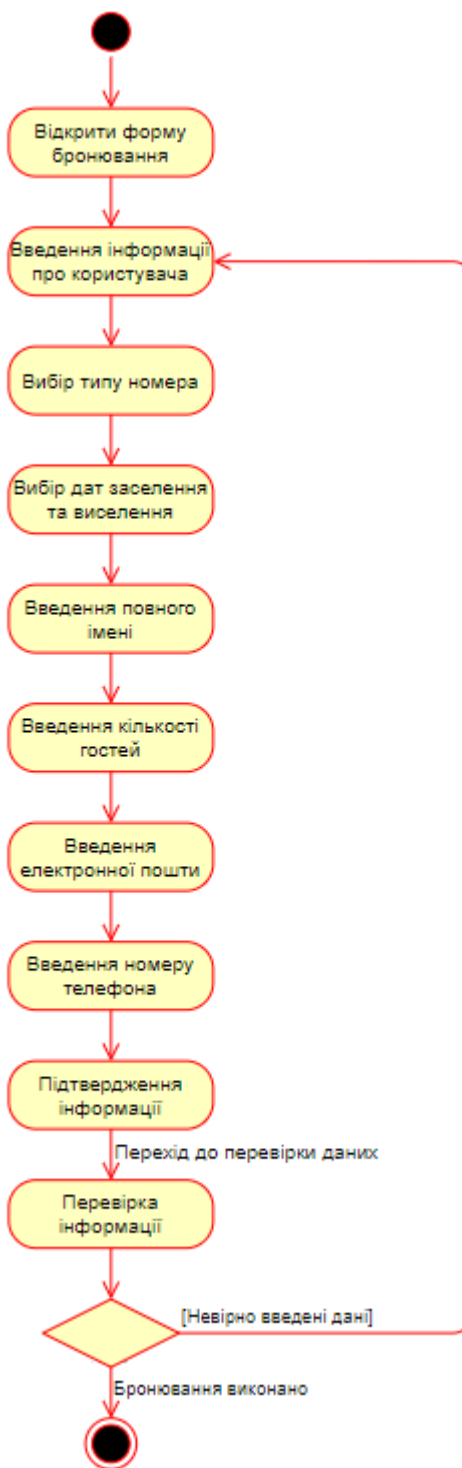


Рисунок 3.4 – Діаграма діяльності для бронювання номеру

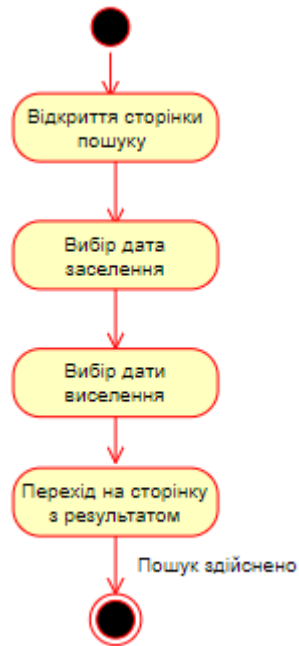


Рисунок 3.5 – Діаграма станів пошуку

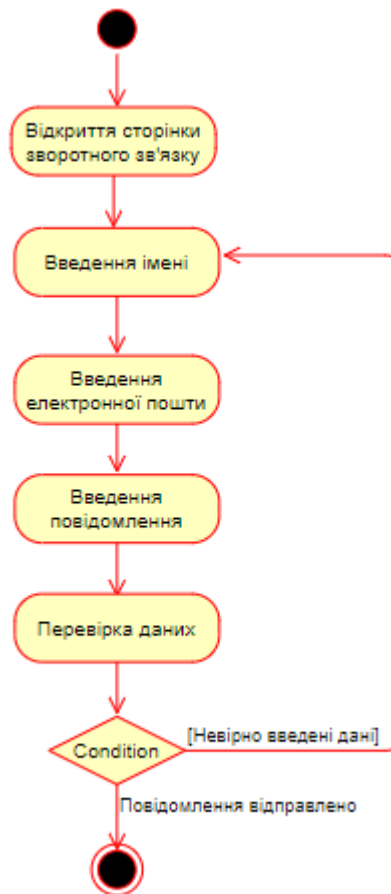


Рисунок 3.6 – Діаграма діяльності зворотного зв'язку

3.4 Побудова моделі потоків даних

Аналіз предметної області управління готельним комплексом проведемо за допомогою CASE-засобу VPwin з використанням двох методів IDEF0 і DFD. Вибір даних методів обумовлений такими факторами:

- IDEF0 – необхідністю визначення відповідних областей в досліджуваній системі, на яких необхідно сфокусувати увагу в першу чергу (моделювання діяльності готелю з метою побудови деякої інформаційної системи);
- DFD – дані діаграми використовуються для опису документообігу та обробки інформації. Вони є доповненням до моделі IDEF0 для більш наочного відображення поточних операцій з документами в системах обробки інформації.

На контекстній діаграмі A0 відображена система управління процесом: основні вхідні потоки, вихідні дані, ресурси та управління. Керуючими впливами є законодавство та інструкції. Ресурсами – співробітники готелю і готельні номери. Контекстна діаграма наведена у додатку А.

Report for Diagram: A-0, Управління готельним комплексом

Activity Name: Управління готельним комплексом

Link Name: Дані користувача

Link Name: Дані клієнта

Link Name: Рахунок

Link Name: Законодавельство

Link Name: Графік заселення

Link Name: Співробітник

Link Name: Побажання користувачів

Link Name: Готельні номери

Link Name: Запит про стан номера

Link Name: Інформація про номер

Link Name: Звіт про проживання

Організація роботи фірми – сукупність технологічних процесів. Основ-

ним результатом цього технологічного процесу є надання різних послуг. Процес роботи поділяється на три безперервних потоку. Були визначені основні процеси:

- організація доступу до системи;
- управління календарним планом;
- реєстрація клієнта.

Дані блоки відображені на діаграмі першого рівня A0 і представлені у додатку Б. Вхідними даними моделі будуть:

- дані користувачів;
- дані клієнтів;
- запити про стан номерів.

Вихідними даними моделі будуть:

- рахунок фактура;
- інформація про номер;
- звіт про проживання;
- графік проживання.

Наступні діаграми – це окремі випадки декомпозиції підсистем розглянутого процесу. У них виділяються основні процеси. Нижче наведені звіти по кожній з діаграм: A1, A2, A3. Дані діаграми наведені у додатку Б.

Report for Diagram: A1, Організація доступу до системи

Activity Name: Перевірка даних

Activity Name: Видача прав доспуа

Link Name: Побажання користувачів

Link Name: Логін і пароль

Link Name: Авторизований користувач

Link Name: Помилка доступу

Link Name: Користувачі

Link Name: Дозвіл доступу

Data Store Name: Користувачі системи

External Name: Користувач

Report for Diagram: A2, Управління календарним планом

Activity Name: Формування списку поселенців

Activity Name: Отримання інформації про номер

Activity Name: Формування звіту

Link Name: Дані користувача

Link Name: Законодательство

Link Name: Графік заселення

Link Name: Авторизований користувач

Link Name: Співробітник

Link Name: Побажання користувачів

Link Name: Готельні номери

Link Name: Запит про стан номера

Link Name: Інформація про номер

Link Name: Інформація про порожньому номері

Link Name: Інформація про заселеному номері

Link Name: Звіт про проживання

Data Store Name: зареєстрованих номера

External Name: Користувач

Report for Diagram: A3, Реєстрація клієнта

Activity Name: Отримання даних

Activity Name: Заповнення інформації

Activity Name: Формування у рахунку

Link Name: Дані клієнта

Link Name: Рахунок

Link Name: Законодательство

Link Name: Авторизований користувач

Link Name: Побажання користувачів

Link Name: Інформація про номер

В логічній моделі виділено чотирнадцять сутностей: Service, Service, Documents, Reserv_Service, Settings, Level_access, Bad_type, room_bad, Clerck, Clients, Reserv, Reserv_room, Rooms, Tariff, Room_type.

