

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

Кафедра системного аналізу

Пояснювальна записка

до магістерської роботи

на ступінь вищої освіти магістр

на тему: **«ІНФОРМАЦІЙНІ НАПРЯМКИ ВДОСКОНАЛЕННЯ
КОМУНІКАЦІЙНОЇ ФУНКЦІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ, ЩО
ВИКОРИСТОВУЮТЬСЯ В СІМЕЙНИХ ГОТЕЛЬНИХ МЕРЕЖАХ»**

Виконав: студент 6 курсу, групи САДМ-61
спеціальності

124 Системний аналіз

(шифр і назва спеціальності)

Голосов В.С.

(прізвище та ініціали)

Керівник Козаченко С.Я.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ

Кафедра Системного аналізу

Ступінь вищої освіти Магістр

Спеціальність 124, Системний аналіз

ЗАТВЕРДЖУЮ

Завідувач кафедри
системного аналізу

_____ Гордієнко Т.Б.
“ _____ ” _____ 20__ року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Голосову Владиславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Інформаційні напрямки вдосконалення комунікаційної функції в інформаційних системах, що використовуються в сімейних готельних мережах

Керівник роботи Козаченко Сергій Якович, к.е.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ _____ ” 20__ року № _____

2. Строк подання студентом роботи _____

3. Вихідні дані до роботи

3.1 Науково-технічна література та електронні джерела

3.2 Характеристика існуючих інформаційних систем, що використовуються в сімейних готельних мережах

3.3 Аналіз ринку готельного бізнесу

3.4 Аналіз інформаційних систем, що використовуються в управлінні в сімейних готельних мережах

4. Концептуальна структура системи функції комунікації

4.1. Проектування структури інформаційної системи

4.2. Розробка структури програмного забезпечення інформаційної системи

4.3. Загальна структура програмного застосування системи та посібник користувача

4.4. Системні вимоги програмних та апаратних засобів, інсталяція системи

5. Перелік графічного матеріалу

5.1 Лістинг програми

5.2 Презентація магістерської роботи

5.3 Додатки

6. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури		Виконав
2	Пошук та аналіз інформації про автоматизовані системи сімейних готельних комплексів		Виконав
3	Пошук та аналіз можливостей вдосконалення комунікаційної функції в інформаційних системах, що використовуються в сімейних готельних мережах		Виконав
4	Аналіз інформаційних систем, що використовуються в управлінні в сімейних готельних мережах		Виконав
5	Проектування структури інформаційної системи		Виконав
6	Розробка структури програмного забезпечення інформаційних системи сімейних готельних мережах		Виконав
7	Розробка обов'язкових демонстраційних матеріалів		Виконав
8	Попередній захист роботи		Виконав
9	Пред'явлення роботи в деканат		Виконав

Студент

(підпис)

Керівник роботи

(підпис)

Голосов В.С.

(прізвище та ініціали)

Козаченко С.Я.

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 71 с., 14 табл., 44 рис., 3 дод., 20 джерел.

АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ, СІМЕЙНІ ГОТЕЛЬНІ МЕРЕЖІ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, SQL-ІНТЕРФЕЙСИ, VISUAL STUDIO, ВИЗНАЧЕННЯ СКЛАДУ БАЗИ ДАНИХ, MICROSOFT SQL SERVER 2008 EXPRESS, ПРОЕКТІВАННЯ БАЗИ ДАНИХ, СТРУКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ, ПОБУДОВА ІНФОРМАЦІЙНОЇ СИСТЕМИ, РОЗРОБКА БАЗИ ДАНИХ, РОЗРОБКА ІНТЕРФЕЙСУ, ІНСТАЛЯЦІЯ ТА ЗАСТОСУВАННЯ СИСТЕМИ.

Об'єкт дослідження – структура ділових процесів готельного закладу з погляду оперативного обліку.

Предмет дослідження – створення прикладу інформаційної системи для автоматизації системи документообігу підприємства сфери обслуговування населення.

Мета роботи – Вдосконалення комунікаційної функції в інформаційних системах, що використовуються в сімейних готельних мережах готельного бізнесу.

Методи дослідження — аналіз існуючих рішень для автоматизації системи документообігу підприємства сфери обслуговування населення та якісного супроводу документообігу та комунікацій в сімейних готельних мережах, що супроводжується зростанням вимог до якості та видів обробки інформації.

Отримані результати – Виконано створення ряду програмних додатків засобами середовища програмування Visual Studio, що демонструють можливості ЕОМ щодо виконання завдань роботи з базами даних з використанням мови програмування C#. Доступ до бази даних у програмній оболонці передбачається здійснено за допомогою використання SQL-інтерфейсу у зв'язку з технологіями програмування .NET. Вибір даної технології обумовлений опрацьованістю та стабільністю зв'язку БД<->додаток, які підтримуються сучасними Windows-системами. Носієм для серверної бази даних вибрано стандартну базу даних у

форматі Microsoft SQL Server 2008 Express (*.mdf) даних. Всі алгоритмічні дії за програмою базуються на основному циклі роботи програми в середовищі Windows, переривання якого виконує поставлені завдання.

Зміст

ВСТУП.....	9
1 ХАРАКТЕРИСТИКА ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ, ЩО ВИКОРИСТОВУЮТЬСЯ В СІМЕЙНИХ ГОТЕЛЬНИХ МЕРЕЖАХ	11
1.1 МЕТА СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	11
1.2 АНАЛІЗ РИНКУ ГОТЕЛЬНОГО БІЗНЕСУ	12
1.2.1 Аналіз діяльності організації.....	17
1.2.2 Рівень розвиненості функції комунікацій в інформаційних системах як одна з конкурентних переваг.....	27
1.3 АНАЛІЗ ІНФОРМАЦІЙНИХ СИСТЕМ, ЩО ВИКОРИСТОВУЮТЬСЯ В УПРАВЛІННІ В СІМЕЙНИХ ГОТЕЛЬНИХ МЕРЕЖАХ	30
2 КОНЦЕПТУАЛЬНА СТРУКТУРА СИСТЕМИ ФУНКЦІЇ КОМУНІКАЦІЇ.....	33
2.1 ВИЗНАЧЕННЯ СКЛАДУ БАЗИ ДАНИХ.....	33
2.2 ФІЗИЧНЕ ПРОЕКТУВАННЯ БАЗИ ДАНИХ.....	40
2.3 ПРОЕКТУВАННЯ СТРУКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	58
3 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	61
3.1 ЗАСОБИ РЕАЛІЗАЦІЇ	61
3.2 ЗАГАЛЬНА СТРУКТУРА ПРОГРАМНОГО ЗАСТОСУВАННЯ СИСТЕМИ.....	62
3.3 ПОСІБНИК КОРИСТУВАЧА	64
3.3.1 Системні вимоги програмних та апаратних засобів	68
3.3.2 Інсталяція системи	69
3.3.3 Результати тестування з прикладами екранних форм	70
ВИСНОВКИ	77

ПЕРЕЛІК ПОСИЛАНЬ.....	79
ДОДАТОК 1.....	82
ДОДАТОК 2.....	123
ДОДАТОК 3.....	133

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

APM	-	Автоматизоване робоче місце
АСУ	-	Автоматизована система управління
IC	-	Інформаційна система
SQL	-	Structured Query Language
PMS	-	Property Management System
СУБД	-	Система управління базами даних
БД	-	База даних
IDEF	-	Integrated DEFinition
ER-модель	-	Entity-relationship model
ЕОМ	-	Електронно-обчислювальна машина
HTML	-	HyperText Markup Language
MDI	-	multiple document interface
ВДТ	-	Відео дисплейні термінали
ПЕОМ	-	Персональні електронно-обчислювальні машини
КПО	-	Коефіцієнт природного освітлення
БНіП	-	Будівельні норми і правила
ДСанПіН	-	Державні санітарні правила і норми
АЦДП	-	Алфавітно-цифровий друкувальний пристрій

ВСТУП

Інформаційно-довідкові системи призначені для управління різними інформаційними та довідковими масивами, до яких входять і бази даних. Ці системи організовують певний інтерфейс користувача з функціями обробки інформації самої системи. Сучасні великі проекти ІС характеризуються, як правило, такими особливостями:

- складність опису (досить велика кількість функцій, процесів, елементів даних та складні взаємозв'язки між ними), що вимагає ретельного моделювання та аналізу даних та процесів;
- наявність сукупності тісно взаємодіючих компонентів (підсистем), що мають свої локальні завдання та цілі функціонування (наприклад, традиційних додатків, пов'язаних з обробкою транзакцій та рішенням регламентних завдань, та додатків аналітичної обробки (підтримки прийняття рішень), які використовують нерегламентовані запити до даних великого обсягу);
- відсутність прямих аналогів, що обмежує можливість використання будь-яких типових проектних рішень та прикладних систем;
- необхідність інтеграції існуючих та новостворених додатків;
- функціонування у неоднорідному середовищі на кількох апаратних платформах;
- роз'єднаність та різнорідність окремих груп розробників за рівнем кваліфікації та сформованим традиціям використання тих чи інших інструментальних засобів;
- суттєва тимчасова протяжність проекту, зумовлена, з одного боку, обмеженими можливостями колективу розробників, і, з іншого боку, масштабами організації-замовника та різним ступенем готовності окремих її підрозділів до впровадження ІС.

Метою написання цієї випускної кваліфікаційної роботи є вдосконалення комунікаційної функції в інформаційних системах, що використовуються в сімейних готельних мережах готельного бізнесу.

Практично в усіх установах сфери обслуговування є необхідність оперативного обміну інформацією між персоналом і клієнтами, зокрема на дистанційному рівні. Така інформація зазвичай враховується в журналах обліку, а також передається за допомогою традиційних документів.

Предметом дослідження є створення прикладу інформаційної системи для автоматизації системи документообігу підприємства сфери обслуговування населення.

Об'єктом дослідження є структура ділових процесів готельного закладу з погляду оперативного обліку. Актуальність побудови цієї системи обумовлена стабільним зростанням споживчого попиту до питань якісного супроводу документообігу в готелях, що супроводжується зростанням вимог до якості та видів обробки інформації.

Гіпотеза дослідження визначена тим, що в процесі створення та функціонування ІС інформаційні потреби користувачів можуть змінюватись або уточнюватись, що ще більше ускладнює розробку та супровід таких систем.

Для досягнення мети роботи було поставлено такі завдання:

- Розробити та налаштувати модель бази даних установи.
- Розробити настільний Windows-додаток, що забезпечує створення та модифікацію даних адміністратором бази даних.
- Розробити екранні інтерфейсні елементи, що забезпечують створення та модифікацію даних на рівні робочих місць відповідальних співробітників мережі готельних установ.
- Розробити екранні інтерфейсні елементи, які забезпечують пошук, перегляд та використання даних на рівні робочих місць відповідальних працівників мережі готельних закладів.

1 ХАРАКТЕРИСТИКА ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ, ЩО ВИКОРИСТОВУЮТЬСЯ В СІМЕЙНИХ ГОТЕЛЬНИХ МЕРЕЖАХ

Основним елементом у сучасних інформаційних системах є автоматизоване робоче місце (АРМ) – проблемно-орієнтований програмно-технічний комплекс, винесений на робоче місце та який автоматизує в режимі діалогу деякий набір управлінських процедур кінцевого користувача за його безпосередньої участі.

Об'єктом практичної розробки цього проекту є модель обліку виробничого процесу мережі установ готельного призначення.

АСУ для мережі готелів (готельних комплексів) є комплексом інтегрованих підсистем, що створює ефективне середовище взаємодії співробітників, клієнтів і ділових партнерів - туристичних агентств, корпоративних клієнтів і туроператорів.

1.1 Мета створення інформаційної системи

Функції проектованої інформаційної служби у мережі готелів визначаються з цілей та завдань її діяльності та включають повний обсяг роботи, що виконується службою.

Предметною областю завдання є дані з обліку виробничої діяльності типового готельного комплексу (мережі готельних установ).

Інформаційна система у разі виконує функції з управлінської діяльності установи та забезпечує її ефективний виробничий процес. Передбачається, що в організації ведеться постійний збір даних про готельний процес, що вводяться до бази даних. Так здійснюється накопичення необхідної на вирішення завдання інформації.

Для виконання завдання необхідно спроектувати та створити базу даних у форматі, який підтримує спільну одночасну роботу групи користувачів. Інтерфейсом для доступу до такої бази даних є настільна Windows-програма, яка, обробляючи дані у своїй серверній частині, виконує клієнтські запити щодо отримання інформації.

Середовище програмування Visual Studio є інструментом, який дозволяє виконати всі завдання в циклі: проектування, розробка схеми і створення бази даних, заповнення початковими даними, забезпечення її роботи в серверному SQL режимі, а також розробку та тестування програмної оболонки для прикладного практичного доступу до неї.

1.2 Аналіз ринку готельного бізнесу

Сучасна готельна мережа є комплексом функціональних ланок, від злагодженості роботи якого залежить успішність існування підприємства над ринком. Вирішення цієї проблеми можливе лише за рахунок впровадження систем автоматизації роботи готелю, тобто впровадження Автоматизованих Систем Управління (АСУ) готелем, або, в англійському варіанті – Property Management System (PMS).

За кордоном перші АСУ готелів з'явилися у другій половині минулого століття. Багаторічний досвід розвитку та експлуатації дозволив їм стати досить досконалими програмними продуктами. І хоча ціна таких систем висока, згідно з дослідженнями корпорації Microsoft, більшість готелів у країнах періодично встановлюють нову систему управління. Це викликано наростаючими темпами зростання конкурентної боротьби та технологічного прогресу – якщо раніше готелі

змінювали технічне обладнання в середньому кожні 7–9 років, сьогодні - кожні 3-5 років, і тенденція до зменшення цього періоду залишається.

У єдине інформаційне поле мережі готелів інтегровані автоматизована система управління (АСУ) готельним бізнесом, комплекси систем безпеки, життєзабезпечення, інформатизації з використанням технології клієнт-сервер.

Цей комплекс включає:

- модулі систем управління службами управління та розміщення (АРМ -портъє);
- управління якістю обслуговування; оптимізація прибутку;
- централізоване бронювання та централізована інформація про клієнтів;
- управління продажами та подіями;
- автоматизація робочих місць (кількість адміністраторів номерів, чергування на підлозі, видобуток, інформація та довідкові та інші послуги);
- інтернет -рішення.

АСУ готельного підприємства узгоджено працює з різними "спорідненими" програмними та апаратними комплексами - бухгалтерською системою та складською системою, АСУ ресторана та міні-бара, системою віддаленого бронювання, управлінням готельною групою та системою контролю доступу. Усі вони поєднуються в єдину локальну комп'ютерну мережу.

Готель є невід'ємною частиною індустрії гостинності, яка, в свою чергу, належить до більшої галузі - туризму. При ознайомленні із предметною областю розглядаються існуючі виробничі та технологічні процеси, потоки та організаційна структура установи.

Мережа готелів може бути визначена як комерційне підприємство, головне завдання якого - надати туристам послуги з розміщення та харчування, а також надання в оренду конференц-залів. Структура мережі готелів складається із структури кожного готелю.

Керівні працівники:

- директор готелю;
- адміністратор готелю.

Департамент прийому та розміщення: основними функціями відділу прийому та розміщення є реєстрація гостей, розподіл номерів, поселення та виписки клієнтів та надання ним численних додаткових послуг. Процес обслуговування гостей можна розбити на кілька етапів:

- бронювання – попереднє замовлення місць у готелі;
- прийом, реєстрація та розміщення гостей;
- надання послуг проживання;
- надання додаткових послуг;
- остаточний розрахунок та оформлення виїзду.

Департамент обслуговування: з погляду гостей служба обслуговування є найважливішою у готелях, оскільки персонал саме цієї служби працює з клієнтами у постійному контакті та виконує всі функції, пов'язані з їх безпосереднім обслуговуванням.

Очолює департамент обслуговування менеджер, якому підпорядковані швейцари, коридорні, підносики багажу, ліфтери, консьєржі, посильні, водії (обслуговують орендовані машини та паркують автомобілі гостей).

Як правило, першими зустрічають гостей швейцари, які стоять біля входу до готелю. Вони мають привітати гостей, допомогти їм вийти із машини. Швейцари повинні володіти інформацією про послуги, наявні в готелі, про готельні заходи (конференції, банкети), про місце розташування готелю та його околиці.

Господарська служба: керівник цього підрозділу несе відповідальність за чистоту всього готелю та його номерів. Зазвичай вона має кілька помічників, старших покоївок (старша покоївка відповідає за прибирання номерів на одному-

двох поверхах). Господарська служба щоденно звіряє свої дані про номери з даними служби прийому. Повідомлення служби прийому про витяг гостя з номера фіксуються у відповідній графі журналу господарської служби. В іншій графі журналу записується прізвище покоївки, якій доручено прибирання номера. Після закінчення збирання у журналі робиться відповідна позначка. Про готовність номера повідомляється у службу прийому.

Інженерна служба: сучасний готель насичений складним інженерним обладнанням (швидкісні ліфти, системи кондиціювання, опалення, водопостачання та каналізації, електричне (великої потужності) та газове обладнання кухні, кабельне телебачення, комп'ютери тощо). Для того щоб не містити повний штат інженерно-технічних працівників для обслуговування та ремонту всього обладнання, готель укладає договори зі спеціалізованими фірмами, які здійснюють його обслуговування та ремонт. Інженерна служба має відносно невеликий штат співробітників, бажано універсалів, здатних усунути нескладні несправності сантехнічного та електроустаткування та здійснювати грамотну експлуатацію всього обладнання.

Департамент безпеки: особиста безпека гостей та гарантована безпека їхньої власності – важливий аспект у роботі будь-якого готелю. Службовці зобов'язані вживати всіх необхідних запобіжних заходів, щоб захистити гостей від пограбування, шахрайства, насильства тощо. Вони несуть відповідальність за збитки, завдані гостям за час їхнього проживання в готелі. До технічних засобів безпеки насамперед відносяться телевізійні камери, за допомогою яких проглядаються всі громадські та багато службових приміщень. Записи повинні зберігатись не менше 24 годин. Введення електронних замків суттєво підвищило безпеку номерного фонду. Також для запобігання крадіжкам у багатьох готелях на прохання клієнтів у номерах встановлюються індивідуальні сейфи. Користування цими сейфами здійснюється за додаткову плату у період проживання. Крім

індивідуальних сейфів багато готелів організують і централізоване зберігання майна клієнтів у камері зберігання та сейфової комори.

Департамент безпеки готелю займається такими проблемами:

- розроблення процедур реагування на надзвичайні події;
- повсякденною безпекою гостьових кімнат;
- контролем ключів;
- запобігання крадіжкам, контролю за замками;
- контролем доступу до будівлі готелю;
- системою охоронної сигналізації;
- контролем території;
- зовнішнім освітленням;
- системою спостереження з допомогою телемоніторів;
- сейфами та вогнетривкими шафами;
- збиранням та зберіганням інформації.

Департамент закупівель (постачання): функція підрозділу - закупівлі необхідних товарів необхідної якості за розумною ціною у потрібний час. При цьому повинен витримуватися оптимальний розмір запасів у готелі, оскільки зберігання коштує грошей, до того ж багато закуплених товарів при тривалому зберіганні можуть зіпсуватися.

Департамент харчування: служба громадського харчування забезпечує обслуговування гостей у ресторанах, кафе чи барах готелю, вирішує питання щодо організації та обслуговування банкетів, презентацій та інших святкових заходів.

Департамент інформаційних технологій: комп'ютерні системи централізованого управління готельним комплексом дозволяють координувати роботу навіть великого готелю без особливих витрат часу та сил. Система дозволяє вирішити такі завдання:

- оптимізація бізнес-процесів; зниження паперового документообігу;
- підвищення контролю за діяльністю служб та персоналу;
- підвищення якості обслуговування гостей;
- оптимізація операційних витрат;
- статистичні та аналітичні функції.

Всі системи автоматизації готельних господарств мережі є інтегрованими пакетами програм, що автоматизують діяльність основних служб готелю: управління номерним фондом, адміністративної, комерційної, інженерної, служби громадського харчування.

1.2.1 Аналіз діяльності організації

За підсумками попереднього аналізу предметної області можна назвати такі інформаційні об'єкти:



Рисунок 1.1 Основні інформаційні об'єкти

Система створюється для обслуговування наступних груп користувачів:

- співробітники та адміністрація мережі готелів;

- постояльці, клієнти та відвідувачі.

При ознайомленні із предметною областю загалом розглядаються існуючі виробничі процеси, потоки, організаційна структура установи.

Серед зазначених інформаційних об'єктів можна припустити існування таких функціональних зв'язків:

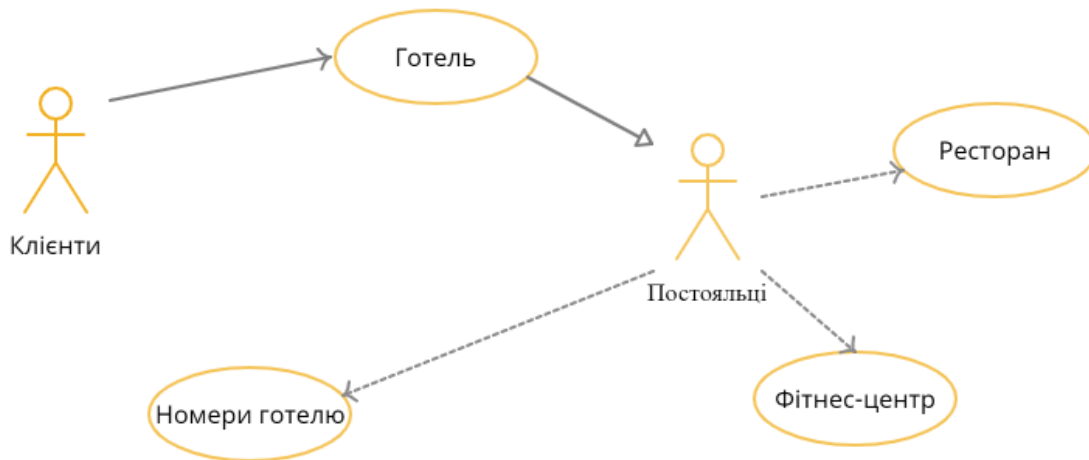


Рисунок 1.2 Можливі функціональні зв'язки ІС

Мається на увазі, що вся інформація зберігатиметься у базі даних, з якою працюватимуть співробітники мережі готелів, такі як керівництво, диспетчери, менеджери, персонал, адміністратори системи. Аналізуючи організаційну структуру умовного готелю, можна виділити такі процеси:

- реєстрація та облік готелів, що входять до мережі готельного комплексу;
- реєстрація та облік замовлень на бронювання номерів;
- реєстрація та облік типів, категорій та додаткових опцій готельних номерів;
- реєстрація та облік готельних номерів;
- реєстрація та облік типів обслуговуючого персоналу;
- реєстрація та облік співробітників готелю;

- реєстрація та облік постояльців готелів;
- реєстрація та облік процесу прибирання та ремонту готельних номерів;
- реєстрація та облік типів занять фітнесом;
- реєстрація та облік занять фітнесом постояльцями;
- реєстрація та облік типів комплексного харчування для постояльців;
- реєстрація та облік харчування постояльців.

На початку здійснення кожного життєвого циклу бізнес-процесу готельної інформаційної системи виконується бронювання готельного номера:

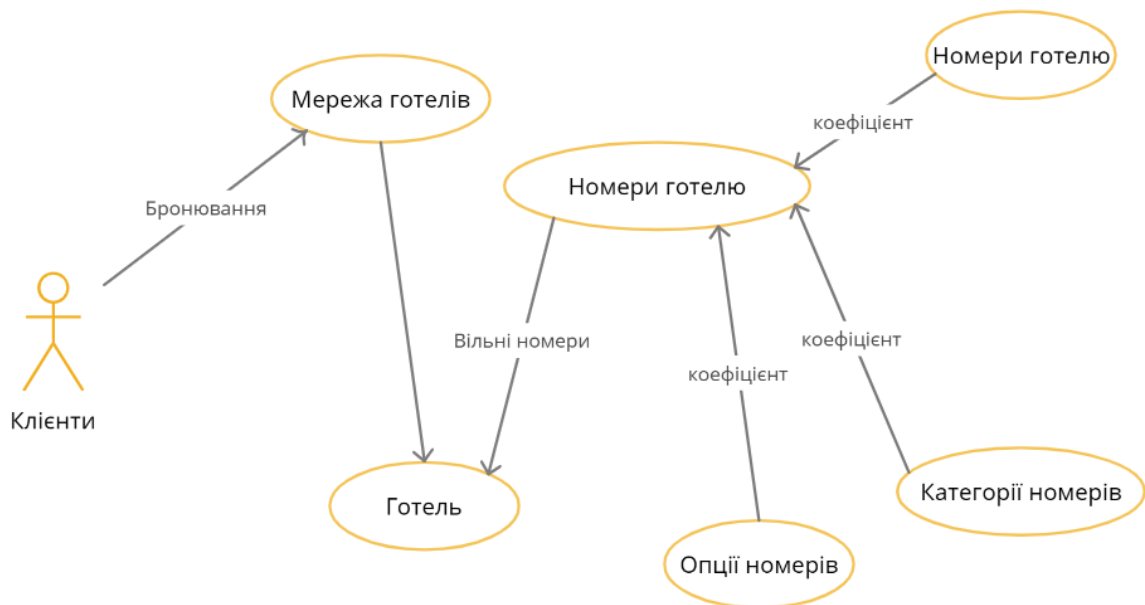


Рисунок 1.3 Опис процесу бронювання номерів

В результаті виконання процесу бронювання адміністратор з розміщення (Ресепшн) отримує інформацію, після підтвердження якої він може здійснити розміщення клієнта в номері на вказаний час та термін.

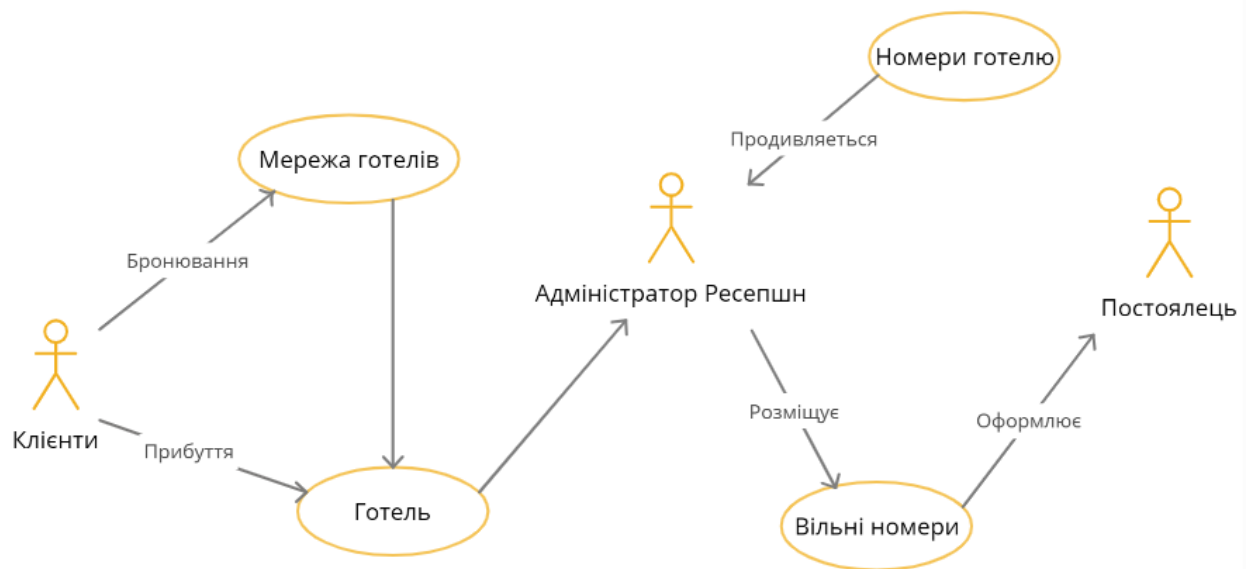


Рисунок 1.4 Опис процесу розміщення постояльця

Аналогічним чином Клієнт може безпосередньо (телефоном чи особисто) звернутися до Адміністратора Ресепшн, внаслідок чого той, на підставі наявної в нього інформації про вільні номери, розміщує клієнта. Отже, виходячи з виконання такої дії клієнт стає Постояльцем. Тут слід зазначити, що під поняттям Постоялець мається на увазі кількість людей, які розміщуються разом з ним: наприклад, сім'я з трьох осіб, де відповідальним Постояльцем є глава сім'ї.

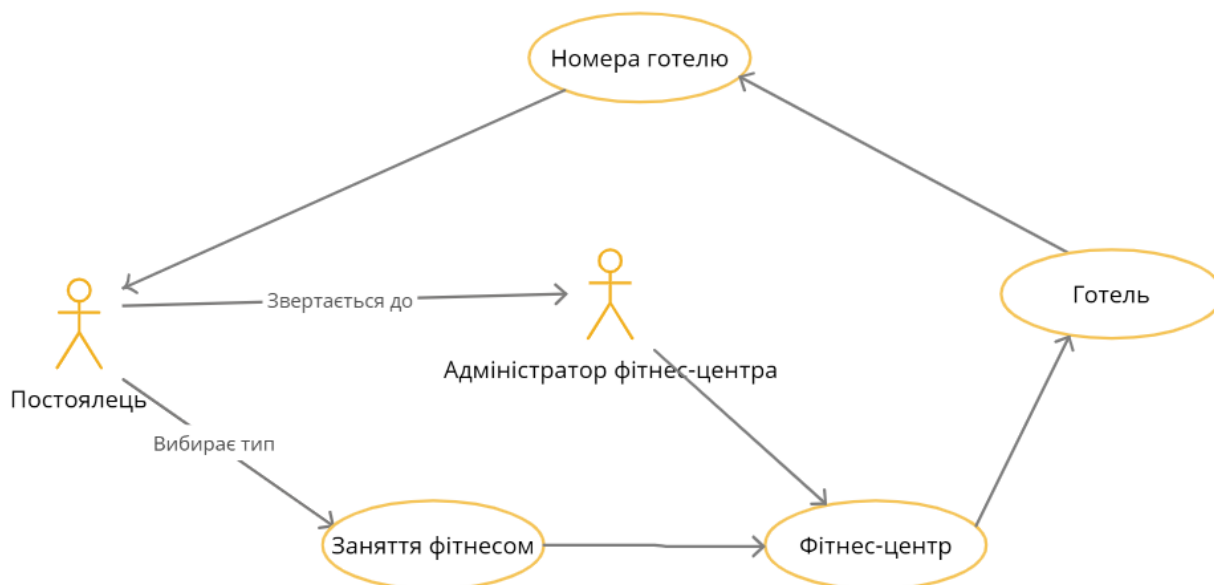


Рисунок 1.5 Опис процесу заняття фітнесом

Далі Постялець може користуватися всіма послугами, що пропонуються готелем.