Кожна з сутностей має свій набір атрибутів і первинних ключів, які відображені на ER-діаграмі. Таким чином, визначається інформація, яка зберігається в конкретній сутності і в конкретному атрибуті, що забезпечує повну інформаційну підтримку для виконання всіх функцій, закладених в інформаційну систему [12].

Нижче опишемо інформацію, що міститься в кожній з сутностей, наведеною на ER-діаграмі (рис.3.8).

Сутність Service містить інформацію про додаткові послуги, що надаються готелем.

Сутність Documents містить інформацію про видані рахунки.

Сутність Reserv містить інформацію про бронювання і реєстрацію.

Сутність Reserv_Service зберігає інформацію про додаткові послуги, що надані клієнту в межах одного рахунку.

Сутність Reserv_room зберігає список зареєстрованих номерів клієнтом.

Сутність Settings зберігає глобальні налаштування програми.

Сутність Rooms зберігає інформацію про кімнати.

Сутність Tariff містить список тарифів для певного типу кімнат.

Сутність Clients зберігає інформацію про клієнтів.

Сутність Clerck зберігає інформацію про користувачів програми.

Сутність Bad_type зберігає інформацію про типи ліжок.

Після створення повної і адекватної логічної моделі можна приймати рішення про вибір платформи реалізації. Вибір платформи – складне завдання, що залежить від вимог до використання даних і стратегічних принципів формування архітектури корпорації. Розробник моделі даних і адміністратор бази даних використовують логічну модель, вимоги до використання та стратегічні принципи формування архітектури корпорації для розробки фізичної

моделі даних.

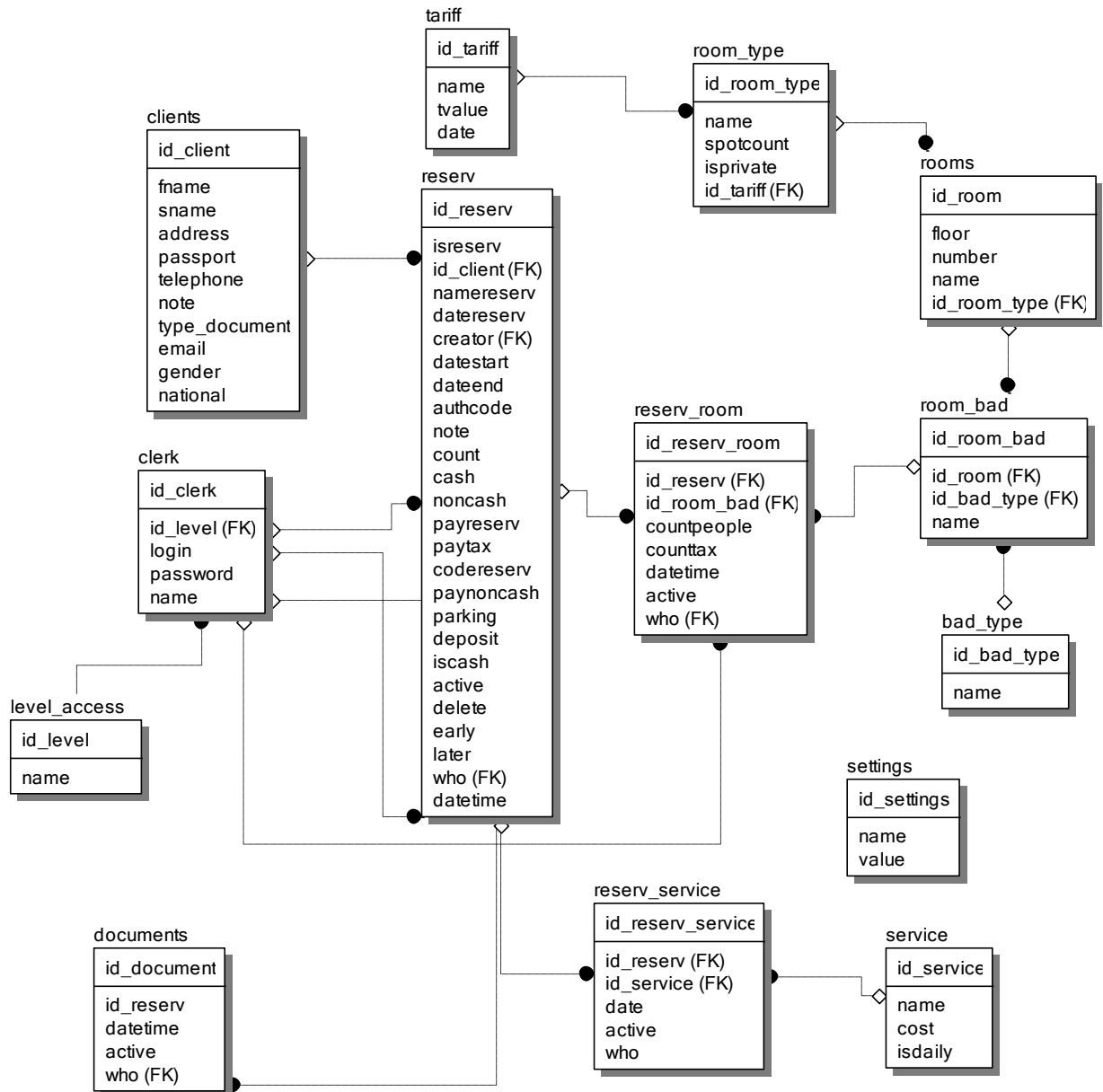


Рисунок 3.8 – Діаграма ERD – рівень атрибутів

Поділ моделі даних на логічну і фізичну вирішує важливу задачу найбільш оптимального представлення даних, зручного для розуміння як фахівцями, так і простим користувачами.

Фізична модель даних залежить від конкретної СКБД, фактично будучи відображенням системного каталогу. У фізичній моделі міститься вся інформація про всі об'єкти БД. Виходячи з цього, можна стверджувати, що од-

на і та ж логічна модель може бути представлена кількома фізичними. Представлені в фізичній моделі атрибути несуть конкретну інформацію про конкретних фізичних об'єктах.

У ERwin фізична модель є графічним представленням реально реалізованої бази даних. Фізична база даних буде складатися з таблиць, стовпців і зв'язків.

3.5 Структура і схема бази даних

На основі діаграми ERD, наведеною на рис.3.8, розроблена база даних для MySQL.

Розроблена база даних складається з 12 таблиць. Нижче наводиться детальна структура цих таблиць.

Таблиця 3.1 – Структура таблиці reserv

Поле	Тип даних	Примітка
id	int	первинний ключ
isreserv	boolean	чи є дана запис бронюванням
idclient	int	код клієнта (зв'язок з таблицею Clients)
namereserv	varchar	ім'я клієнта, який бронює або реєструє
datereserv	datetime	дата створення запису
creator	smallint	код користувача, який створив запис
datestart	datetime	дата вселення
dateend	datetime	дата виселення
authcode	varchar	код
note	text	замітка
count	tinyint	кількість
cash	float	сплачена сума готівковим розрахунком
noncash	float	сплачена сума безготівковим розрахунком
paytax	float	сума сплати податку
parking	boolean	необхідність паркувального місця
deposit	float	завдаток клієнта
iscash	boolean	чи оплачував клієнт готівкою
active	boolean	активна чи запис, клієнт вже виїхав чи ні

delete	boolean	видалена чи запис
early	boolean	ранній виїзд
later	boolean	пізній заїзд
who	int	Ідентифікатор користувача, який вносив зміни
datetime	datetime	дата зміни

Таблиця 3.2 – Структура таблиці clients

Поле	Тип даних	Примітка
id_client	int	первинний ключ
fname	varchar	прізвище
sname	varchar	ім'я
address	varchar	адреса проживання
passport	varchar	номер паспорта
telephone	varchar	телефон
note	varchar	замітка
type_product	varchar	тип документа
email	varchar	поштова скринька
gender	enum	пол
national	varchar	національність

Таблиця 3.3 – Структура таблиці tariff

Поле	Тип даних	Примітка
id_tariff	smallint	первинний ключ
name	varchar	ім'я тарифу
tvalue	float	вартість
date	date	дата початку дії тарифу

Таблиця 3.4 – Структура таблиці room_type

Поле	Тип даних	Примітка
id_room_type	tinyint	первинний ключ
name	varchar	ім'я типу кімнати
spotcount	tinyint	кількість сплячих місць
id_tariff	smallint	зовнішній ключ з таблицею tariff

Таблиця 3.5 – Структура таблиці rooms

Поле	Тип даних	Примітка
id_room	smallint	первинний ключ
floor	tinyint	поверх
number	Smallint	номер кімнати
name	varchar	ім'я кімнати
id_room_type	smallint	зовнішній ключ з таблицею room_type

Таблиця 3.6 – Структура таблиці room_bad

Поле	Тип даних	Примітка
id_room_bad	smallint	первинний ключ
id_room	smallint	зовнішній ключ з таблицею rooms
id_bad_type	smallint	зовнішній ключ з таблицею bad_type
name	varchar	ім'я ліжка

Таблиця 3.7 – Структура таблиці bad_type

Поле	Тип даних	Примітка
id_bad_type	tinyint	первинний ключ
name	varchar	назва ліжка

Таблиця 3.8 – Структура таблиці reserve_room

Поле	Тип даних	Примітка
id_reserv_room	int	первинний ключ
id_room_bad	smallint	зовнішній ключ з таблиці room_bad
countpeople	tinyint	кількість проживаючих людей
counttax	tinyint	кількість оплачених податків
datetime	datetime	коли вносили зміни
active	boolean	активна чи запис
who	int	зовнішній ключ з таблицею clerck, запис того хто вносив зміни

Таблиця 3.9 – Структура таблиці clerck

Поле	Тип даних	Примітка
id_clerck	smallint	первинний ключ
id_level	tinyint	рівень доступу
login	varchar	логін
password	char	пароль

Таблиця 3.10 – Структура таблиці settings

Поле	Тип даних	Примітка
id_settings	tinyint	первинний ключ
name	varchar	ім'я параметра
value	varchar	значення

Таблиця 3.11 – Структура таблиці reserve_service

Поле	Тип даних	Примітка
id_reserv_service	int	первинний ключ
id_reserv	int	зовнішній ключ з таблицею service
id_service	tinyint	зовнішній ключ з таблицею послуги
date	datetime	дата внесення змін
active	boolean	активність цього запису
who	smallint	зовнішній ключ з таблицею clerck, запис того хто вносив зміни

Таблиця 3.12 – Структура таблиці service

Поле	Тип даних	Примітка
id_service	tinyint	первинний ключ
name	varchar	ім'я додаткової послуги
costs	float	вартість додаткової послуги
isdayly	boolean	за послугу ціна знімається щодня або один раз за весь час проживання

3.5 Тестування системи

Основна мета розробки програми – полегшити роботу адміністратору по підрахунку вартості проживання, створенню звіту та контролю за станом номерів у готелі.

Після запуску системи з'явиться сторінка в якій потрібно буде ввести логін і пароль користувача, і натиснути кнопку «Login» (рис. 3.9). Для завершення роботи програми необхідно натиснути на кнопку «File" -> "Exit».

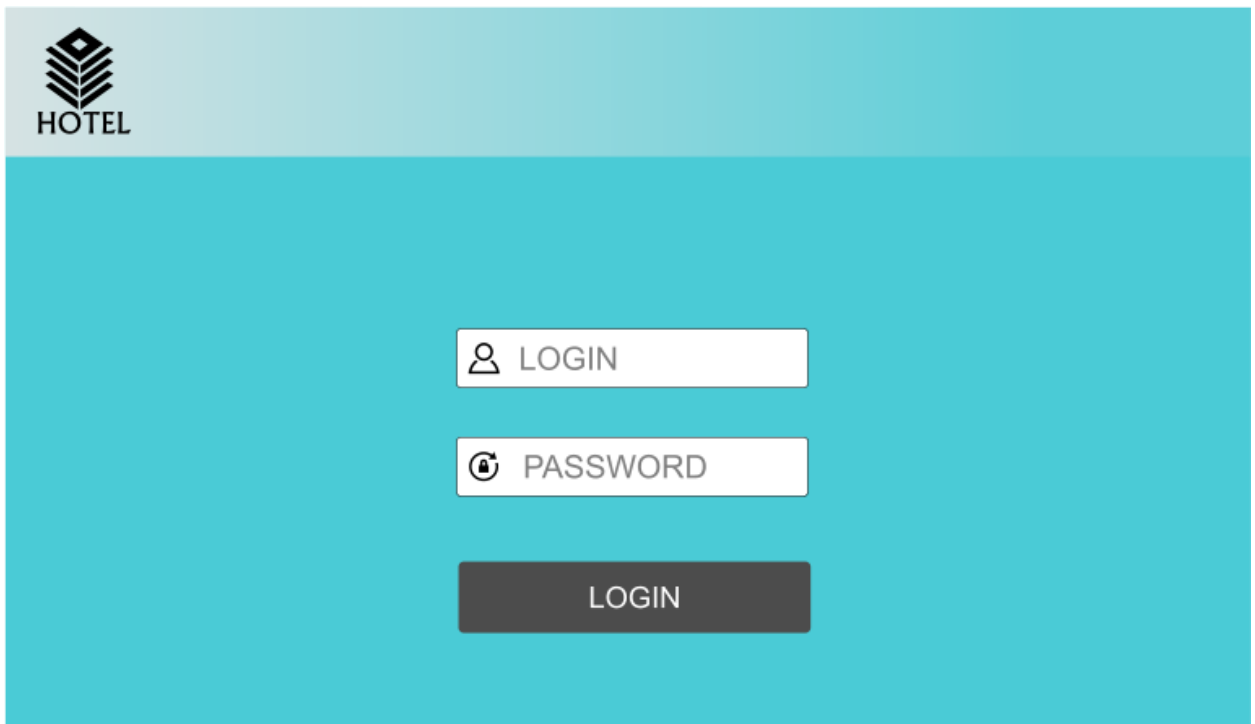


Рисунок 3.9 – Вхід в систему

Коли система запускається, з'являється сторінка, вигляд якої представлений на рис.3.10 і головне меню, що містять такі пункти: «Calendar», «Day info», «Statistic», «Registration», «Booking», для адміністраторів також доступні пункти меню: «Settings Data», «Settings».