Наприклад, кожен постялець може користуватися послугами із заняття фітнесом у залах, що є у складі готельного комплексу: тренажери, аеробіка, басейн, сауна, пляж і т.д. (Рис 1.5).

В даному випадку Постялець Готелю вибирає зі списку можливих типів занять фітнесом необхідне, далі звертається до Адміністратора Фітнес-центру, який призначає Заняття для Постяльця.

Процес проживання може супроводжуватись організованим процесом комплексного харчування Постяльця у Ресторані Готелі (рис. 1.6)

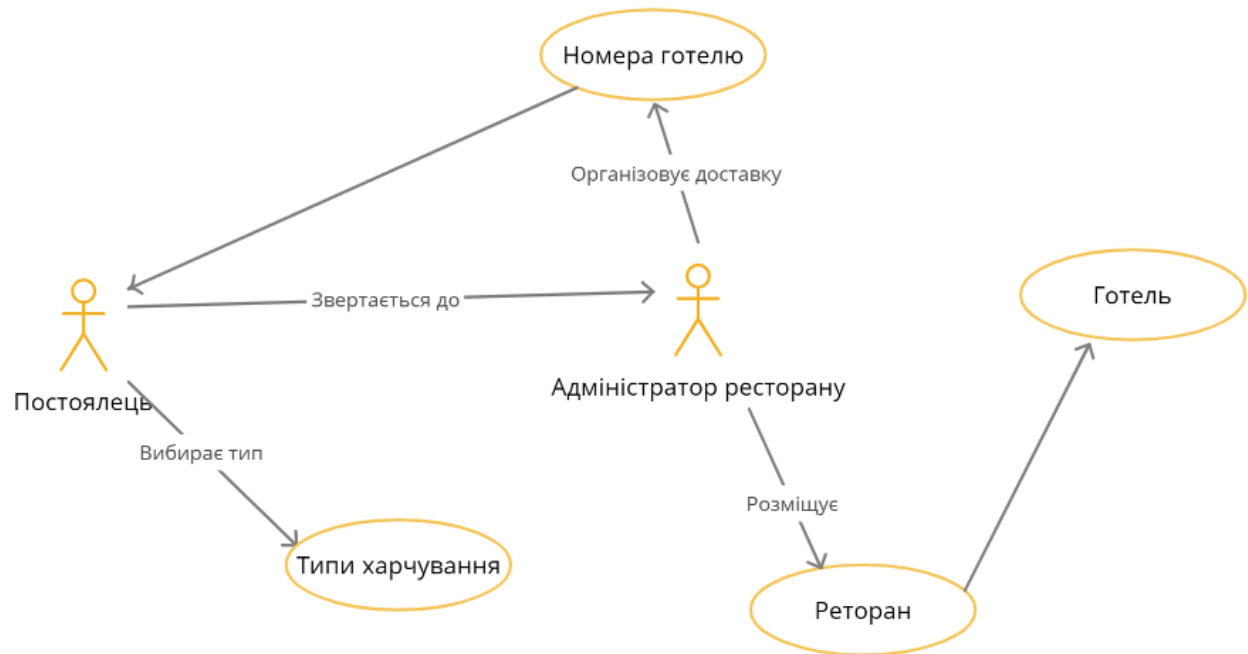


Рисунок 1.6 Опис процесу харчування у Ресторані

У цьому випадку Постялець Готелю також вибирає зі списку можливих типів харчування (сніданок, обід, вечерю, десерт) необхідне, далі він звертається до Адміністратора Ресторана, який призначає Клієнта на харчування в Ресторані або на прохання Клієнта організує доставку Харчування в готельний номер.

Після використання всіх послуг за даними преїскуранту на готельні послуги Постяльцеві формується та виписується рахунок на оплату:

1.

- Кількість Днів = Дні розміщення + Дні Продовження;
- Ціна Дня = Ціна Номери * Коефіцієнт за Тип * Коефіцієнт за Категорії * Коефіцієнт за Опції;
- Кількість мешканців за договором Постяльця;
- Сума по Рахунку на Проживання = Кількість Днів * Ціна Дня * Кількість мешканців.

2. Рахунок на заняття Фітнесом:

- Кількість годин занять по типу №1 фітнесу;
- Кількість годин занять по типу №2 фітнесу;
- Кількість годин занять по типу №N фітнесу;
- Ціна години занять по типу №1 фітнесу;
- Ціна години занять по типу №2 фітнесу;
- Ціна години занять по типу №N фітнесу;
- Кількість людей від одного Постояльця по типу №1 фітнесу;
- Кількість людей від одного Постояльця по типу №2 фітнесу;
- Кількість людей від одного Постояльця по типу №N фітнесу;
- Сума за рахунком за Фітнес = Кількість годин занять за типом №1-N фітнесу *
Ціна години занять за типом №1-N фітнесу * Кількість людей від одного
Постояльця по типу №1-N фітнесу.

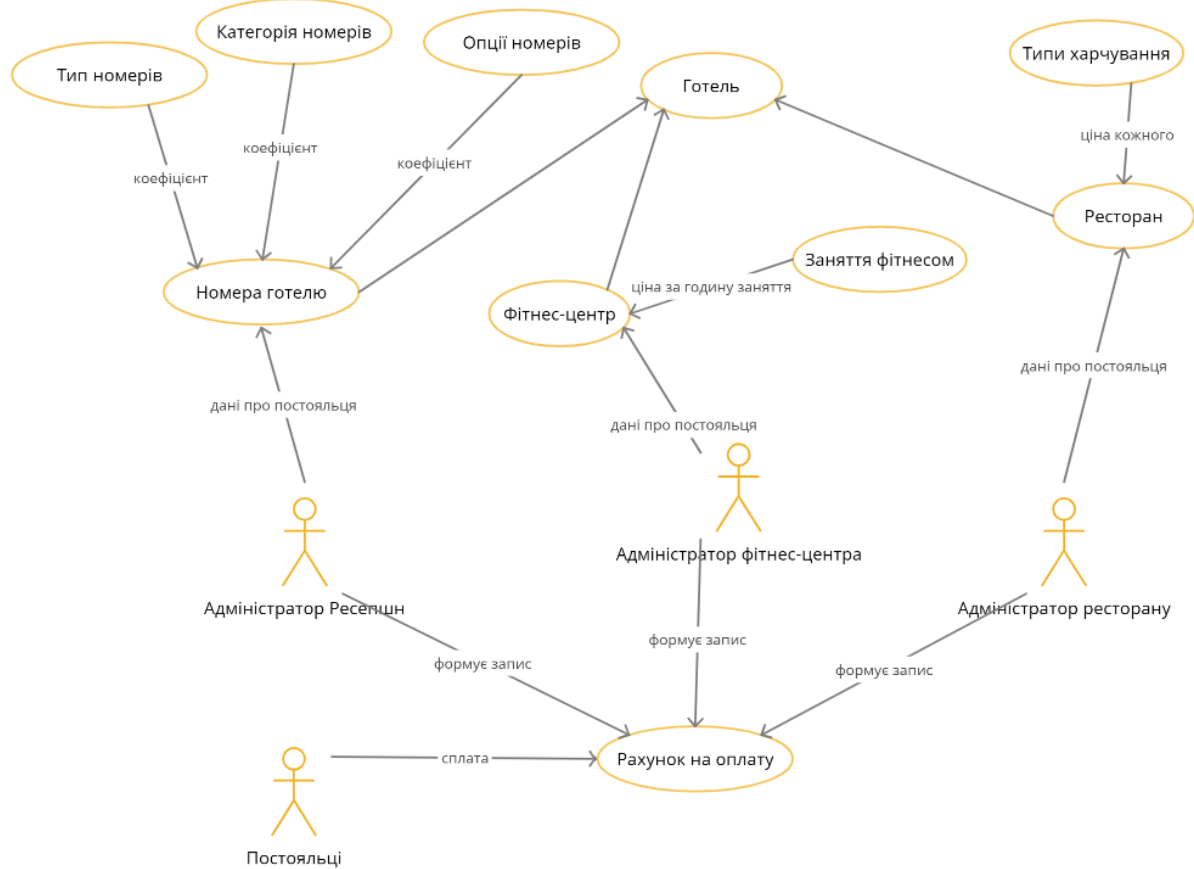


Рисунок 1.7 Процес формування рахунку на оплату

3. Рахунок за харчування:

- Кількість разів харчування за типом №1;
- Кількість разів харчування за типом №2;
- Кількість разів харчування на типом №N;
- Ціна разового харчування за типом №1;
- Ціна разового харчування за типом №2;
- Ціна разового харчування за типом №N;
- Кількість людей від одного Постояльця типу №1 харчування;
- Кількість людей від одного Постояльця типу №2 харчування;
- Кількість людей від одного Постояльця типу №N харчування;

- Сума по Рахунку за Харчування = Кількість разів харчування за видом №1-N * Ціна разового харчування за видом №1-N * Кількість людей від одного Постояльца виду №1-N харчування.

Таким чином: Підсумковий Рахунок за Готельні Послуги: Рахунок на проживання + Рахунок за Заняття Фітнесом + Рахунок за харчування.

Після звільнення номера Постояльцем необхідно упорядкувати його і зробити його «вільним». Процес «Прибирання та обслуговування номерів» відбувається у кілька етапів:

- При проживанні постояльца проводиться поточне прибирання приміщень (щоденне та генеральне, допустимо щотижневе) – при цьому готельний номер залишається «зайнятим»;
- При від'їзді постояльца проводиться підсумкове прибирання та готельний номер переводиться до категорії «вільні»;
- При досягненні якогось незадовільного експлуатаційного чи технічного стану номера після відбуття постояльца в номері проводиться ремонт – номер переводиться в категорію «На ремонті» та в числі вільних не числиться;
- Процес поточного ремонту номерів у системі не враховується.

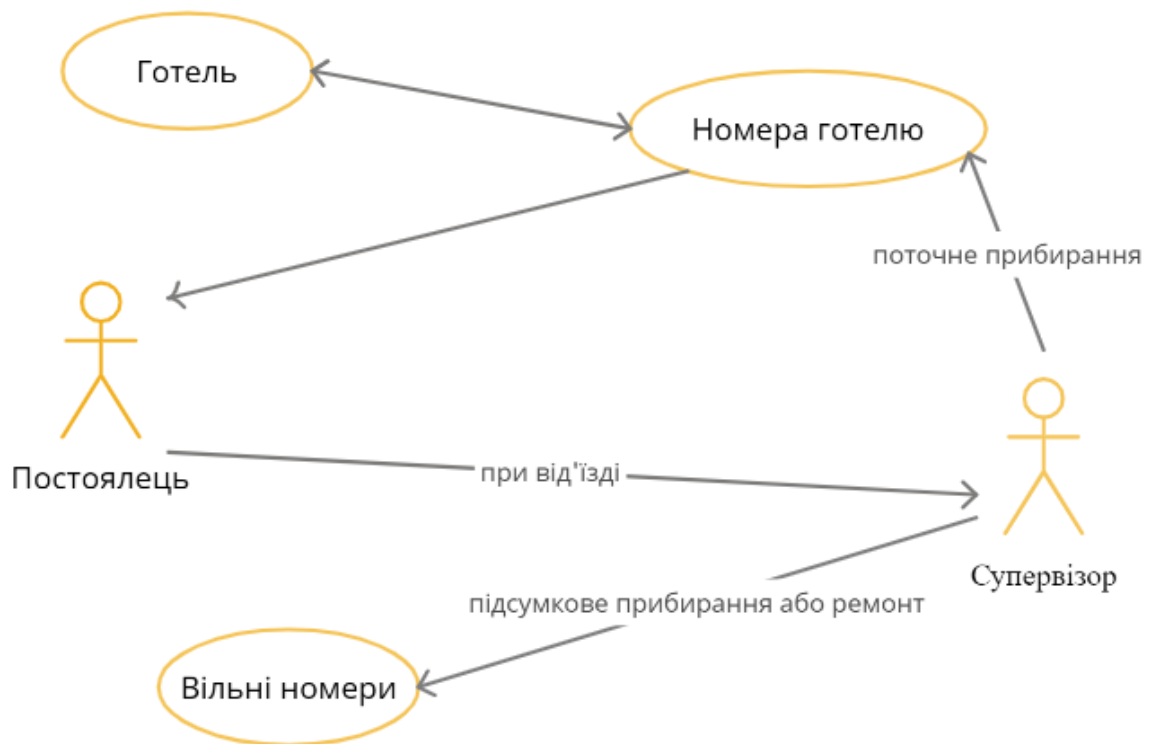


Рисунок 1.8 Прибирання та обслуговування номерів

Зважаючи на те, що проектується інформаційна система, необхідно також розглянути роль її системного Адміністратора.



Рисунок 1.9 Структура діяльності системного адміністратора

Системний адміністратор веде заповнення, коригування та обслуговування всіх вихідних довідників та робочих інформаційних таблиць.

Таким чином, кожен процес реалізує завдання, пов'язані між собою, зумовлюючи загальну структуру системи, що проектується.

1.2.2 Рівень розвиненості функції комунікацій в інформаційних системах як одна з конкурентних переваг

Фахівці зазначають, що в європейському готельному бізнесі широко поширені системи управління. Перш за все це локальні мережі, які централізовано дозволяють автоматично адаптувати номер під клієнта. Робота таких мереж полягає у наступному.

Коли в готель приїжджає клієнт - іноземець, співробітники готелю прямо зі стійки реєстрації адаптують інфраструктуру номера, в який прямує гість, під його рідну мову.

Так що клієнт бачить, наприклад, меню телебачення та IP-телефону своєю рідною мовою. Це, безумовно, підвищує престиж готелю.

Бездротовий доступ до Інтернету настільки ж обов'язковий для готелю, як і меблі під старовину. Близько 70% всіх готелів Києва пропонують своїм клієнтам бездротовий або високошвидкісний доступ до Інтернету.

Готелі змагаються між собою за кількістю впроваджених технологій та витрачають дедалі більше грошей на IT-рішення. Наприклад, у проекті готелю Novotel на IT-інфраструктуру витратили \$1 000 000, при тому, що загальна сума інвестицій у проект склала близько \$30 000 000.

У «Гранд Готелі Європа» було проведено роботи з покриття бездротовою мережею всього готелю. Завдяки цьому підключитися до Інтернету можна з будь-якої точки готелю: бару, ресторану, холу тощо.

В даний час близько 99% нових ноутбуків та мобільних пристроїв оснащені чіпами Wi-Fi, тому це дуже зручне для клієнтів нововведення.

Крім додаткового комфорту для клієнтів, це також дає величезний потенціал внутрішньому менеджменту готелю.

За допомогою бездротових систем можлива передача даних та голосу. Більшість готелів високого рівня поки що впроваджують бездротові системи в певних секторах готелю: тільки на бізнес-території або тільки в номерному фонді.

У готелі встановлена спеціальна система автоматизації управління номерним фондом, яка обмінюється даними із системами Accor Hotels та Global Distribution System, що дозволяє не тільки оперативно відстежувати та виконувати замовлення на бронювання номерів, а й відстежувати пропозиції конкурентів та змінювати

тарифи протягом доби. Системи Wi-Fi набули широкого поширення в українських готелях 4-5 роки тому.

На даний момент послуга бездротового доступу в Internet є такою ж обов'язковою умовою для готелів бізнес- та преміум-класів, як гаряча вода або кондиціонер у номері.

Все частіше великі готелі перетворюються на так звані інтелектуальні будівлі: системами кондиціонування або освітлення можна керувати централізовано з пульта диспетчера або по телефону кожного номера.

Для цього готелі удосконалюють не лише інженерні системи, а й запроваджують IP-телефонію.

Популярністю користується і телебачення IP-TV, за допомогою якого можна вибирати улюблені фільми та передачі з бази готелю.

Internet дає можливість готелям не лише надавати клієнтам послугу дистанційного самостійного бронювання номерів, а й дозволяє керівництву готелів вивчати ринок та миттєво підлаштовуватися під нього.

Незважаючи на те, що важливість систем автоматизації для сучасної індустрії гостинності безперечна, проте, слід зазначити, що використання АСУ не впливає безпосередньо на рішення гостя про вибір того чи іншого готелю.

Найважливішим у готелі, як і раніше, є не наявність автоматизованої системи управління, створеної за останнім словом техніки, а атмосфера затишку і гостинності, що панує в готелі, поряд з професійним обслуговуванням.

1.3 Аналіз інформаційних систем, що використовуються в управлінні в сімейних готельних мережах

Ефективне функціонування сімейних готельних мереж неможливо без використання спеціалізованого програмного забезпечення, що дає можливість автоматизувати всі бізнес-процеси. Сьогодні ми маємо вимоги до автоматизованих систем управління для підприємств індустрії гостинності. Такі інформаційні технології мають комплексний характер, складаються з комп'ютерної системи бронювання, системи проведення телеконференцій, відео-систем, інформаційних систем управління, електронних інформаційних систем авіаліній, електронної пересилки грошей, телефонних мереж, глобальних комп'ютерних систем резервування, системи мультимедіа, смарт-карток. Одним із основних напрямів розвитку сімейного готельного бізнесу в Україні вважається впровадження інформаційних комп'ютерних технологій управління та сучасних систем бронювання на основі вивчення та застосування досвіду розвинених країн у цій області. Основними комп'ютерними системами сімейних готельних мереж, що використовуються: комплекс найбільшої у світі компанії «Micros-Fidelio», Система Fidelio Front Office (FFO) допомагає автоматизувати основні етапи роботи готелю, системи Amadeus, Galileo, Sabre і Worldspan. Основними перевагами глобальної системи Amadeus виступають: заощадження часу завдяки можливості отримання усієї туристичної, клієнтської і агентської інформації; нова інформація поступає кожну годину в режимі on-line; зростаюча продуктивність дозволяє робити інформацію доступною; економія витрат за рахунок стабільної роботи і негайних підтверджень; підвищення прибутків завдяки широкому діапазону можливостей, які забезпечують задоволення усього спектру замовлень клієнта.

Розглянемо різні практичні приклади впровадження АСУ у вітчизняних та зарубіжних готелях.

1. Готель Grand Hyatt Mumbai через два роки після встановлення програмного забезпечення TRITON починає аналізувати результати роботи з ним. TRITON-продукт компанії Knowcross був встановлений у Grand Hyatt Mumbai у жовтні 2010. Система створена для розподілу дзвінків у центрі комунікації та для управління роботами всередині готелю.

Результатом використання програми TRITON стало зменшення часу, протягом якого співробітник готелю відповідає на дзвінок та підвищення ефективності роботи.

2. Менеджери п'ятизіркового готелю «Краун плаза» (Crowne Plaza) у столиці Ірландії – Дубліні поставили собі за мету підвищити заповнюваність готелю (іншими словами – його рентабельність) і при цьому знизити операційні витрати, використовуючи рішення компанії Cisco для готельного бізнесу під назвою Connected Hotel, в основі якого - потужний потенціал інтелектуальних IP-мереж, що дозволяють одночасно пересилати дані та підтримувати голосовий та відеозв'язок.

У системі передбачений і бездротовий високошвидкісний Інтернет-доступ у будь-якій точці бізнес-центру. За допомогою простих у використанні IP-телефонів постояльці «Краун плаза» можуть прийняти та відправити електронні повідомлення, ознайомитися з інформацією про послуги, що пропонуються готелем, дізнатися прогноз погоди, замовити таксі. Крім цього, в IP-телефонах передбачені персональні налаштування: робота різними мовами з індивідуальними вітальними повідомленнями та навіть логотипи конкретних компаній.

Інтелектуальна інформаційна мережа, що лежить в основі рішення Connected Hotels, дає готелям можливість запропонувати постояльцям набір інноваційних послуг, що постійно розширюється, ефективніше використовувати людські ресурси,

налагодити більш чіткий контроль за роботою всіх готельних служб. За рахунок надійного автоматичного доступу до широкого спектру різноманітних послуг для постояльців, людських ресурсів та зручностей, що пропонуються в готельних будинках, потужна інтелектуальна IP-мережа перетворює готель на повнофункціональне робоче середовище. Застосування спільно розроблених технологій та формування нових сервісів, зокрема, мобільної телефонії та високошвидкісного доступу до Internetu, дозволяє значно підвищити продуктивність готельного персоналу та оперативність запропонованих постояльцям послуг. Це, у свою чергу, підвищує лояльність клієнтів та допомагає створити стійку клієнтську базу.

Надання багатофункціональних послуг бездоганної якості, зокрема обладнання готельних номерів модернізованими розважальними системами – принципово нове джерело доходів, що значно підвищує рентабельну наповнюваність готелів. Рішення Connected Hotels об'єднує сервіси автоматизації управління будівлею та корпоративну мережу на єдиній платформі Cisco, завдяки чому знижуються як капітальні інвестиції, так і операційні витрати.

2 КОНЦЕПТУАЛЬНА СТРУКТУРА СИСТЕМИ ФУНКЦІЇ КОМУНІКАЦІЇ

На етапі проектування необхідно орієнтуватися на спосіб використання додатка працівниками сімейної готельної установи, на вбудовування інформаційних систем у реальні технологічні процеси функціонування мережі сімейних готелів.

Перш за все, у нашому випадку, інформація може потрапити до бази даних системи з первинної документації;

Дані про характеристики можуть бути внесені до комп'ютера зі слів персоналу (в даному випадку – менеджери), а документ (у разі потреби) буде сформовано системою автоматично.

Доступ до бази даних у програмній оболонці передбачається здійснити у вигляді використання описаного вище інтерфейсу бази у зв'язку з технологіями програмування .NET. Вибір даної технології обумовлений опрацьованістю та стабільністю зв'язку БД, що підтримуються сучасними Windows-системами. Переважною СУБД є Microsoft SQL Server, у локальній версії Express, використання якої зумовлює швидку, точну і безпомилкову обробку даних, якими оперує програма.

2.1 Визначення складу бази даних

Функціональна модель призначена для опису існуючих бізнес - процесів на підприємстві та ідеального стану речей - того, чого потрібно прагнути.

Методологія IDEF0 наказує побудову ієрархічної системи діаграм - одиничних описів фрагментів системи.

Функціональний блок перетворює Входи на Виходи (тобто вхідну інформацію у вихідну). Управління визначає, коли і як це перетворення може, чи має статися.

Побудова моделі ІС починається з опису функціонування системи загалом як контекстної діаграми. Далі представлена основна логічна діаграма бази даних, що проектується. Відповідно до правил відображення, зв'язки виду «один-багатьом» у діаграмі розташовані зверху вниз.

Блок А0 – функціонування готелю (див. рис. 2.1). Вхід – дані клієнта. Вихід – чек, звіт. Управління – закон про права споживача, перелік програм, правила організації.

Механізм - клієнт, менеджер, програмне забезпечення, клієнтська база, технічні засоби, персонал.

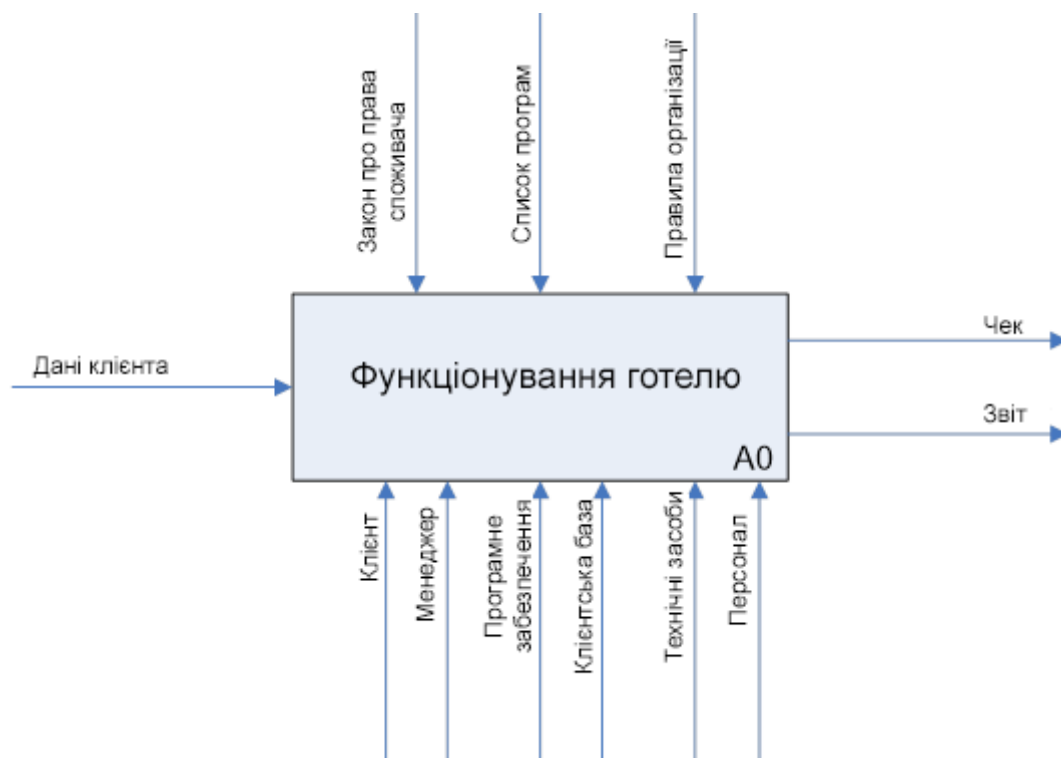


Рисунок 2.1 А-0. Функціонування готелю

До блоку А-0 входять блоки А1, А2, А3.

Блок А1-реєстрація клієнта в готелі. Вхід – дані клієнта. Вихід – номер. Управління – закон про права споживача, правила організації. Механізм - клієнт, менеджер, клієнтська база.

Блок А2-надання послуг. Вхід номер. Вихід – список послуг. Управління – закон про права споживача, перелік програм, правила організації. Механізм – персонал.

Блок А3-оформлення звіту. Вхід – список послуг. Вихід – чек, звіт. Управління – закон про права споживача, перелік програм, правила організації. Механізм - менеджер, програмне забезпечення, технічні засоби.

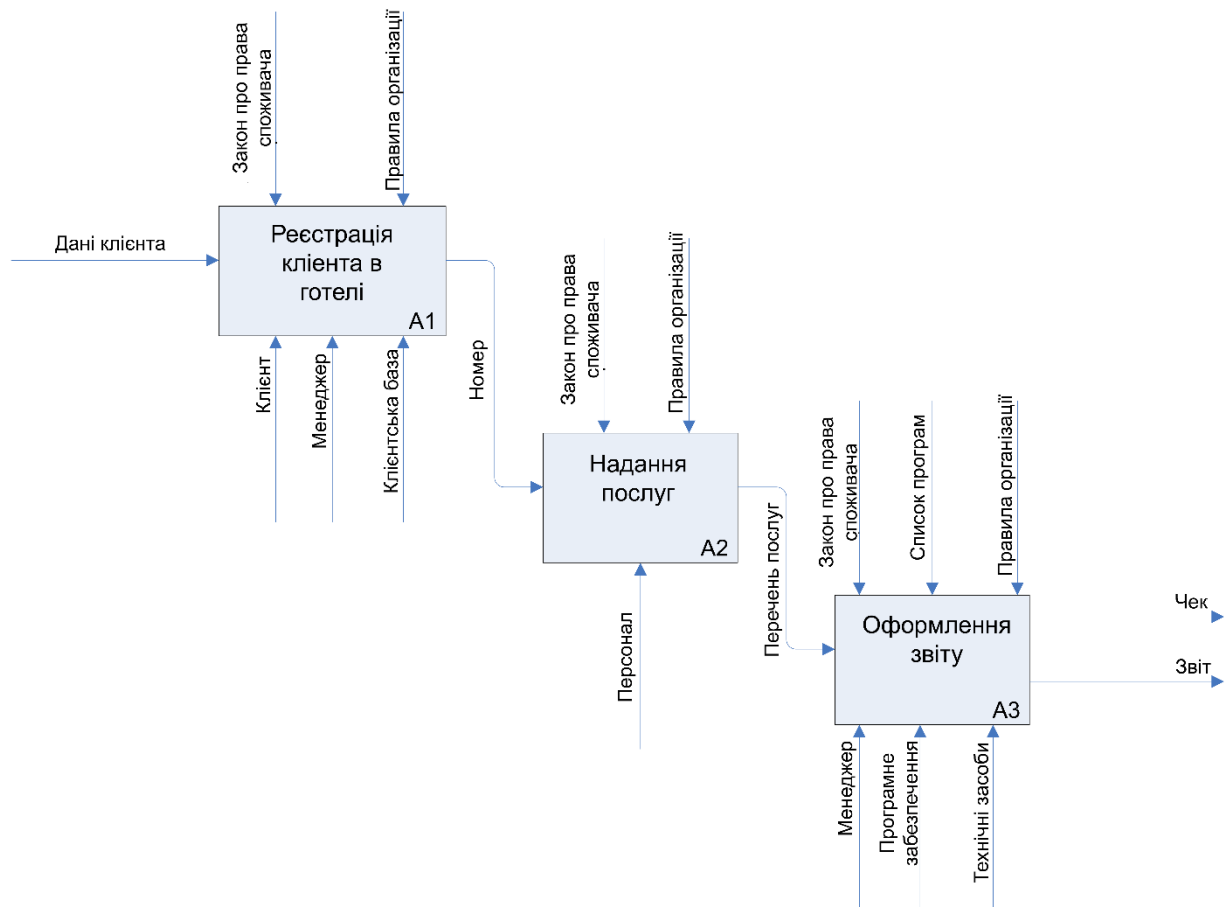


Рисунок 2.2 Блок А0. Процес надання готельних послуг

Блок А1 своєю чергою підрозділяється на блоки А11, А12, А13.

Блок A11 - резервування номера. Входом будуть дані клієнта. Управління – закон про права споживача. Механізм-клієнт. Вихід-список порожніх номерів.