При натисканні на пункт «Calendar» з'являється таблиця поселень мешканців з календарем. У таблиці в рядках записані номери кімнат, а в стовбцях – дати. У календарі можна вибрати дату, за яку адміністратор хоче подивитися резерв номерів. У календарі відображається 7 днів від поточної обраної дати. У день приїзду клієнта в комірці записується прізвище клієнта. Стовпчики таблиці розфарбовуються чотирма кольорами:

- білий – номер вільний;
- жовтий – клієнт забронював номер, але не заселився;
- зелений – клієнт заселився;
- сірий – клієнт проживав, розрахувався і вже покинув номер.

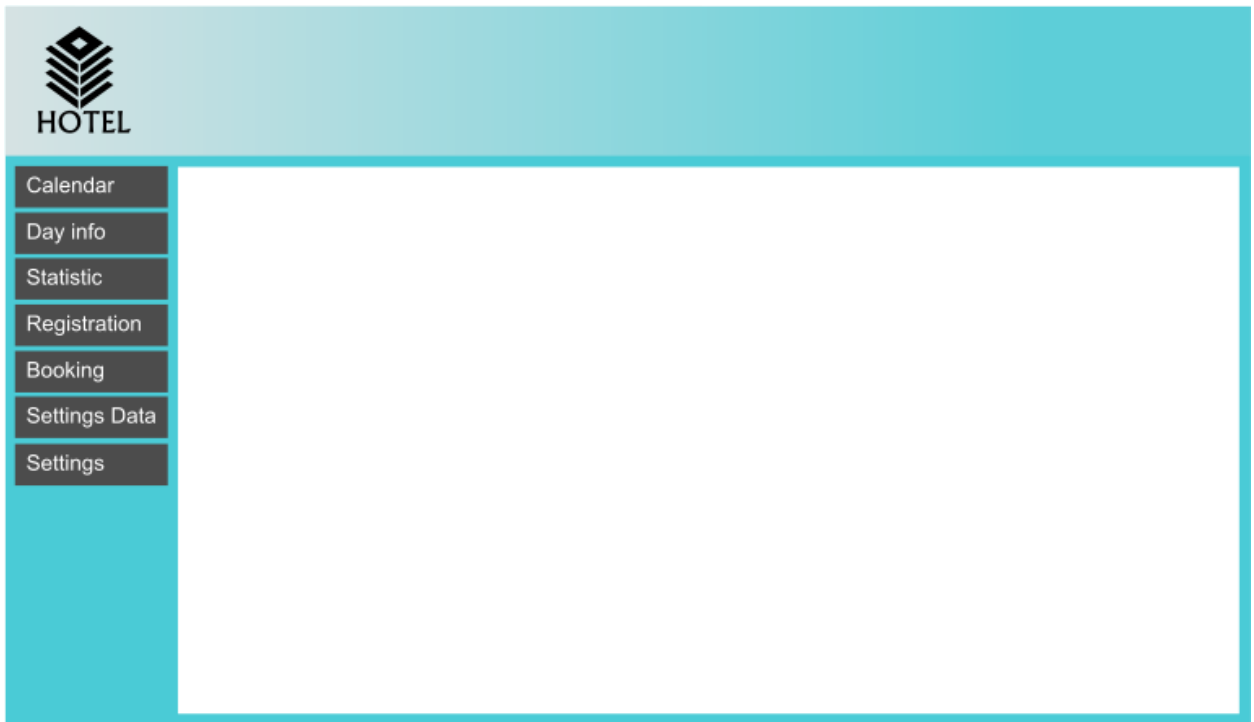


Рисунок 3.10 – Головне вікно

У будь-якому місці таблиці можна клацнути правою кнопкою мишки і з'явиться спливаюче вікно (рис.3.11), в якому можна відзначити номер на поселення; забронювати для клієнта (якщо номер вільний); змінити запис: викликавши форму редагування запису (рис.3.12) або одним натисканням мишки резерв перетворити на реєстрацію.

Сторінка «Реєстрація» (рис 3.12) розбита на кілька частин. На панелі «Client info» відображається інформація про клієнтів. Тут є можливість ввести особисті дані клієнта: ім'я, прізвище, стать, національність, номер документа, тип документа і поштову скриньку. Якщо поставити галочку біля email, то користувачеві прийде лист на пошту з інформацією про поселення.

У частині, яка інформує про номери («Room і Room info»), є можливість вибрати кімнати, що цікавлять адміністратора, і натиснути кнопку «Add», або вибравши з уже обраних видалити їх, натиснувши відповідно

кнопку «Delete». У вибраних кімнатах можна відредагувати кількість мешканців та кількість клієнтів, які платять податок на проживання.

Calendar	Name	Jan25	Jan26	Jan27	Jan28	Jan29	Jan30
Day info	Rock-n-Roll#1	Karen		Toth	Toth	Toth	Tot
Statistic	Rock-n-Roll#2	Karen		Toth	Toth	Toth	Tot
Registration	Rock-n-Roll#3	Karen					
Booking	Rock-n-Roll#4	Karen					
Settings Data	Rock-n-Roll#5	Karen					
Settings	Fusion#1	Karen		Surplice		Pilizari	
	Fusion#2	Karen		Surplice			
	Fusion#3	Karen		Surplice			
	Fusion#4	Karen					
	Blues#1	Karen					
	Blues#2	Karen					
	Blues#3	Karen					
	Blues#4	Karen					
	Blues#5	Karen					
	Blues#6	Karen					
	Opera	Karen					
	Jazz	Karen		Williams			
	Bossa nova	Karen					

Рисунок 3.11 – Календарь приїздів

На цієї ж сторінці в розділі «Credit card» адміністратор або порт'є вводить відомості про оплату клієнта за проживання. Платежі можуть бути різного виду: готівкові або безготівкові. Клієнт може заплатити завдаток. У разі безготівкового розрахунку в вікні вводиться ID транзакції.

На сторінці «Реєстрація» додана можливість вибору додаткових послуг, що надаються готелем клієнтові («Additional service»). До додаткових послуг відноситься: парковка, гараж, хімчистка та ін. Адміністратор може редагувати цей список при необхідності. Вартість послуг автоматично вклю-

чаються в рахунок гостя. Вид поля додавання додаткових послуг, яке автоматично з'являється при натисканні кнопки «Add», представлений на рис.3.13.

Таким чином, сторінка «Реєстрація» є основною сторінкою для введення вихідної інформації про клієнта. На підставі цих даних, пізніше портсь може згенерувати рахунок про оплату.

The screenshot shows a web interface for hotel registration. On the left is a navigation menu with options like Calendar, Day info, Statistic, Registration (highlighted), Booking, Settings Data, and Settings. The main area has tabs for 'New client' and 'Frequent client' (selected), and 'Registration' and 'Booking' (selected). The 'Client info' section contains fields for Name (Justin), Last name (Surplice), National, Telephone, E-mail, Ne document, Type document, Adress, and Gender (M). The 'Room' section has a list of room types (Rock-n-Roll#1-5, Fusion#1-3, Blues#1-2) and 'Add'/'Delete' buttons. Below this is a table:

Room	Cost	People	Count tax
Fusion#1	12.99	1	1
Fusion#2	12.99	1	1
Fusion#3	12.99	1	1

The 'Action' section has a 'Calculate' button and displays: Amount 70.15, Tax 9.9, Total 80.05, with 'Cancel', 'Ok', 'Clear', and 'Print' buttons. The 'Room info' section includes 'Date in' (20.07.2021), 'Date out' (22.07.2021), 'Id book' (111945), and checkboxes for 'Early arrival' and 'Late chek-out'. The 'Credit card' section has 'Authorization code' (51453-24103854), 'Sum' (0.0), 'Pay cash' (checked, 0.0), and 'Deposit' (7.79).

Рисунок 3.12 – Редагування даних реєстрації

На сторінці «Action» адміністратор може підрахувати вартість проживання в номерах, суму податку та загальну вартість з урахуванням податку та обраних клієнтом додаткових послуг. При натисканні на кнопку «Clear» інформація у вікні повністю очиститься, при натисканні на кнопку «OK» результат зберігається, а при натисканні на кнопку «Print» інформація збережеться і буде виведена на друк у вигляді звіту, що представляє собою рахунок-фактуру (рис. 3.14).

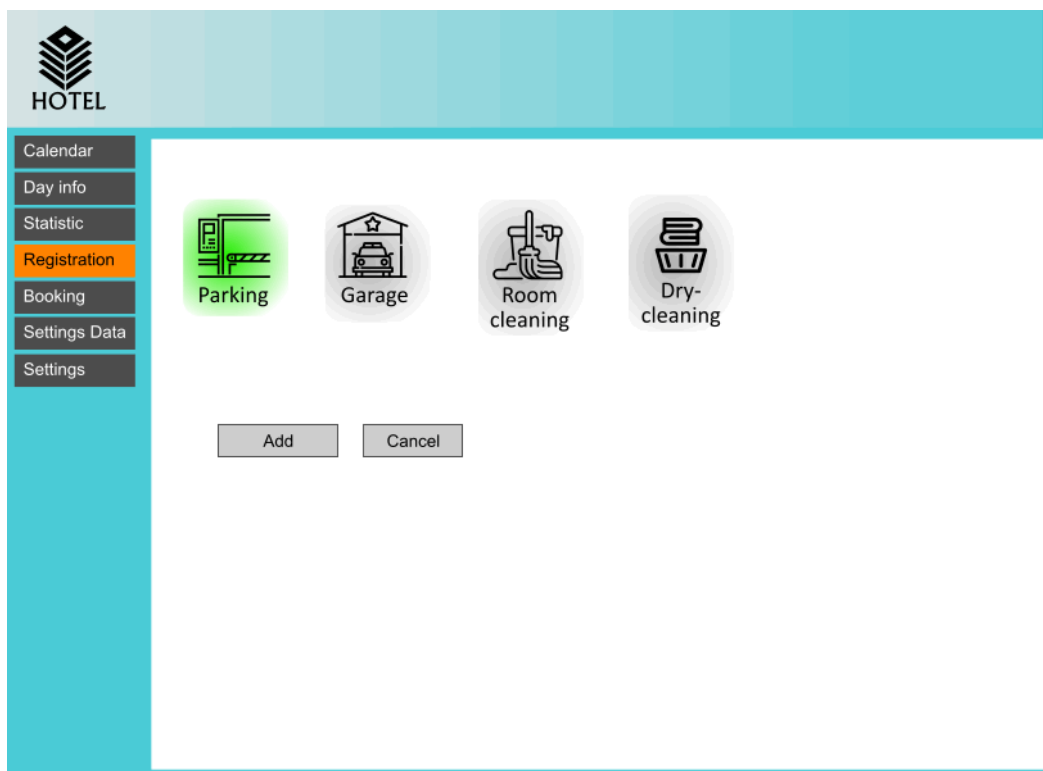


Рисунок 3.13 – Список додаткових послуг



Disco#1

Dates of stay:	Number of nights:	Type of room:	Number of rooms:	Price of room per night:
2021-06-21 2021-06-23	2 n		1	23.99 €

The tax rates:

(Accommodation tax € 3.85 per / person / night)

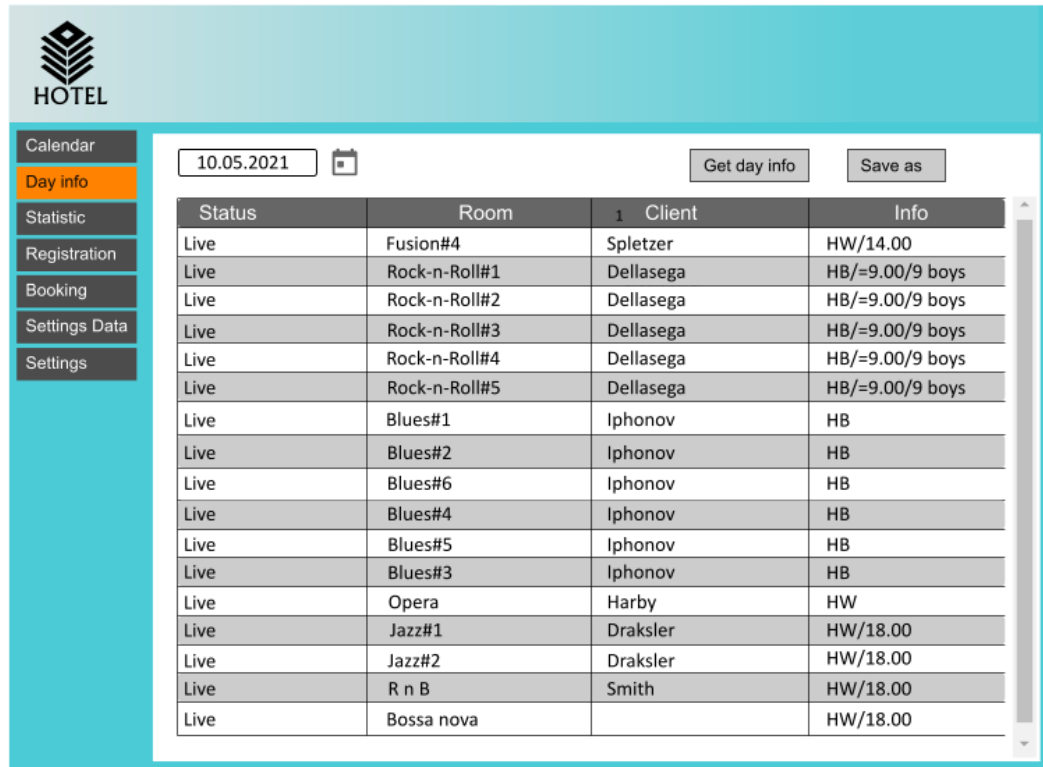
Number of residents: 1 people

Number of nights: 2 n

The total sum to be paid: 7.7 €

Рисунок 3.14 – Звіт «Рахунок фактура»

При виборі на панелі інструментів «Day info» з'являється сторінка (рис. 3.15) на якій можемо вибрати дату і отримати інформацію про здані номери за поточний день. При натисканні на кнопку «Save as» створюється xls-файл з поточною таблицею.