Блок A12 - оформлення поселення. Вхід-список порожніх номерів. Управління – правила організації. Механізм – менеджер. Вихід – узгодження.

Блок A13-прийом передоплати. Вхід-узгодження. Управління-правила організації. Механізм-менеджер. Вихід-номер.

Блок A2 поділяється на блоки A21, A22.

Блок A21 - надання стандартних послуг. Вхід-номер. Управління – закон про права споживача, правила організації.

Механізм-персонал. Вихід-список стандартних послуг.

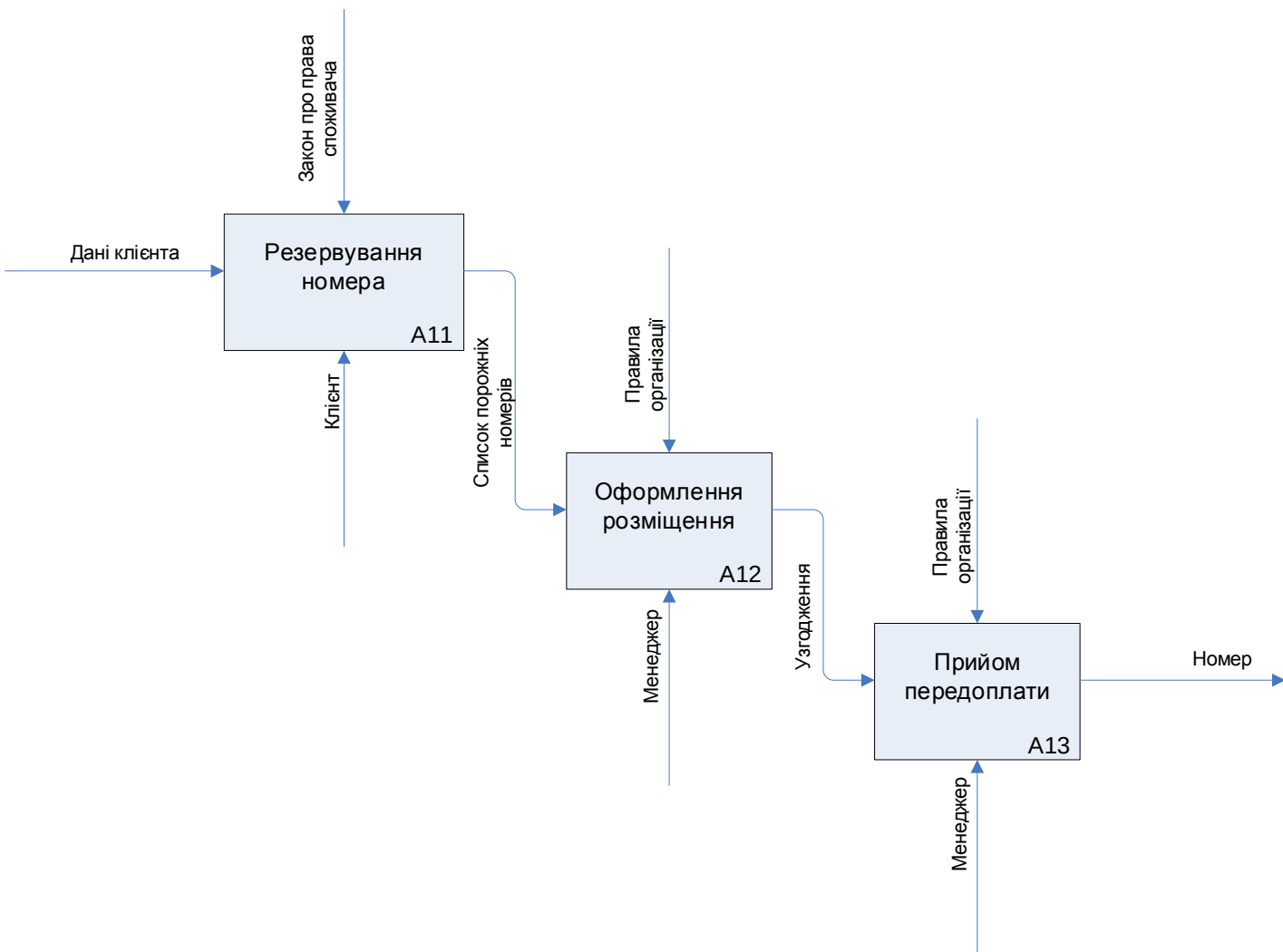


Рисунок 2.3 Блок А1 Процес розміщення гостей

Блок А22-надання додаткових послуг:

- Вхід-список стандартних послуг
- Управління-закон про права споживача, правила організації.
- Механізм-персонал. Вихід-перелік наданих послуг.

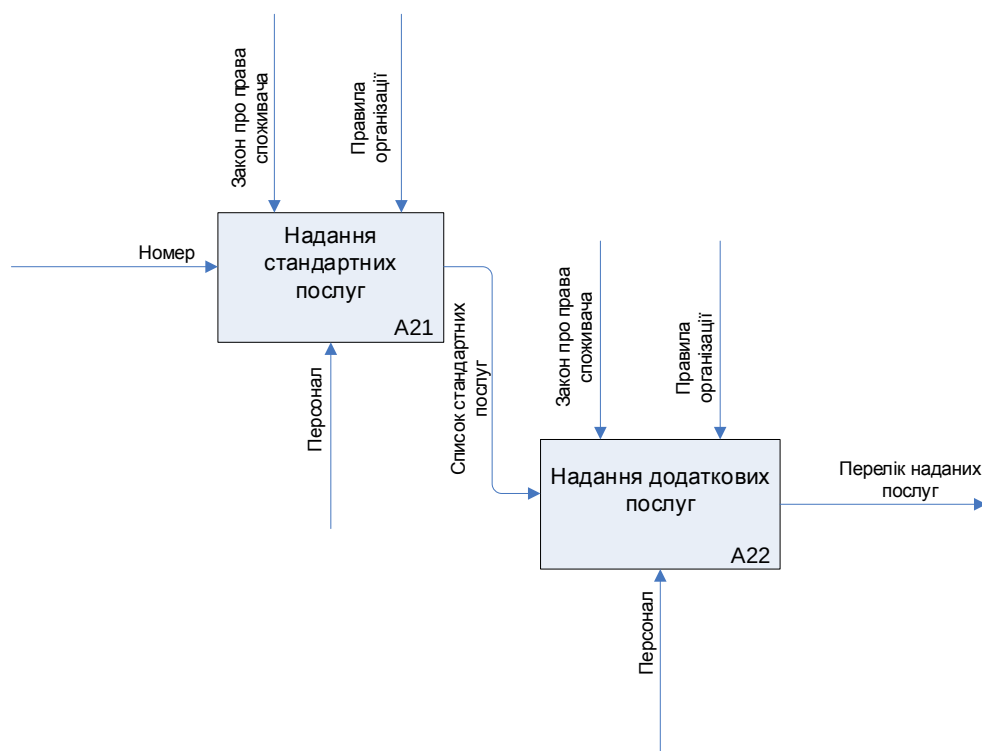


Рисунок 2.4 Блок A2 Процес надання послуг

На підставі виробленої декомпозиції бази даних виявлено такі об'єкти, з'єднані в діаграмі класів системи:

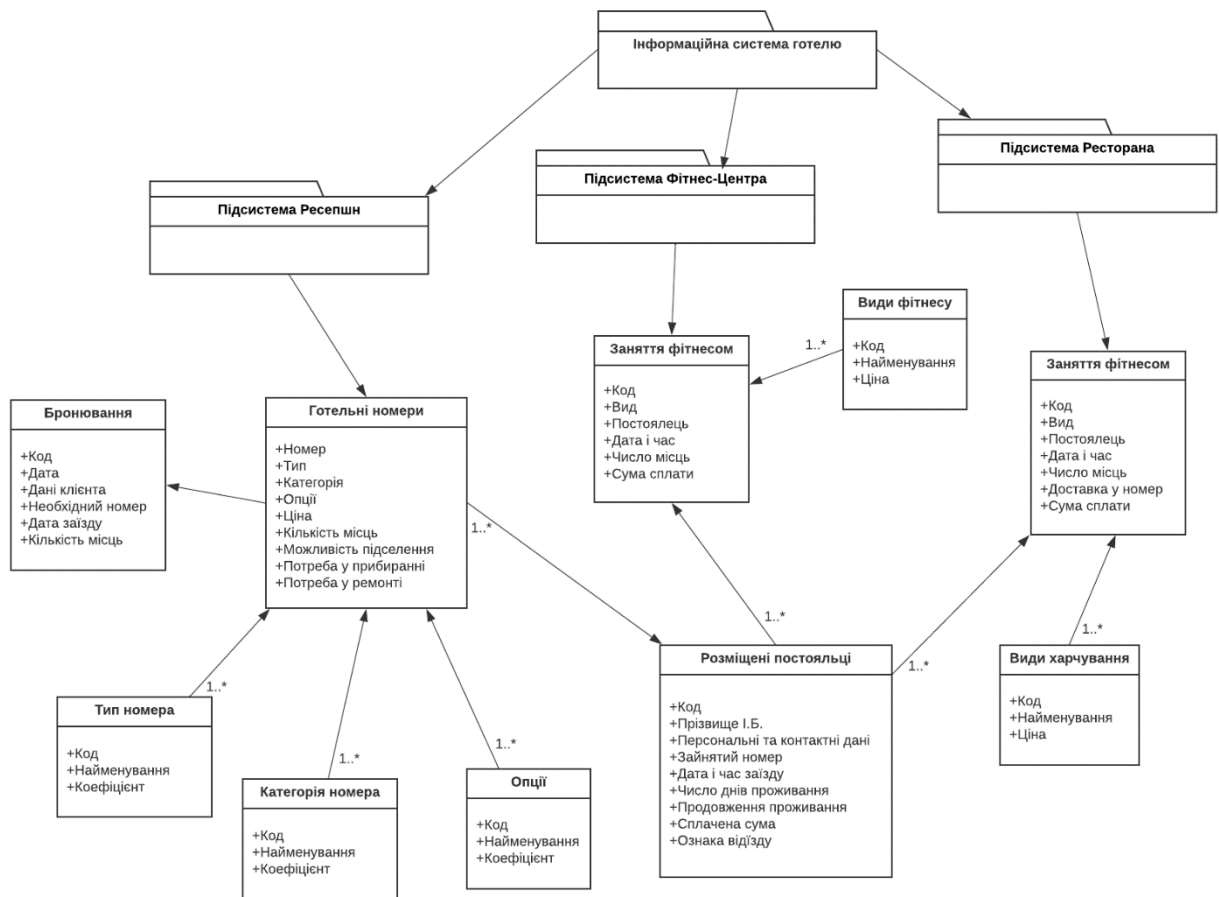


Рисунок 2.5 ER-модель інформаційної системи

У цій діаграмі класів виділено всі необхідні інформаційні об'єкти та супутні їм інформаційні атрибути та характеристики, на підставі яких можна приступити до розробки та фізичного проектування бази даних «Мережа Готель».

Як було зазначено раніше, основною метою та призначенням проектованої інформаційної системи є забезпечення ефективності функціонування системи на досліджуваному закладі – типовому готельному комплексі.

Всі подальші міркування ґрунтуються на правилах побудови попередніх відносин із діаграм ER-типу для реляційних баз даних.

2.2 Фізичне проектування бази даних

На підставі проведеної декомпозиції для бази даних «Готельна Мережа» було виявлено такі інформаційні об'єкти та таблиці даних:

Таблиця 2.1 Об'єкти Мережі Сімейних Готелів

Найменування таблиці	Ім'я поля	Тип даних	Опис
HotelNet	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (15)	Найменування
	Note	Текстовий (10)	Найменування

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Таблиця «Об'єкти Мережі Готелів» є переліком готелів, що входять у готельну мережу. При цьому перший запис є адміністрацією мережі готелів (не є готелем).

	Kod	Name	Note
+	1	Комплекс	Адміністрація мережі
▶	2	Готель №1	Готель №1, адреса, опи
+	3	Готель №2	Готель №2, адреса, опи
+	4	Готель №3	Готель №3, адреса, опи
+	5	Готель №4	Готель №4, адреса, опи
+	6	Готель №5	Готель №5, адреса, опи
*			

Запись: 2 из 6

Рисунок 2.6 Об'єкти Мережі Сімейних Готелів

Для роботи з інформаційною системою передбачено ряд таблиць, в яких зберігаються дані щодо користувачів системи – спеціалістів адміністрації мережі готелів та спеціалістів кожного конкретного готелю.

Таблиця 2.2 Класифікатор посад спеціалістів

Найменування таблиці	Ім'я поля	Тип даних	Опис
Posada	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (30)	Найменування

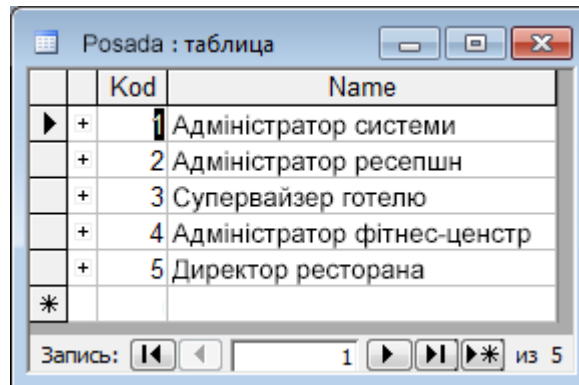


Рисунок 2.7 Класифікатор посад спеціалістів

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Таблиця 2.3 Класифікатор спеціалістів

Найменування таблиці	Ім'я поля	Тип даних	Опис
Personal	Kod	Лічильник	Ідентифікатор
	Prizvyshche	Текстовий (30)	Прізвище
	Imya	Текстовий (20)	Ім'я
	PoBatkovi	Текстовий (25)	По батькові

	Posada	Числовий	Код посади
	Parol	Текстовий (10)	Пароль доступу
	Hotel	Числовий	Код готелю

Kod	Prizvyshche	I	B	Posada	Parol	Hotel
1	Домрачева	Катерина	Ігорівна	Адміністратор системи		Комплекс
2	Погрибняк	Володимир	Степанович	Адміністратор ресепшн		Готель №1
3	Мельник	Ігорь	Сергійович	Супервайзер готелю		Готель №1
4	Ковальчук	Степан	Миколайович	Адміністратор фітнес-центру		Готель №1
*						

Запись: 1 из 4

Рисунок 2.8 Класифікатор спеціалістів

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Також у таблиці присутні пов'язані поля для отримання текстових значень за кодом з інших таблиць: найменування посади спеціаліста та найменування готелю (або адміністрації комплексу), в якому даний спеціаліст працює.

Таблиця 2.4 Протокол доступу

Найменування таблиці	Ім'я поля	Тип даних	Опис
Dostup	Kod	Лічильник	Ідентифікатор
	Dostup	Числовий	Код спеціаліста
	DataIn	Дата та Час	Дата та Час входу у системи
	DataOut	Дата та Час	Дата та Час виходу із системи

Для відстеження протоколу доступу фахівців у систему передбачено спеціальну таблицю «Протокол доступу»:

	Kod	Dostup	DataIn	DataOut
▶		Домрачева К.І.	15.12.2022 19:41:25	15.12.2022 21:40:15
*		0	15.12.2022 19:41:52	

Рисунок 2.9 Протокол доступу

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Також у таблиці є пов'язане поле для отримання текстових значень Прізвища І.Б. відповідального фахівця з коду з таблиці «Класифікатор спеціалістів».

В даному випадку для формування текстового відображення фахівця у форматі «Прізвище І.Б.» застосовано вираз із використанням текстових функцій бази даних:

```
SELECT          Personal.Kod,          Personal.Prizvyshche+"
"+Left(Personal.Imya,1)+"."+Left(Personal.PoBatkovi,1)+"." AS Name FROM Personal;
```

Для відображення дати та часу входу в систему та виходу з неї використані поля дати/часу з форматом «Повний формат дати»

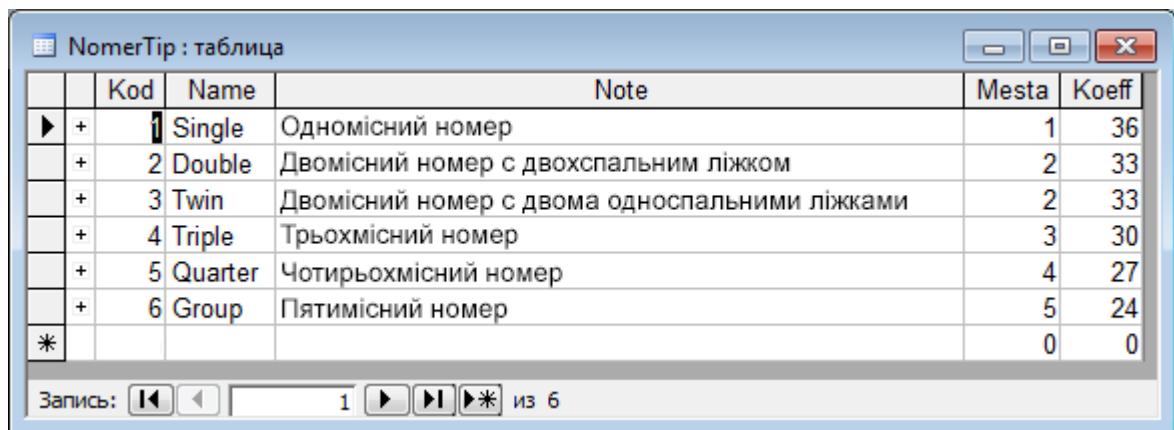
При цьому для дати входу використана можливість бази даних автоматичного внесення дати та часу за умовчанням з використанням виразу з функціями дати та часу: «Date()+Time()».

Дата виходу в будь-який момент невідома, тому необхідне поле в таблиці даних передбачається заповнювати за фактом виходу засобами програмної програми при його завершенні для користувача бази даних.

Далі було створено три аналогічні інформаційні довідники, що зберігають інформацію про сучасні параметри готельних номерів.

Таблиця 2.5 Типи номерів

Найменування	Ім'я поля	Тип даних	Опис
NomerTip	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (15)	Найменування
	Note	Текстовий (50)	Опис
	Mesta	Числовий	Число місць
	Koeff	Числовий	Коефіцієнт



	Kod	Name	Note	Mesta	Koeff
+	1	Single	Одномісний номер	1	36
+	2	Double	Двомісний номер с двохспальним ліжком	2	33
+	3	Twin	Двомісний номер с двома односпальними ліжками	2	33
+	4	Triple	Трьохмісний номер	3	30
+	5	Quarter	Чотирьохмісний номер	4	27
+	6	Group	П'ятимісний номер	5	24
*				0	0

Запись: 1 из 6

Рисунок 2.10 Типи номерів

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Також у таблиці присутні текстові найменування та опис типу номера, числова кількість місць та підвищуючий коефіцієнт. Цей коефіцієнт прийнятий як дослідна величина для отримання вартості проживання шляхом множення на певний номінал проживання по одному місцю в номері для кожного окремого готелю.

Таблиця 2.6 Категорії номерів

Найменування таблиці	Ім'я поля	Тип даних	Опис
----------------------	-----------	-----------	------

NomerCategory	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (15)	Найменування
	Note	Текстовий (50)	Опис
	Koeff	Числовий	Коефіцієнт

	Kod	Name	Note	Koeff
+	1	Standard	Стандартний номер	0
+	2	Superior	Номер більшого розміру, ніж Standart	1
+	3	DeLuxe	Номер більшого розміру, ніж Superior	2
+	4	MiniSuite	Номер Superior більшої комфортності	3
+	5	Suite	Номер MiniSuite з зоною для відпочинку	4
+	6	JuniorSuite	Номер MiniSuite з відгородженим спальним місцем	5
+	7	SeniorSuite	Номер MiniSuite двокімнатний	6
+	8	ExecutiveSuite	Номер MiniSuite з двома спальнями	7
+	9	KingSuite	Номер MiniSuite з двома спальнями та кімнатами	8
+	10	Honeymoon	Номер для наречених	9
+	11	Appartment	Номер з спальнями, кімнатами та кухнею	10
*				0

Запись: 1 из 11

Рисунок 2.11 Категорії номерів

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Також у таблиці присутні текстові найменування та опис категорії номера, числовий та підвищуючий коефіцієнт. Цей коефіцієнт також прийнятий як дослідна величина для отримання вартості проживання шляхом множення на певний номінал проживання по одному місцю в номері для кожного окремого готелю.

Таблиця 2.7 Додаткові опції

Найменування таблиці	Ім'я поля	Тип даних	Опис
NomerOptions	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (15)	Найменування
	Note	Текстовий (50)	Опис
	Koeff	Числовий	Коефіцієнт

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Також у таблиці присутні текстові найменування та опис категорії номера, числовий та підвищуючий коефіцієнт. Цей коефіцієнт також прийнятий як дослідна величина для отримання вартості проживання шляхом множення на певний номінал проживання по одному місцю в номері для кожного окремого готелю.

	Kod	Name	Note	Koeff
▶ +	1	None	Без доповнень	0
+	2	Balcony	Номер з балконом	2
+	3	Connected	Сумістний номер	4
+	4	Duplex	Двоповерховий номер	6
+	5	Extra	Додаткове ліжко	2
+	6	Family	Для проживання сім'ї	12
+	7	Sitting	Номер з зоною відпочинку	8
*				0

Запись: 1 из 7

Рисунок 2.12 Додаткові опції

На підставі даних таблиці, була побудована таблиця «Готельні номери», що є основною інформаційною сутністю системи організації готельних номерів мережі сімейних готелів.

Таблиця 2.8 Готельні номери

Найменування таблиці	Ім'я поля	Тип даних	Опис
Nomers	Kod	Лічильник	Ідентифікатор
	Hotel	Числовий	Код готелю
	Nomer	Числовий	Номер (номера)
	Price	Грошовий	Ціна проживання
	TipNo	Числовий	Код типу номера
	Category	Числовий	Код категорії номера

	Options	Числовий	Код дод. опції
	Clean	Логічний	Необхідність прибирання
	Remont	Логічний	Номер знаходиться на ремонті

	Kod	Hotel	Nomer	Price	TipNo	Category	Options	Clean	Remont
▶ +	1	Готель №1	1	60.00грн.	Single	Standard	None	☐	☐
	2	Готель №1	2	60.00грн.	Single	Standard	Extra	☐	☐
	3	Готель №1	3	60.00грн.	Single	Superior	Balcony	☒	☐
	4	Готель №1	4	60.00грн.	Single	DeLuxe	Sitting	☐	☐
	5	Готель №1	5	60.00грн.	Double	Standard	Balcony	☐	☐
	6	Готель №1	6	60.00грн.	Double	MiniSuite	Extra	☐	☐
	7	Готель №1	7	60.00грн.	Twin	JuniorSuite	None	☒	☐
	8	Готель №1	8	60.00грн.	Triple	JuniorSuite	Balcony	☐	☐
	9	Готель №1	9	60.00грн.	Triple	Standard	Family	☐	☐
	10	Готель №1	10	60.00грн.	Quarter	Suite	Duplex	☐	☐
	11	Готель №1	11	60.00грн.	Quarter	SeniorSuite	Sitting	☐	☒
	12	Готель №1	12	60.00грн.	Group	Standard	None	☐	☐
	13	Готель №1	13	60.00грн.	Group	ExecutiveSuite	Extra	☐	☐
	14	Готель №2	1	70.00грн.	Single	Standard	None	☐	☐
	15	Готель №2	2	70.00грн.	Single	Standard	Extra	☐	☐
	16	Готель №2	3	70.00грн.	Single	Superior	Balcony	☐	☐
*			0	0.00грн.				☐	☐

Запись: 1 из 16

Рисунок 2.13 Готельні номери

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Також у таблиці є пов'язані поля для отримання текстових значень найменування готелю поточного номера за кодом з таблиці «Об'єкти Мережі Готель».

Також у таблиці є числове значення номера приміщення, грошове значення номіналу вартості проживання по одному місцю в даному готелі.

Поряд з цим у таблиці присутні пов'язані поля для отримання текстових та числових значень найменувань та коефіцієнтів за кодом з таблиць типів, категорій та додаткових опцій.

У таблиці «Готельні номери» також передбачено поля для введення та коригування логічних значень: «Необхідність виконання прибирання» та «Номер на ремонті». При дійсних значеннях даних полів виникає ситуація, що ці номери неможливо використовувати.

З даних ряду вищеописаних таблиць можна побудувати таблицю, зберігає інформацію про проживання клієнтів у конкретному номері конкретного готелю.

Таблиця 2.9 Реєстрація постояльців

Найменування таблиці	Ім'я поля	Тип даних	Опис
RegisterClients	Kod	Лічильник	Ідентифікатор
	Prizvyshche	Текстовий (30)	Прізвище
	I	Текстовий (20)	Ім'я
	B	Текстовий (25)	По батькові
	Phone	Текстовий (11)	Номер телефона
	Nomer	Числовий	Код номера
	Mesta	Числовий	Число місць
	Data	Дата/Час	Дата заїзду
	Hour	Дата/ Час	Час заселення
	Days	Числовий	Днів проживання
	Prod	Числовий	Продовження днів
	Splata	Грошовий	Сплатена сума
	Kinets	Логічний	Вибуття

	Kod	Prizvyshch	I	B	Phone	Nomer	Mesta	Data	Hour	Days	Prod	Splata	Kinets
▶ +	1	Іванов	Іван	Іванович	(050)097-68-45	1	1	15.12.2022	11:28	3	0	3 000.00грн	☐
+	2	Петров	Петр	Петрович	(096)546-34-15	4	2	16.12.2022	20:20	2	0		☐
+	3	Миколаїв	Микола	Ігорович	(097)352-32-46	6	2	17.12.2022	15:15	8	1		☐
+	4	Коваленко	Сергій	Сергійович	(050)548-25-73	6	1	18.12.2022	09:00	4	0		☐
*						0	0		00:30	0	0	0.00грн.	☐

Запись: 1 из 4

Рисунок 2.14 Реєстрація постояльців

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею. Далі в таблиці присутні текстові значення для зберігання контактних реквізитів постояльца, є пов'язане поля для отримання числового значення номера приміщення, в якому проживатиме постоялець.

На рівні проектування бази даних прийнято, що при реєстрації записується один постоялець, а також кілька осіб, які проживають разом із ним, для чого призначено числове поле «Число місць».

Для відображення дати заїзду та часу заселення використані поля дати/часу з форматом «Короткий формат дати» та «Короткий формат часу». При цьому, для дати заїзду та часу заселення використана можливість бази даних автоматичного внесення дати та часу за умовчанням з використанням виразу з функціями дати та часу: Date() та Time().

Крім того, для інформації про проживання призначені числові поля заявлених під час заселення «Днів проживання» та «Продовження днів» - за бажання постояльца продовжити проживання на вказану кількість днів. Також передбачено грошове поле «Оплачена сума» та логічне поле «Вибуття», що означає завершення проживання.

Для зберігання та обробки інформації про організацію харчування постояльців у готелі створено відповідні таблиці.

Таблиця 2.10 Види харчування

Найменування таблиці	Ім'я поля	Тип даних	Опис
Pitanie	Kod	Лічильник	Ідентифікатор

	Name	Текстовий (30)	Найменування
	Price	Грошовий	Вартість за особу

	Kod	Name	Price
▶ +	1	Сніданок	300.00грн.
+	2	Обід	200.00грн.
+	3	Вечеря	250.00грн.
+	4	Десерт	150.00грн.
*			0.00грн.

Запись: 1 из 4

Рисунок 2.15 Види харчування

У цій таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею, текстове поле, що зберігає найменування.

Числове поле "Вартість" відображає грошове значення вартості харчування з розрахунку на 1 особу за поточним видом харчування.

На підставі даних таблиць про види харчування та зареєстрованих постояльців створюється таблиця про конкретні факти надання послуг із харчування постояльців.

Таблиця 2.11 Організація харчування

Найменування таблиці	Ім'я поля	Тип даних	Опис
Nomers	Kod	Лічильник	Ідентифікатор
	Hotel	Числовий	Код готелю
	Kharchuvannya	Числовий	Код харчування
	Client	Числовий	Код постояльца
	Mesta	Числовий	Кількість порцій
	Nomer	Логічний	Доставка у номер
	Data	Дата/Час	Дата харчування
	Hour	Дата/Час	Час харчування

	Splata	Грошовий	Сплачена сума
--	--------	----------	---------------

Kod	Hotel	Pitanie	Client	Mesta	Nomer	Data	Hour	Splata
1	Готель №1	Сніданок	Іванов І.І.	2	☐	18.12.2022	12:10	300.00грн
2	Готель №1	Обід	Іванов І.І.	2	☐			
3	Готель №1	Вечеря	Іванов І.І.	2	☐			
4	Готель №1	Десерт	Іванов І.І.	1	☒			
*			0	0	☐			0.00грн.

Рисунок 2.16 Організація харчування

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Далі в таблиці є пов'язане поле для отримання текстових значень найменування готелю поточного номера за кодом з таблиці «Об'єкти Мережі Готель».

Далі в таблиці є пов'язане поле для отримання текстових значень найменування та грошових значень виду харчування поточної послуги з харчування за кодом з таблиці «Види харчування». Також у таблиці є пов'язане поле для отримання текстових значень Прізвища І.Б. постояльця за кодом із таблиці «Реєстрація Постояльців». В даному випадку для формування текстового відображення фахівця у форматі «Прізвище І.Б.» застосовано вираз із використанням текстових функцій бази даних: `SELECT RegisterClients.Kod, RegisterClients.Prizvyshche+" "+Left(RegisterClients.Imya,1)+". "+Left(RegisterClients.PoBatkovi,1)+". " AS Name FROM RegisterClients;`

Так як у готелі оформлений один відповідальний постоялець, то на його ім'я вказується числова кількість місць (порцій) для харчування.