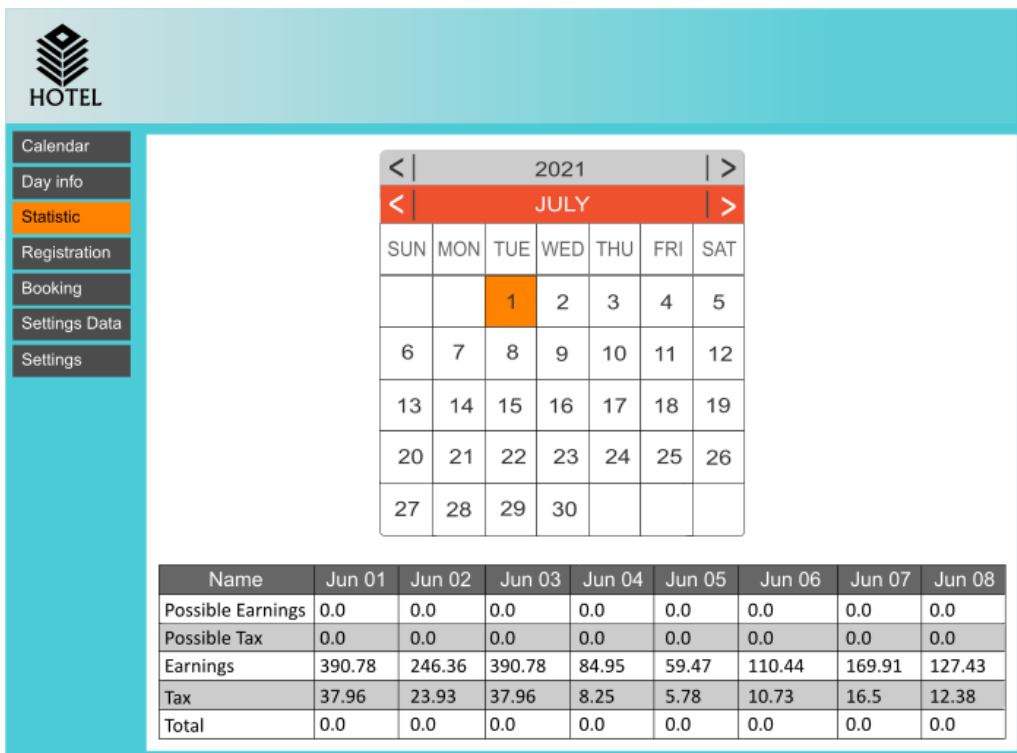


Status	Room	1 Client	Info
Live	Fusion#4	Spletzer	HW/14.00
Live	Rock-n-Roll#1	Dellasega	HB/=9.00/9 boys
Live	Rock-n-Roll#2	Dellasega	HB/=9.00/9 boys
Live	Rock-n-Roll#3	Dellasega	HB/=9.00/9 boys
Live	Rock-n-Roll#4	Dellasega	HB/=9.00/9 boys
Live	Rock-n-Roll#5	Dellasega	HB/=9.00/9 boys
Live	Blues#1	Iphonov	HB
Live	Blues#2	Iphonov	HB
Live	Blues#6	Iphonov	HB
Live	Blues#4	Iphonov	HB
Live	Blues#5	Iphonov	HB
Live	Blues#3	Iphonov	HB
Live	Opera	Harby	HW
Live	Jazz#1	Draksler	HW/18.00
Live	Jazz#2	Draksler	HW/18.00
Live	R n B	Smith	HW/18.00
Live	Bossa nova		HW/18.00

Рисунок 3.15 – Сторінка «Day info»

При виборі на панелі інструментів «Statistic» з'являється сторінка (рис. 3.16) в якій можна вибрати дату і отримати інформацію про поточний і можливий прибуток. Це створено для зняття щотижневої статистики про отримані кошти.

При натисканні на «Settings Data» створюється сторінка (рис. 3.17), в якій можна працювати з даними, що зберігаються в БД. Тут можна переглянути список всіх клієнтів (вкладка «Clients»), список типів ліжок (вкладка «Bed»), список кімнат (вкладка «Rooms»), тарифи (вкладка «Tariff»), типи кімнат (вкладка «Type Room»), список користувачів (вкладка «Clerck»). На кожній вкладці присутня кнопка додати, редагувати і видалити запис.



HOTEL

Calendar
Day info
Statistic
Registration
Booking
Settings Data
Settings

< | 2021 | >
< | JULY | >

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Name	Jun 01	Jun 02	Jun 03	Jun 04	Jun 05	Jun 06	Jun 07	Jun 08
Possible Earnings	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Possible Tax	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Earnings	390.78	246.36	390.78	84.95	59.47	110.44	169.91	127.43
Tax	37.96	23.93	37.96	8.25	5.78	10.73	16.5	12.38
Total	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Рисунок 3.16 – Сторінка «Statistic»



HOTEL

Calendar
Day info
Statistic
Registration
Booking
Settings Data
Settings

Clients Rooms Beds Sensors **Tariff** Equipment Room type Clerck

24.07.2021 

Name	Price	Type room	Date
1 day	73.99	4-bed Private Jazz	2021-07-24
1 day	55.99	2-bed (twin) private Ope...	2021-07-24
1 day	18.99	6-bed dorm Blues	2021-07-24
1 day	73.99	4-bed Private RnB	2021-07-24
1 day	16.99	5-bed dorm Rock-n-Roll	2021-07-24
1 day	65.99	3-bed Private Bossa nova	2021-07-24
1 day	16.99	4-bed dorm Fusion	2021-07-24

Delete Edit Add

Рисунок 3.15 – Таблиця з даними бронювання

На рис. 3.16 зображено сторінку з настройками, яка з'являється після натискання на кнопку «Settings». Тут вказується інформація про поштову адресу: поштова адреса, порт вихідної пошти, адреса, логін і пароль. Тут також вказується вартість паркування, і податку на одну особу. Також в настройку винесено чи необхідно вказувати код транзакції при безготівковому розрахунку.

HOTEL

Calendar
Day info
Statistic
Registration
Booking
Settings Data
Settings

Payment

Tax per day

Parking

Need credit card?

Reset bed number

e-mail

Send message?

Smtp server

Email from

User name

Password

Port

Ok

Рисунок 3.16 – Сторінка з налаштуваннями

ВИСНОВКИ

В ході виконання кваліфікаційної роботи бакалавра була система з web-інтерфейсом, яка дозволяє автоматизувати роботу готельного комплексу, пов'язана насамперед із бронюванням і розміщенням клієнтів.

При створенні програмного продукту була докладно вивчена предметна область. Аналіз документообігу та умов функціонування готельного комплексу проведений за допомогою CASE-засобів з використанням двох методів IDF0 і DFD. Побудовано модель потоків даних і діаграма дерева вузлів, яка показує ієрархію робіт і дозволяє розглянути всю модель цілком.

Побудована логічна і фізична моделі даних БД. Фізична модель бази даних, реалізована для MySQL, яка представляє собою функціонально повну реляційну СКБД. У ній передбачені всі необхідні засоби для визначення і обробки даних, а також для керування ними при роботі з великими обсягами інформації.

В якості засобів розробки були обрані сервер apache Tomcat, мова JavaScript (jQuery) і Spring (Java). В якості інструменту завдання стилів був обраний CSS фреймворк Bootstrap.