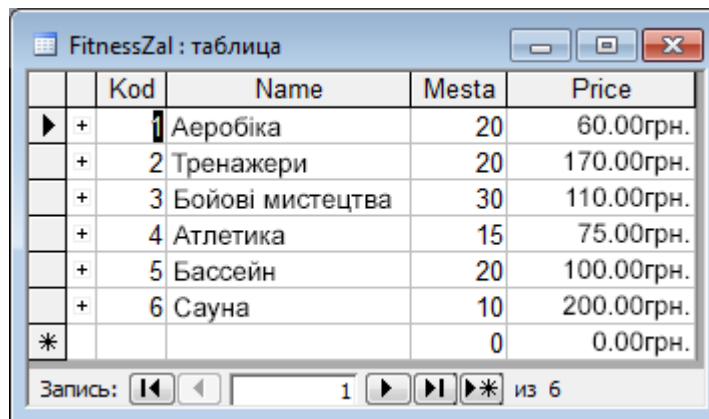
Для відображення дати та часу живлення використані поля дати/часу з форматом «Короткий формат дати» та «Короткий формат часу».

Для оформлення фактичної оплати передбачено спеціальне поле грошового формату, в якому реєструються суми, сплачені постояльцем до повного розрахунку при виселенні.

Для зберігання та обробки інформації про організацію здорового дозвілля постояльців у готелі створено відповідні таблиці.

Таблиця 2.12 Види та приміщення для фітнесу

Найменування таблиці	Ім'я поля	Тип даних	Опис
FitnessZal	Kod	Лічильник	Ідентифікатор
	Name	Текстовий (30)	Найменування
	Mesta	Числовий	Місць у залі
	Price	Грошовий	Вартість на 1 особу



	Kod	Name	Mesta	Price
+	1	Аеробіка	20	60.00грн.
+	2	Тренажери	20	170.00грн.
+	3	Бойові мистецтва	30	110.00грн.
+	4	Атлетика	15	75.00грн.
+	5	Басейн	20	100.00грн.
+	6	Сауна	10	200.00грн.
*			0	0.00грн.

Запис: 1 из 6

Рисунок 2.17 Види та приміщення для фітнесу

У цій таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею, текстове поле, що зберігає найменування.

Числове поле «Місця» відображає кількість місць у кожному «залі» з точки зору одного сеансу занять.

Числове поле «Вартість» відображає грошове значення вартості оздоровчих занять із розрахунку на 1 особу за поточним видом фітнесу.

Для організації занять фітнесом за видами в готелі передбачається створення попередньо організованих груп, які можна відвідувати виходячи із заздалегідь визначеного розкладу фітнес-занять.

Таблиця 2.13 Розклад груп занять

Найменування таблиці	Ім'я поля	Тип даних	Опис
FitnessGroup	Kod	Лічильник	Ідентифікатор
	FitnessZal	Числовий	Код виду
	TimeP	Дата/Час	Час початку
	TimeK	Дата/Час	Час закінчення
	PN	Логічний	Понеділок
	VT...PT	Логічний	Група полів: Вівторок-П'ятниця
	ND	Логічний	Неділя

	Kod	FitnessZal	TimeP	TimeK	PN	VT	SR	CT	PT	SB	ND
▶ +	1	Аеробіка	10:00	12:00	☒	☐	☒	☐	☒	☐	☐
+	2	Аеробіка	13:00	15:00	☒	☐	☒	☐	☒	☐	☐
+	3	Аеробіка	16:00	18:00	☒	☐	☒	☐	☒	☐	☐
+	4	Аеробіка	10:00	12:00	☐	☒	☐	☒	☐	☒	☐
+	5	Аеробіка	13:00	15:00	☐	☒	☐	☒	☐	☒	☐
+	6	Аеробіка	16:00	18:00	☐	☒	☐	☒	☐	☒	☐
+	7	Тренажери	09:00	11:00	☒	☐	☒	☐	☒	☐	☒
+	8	Тренажери	12:00	14:00	☒	☐	☒	☐	☒	☐	☒
+	9	Тренажери	15:00	17:00	☒	☐	☒	☐	☒	☐	☒
+	10	Бойові мистецтва	10:00	13:00	☒	☐	☒	☐	☒	☐	☐
+	11	Бойові мистецтва	15:00	18:00	☒	☐	☒	☐	☒	☐	☐
+	12	Бойові мистецтва	10:00	18:00	☐	☒	☐	☒	☐	☒	☐
+	13	Бойові мистецтва	15:00	18:00	☐	☒	☐	☒	☐	☒	☐
+	14	Атлетика	11:00	13:00	☒	☐	☒	☐	☒	☐	☐
+	15	Атлетика	15:00	17:00	☒	☐	☒	☐	☒	☐	☐
+	16	Атлетика	11:00	13:00	☐	☒	☐	☒	☐	☒	☐
+	17	Атлетика	15:00	17:00	☐	☒	☐	☒	☐	☒	☐
+	18	Басейн	09:00	13:00	☒	☐	☒	☐	☒	☐	☒
+	19	Басейн	14:00	18:00	☐	☒	☐	☒	☐	☒	☒
+	20	Сауна	10:00	14:00	☒	☐	☒	☐	☒	☐	☐
+	21	Сауна	16:00	20:00	☐	☒	☐	☒	☐	☒	☐
*					☐	☐	☒	☐	☐	☐	☐

Запись: 1 из 21

Рисунок 2.18 Розклад груп занять

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Для відображення часу початку та закінчення занять використані поля дати/часу з форматом «Короткий формат часу».

Далі, передбачено ряд із 7 логічних полів у вигляді «Понеділок»...«Неділя», в яких можна, вказуючи «справжні» значення, формувати розклад занять по днях тижня з метою отримання подальшої інформації.

На підставі даних таблиць про види фітнесу та зареєстрованих постояльців створюється таблиця про конкретні факти надання послуг з організації здорового дозвілля постояльців.

Таблиця 2.14 Заняття фітнесом

Найменування таблиці	Ім'я поля	Тип даних	Опис
FitnessGroup	Kod	Лічильник	Ідентифікатор
	Hotel	Числовий	Код готелю
	Client	Числовий	Код постояльца
	FitGroup	Числовий	Код групи
	Mesta	Числовий	Число відвідувачів
	Splata	Грошовий	Сплачена сума

	Kod	Hotel	Client	FitGroup	Mesta	Splata
▶	1	Готель №1	Іванов І.І.	Аеробіка	2	200.00грн
	2	Готель №1	Іванов І.І.	Аеробіка	1	400.00грн
	3	Готель №1	Іванов І.І.	Тренажери	2	450.00грн
*			0	0	0	0.00грн.

Запис: 1 из 3

Рисунок 2.19 Заняття Фітнесом

У даній таблиці передбачено ключове поле Kod, використовуючи яке далі база даних і програмна оболонка будуть отримувати пов'язані відносини з таблицею.

Далі в таблиці є пов'язане поле для отримання текстових значень найменування готелю поточного номера за кодом з таблиці «Об'єкти Мережі Готель».

Також у таблиці є пов'язане поле для отримання текстових значень Прізвища І.Б. постояльца за кодом із таблиці «Реєстрація Постояльців». В даному випадку для формування текстового відображення фахівця у форматі «Прізвище І.Б.»

застосовано вираз із використанням текстових функцій бази даних: `SELECT RegisterClients.Kod, RegisterClients.Prizvyshche+" "+Left(RegisterClients.Imya,1)+"."+Left(RegisterClients.PoBatkovi,1)+"." AS Name FROM RegisterClients;`

Далі в таблиці є пов'язане поле для отримання текстових значень найменування та грошових значень виду поточної послуги з фітнесу за кодом з таблиці «Види фітнесу».

Так як у готелі оформлений один відповідальний постоялець, то на його ім'я вказується числова кількість місць для заняття фітнесом.

Для оформлення фактичної оплати передбачено спеціальне поле грошового формату, в якому реєструються суми, сплачені постояльцем до повного розрахунку при виселенні.

У цілому нині інформаційно-логічна модель бази даних «Мережа Готель» має такий вид (рис. 2.20):

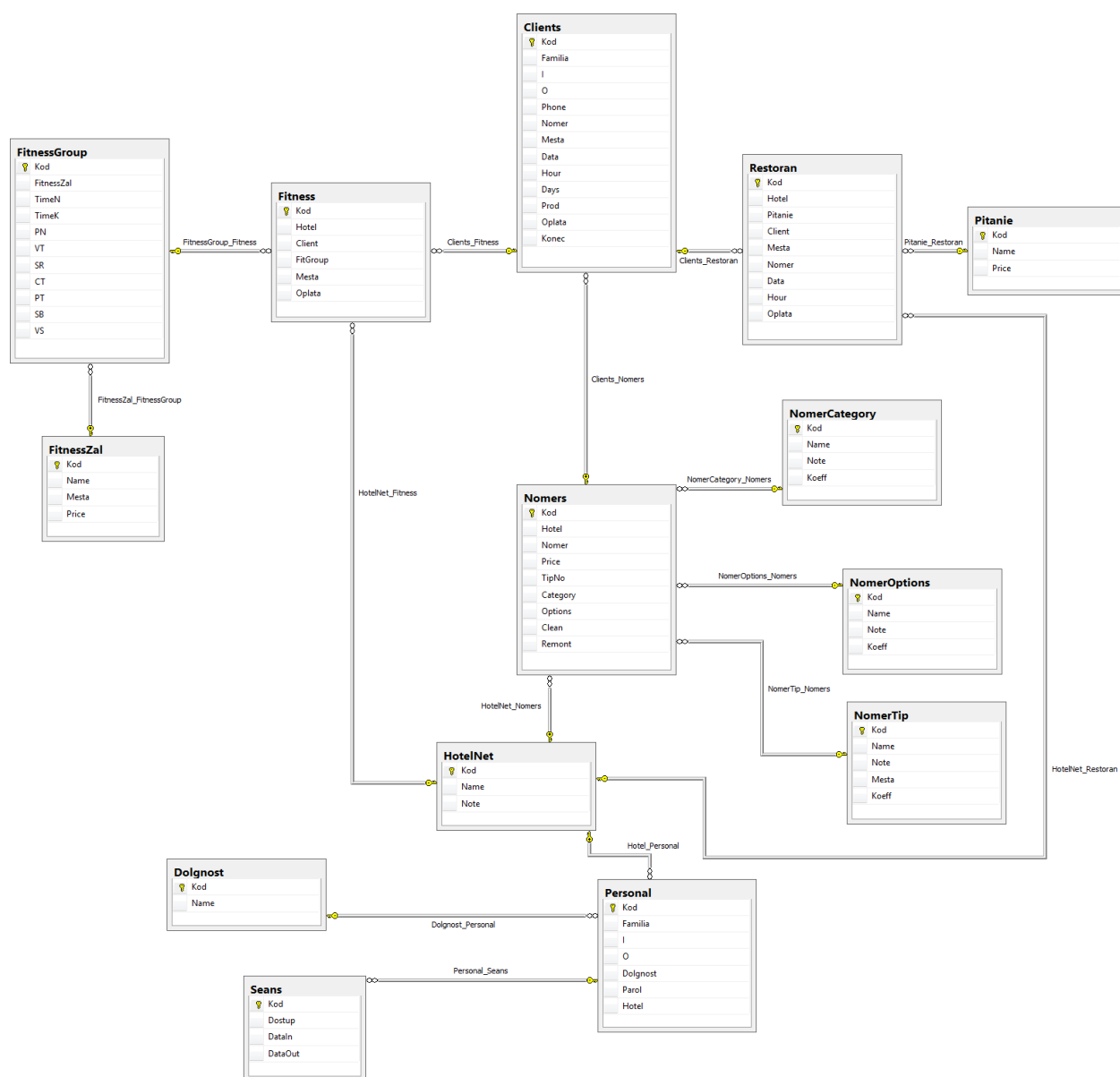


Рисунок 2.20 Діаграма бази даних "Мережа Готелів"

Таким чином, необхідну базу даних було створено. Інші об'єкти обробки бази даних будуть створені в рамках програмного додатку.

2.3 Проектування структури інформаційної системи

Основним елементом у сучасних інформаційних системах є автоматизоване робоче місце (АРМ) – проблемно-орієнтований програмно-технічний комплекс, винесений на робоче місце та який автоматизує в режимі діалогу деякий набір управлінських процедур кінцевого користувача за його безпосередньої участі.

Для здійснення програмної реалізації проекту у вигляді настільного Windows-програми .NET було виділено та визначено інтерфейсні частини програми, як окремі інформаційні об'єкти.

В цілому, виділяється основний інтерфейсний об'єкт «Інформаційна система Мережі Сімейних Готелів» - Windows-додаток з управління базою даних SQL Server.

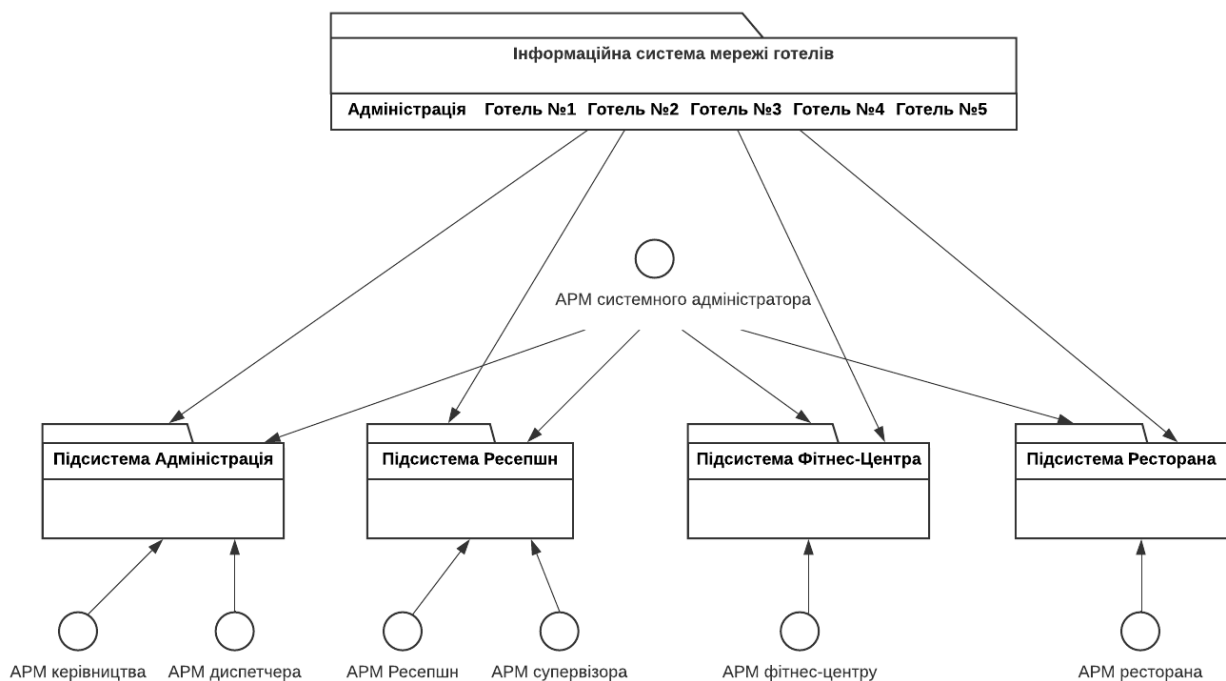


Рисунок 2.21 Діаграма класів – структура інформаційної системи

Основним інтерфейсним об'єктом інформаційної системи є «АРМ Адміністратора»

Виходячи з наведеної (див. рис. 2.21) структури інформаційної системи, вона складається з:

1. Глобальний Інформаційний об'єкт – «Інформаційна мережа мережі сімейних готелів», до нього входять інформаційні сутності:

- Адміністрація;
- Готель №1;
- Готель №2;
- Готель №3;
- Готель №4;
- ...
- Готель №N;

2. Інформаційний об'єкт - "Підсистема Адміністрація":

- Інтерфейс користувача - "АРМ керівництва";
- Інтерфейс користувача - "АРМ диспетчера";
- Редагується АРМ системного адміністратора.

3. Інформаційний об'єкт - "Підсистема Ресепшн":

- Інтерфейс користувача - "АРМ ресепшн";
- Інтерфейс користувача - "АРМ супервізора";
- редагується АРМ системного адміністратора;

4. Інформаційний об'єкт – «Підсистема Фітнес-Центру»:

- Інтерфейс користувача - "АРМ Фітнес-Центру";
- Редагується АРМ системного адміністратора.

5. Інформаційний об'єкт – «Підсистема Ресторану»

- Інтерфейс користувача - "АРМ Ресторана";

- Редагується АРМ системного адміністратора.

Крім того, у проекті передбачена база даних Hotels.mdf - база даних SQL Server по мережі сімейних готелів.

З РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Відповідно до системного підходу проєктована інформаційна система є сукупністю взаємопов'язаних об'єктів, що функціонують спільно для досягнення мети.

3.1 Засоби реалізації

У рамках практичної частини цієї роботи виконано створення ряду програмних додатків засобами середовища програмування Visual Studio, що демонструють можливості ЕОМ щодо виконання завдань роботи з базами даних з використанням мови програмування C#.

Доступ до бази даних у програмній оболонці передбачається здійснити за допомогою використання описаного вище SQL-інтерфейсу у зв'язку з технологіями програмування .NET. Вибір даної технології обумовлений опрацьованістю та стабільністю зв'язку БД<->додаток, які підтримуються сучасними Windows-системами. Носієм для серверної бази даних вибрано стандартну базу даних у форматі Microsoft SQL Server 2008 Express (*.mdf) даних.

Всі алгоритмічні дії за програмою базуються на основному циклі роботи (очікування - Application.Run) програми в середовищі Windows, переривання якого виконує поставлені завдання.

Запуск підпрограм з виконання завдань ґрунтується на властивостях та методах стандартних елементів Windows, доступ до яких організований та передбачений за допомогою елементів «Палітри інструментів» середовища програмування, а саме:

- Доступ до таблиць даних у форматі SQL Express здійснюється за допомогою стандартних невізуальних компонентів DataSet, керуючи властивостями та методами якого можна через середовище програмування безпосередньо проводити зміну та вибірку фізичної таблиці даних;
- Відображення даних у табличному вигляді здійснюється компонентами DataGridView та стандартними елементами. Цей елемент також дозволяє редагувати дані. Для можливості пересування таблицям даних застосований також підкомпонент Pager, який забезпечує посторінкову навігацію по таблиці;
- Для здійснення вибору даних із визначених списків застосовані компоненти ListBox та DropDownList, які на етапі розробки програми прив'язані до таблиць бази даних з метою вибору різних значень зі списків;
- Для здійснення всіх корисних дій застосовані кнопки Button, пов'язані із програмним кодом;

І, нарешті, застосований головний компонент - MainForm, який, за допомогою вказівки якості IsMdiContainer=true; є головним вікном-контейнером програми, що дозволяє програмній оболонці завантажувати в основне вікно ресурси інших віконних форм, фактично використовуючи в рамках програми схожу клієнтську HTML-технологію <FRAME>.

3.2 Загальна структура програмного застосування системи

Таким чином, структура програми є набір з декількох діалогових вікон, активація яких дозволяє отримати доступ до елементів бази даних. Крім того, у проекті передбачена база даних у форматі Microsoft SQL Server Express – «Hotel.mdf».

Додаток має різноманітні візуальні та невізуальні елементи, що дозволяють здійснити відповідний доступ до ресурсів системи.

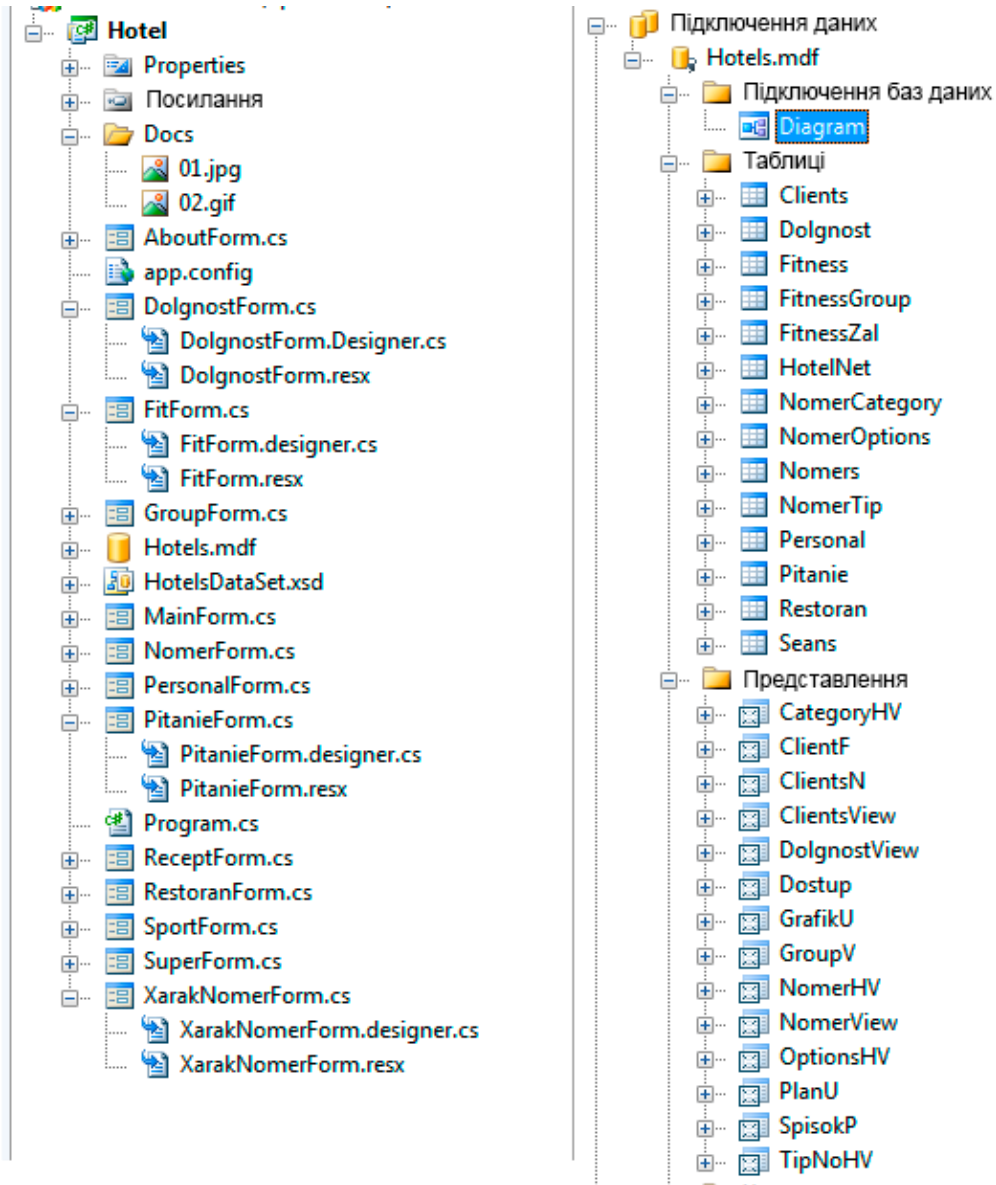


Рисунок 3.1 Структура проекту Програми

3.3 Посібник користувача

При запуску програми відбувається вхід до інформаційної системи – на екран виводиться вікно авторизації користувачів (рис. 3.2):

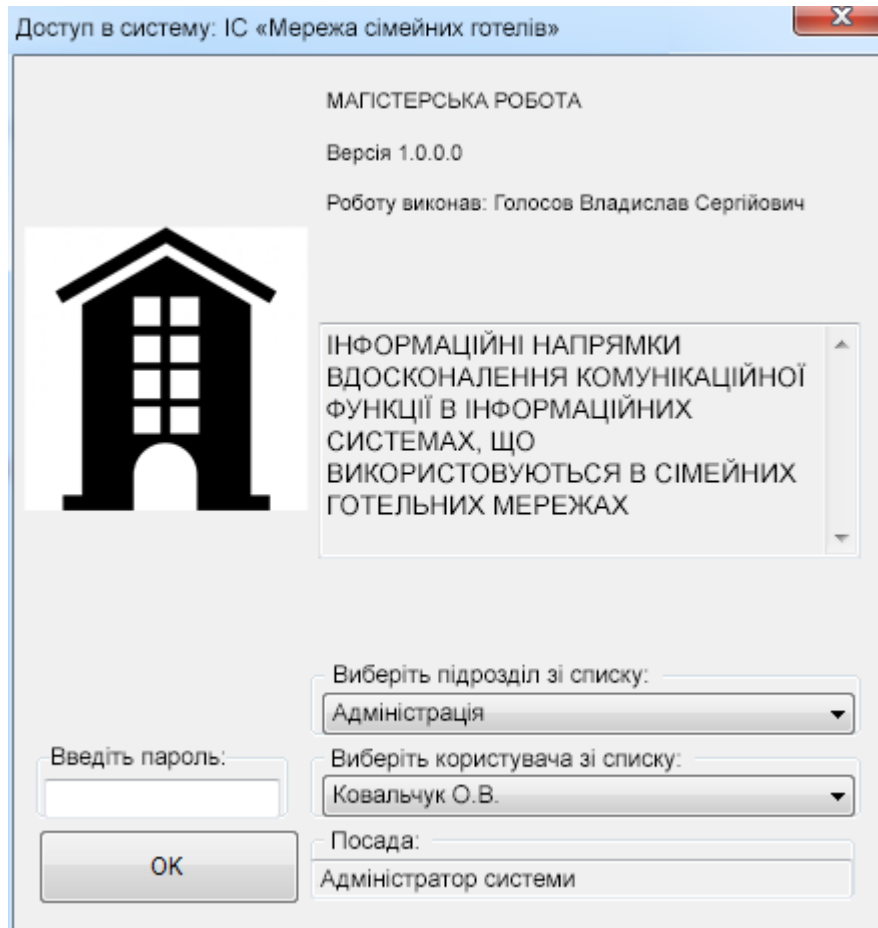


Рисунок 3.2 Вікно авторизації користувачів

У рамках програмних ресурсів даного діалогового вікна передбачена система перевірки здійснення доступу, для чого на формі передбачено три об'єкти: два списки, що випадають, і поле введення. При виборі зі списку користувача системи він вводить пароль і при правильному введенні система продовжує роботу.

Для організації перевірки пароля – вікно "Доступ в систему" виводиться раніше запуску головної контейнерної MDI форми програми, що прописано в стартовому програмному файлі проекту "Program.cs":

При отриманому дозволі відбувається вхід до інформаційної системи - на екран виводиться головна екранна форма (рис. 3.3):

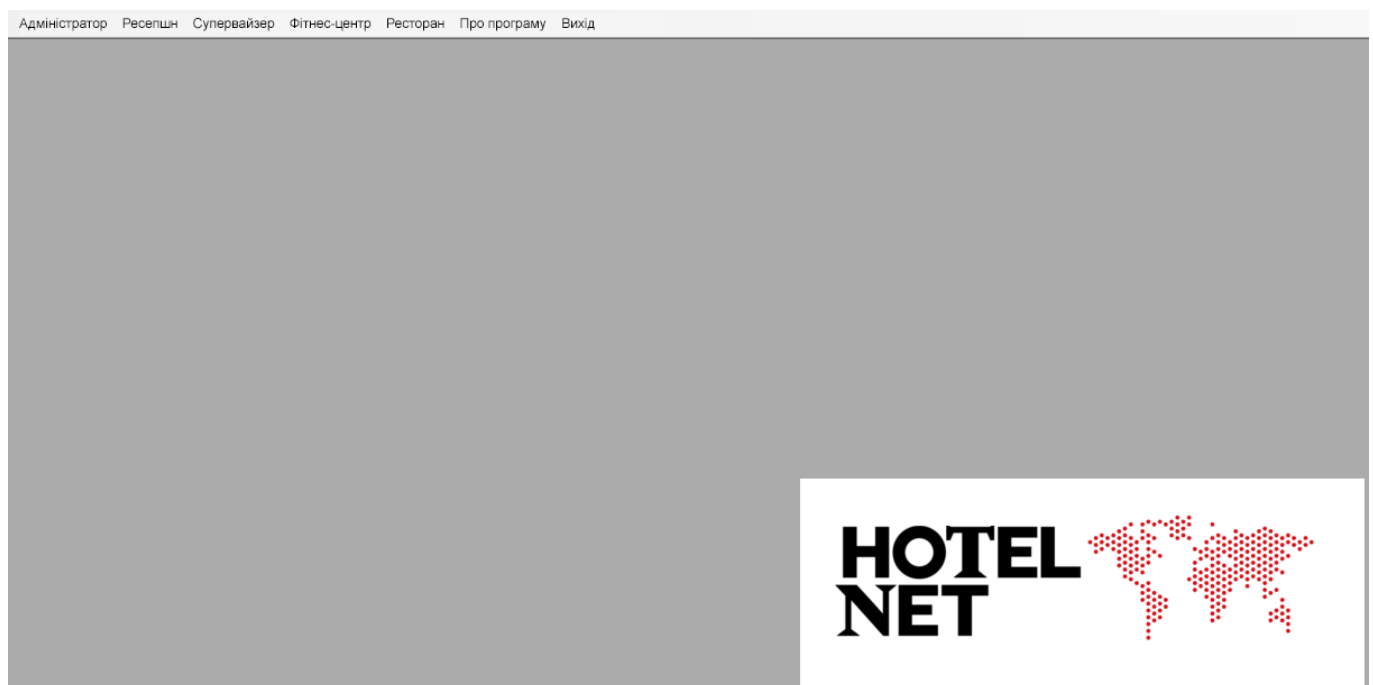


Рисунок 3.3 Головне вікно програмної оболонки

Головна екранна форма забезпечена тематичним малюнком, що містить всі необхідні елементи, які дають можливість повноцінної обробки інформації у вигляді головного меню.

Структура головного меню програмної програми:

1. Адміністратор (робота з довідниками структури):

- а. Персонал (довідник працівників);
- б. Посади (довідник посад);

- в. Характеристики готельних номерів (довідники основних та додаткових параметрів проживання);
 - Типи номерів;
 - Категорії номерів;
 - Додаткові опції;
- г. Готельні номери (зведений довідник готельних номерів у розрізі об'єктів готельного господарства мережі);
- д. Об'єкти мережі сімейних готелів (список об'єктів готельного господарства);
- 2. «Ресепшн» (АРМ менеджера-адміністратора прийому гостей);
- 3. «Супервайзер» (АРМ менеджера з підготовки та прибирання номерів номерів);
- 4. Фітнес-Центр (об'єкти оздоровчих занять клієнтів мережі сімейних готелів):
 - а. Види оздоровчих занять
 - б. Групи та графіки занять
 - в. Запис на заняття (реєстрація клієнтів на заняття)
- 5. Ресторан (об'єкти організації харчування клієнтів):
 - а. Види харчування (сніданки, обіди, ціни);
 - б. Замовлення харчування (реєстрація замовлень харчування, включаючи доставку в номер);
- 6. Про Програму (загальна інформація)
- 7. Вихід (завершення роботи програми)

В рамках програми створено ряд однотипних вікон для редагування простих довідників, прив'язаних до відповідних наборів даних.

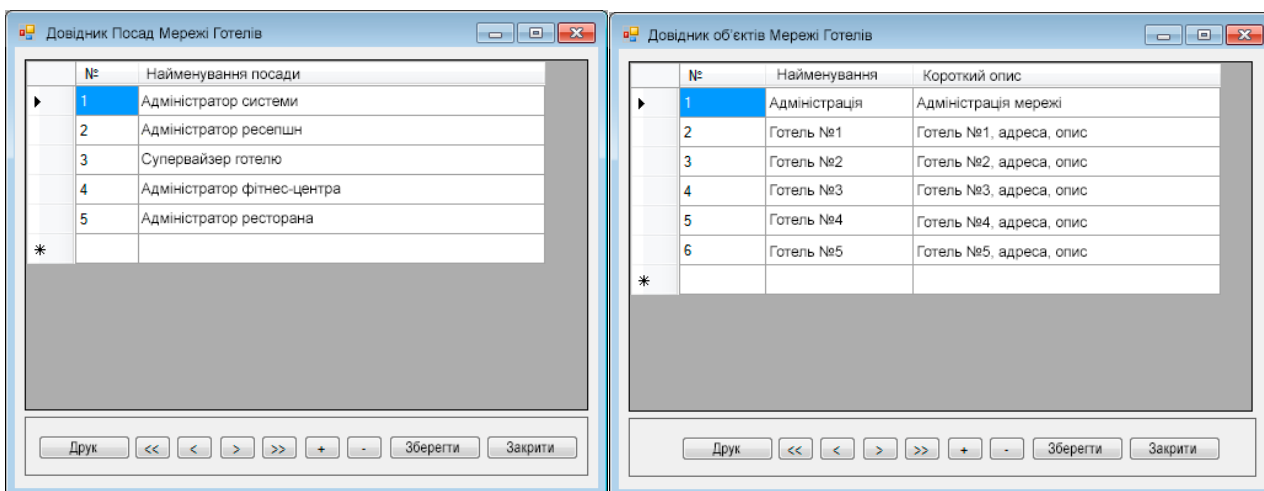


Рисунок 3.4 Довідники посад та готельних об'єктів

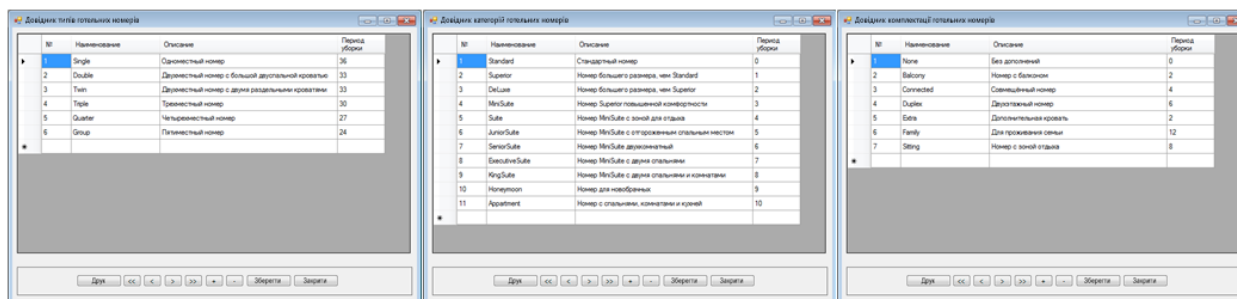


Рисунок 3.5 Довідники характеристик номерів

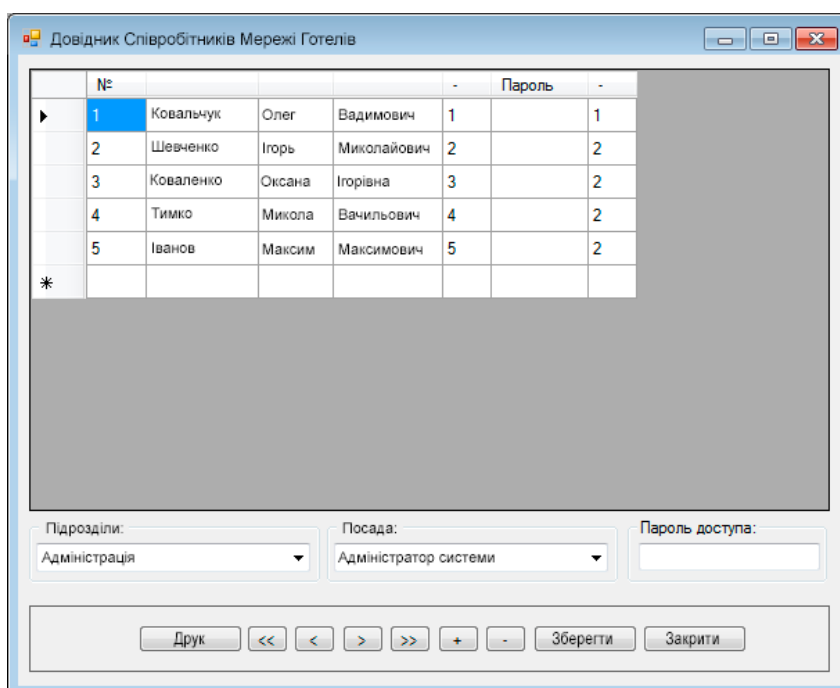


Рисунок 3.6 Довідник працівників

3.3.1 Системні вимоги програмних та апаратних засобів

З огляду на те, що програмний додаток виконано серед програмування Visual Studio, то зазначені для середовища системні вимоги поширюються і створений додаток.

1. Вимоги до програмного забезпечення:

- Програмну програму можна встановити в наступних операційних системах:
- Windows XP (x86) з пакетом оновлень 3 (SP3) - всі випуски крім Starter;
- Windows Vista (x86 і x64) з пакетом оновлень 1 (SP1) - всі випуски крім Starter;
- Windows 7 (x86 та x64);
- Windows Server 2003 (x86 та x64) з пакетом оновлень 2 (SP2)
- Windows Server 2003 R2 (x86 та x64)
- Windows Server 2008 (x86 та x64) з пакетом оновлень 2 (SP2);
- Windows Server 2008 R2 (x64).
- Windows 8 (x86 та x64);
- Windows 10 (x86 та x64).

Підтримувані архітектури:

- 32-розрядна (x86);
- 64-розрядна (x64);

2. Вимоги до обладнання:

- Процесор із частотою 1,6 ГГц або вище;
- 1024 МБ ОЗП;
- 3 ГБ вільного місця на диску;
- Жорсткий диск зі швидкістю 5400 об/хв;
- Відеоадаптер з підтримкою DirectX 9 та роздільною здатністю 1280 x 1024 (або вищою);

- Дискковод DVD-ROM або накопичувач флеш-дисків.

3.3.2 Інсталяція системи

У проекті програми для зв'язку з базами даних було застосовано невізуальні елементи Dataset, BindingSource і TableAdapter, котрим система створила набір табличних елементів.

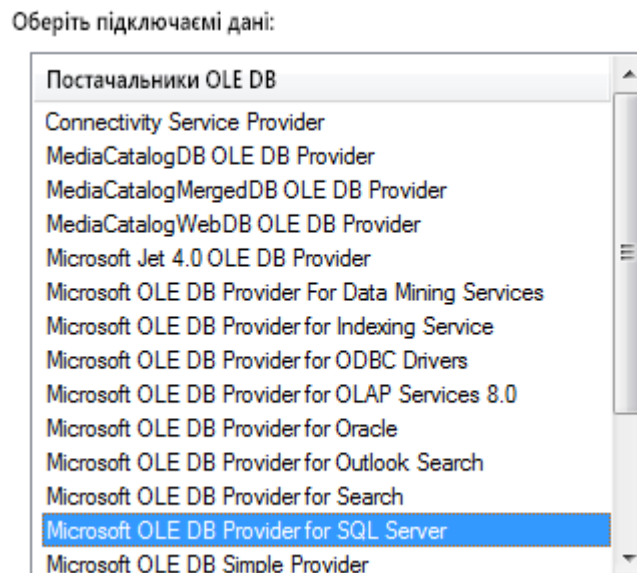


Рисунок 3.7 Вибір постачальника даних - Microsoft OLE DB Provider for SQL Server

Для прив'язки додатку до бази даних був застосований невізуальний компонент SqlConnection, який налаштований таким чином:

1. Було встановлено та налаштовано сервер баз даних Microsoft SQL Server 2008 Express (компактна версія, застосовується на етапі навчального програмування).

2. Засобами утиліти "Імпорт та експорт даних (32-розрядна версія)" була створена база даних "Rabota.mdf", яка була розміщена в підкаталозі каталогу розміщення програми.

3. Налаштовано рядок з'єднання з базою даних.

4. Вибрано постачальника бази даних Microsoft OLE DB Provider for SQL Server, який відповідає серверу баз даних.

5. Далі були налаштовані канали передачі:

- Вказано ім'я сервера – вказується ім'я сервера, що працює у поточній мережі та на локальному комп'ютері;
- Вказано шлях до робочої бази даних.

Після цього було здійснено перевірку з'єднання.

Для адаптації програмного застосування в існуючих архітектурах було здійснено налаштування робочої платформи програми:

Для всіх таблиць і запитів у програмі передбачено універсальний альянс операційної системи «~», що означає, що таблиці даних знаходяться в тій же папці, що і файл програми, що виконується. Це забезпечує запуск Програми на будь-якому комп'ютері.

3.3.3 Результати тестування з прикладами екранних форм

У рамках роботи з програмним додатком було побудовано та протестовано екранні форми, в яких виконуються основні виробничі дії відповідальних співробітників мережі сімейних готелів.

	Код	Прізвище	Ім'я	По батькові	Телефон	Номер	Міся	Дата заїзду	Час виїзду	Днів	Подовження
▶	1	Іванов	Іван	Іванович	0500976845	1	1	15.12.2022	11:28	3	0
	2	Петров	Петро	Петрович	0965463415	4	2	16.12.2022	20:20	2	0
	3	Миколаїв	Микола	Ігорович	0973523415	6	2	17.12.2022	15:15	8	1
	4	Коваленко	Сергій	Сергійович	0505482573	5	1	18.12.2022	9:00	4	0
*											

Рисунок 3.8 Реєстрація та облік проживання клієнтів мережі сімейних готелів

Для роботи з наборами даних на кожній формі впроваджено ряд командних кнопок, що виконують стандартні операції «Друк», «Зберегти», «Закрити» та «Навігація».

Для друку даних в рамках програми застосована можливість створювати вихідний документ на основі шаблону документа Microsoft Excel – до осередків якого здійснюється доступ та зміна за допомогою ресурсів основної програми - бібліотеки Microsoft.Office.Interop.Excel:

```
// Визначаємо ресурси для роботи з Microsoft.Office.Interop.Excel
private Microsoft.Office.Interop.Excel.Application app = null;
//Підключаємося до програми Excel
private Microsoft.Office.Interop.Excel.Workbook workbook = null;
//Підключаємося до робочої книги Excel
private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;
string Sp=System.Windows.Forms.Application.StartupPath+ "/personal.xls";
//Підключаємося до робочих листів Excel
```

Далі робочі листи Excel заповнюються звичайним чином, з використанням адресації посилань типу «R1:C1»

Для побудови звітів було передбачено спеціально сконструйовані SQL-запити.

Реєстр
клієнтів мережі

№	Прізвище	Ім'я	По батькові	Телефон	Розміщення у номері	Число койок-ліжок	Дата заїзду	Час заїзду	Число днів	Подовження днів	Сплачено	Готель
1	Іванов	Іван	Іванович	0500976845	1	1	15.12.2022	11:28	3	0	1 500.00грн.	Готель №1
2	Петров	Петро	Петрович	0965463415	4	2	16.12.2022	20:20	2	0	2 500.00грн.	Готель №1
3	Михайлів	Микола	Ігорович	0973523415	6	2	17.12.2022	15:15	8	1	1 200.00грн.	Готель №1
4	Коваленко	Сергій	Сергійович	0505482573	5	1	18.12.2022	9:00	4	0	0.00грн.	Готель №1

Рисунок 3.9 Отримання реєстру-відомості проживання клієнтів

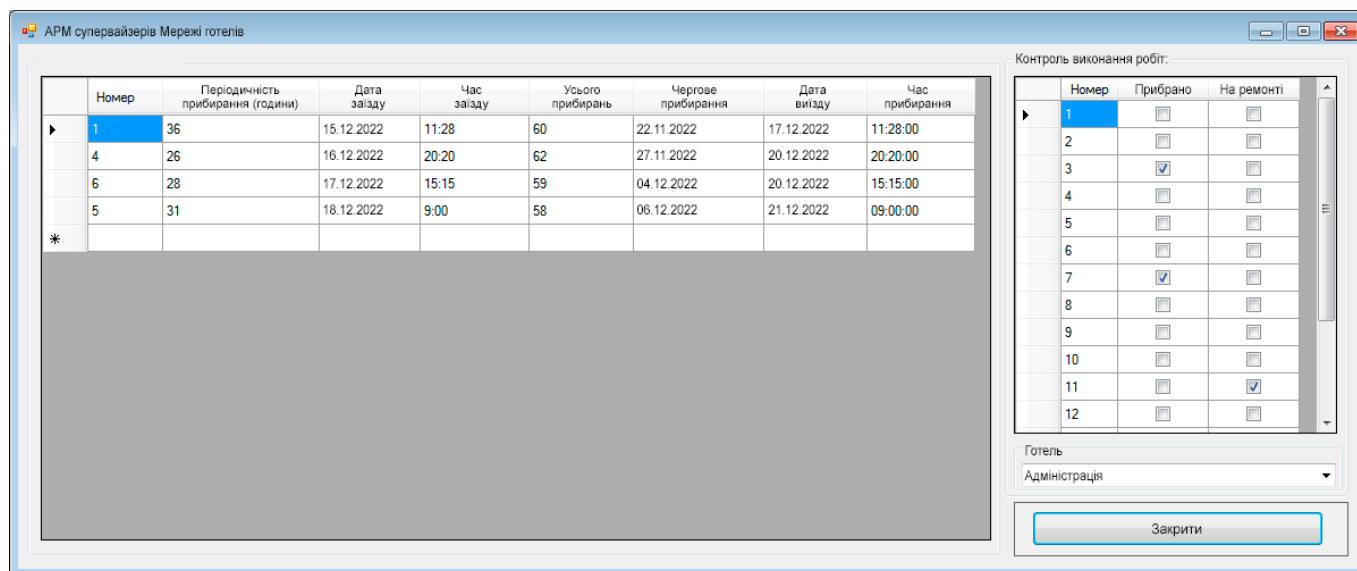


Рисунок 3.10 АРМ Супервайзер мережі сімейних готелів

При роботі супервайзера мережі сімейних готелів він за допомогою ряду каскадних питань отримує вичерпний список поточного стану готельних номерів, маючи можливість спрямовувати сили підлеглих як на поточне (планове) прибирання номерів, так і на підготовку номерів для подальшого використання, у тому числі до ремонту.

Для побудови організованого розпорядку оздоровчих занять для клієнтів побудовано спеціальну форму, що дозволяє налаштовувати такий графік:

№	Початок	Кінець	ПН	ВТ	СР	ЧТ	ПТ	СБ	НД
1	10:00	12:00	☒	☐	☒	☐	☒	☐	☐
2	13:00	15:00	☒	☐	☒	☐	☒	☐	☐
3	16:00	18:00	☒	☐	☒	☐	☒	☐	☐
4	10:00	12:00	☐	☒	☐	☒	☐	☒	☐
5	13:00	15:00	☐	☒	☐	☒	☐	☒	☐
6	16:00	18:00	☐	☒	☐	☒	☐	☒	☐
7	9:00	11:00	☒	☐	☒	☐	☒	☐	☒
8	12:00	14:00	☒	☐	☒	☐	☒	☐	☒
9	15:00	17:00	☒	☐	☒	☐	☒	☐	☒
10	10:00	13:00	☒	☐	☒	☐	☒	☐	☐
11	15:00	18:00	☒	☐	☒	☐	☒	☐	☐
12	10:00	18:00	☐	☒	☐	☒	☐	☒	☐
13	15:00	18:00	☐	☒	☐	☒	☐	☒	☐
14	11:00	13:00	☒	☐	☒	☐	☒	☐	☐
15	15:00	17:00	☒	☐	☒	☐	☒	☐	☐
16	11:00	13:00	☐	☒	☐	☒	☐	☒	☐

Об'єкт фітнес центру
Аеробіка

Друк << < > >> + - Зберегти Закрити

Рисунок 3.11 Побудова графіка оздоровчих занять

Як видно з цієї форми, налаштування графіка проводиться у розрізі видів заняття фітнесом (оздоровчими заходами), при цьому з використанням елементів – «прапорців» користувач легко налаштовує дні занять груп кожного дня тижня.

З даних зв'язної таблиці будується спеціальний наочний звіт, який практично може бути вивішений на загальний огляд.

Реєстр
клієнтів мережі

№	Вид заняття	Вмістимість (чол)	Вартість одної години	Година початку	Година закінчення	Пн	Вт	Ср	Чт	Пт	Сб	Нд
1	Аеробіка	20	150.0000	10.00	12.00							
2	Аеробіка	20	150.0000	13.00	15.00							
3	Аеробіка	20	150.0000	16.00	18.00							
4	Аеробіка	20	150.0000	10.00	12.00							
5	Аеробіка	20	150.0000	13.00	15.00							
6	Аеробіка	20	150.0000	16.00	18.00							
7	Тренажери	20	250.0000	09.00	11.00							
8	Тренажери	20	250.0000	12.00	14.00							
9	Тренажери	20	250.0000	15.00	17.00							
10	Бойові мистецтва	30	120.0000	10.00	13.00							
11	Бойові мистецтва	30	120.0000	15.00	18.00							
12	Бойові мистецтва	30	120.0000	10.00	18.00							
13	Бойові мистецтва	30	120.0000	15.00	18.00							
14	Атлетика	15	145.0000	11.00	13.00							
15	Атлетика	15	145.0000	15.00	17.00							
16	Атлетика	15	145.0000	11.00	13.00							
17	Атлетика	15	145.0000	15.00	17.00							
18	Басейн	20	100.0000	09.00	13.00							
19	Басейн	20	100.0000	14.00	18.00							
20	Сауна	10	200.0000	10.00	14.00							
21	Сауна	10	200.0000	16.00	20.00							

Рисунок 3.12 Звітна форма "Графік занять фітнес-центрів"

Для реєстрації клієнтів передбачено форму, в якій при реєстрації динамічно обчислюється вартість оплати в залежності від виду оздоровчого заняття та кількості відвідувачів по кожному клієнту.

	№	Місця	Сплачено
▶	1	2	200,00грн.
	2	1	400,00грн.
	3	2	450,00грн.
*			

Заняття в готелі:
Готель №1

Група заняття:
Аеробіка

Початок: 10:00 Кінець: 12:00 Ціна: 150,00грн.

Клієнт:
Іванов І.І.

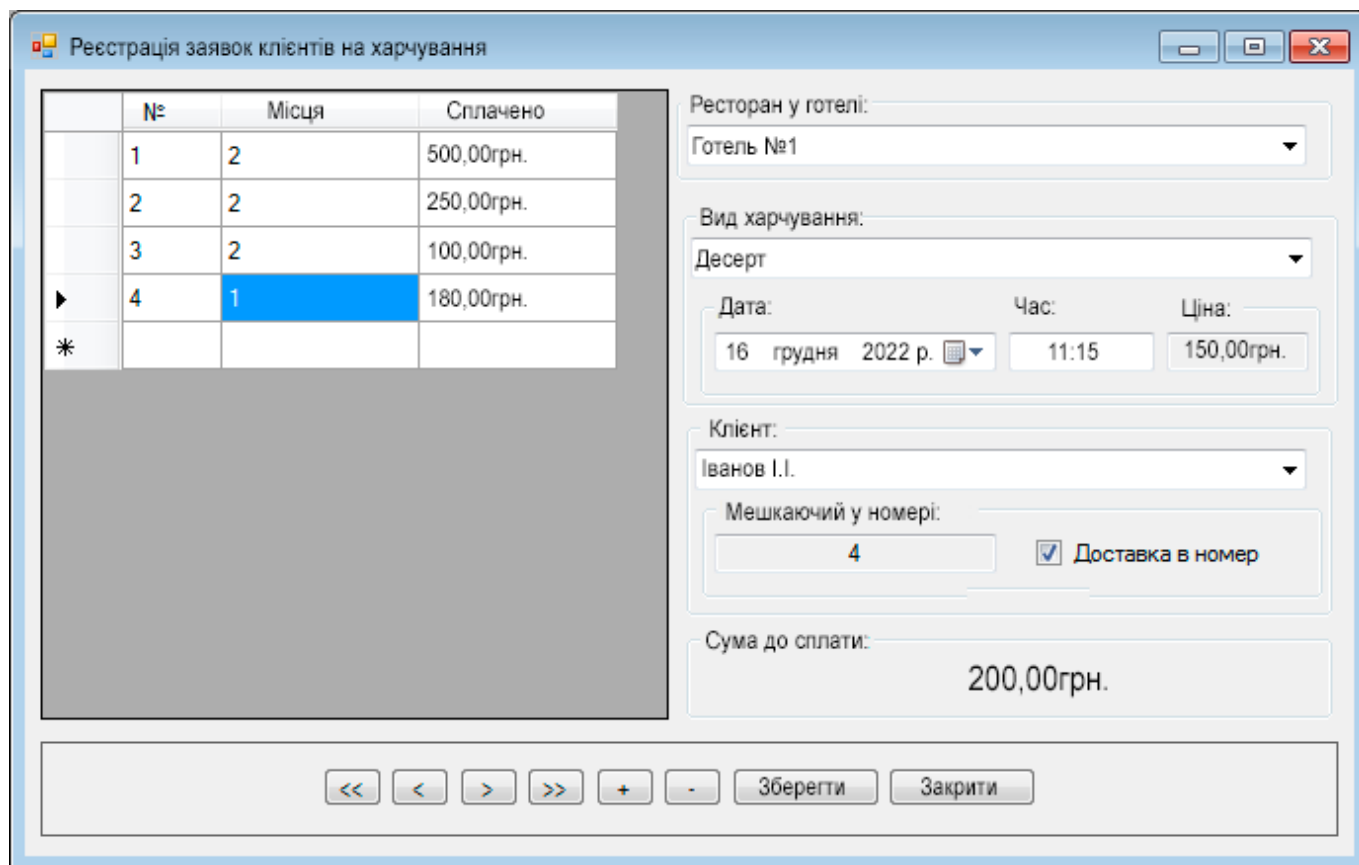
Мешкаючий у номері:
1

Сума до сплати:
200,00грн.

Navigation: << < > >> + - Зберегти Закрити

Рисунок 3.13 Реєстрація клієнтів фітнес-центру

Аналогічним чином реєструються та розраховуються замовлення клієнтів на організацію харчування:



Реєстрація заявок клієнтів на харчування

	№	Місця	Сплачено
	1	2	500,00грн.
	2	2	250,00грн.
	3	2	100,00грн.
▶	4	1	180,00грн.
*			

Ресторан у готелі:
Готель №1

Вид харчування:
Десерт

Дата: 16 грудня 2022 р. Час: 11:15 Ціна: 150,00грн.

Клієнт:
Іванов І.І.

Мешкаючий у номері:
4 ☒ Доставка в номер

Сума до сплати:
200,00грн.

<< < > >> + - Зберегти Закрити

Рисунок 3.14 Реєстрація заявок клієнтів на харчування

Таким чином, у процесі виконання роботи була створена система, яка, за своїми можливостями, повністю виконує всі вимоги, запропоновані завданням.

ВИСНОВКИ

Метою написання цієї випускної кваліфікаційної роботи стала побудова прикладу інформаційної системи мережі сімейних готелів.

Предметом дослідження було завдання створення прикладу інформаційної системи для автоматизації системи документообігу підприємства сфери обслуговування населення.

Об'єктом дослідження була структура ділових процесів готельного закладу з погляду оперативного обліку. Актуальність побудови цієї системи обумовлена стабільним зростанням споживчого попиту до питань якісного супроводу документообігу в готелях, що супроводжується зростанням вимог до якості та видів обробки інформації.

Для досягнення мети роботи було поставлено такі завдання:

- Розробити та налаштувати модель бази даних установи.
- Розробити настільний Windows-додаток, що забезпечує створення та модифікацію даних адміністратором бази даних.
- Розробити екранні інтерфейсні елементи, які забезпечують створення та модифікацію даних на рівні робочих місць відповідальних співробітників мережі готельних установ.
- Розробити екранні інтерфейсні елементи, які забезпечують пошук, перегляд та використання даних на рівні робочих місць відповідальних працівників мережі готельних закладів.

Структура випускної кваліфікаційної роботи така: вступ, основна частина з трьох розділів, висновки, список використаної літератури, додатки.

У теоретичній частині роботи були проведені дослідження, які включають результати аналізу та перед проектного обстеження факторів управління

виробничими процесами. В результаті проведеного перед проектного аналізу було конкретизовано завдання, сформульовано послідовність досягнення цілей.

У практичній частині роботи було визначено інформаційну сутність завдання, створено базу даних, передбачено кілька пов'язаних по посиланням полях кодів відносин таблиць та полів. Усі таблиці, пов'язані одна з одною, у зв'язків передбачена перевірка цілісності і можливість каскадного видалення даних. Для наочного візуального відображення текстових довідкових значень з таблиць джерел у процесі розробки програмного додатку було створено низку запитів – уявлень даних.

Таким чином, у процесі виконання роботи була створена система, яка, за своїми можливостями, повністю виконує всі вимоги, запропоновані завданням.

ПЕРЕЛІК ПОСИЛАНЬ

1. Andrew W. Troelsen Pro C# 7: With .NET and .NET Core - 2017. - 1330 p.
2. Адміністрування Microsoft SQL Server 2012. MCAD/MCSE, MCDBA, 2018 – 640 с.
3. Пилипівський Є.Є. Економіка та організація готельного господарства, 2013. – 176 с.
4. Юрченко І.В. Інформатика та програмування. Частина 1. Навчальний посібник.– Чернівці: Книги–XXI, 2011.– 203 с.
5. Юрченко І.В., Сікора В.С. Інформатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.
6. Семчук А.Р., Юрченко І.В. Економічна інформатика. Навчальний посібник.– Чернівці: МВІЦ "Місто", 2008.– 426 с.
7. Миронов Ю.Б. Інформаційні технології в діяльності санаторно-готельних підприємств // Вісник Львівської комерційної академії. Серія економічна. 2007. Вип.26. С.193-200.
8. Школа А.М. Менеджмент туристичної індустрії: навчальний посібник / За ред. проф. І.М. Школи. Чернівці: ЧТЕІ КНТЕУ, 2003. 662 с.
9. Сокол т.г. Організація обслуговування в готелях і туристичних комплексах Підручник. – К.: Альтерпрес, 2009. – 447 с.
10. Богатирьов, В. А. Оптимізація періодичності контролю захищеності комп'ютерних систем / В. А. Богатирьов, А. В. Богатирьов // Науково-технічний вісник інформаційних технологій, механіки і оптики. - 2015. - № 2. - С. 300- 305.
11. Гужва В. М. Інформаційні системи в міжнародному бізнесі / В. М. Гужва, А. Г. Постовой. — К.: КНЕУ, 1999. — 164 с.
12. О.М. Роїк, А. А. Шиян, Л. О. Нікіфорова. Системний аналіз. Вінниця. ВНТУ. 2015. –83с.
13. [О.М. Шушура, Р. Н. Кветний., І. В. Богач та ін.] Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 1: навчальний посібник з грифом МОНМС України. - Вінниця: ВНТУ, 2013. - 191.
14. [О.М. Шушура, Р. Н. Кветний., І. В. Богач та ін.] Комп'ютерне моделювання систем та процесів. Методи обчислень. Частина 2: навчальний посібник з грифом МОНМС України. - Вінниця: ВНТУ, 2013. - 235.
15. Joseph O'Connor and Ian McDermott. The art of systems thinking. Thorsons. 2006. 256с.

16. В.Б. Толубко, Бурячок В.Л., Г.М. Гулак. Інформаційний та кіберпростори: проблеми безпеки, методи та засоби боротьби: підручник. – Київ: ТОВ «СІК ГРУП УКРАЇНА». 2015– 449 с.

- 17.Н.Б. Шаховська, В.В. Литвин. Проектування інформаційних систем: навчальний посібник. – Львів: «Магнолія - 2006». 2017 – 380 с. 6
- 18.David A. Marca, Clement L. McGowan Sadt: Structured Analysis and Design Techniques (MCGRAW HILL SOFTWARE ENGINEERING SERIES) – 1987. – 392 p.
- 19.Mike Snell , Lars Powers Microsoft Visual Studio: 2012 Unleashed 2nd Edition – 2012. – 1149 p.
- 20.R. Linger, H. Mills, B. Witt Structured programming - theory and practice – 1979. – 418 p.