При створенні системи були враховані всі запропоновані вимоги. Програма має інтуїтивно зрозумілий інтерфейс, що дозволяє швидко заповнити, вибрати, переглянути дані і виконати всі стандартні операції з бронювання та заселення клієнтів. Адміністратор, який буде працювати з системою, зможе легко і без проблем працювати з базою даних: редагувати, видаляти непотрібні або оновлювати застарілі дані, здійснювати пошук даних за різними критеріями, виводити звіти на друк.

Система здійснює:

- ведення настроюваної системи номерів, а також системи бронювання та проживання з урахуванням виїзду, «незаїзду» і втратою готелю від невикористання номерного фонду;
- реєстрацію клієнтів та ведення списків;

- пошук даних за вільними номерами і за бронюванням;
- формування рахунків-фактури з урахуванням тарифів проживання, податків, види оплати та користування додатковими послугами;
- формування звітності.

Створений програмний продукт може мати важливе економічне значення, оскільки дозволить мінімізувати втрати прибутку готельного комплексу за рахунок економії витрат на обробку даних і підвищення продуктивність роботи персоналу.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

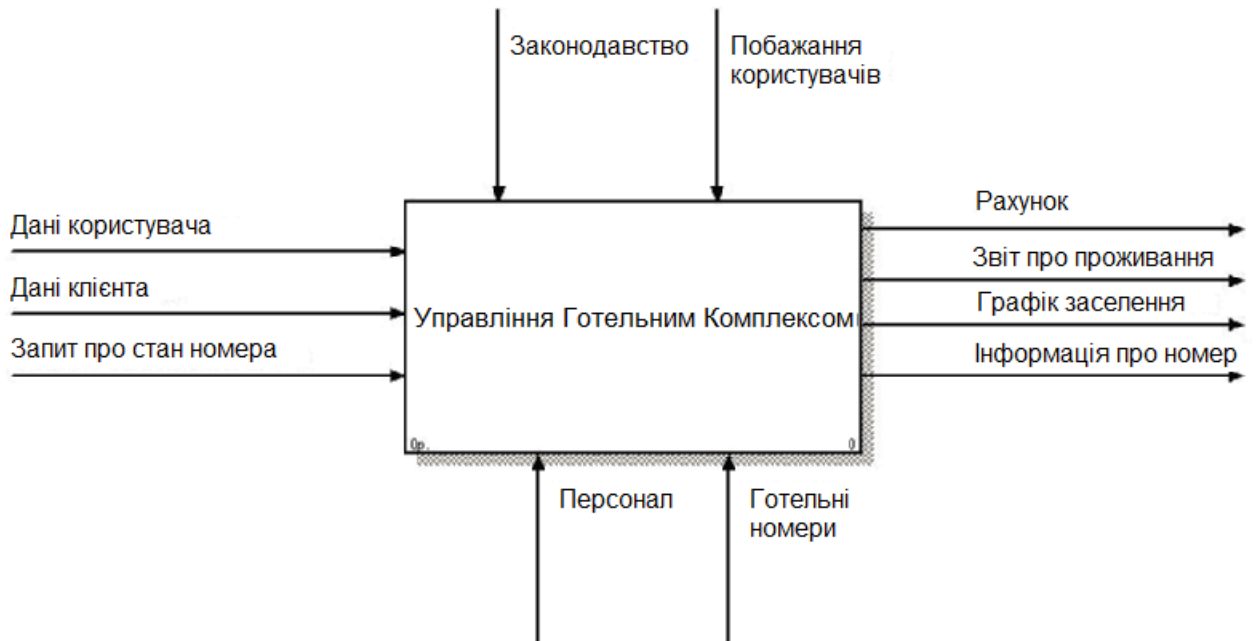
1. Система управління Bnovo PMS – Офіційний сайт. URL: <https://bnovo.ru/> (дата звернення 14.04.2021)
2. Система управління готелем Travelline – Офіційний сайт. URL: <https://www.travelline.ua/> (дата звернення 14.04.2021)
3. Система управління готелем jSolutions – Офіційний сайт. URL: <https://jsolutions.ua/> (дата звернення 14.04.2021)
4. Трофименко О.Г., Прокоп Ю.В., Логінова Н.І., Копитчук І.М. Організація баз даних: навч. посібник. 2-ге вид. виправ. і доповн. Одеса : Фенікс, 2019.– 246 с.
5. TOPDB Top Database index. URL: <https://pypl.github.io/DB.html> (дата звернення 14.04.2021)
6. MySQL – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/MySQL> (дата звернення 14.04.2021)
7. PostgreSQL – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/PostgreSQL> (дата звернення 14.04.2021)
8. SQLite – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/SQLite> (дата звернення 14.04.2021)
9. Рейтинг языков программирования для web-разработки (backend). URL: <https://techrocks.ru/2019/01/04/best-backend-web-devprogramming-languages/> (дата звернення 14.04.2021)
10. JavaScript – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/JavaScript> (дата звернення 14.04.2021)
11. Дэвид Флэнаган. JavaScript. Подробное руководство. 5-е изд. Изд-во: Символ-Плюс, 2008 р. 453 с.
12. PHP – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/PHP> (дата звернення 14.04.2021)
13. Кузнецов М.В, Объектно-ориентированное программирование на PHP. СПб.: «БХВ-Петербург», 2007. 608 с.

14. Laravel – Вікіпедія. (загол. з екрана). URL:<https://uk.wikipedia.org/wiki/Laravel> (дата звернення 14.04.2021).
15. Java – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення 14.04.2021)
16. Oracle Corporation – Вікіпедія. (загол. з екрана). URL: https://uk.wikipedia.org/wiki/Oracle_Corporation (дата звернення 14.04.2021)
17. Блинов, И.Н., Романчик, В. С. Java. Методы программирования : уч.-мет. Пособие. Минск : издательство «Четыре четверти», 2013. 896 с.
18. Python – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Python> (дата звернення 14.04.2021)
19. Django – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Django> (дата звернення 14.04.2021)
20. Bootstrap – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Bootstrap> (дата звернення 04.01.2021)
21. Документація по web серверу apache. URL: <http://httpd.apache.org/docs/> (дата звернення 04.01.2021)