ДОДАТОК 1

Лістинг програмного коду с коментарями

1. Program.cs

```

using System;
using System.Collections.Generic;
using System.Windows.Forms;

namespace WindowsApplication1
{
    static class Program
    {
        /// <summary>
        /// Основна точка входу для програми.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            //Спочатку запускаємо форму "Про програму" для доступу до системи
            AboutForm newAboutForm = new AboutForm();
            newAboutForm.ShowDialog();
            //Якщо доступ не дозволено, то завершуємо роботу
            if (newAboutForm.DialogResult == DialogResult.Abort) Application.Exit();
            //Якщо доступ дозволено, то запускаємо головне вікно системи
            else Application.Run(new MainForm());
        }
    }
}

```

2. AboutForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
using System.Reflection;

namespace WindowsApplication1
{
    partial class AboutForm : Form
    {
        public AboutForm()
        {
            InitializeComponent();

            // Ініціалізація AboutBox для відображення інформації про продукт із інформації про збірку.
            // Змініть налаштування інформації про збірку для вашої програми за допомогою:

```

```

// - Проект->Властивості->Програма->Інформація про збірку
// - AssemblyInfo.cs
this.Text = String.Format("Доступ в систему: {0}", AssemblyTitle);
this.labelProductName.Text = AssemblyProduct;
this.labelVersion.Text = String.Format("Версія {0}", AssemblyVersion);
this.labelCopyright.Text = AssemblyCopyright;
this.labelCompanyName.Text = String.Format("Роботу виконав: {0}", AssemblyCompany);
this.textBoxDescription.Text = AssemblyDescription;
}
#endregion Assembly Attribute Accessors

public string AssemblyTitle
{
    get
    {
        // Отримати всі атрибути Title для цієї збірки
        object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyTitleAttribute),
false);
        // Якщо є хоча б один атрибут Title
        if (attributes.Length > 0)
        {
            // Виберіть перший
            AssemblyTitleAttribute titleAttribute = (AssemblyTitleAttribute)attributes[0];
            // Якщо це не порожній рядок, повернути його
            if (titleAttribute.Title != "")
                return titleAttribute.Title;
        }
        // Якщо атрибут Title не було, або якщо атрибут Title був порожнім рядком, поверніть ім'я .exe
        return System.IO.Path.GetFileNameWithoutExtension(Assembly.GetExecutingAssembly().CodeBase);
    }
}

public string AssemblyVersion
{
    get
    {
        return Assembly.GetExecutingAssembly().GetName().Version.ToString();
    }
}

public string AssemblyDescription
{
    get
    {
        // Отримати всі атрибути Description для цієї збірки
        object[] attributes =
Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyDescriptionAttribute), false);
        // Якщо атрибути Description відсутні, поверніть порожній рядок
        if (attributes.Length == 0)
            return "";
        // Якщо є атрибут Description, повернути його значення
        return ((AssemblyDescriptionAttribute)attributes[0]).Description;
    }
}

public string AssemblyProduct

```

```

    {
        get
        {
            // Отримайте всі атрибути продукту на цій збірці
            object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyProductAttribute),
false);
            // Якщо атрибутів продукту немає, поверніть порожній рядок
            if (attributes.Length == 0)
                return "";
            // Якщо є атрибут Product, поверніть його значення
            return ((AssemblyProductAttribute)attributes[0]).Product;
        }
    }

    public string AssemblyCopyright
    {
        get
        {
            // Отримати всі атрибути авторського права на цю збірку
            object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCopyrightAttribute),
false);
            // Якщо атрибути авторського права відсутні, поверніть порожній рядок
            if (attributes.Length == 0)
                return "";
            // Якщо є атрибут авторського права, поверніть його значення
            return ((AssemblyCopyrightAttribute)attributes[0]).Copyright;
        }
    }

    public string AssemblyCompany
    {
        get
        {
            // Отримати всі атрибути компанії на цій збірці
            object[] attributes = Assembly.GetExecutingAssembly().GetCustomAttributes(typeof(AssemblyCompanyAttribute),
false);
            // Якщо атрибутів компанії немає, поверніть порожній рядок
            if (attributes.Length == 0)
                return "";
            // Якщо є атрибут Company, поверніть його значення
            return ((AssemblyCompanyAttribute)attributes[0]).Company;
        }
    }
}
#endregion

private void okButton_Click(object sender, EventArgs e)
{
    this.Close();
}

private void AboutForm_Load(object sender, EventArgs e)
{
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Dostup". За потреби вона може
    бути переміщена або видалена.
    this.dostupTableAdapter.Fill(this.hotelsDataSet.Dostup);
}

```

```

        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.HotelNet". За потреби вона може
        // бути переміщена або видалена.
        this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
        this.dostupBindingSource.Filter = "Hotel = '" + Convert.ToString(this.comboBox1.SelectedValue) + "'";
    }

    private void AboutForm_FormClosing(object sender, FormClosingEventArgs e)
    {
        if (this.comboBox2.SelectedIndex < 0)
        {
            //У списку підрозділу немає співробітників
            e.Cancel = true;
            return;
        }

        //При виході з вікна, якщо введений пароль та користувач збігаються
        if (this.comboBox2.SelectedValue.ToString() == this.maskedTextBox1.Text)
        {
            //Закриваємо вікно для продовження роботи
            this.DialogResult = DialogResult.OK;
            e.Cancel = false;
        }
        else // якщо ж не збігаються
        {
            //Якщо користувач позитивно відповідає на запит про завершення роботи
            if (MessageBox.Show("Завершити роботу програми?", "Невірно введений пароль!!!",
                MessageBoxButtons.YesNo, MessageBoxIcon.Stop) == DialogResult.Yes)
            {
                //Закриваємо вікно для завершення роботи
                this.DialogResult = DialogResult.Abort;
            }
            else // або залишаємось у вікні
            {
                //Не закриваємо вікно для продовження авторизації
                e.Cancel = true;
            }
        }
    }

    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        //Співробітники прив'язуються до підрозділу
        this.dostupBindingSource.Filter = "Hotel = '" + Convert.ToString(this.comboBox1.SelectedValue) + "'";
    }
}

```

3. MainForm.cs

```

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace WindowsApplication1
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();

            private void HelpMenu_Click(object sender, EventArgs e)
            {
                // Відкриваємо повідомлення - діалогову форму "Про програму"
                MessageBox.Show("Мережа сімейних готелів \n\nВипускна Кваліфікаційна Робота.\n\nВиконав студент  
ДЕРЖАВНОГО УНІВЕРСИТЕТУ ТЕЛЕКОМУНІКАЦІЙ\nГолосов Владислав Сергійович \n\nг.Київ 2023", "Про  
програму...", MessageBoxButtons.YesNo, MessageBoxIcon.Information);
            }

            private void ExitMenu_Click(object sender, EventArgs e)
            {
                // Завершуємо роботу програми
                Application.Exit();
            }

            private void DolgnostMenu_Click(object sender, EventArgs e)
            {
                // Відкриваємо довідник Посад
                DolgnostForm newDolgnostForm = new DolgnostForm();
                newDolgnostForm.MdiParent = this;
                newDolgnostForm.Show();
            }

            private void PersonalMenu_Click(object sender, EventArgs e)
            {
                // Відкриваємо довідник персоналу
                PersonalForm NewPersonalForm = New PersonalForm();
                newPersonalForm.MdiParent = this;
                newPersonalForm.Show();
            }

            private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
            {
                //Якщо користувач негативно відповідає на запит про завершення роботи
                if (MessageBox.Show("Завершити роботу програми?", "Вихід...", MessageBoxButtons.YesNo,
                MessageBoxIcon.Question)==DialogResult.No)
                {
                    //То скасовуємо завершення
                    e.Cancel=true;
                }
            }
        }
    }
}

```

```

private void timer1_Tick(object sender, EventArgs e)
{
    // Дата та час на статусній панелі
    this.toolStripStatusLabel1.Text = DateTime.Now.ToString("dd MMMM yyyyг, dddd HH:mm:ss");
}

private void MainForm_Load(object sender, EventArgs e)
{
    //
}

private void TipNoMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо універсальну форму характеристик номерів
    XarakNomerForm newXarakNomerForm = new XarakNomerForm();
    newXarakNomerForm.MdiParent = this;
    //Пишемо свій заголовок
    newXarakNomerForm.Text = "Довідник типів готельних номерів";
    newXarakNomerForm.Tag = "1"; //Встановлюємо числове значення визначення робочої таблиці даних
    // Відкриваємо форму довідника
    newXarakNomerForm.Show();
}

private void CategoryMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо універсальну форму характеристик номерів
    XarakNomerForm newXarakNomerForm = new XarakNomerForm();
    newXarakNomerForm.MdiParent = this;
    //Пишемо свій заголовок
    newXarakNomerForm.Text = "Довідник категорій готельних номерів";
    newXarakNomerForm.Tag = "2"; //Встановлюємо числове значення визначення робочої таблиці даних
    // Відкриваємо форму довідника
    newXarakNomerForm.Show();
}

private void OptionsMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо універсальну форму характеристик номерів
    XarakNomerForm newXarakNomerForm = new XarakNomerForm();
    newXarakNomerForm.MdiParent = this;
    //Пишемо свій заголовок
    newXarakNomerForm.Text = "Довідник комплектації готельних номерів";
    newXarakNomerForm.Tag = "3"; //Встановлюємо числове значення визначення робочої таблиці даних
    // Відкриваємо форму довідника
    newXarakNomerForm.Show();
}

private void NomerMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно обробки списку номерів готелю
    NomerForm NewNomerForm = New NomerForm();
    newNomerForm.MdiParent = this;
    newNomerForm.Show();
}

```

```

private void ReceptMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно АРМ спеціаліста "Ресепшн"
    ReceptForm newReceptForm = new ReceptForm();
    newReceptForm.MdiParent = this;
    newReceptForm.Show();
}

private void SuperMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно АРМ Супервайзерів Готелів
    SuperForm newSuperForm = new SuperForm();
    newSuperForm.MdiParent=this;
    newSuperForm.Show();
}

private void FitZalMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно видів оздоровчих занять
    SportForm newSportForm = new SportForm();
    newSportForm.MdiParent = this;
    newSportForm.Show();
}

private void FitGroupMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно формування оздоровчих груп
    GroupForm NewGroupForm = New GroupForm();
    newGroupForm.MdiParent=this;
    newGroupForm.Show();
}

private void FitMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно запису на фітнес
    FitForm newFitForm = new FitForm();
    newFitForm.MdiParent=this;
    newFitForm.Show();
}

private void PitanieMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно довідника живлення
    PitanieForm newPitanieForm = new PitanieForm();
    newPitanieForm.MdiParent=this;
    newPitanieForm.Show();
}

private void RestorMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо вікно запису харчування
    RestoranForm newRestoranForm = new RestoranForm();
    newRestoranForm.MdiParent = this;
    newRestoranForm.Show();
}

```



```

private void HotelsMenu_Click(object sender, EventArgs e)
{
    // Відкриваємо довідник об'єктів мережі готелів
    HotelsForm NewHotelsForm = New HotelsForm();
    newHotelsForm.MdiParent=this;
    newHotelsForm.Show();
}
}
}

```

4. DolgnostForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class DolgnostForm : Form
    {
        // Визначаємо ресурси для роботи з Microsoft.Office.Interop.Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

        public DolgnostForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "HotelsDataSet.Dolgnost". За потреби
            // вона може бути переміщена або видалена.
            this.dolgnostTableAdapter.Fill(this.HotelsDataSet.Dolgnost);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "HotelsDataSet.DolgnostView". За
            // потреби вона може бути переміщена або видалена.
            this.dolgnostViewTableAdapter.Fill(this.HotelsDataSet.DolgnostView);
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //Підключаємо програму Microsoft.Office.Interop.Excel
            app = new Microsoft.Office.Interop.Excel.ApplicationClass();

```

```

// Відкриваємо призначену для поточного звіту робочу книгу
string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/dolgnost.xls";
workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
"\t", false, false, 0, true, 0, false);
// Відкриваємо робочий лист - за замовчуванням перший
worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
// TODO: цей рядок коду дозволяє завантажити дані в таблицю "HotelDataSet.dolgnostView". За потреби вона
може бути переміщена або видалена.
this.dolgnostViewTableAdapter.Fill(this.HotelsDataSet.DolgnostView);
// Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
for (int i = 0; i <= this.HotelsDataSet.DolgnostView.Rows.Count-1; i++)
{
    worksheet.Cells[i + 2, 1] = Convert.ToString(i + 1);
    worksheet.Cells[i + 2, 2] = this.HotelsDataSet.DolgnostView.Rows[i][0].ToString();
    worksheet.Cells[i + 2, 3] = this.HotelsDataSet.DolgnostView.Rows[i][1].ToString();
    worksheet.Cells[i + 2, 4] = this.HotelsDataSet.DolgnostView.Rows[i][2].ToString();
}
// Робимо Microsoft.Office.Interop.Excel видимим
app.Visible = true;
// Включаємо режим попереднього перегляду
workbook.PrintPreview(false);
// Закриваємо робочу книгу
workbook.Close(false, Sp, false);
// Закриваємо Microsoft.Office.Interop.Excel
app.Quit();
}

private void buttons_Click(object sender, EventArgs e)
{
    //Кнопки навігації за таблицею даних
    if (sender == button2) { this.dolgnostBindingSource.MoveFirst(); }
    if (sender == button3) { this.dolgnostBindingSource.MovePrevious(); }
    if (sender == button4) { this.dolgnostBindingSource.MoveNext(); }
    if (sender == button5) { this.dolgnostBindingSource.MoveLast(); }
    //Додати новий запис
    if (sender == button7) { this.dolgnostBindingSource.AddNew(); }
    //Видалити поточний запис
    if (sender == button8)
    {
        if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
        else this.dolgnostBindingSource.RemoveCurrent();
    }
}

private void button6_Click(object sender, EventArgs e)
{
    //Процедура збереження таблиці даних
    this.dolgnostBindingSource.EndEdit();
    this.dolgnostTableAdapter.Update(this.HotelsDataSet);
    this.HotelsDataSet.AcceptChanges();
}

private void button9_Click(object sender, EventArgs e)
{

```

```

        //Закрити форму
        this.Close();
    }
}
}

```

5. FitForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class FitForm : Form
    {
        public FitForm()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.GroupV". За потреби вона може
                // бути переміщена або видалена.
                this.groupVTableAdapter.Fill(this.hotelsDataSet.GroupV);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.ClientF". За потреби вона може
                // бути переміщена або видалена.
                this.clientFTableAdapter.Fill(this.hotelsDataSet.ClientF);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.FitnessGroup". За потреби вона
                // може бути переміщена або видалена.
                this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Fitness". За потреби вона може
                // бути переміщена або видалена.
                this.fitnessTableAdapter.Fill(this.hotelsDataSet.Fitness);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelDataSet.GroupV". За потреби вона може
                // бути переміщена або видалена.
                this.groupVTableAdapter.Fill(this.hotelsDataSet.GroupV);
            }

            private void buttons_Click(object sender, EventArgs e)
            {
                //Кнопки навігації за таблицею даних
                if (sender == button2) { this.fitnessBindingSource.MoveFirst(); }
                if (sender == button3) { this.fitnessBindingSource.MovePrevious(); }
                if (sender == button4) { this.fitnessBindingSource.MoveNext(); }
                if (sender == button5) { this.fitnessBindingSource.MoveLast(); }
            }
        }
    }
}

```

```

        //Додати новий запис
        if (sender == button7) { this.fitnessBindingSource.AddNew(); }
        //Видалити поточний запис
        if (sender == button8)
        {
            if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
            else this.fitnessBindingSource.RemoveCurrent();
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.fitnessBindingSource.EndEdit();
        this.fitnessTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }

    private void fitnessBindingSource_PositionChanged(object sender, EventArgs e)
    {
        if (fitnessBindingSource.Position < this.hotelsDataSet.Fitness.Rows.Count)
            //Вираховуємо суму до оплати на текстовий елемент
            this.label1.Text = String.Format("{0:C2}",
            double.Parse(this.textBox4.Text, System.Globalization.NumberStyles.Currency) *
            double.Parse(this.hotelsDataSet.Fitness.Rows[fitnessBindingSource.Position][1].ToString(),
System.Globalization.NumberStyles.Currency));
            //Використовуємо перетворення типів, виходячи з місцевого формату грошових одиниць
        }
    }
}

```

6. GroupForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{

```

```

public partial class GroupForm : Form
{
    // Визначаємо ресурси для роботи з Excel
    private Microsoft.Office.Interop.Excel.Application app = null;
    private Microsoft.Office.Interop.Excel.Workbook workbook = null;
    private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

    public GroupForm()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.GroupV". За потреби вона може
        // бути переміщена або видалена.
        this.groupVTableAdapter.Fill(this.hotelsDataSet.GroupV);
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.FitnessGroup". За потреби вона
        // може бути переміщена або видалена.
        this.fitnessGroupTableAdapter.Fill(this.hotelsDataSet.FitnessGroup);
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.FitnessZal". За потреби вона
        // може бути переміщена або видалена.
        this.fitnessZalTableAdapter.Fill(this.hotelsDataSet.FitnessZal);
    }

    private void buttons_Click(object sender, EventArgs e)
    {
        //Кнопки навігації за таблицею даних
        if (sender == button2) { this.fitnessGroupBindingSource.MoveFirst(); }
        if (sender == button3) { this.fitnessGroupBindingSource.MovePrevious(); }
        if (sender == button4) { this.fitnessGroupBindingSource.MoveNext(); }
        if (sender == button5) { this.fitnessGroupBindingSource.MoveLast(); }
        //Додати новий запис
        if (sender == button7) { this.fitnessGroupBindingSource.AddNew(); }
        //Видалити поточний запис
        if (sender == button8)
        {
            if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
            else this.fitnessGroupBindingSource.RemoveCurrent();
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.fitnessGroupBindingSource.EndEdit();
        this.fitnessGroupTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }
}

```

```

    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        //Підключаємо додаток Excel
        app = new Microsoft.Office.Interop.Excel.ApplicationClass();
        // Відкриваємо призначену для поточного звіту робочу книгу
        string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/Group.xls";
        // Відкриваємо робочий лист - за замовчуванням перший
        workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
        "\t", false, false, 0, true, 0, false);
        // Записуємо в комірку заголовка таблиці необхідний текст
        worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelDataSet.GroupV". За потреби вона може
        бути переміщена або видалена.
        this.groupVTableAdapter.Fill(this.hotelsDataSet.GroupV);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.hotelsDataSet.GroupV.Rows.Count - 1; i++)
        {
            for (int j = 0; j <= 12; j++)
                if ((j == 4) || (j == 5)) worksheet.Cells[i + 2, j + 1] = String.Format("{0:t}",
                Convert.ToDateTime(this.hotelsDataSet).GroupV.Rows[i][j].ToString());
            else worksheet.Cells[i + 2, j + 1] = this.hotelsDataSet.GroupV.Rows[i][j].ToString();
        }
        // Робимо Excel видимим
        app.Visible = true;
        // Включаємо режим попереднього перегляду
        workbook.PrintPreview(false);
        // Закриваємо робочу книгу
        workbook.Close(false, Sp, false);
        // Закриваємо Excel
        app.Quit();
    }
}
}

```

7. HotelsForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class HotelsForm : Form
    {
        public HotelsForm()
    }
}

```

```

{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "HotelsDataSet.Dolgnost". За потреби вона може
    // бути переміщена або видалена.
    this.hotelNetTableAdapter.Fill(this.HotelsDataSet.HotelNet);
}

private void buttons_Click(object sender, EventArgs e)
{
    //Кнопки навігації за таблицею даних
    if (sender == button2) { this.hotelNetBindingSource.MoveFirst(); }
    if (sender == button3) { this.hotelNetBindingSource.MovePrevious(); }
    if (sender == button4) { this.hotelNetBindingSource.MoveNext(); }
    if (sender == button5) { this.hotelNetBindingSource.MoveLast(); }
    //Додати новий запис
    if (sender == button7) { this.hotelNetBindingSource.AddNew(); }
    //Видалити поточний запис
    if (sender == button8)
    {
        if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
        DialogResult.No) return;
        else this.hotelNetBindingSource.RemoveCurrent();
    }
}

private void button6_Click(object sender, EventArgs e)
{
    //Процедура збереження таблиці даних
    this.hotelNetBindingSource.EndEdit();
    this.hotelNetTableAdapter.Update(this.HotelsDataSet);
    this.HotelsDataSet.AcceptChanges();
}

private void button9_Click(object sender, EventArgs e)
{
    //Закрити форму
    this.Close();
}
}

```

8. NomerForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

```

```

using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class NomerForm : Form
    {
        // Визначаємо ресурси для роботи з Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

        public NomerForm()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerHV". За потреби вона
                // може бути переміщена або видалена.
                this.nomerHVTableAdapter.Fill(this.hotelsDataSet.NomerHV);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.HotelNet". За потреби вона може
                // бути переміщена або видалена.
                this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerOptions". За потреби вона
                // може бути переміщена або видалена.
                this.nomerOptionsTableAdapter.Fill(this.hotelsDataSet.NomerOptions);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerCategory". За потреби вона
                // може бути переміщена або видалена.
                this.nomerCategoryTableAdapter.Fill(this.hotelsDataSet.NomerCategory);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerTip". За потреби вона
                // може бути переміщена або видалена.
                this.nomerTipTableAdapter.Fill(this.hotelsDataSet.NomerTip);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Nomers". За потреби вона може
                // бути переміщена або видалена.
                this.nomersTableAdapter.Fill(this.hotelsDataSet.Nomers);
            }

            private void buttons_Click(object sender, EventArgs e)
            {
                //Кнопки навігації за таблицею даних
                if (sender == button2) { this.nomersBindingSource.MoveFirst(); }
                if (sender == button3) { this.nomersBindingSource.MovePrevious(); }
                if (sender == button4) { this.nomersBindingSource.MoveNext(); }
                if (sender == button5) { this.nomersBindingSource.MoveLast(); }
                //Додати новий запис
                if (sender == button7) { this.nomersBindingSource.AddNew(); }
                //Видалити поточний запис
                if (sender == button8)
                {
                    if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
                        DialogResult.No) return;
                    else this.nomersBindingSource.RemoveCurrent();
                }
            }
        }
    }
}

```



```

    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.nomersBindingSource.EndEdit();
        this.nomersTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        //Підключаємо додаток Excel
        app = new Microsoft.Office.Interop.Excel.ApplicationClass();
        // Відкриваємо призначену для поточного звіту робочу книгу
        string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/Nomera.xls";
        // Відкриваємо робочий лист - за замовчуванням перший
        workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
        "\t", false, false, 0, true, 0, false);
        // Записуємо в комірку заголовка таблиці необхідний текст
        worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelDataSet.NomerHV". За потреби вона може
        бути переміщена або видалена.
        this.nomerHVTableAdapter.Fill(this.hotelsDataSet.NomerHV);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.hotelsDataSet.NomerHV.Rows.Count - 1; i++)
        {
            //Типи номерів
            worksheet.Cells[i + 3, 1] = this.hotelsDataSet.NomerHV.Rows[i][0].ToString();
            worksheet.Cells[i + 3, 2] = this.hotelsDataSet.NomerHV.Rows[i][1].ToString();
            worksheet.Cells[i + 3, 3] = this.hotelsDataSet.NomerHV.Rows[i][2].ToString();
            worksheet.Cells[i + 3, 4] = this.hotelsDataSet.NomerHV.Rows[i][3].ToString();
            worksheet.Cells[i + 3, 5] = this.hotelsDataSet.NomerHV.Rows[i][4].ToString();
            worksheet.Cells[i + 3, 6] = this.hotelsDataSet.NomerHV.Rows[i][5].ToString();
            worksheet.Cells[i + 3, 7] = this.hotelsDataSet.NomerHV.Rows[i][6].ToString();
            worksheet.Cells[i + 3, 8] = this.hotelsDataSet.NomerHV.Rows[i][7].ToString();
            worksheet.Cells[i + 3, 9] = this.hotelsDataSet.NomerHV.Rows[i][8].ToString();
        }
        // Робимо Excel видимим
        app.Visible = true;
        // Включаємо режим попереднього перегляду
        workbook.PrintPreview(false);
        // Закриваємо робочу книгу
        workbook.Close(false, Sp, false);
        // Закриваємо Excel
        app.Quit();
    }
}
}

```

9. PersonalForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class PersonalForm : Form
    {
        // Визначаємо ресурси для роботи з Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

        Public PersonalForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.SpisokP". За потреби вона може
            // бути переміщена або видалена.
            this.spisokPTableAdapter.Fill(this.hotelsDataSet.SpisokP);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Dolgnost". За потреби вона може
            // бути переміщена або видалена.
            this.dolgnostTableAdapter.Fill(this.hotelsDataSet.Dolgnost);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.HotelNet". За потреби вона може
            // бути переміщена або видалена.
            this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Personal". За потреби вона може
            // бути переміщена або видалена.
            this.personalTableAdapter.Fill(this.hotelsDataSet.Personal);
        }

        private void buttons_Click(object sender, EventArgs e)
        {
            //Кнопки навігації за таблицею даних
            if (sender == button2) { this.personalBindingSource.MoveFirst(); }
            if (sender == button3) { this.personalBindingSource.MovePrevious(); }
            if (sender == button4) { this.personalBindingSource.MoveNext(); }
            if (sender == button5) { this.personalBindingSource.MoveLast(); }
            //Додати новий запис
            if (sender == button7) { this.personalBindingSource.AddNew(); }
        }
    }
}

```

```

        //Видалити поточний запис
        if (sender == button8)
        {
            if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
            else this.personalBindingSource.RemoveCurrent();
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.personalBindingSource.EndEdit();
        this.personalTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        //Підключаємо додаток Excel
        app = new Microsoft.Office.Interop.Excel.ApplicationClass();
        // Відкриваємо призначену для поточного звіту робочу книгу
        string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/Personal.xls";
        // Відкриваємо робочий лист - за замовчуванням перший
        workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
"\t", false, false, 0, true, 0, false);
        // Записуємо в комірку заголовка таблиці необхідний текст
        worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelDataSet.SpisokP". За потреби вона може
бути переміщена або видалена.
        this.spisokPTableAdapter.Fill(this.hotelsDataSet.SpisokP);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.hotelsDataSet.SpisokP.Rows.Count - 1; i++)
        {
            worksheet.Cells[i + 2, 1] = this.hotelsDataSet.SpisokP.Rows[i][0].ToString();
            worksheet.Cells[i + 2, 2] = this.hotelsDataSet.SpisokP.Rows[i][1].ToString();
            worksheet.Cells[i + 2, 3] = this.hotelsDataSet.SpisokP.Rows[i][2].ToString();
            worksheet.Cells[i + 2, 4] = this.hotelsDataSet.SpisokP.Rows[i][3].ToString();
            worksheet.Cells[i + 2, 5] = this.hotelsDataSet.SpisokP.Rows[i][4].ToString();
            worksheet.Cells[i + 2, 6] = this.hotelsDataSet.SpisokP.Rows[i][5].ToString();
        }
        // Робимо Excel видимим
        app.Visible = true;
        // Включаємо режим попереднього перегляду
        workbook.PrintPreview(false);
        // Закриваємо робочу книгу
        workbook.Close(false, Sp, false);
        // Закриваємо Excel
    }

```

```

        app.Quit();
    }
}

```

10.PitanieForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class PitanieForm : Form
    {
        public PitanieForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Pitanie". За потреби вона може
            // бути переміщена або видалена.
            this.pitanieTableAdapter.Fill(this.hotelsDataSet.Pitanie);
        }

        private void buttons_Click(object sender, EventArgs e)
        {
            //Кнопки навігації за таблицею даних
            if (sender == button2) { this.pitanieBindingSource.MoveFirst(); }
            if (sender == button3) { this.pitanieBindingSource.MovePrevious(); }
            if (sender == button4) { this.pitanieBindingSource.MoveNext(); }
            if (sender == button5) { this.pitanieBindingSource.MoveLast(); }
            //Додати новий запис
            if (sender == button7) { this.pitanieBindingSource.AddNew(); }
            //Видалити поточний запис
            if (sender == button8)
            {
                if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
                DialogResult.No) return;
                else this.pitanieBindingSource.RemoveCurrent();
            }
        }

        private void button6_Click(object sender, EventArgs e)
        {
            //Процедура збереження таблиці даних
            this.pitanieBindingSource.EndEdit();
        }
    }
}

```

```

        this.pitanieTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }
}

```

11.ReceptForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class ReceptForm : Form
    {
        // Визначаємо ресурси для роботи з Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

        public ReceptForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerView". За потреби вона
            // може бути переміщена або видалена.
            this.nomerViewTableAdapter.Fill(this.hotelsDataSet.NomerView);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Clients". За потреби вона може
            // бути переміщена або видалена.
            this.clientsTableAdapter.Fill(this.hotelsDataSet.Clients);
            // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelDataSet.clients". За потреби вона може
            // бути переміщена або видалена.
            this.clientsTableAdapter.Fill(this.hotelsDataSet.Clients);
        }

        private void buttons_Click(object sender, EventArgs e)
        {
            //Кнопки навігації за таблицею даних

```

```

        if (sender == button2) { this.clientsBindingSource.MoveFirst(); }
        if (sender == button3) { this.clientsBindingSource.MovePrevious(); }
        if (sender == button4) { this.clientsBindingSource.MoveNext(); }
        if (sender == button5) { this.clientsBindingSource.MoveLast(); }
        //Додати новий запис
        if (sender == button7) { this.clientsBindingSource.AddNew(); }
        //Видалити поточний запис
        if (sender == button8) { this.clientsBindingSource.RemoveCurrent(); }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.clientsBindingSource.EndEdit();
        this.clientsTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }

    private void button1_Click_1(object sender, EventArgs e)
    {
        //Підключаємо додаток Excel
        app = new Microsoft.Office.Interop.Excel.ApplicationClass();
        // Відкриваємо призначену для поточного звіту робочу книгу
        string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/Clients.xls";
        // Відкриваємо робочий лист - за замовчуванням перший
        workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
        "\t", false, false, 0, true, 0, false);
        // Записуємо в комірку заголовка таблиці необхідний текст
        worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.ClientsView". За потреби вона
        може бути переміщена або видалена.
        this.clientsViewTableAdapter.Fill(this.hotelsDataSet.ClientsView);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.hotelsDataSet.ClientsView.Rows.Count - 1; i++)
        {
            for (int j = 0; j <= 12; j++)
            {
                if (j == 8) worksheet.Cells[i + 2, j + 1] = String.Format("{0:t}",
                Convert.ToDateTime(this.hotelsDataSet.ClientsView.Rows[i][j]). ToString());
                else worksheet.Cells[i + 2, j + 1] = this.hotelsDataSet.ClientsView.Rows[i][j].ToString();
            }
        }
        // Робимо Excel видимим
        app.Visible = true;
        // Включаємо режим попереднього перегляду
        workbook.PrintPreview(false);
        // Закриваємо робочу книгу
        workbook.Close(false, Sp, false);
        // Закриваємо Excel
    }

```