Д О Д А Т К И

ДОДАТОК А

Контекстна діаграма управління готельним комплексом



ДОДАТОК Б

Моделі потоків даних DFD

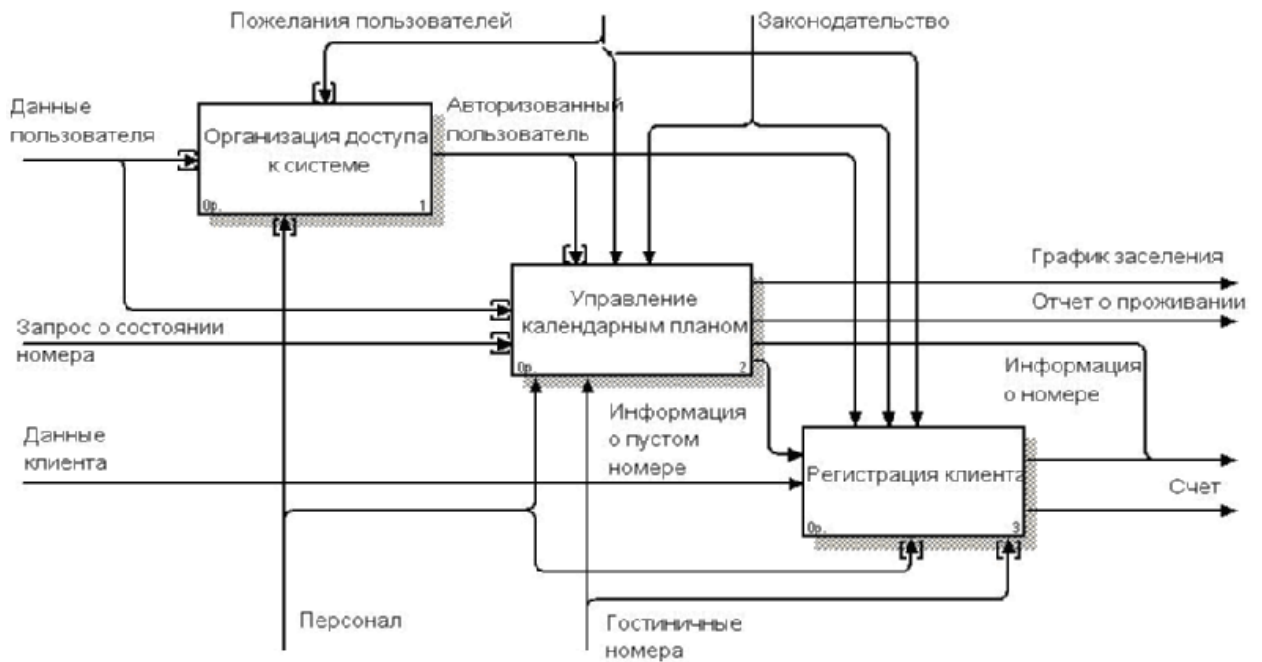


Рисунок Б.1 – Диаграмма декомпозиции управления готельным комплексом



Рисунок Б.2 – Диаграмма декомпозиции организации доступа до системы

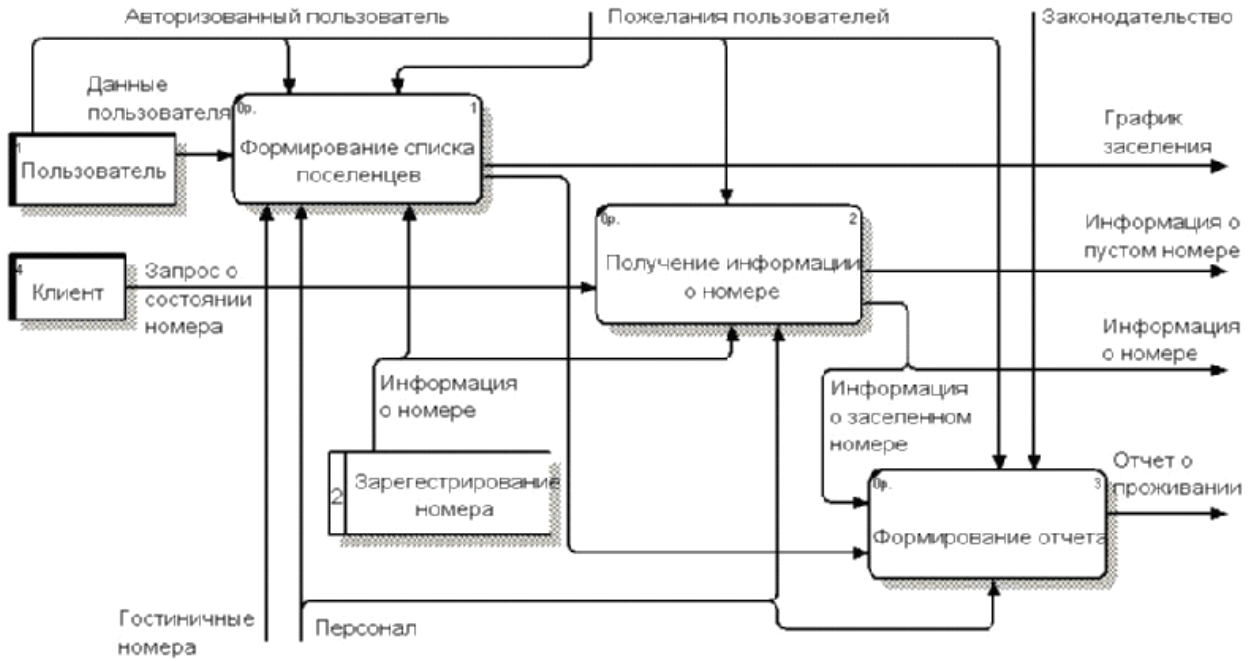


Рисунок Б.3 – Диаграмма декомпозиції управління календарним планом

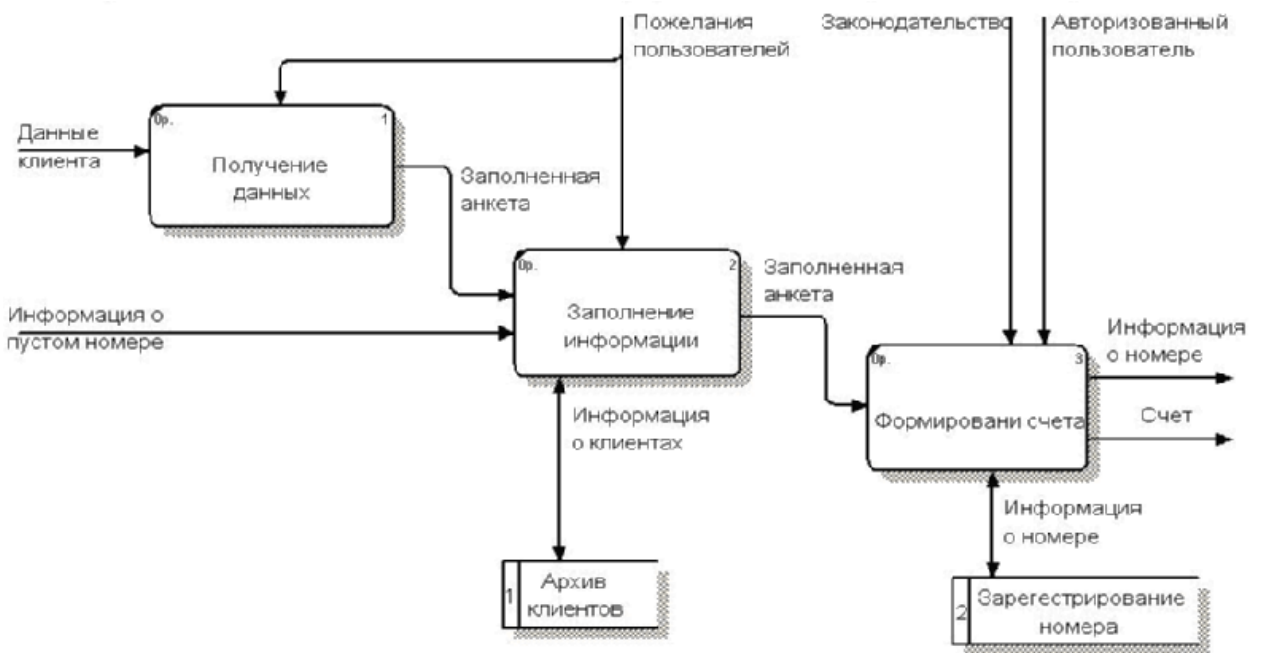


Рисунок Б.4 – Диаграмма декомпозиції реєстрації клієнта

ДОДАТОК В

Діаграма дерева вузлів



ДОДАТОК Г

Фрагмент програмного коду