```

        app.Quit();
    }
}

```

12.RestoranForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class RestoranForm : Form
    {
        public RestoranForm()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.ClientF". За потреби вона може
                // бути переміщена або видалена.
                this.clientFTableAdapter.Fill(this.hotelsDataSet.ClientF);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.restoranGroup". За потреби вона
                // може бути переміщена або видалена.
                this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Pitanie". За потреби вона може
                // бути переміщена або видалена.
                this.pitanieTableAdapter.Fill(this.hotelsDataSet.Pitanie);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.restoran". За потреби вона може
                // бути переміщена або видалена.
                this.restoranTableAdapter.Fill(this.hotelsDataSet.Restoran);
            }

            private void buttons_Click(object sender, EventArgs e)
            {
                //Кнопки навігації за таблицею даних
                if (sender == button2) { this.restoranBindingSource.MoveFirst(); }
                if (sender == button3) { this.restoranBindingSource.MovePrevious(); }
                if (sender == button4) { this.restoranBindingSource.MoveNext(); }
                if (sender == button5) { this.restoranBindingSource.MoveLast(); }
                //Додати новий запис
                if (sender == button7) { this.restoranBindingSource.AddNew(); }
                //Видалити поточний запис
                if (sender == button8)

```

```

        {
            if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
            else this.restoranBindingSource.RemoveCurrent();
        }
    }

    private void button6_Click(object sender, EventArgs e)
    {
        //Процедура збереження таблиці даних
        this.restoranBindingSource.EndEdit();
        this.restoranTableAdapter.Update(this.hotelsDataSet);
        this.hotelsDataSet.AcceptChanges();
    }

    private void button9_Click(object sender, EventArgs e)
    {
        //Закрити форму
        this.Close();
    }

    private void restoranBindingSource_PositionChanged(object sender, EventArgs e)
    {
        if (restoranBindingSource.Position < this.hotelsDataSet.Restoran.Rows.Count)
            //Виразуємо суму до оплати на текстовий елемент
            this.label1.Text = String.Format("{0:C2}",
            double.Parse(this.textBox4.Text, System.Globalization.NumberStyles.Currency) *
            double.Parse(this.hotelsDataSet.Restoran.Rows[restoranBindingSource.Position][1].ToString(),
System.Globalization.NumberStyles.Currency));
            //Використовуємо перетворення типів, виходячи з місцевого формату грошових одиниць
        }
    }
}

```

13.SportForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class SportForm : Form
    {
        // Визначаємо ресурси для роботи з Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;
    }
}

```



```

public SportForm()
{
    InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.FitnessZal". За потреби вона
    // може бути переміщена або видалена.
    this.fitnessZalTableAdapter.Fill(this.hotelsDataSet.FitnessZal);
}

private void buttons_Click(object sender, EventArgs e)
{
    //Кнопки навігації за таблицею даних
    if (sender == button2) { this.fitnessZalBindingSource.MoveFirst(); }
    if (sender == button3) { this.fitnessZalBindingSource.MovePrevious(); }
    if (sender == button4) { this.fitnessZalBindingSource.MoveNext(); }
    if (sender == button5) { this.fitnessZalBindingSource.MoveLast(); }
    //Додати новий запис
    if (sender == button7) { this.fitnessZalBindingSource.AddNew(); }
    //Видалити поточний запис
    if (sender == button8)
    {
        if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
        else this.fitnessZalBindingSource.RemoveCurrent();
    }
}

private void button6_Click(object sender, EventArgs e)
{
    //Процедура збереження таблиці даних
    this.fitnessZalBindingSource.EndEdit();
    this.fitnessZalTableAdapter.Update(this.hotelsDataSet);
    this.hotelsDataSet.AcceptChanges();
}

private void button9_Click(object sender, EventArgs e)
{
    //Закрити форму
    this.Close();
}

private void button1_Click_1(object sender, EventArgs e)
{
    //Підключаємо додаток Excel
    app = new Microsoft.Office.Interop.Excel.ApplicationClass();
    // Відкриваємо призначену для поточного звіту робочу книгу
    string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/Sportzal.xls";
    // Відкриваємо робочий лист - за замовчуванням перший
    workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
    "", false, false, 0, true, 0, false);
    // Записуємо в комірку заголовка таблиці необхідний текст

```

```

worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
// Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
for (int i = 0; i <= this.hotelsDataSet.FitnessZal.Rows.Count - 1; i++)
{
    worksheet.Cells[i + 2, 1] = this.hotelsDataSet.FitnessZal.Rows[i][0].ToString();
    worksheet.Cells[i + 2, 2] = this.hotelsDataSet.FitnessZal.Rows[i][1].ToString();
    worksheet.Cells[i + 2, 3] = this.hotelsDataSet.FitnessZal.Rows[i][2].ToString();
    worksheet.Cells[i + 2, 4] = this.hotelsDataSet.FitnessZal.Rows[i][3].ToString();
}
// Робимо Excel видимим
app.Visible = true;
// Включаємо режим попереднього перегляду
workbook.PrintPreview(false);
// Закриваємо робочу книгу
workbook.Close(false, Sp, false);
// Закриваємо Excel
app.Quit();
}
}
}

```

14. SuperForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class SuperForm : Form
    {
        public SuperForm()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.Nomers". За потреби вона може
                // бути переміщена або видалена.
                this.nomersTableAdapter.Fill(this.hotelsDataSet.Nomers);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.PlanU". За потреби вона може
                // бути переміщена або видалена.
                this.planUTableAdapter.Fill(this.hotelsDataSet.PlanU);
                // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.HotelNet". За потреби вона може
                // бути переміщена або видалена.
                this.hotelNetTableAdapter.Fill(this.hotelsDataSet.HotelNet);
            }
        }
    }
}

```

```

private void button9_Click(object sender, EventArgs e)
{
    //Закрити форму
    this.Close();
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    //Якщо список готелів не порожній, встановлюємо фільтр для переглядових таблиць
    if (this.comboBox1.SelectedIndex > -1)
    {
        if (this.comboBox1.SelectedValue.ToString() == "1")
        {
            //Якщо вибрали адміністрацію, то фільтр не потрібен
            this.nomersBindingSource.Filter = "";
            this.planUBindingSource.Filter = "";
        }
        else
        {
            this.nomersBindingSource.Filter = "Hotel = " + this.comboBox1.SelectedValue.ToString();
            this.planUBindingSource.Filter = "Hotel = " + this.comboBox1.SelectedValue.ToString();
        }
    }
}
} } }

```

15.XarakNomerForm.cs

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
using Microsoft.Office.Interop.Excel;
using System.Configuration;

namespace WindowsApplication1
{
    public partial class XarakNomerForm : Form
    {
        // Визначаємо ресурси для роботи з Excel
        private Microsoft.Office.Interop.Excel.Application app = null;
        private Microsoft.Office.Interop.Excel.Workbook workbook = null;
        private Microsoft.Office.Interop.Excel.Worksheet worksheet = null;

        public XarakNomerForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

```

```

// Залежно від переданого на цю форму з головного меню ознаки Tag відкриваємо потрібну таблицю даних
if (this.Tag.ToString() == "1")
{
    this.xaraknomerBindingSource.DataMember = "NomerTip";
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerTip". За потреби вона
може бути переміщена або видалена.
    this.nomerTipTableAdapter.Fill(this.HotelsDataSet.NomerTip);
}
if (this.Tag.ToString() == "2")
{
    this.xaraknomerBindingSource.DataMember = "NomerCategory";
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerCategory". За потреби
вона може бути переміщена або видалена.
    this.nomerCategoryTableAdapter.Fill(this.HotelsDataSet.NomerCategory);
}
if (this.Tag.ToString() == "3")
{
    this.xaraknomerBindingSource.DataMember = "NomerOptions";
    // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.NomerOptions". За потреби
вона може бути переміщена або видалена.
    this.nomerOptionsTableAdapter.Fill(this.HotelsDataSet.NomerOptions);
}
}

private void buttons_Click(object sender, EventArgs e)
{
    //Кнопки навігації за таблицею даних
    if (sender == button2) { this.xaraknomerBindingSource.MoveFirst(); }
    if (sender == button3) { this.xaraknomerBindingSource.MovePrevious(); }
    if (sender == button4) { this.xaraknomerBindingSource.MoveNext(); }
    if (sender == button5) { this.xaraknomerBindingSource.MoveLast(); }
    //Додати новий запис
    if (sender == button7) { this.xaraknomerBindingSource.AddNew(); }
    //Видалити поточний запис
    if (sender == button8)
    {
        if (MessageBox.Show("Видалити запис?", this.Text, MessageBoxButtons.YesNo, MessageBoxIcon.Question) ==
DialogResult.No) return;
        else this.xaraknomerBindingSource.RemoveCurrent();
    }
}

private void button6_Click(object sender, EventArgs e)
{
    //Процедура збереження таблиці даних
    this.xaraknomerBindingSource.EndEdit();
    if (this.Tag.ToString() == "1") this.nomerTipTableAdapter.Update(this.HotelsDataSet);
    if (this.Tag.ToString() == "2") this.nomerCategoryTableAdapter.Update(this.HotelsDataSet);
    if (this.Tag.ToString() == "3") this.nomerOptionsTableAdapter.Update(this.HotelsDataSet);
    this.HotelsDataSet.AcceptChanges();
}

private void button9_Click(object sender, EventArgs e)
{
    //Закрити форму

```

```

    this.Close();
}

private void button1_Click_1(object sender, EventArgs e)
{
    //Підключаємо додаток Excel
    app = new Microsoft.Office.Interop.Excel.ApplicationClass();
    // Відкриваємо призначену для поточного звіту робочу книгу
    string Sp = System.Windows.Forms.Application.StartupPath + "/Docs/XarakNomer.xls";
    // Відкриваємо робочий лист - за замовчуванням перший
    workbook = app.Workbooks.Open(Sp, 0, true, 5, "", "", true, Microsoft.Office.Interop.Excel.XlPlatform.xlWindows,
"\t", false, false, 0, true, 0, false);
    // Записуємо в комірку заголовка таблиці необхідний текст
    worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.ActiveSheet;
    //Пишемо заголовок звіту залежно від виду звіту
    worksheet.Cells[1, 1] = this.Text;
    // Залежно від того, яка таблиця - робоча, вибираємо запит для звіту:
    if (this.Tag.ToString() == "1")
    {
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.TipNoHV". За потреби вона
        може бути переміщена або видалена.
        this.tipNoHVTableAdapter.Fill(this.HotelsDataSet.TipNoHV);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.HotelsDataSet.TipNoHV.Rows.Count - 1; i++)
        {
            //Типи номерів
            worksheet.Cells[i + 4, 1] = this.HotelsDataSet.TipNoHV.Rows[i][0].ToString();
            worksheet.Cells[i + 4, 2] = this.HotelsDataSet.TipNoHV.Rows[i][1].ToString();
            worksheet.Cells[i + 4, 3] = this.HotelsDataSet.TipNoHV.Rows[i][2].ToString();
            worksheet.Cells[i + 4, 4] = this.HotelsDataSet.TipNoHV.Rows[i][3].ToString();
            worksheet.Cells[i + 4, 5] = this.HotelsDataSet.TipNoHV.Rows[i][4].ToString();
        }
    }
    if (this.Tag.ToString() == "2")
    {
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.CategoryHV". За потреби вона
        може бути переміщена або видалена.
        this.categoryHVTableAdapter.Fill(this.HotelsDataSet.CategoryHV);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші
        for (int i = 0; i <= this.HotelsDataSet.CategoryHV.Rows.Count - 1; i++)
        {
            //Категорії номерів
            worksheet.Cells[i + 4, 1] = this.HotelsDataSet.CategoryHV.Rows[i][0].ToString();
            worksheet.Cells[i + 4, 2] = this.HotelsDataSet.CategoryHV.Rows[i][1].ToString();
            worksheet.Cells[i + 4, 3] = this.HotelsDataSet.CategoryHV.Rows[i][2].ToString();
            worksheet.Cells[i + 4, 4] = this.HotelsDataSet.CategoryHV.Rows[i][3].ToString();
            worksheet.Cells[i + 4, 5] = this.HotelsDataSet.CategoryHV.Rows[i][4].ToString();
        }
    }
    if (this.Tag.ToString() == "3")
    {
        // TODO: цей рядок коду дозволяє завантажити дані в таблицю "hotelsDataSet.OptionsHV". За потреби вона
        може бути переміщена або видалена.
        this.optionsHVTableAdapter.Fill(this.HotelsDataSet.OptionsHV);
        // Рядково пересуваємося по робочій таблиці даних і записуємо результати до таблиці на аркуші

```

```

for (int i = 0; i <= this.HotelsDataSet.OptionsHV.Rows.Count - 1; i++)
{
    //Додаткові опції
    worksheet.Cells[i + 4, 1] = this.HotelsDataSet.OptionsHV.Rows[i][0].ToString();
    worksheet.Cells[i + 4, 2] = this.HotelsDataSet.OptionsHV.Rows[i][1].ToString();
    worksheet.Cells[i + 4, 3] = this.HotelsDataSet.OptionsHV.Rows[i][2].ToString();
    worksheet.Cells[i + 4, 4] = this.HotelsDataSet.OptionsHV.Rows[i][3].ToString();
    worksheet.Cells[i + 4, 5] = this.HotelsDataSet.OptionsHV.Rows[i][4].ToString();
}
}
// Робимо Excel видимим
app.Visible = true;
// Включаємо режим попереднього перегляду
workbook.PrintPreview(false);
// Закриваємо робочу книгу
workbook.Close(false, Sp, false);
// Закриваємо Excel
app.Quit();
}
}
}

```

16.GorodForm.aspx

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace LCompany
{
    public partial class GorodForm : System.Web.UI.Page
    {
        protected void Button1_Click1(object sender, EventArgs e)
        {
            // Формуємо команду додавання міста
            this.SqlDataSource1.InsertCommand = "INSERT INTO Gorod (Name,X,Y,Punkt) VALUES ('" + "Нове Місто" +
                "+" + 0 + ", " + 0 + ", " + false + "'";
            // Додаємо дані
            this.SqlDataSource1.Insert();
            //Оновлюємо дані в таблиці
            this.GridView1.DataBind();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }
    }
}

```

17.PriceList.aspx

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace LCompany
{
    public partial class PriceList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Page_Init(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            // Формуємо команду додавання запису до прайс-листу
            this.SqlDataSource1.InsertCommand = "INSERT INTO Price (Ves,Obyom,Data) VALUES('" + 0 +
            "', '" + 0 + "', '" + DateTime.Now + "')";
            // Додаємо розцінку з поточною датою
            this.SqlDataSource1.Insert();
            //Оновлюємо дані в таблиці
            this.GridView1.DataBind();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }
    }
}

```

18.StatusList.aspx

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace LCompany
{
    public partial class StatusList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Page_Init(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            // Формуємо команду додавання запису до статусів
            this.SqlDataSource1.InsertCommand = "INSERT INTO Status (Name) VALUES ('" + "Новий статус" + "')";
            // Додаємо розцінку з поточною датою
            this.SqlDataSource1.Insert();
            //Оновлюємо дані в таблиці
            this.GridView1.DataBind();
        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }
    }
}

```

19.TransportList.aspx

```

using System;
using System.Collections.Generic;

```



```

using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace LCompany
{
    public partial class TransportList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            // Якщо нічого в полях не введено, нічого не робимо і виходимо
            if ((this.TextBox1.Text == "") || (this.TextBox2.Text == "") || (this.TextBox3.Text == "") || (this.TextBox4.Text == " ")
            || (this.TextBox5.Text == "")) return;
            // Формуємо команду додавання автомобіля
            this.SqlDataSource1.InsertCommand = "INSERT INTO Transport (Marka,Nomer,FIO,Phone) VALUES(" +
            this.TextBox1.Text +
            ", " + this.TextBox2.Text + ", " + this.TextBox3.Text + ", " + this.TextBox4.Text + ", " + this.TextBox5.Text +
            " )";
            // Додаємо автомобіль
            this.SqlDataSource1.Insert();
            //Оновлюємо дані в таблиці
            this.GridView1.DataBind();
            // Очищаємо поля введення
            this.TextBox1.Text = "";
            this.TextBox2.Text = "";
            this.TextBox3.Text = "";
            this.TextBox4.Text = "";
        }
    }
}

```

20.KlientList.aspx

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;

```

```

using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace LCompany
{
    public partial class KlientList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Page_Init(object sender, EventArgs e)
        {

        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }
    }
}

```

21.OperatorList.aspx

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

namespace LCompany
{
    public partial class OperatorList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}

```

```

protected void Page_Init(object sender, EventArgs e)
{

}

protected void Button1_Click(object sender, EventArgs e)
{
    // Формуємо команду додавання співробітника
    this.SqlDataSource1.InsertCommand = "INSERT INTO Operators (FIO,Parol) VALUES ('" + "Новий Оператор" +
        "', '" + "'" + "'")";
    // Додаємо Співробітника
    this.SqlDataSource1.Insert();
    //Оновлюємо дані в таблиці
    this.GridView1.DataBind();
}

protected void Button2_Click(object sender, EventArgs e)
{
    // Переходимо на головну сторінку співробітника
    this.Response.Redirect("OperatorMenu.aspx");
}
}
}

```

22.ZayavkaForm.aspx

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace LCompany
{
    public partial class ZayavkaForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }

        protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {

```

```

        //При збереженні в GridView1 оновлюємо значення списку в поточному рядку
        this.SqlDataSource1.UpdateCommand = "UPDATE [Zayavka] SET [DataZ] = @DataZ, [Kontragent] = @Kontragent,
        [Tekst] = @Tekst, [Chat] = @Chat, [Operator] = 1 WHERE [Kod] = @Kod";
    }
}

```

23.ZakazForm.aspx

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace LCompany
{
    public partial class ZakazForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }

        protected void Button1_Click1(object sender, EventArgs e)
        {
            //Зберігаємо нове порожнє замовлення, формуючи рядок додавання
            this.SqlDataSource1.InsertCommand = "INSERT INTO [Dostavka] (Odpravitel, Poluchatel, DataO, Gorod, Transport,
            Obyom, Ves, Kolvo, DataP, Status, Price, Summa) VALUES ('" +
                2 + "', '" + 2 + "', '" + DateTime.Now + "', '" + 1 + "', '" + 1 + "', '" + 0 + "', '" + 0 + "', '" + 0 + "', '" + 0 + "', '" + DateTime.Now
            + "', '" +
                1 + "', '" + 1 + "', '" + 0 + "')";
            //Додаємо дані
            this.SqlDataSource1.Insert();
            //Оновлюємо дані
            this.GridView1.DataBind();
        }

        protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {
            //При збереженні оновлюємо значення замовлення у поточному рядку
            this.SqlDataSource1.UpdateCommand = "UPDATE [Dostavka] SET [Otpravitel] = @Otpravitel, [Poluchatel] =
            @Poluchatel, [DataO] = @DataO, " +
                "[Gorod] = @Gorod, [Transport] = @Transport, [Obyom] = @Obyom, [Ves] = @Ves, [Kolvo] = @Kolvo, [DataP]
            = @DataP, [Status] = @Status , " +

```

```

        "[Price] = @Price, [Summa] = @Summa WHERE [Kod] = @Kod";
    }
}
}

```

24.RouteForm.aspx

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Drawing;
using System.Drawing.Drawing2D;
using System.Drawing.Imaging;

namespace LCompany
{
    public partial class RouteForm : System.Web.UI.Page
    {
        int Razmer; //Кількість пунктів, включених у маршрут
        int MaxX, MaxY, MaxZ; // Змінні MaxX і MaxY призначені для зберігання даних про стовпчик
        // та рядку максимального елемента матриці. Змінна MaxZ
        // призначена зберігання значення максимального елемента матриці.
        int[,] RasTable; //Розрахункова таблиця
        int[,] IsxTable; //Вихідна таблиця

        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void Page_Init(object sender, EventArgs e)
        {

        }

        protected void Button2_Click(object sender, EventArgs e)
        {
            // Переходимо на головну сторінку співробітника
            this.Response.Redirect("OperatorMenu.aspx");
        }

        protected void Button4_Click(object sender, EventArgs e)
        {

```

```

DataSourceReload(SqlDataSource1);
//Заповнюємо базу та розрахункову таблицю
this.SqlDataSource1.FilterExpression = "Punkt = true";
GridView1.DataBind();
Razmer = GridView1.Rows.Count;
//Очищаємо зображення
Image2.ImageUrl = "";
//Таблицю заповнили – можна розраховувати
Button3.Enabled = true; //Включаємо кнопку розрахунку
Button4.Enabled = false; //Вимикаємо кнопку заповнення
}

private void Raschot()
{
    //Процедура розрахунку матриці методом Кларка-Райту - обчислюємо дуги (вектори) економій
    // Спочатку максимальне число дорівнює 0}
    MaxZ = 0;
    int k0 = 0;
    int k = 0;
    // Завжди починаємо з другого рядка
    for (int i = 1; i < Razmer; i++)
    {
        k0 = RasTable[0, i];

        // і з першого елемента
        for (int j = 1; j < Razmer; j++)
        {
            k = 0;
            //Далі, якщо елемент не дорівнює 0, тобто він або не на діагоналі або ще не "викреслено" то вважаємо
            його значення
            if (RasTable[j, i] > 0)
            {
                //за принципом "прямого кута" методу Кларка-Райту
                k = Math.Abs (k0 + RasTable [j, 0] - RasTable [j, i]);
                if (k >= MaxZ)
                {
                    //якщо знайдене значення >= MaxZ то запам'ятовуємо його
                    MaxZ = k;
                    //Одночасно запам'ятовуємо координати поточного максимального елемента
                    MaxX = j;
                    MaxY = i;
                }
            }
        }
    }
}

private int XYFind(int KL)
//Обчислюємо абсолютний базовий номер міста в залежності від обраного номера
{
    int Re = 0;

    for (int i = 0; i < Razmer; i++)
        if (KL == Convert.ToInt32(this.GridView1.Rows[i].Cells[1].Text))
        {

```

```

        Re = i;
        break;
    }
    else
    {
        Re = 0;
    }
    return Re;
}

private void Udalenie()
{
    // Видаляємо з робочої матриці, тобто прирівнюємо 0 всі значення зі стовпця та рядки поточного
    // максимального числа, крім першого пункту}
    for (int i = 1; i < Razmer - 1; i++) RasTable [MaxX, i] = 0; // стовпець
    for (int j = 1; j < Razmer; j++) RasTable [j, MaxY] = 0; //Рядок
    // А також видаляємо зворотний шлях, наприклад, якщо віддалений 3-4, то 4-3 вже немає сенсу}
    RasTable[MaxY, MaxX] = 0;
    RasTable[MaxX, MaxY] = 0;
}

private void DataSourceReload(SqlDataSource DataSource)
{
    //Динамічно оновлюємо джерела даних
    string SelectCommand = DataSource.SelectCommand;
    DataSource.SelectCommand = "";
    DataSource.SelectCommand = SelectCommand;
    string FilterExpression = DataSource.FilterExpression;
    DataSource.FilterExpression = "";
    DataSource.FilterExpression = DataSource.FilterExpression;
    //Оновлюємо пов'язані елементи
    DataSource.DataBind();
}

protected void Button3_Click(object sender, EventArgs e)
{
    //Процедура розрахунку шляху
    GridView2.Visible = true; //Показуємо перехресну таблицю для заповнення та вилучення даних
    Razmer = GridView1.Rows.Count; //Розміри маршруту
    if (Razmer < 4) return; //Якщо вибрано менше 4-х пунктів - менше немає сенсу для розрахунку - виходимо
    string ColList = "";
    GridView2.DataSourceID = "";
    this.SqlDataSource2.SelectCommand = "";
    DataSourceReload(SqlDataSource2);
    //Створимо список стовпців для перехресного запиту
    for (int i = 0; i < Razmer; i++)
    {
        ColList = ColList + "[" + GridView1.Rows[i].Cells[1].Text + "]";
        if (i < Razmer - 1) ColList = ColList + ",";
    }
    //Створюємо перехресний запит PIVOT

```

```

this.SqlDataSource2.SelectCommand = "SELECT * FROM Rastab PIVOT (SUM(Dist) FOR Kod2 IN (" + ColList +
")) pvt;";
DataSourceReload(SqlDataSource2);
//Призначаємо джерело запиту для GridView2
GridView2.DataSourceID = "SqlDataSource2";
//Оновлюємо дані
GridView2.DataBind();
//Створюємо розрахункову та вихідну таблиці
RasTable = new int [Razmer, Razmer];
IsxTable = new int [Razmer, Razmer];
//Переносимо всі дані в розрахункову та вихідну таблиці
for (int i = 0; i < Razmer; i++)
{
    for (int j = 0; j < Razmer; j++)
    {
        IsxTable[j, i] = Convert.ToInt32(Convert.ToDecimal(GridView2.Rows[i].Cells[j + 1].Text));
        RasTable[j, i] = IsxTable[j, i];
    }
}
GridView2.Visible = false; //Приховуємо перехресну таблицю
GridView3.Visible = true; //Показуємо список шляхів
//Очищаємо таблицю маршрутів від попередніх розрахунків
this.SqlDataSource3.DeleteCommand = "DELETE FROM [Route]";
this.SqlDataSource3.Delete();
DataSourceReload(SqlDataSource3);
//Створюємо таблицю оптимального шляху + 1 елемент повернення у вихідний пункт
int[,] RouteTable = new int[2, Razmer + 1];
RouteTable[0, 0] = 1; //Перший пункт
RouteTable[1, 0] = IsxTable[0, 0]; // Відстань - з вихідної таблиці
//Виконуємо розрахунок методу Кларка-Райту – Razmer – 1 раз, тобто N-1 число шляхів між N містами
for (int i = 0; i < Razmer - 1; i++)
{
    // Визначаємо максимальний елемент
    Raschot(); //Будуємо шлях
    //Для коректного відображення знаходимо справжні номери пунктів, що відповідають MaxY та MaxX
    int XMax = Convert.ToInt32(this.GridView1.Rows[MaxX].Cells[1].Text);
    int YMax = Convert.ToInt32(this.GridView1.Rows[MaxY].Cells[1].Text);
    // Записуємо до таблиці підсумкового шляху черговий пункт
    RouteTable[0, i + 1] = YMax;
    RouteTable[1, i + 1] = IsxTable[XYFind(RouteTable[0, i]), XYFind(YMax)];
    //Видаляємо вже непотрібні шляхи
    Udalenie();
}
// Записуємо до таблиці підсумкового шляху останній пункт
RouteTable[0, Razmer] = 1; //Номер пункту
RouteTable[1, Razmer] = IsxTable[XYFind(RouteTable[0, Razmer - 1]), 0]; // Відстань - з вихідної таблиці
int Dlina = 0; //Довжина шляху
//Вставляємо ланцюжки з першого пункту та інших пунктів туди й назад у таблицю маршрутів - звідки
виходитиме послідовність для малювання
for (int i = 0; i < Razmer + 1; i++)
{
    //Розраховуємо довжину шляху та заносимо їх у інформаційний напис
    Dlina = Dlina + RouteTable[1, i];
    // Записуємо черговий пункт
    this.SqlDataSource4.InsertCommand = "INSERT INTO Route (Punkt) VALUES (" + RouteTable[0, i] + ")";
}

```



```

        this.SqlDataSource4.Insert();
    }
    // Зберігаємо зміни
    DataSourceReload(SqlDataSource4);
    DataSourceReload(SqlDataSource5);
    GridView3.DataBind();
    //Заносимо довжину шляху в інформаційний напис
    this.Label1.Text = Dlina.ToString();
    Button3.Enabled = false; //Вимикаємо кнопку розрахунку
    Button5.Enabled = true; //Включаємо кнопку відтворення
}

protected void Button5_Click(object sender, EventArgs e)
{
    //Малюємо карту маршруту
    using (Bitmap PathImage = new Bitmap(1200, 600, System.Drawing.Imaging.PixelFormat.Format32bppArgb))
    {
        int X1, Y1, X2, Y2; //Змінні для координат
        using (Graphics Paths = Graphics.FromImage(PathImage))
        {
            //Встановлюємо тип лінії
            Pen PathsPen = new Pen(Color.Blue, 3); //Синій колір
            PathsPen.DashStyle = DashStyle.Solid; //Суцільна лінія
            PathsPen.EndCap = System.Drawing.Drawing2D.LineCap.ArrowAnchor; //В кінці стрілка
            //Встановлюємо атрибути тексту
            var TextFont = new Font("Calibri", 14, FontStyle.Underline); //Підкреслений текст
            var TextBrush = new SolidBrush(System.Drawing.Color.Gold); //Золотий колір
            Paths.TextRenderingHint = System.Drawing.Text.TextRenderingHint.SingleBitPerPixelGridFit;
            DataSourceReload(SqlDataSource6);
            //Тимчасово включаємо таблицю з результатами для читання даних
            GridView4.Visible = true;

            // Задаємо пропорції для масштабування на екрані - основне місто в центрі
            int X = Convert.ToInt32(this.Image2.Width.Value / 2);
            int Y = Convert.ToInt32(this.Image2.Height.Value / 2);
            double KX = 0.7; //Коефіцієнт по X
            double KY = 0.9; //Коефіцієнт по Y
            //Послідовно зчитуємо результати
            for (int i = 0; i < GridView4.Rows.Count - 1; i++)
            {
                //Отримуємо координати сусідніх пунктів
                X1 = Convert.ToInt32(X + Convert.ToInt32(GridView4.Rows[i].Cells[2].Text) * KX);
                Y1 = Convert.ToInt32(Y - Convert.ToInt32(GridView4.Rows[i].Cells[3].Text) * KY);
                X2 = Convert.ToInt32(X + Convert.ToInt32(GridView4.Rows[i + 1].Cells[2].Text) * KX);
                Y2 = Convert.ToInt32(Y - Convert.ToInt32(GridView4.Rows[i + 1].Cells[3].Text) * KY);
                //Якщо останній пункт, то малюємо пунктиром
                if (i == GridView4.Rows.Count - 2)
                {
                    PathsPen.DashStyle = DashStyle.DashDot; //Пунктирна лінія
                    //Повертаємось додому
                    Paths.DrawLine(PathsPen, X1, Y1, X, Y);
                }
                else Paths.DrawLine(PathsPen, X1, Y1, X2, Y2); //Інакше малюємо лінію між пунктами
                //Малюємо текст найменування пункту з номером крім останнього
            }
        }
    }
}

```

```

        if (i < GridView4.Rows.Count - 1) Paths.DrawString(Convert.ToString(i + 1) + ". " +
GridView4.Rows[i].Cells[1].Text, TextFont, TextBrush, X1, Y1 );
    }
    DataSourceReload(SqlDataSource7);
    //Послідовно друкуємо інші міста для повноти карти
    //Тимчасово включаємо таблицю з результатами для читання даних
    GridView5.Visible = true;
    //Встановлюємо атрибути тексту
    TextFont = new Font("Calibri", 12, FontStyle.Regular); //Звичайний текст дрібніший
    TextBrush = new SolidBrush (System.Drawing.Color.WhiteSmoke); //Світлосірий колір
    for (int i = 0; i < GridView5.Rows.Count; i++)
    {
        //Отримуємо координати інших пунктів
        X1 = Convert.ToInt32(X + Convert.ToInt32(GridView5.Rows[i].Cells[1].Text) * KX);
        Y1 = Convert.ToInt32(Y - Convert.ToInt32(GridView5.Rows[i].Cells[2].Text) * KY);
        //Малюємо текст найменування пункту
        Paths.DrawString("." + GridView5.Rows[i].Cells[0].Text, TextFont, TextBrush, X1, Y1);
    }
    Paths.Dispose();
}
//Вимикаємо таблицю з результатами для читання даних
GridView4.Visible = false;
//Вимикаємо таблицю з іншими містами
GridView5.Visible = false;
//Зберігаємо та завантажуюмо зображення
PathImage.Save(Server.MapPath("~/") + "/PathImage.jpg");
Image2.ImageUrl = "~/PathImage.jpg";
}
//Підсвічуємо робочу кнопку
Button5.Enabled = false; //Вимикаємо кнопку відтворення
}
}
}

```

ДОДАТОК 2

Вимоги державних санітарних правил і норм (ДСанПіН) під час роботи з комп'ютерною технікою.

Для запобігання несприятливому впливу на людину шкідливих факторів, що супроводжують роботи з відеодисплейними терміналами (далі – ВДТ) та персональними електронно-обчислювальними машинами (далі – ПЕОМ) визначається низка вимог.

Висота робочої поверхні столу для дорослих користувачів має регулюватися в межах 680 - 800 мм; за відсутності такої можливості висота робочої поверхні столу має становити 725 мм.

Модульними розмірами робочої поверхні столу для ВДТ і ПЕОМ, на підставі яких повинні розраховуватися конструктивні розміри, слід вважати: ширину 800, 1000, 1200 і 1400 мм, глибину 800 і 1000 мм при нерегульованій його висоті, що дорівнює 725 мм.

Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600 мм, шириною - не менше ніж 500 мм, глибиною на рівні колін - не менше ніж 450 мм і на рівні витягнутих ніг - не менше ніж 650 мм.

Робочий стілець (крісло) має бути підйомно - поворотним і регульованим по висоті та кутам нахилу сидіння та спинки, а також - відстані спинки від переднього краю сидіння.

Конструкція його має забезпечувати:

- ширину та глибину поверхні сидіння не менше 400 мм;
- поверхня сидіння із закругленим переднім краєм;
- Регулювання висоти поверхні сидіння в межах 400 - 550 мм і кутів нахилу вперед до 15 град. та назад до 5 град.;

- Висоту опорної поверхні спинки 300 ± 20 мм, ширину - не менше 380 мм і радіус кривизни горизонтальної площини - 400 мм;
- кут нахилу спинки у вертикальній площині в межах 0° - 30° градусів;
- регулювання відстані спинки від переднього краю сидіння в межах 260 – 400 мм;
- стаціонарні або знімні підлокітники довжиною не менше 250 мм та шириною 50 - 70 мм;
- регулювання підлокітників за висотою над сидінням у межах 230 ± 30 мм та внутрішньої відстані між підлокітниками в межах 350 - 500 мм.

Робоче місце повинно бути обладнане підставкою для ніг, що має ширину не менше ніж 300 мм, глибину не менше ніж 400 мм, регулювання по висоті в межах до 150 мм та по куту нахилу опорної поверхні підставки до 20° градусів. Поверхня підставки повинна бути рифлена і мати по передньому краю бортик заввишки 10 мм.

Робоче місце з ВДТ і ПЕОМ має бути оснащено попітром для документів, що легко переміщається.

При організації робочих місць для роботи на технологічному обладнанні, до складу яких входять ВДТ або ПЕОМ (верстати з програмним управлінням, роботизовані та технологічні комплекси, гнучке автоматизоване виробництво, диспетчерські пульти управління та ін.), слід передбачати:

- простір по глибині щонайменше 850 мм з урахуванням виступаючих частин устаткування перебування людини - оператора;
- простір для стоп глибиною та висотою не менше 150 мм та шириною не менше 530 мм;
- Розташування пристроїв введення - виведення інформації, що забезпечує оптимальну видимість екрана;

- легку досяжність органів ручного управління в зоні моторного поля: за висотою – 900 – 1300 мм, по глибині – 400 – 500 мм;

- розташування екрану ВДТ або ПЕОМ у місці робочої зони, що забезпечує зручність зорового спостереження у вертикальній площині під кутом ± 30 градусів від нормальної лінії погляду оператора, а також зручність використання ВДТ або ПЕОМ (введення - виведення інформації при коригуванні основних параметрів технологічного процесу, налагодження програм та ін.) одночасно з виконанням основних виробничих операцій (спостереження за зоною обробки на верстаті з програмним управлінням, під час обслуговування роботизованого технологічного комплексу та ін.));

- Можливість повороту екрану ВДТ або ПЕОМ навколо горизонтальної та вертикальної осей.

Клавіатуру слід розташовувати на поверхні столу на відстані 100 - 300 мм від краю, зверненого до користувача, або на спеціальній робочій поверхні, що регулюється по висоті, відокремленою від основної стільниці.

Приміщення з ВДТ та ПЕОМ повинні мати природне та штучне освітлення.

Природне освітлення має здійснюватися через світлопройоми, орієнтовані переважно на північ і північний схід і забезпечувати коефіцієнт природного освітлення (КПО) не нижче 1,2% у зонах зі стійким сніговим покривом і не нижче 1,5% на решті території. Зазначені значення КПО нормуються для будівель, розташованих у поточному світловому та кліматичному поясі.

Розрахунок КПО для інших поясів світлового клімату проводиться за загальноприйнятою методикою згідно з БНіП "Природне та штучне освітлення".

Розміщення робочих місць з ВДТ та ПЕОМ для дорослих користувачів у підвальних приміщеннях не допускається. Розміщення робочих місць з ВДТ та

ПЕОМ у всіх навчальних закладах та дошкільних закладах не допускається у цокольних та підвальних приміщеннях.

У випадках виробничої необхідності експлуатація ВДТ та ПЕОМ у приміщеннях без природного освітлення може проводитись лише за погодженням з органами та установами Державного санітарно-епідеміологічного нагляду.

Площа на одне робоче місце з ВДТ або ПЕОМ для дорослих користувачів має становити щонайменше 6,0 кв. м, а обсяг – не менше 20,0 куб. м.

Площа на одне робоче місце з ВДТ та ПЕОМ у всіх навчальних та дошкільних закладах має бути не менше 6,0 кв. м, а обсяг – не менше 24,0 куб. м.

При будівництві нових та реконструкції діючих середніх, середніх спеціальних та вищих навчальних закладів приміщення для ВДТ та ПЕОМ слід проектувати висотою (від підлоги до стелі) не менше 4,0 м.

При вході до навчального приміщення з ВДТ та ПЕОМ у середніх та вищих навчальних закладах слід передбачити вбудовані або пристінні шафи (полиці) для зберігання портфелів, сумок учнів та студентів.

Виробничі приміщення, в яких для роботи використовуються переважно ВДТ та ПЕОМ (диспетчерські, операторські, розрахункові та ін.), та навчальні приміщення (аудиторії обчислювальної техніки, дисплейні класи, кабінети та ін.) не повинні межувати з приміщеннями, в яких рівні шуму та вібрації перевищують нормовані значення (механічні цехи, майстерні, гімнастичні зали тощо).

Звукоізоляція огорожувальних конструкцій приміщень з ВДТ та ПЕОМ повинна відповідати гігієнічним вимогам та забезпечувати нормовані параметри шуму відповідно до вимог розділу 6 цих Санітарних правил.

Приміщення з ВДТ та ПЕОМ повинні обладнатися системами опалення, кондиціонування повітря або ефективною вентиляцією припливно-витяжної. Розрахунок повітрообміну слід проводити за тепло надлишком від машин, людей,

сонячної радіації та штучного освітлення. Нормовані параметри мікроклімату, іонного складу повітря, вміст шкідливих речовин, у ньому повинні відповідати вимогам розділу 5 цих Санітарних правил.

Навчальні кабінети обчислювальної техніки або дисплейні аудиторії (класи) повинні мати суміжне приміщення – лаборантську площу не менше ніж 18,0 кв. м із двома входами: до навчального приміщення та на сходовий майданчик чи рекреацію.

У дитячих дошкільних закладах суміжно із приміщенням, де встановлені ПЕОМ чи ВДТ, має розташовуватися ігровий зал площею не менше 24 кв. м.

Для внутрішнього оздоблення інтер'єру приміщень з ВДТ і ПЕОМ повинні використовуватися дифузно - відбивні матеріали з коефіцієнтом відбиття для стелі - 0,7 - 0,8; для стін – 0,5 – 0,6; для підлоги – 0,3 – 0,5.

Полімерні матеріали, що використовуються для внутрішнього оздоблення інтер'єру приміщень з ВДТ та ПЕОМ, повинні бути дозволені для застосування органами та установами Державного санітарно-епідеміологічного нагляду.

У дошкільних та всіх навчальних закладах, включаючи вузи, забороняється для оздоблення внутрішнього інтер'єру приміщень з ВДТ та ПЕОМ застосовувати полімерні матеріали (деревностружкові плити, шаруватий паперовий пластик, синтетичні килимові покриття тощо), що виділяють у повітря шкідливі хімічні речовини.

Поверхня підлоги в приміщеннях експлуатації ВДТ і ПЕОМ повинна бути рівною, без вибоїн, неслизькою, зручною для очищення та вологого прибирання, мати антистатичні властивості.

У виробничих приміщеннях, у яких робота на ВДТ та ПЕОМ є допоміжною, температура, відносна вологість та швидкість руху повітря на робочих місцях

повинні відповідати діючим санітарним нормам мікроклімату виробничих приміщень.

У виробничих приміщеннях, в яких робота на ВДТ та ПЕОМ є основною (диспетчерські, операторські, розрахункові, кабіни та пости управління, зали обчислювальної техніки та ін.) повинні забезпечуватися оптимальні параметри мікроклімату.

У приміщеннях з ВДТ та ПЕОМ у дошкільних, середніх спеціальних та вищих навчальних закладах мають забезпечуватись оптимальні параметри мікроклімату.

Для підвищення вологості повітря в приміщеннях з ВДТ та ПЕОМ слід застосовувати зволожувачі повітря, що заправляються щодня дистильованою або кип'яченою питною водою.

Приміщення з ВДТ та ПЕОМ перед початком та після кожної академічної години навчальних занять, до та після кожного заняття у дошкільній установі мають бути провітрені, що забезпечує покращення якісного складу повітря, у тому числі й аероіонний режим.

Рівні позитивних та негативних аероіонів у повітрі приміщень з ВДТ та ПЕОМ повинні відповідати нормам.

Вміст шкідливих хімічних речовин у повітрі виробничих приміщень, в яких робота на ВДТ та ПЕОМ є допоміжною, не повинна перевищувати "гранично допустимих концентрацій шкідливих речовин у повітрі робочої зони".

Зміст шкідливих хімічних речовин у виробничих приміщеннях, робота на ВДТ та ПЕОМ у яких є основною (диспетчерські, операторські, розрахункові, кабіни та пости управління, зали обчислювальної техніки та ін.), не має перевищувати "гранично допустимих концентрацій забруднюючих речовин в атмосферному повітрі населених місць"

Вміст шкідливих хімічних речовин у повітрі приміщень використання ВДТ та ПЕОМ у дошкільних та всіх навчальних закладах, включаючи вищі, не повинен перевищувати середньодобових концентрацій для атмосферного повітря.

Забороняється проводити ремонт ВДТ та ПЕОМ безпосередньо у робочих, навчальних та дошкільних приміщеннях.

Розташування робочих місць по відношенню до світлових отворів. Штучне освітлення у приміщеннях експлуатації ВДТ та ПЕОМ. має здійснюватися системою загального рівномірного висвітлення. У виробничих та адміністративно-громадських приміщеннях, у випадках переважної роботи з документами, допускається застосування системи комбінованого освітлення (до загального освітлення додатково встановлюються світильники місцевого освітлення, призначені для освітлення зони розташування документів).

Освітленість на поверхні столу в зоні розміщення робочого документа має бути 300 – 500 лк. Дозволяється встановлення світильників місцевого освітлення для підсвічування документів. Місцеве освітлення не повинно створювати відблисків на поверхні екрана та збільшувати освітленість екрану більше 300 лк.

Слід обмежувати пряму блискучість від джерел освітлення, при цьому яскравість поверхонь, що світяться (вікна, світильники та ін.), що знаходяться в полі зору, повинна бути не більше 200 кд/кв. м.

Слід обмежувати відбиту блискітку на робочих поверхнях (екран, стіл, клавіатура та ін.) за рахунок правильного вибору типів світильників та розташування робочих місць по відношенню до джерел природного та штучного освітлення, при цьому яскравість відблисків на екрані ВДТ та ПЕОМ не повинна перевищувати 40 кд / кв. м та яскравість стелі при застосуванні системи відбитого освітлення не повинна перевищувати 200 кд/кв. м.

Показник засліпленості для джерел загального штучного освітлення у виробничих приміщеннях має бути не більше 20, показник дискомфорту в адміністративно – громадських приміщеннях – не більше 40, у дошкільних та навчальних приміщеннях – не більше 25.

Слід обмежувати нерівномірність розподілу яскравості у полі зору користувача ВДТ та ПЕОМ, при цьому співвідношення яскравості між робочими поверхнями не повинно перевищувати 3:1 - 5:1, а між робочими поверхнями та поверхнями стін та обладнання - 10:1.

Як джерела світла при штучному освітленні повинні застосовуватися переважно люмінесцентні лампи типу ЛБ. При влаштуванні відбитого освітлення у виробничих та адміністративно - громадських приміщеннях допускається застосування металогалогенових ламп потужністю до 250 Вт. Допускається застосування ламп розжарювання у світильниках місцевого освітлення.

Загальне освітлення слід виконувати у вигляді суцільних або переривчастих ліній світильників, розташованих збоку від робочих місць, паралельно лінії зору користувача при рядному розташуванні ВДТ та ПЕОМ. При периметральному розташуванні комп'ютерів лінії світильників повинні розташовуватися локалізовано над робочим столом ближче до переднього краю, зверненому до оператора.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50 до 90 градусів з вертикаллю в поздовжній та поперечній площинах повинна становити не більше ніж 200 кд/кв. м, захисний кут світильників має бути не менше 40 градусів.

Світильники місцевого освітлення повинні мати відбивач, що не просвічує, із захисним кутом не менше 40 градусів.

Коефіцієнт запасу (Кз) для освітлювальних установок загального освітлення повинен прийматися рівним 1,4.

Коефіцієнт пульсації не повинен перевищувати 5%, що повинно забезпечуватись застосуванням газорозрядних ламп у світильниках загального та місцевого освітлення з високочастотними пускорегулюючими апаратами (ВЧ ПРА) для будь-яких типів світильників. За відсутності світильників з ВЧ ПРА лампи багатолампових світильників або поруч розташовані світильники загального освітлення слід включати різні фази трифазної мережі.

Для забезпечення нормованих значень освітленості в приміщеннях використання ВДТ і ПЕОМ слід проводити чищення шибок віконних рам і світильників не рідше двох разів на рік і проводити своєчасну заміну ламп, що перегоріли.

Для даного виробничого приміщення з комп'ютерами: довжиною 6 м, шириною 5 м, висотою 3 м необхідно мати 6 світильників по 2 лампи в кожному споживана потужність якої становить 960Вт.

У виробничих приміщеннях, в яких робота на ВДТ та ПЕОМ є допоміжною, рівні шуму на робочих місцях не повинні перевищувати значень, встановлених для даних видів робіт "Санітарними нормами допустимих рівнів шуму на робочих місцях"

При виконанні основної роботи на ВДТ та ПЕОМ (диспетчерські, операторські, розрахункові кабінки та пости управління, зали обчислювальної техніки та ін.), у всіх навчальних та дошкільних приміщеннях з ВДТ та ПЕОМ рівень шуму на робочому місці не повинен перевищувати 50 дБА.

У приміщеннях, де працюють інженерно-технічні працівники, які здійснюють лабораторний, аналітичний чи вимірювальний контроль, рівень шуму не повинен перевищувати 60 дБА.

У приміщеннях операторів ЕОМ (без дисплеїв) рівень шуму має перевищувати 65 дБА.

На робочих місцях у приміщеннях для розміщення шумних агрегатів обчислювальних машин (АЦДП, принтери тощо) рівень шуму не повинен перевищувати 75 дБа.

При виконанні робіт з ВДТ та ПЕОМ у виробничих приміщеннях рівень вібрації не повинен перевищувати допустимих значень відповідно до "Санітарних норм вібрації робочих місць" (категорія 3, тип "в").

У виробничих приміщеннях, у яких робота з ВДТ та ПЕОМ є основною, а також у всіх навчальних та дошкільних приміщеннях з ВДТ та ПЕОМ вібрація на робочих місцях не повинна перевищувати допустимих норм вібрації.

Шумливе обладнання (АЦДП, принтери тощо), рівні шуму якого перевищують нормовані, має знаходитися поза приміщенням з ВДТ та ПЕОМ.

Знизити рівень шуму в приміщеннях з ВДТ та ПЕОМ можна використанням звукопоглинаючих матеріалів з максимальними коефіцієнтами звукопоглинання в області частот 63 - 8000 Гц для обробки приміщень (дозволений органами та установами Держсанепіднагляду України), підтверджених спеціальними акустичними розрахунками.

Додатковим звукопоглинанням служать однотонні завіси із щільної тканини, що гармонують із забарвленням стін та підвішені у складку на відстані 15 - 20 см від огорожі. Ширина завіси повинна бути в 2 рази більша за ширину вікна.

ДОДАТОК 3

Правила техніки безпеки: Інструкція з охорони праці під час роботи на відеодисплейних терміналах (ВДТ) та персональних електронно-обчислювальних машинах (ПЕОМ); Вправи для очей під час роботи за комп'ютером.

Робочі місця з ВДТ і ПЕОМ по відношенню до світлових прорізів повинні розташовуватися так, щоб природне світло падало збоку, переважно зліва.

Схеми розміщення робочих місць з ВДТ та ПЕОМ повинні враховувати відстані між робочими столами з відеомоніторами (в напрямку тилу поверхні одного відеомонітора та екрана іншого відеомонітора), яка повинна бути не менше 2,0 м, а відстань між бічними поверхнями відеомоніторів – не менше 1, 2м.

Робочі місця з ВДТ та ПЕОМ у залах електронно-обчислювальних машин або у приміщеннях з джерелами шкідливих виробничих факторів повинні розміщуватися в ізольованих кабінах з організованим повітрообміном.

Віконні отвори в приміщеннях використання ВДТ та ПЕОМ повинні бути обладнані регульованими пристроями типу жалюзі, завісами, зовнішніми козирками та ін.

Робочі місця з ВДТ і ПЕОМ у виконанні творчої роботи, потребує значного розумового напруги чи високої концентрації уваги, слід ізолювати друг від друга перегородками висотою 1,5 - 2,0 м.

Шафи, сейфи, стелажі для зберігання дисків, дискет, комплектуючих деталей, запасних блоків ВДТ та ПЕОМ, інструментів слід розташовувати у підсобних приміщеннях, для навчальних закладів – у лаборантських.

За відсутності підсобних приміщень або лаборантських допускається розміщення шаф, сейфів та стелажів у приміщеннях безпосереднього використання

ВДТ та ПЕОМ за дотримання вимог до площі приміщень та вимог, викладених у цьому розділі.

У підсобних приміщеннях або в лаборантських повинні розміщуватися робочий стіл та радіомонтажний стіл, обладнаний місцевим відсмоктуванням на телескопічному повітроводі з шарнірним з'єднанням, що дозволяє встановлювати повітроприймач у потрібному положенні, з вихідною швидкістю 5 - 6 м/с у всмоктувальній площині.

При конструюванні обладнання та організації робочого місця користувача ВДТ та ПЕОМ слід забезпечити відповідність конструкції всіх елементів робочого місця та їх взаємного розташування ергономічним вимогам з урахуванням характеру виконуваної користувачем діяльності, комплексності технічних засобів, форм організації праці та основного робочого положення користувача.

Конструкція робочого столу повинна забезпечувати оптимальне розміщення на робочій поверхні устаткування, що використовується з урахуванням його кількості та конструктивних особливостей (розмір ВДТ та ПЕОМ, клавіатури, пюпітра та ін.), характеру виконуваної роботи. При цьому допускається використання робочих столів різних конструкцій, що відповідають сучасним вимогам ергономіки.

Конструкція робочого крісла повинна забезпечувати підтримку раціональної робочої пози під час роботи на ВДТ і ПЕОМ, дозволяти змінювати позу з метою зниження статичної напруги м'язів шийно - плечової області та спини для запобігання розвитку втоми.

Тип робочого стільця (крісла) повинен вибиратися залежно від характеру та тривалості роботи з ВДТ та ПЕОМ з урахуванням зростання користувача.

Робочий стілець (крісло) повинен бути підйомно - поворотним і регульованим по висоті та кутам нахилу сидіння та спинки, а також відстані спинки від переднього

краю сидіння, при цьому регулювання кожного параметра має бути незалежним, легко здійснюваним і мати надійну фіксацію.

Поверхня сидіння, спинки та інших елементів стільця (крісла) повинна бути напівм'якою, з нековзним, неелектризується і повітропроникним покриттям, що забезпечує легке очищення від забруднень.

Екран відеомонітора повинен знаходитись від очей користувача на оптимальній відстані 600 - 700 мм, але не ближче 500 мм з урахуванням розмірів алфавітно-цифрових знаків та символів.

У приміщеннях з ВДТ та ПЕОМ щодня повинно проводитися вологе прибирання.

Приміщення з ВДТ та ПЕОМ повинні бути оснащені аптечкою першої допомоги та вуглекислотними вогнегасниками.

Режими праці та відпочинку при роботі з ПЕОМ та ВДТ повинні організовуватися залежно від виду та категорії трудової діяльності.

Види трудової діяльності поділяються на 3 групи: група А - робота з зчитування інформації з екрану ВДТ чи ПЕОМ із попереднім запитом; група Б - робота із введення інформації;

група В - творча робота у режимі діалогу з ЕОМ. При виконанні протягом робочої зміни робіт, що належать до різних видів трудової діяльності, за основну роботу з ПЕОМ та ВДТ слід приймати таку, що займає не менше 50% часу протягом робочої зміни або робочого дня.

Для видів трудової діяльності встановлюється 3 категорії тяжкості та напруженості роботи з ВДТ та ПЕОМ, які визначаються: для групи А - за сумарною кількістю зчитуваних знаків за робочу зміну, але не більше 60000 знаків за зміну; для групи Б - за сумарною кількістю зчитуваних або введених знаків за робочу зміну, але не більше 40 000 знаків за зміну; для групи В - за сумарним часом

безпосередньої роботи з ВДТ та ПЕОМ за робочу зміну, але не більше 6 годин за зміну.

Для викладачів вищих та середніх спеціальних навчальних закладів, вчителів загальноосвітніх шкіл встановлюється тривалість роботи у дисплейних класах та кабінетах інформатики та обчислювальної техніки не більше 4 годин на день.

Для інженерів, які обслуговують навчальний процес у кабінетах (аудиторіях) з ВДТ та ПЕОМ, тривалість роботи не повинна перевищувати 6 годин на день.

Тривалість обідньої перерви визначається чинним законодавством про працю та Правилами внутрішнього трудового розпорядку підприємства (організації, установи).

Для забезпечення оптимальної працездатності та збереження здоров'я професійних користувачів протягом робочої зміни повинні встановлюватись регламентовані перерви.

Час регламентованих перерв протягом робочої зміни слід встановлювати залежно від її тривалості, виду та категорії трудової діяльності.

Тривалість безперервної роботи з ВДТ без регламентованої перерви не повинна перевищувати 2 години.

При роботі з ВДТ та ПЕОМ у нічну зміну (з 22 до 6 годин), незалежно від категорії та виду трудової діяльності, тривалість регламентованих перерв має збільшуватися на 60 хвилин.

При 8-годинній робочій зміні та роботі на ВДТ та ПЕОМ регламентовані перерви слід встановлювати:

- для I категорії робіт - через 2 години від початку робочої зміни та через 2 години після обідньої перерви тривалістю 15 хвилин кожен;
- для II категорії робіт - через 2 години від початку робочої зміни та через 1.5 - 2 години після обідньої перерви тривалістю 15 хвилин кожен або тривалістю 10

хвилин через кожну годину роботи; - для III категорії робіт - через 1.5 - 2 години від початку робочої зміни та через 1.5 - 2 години після обідньої перерви тривалістю 20 хвилин кожен або тривалістю 15 хвилин через кожну годину роботи.

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин роботи аналогічно до перерв при 8-годинній робочій зміні, а протягом останніх 4 годин роботи, незалежно від категорії та виду робіт, кожну годину тривалістю 15 хвилин.

Під час регламентованих перерв з метою зниження нервово – емоційної напруги, втоми зорового аналізатора, усунення впливу гіподинамії та гіпокінезії, запобігання розвитку познотонічної втоми доцільно виконувати комплекси вправ.

З метою зменшення негативного впливу монотонії доцільно застосовувати чергування операцій осмисленого тексту та числових даних (зміна змісту робіт), чергування редагування текстів та введення даних (зміна змісту роботи).

У випадках виникнення у працюючих з ВДТ та ПЕОМ зорового дискомфорту та інших несприятливих суб'єктивних відчуттів, незважаючи на дотримання санітарно-гігієнічних, ергономічних вимог, режимів праці та відпочинку слід застосовувати індивідуальний підхід в обмеженні часу робіт з ВДТ та ПЕОМ, корекцію тривалості перерв для відпочинку проводити зміну діяльності на іншу, не пов'язану з використанням ВДТ та ПЕОМ.

Працюючим на ВДТ та ПЕОМ з високим рівнем напруженості під час регламентованих перерв та наприкінці робочого дня показано психологічне розвантаження у спеціально обладнаних приміщеннях (кімната психологічного розвантаження).

Спеціальна гімнастика для очей на комп'ютері допоможе розслабитися та зберегти зір.

Щоб зменшити негативний вплив комп'ютера на очі та зберегти зір, необхідно дотримуватись простих правил, які полягають у наступному:

- відстань до монітора має становити щонайменше 60 сантиметрів. Якщо ви погано бачите інформацію на екрані, слід поміняти розмір шрифту;
- монітор необхідно розташовувати перед собою. Він не повинен стояти навскіс, інакше для перегляду зображення на ньому Вам доведеться постійно повертати голову. А це може призвести до втоми очей;
- необхідно встановити монітор так, щоб кут падіння погляду на нього дорівнював приблизно 15 градусів. Тобто верхня кромка екрана повинна бути трохи нижче за рівень очей;
- періодично протирайте монітор, не чіпайте його руками. Для чищення можна скористатися сухими безворсовими серветками і спеціальною рідиною;
- зверніть увагу на освітлення у робочому приміщенні. Воно не повинно бути надто яскравим або тьмяним. Віддайте перевагу приглушеному розсіяному світлу. Не працюйте за комп'ютером у темряві;
- встановіть мінімальний рівень яскравості екрана. Не слід збільшувати контрастність зображення. Для обробки тексту на світлому фоні можна пригасити яскравість монітора, а при перегляді відео – зробити яскравішим;
- намагайтеся уникати тривалої роботи з сайтами з невдалою гамою кольорів. Щоб розглянути маленькі жовті літери на білому тлі, доведеться сильно напружити очі, це велике навантаження для них;
- при роботі за комп'ютером слід частіше моргати. Так Ви забезпечите очам постійне зволоження і запобігти висиханню рогівки, що може призвести до подразнення;

- вважається, що доросла людина може безперервно проводити за комп'ютером трохи більше 2 годин. При цьому в день рекомендується відводити не більше 6 годин.

Стандартний комплекс вправ для очей під час роботи за комп'ютером:

1. Закрийте очі і сильно напружте м'язи очей. Зачекайте близько 4 секунд. Розплющте очі і розслабтеся. Подивіться у вікно вдалину близько 6 секунд. Повторіть вправу 4-5 разів.
2. Подивіться на перенісся і затримайте очі приблизно на 4 секунди. Потім переведіть погляд на краєвид за вікном та дивіться туди близько 6 секунд. Виконайте вправу 4-5 разів.
3. Подивіться вліво, не повертаючи голову. Зафіксуйте очі у цьому положенні приблизно на 4 секунди. Повторіть цю вправу, тільки дивлячись ліворуч, вниз та вгору. Необхідно виконати це коло 3-4 рази.
4. Виконайте повороти очима в наступних напрямках: ліворуч, вниз, праворуч, вгору, потім прямо в далечінь у вікно. Потім праворуч, вниз, ліворуч, вгору, а далі прямо в далечінь у вікно. Виконайте всі дії ще 3-4 рази.
5. Моргніть максимально швидко, відрахувавши до 10, потім закрийте очі на кілька секунд. Тепер ще раз поморгайте протягом хвилини. Знову закрийте очі на 2-3 секунди. Відкрийте їх і подивіться у далечінь у вікно. Повторіть вправу 2-3 рази.
6. Розгляньте уважно будь-який добре видимий об'єкт (гілку, пташку, аркуш тощо) протягом 30 секунд. Потім переведіть очі на найвіддаленіший предмет. Це може бути будинок, автомобіль, дерево. Дивіться на нього протягом 30 секунд. Потім поверніть погляд на об'єкт. Повторіть таку вправу 6 разів.

Після закінчення зарядки даємо очам розслабитися. Можна просто закрити їх на п'ять хвилин. Не опускайте голову вперед. Виконання подібних вправ для очей при роботі за комп'ютером сприяє їхньому розслабленню та тренуванню.